Simultaneous Clustering and Ranking from Pairwise Comparisons

Jiyi Li[1,4], Yukino Baba[2,4], Hisashi Kashima[3,4]
[1] University of Yamanashi, Japan
[2] University of Tsukuba, Japan
[3] Kyoto University, Japan
[4] RIKEN Center for AIP, Japan
jyli@yamanashi.ac.jp, baba@cs.tsukuba.ac.jp, kashima@i.kyoto-u.ac.jp

Abstract

When people make decisions with a number of ideas, designs, or other kinds of objects, one attempt is probably to organize them into several groups of objects and to prioritize them according to some preference. The grouping task is referred to as *clustering* and the prioritizing task is called as *ranking*. These tasks are often outsourced with the help of human judgments in the form of *pairwise comparisons*. Two objects are compared on whether they are similar in the clustering problem, while the object of higher priority is determined in the ranking problem. Our research question in this paper is whether the pairwise comparisons for clustering also help ranking (and vice versa). Instead of solving the two tasks separately, we propose a unified formulation to bridge the two types of pairwise comparisons. Our formulation simultaneously estimates the object embeddings and the preference criterion vector. The experiments using real datasets support our hypothesis; our approach can generate better neighbor and preference estimation results than the approaches that only focus on a single type of pairwise comparisons.

1 Introduction

When people have a number of ideas, designs, or other kinds of objects, they may try to organize them into several groups of objects in order to make sense of their landscape and to prioritize them in order to decide the next actions. The grouping task is referred to as *clustering* and the prioritizing task is called *ranking*. These tasks are often outsourced with the judgments by the users themselves or some others such as crowdsourcing workers, and the judgments are often in the form of *pairwise comparisons* because humans are better at comparing objects rather than investigating each single object. Depending on the tasks, different types of pairwise comparisons are used: *pairwise similarity comparisons* and *pairwise preference comparisons*.

On one hand, in the object clustering task, two objects are compared to determine their similarity. The labels of such pairwise similarity comparisons are used to estimate the embedding of the objects in a latent space so that the distances among the objects in the space preserve the object similarity [Hinton and Roweis, 2003; van der Maaten and Hinton, 2008; Tamuz *et al.*, 2011; van Der Maaten and Weinberger, 2012; Agarwal *et al.*, 2007; Gomes *et al.*, 2011]. On the other hand, in the object ranking task, two objects are compared to determine which object is preferred. The labels of such pairwise preference comparisons are aggregated to a ranking list [Bradley and Terry, 1952; Chen *et al.*, 2013; Raman and Joachims, 2014; Chen and Joachims, 2016].

The existing approaches for these two kinds of tasks are separated from each other. They only utilize a single type of pairwise comparisons of the corresponding tasks. Here our research questions in this paper arise: *"Do the pairwise similarity comparisons (primarily used for clustering) also help object ranking?"* and *"Do the pairwise preference comparisons (primarily used for ranking) also help object clustering?"*. Our expectation is that the quality of both tasks is improved by simultaneously solving the two different tasks, rather than solving them separately. For example, in the ranking task, the objects with similar contents would probably have close ranks (while the reverse may be not always true). Similarly, in the clustering task, the objects with far different ranks would probably have dissimilar contents (again, the reverse is not always true.)

In this paper, we propose *Simultaneous Clustering And Ranking from PAirwise comparisons (SCARPA)*, a unified formulation to bridge the two types of pairwise comparisons. Our formulation depends on both of the object embeddings preserving the pairwise similarity among objects and the preference criterion vector; the object embeddings projected onto the direction of the vector represent the preference of the objects. Our approach iteratively learns the object embeddings and the preference scores by maximizing a mixed objective function which includes both pairwise preference and similarity information.

A typical usage of our method is to organize a large number of ideas generated by a variety of people [Siangliulue *et al.*, 2015; Hope *et al.*, 2017]. Due to the scale, it becomes challenging for users to explore a pool of the ideas and to identify the superior ones. Our method provides users with an efficient way to group and prioritize the ideas to support the decision-making process; the embedding of each idea obtained by our method can be used for visualizing the idea clusters so that users can easily see an overview of a diverse set of ideas, and the preference scores of the ideas help users decide priorities for investigation. Another example is to organize a large number of graphic designs obtained by a design competition. By using our method, stakeholders are able to group and prioritize them to decide which designs are similar and which ones are finally selected as the winners. Furthermore, besides the proposed simultaneous clustering and ranking tasks, our method can also be utilized for single clustering or ranking tasks by collecting additional labels of another type of comparisons.

