Quaternion Ordinal Embedding

Wenzheng Hou\textsuperscript{1,2}, Qianqian Xu\textsuperscript{1*}, Ke Ma\textsuperscript{2}, Qianxiu Hao\textsuperscript{1,2} and Qingming Huang\textsuperscript{1,2,3,4*}
\textsuperscript{1}Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, CAS
\textsuperscript{2}School of Computer Science and Technology, University of Chinese Academy of Sciences
\textsuperscript{3}Key Laboratory of Big Data Mining and Knowledge Management, Chinese Academy of Sciences
\textsuperscript{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.

Most of the existing ordinal embedding methods pursue the \textit{vector-based} representations of objects. Specifically, denote $x_a$, $x_b$ and $x_c$ 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 $x_a$ and $x_b$ together while pull $x_a$ and $x_c$ 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, \textit{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 $4 \times 4$ matrix and a $3 \times 3$ 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, GNMD [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-Terry-Luce 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

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

Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22)
where \([N] = \{1, \ldots, N\}\) 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 \([x_1, x_2, \ldots, x_N]\), such that the following objectives hold as much as possible:

\[
d(x_a, x_b) < d(x_a, x_c), \forall (a, b, c) \in S.
\]

### 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 \(q = w + x \cdot i + y \cdot j + z \cdot k\), where \(w, x, y, z\) are \(D\) dimensional real vectors and \(i, j, k\) are imaginary unit vectors that satisfy the following Hamilton’s rules:

\[
i^2 = j^2 = k^2 = ijk = -1,
ij = k = -ji, \ jk = i = -kj, \ ki = j = -ik.
\]

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

### Representation Matrix of a Quaternion

Let \(w_i, x_i, y_i, z_i\) be the \(i\)-th dimension of \(w, x, y, z\) respectively. Then quaternion \(Q_i = w_i + x_i \cdot i + y_i \cdot j + z_i \cdot k\) is associated with a \(4 \times 4\) matrix as follows:

\[
Q_i = \begin{bmatrix}
w_i & -x_i & -y_i & -z_i \\
x_i & w_i & -z_i & y_i \\
y_i & z_i & w_i & -x_i \\
z_i & -y_i & x_i & w_i
\end{bmatrix}.
\]

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 \(Q_i\) corresponds to the transpose of \(Q_i\); \(Q_i\) 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 \(-I\)).

Besides, from a representation matrix \(Q_i\), we could easily obtain the original quaternion \(Q_i\). 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 \(p' = Q_i p Q_i^{-1}\) can be algebraically manipulated into a matrix rotation \(p' = R_i p\), where \(R_i\) is the rotation matrix given by:

\[
R_i = \begin{bmatrix}
r_{11} & r_{12} & r_{13} \\
r_{21} & r_{22} & r_{23} \\
r_{31} & r_{32} & r_{33}
\end{bmatrix}
\]

where

\[
\begin{align*}
r_{11} &= w_i^2 + x_i^2 - y_i^2 - z_i^2 \\
r_{12} &= 2x_i y_i - 2w_i z_i \\
r_{13} &= 2x_i z_i + 2y_i y_i \\
r_{21} &= 2x_i y_i + 2w_i z_i \\
r_{22} &= w_i^2 - x_i^2 + y_i^2 - z_i^2 \\
r_{23} &= 2y_i z_i - 2w_i x_i \\
r_{31} &= 2x_i z_i - 2w_i y_i \\
r_{32} &= 2y_i z_i + 2w_i x_i \\
r_{33} &= w_i^2 - x_i^2 - y_i^2 + z_i^2.
\end{align*}
\]

This matrix form preserves the geometric properties of quaternions. Besides, according to [Bar-Itzhack, 2000; Markley, 2008], the original quaternion \(Q_i\) can also be restored from the rotation matrix \(R_i\).
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 $q \in \mathbb{R}^{D \times 4}$, we can obtain a vector $\tilde{q} = \{ w, x, y, z \} \in \mathbb{R}^{4D}$. Then the distance between two quaternion vectors $q_1, q_2$ can be calculated by the inner product, i.e., $d_Q(q_1, q_2) = \sqrt{(q_1 - q_2)^T(q_1 - q_2)}$. 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

