Quaternion Ordinal Embedding

Wenzheng Hou[1,2] , **Qianqian Xu**[1*] , **Ke Ma**[2] , **Qianxiu Hao**[1,2] and **Qingming Huang**[1,2,3,4*]
[1]Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, CAS
[2]School of Computer Science and Technology, University of Chinese Academy of Sciences
[3]Key Laboratory of Big Data Mining and Knowledge Management, Chinese Academy of Sciences
[4]Artificial Intelligence Research Center, Peng Cheng Laboratory
houwenzheng20@mails.ucas.ac.cn, xuqianqian@ict.ac.cn, make@ucas.ac.cn
haoqianxiu19@mails.ucas.ac.cn qmhuang@ucas.ac.cn

Abstract

Ordinal embedding (OE) aims to project objects into a low-dimensional space while preserving their ordinal constraints as well as possible. Generally speaking, a reasonable OE algorithm should simultaneously capture a) semantic meaning and b) the ordinal relationship of the objects. However, most of the existing methods merely focus on b). To address this issue, our goal in this paper is to seek a generic OE method to embrace the two features simultaneously. We argue that different dimensions of vector-based embedding are naturally entangled with each other. To realize a), we expect to decompose the D dimensional embedding space into D different semantic subspaces, where each subspace is associated with a matrix representation. Unfortunately, introducing a matrix-based representation requires far more complex parametric space than its vector-based counterparts. Thanks to the algebraic property of quaternions, we are able to find a more efficient way to represent a matrix with quaternions. For b), inspired by the classic chordal Grassmannian distance, a new distance function is defined to measure the distance between different quaternions/matrices, on top of which we construct a generic OE loss function. Experimental results for different tasks on both simulated and real-world datasets verify the effectiveness of our proposed method.

1 Introduction

Ordinal Embedding (OE) aims to embed objects into a space such that the distances between the embeddings satisfy the constraints of the relative comparisons (also called ordinal relationships) as much as possible. This problem has first been studied by Shepard [Shepard, 1962a; Shepard, 1962b] and Kruskal [Kruskal, 1964a; Kruskal, 1964b], and lately has drawn wide interests in the multimedia, information retrieval and computer vision communities.

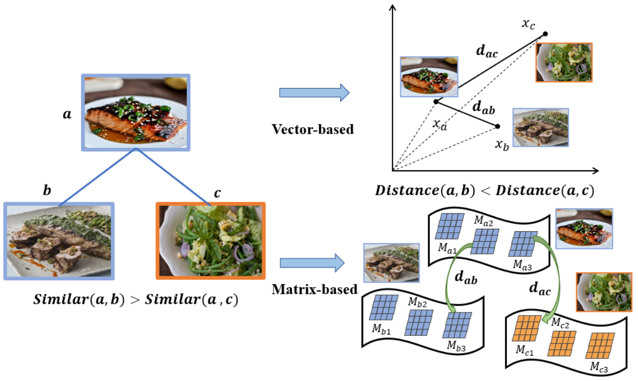


Figure 1: Comparison between two types of ordinal embedding mechanism.The upper part shows that the vector-based method might be hard to simultaneously capture the semantic meaning and ordinal relationships. The lower part demonstrates the proposed method represents each object as a set of subspace, modeling the semantic concepts in an explicit manner.

Most of the existing ordinal embedding methods pursue the ***vector-based*** representations of objects. Specifically, denote $\boldsymbol{x}_a$, $\boldsymbol{x}_b$ and $\boldsymbol{x}_c$ as the vector representations of object a, b and c, respectively. If object a is more similar to object b than object c, then OE tries to push $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$ together while pull $\boldsymbol{x}_a$ and $\boldsymbol{x}_c$ away. We illustrate this convention in the upper part of Fig. (1).

Intuitively, a good representation should convey both the semantic meanings (for example, the toasted salmon and the marinated pork in Fig. (1) are rich in protein) and the ordinal relationships (for example, the toasted salmon is more similar to the marinated pork than the Herb salad). However, the vector-based representation and the corresponding element-wise similarity function, *e.g.*, Minkowski distance, cosine similarity, hamming distance, etc, might fail to simultaneously capture the above two information well. This is because the objects are mapped into a single common high-dimensional space so that it is hard to disentangle different dimensions to express different semantics. For example, the toasted salmon and the marinated pork are semantically related w.r.t the concept 'rich in protein'. However, a simple vector-based representation is hard to explicitly model such a concept.

To avoid such a problem, we propose to model an object as a set of semantics subspace, with each subspace representing

[*] Corresponding Author

a potential semantic concept, as opposed to a single vector-based representation. By doing so, we hope to establish an explicit connection between the overall similarity in the embedding space and the semantic concepts.

Then two key problems arise: 1) how to formulate a subspace? 2) how to measure the distance across subspaces? To solve the first problem, we might resort to a matrix-based representation in the sense that each linear subspace corresponds to a matrix. At first glance, one might directly represent an object by a matrix. However, this requires a huge number of parameters. In this paper, we propose to embed objects into a D-dimensional quaternion space. Thanks to the algebraic and geometric properties of quaternions, we are able to find a more efficient way to represent a matrix. To be more specific, a quaternion $q = w + x \cdot i + y \cdot j + z \cdot k$ corresponds to at least a 4×4 matrix and a 3×3 matrix. In this sense, these matrices only require 4 parameters. Therefore, a D-dimensional quaternion suffices to express D semantic subspaces in a parsimonious manner.

