Chapter 5

HyperMap: Parametric Linear Visualization for High-Dimensional Data Clustering

In this chapter, we develop a clustering tool called parametric visualization. The current state of the art technique of visualization has fixation target space to visualize. In this chapter, we propose a dynamic target space changed by tuning parameters.

5.1 Introduction

Visualizing data clustering in a high dimensionality space is difficult due to the nature of data scatter and the way of seeing data in such a high dimensional space. The data scatter is explained below. When dimensionality becomes high, accordingly, the data embed in such a high dimensionality space becomes much more sparse[11, 9]. The distance between objects increase while the dimensionality increases. An example is shown in Figure 5.1. Assume the radius of the circle is 1. When the space is a 2-dimensional space, the distance between A and B is $\sqrt{2} - 1 = 0.414$, and the distance between B and O is 1. The ratio between the two line segments is considerably small. When the dimensionality reaches 100, the distance between A and B becomes $\sqrt{100} - 1 = 9$. In other words, the line segment between A and B becomes 9 times of the line segment between B and O, because the circle becomes smaller in a high dimensionality space and the volume exclusive of the circle becomes larger exponentially.

In order to visualize data in a 2 or 3- dimensional space, dimensionality reduction needs to be performed. Principle Component Analysis (PCA) is the classical method. Because most datasets have high dimensionality (or inherent dimensionality), PCA[24] is impossible to select "good" dimensions in which the variance of coordinates are much bigger than the others. Many visualization algorithms are proposed including eigenvector-based techniques, such as, MDS [37], LLE [33], and Isomap [36]. The data scatter in the target space created by those algorithm is fixation. User analyzes the data scatter from the *only one* representation in target space. There are no rooms for users to change their viewpoints. Because information is significantly lost when mapping to a 2 or 3- dimensional space, it is difficult that all the clusters are picked up in only one scatter in target space.

The aforementioned approaches do not offer an interactive way for users to interpret data clustering. The challenge issues are: why and how such clusters are found? If a user

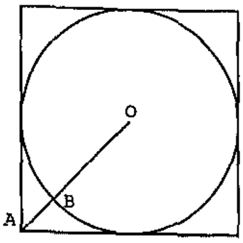

Figure 5.1: The higher dimensionality, the "smaller" circle is

changes his/her view point(s), is it possible for him/her to find different clusters or will the existing clusters disappear depending on the user's viewpoint changes? If a user's viewpoint will affect the ways of exploring clusters, how do we allow users to interactively change their viewpoints and see the clusters in linear time. These issues have not well studied for high-dimensional data clustering. In this paper, we focus ourselves on visualizing high-dimensional data in a 3/2-dimensional or even 1-dimensional space. We introduce a parametric approach allowing users to interactively change their viewpoints and see the changes on-line.[1]

Our parametric approach is developed on top of a new novel mapping algorithm called *HyperMap* we propose in this paper. HyperMap is a generalization of FastMap [18]. In FastMap, 2 objects (called pivots) are extracted at one time to determine a axis to be reduced. The dimensionality reduction is done one-dimension by one-dimension. There are two drawbacks with FastMap. First, every pivot is crucial for discrimination of data in target space. The quality of data clustering declines from the time that a bad pivot is selected. Second, the target space is fixation, users can not observe the scatter of data in visual space

[1] Our online demo is available at `http://www.dblab.is.tsukuba.ac.jp/~an/HyperMap/HyperMapDemo.htm`.

interactively. Different from FastMap, in HyperMap, an axis consists of k (≥ 2) pivots, which determines a (k-1)-hyperaxis. The parametric approach developed on top of HyperMap allows users to observe the data scatter in 2 or 3- dimensional space interactively. By using HyperMap, our approach allows users to use the data scatters in linear time, and reduce the impacts of selecting a bad pivot. The main contributions of this paper are given below.

- We define a *hyperaxis* and *coordinate value* of hyperaxis. We break through the limitation of Euclidean space that an axis is a line. In our approach, an axis can be a line, a plane, or a hyperplane.
- We derive a formula for translating coordinate values from original space to destination space.
- We develop a novel interactive technique that allows users to change their viewpoints by tuning tuning the weight associated with each hyperaxis. By changing the weights, users can see the clusters from different viewpoints, interactively and efficiently.

The rest of the paper is organized as follows. Section 5.2 outlines our parametric approach. We introduce HyperMap in Section 5.3. The analyses of HyperMap are given in Section 5.4. We conclude the paper in Section 5.5.

5.2 Visualizing Large High-Dimensional Data in a 3-Dimensional Space

In data visualization, the most important issue is that the neighborhood of objects that are close in the original high dimensionality space should be close in the reduced data space to be visualized. At the same time, objects that are far away in the original high dimensionality space should not be close in the reduced data space to be visualized.