We conduct experiments using several real datasets collected using crowdsourcing. These datasets include the examples of the idea and design collections that we aim to make decisions on them. The experimental results illustrate that our approach can generate better neighbor and preference estimation results than the approaches that only focus on a single type of pairwise comparisons, by only increasing a small number of cost on collecting additional labels.

The contributions of our work are mainly two-fold:

1. We propose the new problem of simultaneous clustering and ranking from two types of pairwise comparisons: the pairwise similarity comparison and the pairwise preference comparison.
2. We propose a unified formulation that bridges the two different types of pairwise comparisons so that we can utilize the information of both types of pairwise comparison to improve the quality of both clustering and ranking results.

2 Simultaneous Clustering and Ranking Problem

2.1 Clustering Problem and Ranking Problem

Suppose we have a set of n objects denoted by $\mathcal{O}$. We assume that no feature vector representations of the objects are available. Our goal is to organize the objects into several groups and to prioritize the objects. The former task is usually called clustering, and the latter is called ranking.

In the clustering task, we assume we first seek for a representation (or an embedding) of each object in a d-dimensional latent feature space; we denote by $\boldsymbol{x}_i \in \Re^d$ the representation of object i. Those representations reflect proximity relations among the objects, and then they can be used for clustering the objects (by e.g. the k-means clustering algorithm).

In the ranking task, our goal is to obtain a ranking list of the n objects. The ranking list reflects the relative preference among the objects.

2.2 Pairwise Comparisons

In order to obtain the embeddings and the ranking list of the set of objects $\mathcal{O}$, we assume we use human judgments, especially in terms of pairwise comparison. For a given object pair o_i and o_j in $\mathcal{O}$, we consider two different types of pairwise comparison: pairwise similarity comparison and pairwise preference comparison.

Pairwise similarity comparison is a type of questions asking the degree of similarity between two objects. In this paper, we consider the simplest kind of pairwise similarity comparison which is a binary-answer question, e.g. "Are the two objects are similar?" of which the candidate answers are "Yes" and "No". We can collect the answers for a number of object pairs, e.g., by using crowdsourcing, and the answers are aggregated to estimate their embeddings in a latent space, and they are further used for object clustering [Gomes *et al.*, 2011; Yi *et al.*, 2012; Korlakai Vinayak and Hassibi, 2016].

Pairwise preference comparison is a type of questions asking which of two given objects is of higher priority. This is also a binary-answer question, e.g. "Which of the two objects is preferred to the other one?" Collected answers are aggregated into a single ranking list through estimation of statistical models [Bradley and Terry, 1952; Chen *et al.*, 2013; Raman and Joachims, 2014].

In this paper, for the pairs of objects, we ask humans such as crowdsourcing workers to answer two types of pairwise comparison. Because the number of object pairs is quadratic in the number of objects, it costs too much budget and time if we ask all object pairs. Furthermore, if we use crowdsourcing, the comparison results can be rather noisy, and therefore multiple comparisons for the same object pairs are required to integrate them to obtain reliable results, which further increase the number of comparisons. Those facts motive us to obtain accurate clustering and ranking results using a limited number of comparison results.

2.3 Problem Definition

The key idea to address the present problem is to perform the clustering task and ranking simultaneously. Our hypotheses behind the idea is that (i) the pairwise similarity comparisons (that are primarily used for the clustering task) also help object ranking, and at the same time, (ii) the pairwise preference comparisons (that are primarily used for the ranking task) also help object clustering. Our expectation is that the quality of both tasks is improved by simultaneously solve the two different tasks, rather than solving them separately.

Finally, the *simultaneous clustering and ranking problem* is summarized as follows:

INPUT: We are given a set of pairwise preference labels $\mathcal{P} = \{p_{ij}\}_{i,j}$, where $p_{ij} \in [0, 1]$ is the result of preference comparison between objects o_i and o_j, which indicates the ratio of votes for o_i. We also have a set of pairwise similarity labels $\mathcal{S} = \{s_{ij}\}_{i,j}$, where s_{ij} is the result of similarity comparison between objects o_i and o_j, which indicates the ratio of the "Yes" answer (i.e., "similar"). Both pairwise preference labels and pairwise similarity labels have not necessarily been given for all of the object pairs.

OUTPUT: Our goal is to output the embeddings $\{\boldsymbol{x}_i\}_{i=1}^n$ and the preference scores $\{\tau_i\}_{i=1}^n$.

