Label Aggregation for Crowdsourced Triplet Similarity Comparisons

Jiyi Li[1], Lucas Ryo Endo[2], and Hisashi Kashima[2]

[1]Universiyt of Yamanashi, jyli@yamanashi.ac.jp
[2]Kyoto University, {lucas, kashima}@ml.ist.i.kyoto-u.ac.jp

Abstract. Organizing objects such as human ideas, opinions, and designs based on their similarity relationships is an important first step in data exploration and decision making. Those similarity comparisons are often cast as triplet comparisons asking which of two given objects is more similar to another given object, because humans are better at this type of relative judgments than pairwise similarity comparisons which ask for absolute judgments, especially in sensory domains. Crowdsourcing is an effective way to collect such human judgments easily and on a large scale; however, there is a large variation in abilities among workers and the difficulties of evaluating the target objects. How to aggregate the labels of crowdsourced triplet similarity comparisons for estimating similarity relations of all objects when there are only a smaller number of labels remains a challenge. In this work, we construct two novel real datasets for investigating this research topic. For label aggregation approach, we propose a family of models to learn the object embeddings from crowdsourced triplet similarity comparisons by incorporating worker abilities and object difficulties. Because of the diverse properties of real datasets, we automatically search for the optimal model from all variants of the proposed model. The experimental results verified the effectiveness of our approach. We also investigated how the data properties and model options influence the performance.

Keywords: Crowdsourcing, Label Aggregation, Triplet Similarity Comparison

1 Introduction

Organizing various objects such as human ideas, opinions, and designs based on their similarity relationships is an important first step in data exploration and decision making. Crowdsourcing is widely used as an inexpensive and rapid way to collect data for pairwise similarity comparisons of objects (e.g., [4]), especially when feature representations or similarities of the objects are not readily available. Pairwise similarities among objects are usually evaluated in terms of binary judgments indicating whether or not two objects are similar to each other. These pairwise similarity comparisons are aggregated to organize the objects into groups to elucidate the landscape of these objects for decision-making. However, there are at least two problems when using these crowdsourced pairwise

similarity labels. First, it is often difficult for humans to make absolute judgments, especially when the judgments are subjective. Second, there are large differences in the abilities and label numbers among workers and difficulties among objects.

For the first problem, the threshold for distinguishing similarity and dissimilarity is difficult to determine. For some object pairs, people can easily judge whether they are similar, while for some object pairs, it can be difficult to provide absolute judgments. For such cases, judgments based on relative comparisons are more human-friendly. For example, a triplet similarity comparison can be described as "object a is more similar to object b than to object c" [11,3,13], and a quadruplet similarity comparison can be described as "objects a and b are more similar than objects c and d." [1,12]. Such relative comparisons can assist a model to better estimate the similarities among objects. In this paper, we focus on triplet similarity comparisons. One of the obstacles in conducting relative comparisons is that the total number of combinations of objects is huge. Given n objects, the number of triplets is $\mathcal{O}(n^3)$. Therefore, a worker cannot label all triplets and can only evaluate a small subset. We need a method that can aggregate a small number of labels and estimate the similarity relations of all objects. To investigate this research topic and the label aggregation methods, we need datasets that contain crowdsourced triplet similarity comparison labels. Because there were no existing datasets available to the best of our knowledge, we created two novel real datasets using a crowdsourcing platform.

For the label aggregation methods, in the early years, one type of solutions is to learn the similarity matrix of objects, i.e., multi-dimensional scaling [11]. After that, another type of solutions is proposed which first learns the objects embeddings from labeled triplets, i.e., stochastic triplet embeddings [3,13], based on Gaussian kernel (STE) or Student-t kernel (tSTE), and then estimate the object similarities based on the embeddings. However, these existing approaches were not proposed for addressing the second problem. On the topic of label aggregation for *categorical labels* [14] and *pairwise preference comparison labels* [2] in the crowdsourcing context, researchers always incorporate worker abilities and/or object difficulties as the crucial factors in constructing the probabilistic models. In this paper, we propose a family of stochastic triplet embedding models incorporating worker abilities and object difficulties to learn the object embeddings from *triplet similarity comparison labels*, namely, Crowdsourced (t-distributed) Stochastic Triplet Embeddings (Crowd-(t)STE). The performances of the variants of the proposed model depend on the diverse properties of the real datasets; thus we also automatically search the optimal model with the best validation performance from all variants of our model.