We use an example to explain the interactive visualization shown in Figure 5.2. There are three clusters. We assume that a pivot is the center of a cluster (Figure 5.2 (A)). For each object, we can calculate three distances between the object and any of the pivots, denoted as d_1, d_2 and d_3. We call these distances as *ingredient* of the object in a hyperaxis. The coordinates value of the hyperaxis are obtained using Equation 5.1.

$$z = \alpha_1 d_1 + \alpha_2 d_2 + \alpha_3 d_3 \tag{5.1}$$

A user can interactively tune the weights α_1,α_2 and α_3 to visualize the 3 clusters. For instance, setting $\alpha_1 = 0.0$, $\alpha_2 = \alpha_3 = 0.5$, the cluster C_1 is recognized as shown in Figure 5.2(B), because the coordinate values of the data points in C_1 are smaller than those of the data points in other clusters C_2 and C_3. In a similar fashion, by setting $\alpha_2 = 0.0$ and $\alpha_1 = \alpha_3 = 0.5$, the cluster C_2 can be separated as shown in Figure 5.2(C).

It is important to note that Equation (5.1) shows how to select one coordinate value to visualize data. We need to determine other coordinate values accordingly. To the best of

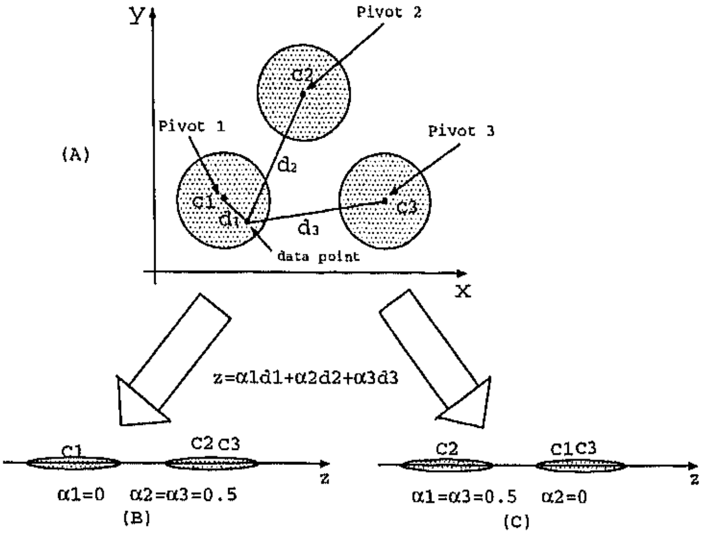

Figure 5.2: Motivation of HyperMap. There are 3 clusters in 2-dimensional space (A), we can store three ingredients of one hyperaxis. By tuning weights in a hyperaxis, the 3 clusters are recognized one by one.

our knowledge, there is no reported study on selecting coordinate values, because of the difficulties of the recursive calculation of projected distances. In this paper, we propose a novel algorithm that recursively chooses pivot objects employing k-center technique [21].

5.3 HyperMap

5.3.1 FastMap

FastMap algorithm can be described in brief as follows.

1. Pick up 2 pivot objects p_1, p_2 to construct the first axis p_1p_2.

2. All the data is projected into the axis p_1p_2, the first coordinate in target space is

gotten.

3. Project all data into the hyper plane $\mathcal{H}$ perpendicular to the axis p_1p_2.

4. In the hyper plane $\mathcal{H}$, steps 1,2,3 are repeated until all the axes of target space are gotten.

FastMap argorithm is generalized by extending the number of pivot objects of the axis of target space. Two problems arise. One is how to select pivot objects effectively. the other is what is the coordinate (called *hypercoordinate*) in target space.

HyperMap is to map objects in a n-dimensional space into a (k-1)-dimensional hyperplan, that is determined by the corresponding (k-1)-hyperaxis using k n-dimensional objects (called pivots), $p_1, p_2, \ldots, p_k$, for $k \leq n$. In the following, we use $\overline{p_1p_2...p_k}$ to denote the complementary space that is orthogonal to the the hyperaxis. In addition, given an object o in the n-dimensional space, we use $o|_{p_1p_2...p_i}$ (or simply o') to denote the projection of o onto a hyperaxis $p_1p_2...p_i$. Notations and symbols are summarized in Table 5.1.