3 Our Approach

3.1 Relation of Different Pairwise Comparisons

When estimating the ranking list of the objects from the pairwise preference comparison labels, we estimate the compe-

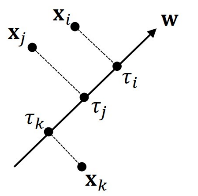


Figure 1: Relation between the competency and the object embeddings assumed in the proposed model. The competency of each object (τ_i, τ_j, and τ_k) is represented by the projection of its embeddings ($\boldsymbol{x}_i$, $\boldsymbol{x}_j$, and $\boldsymbol{x}_k$) on the preference criterion vector $\boldsymbol{w}$. Objects o_i and o_j are similar and have close competency; objects o_j and o_k have close competency but they can be dissimilar.

tency scores of the objects which are as consistent as possible with the preference labels by the pairwise preference models such as the Bradley-Terry model [Bradley and Terry, 1952; Chen *et al.*, 2013]. We also assume the standard Bradley-Terry model which gives the probability that object o_i is preferred to object o_j as

$$\theta_{ij} = \Pr(o_i \succ o_j) = \frac{1}{1 + \exp\left(-\left(\tau_i - \tau_j\right)\right)}, \tag{1}$$

where τ_i denotes the competency score of object o_i.

When estimating the neighborhood of the objects from the pairwise similarity comparison labels, we learn the embeddings of the objects so that they are as consistent as possible with the similarity labels by the neighbor embedding models [Hinton and Roweis, 2003; van Der Maaten and Weinberger, 2012]. We employ the simplest neighborhood model that gives the probability that two objects o_i and o_j are neighbours of each other as

$$\phi_{ij} = \exp(-||\boldsymbol{x}_i - \boldsymbol{x}_j||^2), \tag{2}$$

where the embedding of object o_i is denoted by a d-dimensional vector $\boldsymbol{x}_i$.

Our idea is to bridge the different types of pairwise comparison by assuming a relation between the competency score τ_i and the object embedding $\boldsymbol{x}_i$; namely, we assume

$$\tau_i = \boldsymbol{w}^\top \boldsymbol{x}_i, \tag{3}$$

where $\boldsymbol{w}$ is a d-dimensional parameter vector which represents the preference criterion. Each dimension of $\boldsymbol{w}$ indicates the impact of the corresponding feature in the latent space on the object competency. The competency τ_i of an object o_i can be represented by the projection of its embeddings $\boldsymbol{x}_i$ on the preference criterion vector $\boldsymbol{w}$. Figure 1 gives a graphical explanation of the relation between the competency scores and the object embeddings.

The relation given by Eq.(3) implies that (i) a similar object pair have close competencies, while an object pair with close competencies are not necessarily to be similar, and (ii) an object pair with far competencies is dissimilar, while a dissimilar object pair does not necessarily have far competencies. Figure 1 illustrates the assumptions, i.e., the similar object pair o_i and o_j have close competencies, while o_j and o_k have close competencies but they are dissimilar. By using Eq. (3), the preference model (Eq. (1)) is written as:

$$\theta_{ij} = \frac{1}{1 + \exp(-\boldsymbol{w}^\top(\boldsymbol{x}_i - \boldsymbol{x}_j))}. \tag{4}$$

3.2 Simultaneous Clustering and Ranking

Now we formulate the simultaneous clustering and ranking problem as an optimization problem that finds both the object embeddings $\{\boldsymbol{x}_i\}_i$ and the preference criterion vector $\boldsymbol{w}$. The overall objective function F is defined as

$$\begin{aligned} F\left(\{\boldsymbol{x}_i\}_i, \boldsymbol{w}\right) =& \alpha R(\{\boldsymbol{x}_i\}_i, \boldsymbol{w}) + \beta E(\{\boldsymbol{x}_i\}_i) \\ & - \eta \parallel \boldsymbol{w} \parallel_2^2 - \gamma \sum_i \parallel \boldsymbol{x}_i \parallel_2^2, \end{aligned} \tag{5}$$

where $R(\cdot, \cdot)$ and $E(\cdot)$ are two objective functions for ranking and embedding, respectively, and α and β are their mixture constants; the last two terms are the regularization terms with constants $\eta \geq 0$ and $\gamma \geq 0$.

The objective function for ranking and embedding are defined as follows, respectively:

$$\begin{aligned} R(\{\boldsymbol{x}_i\}_i, \boldsymbol{w}) =& \sum_{p_{ij} \in \mathcal{P}} \left\{ p_{ij} \log \frac{1}{1 + \exp(-\boldsymbol{w}^\top(\boldsymbol{x}_i - \boldsymbol{x}_j))} \right. \\ & \left. + (1 - p_{ij}) \log \left(1 - \frac{1}{1 + \exp(-\boldsymbol{w}^\top(\boldsymbol{x}_i - \boldsymbol{x}_j))}\right) \right\}. \end{aligned} \tag{6}$$

$$\begin{aligned} E(\{\boldsymbol{x}_i\}_i) =& \sum_{s_{ij} \in \mathcal{S}} \left\{ s_{ij} \log(\exp(-||\boldsymbol{x}_i - \boldsymbol{x}_j||^2)) \right. \\ & \left. + (1 - s_{ij}) \log(1 - \exp(-||\boldsymbol{x}_i - \boldsymbol{x}_j||^2)) \right\}. \end{aligned} \tag{7}$$

By using the gradient descent method and updating $\{\boldsymbol{x}_i\}_i$ and $\boldsymbol{w}$ iteratively, we obtain $\{\boldsymbol{x}_i\}_i$ and $\boldsymbol{w}$ that maximizes the objective function Eq. (5).

