凸多面体を用いた次元縮小法と高次元索引機構

安 際元[†] 古瀬 一隆[††] 陳 漢雄[††]
石川 雅弘[†††] 大保 信夫[††]

本稿では，高次元データの非均一性に着目した新しい次元縮小法とそれを用いた索引機構を提案する．提案手法は，データ空間の凸多面体によって次元縮小を行う．この手法の特徴は，局所的に次元を縮小する点にあり，それによりコンパクトな索引構造の実現が可能となる．この手法の有効性を示すため，本稿では提案手法を VA-file に適用した新しい索引構造 CVA-file（Compact VA-file）を考案した．この索引構造は次元縮小法によって索引ファイルを大幅に縮小する．また，凸多面体の幾何的性質を利用して，各データの縮小した次元の境界（bound）を計算することにより，精度を保ちながら索引ファイルを縮小することが可能になる．実データを用いた実験では CVA-file は主成分抽出により全局次元縮小法から構成した KLT（Karhunun Loeve Transform）空間の VA-file や SR-tree より良い結果を示した．

Dimensionality Reduction Technique with Convex Polyhedra and High-dimensional Index Structure

Jiyuan An,[†] Kazutaka Furuse,[††] Hanxiong Chen,[††]
Masahiro Ishikawa[†††] and Nobuo Ohbo[††]

This paper proposes a new dimensionality reduction technique and an indexing mechanism for high dimensional data sets in which data points are not uniformly distributed. The proposed technique decomposes a data space into convex polyhedra, and the dimensionality of each data point is reduced according to which polyhedron includes the data point. One of the advantages of the proposed technique is that it reduces the dimensionality *locally*. This local dimensionality reduction contributes to improve indexing mechanisms for non-uniformly distributed data sets. To show the applicability and the effectiveness of the proposed technique, this paper describes a new indexing mechanism called CVA-file (Compact VA-File) which is a revised version of the VA-file. With the proposed dimensionality reduction technique, the size of data points stored in index files can be reduced. Furthermore, it can estimate upper and lower bounds of each entry in index files by using geographic properties of convex polyhedra. Results from experimental simulations show that the CVA-file is better than VA-file with dimensionality reduction using KLT (Karhunun Loeve Transform) and SR-tree for non-uniformly distributed real data sets.

1. はじめに

画像や音声をはじめとするマルチメディアデータに対する類似検索に多次元索引構造を用いるのは一般的な手法である．しかし，“高次元の呪い” といわれているように，高次元データに対しては従来の木構造索引が十分な性能を発揮できない．特に，一様データに対しては，類似検索の意味までなくなることが，近年の理論的な研究により明らかにされている[4),6)]．

このような問題に対して，次元縮小手法が有効な手段の 1 つとして知られており[5),7),9)]，これまでに，ピラミッド手法，KLT，FastMap などが提案されている．

ピラミッド手法[5),11)]は，d 次元空間データを一次元で表現し，B+-tree などを用いて索引構造を構築する試みである．**図 1** はピラミッド手法を用いた範囲検索の様子を示したものである．辺長が 1 のデータ空間に辺長 α の検索範囲 q が与えられたとき，検索範囲の中心が白い三角形の中にある場合，検索はそのピラミッド内だけで済む．そうでない場合には，他のピラミッドも検索しなければならない．しかしながら，

† 筑波大学工学研究科
Doctoral Program in Engineering, University of Tsukuba
†† 筑波大学電子・情報工学系
Institute of Information Sciences and Electronics, University of Tsukuba
††† 農業生物資源研究所
National Institute of Agrobiological Sciences

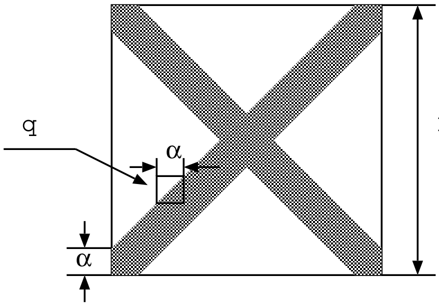

図 1 ピラミッドの範囲検索
Fig. 1 Range retrieval with pyramid technique.