Definition 1 (hypercoordinate) *Given an object o, weight $W = (\alpha_1, \alpha_2, \ldots, \alpha_k)$ and $V = (v_1, v_2 \ldots, v_k)$ where each v_i is a ingredient of a hyperaxis. The hypercoordinate of o on a hyperaxis $p_1p_2...p_k$, denoted $o.x$, is defined below.*

$$o.x = W \cdot V^T = \sum_{i=1}^{k} \alpha_i \cdot v_i = \sum_{i=1}^{k} \alpha_i \cdot dist(o|_{p_1...p_k}, p_i) \tag{5.2}$$

Here, $o|_{p_1...p_k}$ is the projection of object o on the (k-1)-hyperaxis, v_i is an ingredient indicating the distance between $o|_{p_1...p_k}$ and a pivot object p_i $(1 \leq i \leq k)$. α_i is a weight such

Symbol	**Meaning**
$\mathcal{O}$	Dataset of objects $\{o_i\}$
N	number of objects in $\mathcal{O}$
W	a sequence of weights $(\alpha_1, \alpha_2, \ldots)$
k	number of pivots of a hyperaxis
$p_1p_2...p_k$	hyperaxis determined by $p_1, p_2, ..., p_k$
$\overline{p_1p_2...p_k}$	complementary space orthogonal to $p_1p_2...p_k$
$o\|_{p_1...p_k}$	projected point of p on $p_1p_2...p_k$
$dist(p, q)$	distance between p and q
$\overline{dist}(p, q)$	projected distance on the complementary space
$D(o, p_1...p_k)$	the relative coordinate for o on hyperaxis $p_1...p_k(k = 1, \ldots, n)$

Table 5.1: Notations

that

$$|\alpha_1| + |\alpha_2| + \ldots + |\alpha_k| = 1$$

The condition $\sum \alpha_i = 1$ specifies that the distance between two hypercoordinates, $x.o_i$ and $x.o_j$, cannot be larger than the distance between $o_i|_{p_1p_2...p_k}$ and $o_j|_{p_1p_2...p_k}$ in hyperaxis $p_1p_2...p_k$, because a hypercoordinates are "extracted" from the corresponding hyperplane determined by the hyperaxis. Let o_i' and o_j' be $o_i|_{p_1p_2...p_k}$ and $o_j|_{p_1p_2...p_k}$ in hyperaxis

$p_1p_2 ... p_k$. We show our proof below.

$$\begin{aligned}
|o_i.x - o_j.x| &= |\alpha_1(dist(o'_i, p_1) - dist(o'_j, p_1)) + \\
&\quad \alpha_2(dist(o'_i, p_2) - dist(o'_j, p_2)) + \ldots \\
&\quad + \alpha_k(dist(o'_i, p_k) - dist(o'_j, p_k))| \\
&\leq \max_{1 \leq m \leq k} |dist(o'_i, p_m) - dist(o'_j, p_m)| \\
&\leq dist(o'_i, o'_j)
\end{aligned}$$

It is also illustrated in Figure 5.3.

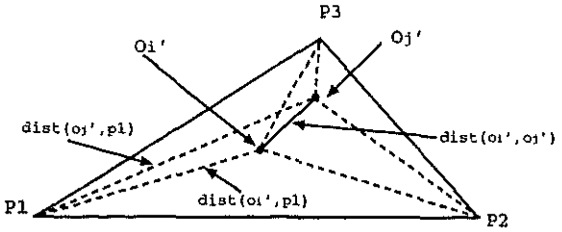

Figure 5.3: $|dist(o'_i, p_1) - dist(o'_j, p_1)| \leq dist(o'_i, o'_j)$.

5.3.2 HyperMap Overview

HyperMap maps objects in a n-dimensional space into a hyperaxis $p_1p_2...p_k$ in four steps.

1. **Step-1 (Pivot selection)**: Determine a (k-1) hyperaxis by selecting k pivots $(p_1, p_2, \ldots, p_k)$.
2. **Step-2 (Computing ingredient)**: For each object o, compute the distance $dist(o|_{p_1p_2...p_k}, p_i)$, for all pivots p_i, in the hyperaxis. Recall that $o|_{p_1p_2...p_k}$ is the projection of object o

on the hyperaxis.

3. **Step-3 (Computing projected distance in the complementary space):** Calculate distances between all pairs of o_i and o_j on the complementary space $\overline{p_1 p_2 ... p_k}$ that is orthogonal to the hyperaxis $p_1 p_2 ... p_k$.

4. **Step-4 (Looping):** In the hyperplane, steps 1-3 are repeated until all the axes of target space are obtained.

5. **Step-5 (Hypercoordinate calculation):**

   Compute the hypercoordinate on the hyperaxis of o for all data points using Equation (5.2).