In terms of the second problem, on top of the quaternions, we only need to find a distance function for quaternion embeddings. Traditional methods [Zhang *et al.*, 2019] project the quaternion embeddings into the high-dimensional Euclidean space and then adopt the Euclidean distance. Obviously, this strategy loses the algebraic and geometric properties of quaternions. Hence, it may not utilize the semantic subspace well. To overcome this limitation, inspired by the chordal Grassmannian distance [Bengtsson *et al.*, 2007], we define a new distance function for the matrices corresponding to quaternions rather than the quaternions vectors. As a key trait, the proposed distance function could measure the similarity between two sets of matrices, encapsulating multiple semantic subspaces into a unified measurement.

We summarize the contributions of this work as follows:

- We present an early trial to reformulate the ordinal embedding problem in a quaternion space with matrix-based representations. Using matrix-based representation learning, we model an object as a set of subspaces with each subspace implying an inherent concept in the similarity comparisons.
- Inspired by chordal Grassmannian distance, a new distance function is proposed to measure the similarity for matrix-based representations established from quaternions.
- Experiments on both simulated and real datasets demonstrate that the proposed method achieves a convincing improvement over existing vector-based methods.

2 Related Work

2.1 Ordinal Embedding

In recent years, the ordinal embedding in the Euclidean space has been extensively studied. This problem is first studied by [Shepard, 1962a; Shepard, 1962b; Kruskal, 1964a; Kruskal, 1964b]. Subsequently, under a margin-loss-based setting, GNMDS [Agarwal *et al.*, 2007] find a low-rank kernel matrix to make embedding satisfy the triplet constraints. Based on the probabilistic model setting, CLK [Tamuz *et al.*, 2011] samples responses to adaptively choose triplet-based relative-similarity queries. STE and t-STE [Van Der Maaten and Weinberger, 2012], assuming the Bradley-TerryLuce noise model [Luce, 2012], think relative-similarity triplets are insufficient for obtaining a truthful embedding of objects, where the embeddings are learned from both relatively similar and dissimilar triplets instead. Since the similarity of objects would not be consistent with different tasks, POE [McFee and Lanckriet, 2011] integrates heterogeneous data to conform to the similarity measure optimally. SPE [Shaw and Jebara, 2009] and LOE [Terada and Luxburg, 2014] are proposed to generate embedding that preserves the ordinal constraints and the dataset's density structure simultaneously. Meanwhile, SVRG-SBB [Ma *et al.*, 2019] accelerates the calculation of the ordinal embedding process by stochastic variance reduced gradient. However, all the above methods consider OE in the Euclidean space. HOE [Suzuki *et al.*, 2019] is the first to extend the problem to the hyperbolic space, which can express hierarchical structure more effectively. Two types of ordinal embedding methods with different embedding spaces are summarized in Appendix C.

2.2 Distance Function

Generally speaking, there are three kinds of distance functions defined on point to point, point to subspace, and subspace to subspace respectively. Point to point distance can be denoted by $d(x_1, x_2)$, the distance between two points x_1 and x_2, usually called Euclidean distance. Point to subspace distance can be denoted by $d(x, S)$, the distance from point x to the subspace S, which is called L2-Hausdorff distance in [Moghaddam and Pentland, 1997; Turk and Pentland, 1991]. Subspace to subspace distance can be denoted by $d(S_1, S_2)$, the distance between two subspaces S_1 and S_2. The principal angle is a commonly used concept in subspaces, which has a good performance [Nishiyama *et al.*, 2005; Wolf and Shashua, 2004]. [Sun and Cheng, 2006] gives the relationship between the similarity and distance of the subspaces, designing a general construction framework of subspace. [Zuccon *et al.*, 2009] analyzes the semantic difference between subspace distance and Euclidean distance in information retrieval. As for applications, subspace distance is widely used in many computer vision fields such as classification [Liu *et al.*, 2011], clustering [Wang *et al.*, 2016], and face recognition [Wang *et al.*, 2008; Pang *et al.*, 2009], which all achieve promising performance.

3 Methodology

In this section, we describe our proposed method in detail. As shown in Fig. 2, we try to model each object with a set of subspaces/matrices. We start by elaborating on the problem setting. Then introduce how to get matrix-based representation in the quaternion space and define a distance function to measure the similarity between two objects' representations.