We conducted experiments on the created datasets with diverse settings to verify our approaches. We also investigated how the data properties and model options influence the performance. The contributions of this paper are as follows:

1. We created two novel real datasets with large-scale crowd labels available that can be used for the research on label aggregation from crowdsourced stochastic triplet comparisons for estimating similarity relations of all objects.

2. We propose an approach to solve the problems of stochastic triplet embedding in the crowdsourcing context.
3. We propose a family of models by incorporating the worker ability and object difficulty by constructing diverse similarity kernels and loss objectives. There are 10 variants in total and we automatically search the optimal model.

2 Our Approach

2.1 Notations

We denote the set of objects by $\mathcal{O} = \{o_i\}_{i=1}^n$. We assume that no feature representations of the objects are available. Our goal is to estimate the similarity relations among all objects. For this purpose, we learn the representations $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$ of the objects in a d-dimensional latent feature space, i.e., $\mathbf{x}_i \in \mathbb{R}^d$, by utilizing the triplet similarity comparisons of objects. We use crowdsourcing to collect the triplet similarity comparison labels. We denote the set of crowd workers by $\mathcal{A} = \{a_l\}_{l=1}^m$.

For three given objects o_i, o_j, and o_k in $\mathcal{O}$, the triplet similarity comparison we consider is a type of questions that ask crowd workers to annotate the relations of pairwise similarities among them. More specifically, we ask a question that is "which object of o_j or o_k is more similar to the anchor object o_i?"; the candidate answers are either of "o_j" or "o_k". If a worker a_l annotates that o_i and o_j are more similar, then the triplet similarity label is $y_{ijk}^l = 1$; otherwise $y_{ijk}^l = 0$. The set of triplet similarity labels is defined as $\mathcal{Y} = \{y_{ijk}^l\}_{i,j,k,l}$ for the set of triplets $\mathcal{T} = \{(o_i, o_j, o_k)\}_{i,j,k}$. The set of labels given by worker a_l is denoted by$\mathcal{Y}^l$; the label set of a triplet is defined as $\mathcal{Y}_{ijk}$.

2.2 Problem Definition

Since the total number of object triplets is cubic in the number of objects, it costs too much budget and time to collect the labels for all of the triplets. In addition, due to the diverse ability and diligence of crowd workers, the collected labels are more likely to be noisy. We thus must collect multiple labels for a triplet and aggregate them to obtain more reliable labels, which further increases the total number of collected labels. These facts motivate us to estimate more accurate similarities of all objects based on a smaller number of similarity comparison labels. We provide large number of labels in the proposed datasets to construct ground truth for evaluation, while the problem setting is using only a small subset of our dataset in the training stage, i.e., only using a subset of triplets $\mathcal{T}_t \subset \mathcal{T}$ and only using a subset of labels $\mathcal{Y}_{t,ijk} \subset \mathcal{Y}_{ijk}$ for each triplet. The problem setting can be summarized as follows.

INPUTS: A set of objects $\mathcal{O}$, a set of crowd workers $\mathcal{A}$, and a subset of triplet similarity comparison labels $\mathcal{Y}_t \subset \mathcal{Y}$, with the subset $\mathcal{Y}_{t,ijk} \subset \mathcal{Y}_{ijk}$ for each triplet, for the subset of triplets $\mathcal{T}_t \subset \mathcal{T}$.

OUTPUTS: The object representations $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$. In addition, by some variants of the proposed model, we can also obtain the estimated object difficulty

$\mathbf{H} = \{\mathbf{h}_i\}_{i=1}^{n}$, $\mathbf{h}_i \in \mathbb{R}^d$ and the estimated worker ability $\mathbf{W} = \{\mathbf{W}^l\}_{l=1}^{m}$, where the size of $\mathbf{W}^l$ depends on the variants of the model.

2.3 Label Aggregation for Triplet Similarity Comparisons

Given a triplet of objects (o_i, o_j, o_k), we define the probability that "object o_i is more similar to object o_j than to object o_k". In the existing general models of Stochastic Triplet Embedding (STE) [3,13], the probability is given as

$$p_{ijk} = \frac{\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j)}{\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) + \mathbf{K}(\mathbf{x}_i, \mathbf{x}_k)}, \tag{1}$$

where

$$\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-(\mathbf{x}_i - \mathbf{x}_j)^\top (\mathbf{x}_i - \mathbf{x}_j)\right) \tag{2}$$

is a kernel function for measuring the similarity of two objects. The object embeddings can be learned by minimizing the cross-entropy loss function:

