番号：C２－１０

HyperMap:高次元空間における写像アルゴリズムとその次元縮小、クラスタリングへの応用

安際元[1]　古瀬一隆[2]　陳漢雄[2]　石川雅弘[3]　于旭[4]　大保信夫[2]

1 筑波大学工学研究科　　2 筑波大学　電子・情報工学系
3 農業生物資源研究所　　4 香港中文大学

Abstract

本稿では FastMap 射影法を一般化した写像手法 HyperMap を提案する。HyperMap は特異値分解(SVD)の複雑なアルゴリズムを避けることができる。また、射影による次元情報抽出を加速するため、FastMap のピボットを超平面に拡張し、「超軸」という概念を導入する。「超軸」への射影アルゴリズムは残存情報をより速く0(ゼロ)に収束させることを可能にする。実データを用いた実験では HyperMap の有効性が確認された。
キーワード：データマイニング　情報可視化　知識発見

1. はじめに

文書、動画、音声をはじめとするマルチメディアデータに対する効率的な検索、クラスタリング、可視化などの処理の実現はデータベース分野において重要かつ挑戦的な課題となっている。これらの処理については、マルチメディアデータから特徴量を抽出し、高次元空間の点として対処することが一般的な手法である。しかし、高次元空間では次元数が大きくなるに伴い空間が疎らになり、空間内の各点がお互いに遠ざかるという現象が起こる。この現象について、以下に例を示す。図 1 は半径１の円とその外接正方形を表している。ここで OB は円の半径であり、2 次元の場合、$\overline{AB}=\sqrt{2}-1$、$\overline{OB}=1$である。しかし、100 次元の場合、$\overline{AB}=\sqrt{100}-1=9$ であるから、$\overline{AB}$ は$\overline{OB}$ の９倍にもなり、３次元空間に慣れた人間にとっては不思議に感じる現象が現れることがわかる。このような高次元データ空間に対しては、R-tree や BIRCH[4]などの低い次元空間で効率的な検索やクラスタリングアルゴリズムは性能を発揮できなくない。これは、いわゆる“次元の呪い”という厄介な問題として知られている。

また、高次元空間データに対するデータマイニングにおいては、計算の高速性が要求され、データサイズに対してほぼ線形の計算時間しか許されない。このため、ある程度正確性を犠牲にしたよりシンプルなクラスタリングアルゴリズムが多く提案されている。その中で、FastMap[2]というユークリッド距離写像法が、線形の計算時間でピボットを探し出し、ピボットとの距離遠近により、有効なクラスタリングの手法としてよく知られている。

本稿では FastMap を一般化した HyperMap 射影法を提案する。FastMap が二つのピボット毎に再帰的に射影空間の座標値を求めるのに対し、HyperMap は任意個のピボット毎に再帰的に射影空間の座標値を求める。FastMap が線（一次元）を軸とするに対し、HyperMap で構成した軸はｋ次元の平面(あるいは超平面)である。本稿ではそれらの軸は“超軸”と呼ぶ。超軸を採用することにより、残存情報が

速く0に収束することができた。もう一方、FastMap では、射影毎にピボット対が離れた点が選んだが、違い射影のピボットが離れることが保証されない。しかし、HyperMap の場合、一つの超軸において、各ピボットが離れたピボットを選択された。データマイニングの分野で、medoid の抽出するのはもっとも難しい課題である。本稿は HyperMap のピボット選択手法を応用され、その結果を検証した。

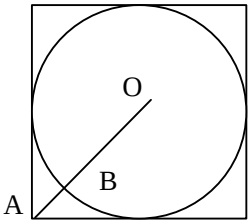

図 1　高次元現象

2. 関連研究

これまで、空間データのクラスタリングに関してさまざまな研究が行われてきた。古典的な PAM(partition around medoids)では、k 個のデータ $O_1, O_2 \cdots O_k$ をクラスタの代表データとして選び、対応するクラスタ $C_1, C_2 \cdots C_k$ を作成する。各データ A は $\min_{1 \le j \le k} dist(A, O_j)$ を満たす O_j、すなわち最も距離の近い medoid が存在するクラスタ C_j に分類される。しかし、どのようにして medoid を選択するかが重要な問題として残っている。PAM では最初に適当に選択し、それを逐次的に改良していくというアプローチをとっている。具体的には、クラスタリングによって割り当てられたデータと medoid の相違度の総和をポテンシャル(クラスタリングの評価関数)として選び、これが小さくなるように改良する。ポテンシャルは