3.1 Problem Setting

Given N objects and the ordinal relationship triplets

$$\mathcal{S} = \{(a, b, c) \mid a, b, c \in [N],\ a \neq b \neq c\},$$

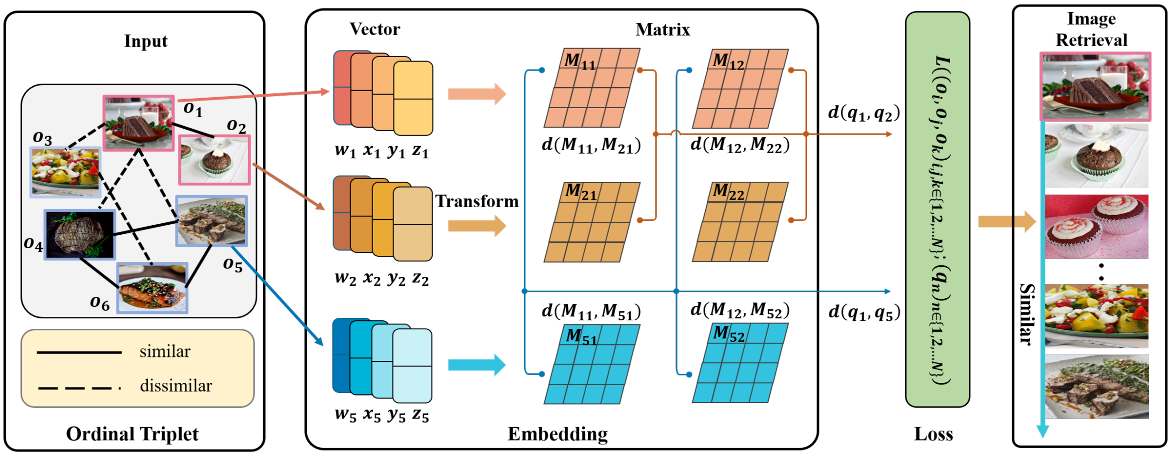


Figure 2: The framework of our proposed method. We first embed objects in the ordinal triplets into the quaternion space, then transform the quaternion vectors to collections of matrices. We aim to make the obtained quaternion embeddings satisfy the ordinal constraints as much as possible. Finally, we apply them for image retrieval tasks. For example, when finding the similar food of Chocolate Cake with the pink edge, we get the Chocolate Muffins with similar ingredients and Cupcakes with similar taste shown on the left of the figure.

where $[N] = \{1, \dots, N\}$ and object a is more similar to object b than it is to object c. Given some distance function $d(\cdot, \cdot)$, the OE problem aims to learn the representations of objects, denoted as $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N\}$, such that the following objectives hold as much as possible:

$$d(\boldsymbol{x}_a, \boldsymbol{x}_b) < d(\boldsymbol{x}_a, \boldsymbol{x}_c), \forall\ (a, b, c) \in \mathcal{S}.$$

3.2 Representations in Quaternion Space

Existing OE methods like EOE[Agarwal *et al.*, 2007; Terada and Luxburg, 2014] and HOE[Suzuki *et al.*, 2019] represent the objects as vectors. However, as is stated in Sec. 1, traditional methods might be insufficient to capture latent semantic information. To overcome the limitation, in this paper, we adopt quaternions to represent an object as a set of subspaces.

A quaternion vector consists of one real part and three imaginary parts, defined as $\boldsymbol{q} = \boldsymbol{w} + \boldsymbol{x} \cdot \boldsymbol{i} + \boldsymbol{y} \cdot \boldsymbol{j} + \boldsymbol{z} \cdot \boldsymbol{k}$, where $\boldsymbol{w}, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$ are D dimensional real vectors and $\boldsymbol{i}$, $\boldsymbol{j}$, $\boldsymbol{k}$ are imaginary units that satisfy the following Hamilton's rules:

$$\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = \boldsymbol{ijk} = -1,$$
$$\boldsymbol{ij} = \boldsymbol{k} = -\boldsymbol{ji},\ \boldsymbol{jk} = \boldsymbol{i} = -\boldsymbol{kj},\ \boldsymbol{ki} = \boldsymbol{j} = -\boldsymbol{ik}.$$

In the following two subsections, we show that a quaternion corresponds to at least two matrices. One is a 4 × 4 matrix termed as representation matrix, the other one is a 3 × 3 matrix called rotation matrix.

Representation Matrix of a Quaternion

Let w_i, x_i, y_i, z_i be the i-th dimension of $\boldsymbol{w}, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$ respectively. Then quaternion $\boldsymbol{Q}_i = w_i + x_i \cdot i + y_i \cdot j + z_i \cdot k$ is associated with a 4 × 4 matrix as follows:

$$\boldsymbol{Q}_i = \begin{bmatrix} w_i & -x_i & -y_i & -z_i \\ x_i & w_i & -z_i & y_i \\ y_i & z_i & w_i & -x_i \\ z_i & -y_i & x_i & w_i \end{bmatrix}. \qquad (1)$$

We term this matrix form as the representation matrix of a quaternion. Interestingly, this matrix formulation preserves many algebraic properties of quaternions, such as: the quaternion addition and multiplication correspond to matrix addition and matrix multiplication; the conjugate of $\boldsymbol{Q}_i$ corresponds to the transpose of $\boldsymbol{Q}_i$; $\boldsymbol{Q}_i$ could be projected onto four matrix basis with projected values being w_i, x_i, y_i, z_i, respectively, and the four matrix basis satisfying similar Hamilton's rules (expect that the multiplication of three skew-symmetric matrix basis equals to the identity matrix $-\boldsymbol{I}$).

Besides, from a representation matrix $\boldsymbol{Q}_i$, we could easily obtain the original quaternion $\boldsymbol{Q}_i$. Seeing the above good properties, we consider the representation matrix as a candidate matrix formulation of quaternions. Next, we continue to derive another matrix formulation.