$$\mathcal{L} = -\sum_{(o_i, o_j, o_k) \in \mathcal{T}} (s_{ijk} \log p_{ijk} + (1 - s_{ijk}) \log(1 - p_{ijk}))) + \lambda_0 ||\mathbf{X}||_2^2, \tag{3}$$

$$s_{ijk} = \sum_{y_{ijk}^l \in \mathcal{Y}_{ijk}} \mathbb{1}(y_{ijk}^l = 1)/|\mathcal{Y}_{ijk}|, \tag{4}$$

where s_{ijk} is the normalized sum computed from y_{ijk}^l and indicates the proportion of answers saying o_i and o_j being more similar. $\mathbb{1}$ is an indicator function. λ_0 is a regularization hyperparameter. Then the estimated labels of triplet similarity comparisons of all objects can be computed by using the object embeddings. ven Der Maaten et al. ([3]) proposed a t-distributed Stochastic Triplet Embedding (tSTE), which uses a heavy-tailed Student-t kernel with α degrees of freedom instead of the Gaussian kernel (2), expressed as

$$\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = \left(1 + \frac{(\mathbf{x}_i - \mathbf{x}_j)^\top (\mathbf{x}_i - \mathbf{x}_j)}{\alpha}\right)^{-\frac{\alpha+1}{2}}. \tag{5}$$

2.4 Label Aggregation for Crowdsourced Triplet Similarity Comparisons

In the crowdsourcing context, there are large differences in the abilities among workers and difficulties among objects. Existing (t)STE methods do not consider them. We thus propose Crowd-(t)STE to solve this problem.

Worker Ability Modeling. The previously mentioned (t)STE model assumes that all crowd workers perform equally well. It does not distinguish the labels from different workers and utilizes the normalized sum (majority voting) of the

labels in Eq. (4). However, in the crowdsourcing context, the ability and diligence of workers are diverse. We thus propose a model that incorporates the worker abilities. We define an ability matrix $\mathbf{W}^l \in \mathbb{R}^{d' \times d'}$ for a worker a_l, and propose the probabilistic model and the two kernels as

$$p^l_{ijk} = \frac{\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{W}^l)}{\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{W}^l) + \mathbf{K}(\mathbf{x}_i, \mathbf{x}_k, \mathbf{W}^l)}, \tag{6}$$

$$\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{W}^l) = \exp\left(-(\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{W}^l (\mathbf{x}_i - \mathbf{x}_j)\right), \tag{7}$$

$$\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{W}^l) = \left(1 + \frac{(\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{W}^l (\mathbf{x}_i - \mathbf{x}_j)}{\alpha}\right)^{-\frac{\alpha+1}{2}}. \tag{8}$$

When $d' = 1$, $\mathbf{W}^l$ is a scalar variable, which means worker a_l performs equally on the entire dataset. When $d' = d$, we expect it potentially learn the rich representation of worker ability by interacting with each dimension in the object embeddings $\mathbf{x}_i$ when computing the probability of the similarity relations among objects.

Object Difficulty Modeling. Besides the worker ability, object difficulty is also an important factor that can influence the correctness of the judgments by workers, i.e., workers are more likely to assign incorrect answers to difficult object triplets. We thus also propose models that consider the object difficulty. In the existing work, methods for categorical labels, such as GLAD [14], leverage a scalar variable to represent the object difficulty; methods for pairwise labels, such as CrowdBT [2], do not model the object difficulty which is required to consider the interactions of the difficulties of two objects for the pairwise preference comparisons.

In this study that focuses on object triplet similarity comparisons, we define the difficulty based on an object, i.e., we utilize a d-dimensional vector $\mathbf{h}_i$ to represent the difficulty of an object o_i. To model the difficulty of a triplet, we need to interact the difficulties of the three objects in it. We thus use the dot product on the difficulties of two objects to compute a scalar for each object pair, to represent the difficulties of judging the similarity of the object pair. We propose the probabilistic model and candidate kernel functions as

$$p^l_{ijk} = \frac{\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{h_i}, \mathbf{h_j})}{\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{h_i}, \mathbf{h_j}) + \mathbf{K}(\mathbf{x}_i, \mathbf{x}_k, \mathbf{h_i}, \mathbf{h_k})}, \tag{9}$$

$$\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{h_i}, \mathbf{h_j}) = \exp\left(-(\mathbf{x}_i - \mathbf{x}_j)^\top (\mathbf{h}_i^\top \mathbf{h}_j)(\mathbf{x}_i - \mathbf{x}_j)\right), \tag{10}$$

$$\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{h_i}, \mathbf{h_j}) = \left(1 + \frac{(\mathbf{x}_i - \mathbf{x}_j)^\top (\mathbf{h}_i^\top \mathbf{h}_j)(\mathbf{x}_i - \mathbf{x}_j)}{\alpha}\right)^{-\frac{\alpha+1}{2}}. \tag{11}$$