$$\sum_{A \in S} \min_{1 \le j \le k} dist(A,\ O_j)$$

により定義される。ここで、S はデータベースのレコード全体である。PAM 手法は単純であるが、計算時間が非常にかかるため、大規模データベースには適用できない。

この欠点を克服する手法として、ランダムサンプルを用いた CLARA(Clustering LARge Applications)がある。この手法では代表データをデータベース全体から探す代わりに、まず k の 2 倍強程度の個数のランダムサンプルをデータベースからとり、この中で PAM を適用する。しかし、最初のサンプルがクラスタリング結果を左右するため、CLARA では数回（5 回程度）サンプルを取り直し、最もポテンシャルが小さいクラスタリングを出力する。

CLARA は確かに高速であるが、サンプルの選び方以外は PAM をそのまま適用するので、クラスタリングの精度が保証されないという問題がある。この点を踏まえた折衷案的な手法として CLARANS（Clustering Large Applications based on RANdomized Search）[3]がある。CLARA が最初にランダムにサンプルを選ぶのに対し、CLARANS では PAM のアルゴリズムで代表の取り替え候補 X をとるところで、ランダムサンプルから X を選ぶことにより、高速化を可能にした。原論文ではサンプルサイズは n/80 と 250 の大きい方を使う。

本研究では、線形の計算時間の写像アルゴリズム HyperMap を提案する。この手法を利用すると、CLARA や CLARANS などのようにランダムサンプルから候補 medoid を選ぶことなく、最初に最も離

れた候補 medoids を求めることが可能となる。

3. 高次元データの手法

3.1. FastMap の射影法

FastMap では、高次元空間中のオブジェクトの k 個の座標値を以下のように決定する。まず、データ集合中のオブジェクトの対(O_a, O_b)を選び、直線 O_aO_b を k 次元の一本目の軸にとる。オブジェクト O_i の一つ目の座標値 X_i は、O_i の直線 O_aO_b への正射影を E とするとき、X_i で与えられる（図 2)。この X_i は、余弦定理から、次のように求めることできる。ここで、$D(A,B)$ は 2 点 A,B 間の距離を表す。

$$X_i = \frac{(D(O_a,O_b))^2 + (D(O_a,O_b))^2 - (D(O_b,O_i))^2}{2D(O_a,O_{b)}}$$

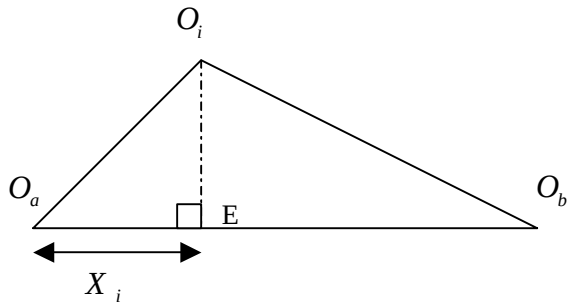

図 2　線 OaOb 上の射影

ここで、X_i を求めるためには O_a,O_b,O_i の 2 点間の距離が与えられていればよく、O_a,O_b,O_i の n 次元空間における座標値は必要ないことに注意されたい。そして、残りの二本目から k 本目の軸は次のように定める。

すべてのデータが n 次元空間上にある場合、直線 O_aO_b と直交する $n-1$ 次元の超平面にすべてのデータを正射影できる。この超平面に射影されたオブジェクト間のユークリッド距離を求めることができれば、一本目の軸と同様に、二本目の軸とその座標値を求めることができる。以上の過程を k 回を繰り返すことにより、元のデータ集合に対して k 次元の情報を抽出することができる。