Rotation Matrix of a Quaternion

It is noteworthy that a quaternion rotation $p' = \boldsymbol{Q}_i p \boldsymbol{Q}_i^{-1}$ can be algebraically manipulated into a matrix rotation $p' = \boldsymbol{R}_i p$, where $\boldsymbol{R}_i$ is the rotation matrix given by:

$$\boldsymbol{R}_i = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}. \qquad (2)$$

where

$$\begin{aligned} r_{11} &= {w_i}^2 + {x_i}^2 - {y_i}^2 - {z_i}^2 \\ r_{12} &= 2x_i y_i - 2w_i z_i, \\ r_{13} &= 2x_i z_i + 2w_i y_i, \\ r_{21} &= 2x_i y_i + 2w_i z_i, \\ r_{22} &= {w_i}^2 - {x_i}^2 + {y_i}^2 - {z_i}^2, \\ r_{23} &= 2y_i z_i - 2w_i z_i, \\ r_{31} &= 2x_i z_i - 2w_i y_i, \\ r_{32} &= 2y_i z_i + 2w_i z_i, \\ r_{33} &= {w_i}^2 - {x_i}^2 - {y_i}^2 + {z_i}^2. \end{aligned}$$

This matrix form preserves the geometric properties of quaternions. Besides, according to [Bar-Itzhack, 2000; Markley, 2008], the original quaternion $\boldsymbol{Q}_i$ can also be restored from the rotation matrix $\boldsymbol{R}_i$.

3.3 Distance between Two Quaternions

Next, we elaborate on how to calculate the distance between two quaternion embeddings. Traditional methods [Zhang *et al.*, 2019] project the quaternion embedding into the high-dimensional Euclidean space. Specifically, given a quaternion vector $\boldsymbol{q} \in \mathbb{R}^{D\times 4}$, we can obtain a vector $\tilde{\boldsymbol{q}} = \{\boldsymbol{w}, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}\} \in \mathbb{R}^{4D}$. Then the distance between two quaternion vectors $\tilde{\boldsymbol{q}}_1, \tilde{\boldsymbol{q}}_2$ can be calculated by the inner product, i.e., $d_E(\tilde{\boldsymbol{q}}_1, \tilde{\boldsymbol{q}}_2) = \sqrt{(\tilde{\boldsymbol{q}}_1 - \tilde{\boldsymbol{q}}_2)^T(\tilde{\boldsymbol{q}}_1 - \tilde{\boldsymbol{q}}_2)}$. Straightforward as it seems to be, such a distance function makes the quaternion space degenerate into the Euclidean space.

Remind that we aim to represent each object with a set of subspaces so that the distance should reflect the distance between two subspaces. We know that a matrix could span a linear subspace. Then it is a natural idea to measure the distance of two quaternion embeddings based on their matrix reformulation.

Then how to calculate the distance between two matrices? In this paper, inspired by the classic chordal Grassmannian distance [Bengtsson *et al.*, 2007], we proposed a distance function to calculate the distance between two quaternion embeddings. The chordal Grassmannian distance is defined as follows

$$d(\boldsymbol{A}, \boldsymbol{B}) = \sqrt{m - tr(\boldsymbol{AB})},\ \boldsymbol{A} \in \mathbb{O}^m,\ \boldsymbol{B} \in \mathbb{O}^m \tag{3}$$

where $\boldsymbol{A}$ and $\boldsymbol{B}$ denote a pair of matrices, $tr(\cdot)$ is the trace of a matrix, m is the dimension of the matrix, and $\mathbb{O}^m$ denotes the set of all $m \times m$ orthogonal matrices.

Herein, we would like to point out that the representation matrix $\boldsymbol{Q}_i$ and the rotation matrix $\boldsymbol{R}_i$ are both orthogonal matrices as long as $w_i^2 + x_i^2 + y_i^2 + z_i^2 = 1$ for a quaternion $q_i = w_i + x_i \cdot i + y_i \cdot j + z_i \cdot k$. On top of this, our distance function is based on the chordal Grassmannian distance. For each D-dimension quaternion embedding $\boldsymbol{q} = \{q_1, \ldots, q_i, \ldots, q_D\}$, we could obtain D 4×4 representation matrices $\{\boldsymbol{Q}_1, \cdots, \boldsymbol{Q}_i, \cdots, \boldsymbol{Q}_D\}$ and D 3×3 rotation matrices $\{\boldsymbol{R}_1, \cdots, \boldsymbol{R}_i, \cdots, \boldsymbol{R}_D\}$. Let $\boldsymbol{M}_i \in \mathbb{R}^{m\times m}$ denote either $\boldsymbol{Q}_i$ $(m = 4)$ or $\boldsymbol{R}_i (m = 3)$. We could construct a new diagonal matrices $\mathcal{M}$ with $\boldsymbol{M}_1, \boldsymbol{M}_2, ..., \boldsymbol{M}_D$ as the diagonal sub-matrices, i.e.,

$$\mathcal{M} = \begin{bmatrix} \boldsymbol{M}_1 & \ldots & \ldots & \boldsymbol{0} \\ \vdots & \boldsymbol{M}_2 & & \vdots \\ \vdots & & \ddots & \vdots \\ \boldsymbol{0} & \ldots & \ldots & \boldsymbol{M}_D \end{bmatrix}_{mD\times mD}, \tag{4}$$

where the omitted parts are all zero matrices. The distance function between two quaternion embeddings $\boldsymbol{q}_a$ and $\boldsymbol{q}_b$ is defined as follows

$$\begin{aligned} d_Q(\boldsymbol{q}_a, \boldsymbol{q}_b) &= \sqrt{mD - tr(\mathcal{M}^a\mathcal{M}^b)} \\ &= \sqrt{mD - \sum\nolimits_{i=1}^{D} tr(\boldsymbol{M}_i^a\boldsymbol{M}_i^b)}, \end{aligned} \tag{5}$$

where $\boldsymbol{M}_i^a$ is the i-th diagonal sub-matrices of the matrix $\mathcal{M}^a$ corresponding to the quaternion vector $\boldsymbol{q}_a$.