Label Aggregation by Learning Object Embeddings. In summary, we propose a family of crowdsourced stochastic triplet embedding models by incorporating worker ability and object difficulty. The generalized model is described

as

$$p_{ijk}^{l} = \frac{\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{h}_i, \mathbf{h}_j, \mathbf{W}^l)}{\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{h}_i, \mathbf{h}_j, \mathbf{W}^l) + \mathbf{K}(\mathbf{x}_i, \mathbf{x}_k, \mathbf{h}_i, \mathbf{h}_k, \mathbf{W}^l)}. \tag{12}$$

$$\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{h_i}, \mathbf{h_j}, \mathbf{W}^l) = \exp\left(-(\mathbf{x}_i - \mathbf{x}_j)^\top (\mathbf{h}_i^\top \mathbf{h}_j) \mathbf{W}^l (\mathbf{x}_i - \mathbf{x}_j)\right), \tag{13}$$

$$\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{h_i}, \mathbf{h_j}, \mathbf{W}^l) = \left(1 + \frac{(\mathbf{x}_i - \mathbf{x}_j)^\top (\mathbf{h}_i^\top \mathbf{h}_j) \mathbf{W}^l (\mathbf{x}_i - \mathbf{x}_j)}{\alpha}\right)^{-\frac{\alpha+1}{2}}. \tag{14}$$

When $\mathbf{H} = \mathbf{1}$, it is equivalent to the model with Eqs. (6), (7), and (8); when $\mathbf{W} = \mathbf{1}$, it is equivalent to the model with Eqs. (9), (10), and (11). Note that although seemingly similar forms are utilized for two different factors of worker ability and object difficulty, they can be distinguished in the optimization because, when computing the loss, worker ability is aggregated by the labels of a worker and object difficulty is aggregated by the labels of an object.

The loss function in Eq. (3) of (t)STE utilizes an aggregated label s_{ijk}. In contrast, because we individually consider the influences of the triplet labels from different workers, we modify the loss to differ these labels. The cross-entropy loss function of Crowd-(t)STE can be formulated as follows:

$$\begin{aligned}\mathcal{L} = &- \sum_{y_{ijk}^l \in \mathcal{Y}} \left(y_{ijk}^l \log p_{ijk}^l + (1 - y_{ijk}^l) \log(1 - p_{ijk}^l)\right)\\ &+ \lambda_0 ||\mathbf{X}||_2^2 + \lambda_1 ||\mathbf{W}||_2^2 + \lambda_2 ||\mathbf{H}||_2^2.\end{aligned} \tag{15}$$

With the combinations of two types of worker ability, one type of object difficulty, and two types of kernel functions (Gaussian and Student-t), there are 10 variants of the proposed model in total. The combinations only without worker ability (or object difficulty) are included.

Because of the diverse properties of real datasets, certain variants are more appropriate for certain cases. We thus automatically search the variants when utilizing our approach. Recently, automatic machine learning, such as network architecture search [17,7], has been extensively studied. In contrast to these works, we focus on searching the candidate kernels. Because the search space is small, we utilize a brute force search to select the variant with optimal performance on the validation subset.

3 Experiments

3.1 Dataset Collection

In order to investigate the performance of the label aggregation approaches in the context of crowdsourced triplet similarity comparisons, we require real datasets that contain crowdsourced triplet similarity labels. However, to the best of our knowledge, no public datasets were available, thus we created two novel datasets by collecting triplet similarity comparison labels using a real-world crowdsourcing

Table 1. Statistics of the datasets. $|\mathcal{O}|$: number of objects, $|\mathcal{T}|$: number of triplets, $|\mathcal{Y}_{ijk}|$: number of workers for each triplet, $|\mathcal{A}|$: total number of workers, $|\mathcal{Y}|$: total number of labels; $|\mathcal{Y}^l|_{\min}$: minimum label numbers of workers, $|\mathcal{Y}^l|_{\max}$: maximum label numbers of workers, and $|\mathcal{Y}^l|_{\text{avg}}$: average label numbers of workers.

