Project: Laplacian Eigenmap

Examples

Group 1

Introduction

In the fields of artificial intelligence, information retrieval, and data mining, it is usually encountered that low-dimensional data are embedded in high-dimensional space. Therefore, the appropriate representation of complex data is a topic of interest to researchers. Take biological perception as an example: the stimulus signals usually have both high-dimensional representations and low-dimensional intrinsic structures. If the perceived high-dimensional signals can be restored to low-dimensional ones, it will be of great benefit to the research of biological perception.

There are many papers and methods on dimensionality reduction, but most of them do not take advantage of the structure of the manifold in which the data reside. Belkin and Niyogi(2022) proposed the Laplacian Eigenmaps method as well as a new analytical framework for geometrically motivated data dimensionality reduction algorithms.

The results can reflect the intrinsic geometric structure of the manifold, because the optimal embedding is provided by the Laplacian Beltrami operator on the manifold. The Laplacian of the graph-adjacent matrix can be regarded as an approximation to the Laplacian Beltrami operator on the manifold, while the embedding maps (discrete) of the data points come from the approximation of eigenmaps/natural maps (continuous) defined on the whole manifold.

There are several properties of the Laplacian Eigenmaps method (hereafter referred to as LE):

1. The computation required after building the map is relatively simple. The process of building a map requires searching the nearest neighbourhood, which is not as easy.

2. It preserves locality, and therefore is relatively insensitive to outliers and noise.

3. A by-product of the LE method's dimensionality reduction is its clustering property: when the data is represented on a two-dimensional plane, the original clustering relationships are easily revealed. The principle is similar to the **spectral clustering methods** developed in machine learning and computer vision. In contrast, global methods do not show clustering tendency. However, not all datasets have meaningful clustering and should be used with care.

4. The choice of t becomes more important when N is large. This is because if a point is connected to most of the points on the graph, the parameter t of the heat kernel should be used to create a "difference pattern", so as to distinguish the really close and distant points, and to achieve the effect of "locality preserving".

Corpus Dataset

A linguistic sample

In natural language processing tasks, numeric vectors are often used to represent words. Since computers cannot understand the words directly, computer scientists proposed to learn sequence level semantics by considering the sequence of all words in the documents. The basic idea is that words appearing in similar contexts tend to have similar properties, and the properties of a word can be determined by the types and proportions of neighbouring words. The basic idea is that words appearing in similar contexts often have similar roles and properties, and the category of a word can be determined by the types and proportions of its neighbouring words, making it possible to distinguish between different words and find similar words.

Brown Corpus is a famous English corpus, with a variety of text sources and a total volume of more than one million words. Following Belkin and Niyogi (2002), we use Brown Corpus to construct a dataset of contextual vector representations of the top 300 words in terms of frequency of co-occurrence, and apply the LE method to the dataset to reduce the data dimensionality. The dataset is constructed as follows.

1. After excluding the stop words - the, of, and, to, in, a, an, I - the 300 words with the highest frequency of occurrence in the corpus are counted and denoted as T300.

2. for each word in T300, find the position in the corpus where it occurs.

3. from this position, check two words to the left and two words to the right to see if there are any words in T300.

4 For words A,B,C in T300, each time word B is observed as the first or second nearest neighbor in the left of word A, add 1 count to row A, column B of the dataset; each time word C is observed as the first and second nearest neighborin the right of word A, add 1 count to row A, column 300+C of the dataset. There are 600 columns at total. The first 300 columns represent the left context words of a certain word, and the last 300 columns represent the right context words of a certain word.

Experiment 1: Using knn, comparing different k

KNN, short for "k nearest neighbors", select the first k nearest neighbors to construct the adjacent graph. When k is set too small, the algorithm may only focus on a narrow neighborhood of the data points, making it hard to capture the geometric structure of the manifold. When k is set too large, the algorithm may try to preserve global distances, and would not be able to extract the structure of low dimensional manifold. Therefore, it is important to try, compare and find the best k setting for a certain dataset.