Once the embeddings $\{\boldsymbol{x}_i\}_i$ and the preference criterion vector $\boldsymbol{w}$ are obtained, we compute the competency scores of the objects by using Eq. (3) to make a ranking list, and the similarity scores of two objects based on Eq. (2) to use them for further applications such as clustering.

4 Experiments

4.1 Baselines

We compare SCARPA, the proposed approach which bridges both two types of comparison labels, with the baseline approaches which only utilize either preference labels or similarity labels:

- **Preference Comparison Embeddings (PCE)**: it uses the preference labels only. It is similar to the standard Bradley-Terry model [Bradley and Terry, 1952] but it learns both embeddings and preference criterion by objective function R (Eq. (6)) with regularization terms.
- **Similarity Comparison Embeddings (SCE)**: it uses the similarity labels only to estimate the embeddings by optimizing the objective function E (Eq. (7)) with regularization terms. It represents the existing work for similarity embedding such as [Hinton and Roweis, 2003; van Der Maaten and Weinberger, 2012].

These baselines are representative in the related topics. Indeed, the extensions of these baselines have been proposed with more sophisticated models. For example, PCE can be extended by adding worker ability [Chen *et al.*, 2013]; SCE

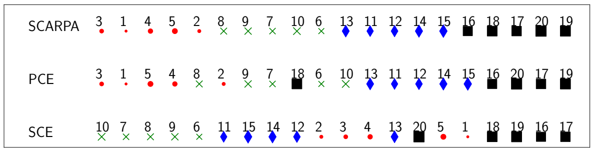
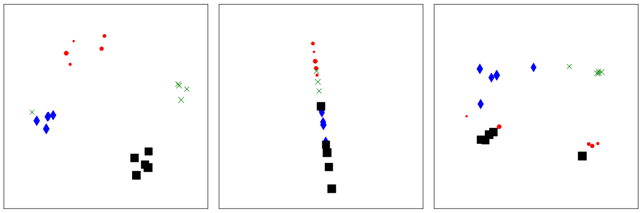


Figure 2: Comparison on ranking task. The size value at the top of each object indicates the true competency. SCE fails to correctly rank the objects, and SCARPA has more accurate ranking than PCE.



(a) SCARPA (b) PCE (c) SCE

Figure 3: Comparison on clustering task. The shape (with specific color) of each object indicates the true cluster. Sampling rate $r = 0.2$ for all approaches. SCARPA successfully categorized the objects based on their shapes; PCE failed to generate clusters and SCE made more incorrect assignments.

can be extended based on more robust distributions such as the t-distribution [van Der Maaten and Weinberger, 2012]. However, the choice of the fundamental models of ranking or clustering is beyond our scope, and our approach can also be extended in the similar ways.

4.2 Experiment with a Synthetic Dataset

We first conducted an experiment with a synthetic dataset to easily observe the characteristics of the approaches by using the ground truth of the competencies and the embeddings. We generated objects that are characterized by two attributes: shape and size. There are four shape types (circle, cross, diamond, and square) and a size. For each of the shape types, we generate five objects. The sizes of circles, crosses, diamonds, and squares are 1–5, 6–10, 11–15, and 16–20, respectively. We then generated a preference comparison label and a similarity preference comparison label for each object pair. The competency of an object is determined by its size; the preference label is $p_{ij} = 1$ if o_i is smaller than the o_j. The similarity comparison is evaluated by the shape; the similarity label is $s_{ij} = 1$ if the objects have the same shape. We sampled 20% of the object pairs and applied each method. The hyperparameter tuning is same with that of the experiments on the real datasets which will be introduced in Section 4.5.

Figure 2 visualizes the results on the ranking task. It shows that SCE failed to correctly rank the objects and SCARPA provided more accurate ranking than PCE; for example, in SCARPA, the objects with size 4, 2, 10 and 18 have the better rank. Figure 3 visualizes the results on the clustering task. SCARPA almost completely correctly distinguished the objects based on their shapes while SCE made more incorrect assignments, and PCE failed to generate clusters. These results demonstrate the effectiveness of SCARPA on both ranking and clustering tasks.