3.4 Optimization Objective

On top of the quaternion embeddings and the developed distance function, our optimization objective is to learn the representations of objects $\{\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_N\}$, such that:

$$d_Q(\boldsymbol{x}_a, \boldsymbol{x}_b) < d_Q(\boldsymbol{x}_a, \boldsymbol{x}_c), \forall\ (a, b, c) \in \mathcal{S}.$$

hold as much as possible. Let ℓ be the loss function of one triplet. Then the loss function is developed as:

$$\mathcal{L} = \frac{1}{|\mathcal{S}|} \sum_{(a,b,c)\in\mathcal{S}} \ell\Big(\{\boldsymbol{q}_a, \boldsymbol{q}_b, \boldsymbol{q}_c\}; d_Q\Big), \tag{6}$$

where $|\mathcal{S}|$ is the number of triplets. In the experiments, we implement different loss functions such as the GNMDS loss function [Agarwal *et al.*, 2007; Terada and Luxburg, 2014] and the CKL loss function [Tamuz *et al.*, 2011]. The readers are referred to Appendix A for more details.

4 Experiments

In this section, we conduct extensive experiments to verify the effectiveness of our proposed method on one synthetic dataset and three real-world datasets. We show a detailed description of the four datasets in the Appendix B.

4.1 Synthetic Dataset

Evaluation metrics. We adopt testing error as our evaluation metric, which is defined as the ratio of the wrongly predicted triples in the test set. For sake of better credibility of our experiments, we also provide the mean and standard deviation of the test error.

Competitors. We compare our method against two types of vector-based ordinal embedding: one is in the Euclidean space (EOE [Agarwal *et al.*, 2007]) and other one is in the Hyperbolic space (HOE [Suzuki *et al.*, 2019]). To further demonstrate the effectiveness of our proposed QOE, we apply five different loss functions (see Appendix A) on top of EOE, HOE and QOE (ours), respectively. More specifically, the five losses include: **1) GNMDS**$_1$: GNMDS loss function [Terada and Luxburg, 2014] with $q = 1$; **2) GNMDS**$_2$: GNMDS loss function with $q = 2$; **3) CKL** [Tamuz *et al.*, 2011]; **4) STE** [Van Der Maaten and Weinberger, 2012]; **5) t-STE**. **Ours**$_1$ and **Ours**$_2$ represent QOE based on the representation matrix $\boldsymbol{Q}$ in Eq.(1), and the rotation matrix $\boldsymbol{R}$ in Eq.(2), respectively. For HOE and Ours, we report the best results on five loss functions.

Implementation details. To ensure a similar total number of parameters to make fair comparisons, when the dimension of real vectors in EOE or HOE is D, we set the dimension of quaternion vectors is $D/4$. We conduct experiments for $D = \{4, 8, 16\}$. We apply random initialization for embedding and choose Adam [Kingma and Ba, 2015] as optimizer. The batch size is set to 512, learning rating $\lambda = 0.1$ and the number of epochs is set to 200 for all methods. During training, if the optimal results of several epochs are not updated, we will end the training early. The experimental environment is shown in the appendix D. All experiments are conducted on Ubuntu 16.04.6 LTS, with NVIDIA TITAN RTX. The algorithm is written in Python 3.6.8 and uses the deep learning framework called TensorFlow 1.14.

		D=4	D=8	D=16
EOE	GNMDS$_1$	0.2769 ± 0.0027	0.2638 ± 0.0014	0.2573 ± 0.0015
	GNMDS$_2$	0.4233 ± 0.0066	0.3317 ± 0.0021	0.3112 ± 0.0019
	CKL	0.2744 ± 0.0010	0.2713 ± 0.0028	0.2706 ± 0.0003
	STE	0.3380 ± 0.0050	0.3199 ± 0.0130	0.3099 ± 0.0044
	t-STE	0.2644 ± 0.0016	0.2461 ± 0.0034	0.2367 ± 0.0006
HOE		0.2839 ± 0.0020	0.2574 ± 0.0051	0.2259 ± 0.0214
	Ours$_1$ w/o dis	0.2770 ± 0.0013	0.2684 ± 0.0069	0.2382 ± 0.0015
	Ours$_1$	0.2756 ± 0.0010	0.2588 ± 0.0071	0.1672 ± 0.0012
	Ours$_2$ w/o dis	0.2643 ± 0.0010	0.2368 ± 0.0073	0.1762 ± 0.0014
	Ours$_2$	0.2007 ± 0.0003	0.1322 ± 0.0020	0.0714 ± 0.0012

Table 1: Testing errors (mean ± standard deviation) on synthetic dataset. The best method is marked as red, and the second is marked as blue.

		D=4	D=8	D=16
EOE	GNMDS$_1$	0.2323 ± 0.0012	0.2087 ± 0.0005	0.2028 ± 0.0003
	GNMDS$_2$	0.2244 ± 0.0004	0.2084 ± 0.0007	0.2010 ± 0.0002
	CKL	0.2766 ± 0.0001	0.2532 ± 0.0002	0.2516 ± 0.0020
	STE	0.2239 ± 0.0014	0.1953 ± 0.0005	0.1805 ± 0.0003
	t-STE	0.2896 ± 0.0018	0.2736 ± 0.0011	0.2592 ± 0.0005
HOE		0.2572 ± 0.0013	0.2153 ± 0.0007	0.1985 ± 0.0005
	Ours$_1$ w/o dis	0.2847 ± 0.0093	0.2076 ± 0.0053	0.2011 ± 0.0019
	Ours$_1$	0.3148 ± 0.0114	0.2032 ± 0.0044	0.1445 ± 0.0017
	Ours$_2$ w/o dis	0.2294 ± 0.0017	0.2042 ± 0.0012	0.1739 ± 0.0012
	Ours$_2$	0.2095 ± 0.0013	0.1771 ± 0.0013	0.1444 ± 0.0005

Table 2: Testing errors (mean ± standard deviation) on Music Artists dataset.