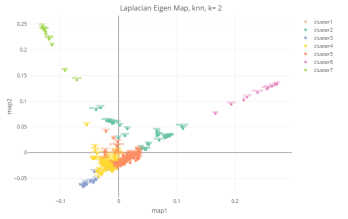

Figure 1: k=2

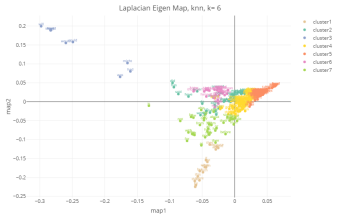

Figure 2: k=6

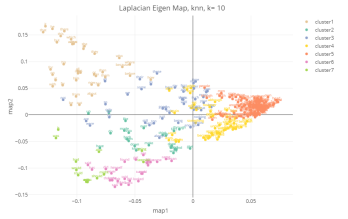

Figure 3: k=10

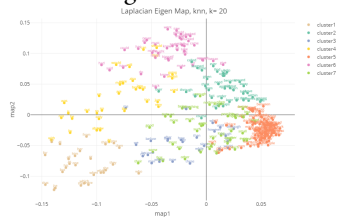

Figure 4: k=20

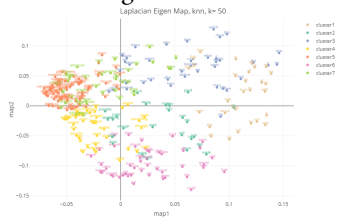

Figure 5: k=50

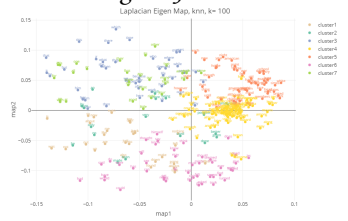

Figure 6: k=100

Since the objective function penalises the distance between two connected vertices, it can be imagined that there are "gravity" between connected vertices, which prevents them from getting far away. from each other. When k is small, each data point is only attracted by a few neighbours, so it can maintain the geometrical shape of "sequentially connected" on a one-dimensional curve. Smaller clusters of points that are farther away from the massive cluster can be fully connected and not penalised too much in the objective function. When k is large, each data point has a bunch of edges and will be pulled by the global data points. Therefore, the data points are distributed evenly in the two-dimensional plane. It is thought inappropriate to make the pulling effect caused by the nearest neighbor and the 10th nearest neighbor equal to each other. Therefore, we make experiment 3 to try the weighted edge.

Experiment 2: Comparing the result of scaling and not scaling.

Generally speaking, data are often standardised in pre-process procedure to eliminate the effect caused by the different scales of the attributes. Then, researchers could compare the effect of different variables fairly. However, in our experiments, all 600 attributes are of the same category: they are counts of co-occurrences of the words represented by rows and columns. Although the magnitudes are different, the values of each scale are not completely incomparable. Therefore, standardisation by columns is not a natural must, and the need for standardisation needs to be determined according to the effectiveness of data reduction. In Experiment 2, we choose the data after column-wise standardisation and keep the other parameter settings the same as in Experiment 1.

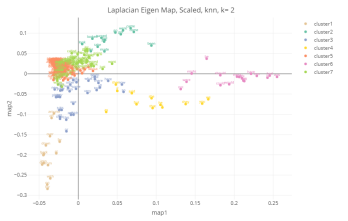

Figure 7: Scaled, k=2

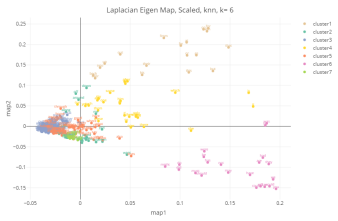

Figure 8: Scaled, k=6

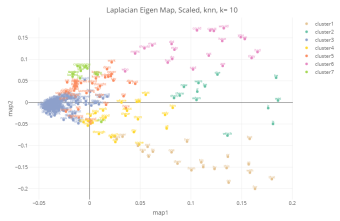

Figure 9: Scaled, k=10

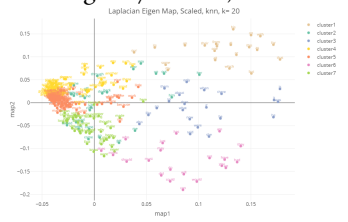

Figure 10: Scaled, k=20

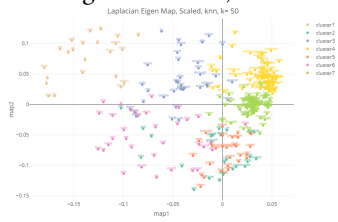

Figure 11: Scaled, k=50

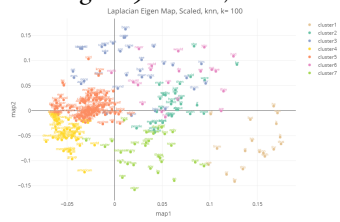

Figure 12: Scaled, k=100

After standardisation, smaller k reveal a more differentiated point layout. If the 2D presentation in experiment 1 appears in a broom shape, the layout in experiment 2 appears like a snail shape. The similarity is that most of the data points are still centred around the origin, i.e. the area of the broomstick or the snail's head. The difference is that in Experiment 1, the dispersed points are roughly distributed on a nearly parallel curve, with a sense of sequentially connection. In Experiment 2, the dispersed points were distributed near the contour of the circle, and the points in the middle of the circle were quite sparse.