4.3 Real Datasets

We construct five real datasets with tasks consisting of preference comparisons and similarity comparisons. These datasets contain the examples of idea and design collections that we aim to group and prioritize to make decisions. The pairwise comparison labels are collected by a commercial crowdsourcing platform *Lancers*[1]. We collect the labels for all object pairs so that we can use any object pair in the experiments. 10 labels are collected for each object pair, which are aggregated by majority voting to create the ground truth labels.

Design: We can easily collect many diverse designs such as drawings and charts from crowds. The design preference can be used for selecting best designs and the design similarity can be used to group the designs. To collect the pairwise comparisons, our solution is to ask another group of crowds to compare the pairs of designs. We focus on designed logos used for the home pages of companies. 21 designs and 20 designs are collected for the two companies respectively, which are used as two datasets, i.e., *design1* (with 52 workers) and *design2* (with 65 workers).

Idea: We can ask a group of crowds to provide ideas for a problem. Another group of crowds evaluates these ideas on the preference and similarity. The ideas for two problems are collected. One problem is "how to decrease the cheating behaviors in the examination" (*idea1* dataset, with 40 ideas and 189 workers). An example of the ideas is "using different sequences of the questions for different students". The other problem is "how to decrease the absences and lateness for the meetings" (*idea2* dataset, with 40 ideas and 184 workers).

Dog: One useful application of our work is the subjective evaluation of images on the issues such as aesthetics. We ask the crowds to evaluate a set of dog images. For the preference tasks, the question is "which dog in the images is cuter?". For the neighbor estimation task, the question is "whether two dogs are similar or belong to the same breed". We select six dog breeds and five images for each breed (30 images in total) from the Stanford Dogs Dataset [Khosla *et al.*, 2011]. There are 74 workers.

4.4 Evaluation Metrics

We evaluate the different approaches on their ability to estimate the pairwise comparison labels of all object pairs by only a small number of labeled object pairs. In details, in one experimental trial, we randomly select a subset of all object pairs with sampling rate $r = 0.1$. For both two types of pairwise comparison, we only use five labels in the ten labels of each object pair. We denote the subset of labels as $\mathcal{P}_k$ and $\mathcal{S}_k$, where k is the index of the subset. We evaluate the average performance of ten trials.

There are two different settings on the selected objects pairs in $\mathcal{P}_k$ and $\mathcal{S}_k$. One is to use the same object pairs for both $\mathcal{P}_k$ and $\mathcal{S}_k$. It only increases a small number of budget cost because workers do not need to judge more objects and only need to provide an extra label. The other setting is to use

[1] www.lancers.jp/

Dataset	Metric	SCARPA	PCE	SCE
Design1	preference	**0.7210 ± 0.0468**	0.7057 ± 0.0397	0.4519 ± 0.0867
	similarity	**0.6439 ± 0.0584**	0.5617 ± 0.1157	0.6044 ± 0.0857
Design2	preference	**0.6159 ± 0.0635**	0.6147 ± 0.0632	0.5016 ± 0.0459
	similarity	**0.5707 ± 0.0757**	0.5567 ± 0.0761	0.5136 ± 0.0718
Idea1	preference	**0.6719 ± 0.0256**	0.6609 ± 0.0324	0.5173 ± 0.0537
	similarity	**0.6209 ± 0.0238**	0.5640 ± 0.0377	0.6077 ± 0.0241
Idea2	preference	**0.5936 ± 0.0197**	0.5921 ± 0.0193	0.5032 ± 0.0422
	similarity	**0.6939 ± 0.0329**	0.5014 ± 0.0535	0.6869 ± 0.0319
Dog	preference	**0.7907 ± 0.0254**	0.7848 ± 0.0222	0.5191 ± 0.0866
	similarity	**0.6926 ± 0.0410**	0.6481 ± 0.0283	0.6506 ± 0.0518

Table 1: Comparison on the real datasets. The winners are bold-faced. SCARPA outperforms the Baselines (PCE and SCE).

different object pairs. It doubles the budget cost for collecting the same total number of labels in both $\mathcal{P}_k$ and $\mathcal{S}_k$ because the workers need to judge different objects.

We use two performance evaluation metrics: *pairwise preference accuracy* and *pairwise similarity accuracy*. The pairwise preference accuracy is the accuracy of the estimated preference. For an object pair o_i and o_j, if o_i is preferred to o_j in the ground truth labels, the estimated competency score $\hat{\tau}_i$ should be higher than $\hat{\tau}_j$. The pairwise similarity accuracy is the accuracy of the relations of estimated similarity of two object pairs. Without a similar-dissimilar threshold, we cannot assign a similarity label to the estimated similarity $\hat{s}_{ij}$. Instead, for a similar object pair (o_a, o_b) and a dissimilar object pair (o_c, o_d) in the ground truth, if the estimated similarity $\hat{s}_{ab}$ is higher than $\hat{s}_{cd}$, we judge the relation of the estimated similarity of these two object pairs is correct.