Results. The results are presented in Tab. 1, from which we can observe that Our QOE achieves a clear margin in performance gain over all baselines on three different dimensions. This shows that the matrix-based representation can better model ordinal relationships and better capture the latent semantic information of objects than vector-based representation methods. Besides, Ours$_2$ obtains the minimum error in most cases while Ours$_1$ is the second best in most cases. This shows that the rotation matrix may better model the inherent properties of quaternions than the representation matrix.

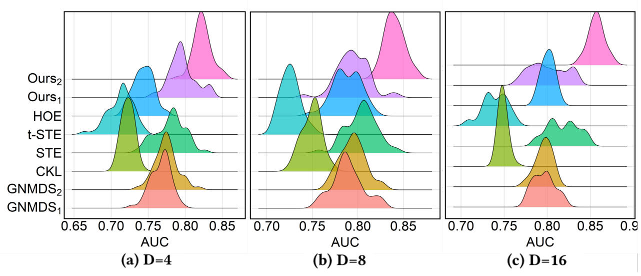
4.2 Music Artists Dataset

Implementation details. The batch size 1,024, learning rating $\lambda = 0.5$ and the epochs number 100 are used for all methods. During training, if the optimal results of several epochs are not updated, we will end the training early. The other settings are the same as setting on the synthetic dataset.

Results. As shown in Tab. 2, our proposed method also outperforms competitors. In particular, Ours$_2$ improves the best competitor's performance by 6.43%, 9.31%, 20.00% on three dimensions, respectively. We also report AUC (the other datasets do not have class labels, thus we can not calculate AUC on them), which is shown in Fig. 3, from which we can see our proposed method also outperforms the competitors. Therefore, the effectiveness of matrix-based representation is again validated.

4.3 Food Dataset

Implementation details. We set the batch size to $1,024$, the learning rating $\lambda = 0.05$ and the number of epochs to 80 for all methods. Other settings are the same as the synthetic dataset.



(a) D=4 (b) D=8 (c) D=16

Figure 3: The AUC of different methods on Music dataset.

		D=4	D=8	D=16
EOE	GNMDS$_1$	0.4488 ± 0.0127	0.4247 ± 0.0139	0.4165 ± 0.0115
	GNMDS$_2$	0.4467 ± 0.0144	0.4337 ± 0.0134	0.4281 ± 0.0094
	CKL	0.4033 ± 0.0094	0.3833 ± 0.0125	0.3040 ± 0.0008
	STE	0.4488 ± 0.0127	0.3850 ± 0.0161	0.3782 ± 0.0149
	t-STE	0.4007 ± 0.0163	0.3792 ± 0.0092	0.3382 ± 0.0144
HOE		0.4271 ± 0.0128	0.4037 ± 0.0140	0.3793 ± 0.0111
	Ours$_1$ w/o dis	0.3831 ± 0.0100	0.3694 ± 0.0095	0.3295 ± 0.0132
	Ours$_1$	0.3221 ± 0.0188	0.2097 ± 0.0113	0.1777 ± 0.0108
	Ours$_2$ w/o dis	0.3789 ± 0.0179	0.3643 ± 0.0145	0.3285 ± 0.0127
	Ours$_2$	0.2792 ± 0.0202	0.2043 ± 0.0075	0.1740 ± 0.0095

Table 3: Testing errors (mean ± standard deviation) on Food dataset.

		D=4	D=8	D=16
EOE	GNMDS$_1$	0.2953 ± 0.0045	0.2840 ± 0.0055	0.2784 ± 0.0020
	GNMDS$_2$	0.3301 ± 0.0352	0.3261 ± 0.0029	0.3227 ± 0.0051
	CKL	0.2791 ± 0.0005	0.2742 ± 0.0005	0.2759 ± 0.0002
	STE	0.2703 ± 0.0020	0.2531 ± 0.0021	0.2470 ± 0.0025
	t-STE	0.3255 ± 0.0065	0.3156 ± 0.0036	0.3078 ± 0.0044
HOE		0.3896 ± 0.0193	0.2791 ± 0.0047	0.2606 ± 0.0029
	Ours$_1$ w/o dis	0.2821 ± 0.0188	0.2642 ± 0.0099	0.2589 ± 0.0131
	Ours$_1$	0.2608 ± 0.0123	0.2454 ± 0.0089	0.2427 ± 0.0027
	Ours$_2$ w/o dis	0.2749 ± 0.0188	0.2458 ± 0.0161	0.2389 ± 0.0064
	Ours$_2$	0.1706 ± 0.0089	0.1513 ± 0.0073	0.1422 ± 0.0043

Table 4: Testing errors (mean ± standard deviation) on Bird dataset.

Results. The testing errors for split ratio of 0.3 are recorded in Tab. 3. It shows that our proposed method still consistently outperforms all baselines over test error metric. Specifically, Ours$_2$ outperforms the best competitor by 30.32%, 46.12%, 42.76% on three dimensions, respectively. As the proportion of the test set increases, the errors of each method are increasing. It is easy to understand that the smaller training set will result in over fitting, and embedding cannot satisfy the ordinal constraints well. This shows that QOE is more robust.