3.2. HyperMap の射影法

HyperMap では FastMap を一般化し、再帰的に選ぶ射影軸は線（一次元）ではなく、任意の次元数に拡張した。通常、軸は線であるが、本研究の「軸」は任意次元の超平面となり、本稿ではそれを「超軸」と呼ぶ。FastMap は再帰的に一本の線への射影を行うのに対し、HyperMap は再帰的に一つの超平面（点、線も含まれている）への射影を行う。そうした座標系の座標値を求めるため、表 1 の記号を用い、HyperMap の座標値を次のように定義する。

定義１．データ P の超軸の座標値の絶対値は $P \mid H_{p_1 p_2 \cdots p_n}$ から超平面 $H_{p_1 p_2 \cdots p_n}$ までの距離であり、符号は $H_{p_1 p_2 \cdots p_{n-1}}$ を境とし、 p_n と同一方向なら“＋”、そうでないとき“－”である。

n 個ピボットから構成した $(n-1)$ 次元の超軸において、$(n-1)$ の座標値が存在することを注意されたい。

表 1　記号

記号	意味
n	ある超軸の次元数
$H_{p_1 p_2 \cdots p_n}$	n 個の点 $p_1, p_2 \cdots p_n$ によって定まる超平面
$P \mid H_{p_1 p_2 \cdots p_n}$	点 P の超軸 $H_{p_1 p_2 \cdots p_n}$ 上への投影点
$Dist(P, P')$	元空間における点 P と P' の距離
$Dist^{(n)}(P, P')$	$Dist^{(n-1)}(P, P')$ の射影の距離
$D(P, P_1 \cdots P_k)$ $(2 \le k \le n)$	データ P の超軸 $H_{p_1 p_2, \cdots, p_n}$ 上の座標値