高次元データに対しては，検索範囲がいくら小さくても，多くのピラミッドを検索する必要が生じる．これは，白い三角形の面積 $(1-2\alpha)^d/(2d)$ が次元 d の増加に従って急激に 0 に近付くことから分かる．

また，KLT（Karhunun Loeve Transform）[8]はGDR（Global Dimensional Reduction）による次元縮小の方法として，相関関係にあるデータのいくつかの要因を合成して，失う情報量を最小にできるとされ，よく用いられる．

FastMap [7]はユークリッド空間上の高次元データを低次元に射影することにより，次元を縮小する方法として提案された．また，FastMap 手法を L_1 距離空間に適用した提案もある[9]．

本稿は，高次元データの非均一性に着目し，データごとに有効次元と非有効次元を分け，有効軸に縮小した次元を用いる手法を提案する．この手法では高次元データ空間を凸多面体によって分割し，その凸多面体によってそれぞれのデータの有効次元と非有効次元を決定する．0 に近い座標値の軸は省略可能であるので，それらの軸を非有効次元とする．この次元縮小手法のもう 1 つの利点は，FastMap などの手法とは異なり，凸多面体の幾何的性質を利用し，非有効軸の座標値の境界値を見積もることができる点にある．

本稿の次元縮小手法は多次元索引機構に幅広く応用可能である．その 1 つの例として，本手法を高次元索引機構 VA-file [10]に適用し，新しい索引機構 CVA-file（Compact VA-file）[3]を提案する．次元縮小手法を用いることにより，索引ファイルを縮小することが可能になると同時に，非有効次元の下界（lower bound）と上界（upper bound）を見積もることにより，精度を保ちながら，よりコンパクトな索引機構が実現できる．

以下，2 章において，高次元データの分布の非一様性を観察する．3 章において凸多面体の定義とその幾何的な性質を述べる．4 章では CVA-file の構造と索引機構を示し，凸多面体の性質を利用して境界の計算を行う手法について説明する．5 章では評価実験の結果と他の索引機構との比較結果について述べ，6 章に結論と今後の課題を述べる．

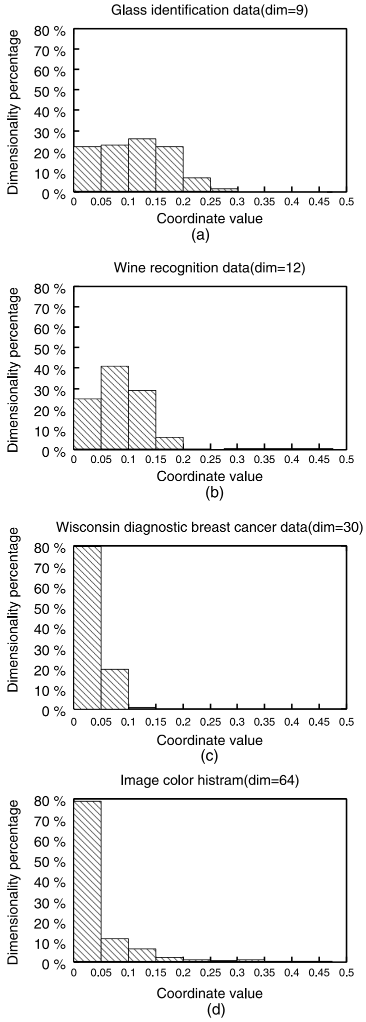

図 2 座標値区間による軸の割合
Fig. 2 Percentage of dimensions according to coordinate.

2. 高次元実データの性質

ここでは，高次元実データの分布の非一様性について述べる．

図 2 は UCI Machine Learning Repository☆のデータを対象としてデータの次元に対する分布を調べたものであり，各データを $(0,1)$ に正規化したときの各座標値に対する次元数の分布を示している．