Data	$\|\mathcal{O}\|$	$\|\mathcal{T}\|$	$\|\mathcal{Y}_{ijk}\|$	$\|\mathcal{A}\|$	$\|\mathcal{Y}\|$	$\|\mathcal{Y}^l\|_{\min}$	$\|\mathcal{Y}^l\|_{\max}$	$\|\mathcal{Y}^l\|_{\text{avg}}$
Food	50	20,000	20	433	400,000	50	19,950	923.79
Scene	50	20,000	20	528	400,000	50	19,950	757.58

platform[1]. We first extracted sets of objects from existing image collections and then generated the triplets to publish them on the crowdsourcing platform. The questions we asked the crowd workers were "which image o_j and o_k is more similar to image o_i?".

The *food dataset* consists of images of five categories (bread, dessert, meat, soup, and vegetable/fruit) from the Food-11 image collection [10] (No Licence Required). For each category, we randomly selected ten images. The *scene dataset* consists of images of five categories (coast, forest, highway, open country, and street) from a collection of urban and natural scenes [9] (Licence CC BY 4.0). We also randomly selected ten images for each category. In each dataset, there are 50 images (objects) in total. We randomly sampled 20,000 triplets in each dataset and published them on the crowdsourcing platform to obtain comparison labels. Each crowd task includes 50 triplets. For each triplet, we collected labels from 20 workers. Each worker did not need to judge all of the triplets and only provided labels for a subset.

Table 1 shows the statistics of the datasets. Although the number of objects is not large, there are 400,000 crowd labels in total in each dataset, which is large-scale. The number of workers $|\mathcal{A}|$ are 433 for the Food dataset and 528 for the Scene dataset. The label numbers $|\mathcal{Y}^l|$ by a worker are diverse, i.e., the minimum and maximum are 50 and 19,950. In addition, although the total numbers of objects and triplets are the same for these two datasets, the total numbers of workers are different because the the label numbers of the workers are different. On average, one worker tended to label less triplets in the scene dataset than in the food dataset. Because the workers can decide the number of tasks they complete, this shows that workers stopped their annotations for the scene dataset earlier than for the food dataset. One possible reason for this is that the scene dataset is more difficult than the food dataset.

3.2 Experimental Settings

We compared our approach to two typical baselines that extend the existing methods that do not consider the factors of worker ability and object difficulty. We adapted and extended the vanilla STE and tSTE [3] for the crowdsourcing context as the baselines. Specifically, we extended the STE method using Eqs.

[1] Lancers: `www.lancers.jp`

(1), (2), and (15), and we extended tSTE method using Eqs. (1), (5), and (15), without using the regularization terms of worker ability and object difficulty in Eq. (15).

We define the names of the variants of our approach by using several suffixes to Crowd-(t)STE; '-s' denotes the scalar worker ability, '-m' for the matrix worker ability, '-d' for the object difficulty. The detailed hyperparameter settings of our approach are as follows. We carried out the experiments for each type of kernel function (Gaussian and Student-t) respectively. The degree of freedom α in all approaches using Student-t kernels is set to $d-1$. The regularization terms of all approaches are set to $\lambda_0, \lambda_1, \lambda_2 \in \{0.001, 0.005, 0.01, 0.05, 0.1\}$, $\lambda_0{=}\lambda_1{=}\lambda_2$. Although it is possible to tune different values for each λ ($\lambda_0 \neq \lambda_1 \neq \lambda_2$) to improve the performance, we mainly investigate the influence of data properties and model options and utilize equal λ. We tuned the hyperparameters λ and search the model variants based on their performance on the validation set. We implemented the approaches by Python and Theano.

3.3 Evaluation Methods

We verify the approaches on their capability to estimate the triplet comparison labels of all object triplets using only a small number of labeled object triplets. In one experimental trial, we first created a subset $\mathcal{Y}_{\mathcal{T}}^{u}$ by only using $u \in \{3, 5, 10\}$ labels in all labels of each triplet in $\mathcal{Y}$. $\mathcal{Y}_{\mathcal{T}}^{u}$ still contains all object triplets in $\mathcal{T}$. We then randomly selected a subset $\mathcal{T}_t$ of all object triplets in $\mathcal{T}$ with sampling rate $r \in \{0.05, 0.1, 0.2\}$. This triplet subset is defined as $\mathcal{Y}_t^u$ and was used as the training set. A subset $\mathcal{Y}_t^{u'}$ with the same size and settings was also created and used as the validation set. d is not tuned so that the proposed approaches and baselines are compared using same d, $d = 10$. The additional experiments with different d is in the appendix. We evaluated the average performance of ten trials for each (u, r, d) group. u and r are data properties and d is a hyperparameter.