In the case when $k = 2$, there exists two pivots (p_1 and p_2). If one of the two distances $dist(o, p_1)$ and $dist(o, p_2)$ is taken as a hypercoordinate, the HyperMap will perform in a similar manner like FastMap [18], which is a special case of HyperMap.

5.3.3 Pivot Selection

In order to select k pivots in linear time, for large high dimensional datasets, we employ the k-center algorithm given in [21], which selects k repulsive representatives in linear time, as shown in Algorithm 6. In brief, it takes two steps as follows.

1. Select an object $o_i \in \mathcal{O}$, randomly. The first pivot p_1 is selected as the object that has the max distance from o_i, and is added into pivot set S, which is empty initially.

2. Repeatedly find a pivot p_i that satisfies the following inequation,

$$\max_{o' \in \mathcal{O}}(\min_{o \in S}(dist(o, o'))) \tag{5.3}$$

until all k pivots are selected.

```
Input:
  Dataset $\mathcal{O}$,
  Distance function $dist(\cdot,\ \cdot)$,
  Number of pivots $k$,
Output:
  A set of $k$ objects $S$.
Select-pivot($\mathcal{O}$,$dist()$, $k$)

begin
    $S \leftarrow \emptyset$
    Choose an object $o_i$ from $\mathcal{O}$ randomly.
    Choose $p_1$ such that for any $o_j \in \mathcal{O}$,
        $\max_{o_j \in \mathcal{O}}(dist(o_i,\ o_j))$
    Add $p_1$ to $S$
    while $|S| \leq k$ do
        Choose next pivot object $o'$ from Eq. (5.3),
        Add $o'$ to $S$.
    endwhile
    return $S$
end
```

Algorithm 6: Pivot Selection

Like FastMap, the computation complexity of this algorithm is linear.

5.3.4 Computing Relative Coordinate

For indicating the location of projected point o' on hyperaxis $p_1p_2 \ldots p_k$, $k-1$ real numbers are needed. Consequently, in this paper, *relative coordinate* $D(o, p_1p_2p_i)(2 \leq i \leq k)$ is

$$D(o, p_1p_2\ldots p_i) = \frac{(dist(o,p_1))^2 - (dist(o,p_i))^2 + (dist(p_1,p_i))^2 - 2\sum_{j=2}^{i-1} D(p_i, p_1p_2\ldots p_j)\cdot D(o, p_1p_2\ldots p_j)}{2\sqrt{(dist(p_1,p_i))^2 - \sum_{j=2}^{i-1} D(p_i,p_1p_2\ldots p_j)^2}} \quad (2 \le i \le k) \qquad (5.4)$$

$$\overline{dist}(o_i,o_j) = \sqrt{(dist(o_i,o_j))^2 - \sum_{t=2}^{k}(D(o_i,p_1p_2 ... p_t) - D(o_j,p_1p_2 ... p_t))^2} \qquad (5.5)$$

proposed. In the following, we will expain how to calculate ingredient and distance in complemently space by using relative coordinate.

Definition 2 (relative coordinate $D(\cdot,\cdot)$) *Given a (k-1)-hyperaxis, $p_1p_2...p_k$, for $k > 1$, an object o's relative coordinate is the distance between $o|_{p_1p_2...p_i}$ and an (i-1)-hyperplane $p_1p_2...p_i$, for $1 \le i \le k$. If the object o is on the same side of the last pivot object p_{i+1}, with respect to the hyperplane $p_1p_2...p_i$, its relative coordinate is positive, otherwise negative. Let $dist(o_i, o_j)$ be a Euclidean distance between two objects o_i and o_j. The formal definition, $D(o|p_1p_2...p_i, p_1p_2...p_i)$, is given in Equation (5.4), for $2 \le i < k$. When $i = 2$, a 1-hyperaxis is a line, p_1p_2, an object o's relative coordinate is computed by Equation (5.6), which is the same as FastMap. When $i = 1$, $D(o|p_1, p_1) = 0$.*

Figure 5.4 (A) illustrates $D(\cdot,\cdot)$ in a (k-1)-hyperaxis, $p_1p_2...p_k$, whereas Figure 5.4 (B) shows a 2-hyperaxis, $p_1p_2p_3$. In Figure 5.4 (B), an object o's relative coordinate $D(o, p_1p_2)$ and $D(o, p_1p_2p_3)$ are illustrated. As a special case, $D(o, p_1) = 0$ for an arbitrary data point o, because the distance from o's projection point to p_1 is zero.