次元の増加につれ，座標値が $(0.0, 0.05)$ の区間にある軸の割合が増加する．64 次元のカラーヒストグ

☆ http://www.ics.uci.edu/mlean/MLRepository.html

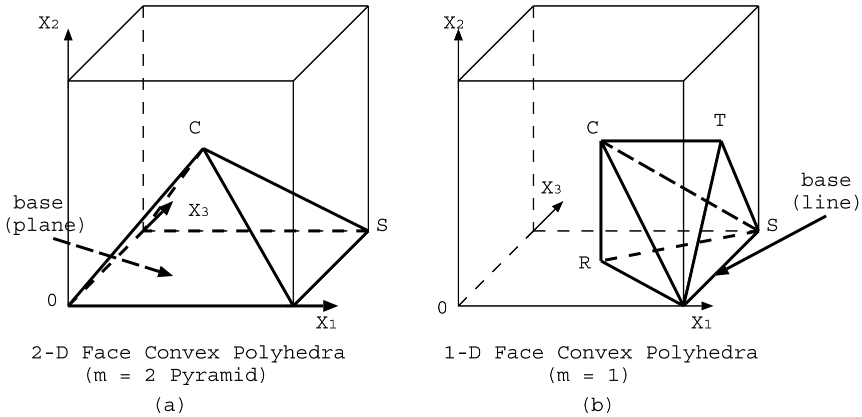

図 3　3 次元凸多面体構造
Fig. 3　Structure of 3-dimension convex polyhedra.

ラムデータについては，78%の座標値は $(0.0, 0.05)$ の区間に，また，11%の座標値は $(0.05, 0.1)$ の区間に入っていることが分かる．つまり，0.1 以下の座標値が約 9 割を占めている．逆の言い方をすれば，高次元データの座標値の 9 割は 0 に近い 1 割の区間に集中している．このことから，残る 1 割の軸のみから構成したコンパクトな索引機構が有効であると考えられる．同様の考えは射影クラスタリングのアルゴリズム PROCLUS [2)]でも用いられている．この手法は，クラスタごとに次元とその数を決定し，その部分空間でクラスタリングを行う．特徴として，新しい低次元座標系を作らず，元の座標系の次元による部分区間でクラスタを見つける点があげられる．データの多くの次元の座標値が 0 に近く，大きい座標値を持つ次元はきわめて少ないデータセットに対しては，主成分分析手法で主成分ベクトル抽出することはきわめて困難であることが予想できる．

3. 高次元空間の凸多面体

d 次元の単位超立方体は $d-1$ 次元の超面によって覆われていると考えられる．同様に $d-1$ 次元の超面は $d-2$ 次元の超面によって覆われているとも考えられる．一般に，d 次元の超立方体は $d-1, d-2, \ldots, 0$ 次元の超面，面，線，点に覆われている．たとえば，立方体は 6 個の面，12 本の線，8 個の点によって覆われている．d 次元の超立方体を覆う m 次元超面の個数は $2^{(d-m)} * \binom{d}{d-m}$ となり，各超面をベース（底面）として，超立方体を等分割できる．このとき，分割した幾何形状は凸多面体になる．

本稿の手法を説明するため，まず，凸多面体の一種であるピラミッド手法[5)]によって 3 次元空間を分割する例について述べる．単位立方体は，その中心 $(0.5, 0.5, 0.5)$ を頂点とし，$(d-1)$ すなわち 2 次元面をベースとした 2×3 個のピラミッドに分割できる．**図 3** (a) は立方体を $x_2 = 0$ の面をベースとして分割したピラミッドを示している．このとき，ピラミッド内の点はすべての面の中でベースとの距離が一番短いことが分かる．すなわち，以下が成り立つ．

$$\begin{cases} x_1 & \geq & x_2 \\ x_3 & \geq & x_2 \end{cases}$$

以下にこの性質を d 次元に一般化したものを示す．

ピラミッドにおける標高の性質：ピラミッド π に対し，π のベースは $x_i = 0$ *or* $x_i = 1$ になる．π 内のデータ $p(x_1, x_2, \ldots, x_d)$ に対し，下記の式が成り立つ．

$$x_j{}' \geq x_i{}' \qquad (j \neq i)$$

ここで

$$x'_k = \begin{cases} x_k & (x_k \leq 0.5) \\ 1.0 - x_k & (x_k > 0.5) \end{cases} \qquad (1 \leq k \leq d) \tag{1}$$

以降，x'_k をデータ p の軸 k における**標高**という．

一般に，ピラミッド手法は d 次元の超立方体に対し，$(d-1)$ 次元の超平面をベースとした分割手法である．本稿ではこの手法をさらに一般化し，m 次元の超平面をベースとした分割手法を提案する．本稿の手法は d 次元の空間を $2^{(d-m)} * \binom{d}{d-m}$ 個の凸多面体によって等分割するが，$m = d-1$ のときはピラミッド手法となり，分割したピラミッドの個数は $2d$ になる．