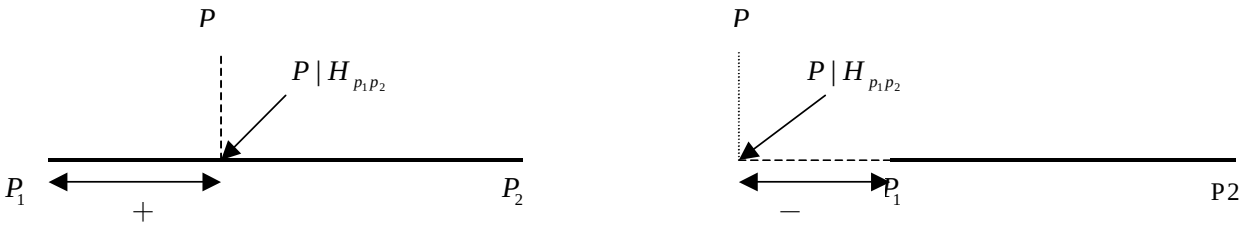

図 3　1 次元の場合の座標値の定義

図 3 で示しているとおり、定義１より、１次元軸の場合、座標値は FastMap の座標値と同じ定義であることが分かる。

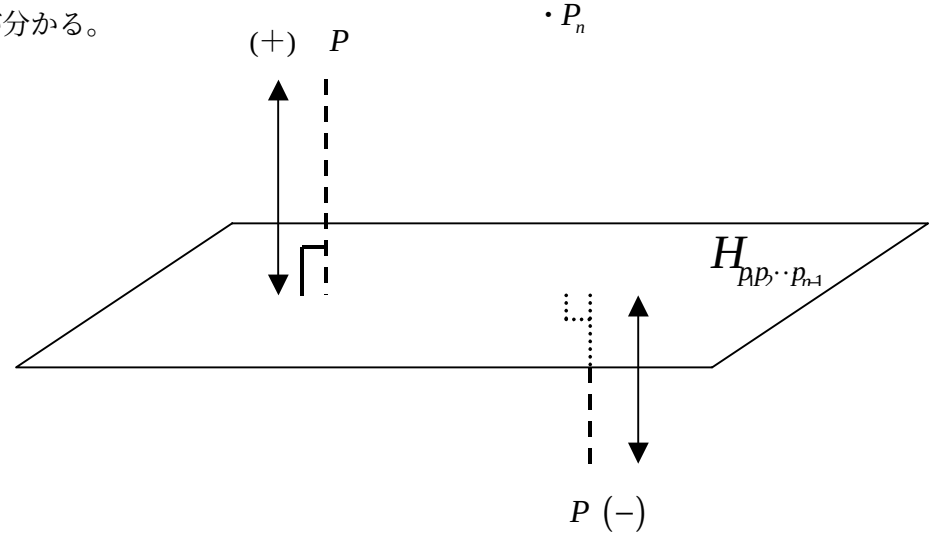

図 4　n 個のピボットから構成した $n-1$ 次元の超軸の射影座標

3.3. 座標値の求め方

① 超軸が線(一次元)の場合、FastMap と同様、

$$D(P,P_1P_2)=\frac{(Dist(P,P_1))^2-(Dist(P,P_2))^2+(Dist(P_1,P_2))^2}{2(Dist(P_1,P_2))} \tag{1.1}$$

② 超軸が平面（２次元）の場合

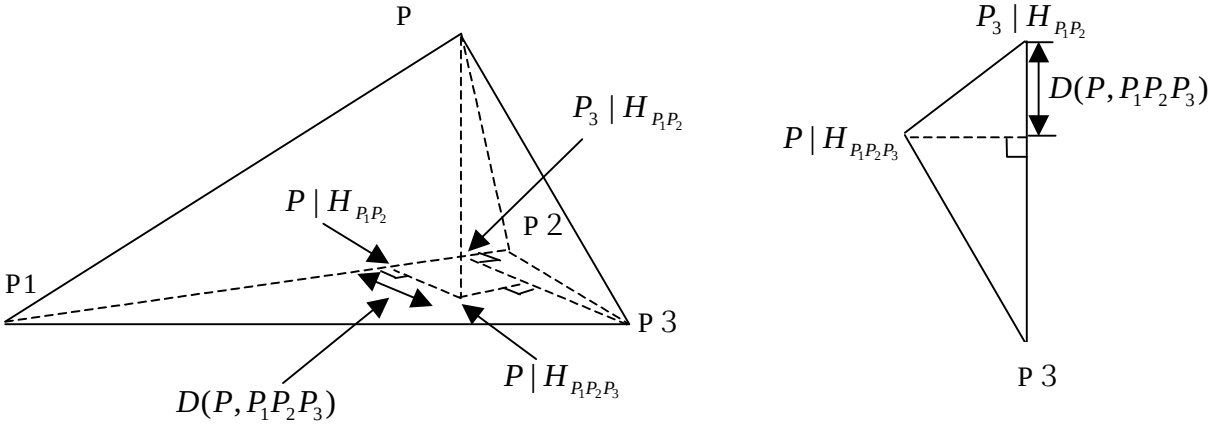

図 5　2 次元超軸の座標計算

$$(Dist(P_3,P_3 \mid H_{P_1P_2}))^2=(Dist(P_1,P_3))^2-(D(P_3,P_1P_2))^2$$

$$(Dist(P \mid H_{P_1P_2P_3},P_3 \mid H_{P_1P_2}))^2=D^2(P,\ P_1P_2P_3)+(D(P_3,\ P_1P_2)-D(P,P_1P_2))^2$$

$$(Dist(P \mid H_{P_1P_2P_3},P_3))^2=D^2(P,P_1P_2P_3)+D^2(P,P_1P_2)-D^2(P,P_1P_3)+D^2(P,P_3P_1)$$

$$=D^2(P,P_1P_2P_3)+D^2(P,P_1P_2)-(Dist(P,P_1))^2+(Dist(P,P_3))^2$$

$$D(P,P_1P_2P_3)=\frac{(Dist(P \mid H_{P_1P_2P_3},P_3 \mid H_{P_1P_2}))^2-(Dist(P \mid H_{P_1P_2P_3},P_3))^2+(Dist(P_3,P_3 \mid H_{P_1P_2}))^2}{2Dist(P_3,P_3 \mid H_{P_1P_2})}$$

$$D(P,P_1P_2P_3)$$

$$=\frac{(D(P_3,P_1P_2)-D(P,P_1P_2))^2-(D(P,P_1P_2))^2+(Dist(P,P_1))^2-(Dist(P,P_3))^2+(Dist(P_1,P_3))^2-(D(P_3,P_1P_2))^2}{2\sqrt{(Dist(P_1,P_3))^2-(D(P_3,P_1P_2))^2}}$$

$$=\frac{(Dist(P,P_1))^2-(Dist(P,P_3))^2+(Dist(P_1,P_3))^2-2D(P_3,P_1P_2)\cdot D(P,P_1P_2)}{2\sqrt{(Dist(P_1,P_3))^2-(D(P_3,P_1P_2))^2}}$$