We have two evaluation metrics to verify the proposed approach on the gold standards from different aspects. The objects in the real datasets we create have the category labels in the raw collections that they are from. We can use this category information to verify the performance of object embeddings by using the estimated object similarities. We utilized two object o_i and o_j from the same category and an object o_k from another category to create a triplet (o_i,o_j,o_k); then, we utilized all the triplet combinations from the objects in $\mathcal{O}$ that satisfy the above condition as the triplets in the ground truth. We computed the estimated similarity comparisons of these triplets using the estimated object embeddings and evaluated the accuracy of the estimated triplet comparisons. For an object triplet (o_i, o_j, o_k) in the ground truth, if the estimated similarity of (o_i, o_j) is higher than that of (o_i, o_k), we judge the estimated similarity comparison of this object triplet to be correct. We call this accuracy *category-based accuracy*.

The raw category information is not usually available in the scenario of our research topic, e.g., a set of logo designs have no clear categories. To evaluate the approach in a universal manner, we also use an evaluation metric

Table 2. Results for different sampling rates r. The number of dimensions $d = 10$. The number of labels (workers) per triplet $u = 5$. The numbers in the bold font indicate the best performance in each dataset in a group of approaches (STE or tSTE). We show the name of the optimal variant selected by the automatic search.

(a). Category-based accuracy.

Data	$r = 0.05$			$r = 0.10$			$r = 0.20$		
	STE	Crowd-STE		STE	Crowd-STE		STE	Crowd-STE	
Food	0.8700	**0.8927**	-s-d	0.9023	**0.9155**	-s-d	0.9179	**0.9127**	-s-d
Scene	0.8531	**0.9083**	-s-d	0.8978	**0.9174**	-s	0.9219	**0.9312**	-s
	tSTE	Crowd-tSTE		tSTE	Crowd-tSTE		tSTE	Crowd-tSTE	
Food	0.8397	**0.8833**	-s-d	0.8943	**0.9116**	-s-d	0.9187	**0.9175**	-s-d
Scene	0.8166	**0.8904**	-s	0.8828	**0.9203**	-s-d	0.9184	**0.9268**	-s-d

(b). Aggregation-based accuracy.

Data	$r = 0.05$			$r = 0.10$			$r = 0.20$		
	STE	Crowd-STE		STE	Crowd-STE		STE	Crowd-STE	
Food	0.7915	**0.8168**	-s-d	0.8276	**0.8494**	-s-d	0.8484	**0.8630**	-s-d
Scene	0.7826	**0.8165**	-s-d	0.8203	**0.8381**	-s	0.8403	**0.8567**	-s
	tSTE	Crowd-tSTE		tSTE	Crowd-tSTE		tSTE	Crowd-tSTE	
Food	0.7691	**0.8161**	-s-d	0.8171	**0.8528**	-s-d	0.8435	**0.8730**	-s-d
Scene	0.7512	**0.8116**	-s	0.8099	**0.8441**	-s-d	0.8354	**0.8579**	-s-d

called *aggregation-based accuracy* to measure the performance. The golden standard labels are computed by majority-voting based aggregation using $s_{ijk} = \sum_{y^l_{ijk} \in \mathcal{Y}_{ijk}} \mathbb{1}(y^l_{ijk} = 1)/|\mathcal{Y}_{ijk}|$ on all labels in $\mathcal{Y}$. If $s_{ijk} > 0.5$, the golden standard label is 1; otherwise, the golden standard label is 0. We trained the model based on a small subset $\mathcal{Y}^u_t$ and evaluated the accuracy of the estimated triplet comparisons to the aggregated labels of $\mathcal{Y}$. Furthermore, in the stage of tuning the hyperparameters λ and searching the model variants based on the performances on the validation set $\mathcal{Y}^{u'}_t$, we utilized the aggregation-based accuracy and the labels in the ground truth are the aggregated labels of the validation set $\mathcal{Y}^{u'}_t$.

3.4 Experimental Results

Q1: Is the proposed Crowd-(t)STE effective?

We show the results with representative hyperparameter setting and data properties as the conditions to compare the approaches: $d = 10$, which is a moderate number of dimensions for representing an object; $r = 0.1$ because we want to verify the performance when there are not many labels available; and $u = 5$ which is a moderate number of workers for annotating a triplet. The columns of $r = 0.10$ in Table 2 lists the results. We organize our methods and the baselines into two groups: Gaussian kernel-based group and Student-t kernel-based group. First, our proposed approach has better performance than the baselines in all of the cases in the columns of $r = 0.10$ in Table 2, regardless of which of the kernels (Gaussian or Student-t) being used. This shows that modeling crowdsourced factors such as worker ability and object difficulty is crucial for the task of label aggregation from crowdsourced triplet similarity

Table 3. Results for different numbers of labels (workers) per triplet u. The number of dimensions $d = 10$. The sampling rate $r = 0.10$. The numbers in the bold font indicate the best performance in each dataset in a group of approaches (STE or tSTE). We show the name of the optimal variant selected by the automatic search.