The basic idea of mapping is to use "cosine law" (Equation (5.6)), as illustrated in Figure 5.5(A), using two pivots, p_1 and p_2. An object o is projected onto 1-hyperaxis determined by p_1p_2 at $o|_{p_1p_2}$ (simply o'). o's relative coordinate in the axis is computed by

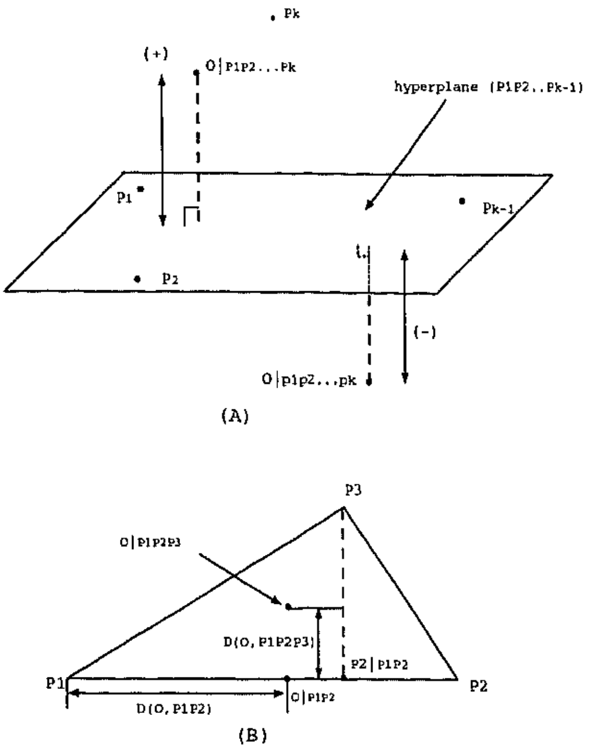

Figure 5.4: Definition of Relation Coordinate of Hyperaxis

$D(o', p_1p_2)$ below.

$$D(o', p_1p_2) = dist(p_1, o') = \frac{(dist(o', p_1))^2 - (dist(o', p_2))^2 + (dist(p_1, p_2))^2}{2 \cdot dist(p_1, p_2)} \tag{5.6}$$

The relative coordinate $D(p, p_1p_2...p_k)$ can be computed using an inductive method, as given in Appendix A.

Lemma 1 *Given data point o, it has a projected point o' on a hyperaxis $p_1p_2...p_3$. The vertical distance h from o to the hyperaxis can be calculated below.*

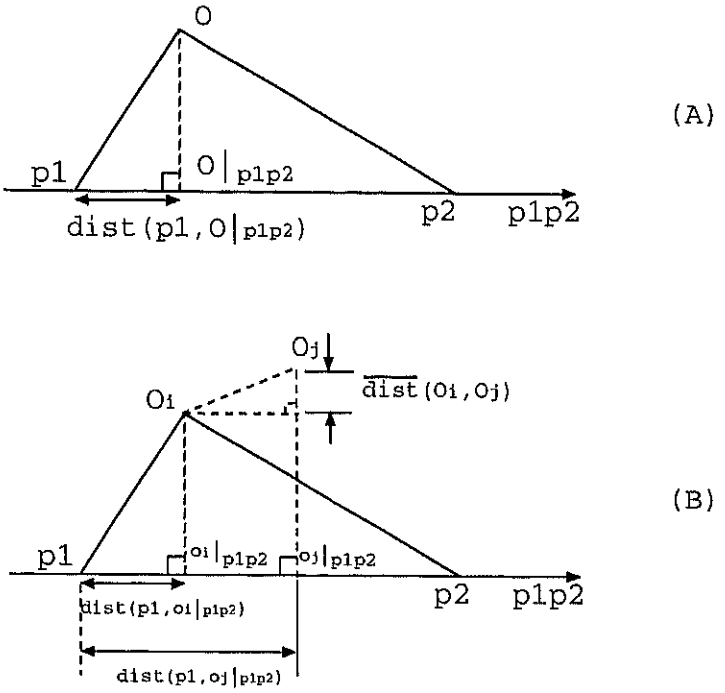

Figure 5.5: 1-hyperaxis(line). (A)Relative coordinate can be simply calculated by using ''Cosine Law'. (B) shows projected distance $\overline{dist}(o_i, o_j)$ between data point o_i and o_j in the complementary space of hyperaxis p_1p_2.

$$h = dist(o, o') = \sqrt{((dist(o, p_1))^2 - \sum_{i=2}^{k} D^2(o, p_1p_2...p_k)} \qquad (5.7)$$

Proof:

As shown in Figure 5.6, from the definition of relative coordinate,

$$(dist(p_1, o))^2 = (dist(p_1, o|_{p_1p_2}))^2 + (dist(o|p_1p_2, o))^2$$

$$= D^2(o, p_1p_2) + (dist(o|p_1p_2, o))^2$$

$\because \quad o|_{p_1p_2}\, o|p_1p_2p_3$ is on the hyperplane $p_1p_2p_3$

$\therefore \quad o\, o|_{p_1p_2} \perp o|_{p_1p_2}\, o|p_1p_2p_3$

$$(dist(o|_{p_1p_2}, o))^2 = (dist(o|_{p_1p_2}, o|_{p_1p_2p_3}))^2 + (dist(o|p_1p_2p_3, o))^2$$

$$= D^2(o, p_1p_2p_3) + (dist(o|p_1p_2p_3, o))^2$$

In generally, we have,

$$(dist(o|_{p_1p_2...p_i}, o))^2 = D^2(o, p_1p_2...p_{i+1}) + (dist(o|p_1p_2...p_{i+1}, o))^2$$

As a result,

$$(dist(p_1, o))^2 = \sum_{i=2}^{k} D^2(o, p_1p_2...p_i) + (dist(o|p_1p_2...p_k, o))^2$$

$$= \sum_{i=2}^{k} D^2(o, p_1p_2...p_i) + (dist(o', o))^2$$

5.3.5 Computing Projected Distance in the Complementary Space

Given two objects, o_i and o_j, in a n-dimensional space, the distance between o_i and o_j in the complementary space orthogonal to (k-1)-hyperaxis can be computed using Equation (5.5). Figure 5.5(B) shows a 1-hyperaxis with $\overline{dist}(o_i, o_j)$ between two objects, o_i and o_j, in the complementary space orthogonal to 1-hyperaxis p_1p_2.

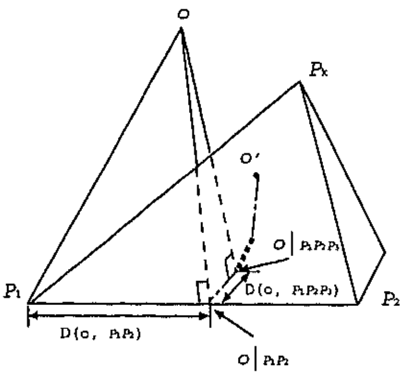

Figure 5.6: The distance from point data to hyperaxis by using relative coordinate

5.3.6 Hypercoordinate Calculation

According the Lemma 1. an ingredient $o.v_i$ of data point o can be obtained from relative coordinate below.

$$\begin{aligned} o.v_i^2 &= (dist(p_i, o))^2 - h^2 \\ &= (dist(p_i, o))^2 - (dist(p_1, o))^2 + \sum_{j=2}^{k}(D^2(o, p_1p_2 \ldots p_k)) \end{aligned} \tag{5.8}$$

5.3.7 Algorithm of HyperMap

The HyperMap algorithm is outlined in Algorithm 7. By replacing $dist(\cdot, \cdot)$ with $\overline{dist}(\cdot, \cdot)$, the relative coordinates $D(\cdot, \cdot)$ in the complementary space can be computed with respect to data point and pivot objects.

```
Global variables:
$N \times n \times npivots$ array D$[\cdot,\cdot,\cdot]$
{using to saving relative coordinate to
each pivots all layers for each data}
Number of pivots of each level $npivot[1]$,
        $npivot[2], \ldots, npivot[n]$
Input:
 Number of pivots $n$
 Distance function $dist(\cdot,\cdot)$
 Dataset $\mathcal{O}$,
Output:
 distance from data $o$ to each pivot, w.r.t.
$level$, $o.coor[level\#][pivot\#]$
HyperMap($n$,$dist()$,$\mathcal{O}$)

begin
  if ($n \leq 0$)
    return
  Select-pivot($\mathcal{O}$, $dist()$, $npivot[n]$){Algorithm 6}
  foreach data $o$ in $\mathcal{O}$ do
   foreach pivot $p_j$ ($j = 2 : npivot[n]$)
   {Relative coordinates are calculated by Eq.(5.4)}
     D$[o.\#, n, j] \leftarrow D(o, p_1 p_2 \ldots p_j)$
     {$o.\#$ is No. of data $o$}
   endforeach
   {compute the distance from $o$ to hyperaxis(Lemma 1)}
   $h \leftarrow (dist(o,p_1))^2 - \sum_{2 \leq i \leq npivot[n]} (D(o,p_1 p_2 \ldots p_i))^2$
   foreach pivot $p_j$ ($j = 1 : npivot[n]$)
    $o.coor[n][j] \leftarrow \sqrt{(dist(p_j,o))^2 - h}$
   endforeach
  endforeach
  {consider the projections of the objects on a
   hyperplane perpendicular to the hyperplane,
   the distance function $dist'()$
   is given by Eq. (5.5)}
  HyperMap($n-1, dist'(), \mathcal{O}$)
end
```

Algorithm 7: HyperMap

5.4 Empirical Results

Experiments were performed on a PC system of a single PentiumIII 700 MHz CPU and 128MB main memory, running Debian GNU/Linux 2.2. With this system, we aimed at evaluating of the feasibility of HyperMap.