一般に n 個のピボットから構成された超軸の座標値は以下で表される。

$$D(P,P_1P_2\cdots P_k)=\frac{(Dist(P,P_1))^2-(Dist(P,P_k))^2+(Dist(P_1,P_k))^2-2\sum_{i=2}^{k-1}D(P,P_1\cdots P_i)\bullet D(P_k,P_1\cdots P_i)}{2\sqrt{(Dist(P_1,P_k))^2-\sum_{i=2}^{n-1}(D(P_k,P_1P_2\cdots P_i))^2}} \quad (1.2)$$

$$(k=3,\cdots n)$$

n 個のピボットから構成された $n-1$ 次元超軸において、$n-1$ 個の座標値がある。それらの座標値を求めるには、再帰的に行う必要がある。まず、式(1.1)を用いて、ピボット P_1P_2 に対応する座標値 $D(P,P_1P_2)$ を算出する。次に、式(1.2)を利用し、再帰的に $D(P,P_1P_2\cdots P_k)$ まで $n-1$ 個の座標値を算出する。

3.4. 射影空間上二つデータの距離の求め方

前述のように、超軸と直交する射影空間上の任意二つの点の射影距離が分かれば、再帰的に次の超軸の各座標値の計算が可能である。図 6 で射影空間上の二つ点の距離の求め方が示されている。データ P と P'の超平面 $H_{P_1P_2P_3}$ への射影点をそれぞれ、$P\,|\,H_{P_1P_2P_3}$ と $P\,|\,H_{P_1P_2P_3}$ とすると、二つの射影点の距離は

$$(Dist(P\,|\,H_{P_1P_2P_3},P'\,|\,H_{P_1P_2P_3}))^2=(D(P,P_1P_2)-D(P',P_1P_2))^2+(D(P,P_1P_2P_3)-D(P',P_1P_2P_3))^2$$

から求められる。データ PP'の超平面 $H_{P_1P_2P_3}$ と直交する超平面上の距離 $Dist'(P,P')$ は次から計算される。

$$\begin{aligned}(Dist'(P,P'))^2&=(Dist(P,P'))^2-(Dist(P\,|\,H_{P_1P_2P_3},P'\,|\,H_{P_1P_2P_3}))^2\\&=(Dist(P,P'))^2-(D(P,P_1P_2)-D(P',P_1P_2))^2+(D(P,P_1P_2P_3)-D(P',P_1P_2P_3))^2\end{aligned} \quad (1.3)$$

一般に

$$(Dist^{(n)}(P,P'))^2=(Dist(P,P'))^2-\sum_{i=2}^{n}(D(P,P_1\cdots P_i)-D(P',P_1\cdots P_i))^2 \quad (1.4)$$