4.5 Tuning Hyperparameters

Since we have no access to a subset of the ground truth labels in our unsupervised problem setting, we cannot use them for tuning the hyperparameters. Instead, we leverage surrogate ground truth and the surrogate performance on them. We use the label subsets $\mathcal{P}_k$ and $\mathcal{S}_k$ which are the same ones used by our approach to learning the embeddings and preference criterion vector as the surrogate ground truth. We tune the parameters by the measures of the pairwise preference accuracy on the held out subset of $\mathcal{P}_k$ and the pairwise similarity accuracy on the held out subset of $\mathcal{S}_k$. For the ranking (clustering) task, the preference (similarity) accuracy on $\mathcal{P}_k$ ($\mathcal{S}_k$) has higher priority in sorting the results generated by different hyperparameter groups. In the case that there are multiple hyperparameter groups which can reach same value on these two measures, we use the average performance on these groups as the experimental results.

The detailed hyperparameter settings of our approach are as follows. The dimension of embeddings d is set to 10. The regularization terms are set to $\eta = 0.1$ and $\gamma = 0.1$. Although it is possible to tune d, η and γ to improve the performance, we mainly investigate the influence of the mixture constants of preference information and similarity information in our approach. The value of α is chosen from $\{0.25, 0.5, 1, 2, 4\}$; the value of β is chosen from $\{0.25, 0.5, 1, 2, 4\}$. We use R (Eq. (6)) with regularization for the initialization of our approach because it can initialize both the embeddings and preference criterion vector. R (Eq. (6)) with regularization uses random initialization. The hyperparameters of the baselines are also tuned in a similar way.

4.6 Results

Comparison of Different Approaches

Table 1 shows the results of the comparison on the performance between our approach and other baseline approaches. In this experiment, the settings for object pair selection is using same object pairs. PCE shows the better performance than SCE on estimated preference labels and worse performance than SCE on estimated similarity labels in most of the datasets. It is because that PCE focuses on preference comparisons and SCE focuses on similarity comparisons.

Table 1 shows that SCARPA generates better results on both estimated preference and similarity labels than PCE and SCE. Our proposed approach can effectively bridge the heterogeneous pairwise comparisons and generate better embeddings and preference criterion vector than the approaches which only utilize a single type of pairwise comparisons.

Costs and Object Pair Selection

There are at least two kinds of costs concerned in such kinds of tasks: time cost and budget cost. Regarding the time cost of running the approaches, SCARPA has the same order of time complexity as the existing work like PCE and SCE. The other is the time and budget cost of collecting the labels. The cost of collecting the labels is more sensitive than the cost of carrying out a ranking or clustering approach like SCARPA. We thus discuss the increase of budget cost and profit of performance improvement between using same and different object pairs.

For this purpose, we construct experiments with two different settings on the selected object pairs $\mathcal{P}_k$ and $\mathcal{S}_k$ discussed in Section 4.4. Table 2 shows the results in the 'same object pairs' scenario and the 'different object pairs' scenario.

First, we compare the results in the columns of 'different object pairs'. The underline font indicates the best results in these columns. SCARPA generally performs better on the estimated preference and similarity labels for most of the datasets when using different object pairs.

Second, we compare the results in the column of 'same object pairs' with the columns of 'different object pairs'. The winners in all columns are bold-faced. On one hand, for the results between SCARPA with same object pairs and

		Same Object Pairs	Different Object Pairs		
Dataset	Metric	SCARPA	SCARPA	PCE	SCE
Design1	preference	0.7210 ± 0.0468	***0.7481 ± 0.0592***	0.7412 ± 0.0573	0.4567 ± 0.0857
	similarity	***0.6439 ± 0.0585***	0.6216 ± 0.0673	0.4965 ± 0.0698	0.6040 ± 0.0891
Design2	preference	*0.6159 ± 0.0635*	0.6068 ± 0.0357	**0.6184 ± 0.0446**	0.5016 ± 0.0459
	similarity	***0.5707 ± 0.0757***	0.5301 ± 0.0993	0.5442 ± 0.0928	0.5136 ± 0.0718
Idea1	preference	***0.6719 ± 0.0256***	0.6635 ± 0.0137	0.6577 ± 0.0177	0.5173 ± 0.0537
	similarity	***0.6209 ± 0.0238***	0.6247 ± 0.0238	0.5704 ± 0.0310	0.6077 ± 0.0241
Idea2	preference	***0.5936 ± 0.0197***	0.5879 ± 0.0303	0.5871 ± 0.0277	0.5032 ± 0.0422
	similarity	***0.6939 ± 0.0329***	0.6914 ± 0.0248	0.5220 ± 0.0366	0.6869 ± 0.0319
Dog	preference	0.7907 ± 0.0254	***0.8055 ± 0.0224***	0.7880 ± 0.0237	0.5191 ± 0.0866
	similarity	0.6926 ± 0.0410	***0.7504 ± 0.0384***	0.6689 ± 0.0508	0.6506 ± 0.0518