Experiment 3: Using weighted matrix and trying different parameters t.

When constructing a graph, you can optionally use a heat kernel to assign different edge weights to pairs of points according to their distance in the original space. The closer the distance is in the original space, the larger the edge weight the adjacent graph, indicating a close connection between the data pair. The further the distance in the original space, the smaller the edge weight in the adjacent graph, which means the connection between the data pair is loose.

When the number of connected edges is large, heat kernel can distinguish the importance of edges and balance the emphasis on locality and global distances. Heat kernel is useful when the distances between points in the original space are not too large. When there are outlying clusters in the original space (the distance between clusters is much larger than the distance within clusters), the selection of heat kernel weights will be a dilemma: if the parameter t is set to a large value, all edges within clusters will be close to one; if the parameter t is set to a small value, the inter-cluster edge weights will be close to 0, and the whole map is approximately divided into two or more connected components. Applying the Laplace feature mapping method to a non-connected graph will lead to unexpected problems, see the following example.

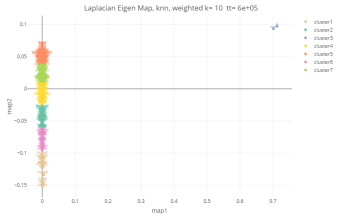

Figure 13: $k = 10, t = 6 \times 10^5$

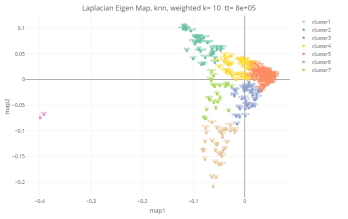

Figure 14: $k = 10, t = 8 \times 10^5$

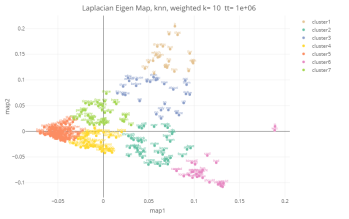

Figure 15: $k = 10, t = 1 \times 10^6$

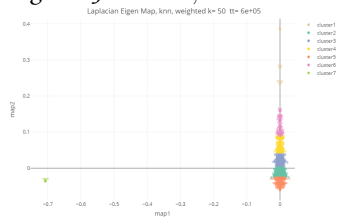

Figure 16: $k = 50, t = 6 \times 10^5$

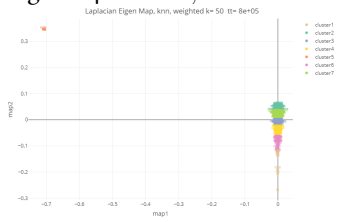

Figure 17: $k = 50, t = 8 \times 10^5$

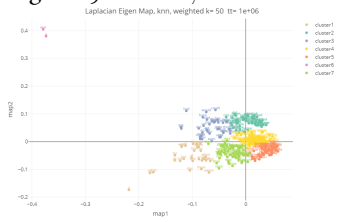

Figure 18: $k = 50, t = 1 \times 10^6$

Lemma 1. *For a fully connected graph, the dimensionality of the eigen space of the graph Laplacian corresponding to the eigen value* 0 *is exactly* 1.

Proof. For the constructed graph, denote the vertices as $\{x_1, ...x_n\}$, the graph map g can be represented by a vector f, where the ith element of f is the image of vertex x_i in eigen map $g(x)$, that is

$$f_i = g(x_i), i = 1, 2, ...n$$

As we already know,

$$\sum_{i,j} ||f_i - f_j||^2 W_{ij} = tr(f^T L f)$$

Now, prove by contradiction. Assume we have an eigen value corresponding to 0 with dimensionality no smaller than 2, then we can take a pair of orthogonal vectors $f^{(1)}$ and $f^{(2)}$ such that

$$f^{(1)T} L f^{(1)} = f^{(2)T} L f^{(2)} = 0$$

therefore, we have

$$\sum_{i,j} ||f_i^{(1)} - f_j^{(1)}||^2 W_{ij} = \sum_{i,j} ||f_i^{(1)} - f_j^{(1)}||^2 W_{ij} = 0$$