が成り立つ

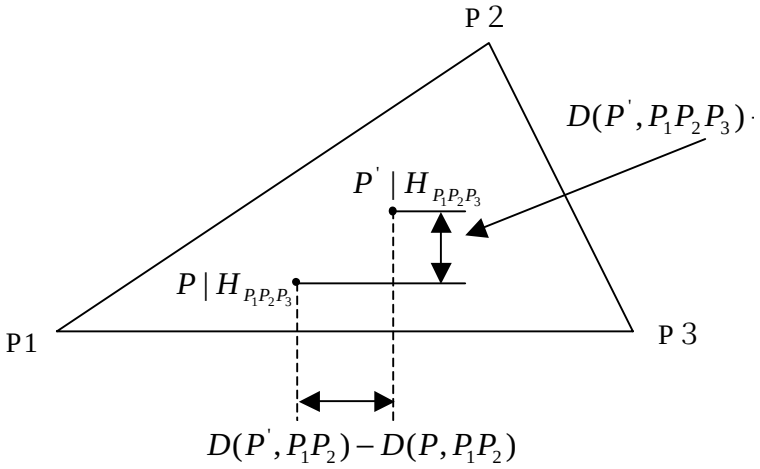

図 6　超軸上のデータ間の距離

3.5. HyperMap アルゴリズム

3.5.2. ピボットの選び法

超軸はピボットによって決められるが、FastMap と同様、最も離れた点をピボットとするのが望ましい。ピボットが３つの場合、線形の計算時間を保つため、まず FastMap と同じ方法で O_a と O_b を定める。次に、Algorithm1(図 7)を用いて、線 O_aO_b まで最も遠いデータを三つ目のピボット O_c とする。同様に、四つ目のピボットは面 $O_aO_bO_c$ までの最も遠いデータを選ぶ。そうすると、互いに遠く離れたピボットから超軸が生成される。点 P から超平面 $H_{P_1P_2\cdots P_k}$ までの距離は式(1.1)及び式(1.2)から得られる。

$$height = (Dist(P,P_1))^2 - \sum_{i=2}^{k}(D(P,P_1\cdots P_i))^2 \qquad (1.5)$$

したがって、O(N)でピボットを求めるが可能である。図７にピボットを選ぶアルゴリズムを示している。ここで、N 個のデータオブジェクトからなる集合は O で表され、各超軸を構成するピボットは npivots で表わされ、二つのオブジェクトの距離関数は dist(O1, O2)で表す。$P_1P_2\cdots P_n$ はその時点の超軸を構成するピボットである。

Algorithm1 choose-distant-object(*O,dist*(),*npivots*)

Begin

1) Set *resultSet* = $\varnothing$
2) Chose Arbitrarily an object called *P*
3) Set P_1 =(the object that is farthest apart from *P*)(according to the distance function *dist*())
4) Add P_1 to *resultSet*
5) While Number of result < *npivots*
   - 5.1) Set P_i = (the object that is farthest apart from hyperplane consisted by *resultSet* Using **Eq** (1.5))
   - 5.2) Add P_i to *resultSet*
6) Report *resultSet* as the desired set of objects.

End

図 7　ピボットを求める

3.5.2. 射影空間の座標値を求める方法

FastMap 手法と同様、HyperMap は元データの座標値を一切利用しない。かわりに元空間上の距離から射影空間を生成する。これは、式(1.4)により残り空間上の各データ間の距離を算出し、また式(1.1)と式(1.2)により射影空間の座標値を求めることができるからである。図８は射影空間の座標値アルゴリズムを示している。このアルゴリズムの入力は表 2 で示す五つのパラメータである。出力は各オブジェクトを各超軸に射影した座標値である。

表 2　HyperMap の入出力変数

入力	N	データ数
	O	オブジェクト集合
	$D(\ )$	距離関数
	k	超軸数
	npivots[]	k本の超軸をそれぞれ定めるピボット数
出力	D[]	3 次元配列(データ順、超軸順、ピボット順)値
	PA[]	2 次元配列（超軸順、ピボット順）元空間の座標値

Algorithm 2 HyperMap

Begin

Global variables:

$N \times k \times npivots$ array D[]

/* using to saving coordinate to every pivots all layers of hypermap for every data*/

$npivots \times k$ pivot array *PA*[]

/* stores the ids of the pivot objects *npivots* per recursive call*/

int *col#*= 0

/* points to the column of the D[] array currently being updated*/

Algorithm *HyperMap(k,Dist( ),O)*

1) If $(k \le 0)$

then {return;}

else {*col#*++;}

2) /* choose pivot objects*/

let *resultSet* be the result of *choose-distant-objects*(*O,D*( ),*npivots*);

3) /* record the ids of the pivot objects*/

PA[] = *resultSet*

4) if (distance of all pivots is 0)

Set D[] = 0 for every data and return

*/*because all inter-object distance are zeros */*

5) /*project the objects on the hyperplane */

for each object,D[] is computed by using **Eq**.(1.1)(1.2).