4.4 Bird Dataset

The testing errors of bird dataset are recorded in Tab. 4. It shows that our proposed method still consistently outperforms both baselines over testing errors metric. Specifically, Ours$_2$ outperforms the best competitor by 36.88%, 40.22%, 42.42% on three dimensions, respectively. The clear margin in performance gain validates the effectiveness of the proposed QOE methods.

	matrix(3×3)	ours$_1(D=8)$	ours$_2(D=8)$
GNMDS$_1$	0.2284 ± 0.0039	0.2558 ± 0.0013	0.1436 ± 0.0009
GNMDS$_2$	0.2445 ± 0.0018	0.2035 ± 0.0021	0.1589 ± 0.0008
STE	0.2269 ± 0.0045	0.1644 ± 0.0040	0.1698 ± 0.0046
t-STE	0.2581 ± 0.0086	0.2032 ± 0.0044	0.1771 ± 0.0013

Table 5: Testing errors on Music Artists dataset for QOE ($D = 8$) and random matrix (the size is 3×3).

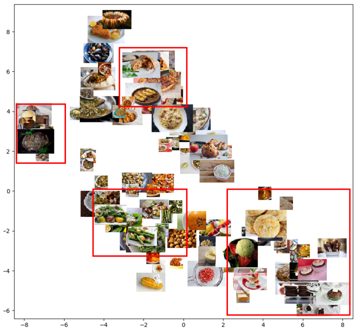


Figure 4: T-SNE of the learned embeddings on Food dataset.

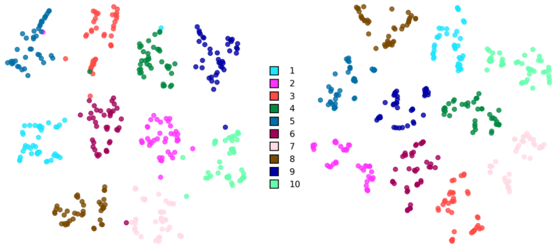
4.5 Embedding Visualization

We visualize the embeddings learned by our methods on Food dataset and Bird dataset, as shown in Fig. 4 and Fig. 5. For Food dataset, from Fig. 4, we see that the learned embeddings shows good clustering behavior with desserts gathered into the bottom right. The meats are close to each other into the left, as are the salads into the bottom middle and baking goods into top left. As for the Bird dataset, from Fig. 5 we see that the learned embeddings also cluster into different groups and each cluster represent category of birds. The classification effect shown in both figures demonstrate that the proposed QOE method is able capture semantic meanings thanks to the proper-structured matrix-based representations.

4.6 Ablation Study

We compare the performance of using quaternions (Ours) and using random matrix. To ensure that the total number of parameters is similar, we compare between 3×3 random-initialized matrices and 2-dimensional quaternion vectors ($D = 8$). The results are shown in Tab. 5. As shown in Tab. 5, in most cases, QOE performs better than using random matrices, which shows that our QOE method is both efficient and effective.

Furthermore, we split datasets into training/test sets by different split ratios $[0.2, 0.3, ..., 0.8]$, where the split ratio is equal to the proportion of the test set to the total dataset. For each split ratio, several experiments are performed and we report the mean and standard deviation of testing errors, as shown in Fig. 6. From the figure, as the proportion of the test set increases, the errors of each method are increasing. However, our QOE is shown to perform better when there is less training samples, which shows that our method is more robust to over-fitness and has better generalization performance.



(b) Ours$_1$ **(c) Ours$_2$**

Figure 5: Embeddings of the Bird dataset constructed by the Ours$_1$ and Ours$_2$ with $D = 4$ and GNMDS$_1$ loss function. The $1 - 10$ in the legend represent different categories of birds.

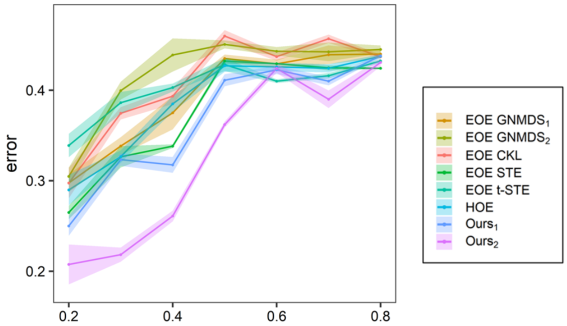


Figure 6: The testing error on Food dataset with different split ratios with $D = 4$.

5 Conclusion

In this work, we propose a matrix-based representation learning to model the similarity relationships among objects. Specifically, we adopt the quaternion space as the embedded space, where every quaternion vector corresponding to an object can be converted into a collection of matrices, which spans a subspace of the embedded space. Furthermore, inspired by the classic chordal Grassmannian distance, a new distance function is defined to measure the similarity of different subspaces. We also conduct experiments on synthetic and several real datasets to verify our proposed method's effectiveness, which achieve promising experimental results.

Ethical Statement

There are no ethical issues.

Acknowledgements

This work was supported in part by the National Key R&D Program of China under Grant 2018AAA0102000, in part by National Natural Science Foundation of China: U21B2038, 61931008, 61836002, 6212200758, 61976202 and 62006217, in part by the Fundamental Research Funds for the Central Universities, in part by Youth Innovation Promotion Association CAS, in part by the Strategic Priority Research Program of Chinese Academy of Sciences, Grant No. XDB28000000, and in part by China Postdoctoral Science Foundation 2021T140653 and 2020M680651.