let $f^{(1)} = c1_n \in \mathbf{R}^{n \times 1}$ be the all one vector, which maps all vertices to a same image. Then, investigate $f^(2)$. If $f^{(2)} \perp f^{(1)}$, we can take at least one vertices pair (x_i, x_j), such that $f_i^{(2)} \neq f_j^{(2)}$. Since the graph is fully connected, there must be a path of edges $\{x_i, x_{k_1}, x_{k_2}, ..., x_{k_m}, x_j\}$ that connect x_i and x_j, and $W_{ik_1}, W_{k_1k_2}, ...W_{k_mj} \neq 0$. Since

$$\sum_{i,j} ||f_i^{(1)} - f_j^{(1)}||^2 W_{ij} = 0$$

There must be

$$||f_i^{(2)} - f_{k_1}^{(2)}||^2 = ||f_{k_1}^{(2)} - f_{k_2}^{(2)}||^2 = ... = ||f_{k_m}^{(2)} - f_j^{(2)}||^2 = 0$$

and consequently

$$f_i^{(2)} = f_{k_1}^{(2)} = ... = f_{k_m}^{(2)} = f_j^{(2)}$$

which makes contradiction. □

Lemma 2. *For a graph with k connected components, the dimensionality of the eigen space of the graph Laplacian corresponding to the eigen value* 0 *is at least* k*.*

Proof. Now, consider the graph with k connected components, by adjusting the rows and columns, we can get a block diagnonal Laplacian.

$$L = \begin{bmatrix} L_1 & 0 & ... & 0 \\ 0 & L_2 & ... & 0 \\ ... & ... & ... & ... \\ 0 & 0 & ... & L_k \end{bmatrix}$$

According to **Lemma 1**, the eigen space of a fully connected graph's Laplacian is of 1 dimension, and the only unit eigenvector is an scaled all one vector 1_n. Then, for $L_i \in \mathbf{R}^{m_i \times m_i}$, we have eigen vectors l_{m_i}'s such that

$$1_{m_i}^T L_i 1_{m_i} = 0$$

Then, they can be constructed into an orthogonal basis for the eigen space of the origianl Laplacian corresponding to the eigen value 0.

$$e^{(1)} = \frac{1}{\sqrt{m_1}} \begin{bmatrix} 1 \\ ... \\ 1 \\ 0 \\ ... \\ 0 \\ 0 \\ ... \\ 0 \end{bmatrix}, e^{(i)} = \frac{1}{\sqrt{m_i}} \begin{bmatrix} 0 \\ ... \\ 0 \\ 1 \\ ... \\ 1 \\ 0 \\ ... \\ 0 \end{bmatrix}, e^{(k)} = \frac{1}{\sqrt{m_k}} \begin{bmatrix} 0 \\ ... \\ 0 \\ 0 \\ ... \\ 0 \\ 1 \\ ... \\ 1 \end{bmatrix}$$

$$e^{(i)} \perp e^{(j)}, i \neq j, i, j = 1, 2, ...k$$

$$e^{(i)T} L e^{(i)} = 0$$

□

Experiment 4: Observing from different perspectives.

In Laplacian Eigenmaps algorithm, each eigen map contains information. As mentioned above, the first and second eigen maps may be affected by the connected components and contain little information. Sometimes, the information

contained in the third and fourth eigen maps may have better interpretability than the results of the first and second eigen maps. It is necessary to extend our attention to the combination of different dimensions and examine them to obtain a low-dimensional representation that is easy to understand and conforms to the true clustering labels.

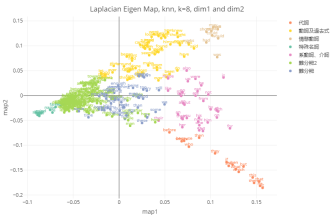

Figure 19: Dimension 1 and 2.

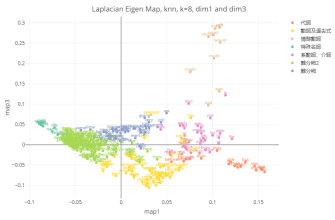

Figure 20: Dimension 1 and 3.

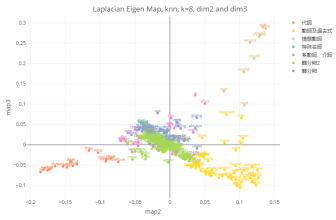

Figure 21: Dimension 2 and 3.