Finding clusters with visualization is an effective technique[18, 6]. We mapped high dimensional datasets to $1, 2, 3$ dimensional spaces by HyperMap, and evaluated the quality of clustering with synthetic and real datasets. The clusters were found by tuning the weight W. As mentioned in the previous sections, it is very hard to find all clusters at once. Rather, we tried to separate each cluster by tuning the weight W with several times.

5.4.1 Synthetic Data Generation

In order to generate datasets, we used a method similar to the one discussed in [40]. In this method, anchor points of clusters are firstly determined. Then, how many points are associated with each anchor point are determined, and finally cluster points are generated.

More detailedly, anchor points of clusters are obtained by generating k uniformly distributed points in d dimensional space of $[0.0,\ 1.0]$. All clusters have the same number of points. The positions of data of each cluster follow the normal distribution, with the anchor point as its mean, and variance μ. In our experiments, we set the number of clusters to $k = 5$, and the variance of all clusters to $\mu = 0.2$.

5.4.2 Getting Accurate Results by Tuning Weight W

As shown in Figure 5.7, cluster 5 among 5 clusters is separated clearly, where the number of pivots for all hyperaxis are set to 3. The weight W is shown in Table 5.2.

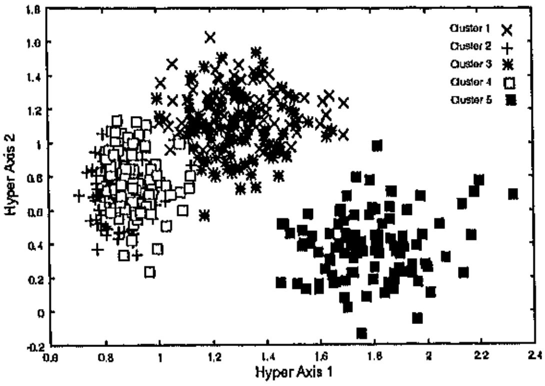

Figure 5.7: HyperMap with Synthetic Dataset. The Cluster 5 is found with the W of Table 5.2.

Table 5.2: Weight Used in the Experiment of Figure 5.7

Hyperaxis 1			Hyperaxis 2		
w_1	w_2	w_3	w_1	w_2	w_3
0.41	-0.09	0.5	0.83	0.04	-0.13

Similarly, by setting the weight W as in Table 5.3, cluster 4 is separated from other 5 clusters as shown in Figure 5.8.

Table 5.3: Weight Used in the Experiment of Figure 5.8

Hyperaxis 1			Hyperaxis 2		
w_1	w_2	w_3	w_1	w_2	w_3
0.93	0.0	-0.07	0.72	0.01	-0.27

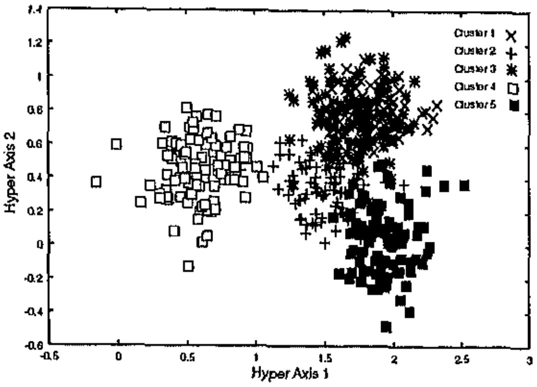

Figure 5.8: HyperMap with Synthetic Dataset. The Cluster 4 is found with W of Table 5.3.

5.4.3 Real dataset

We also applied HyperMap algorithm to a real dataset WINE[2]. There are 178 rows falling into 3 clusters in the wine dataset. Each row has 13 attributes, and indicates a specific sample of wine. Setting the weight W as described in Table 5.4, cluster 3 is clearly separated from others as shown in Figure 5.9.