6) /*consider the projections of the objects on a hyperplane perpendicular to the hyperplane; the distance function $Dist'()$ between two projection is given by **Eq**.(1.4)*/

call *HyperMap*(*k-1, Dist*(), *O*)

End

図 8 HyperMap アルゴリズム

4. 実験評価

有効性を確認するため、この射影手法をクラスタリングに応用し、シミュレーションを行った。このでは、その結果をしめす。

前述したように、これまでに CLARA、CLARANS などのクラスタリング手法が多く提案されている。これらの手法では計算時間を減少するため、全てのデータを対象にするのでなく、ランダムで候補 medoid を選ぶ手法を用いている。本研究では、合成データを用いた実験によりランダムで候補 medoid を選ぶ手法と HyperMap のピボットを medoid とした手法を比較を行った。合成データの指標は表 3 に示す、各クラスタのシードはランダムに生成し、データは平均に分配される。各クラスタにおいて、シードと近いほどデータの密度は高くなり、密度はシードとの距離により正規分布に従う。

クラスタリングの評価はポテンシャルによって行う。すなわちポテンシャル値が小さければ小さいほど、良い結果が得られたものとみなす。

表 3 合成データの指標

データ数	5K, 10K, 50K, 100K
次元数	64, 80, 96, 112
クラスタ数	24, 56

データ数 1000、クラスタ数 24 及び 56 の場合のポテンシャル値を図 9 に示す。縦軸のポテンシャル値は対数目盛で表示されていることに注意されたい。HyperMap と FastMap により選んだピボットを medoid とする場合のポテンシャル値はランダムで選んだmedoid より大幅に減少したことが分かる。FastMap の射影回数はクラスタ数の 1/2 である。HyperMap は射影を 3 回に固定している。実験から最も良い結果となる超軸の分配方法は 1/2,1/3.1/6 である。つまり、クラスタ個数の 1/2 のピボット数を選び、一本目の超軸が構成される。従って、24 クラスタの場合、一本目の超軸の次元数は $24/2-1=11$次元である。二本目超軸クラスタ数の 1/3 のピボット数から構成し、24 個クラスタの場合$24/3-1=7$次元である。三本目超軸のピボット数は$24-12-8=4$で、次元数は 3 である。