Table 2: Same VS. Different Object Pairs. The winners are bold-faced and the winners in the different object pair cases are underlined. SCARPA outperforms PCE and SCE; SCARPA with same and different objects pairs win each other in different evaluations (italicized), while SCARPA with same object pairs requires much less additional labels than that with different object pairs.

SCARPA with different object pairs, we use the italic font to mark the better results in these two columns. The observation is that although using different object pairs for different types of comparisons may generate better results in some cases (3 in 10 evaluations, e.g., dog dataset), using same object pairs for heterogeneous pairwise comparisons is also possible to have better results in some cases (7 in 10 evaluations, e.g., two idea datasets). We can regard that these observations are influenced by the object pair selection settings. On the other hand, the results of SCARPA with same object pairs are better on both preference and similarity estimation than that of PCE and SCE with different object pairs in most of the cases (8 in 10 evaluations for PCE and 10 in 10 evaluations for SCE).

From the aspect of budget cost, in contrast to using different objects pairs for different types of pairwise comparisons which doubles the cost to collect same number of labels, using same object pairs does not increase the cost of workers a lot, because the number of objects that a worker needs to review for judgment does not increase. Actually, answering one more pairwise comparison to an object pair may help the worker to understand the objects and provide better labels.

From the aspect of performance improvement, on one hand, from the results of idea datasets, when using same object pairs for both two types of comparisons, the label information from different types provide effective complementary to each other. Overlaps between preference pairs and similarity pairs can bolster each other. The same object pairs setting can most efficiently benefit from the overlap which explains the 7/10 stable winning rate in the experiments, but it is rather conservative.

On the other hand, from the results of dog dataset, a different set of object pairs which have data about twice as many objects are potentially advantageous with more effective information. In other words, using different object pairs may reach better performance. However, without a rational method to properly select the different set of object pairs, randomly selection is difficult to always reach the effect ones. It has the risk to reach the ineffective ones, which explains the 3/10 winning rate in the experiments. It may perform well when we luckily draw moderately overlapped pairs.

In conclusion, when using same object pairs for both types of pairwise comparisons, our approach can generate better performance than the approach using only a single type of comparisons, by only increasing a small number of cost on collecting the labels. The optimal solution is probably somewhere between the settings of using same and different object pairs, i.e., moderately overlapped cases. How to effectively select different sets of object pairs for different types of pairwise comparison is one of our future work.

5 Related Work

Pairwise preference comparison and ranking had been discussed for decades [Cattelan, 2012]; a typical solution is the Bradley-Terry model [Bradley and Terry, 1952] and its various extensions or generalizations had been proposed, e.g., multiple dimensions [Causeur and Husson, 2005] and intransitivity [Chen and Joachims, 2016]. Recent work also discussed extensions in modern settings, e.g., modeling worker ability in the context of crowdsourcing [Chen *et al.*, 2013] and peer grading in MOOCs [Raman and Joachims, 2014].

In pairwise similarity comparison and embedding, the objects were usually represented in a low-dimensional space so that pairwise similarities were preserved [Hinton and Roweis, 2003; van der Maaten and Hinton, 2008; Xie *et al.*, 2011]. In contrast to embedding with absolute pairwise comparisons, some work utilized relative comparisons with more than two objects, e.g., triplet comparisons [Tamuz *et al.*, 2011; van Der Maaten and Weinberger, 2012; Wah *et al.*, 2014] and quadruplet comparisons [Agarwal *et al.*, 2007; Ukkonen *et al.*, 2015]. Crowdsourced similarity labels were also leveraged for object clustering [Gomes *et al.*, 2011] and learning similarity matrices [Tamuz *et al.*, 2011]. Additional context information were utilized [Yi *et al.*, 2012]. The costs of special multiple pairwise questions were also discussed [Korlakai Vinayak and Hassibi, 2016]. In contrast to the existing work focus only on a single type of pairwise comparisons, our work can leverage the both types of pairwise comparisons.