(a). Category-based accuracy.

Data	$u = 3$			$u = 5$			$u = 10$		
	STE	Crowd-STE		STE	Crowd-STE		STE	Crowd-STE	
Food	0.8911	**0.9132**	-s-d	0.9023	**0.9155**	-s-d	0.9097	**0.9109**	-s
Scene	0.8732	**0.9167**	-s-d	0.8978	**0.9174**	-s	0.9176	**0.9364**	-s-d
	tSTE	Crowd-tSTE		tSTE	Crowd-tSTE		tSTE	Crowd-tSTE	
Food	0.8824	**0.9066**	-s-d	0.8943	**0.9116**	-s-d	0.9001	**0.9108**	-s-d
Scene	0.8672	**0.9078**	-s	0.8828	**0.9203**	-s-d	0.9076	**0.9299**	-s-d

(b). Aggregation-based accuracy.

Data	$u = 3$			$u = 5$			$u = 10$		
	STE	Crowd-STE		STE	Crowd-STE		STE	Crowd-STE	
Food	0.8152	**0.8365**	-s-d	0.8276	**0.8494**	-s-d	0.8394	**0.8622**	-s
Scene	0.8042	**0.8302**	-s-d	0.8203	**0.8381**	-s	0.8357	**0.8568**	-s-d
	tSTE	Crowd-tSTE		tSTE	Crowd-tSTE		tSTE	Crowd-tSTE	
Food	0.8056	**0.8381**	-s-d	0.8171	**0.8528**	-s-d	0.8289	**0.8635**	-s-d
Scene	0.8006	**0.8257**	-s	0.8099	**0.8441**	-s-d	0.8263	**0.8598**	-s-d

comparisons. Secondly, the selected optimal variant of our approach is diverse. All two factors of worker ability and object difficulty have ever been selected in some cases. This shows that all these factors are effective for improving the overall performance and automatically search the optimal variant is important because certain variants are appropriate for certain data properties. Furthermore, because the variants modeling worker ability are always selected, it shows that worker ability is the required crowdsourced factor which need to be considered in the label aggregation approaches.

Q2: How do the data properties influence the performance?

We investigated the influences of two data properties, i.e., label (worker) number per triplet u and triplet sampling rate r. We changed one factor and fixed the others ($u = 5$, $r = 0.10$ and $d = 10$ as default), and verified the changes in the performance. Tables 2 and 3 list the results. First, the family of the proposed approaches always outperforms the baselines in all these cases. This observation is consistent with the results in the columns of $r = 0.10$ in Table 2. Secondly, when u and r increase, the amount of training data increases. The performances on the two accuracy metrics generally increase. Thirdly, when the data are very few and sparse, i.e., $r = 0.05$ in Table 2 and $u = 3$ in Table 3, our approach still performs well. The performance difference between the baselines and our approach when the data are relatively few (e.g., $u = 3$ in Table 3) is larger than that when the data are relatively abundant (e.g., $u = 10$ in Table 3). This shows that our approach is very efficient when the number of collected labels is small. It also shows that although collecting more labels can improve the performance, the improvement may be small. As shown in Table 3, $u = 10$ doubles the number of collected labels from $u = 5$, but the performance only increases by approximately

1%. Therefore, cares must be taken in the trade-off between budget cost and accuracy. Fourthly, the selected variants are different for each dataset with same u and r, which also shows the importance and effectiveness of automatic variant search.

Q3: How do the model options influence the performance?

We investigated the influences of the model options, i.e., the worker ability and object difficulty. We draw observations from Tables 2, 3. First, modeling worker ability is always required in all cases. For the options of modeling worker ability, the results show that the scalar worker ability '-s', rather than the matrix worker ability '-m', is always selected. This shows that the complexity of worker abilities on these two datasets is not high; a scalar value is capable of representing the worker ability, and using a large worker ability matrix may be overparameterized and can easily lead to overfitting. Secondly, the variants with object difficulty are sometimes selected depending on data properties; this shows that automatically selecting the optimal variant is useful for label aggregation from crowdsourced triplet similarity comparisons. Thirdly, we observed that the variants based on the Student-t kernel are not always better than those based on the Gaussian kernel. One candidate solution for this issue is searching the optimal models from both the Crowd-STE and Crowd-tSTE variants of the model.