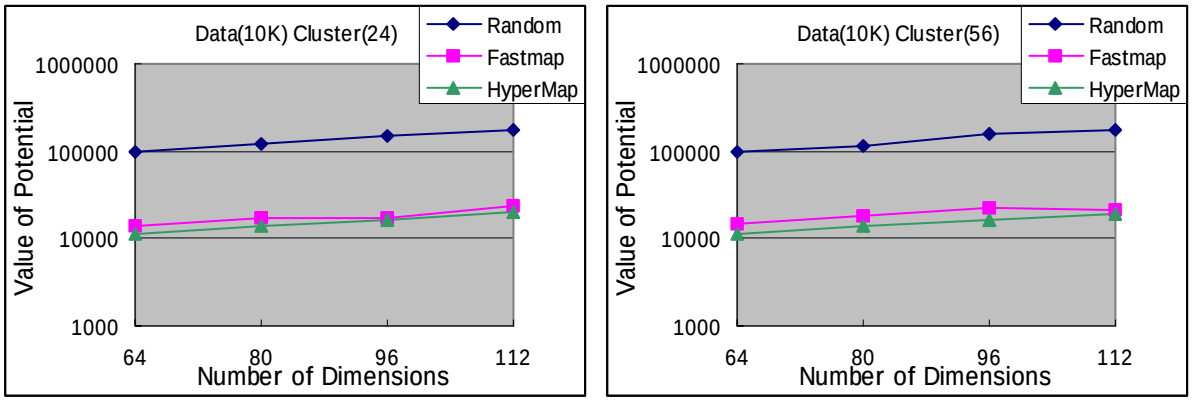

図 9 データ数１００００のポテンシャル値

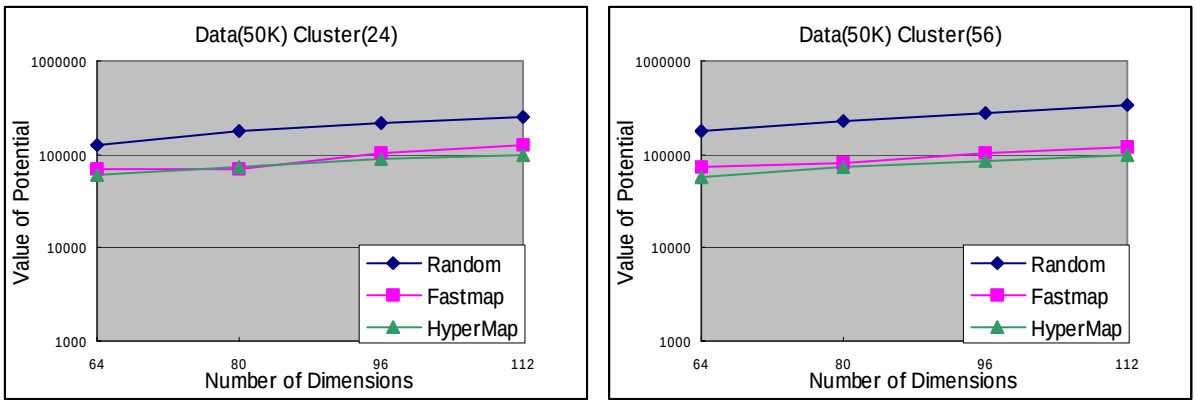

図 10　データ数５００００のポテンシャル値

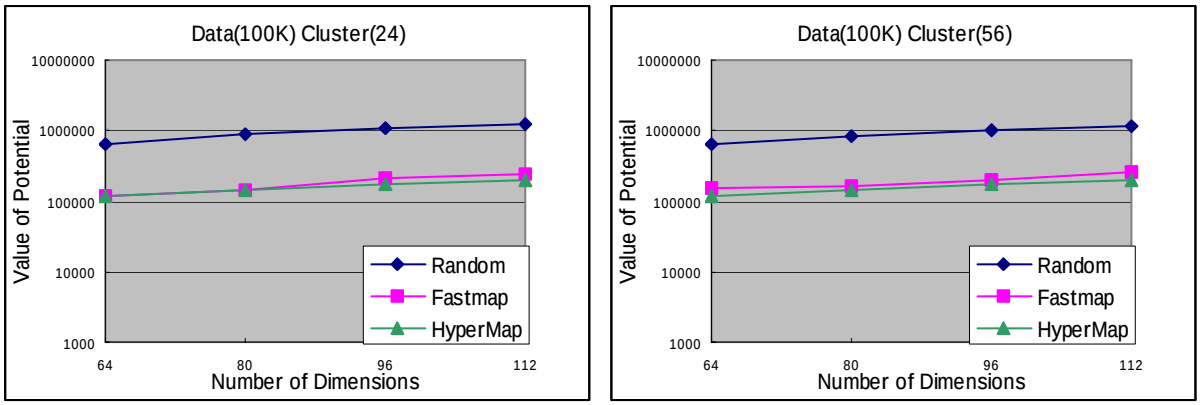

図 11　データ 100,000 のポテンシャル値

図 10 及び図 11 はそれぞれ 50,000 と 100,000 件の実験データに対するポテンシャル値を示している。

5. おわりに

本稿では、FastMap を一般化した射影手法である HyperMap を提案した。この手法により、FastMap の高速性を保ちながら、多くのピボットの情報を利用し、次元情報の迅速な抽出、すなわち次元縮小による情報損失の減少を可能とした。今後は、HyperMap 手法を用い、多次元索引、射影クラスタリングに応用する予定である。

参考文献

[1] Aggarwal C. Charu P. S. Yu et. al..Fast Algorithms for Projected Clustering, Proc. ACM SIGMOD 1999 163-174

[2] Faloutsos, C. and Lin, K. I.: FastMap:A Fast Algorithm for Indexing, DataMining and Visualization fo Traditional and MultiMedia Datasets, Proc ACM SIGMOD June 1995.

[3] Ng, R. T. and Han, J.:Efficient and Effective Clustering Methods for Spatial Data Mining. Proc. VLDB 1994 144-155

[4] Zhang, T. , Ramakrishnan R. and Livny, M.:An Effective Data Clustering Method for Very Large Databases, proc. ACM SIGMOD 1996 103-114

[5] 福田剛志、森本康彦、徳山豪：データマイニング　共立出版　2001