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

where $A$ and $B$ denote a pair of matrices, $\text{tr}(\cdot)$ 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 $Q_i$ and the rotation matrix $R_i$ are both orthogonal matrices as long as $w^2 + x^2 + y^2 + z^2 = 1$ 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 $q = \{ q_1, \ldots, q_D \}$, we could obtain $D \times 4 \times 4$ representation matrices $\{ Q_1, \ldots, Q_D \}$ and $D \times 3 \times 3$ rotation matrices $\{ R_1, \ldots, R_D \}$. Let $M_i \in \mathbb{O}^{m \times m}$ denote either $Q_i$ ($m = 4$) or $R_i$ ($m = 3$). We could construct a new diagonal matrices $\mathcal{M}$ with $M_1, M_2, \ldots, M_D$ as the diagonal sub-matrices, i.e.,

$$\mathcal{M} = \begin{bmatrix} M_1 & & & 0 \\ & M_2 & & \\ & & \ddots & \\ 0 & & & M_D \end{bmatrix}_{mD \times mD},$$

where the omitted parts are all zero matrices. The distance function between two quaternion embeddings $q_a$ and $q_b$ is defined as follows

$$d_Q(q_a, q_b) = \sqrt{mD - \text{tr}(\mathcal{M}^a \mathcal{M}^b)}$$

$$= \sqrt{mD - \sum_{i=1}^{D} \text{tr}(M_i^a M_i^b)},$$

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

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 $\{ q_1, q_2, \ldots, q_N \}$, such that:

$$d_Q(x_a, x_b) < d_Q(x_a, x_c), \forall (a, b, c) \in \mathcal{S}.$$
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, Ours2 obtains the minimum error in most cases while Ours3 is the second best in most cases. This shows that the matrix representation 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 rate \( \lambda = 0.5 \) 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, Ours2 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 cannot 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 rate \( \lambda = 0.05 \) and the number of epochs to 80 for all methods. Other settings are the same as the synthetic 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, Ours2 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 have a larger error.}