4 Related Work

To learn the stochastic pairwise embedding from high-dimensional feature representations or pairwise similarity comparison labels, the objects are usually represented in a low-dimensional space so that pairwise similarities or neighborhoods are preserved [5,8,15]. Crowdsourced pairwise similarity comparison labels have also been leveraged for object clustering [4] and learning similarity matrices [11]; additional context information has also been utilized [16]. The costs of special multiple pairwise questions have also been discussed [6].

Because it is often difficult for humans to make absolute subjective judgements, in contrast to learning the embedding with absolute pairwise comparisons, some existing works have utilized relative comparisons with more than two objects, e.g., triplet comparisons [11,3,13] and quadruplet comparisons [1,12]. However, these works did not consider crowdsourced factors such as the diverse worker ability and object difficulty. In this study, we focused on crowdsourced triplet similarity comparisons and label aggregation for estimating similarity relations of all objects when only a small number of labels are available.

5 Conclusion

We considered label aggregation from crowdsourced triplet similarity comparisons. It can also be regarded as stochastic triplet embedding in the crowdsourcing context. We created two novel real datasets for investigating this research topic. We proposed a family of models by incorporating worker ability and object difficulty with different similarity kernels as well as automatically searching the

optimal model from the possible model variants. We also investigated how the data properties and model options influence the performance. A limitation of these datasets is that the objects are only images, other types of objects will be considered in future work. We can apply the proposed method to various domains including text opinions and 3D designs since our approach is not limited to image objects. Another interesting direction is efficient sampling method for adaptively collecting labels.

References

1. Agarwal, S., Wills, J., Cayton, L., Lanckriet, G., Kriegman, D., Belongie, S.: Generalized non-metric multidimensional scaling. In: AISTATS. pp. 11–18 (2007)
2. Chen, X., Bennett, P.N., Collins-Thompson, K., Horvitz, E.: Pairwise ranking aggregation in a crowdsourced setting. In: WSDM. pp. 193–202 (2013)
3. van Der Maaten, L., Weinberger, K.: Stochastic triplet embedding. In: MLSP. pp. 1–6 (2012)
4. Gomes, R.G., Welinder, P., Krause, A., Perona, P.: Crowdclustering. In: NIPS. pp. 558–566 (2011)
5. Hinton, G.E., Roweis, S.T.: Stochastic neighbor embedding. In: NIPS. pp. 857–864 (2003)
6. Korlakai Vinayak, R., Hassibi, B.: Crowdsourced clustering: Querying edges vs triangles. In: NIPS. pp. 1316–1324 (2016)
7. Liu, H., Simonyan, K., Yang, Y.: DARTS: Differentiable architecture search. arXiv preprint arXiv:1806.09055 (2018)
8. van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. JMLR **9**(Nov), 2579–2605 (2008)
9. Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. International Journal of Computer Vision **42**(3), 145–175 (2001). https://doi.org/10.1023/A:1011139631724
10. Singla, A., Yuan, L., Ebrahimi, T.: Food/non-food image classification and food categorization using pre-trained GoogLeNet model. In: MADiMa. pp. 3–11 (2016)
11. Tamuz, O., Liu, C., Belongie, S., Shamir, O., Kalai, A.T.: Adaptively learning the crowd kernel. In: ICML. pp. 673–680 (2011)
12. Ukkonen, A., Derakhshan, B., Heikinheimo, H.: Crowdsourced nonparametric density estimation using relative distances. In: HCOMP (2015)
13. Wah, C., Van Horn, G., Branson, S., Maji, S., Perona, P., Belongie, S.: Similarity comparisons for interactive fine-grained categorization. In: CVPR. pp. 859–866 (2014)
14. Whitehill, J., fan Wu, T., Bergsma, J., Movellan, J.R., Ruvolo, P.L.: Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In: NIPS. pp. 2035–2043 (2009)
15. Xie, B., Mu, Y., Tao, D., Huang, K.: m-sne: Multiview stochastic neighbor embedding. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) **41**(4), 1088–1096 (2011)
16. Yi, J., Jin, R., Jain, S., Yang, T., Jain, A.K.: Semi-crowdsourced clustering: Generalizing crowd labeling by robust distance metric learning. In: NIPS. pp. 1772–1780 (2012)
17. Zoph, B., Le, Q.V.: Neural architecture search with reinforcement learning. arXiv preprint arXiv:1611.01578 (2016)