Table 5.4: Weight Used in the Experiment of Figure 5.9

Hyperaxis 1			Hyperaxis 2		
w_1	w_2	w_3	w_1	w_2	w_3
-0.42	-0.04	0.54	0.49	-0.34	0.17

We tested 1-dimensional visualization with the WINE dataset. One hyperaxis, 5 pivots are selected. Cluster 1 and 3 is separated as shown in Figure 5.10 by tuning parameter W. The value of W is adjusted in this case as shown in Table 5.5. In Figure 5.10(A), 6 (10.17 %) data points of Cluster 1 are migrated into other 2 clusters. In Figure 5.10(B), only 3

[2] http://kdd.ics.uci.edu/

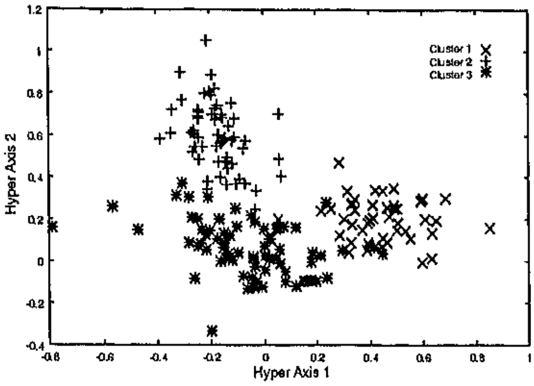

Figure 5.9: HyperMap with real dataset. The cluster 3 is separated at one W value

(6.25 %) data points of Cluster 3 fall into other 2 clusters.

Table 5.5: weight of Figure 5.10

	Hyperaxis 1				
	w_1	w_2	w_3	w_4	w_5
A	0.21	0.01	0.02	0.78	0.02
B	0.85	0.01	-0.06	0.01	0.07

5.5 Conclusions

In this paper we proposed a novel approach called HyperMap for mapping high dimensional data. In this technique, a new concept called hyperaxis is introduced. A hyperaxis is a hyperplane passing through k data points chosen by employing *k-center* algorithm. Especially, when k is 2, HyperMap is coincident with FastMap.

By mapping data points to the target space spanned by the hyperaxes, hyper coordinates are obtained. Experiments on real and synthesis datasets show the effectiveness of using

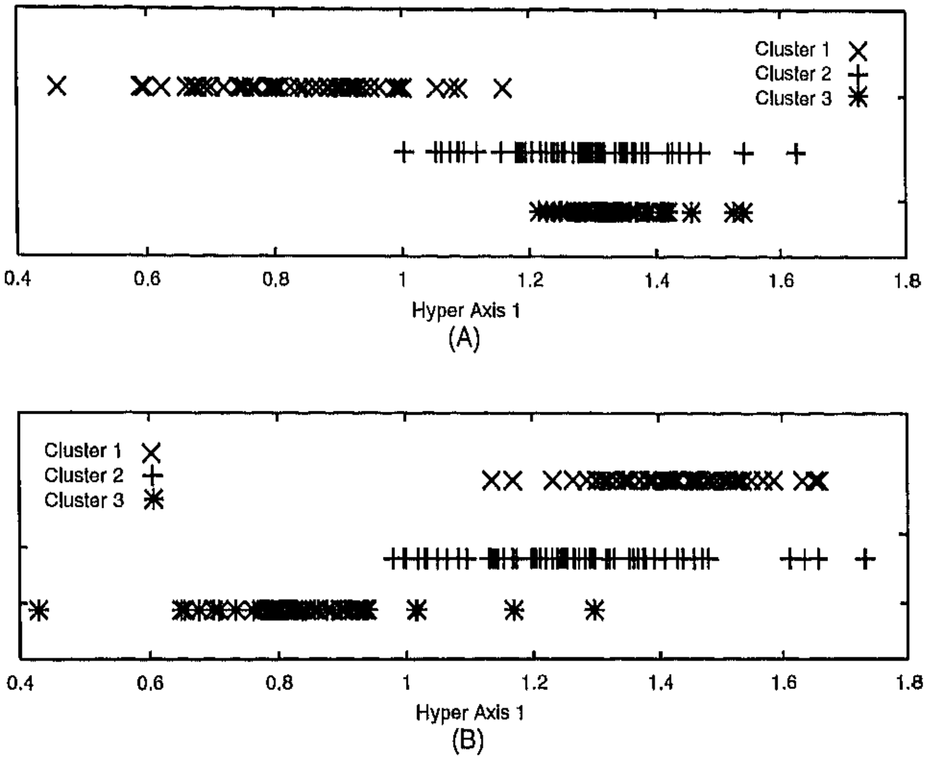

Figure 5.10: HyperMap with Real Dataset. The Cluster 1 and 3 are separated with the *W* value A and B of Table 5.5 respectively

such coordinates for classification and visualization. HyperMap is flexible because the weights can be tuned by users.

As future work, we are clarifying the properties of weight, and are investigating to discover a systematic method for determining the weights. Also, applying HyperMap to high dimensional indexing is considered to be an interesting work. We are sure that it is a good dimensionality reduction method for overcoming the curse of dimensionality.