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

<table>
<thead>
<tr>
<th>Method</th>
<th>D=4</th>
<th>D=8</th>
<th>D=16</th>
</tr>
</thead>
<tbody>
<tr>
<td>EOE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GNMDS1</td>
<td>0.2760 ± 0.0027</td>
<td>0.2638 ± 0.0014</td>
<td>0.2573 ± 0.0015</td>
</tr>
<tr>
<td>GNMDS2</td>
<td>0.4233 ± 0.0066</td>
<td>0.3317 ± 0.0021</td>
<td>0.3112 ± 0.0019</td>
</tr>
<tr>
<td>CKL</td>
<td>0.2744 ± 0.0010</td>
<td>0.2713 ± 0.0028</td>
<td>0.2706 ± 0.0003</td>
</tr>
<tr>
<td>STE</td>
<td>0.3380 ± 0.0050</td>
<td>0.3199 ± 0.0030</td>
<td>0.3099 ± 0.0044</td>
</tr>
<tr>
<td>t STE</td>
<td>0.2644 ± 0.0016</td>
<td>0.2611 ± 0.0013</td>
<td>0.2367 ± 0.0006</td>
</tr>
<tr>
<td>HOE</td>
<td>0.2839 ± 0.0020</td>
<td>0.2574 ± 0.0051</td>
<td>0.2259 ± 0.0214</td>
</tr>
<tr>
<td>Ours1 w/o dis</td>
<td>0.2779 ± 0.0013</td>
<td>0.2684 ± 0.0069</td>
<td>0.2382 ± 0.0015</td>
</tr>
<tr>
<td>Ours2 w/o dis</td>
<td>0.2756 ± 0.0010</td>
<td>0.2588 ± 0.0071</td>
<td>0.1672 ± 0.0012</td>
</tr>
<tr>
<td>Ours2</td>
<td>0.2007 ± 0.0003</td>
<td>0.1322 ± 0.0020</td>
<td>0.0714 ± 0.0012</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Method</th>
<th>D=4</th>
<th>D=8</th>
<th>D=16</th>
</tr>
</thead>
<tbody>
<tr>
<td>EOE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GNMDS1</td>
<td>0.2323 ± 0.0012</td>
<td>0.2087 ± 0.0005</td>
<td>0.2028 ± 0.0003</td>
</tr>
<tr>
<td>GNMDS2</td>
<td>0.2244 ± 0.0004</td>
<td>0.2084 ± 0.0007</td>
<td>0.2010 ± 0.0002</td>
</tr>
<tr>
<td>CKL</td>
<td>0.2766 ± 0.0001</td>
<td>0.2532 ± 0.0002</td>
<td>0.2516 ± 0.0020</td>
</tr>
<tr>
<td>STE</td>
<td>0.2239 ± 0.0014</td>
<td>0.1953 ± 0.0005</td>
<td>0.1805 ± 0.0003</td>
</tr>
<tr>
<td>t STE</td>
<td>0.2896 ± 0.0018</td>
<td>0.2736 ± 0.0011</td>
<td>0.2592 ± 0.0005</td>
</tr>
<tr>
<td>HOE</td>
<td>0.2572 ± 0.0013</td>
<td>0.2153 ± 0.0007</td>
<td>0.1985 ± 0.0005</td>
</tr>
<tr>
<td>Ours1 w/o dis</td>
<td>0.2847 ± 0.0093</td>
<td>0.2076 ± 0.0053</td>
<td>0.2011 ± 0.0019</td>
</tr>
<tr>
<td>Ours1</td>
<td>0.3148 ± 0.0114</td>
<td>0.2032 ± 0.0044</td>
<td>0.1445 ± 0.0017</td>
</tr>
<tr>
<td>Ours2 w/o dis</td>
<td>0.2924 ± 0.0017</td>
<td>0.2042 ± 0.0012</td>
<td>0.1739 ± 0.0012</td>
</tr>
<tr>
<td>Ours2</td>
<td>0.2929 ± 0.0011</td>
<td>0.1771 ± 0.0013</td>
<td>0.1444 ± 0.0005</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Method</th>
<th>D=4</th>
<th>D=8</th>
<th>D=16</th>
</tr>
</thead>
<tbody>
<tr>
<td>EOE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GNMDS1</td>
<td>0.2953 ± 0.0045</td>
<td>0.2840 ± 0.0005</td>
<td>0.2784 ± 0.0020</td>
</tr>
<tr>
<td>GNMDS2</td>
<td>0.3301 ± 0.0352</td>
<td>0.3261 ± 0.0029</td>
<td>0.3227 ± 0.0051</td>
</tr>
<tr>
<td>CKL</td>
<td>0.2791 ± 0.0005</td>
<td>0.2742 ± 0.0005</td>
<td>0.2759 ± 0.0002</td>
</tr>
<tr>
<td>STE</td>
<td>0.2703 ± 0.0020</td>
<td>0.2551 ± 0.0021</td>
<td>0.2470 ± 0.0025</td>
</tr>
<tr>
<td>t STE</td>
<td>0.3255 ± 0.0065</td>
<td>0.3156 ± 0.0036</td>
<td>0.3078 ± 0.0044</td>
</tr>
<tr>
<td>HOE</td>
<td>0.3986 ± 0.0193</td>
<td>0.2791 ± 0.0047</td>
<td>0.2606 ± 0.0029</td>
</tr>
<tr>
<td>Ours1 w/o dis</td>
<td>0.2821 ± 0.0188</td>
<td>0.2642 ± 0.0099</td>
<td>0.2589 ± 0.0131</td>
</tr>
<tr>
<td>Ours1</td>
<td>0.2668 ± 0.0123</td>
<td>0.2454 ± 0.0089</td>
<td>0.2427 ± 0.0027</td>
</tr>
<tr>
<td>Ours2 w/o dis</td>
<td>0.2749 ± 0.0188</td>
<td>0.2458 ± 0.0161</td>
<td>0.2389 ± 0.0064</td>
</tr>
<tr>
<td>Ours2</td>
<td>0.1706 ± 0.0089</td>
<td>0.1513 ± 0.0073</td>
<td>0.1422 ± 0.0043</td>
</tr>
</tbody>
</table>

Table 4: Testing errors (mean ± standard deviation) on Bird dataset.
<table>
<thead>
<tr>
<th></th>
<th>matrix((3 \times 3))</th>
<th>ours(_1)((D = 8))</th>
<th>ours(_2)((D = 8))</th>
</tr>
</thead>
<tbody>
<tr>
<td>GNMDS(_1)</td>
<td>0.2284 ± 0.0039</td>
<td>0.2558 ± 0.0013</td>
<td>0.1436 ± 0.0009</td>
</tr>
<tr>
<td>GNMDS(_2)</td>
<td>0.2445 ± 0.0018</td>
<td>0.2035 ± 0.0021</td>
<td>0.1589 ± 0.0008</td>
</tr>
<tr>
<td>STE</td>
<td>0.2269 ± 0.0045</td>
<td>0.1644 ± 0.0040</td>
<td>0.1698 ± 0.0046</td>
</tr>
<tr>
<td>t-STE</td>
<td>0.2581 ± 0.0086</td>
<td>0.2032 ± 0.0044</td>
<td>0.1771 ± 0.0013</td>
</tr>
</tbody>
</table>

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

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 \(3 \times 3\) 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 \([0.2; 0.3; \ldots ; 0.8]\), 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.

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