References

[Agarwal *et al.*, 2007] Sameer Agarwal, Josh Wills, Lawrence Cayton, Gert Lanckriet, David Kriegman, and Serge Belongie. Generalized non-metric multidimensional scaling. In *AIS*, pages 11–18, 2007.

[Bar-Itzhack, 2000] Itzhack Y Bar-Itzhack. New method for extracting the quaternion from a rotation matrix. *JGCD*, 23:1085–1087, 2000.

[Bengtsson *et al.*, 2007] Ingemar Bengtsson, Wojciech Bruzda, Åsa Ericsson, Jan-Åke Larsson, Wojciech Tadej, and Karol Życzkowski. Mutually unbiased bases and hadamard matrices of order six. *J Math Phys*, 48:052106, 2007.

[Kingma and Ba, 2015] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

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

[Kruskal, 1964b] Joseph B Kruskal. Nonmetric multidimensional scaling: a numerical method. *Psychometrika*, 29:115–129, 1964.

[Liu *et al.*, 2011] Yiguang Liu, Shuzhi Sam Ge, Chunguang Li, and Zhisheng You. k-ns: A classifier by the distance to the nearest subspace. *IEEE Trans*, 22:1256–1268, 2011.

[Luce, 2012] R Duncan Luce. *Individual choice behavior: A theoretical analysis*. Courier Corporation, 2012.

[Ma *et al.*, 2019] Ke Ma, Qianqian Xu, Zhiyong Yang, and Xiaochun Cao. Less but better: Generalization enhancement of ordinal embedding via distributional margin. In *AAAI*, pages 2978–2985, 2019.

[Markley, 2008] F Landis Markley. Unit quaternion from rotation matrix. *JGCD*, 31:440–442, 2008.

[McFee and Lanckriet, 2011] Brian McFee and Gert Lanckriet. Learning multi-modal similarity. *JMLR*, 12:491–523, 2011.

[Moghaddam and Pentland, 1997] Baback Moghaddam and Alex Pentland. Probabilistic visual learning for object representation. *IEEE TPAMI*, 19:696–710, 1997.

[Nishiyama *et al.*, 2005] Masashi Nishiyama, Osamu Yamaguchi, and Kazuhiro Fukui. Face recognition with the multiple constrained mutual subspace method. In *AVBPA*, pages 71–80, 2005.

[Pang *et al.*, 2009] Y. Pang, Y. Yuan, and X. Li. Iterative subspace analysis based on feature line distance. *IEEE TIP*, 18:903–907, 2009.

[Shaw and Jebara, 2009] Blake Shaw and Tony Jebara. Structure preserving embedding. In *ICML*, pages 937–944, 2009.

[Shepard, 1962a] Roger N Shepard. The analysis of proximities: multidimensional scaling with an unknown distance function. i. *Psychometrika*, 27:125–140, 1962.

[Shepard, 1962b] Roger N Shepard. The analysis of proximities: Multidimensional scaling with an unknown distance function. ii. *Psychometrika*, 27:219–246, 1962.

[Sun and Cheng, 2006] Xichen Sun and Qiansheng Cheng. On subspace distance. In *ICIAR*, pages 81–89, 2006.

[Suzuki *et al.*, 2019] Atsushi Suzuki, Jing Wang, Feng Tian, Atsushi Nitanda, and Kenji Yamanishi. Hyperbolic ordinal embedding. In *ACML*, pages 1065–1080, 2019.

[Tamuz *et al.*, 2011] Omer Tamuz, Ce Liu, Serge Belongie, Ohad Shamir, and Adam Tauman Kalai. Adaptively learning the crowd kernel. In *ICML*, pages 673–680, 2011.

[Terada and Luxburg, 2014] Yoshikazu Terada and Ulrike Luxburg. Local ordinal embedding. In *ICML*, pages 847–855, 2014.

[Turk and Pentland, 1991] Matthew A Turk and Alex P Pentland. Face recognition using eigenfaces. In *CVPR*, pages 586–587, 1991.

[Van Der Maaten and Weinberger, 2012] Laurens Van Der Maaten and Kilian Weinberger. Stochastic triplet embedding. In *MLSP*, pages 1–6, 2012.

[Wang *et al.*, 2008] Ruiping Wang, Shiguang Shan, Xilin Chen, and Wen Gao. Manifold-manifold distance with application to face recognition based. *CVPR*, pages 1–8, 2008.

[Wang *et al.*, 2016] Jun Wang, Zhaohong Deng, Kup-Sze Choi, Yizhang Jiang, Xiaoqing Luo, Fu-Lai Chung, and Shitong Wang. Distance metric learning for soft subspace clustering in composite kernel space. *PR*, 52:113–134, 2016.

[Wolf and Shashua, 2004] Lior Wolf and Amnon Shashua. Learning over sets using kernel principal angles. *JMLR*, 4:913–931, 2004.

[Zhang *et al.*, 2019] Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. Quaternion knowledge graph embeddings. In *NIPS*, pages 2731–2741, 2019.

[Zuccon *et al.*, 2009] Guido Zuccon, Leif A Azzopardi, and CJ Van Rijsbergen. Semantic spaces: Measuring the distance between different subspaces. In *ISQI*, pages 225–236, 2009.