6 Conclusion

We propose the new problem of simultaneous clustering and ranking from two types of pairwise comparisons: the pairwise similarity and preference comparison. We propose a unified formulation which bridges the two different types of pairwise comparisons. The experiments illustrate that our approach can generate better neighbor and preference estimation results than the approaches that only focus on a single type of pairwise comparisons by only increasing a small number of cost on collecting additional labels. In future work, we will focus on how to effectively select different object pairs.

Acknowledgments

This research was supported by JSPS KAKENHI Grant Number 15H01704, Japan.

References

[Agarwal *et al.*, 2007] Sameer Agarwal, Josh Wills, Lawrence Cayton, Gert Lanckriet, David Kriegman, and Serge Belongie. Generalized non-metric multidimensional scaling. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, pages 11–18, 2007.

[Bradley and Terry, 1952] Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

[Cattelan, 2012] Manuela Cattelan. Models for paired comparison data: A review with emphasis on dependent data. *Statistical Science*, pages 412–433, 2012.

[Causeur and Husson, 2005] David Causeur and François Husson. A 2-dimensional extension of the bradley–terry model for paired comparisons. *Journal of Statistical Planning and Inference*, 135(2):245–259, 2005.

[Chen and Joachims, 2016] Shuo Chen and Thorsten Joachims. Modeling intransitivity in matchup and comparison data. In *Proceedings of the 9th ACM International Conference on Web Search and Data Mining*, pages 227–236, 2016.

[Chen *et al.*, 2013] Xi Chen, Paul N. Bennett, Kevyn Collins-Thompson, and Eric Horvitz. Pairwise ranking aggregation in a crowdsourced setting. In *Proceedings of the 6th ACM International Conference on Web Search and Data Mining*, pages 193–202, 2013.

[Gomes *et al.*, 2011] Ryan G. Gomes, Peter Welinder, Andreas Krause, and Pietro Perona. Crowdclustering. In *Advances in Neural Information Processing Systems 24*, pages 558–566. 2011.

[Hinton and Roweis, 2003] Geoffrey E Hinton and Sam T. Roweis. Stochastic neighbor embedding. In *Advances in Neural Information Processing Systems 15*, pages 857–864. 2003.

[Hope *et al.*, 2017] Tom Hope, Joel Chan, Aniket Kittur, and Dafna Shahaf. Accelerating innovation through analogy mining. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 235–243, 2017.

[Khosla *et al.*, 2011] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization. In *Proceedings of the 1st Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[Korlakai Vinayak and Hassibi, 2016] Ramya Korlakai Vinayak and Babak Hassibi. Crowdsourced clustering: Querying edges vs triangles. In *Advances in Neural Information Processing Systems 29*, pages 1316–1324. 2016.

[Raman and Joachims, 2014] Karthik Raman and Thorsten Joachims. Methods for ordinal peer grading. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1037–1046, 2014.

[Siangliulue *et al.*, 2015] Pao Siangliulue, Kenneth C. Arnold, Krzysztof Z. Gajos, and Steven P. Dow. Toward collaborative ideation at scale: Leveraging ideas from others to generate more creative and diverse ideas. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work*, pages 937–945, 2015.

[Tamuz *et al.*, 2011] Omer Tamuz, Ce Liu, Serge Belongie, Ohad Shamir, and Adam Tauman Kalai. Adaptively learning the crowd kernel. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pages 673–680, 2011.

[Ukkonen *et al.*, 2015] Antti Ukkonen, Behrouz Derakhshan, and Hannes Heikinheimo. Crowdsourced nonparametric density estimation using relative distances. In *Proceedings of the 3rd AAAI Conference on Human Computation and Crowdsourcing*, 2015.

[van der Maaten and Hinton, 2008] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[van Der Maaten and Weinberger, 2012] Laurens van Der Maaten and Kilian Weinberger. Stochastic triplet embedding. In *Proceedings of 2012 IEEE International Workshop on Machine Learning for Signal Processing*, pages 1–6, 2012.

[Wah *et al.*, 2014] Catherine Wah, Grant Van Horn, Steve Branson, Subhransu Maji, Pietro Perona, and Serge Belongie. Similarity comparisons for interactive fine-grained categorization. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 859–866, 2014.

[Xie *et al.*, 2011] Bo Xie, Yang Mu, Dacheng Tao, and Kaiqi Huang. m-sne: Multiview stochastic neighbor embedding. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 41(4):1088–1096, 2011.

[Yi *et al.*, 2012] Jinfeng Yi, Rong Jin, Shaili Jain, Tianbao Yang, and Anil K. Jain. Semi-crowdsourced clustering: Generalizing crowd labeling by robust distance metric learning. In *Advances in Neural Information Processing Systems 25*, pages 1772–1780. 2012.