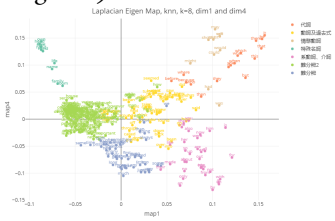

Figure 22: Dimension 1 and 4.

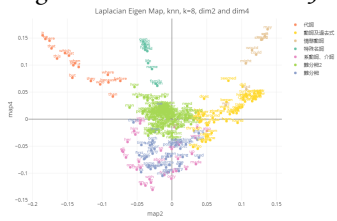

Figure 23: Dimension 2 and 4.

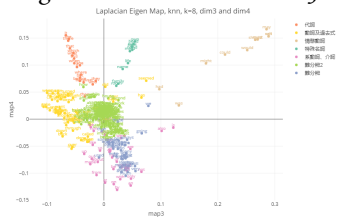

Figure 24: Dimension 3 and 4.

Among the 6 scatter plots of dimension pairs, pair $(2, 3)$, $(2, 4)$ turn out to be most elegant. In Figure21, orange, brown and yellow clusters are separated clearly, while in Figure22, cyan, green, violet and pink clusters are separated clearly. Here comes the zoomed in pictures of these clusters, and in 5 of 7 clusters, a cluster name could be generalized according to the dominant words' category.

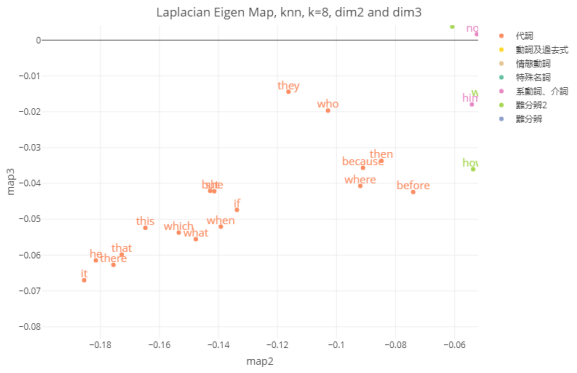

Figure 25: Mainly Pronouns.

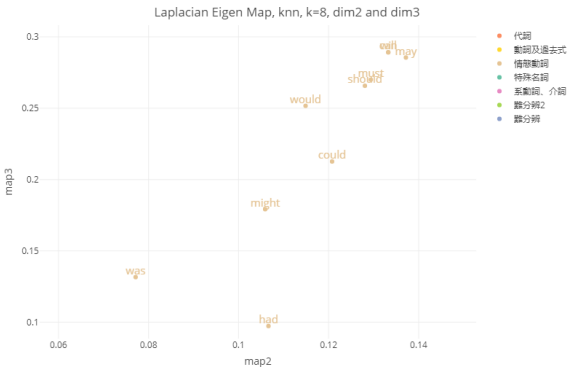

Figure 26: Mainly modal verbs.

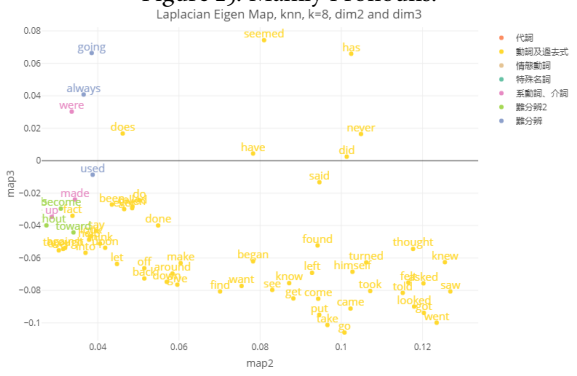

Figure 27: Verbs and past tense.

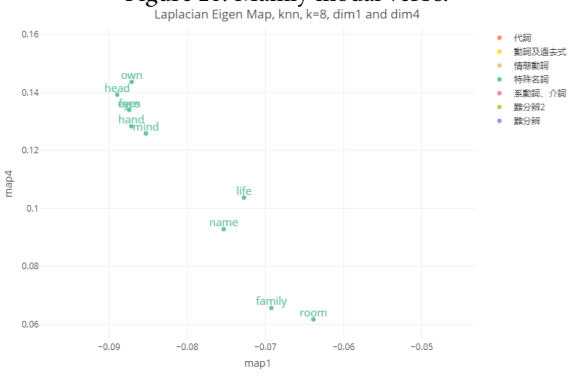

Figure 28: Special nouns.

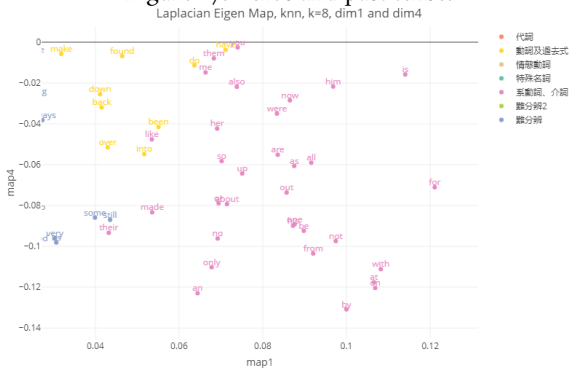

Figure 29: Prepositions and adverbs.

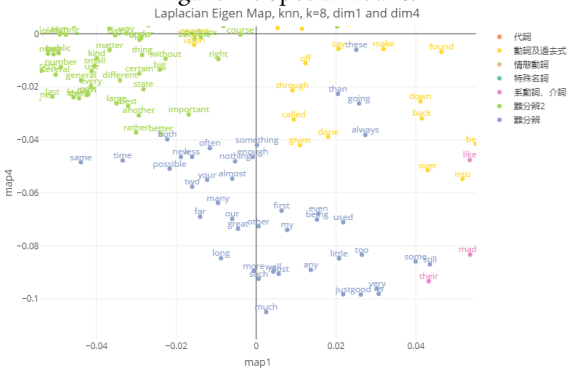

Figure 30: Hard to distinguish.

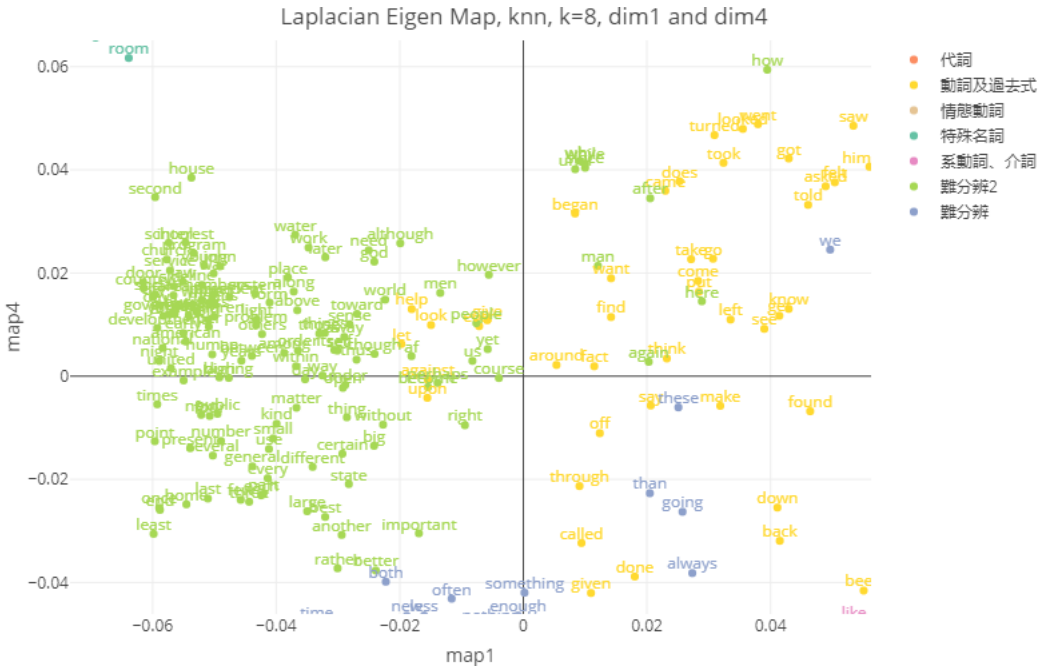

Figure 31: Hard to distinguish.

Experiment 5: the dilemma of enn

When epsilon is large, locality preservation may not be guaranteed, and when epsilon is small, it may result in connected components (not fully connected graphs). It can be considered to apply LE to each connected component separately, but the low-dimensional representations of the points between two different parts can not be compared with each other. Knn limits the number of connected points to guarantee locality preservation. When k is large enough, the graph is fully connected.