例として図 3 (b) に $d = 3$，$m = 1$ の場合の分割手法を示している．$m = 1$ であるため，すべてのベースは線である．図示されている凸多面体のベースは下記によって定義可能である．

$$\begin{cases} x_1 = 1 \\ x_2 = 0 \end{cases}$$

ピラミッド手法と同様に，凸多面体内のデータはベースとなる線との距離が他の線より短いという性質を持っている．すなわち，以下が成り立つ．

$$\begin{cases} x_3{}' \geq x_1{}' \\ x_3{}' \geq x_2{}' \end{cases}$$

立方体には 12 本の線があるため，本稿の手法では立方体が 12 等分される．

ここでは，ベースを構成する軸 $x_{j_1}, x_{j_2}, \ldots, x_{j_m}$ を**ベース軸**という．m 次元のベースは下記によって定義できる．

$$x_{j_t} = 0 \;\; or \;\; x_{j_t} = 1 \qquad (m+1 \leq t \leq d) \quad (2)$$

ピラミッドの性質と同様に，以下の性質が成り立つ．

凸多面体における標高の性質：任意の点 $p\,(x_1, x_2, \ldots, x_d)$ に対し，ベース軸における標高は非ベース軸の標高より大きい．つまり，下記の式が成り立つ：

$$x_{j_s}{}' \geq x_{j_t}{}' \quad (1 \leq s \leq m,\; m+1 \leq t \leq d) \;\; (3)$$

$x_{j_t}{}'$ と $x_{j_s}{}'$ は軸 j_t，j_s の標高である．

4. CVA-file の構造と機構

ここでは，前章で述べた凸多面体分割法を高次元索引機構 VA-file に適用した CVA-file について説明する．

4.1 VA-file

VA-file はデータを圧縮することにより線形走査を高速化した索引機構である．その索引機構は各次元の座標値を量子化したビットデータ（近似データ）ファイルとデータファイルから構成されている．2 つのファイルのデータはソートされず，データは同じ順序で対応している．VA-file を用いた k-NN 検索（k-最近傍検索）は 2 段階に分けられる．まず，ビットファイルを走査し，質問点と近似データの距離の下界と上界によってフィルタリングを行い，検索結果の候補を抽出する．次に，候補の座標値をデータファイルから読み込み，質問点との正確な距離を計算する．候補は質問点との距離の下界の順に並んでいる．したがって，すべての候補の座標値をデータファイルから読み込む必要はなく，候補列の中にあるすべての候補の下界が k-NN 検索の k 個目の距離より遠いと分かったとき，残りの候補列に質問点ともっと近いデータは存在しないと断定し，検索を終了することができる．実際の距離と下界の差が小さいときには，正確な距離を計算しなければならない候補の数が少くて済むので，データファイルに対するアクセス回数を減らすことが可能である．

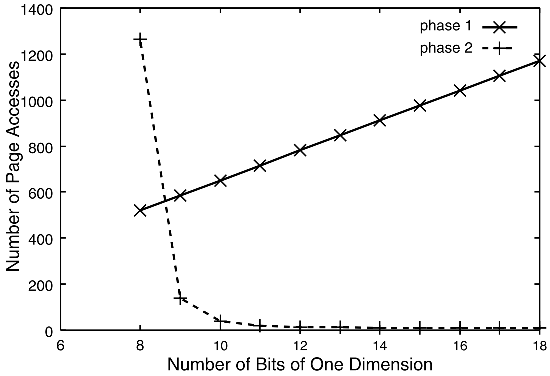

図 4　pruning の効果
Fig. 4　Effect of pruning.

図 4 は 2 つの段階のページアクセス数を示している．実験のデータは 64 次元の一様分布データで，ページサイズは 8 K バイトである．図から分かるように，シーケンシャルアクセスによる第 1 段階のページアクセス数はビットファイルの大きさに比例する．この段階でフィルタリングの役割を十分に果たせないとランダムアクセス方式の第 2 段階のページアクセス数が莫大になることが分かる．第 2 段階のページアクセス数を減らすために下界と上界の幅を狭めることが必要になり，このためには各次元の量子化ビット数を増やさなければならない．一方，量子化ビット数の増加はビットファイルの増大につながり，第 1 段階のアクセスページ数を増加させる結果を招くという問題が生じる．

4.2 凸多面体を用いた次元縮小

前節で述べたように境界とビットファイルのサイズの間にはトレードオフが存在する．VA-file のビットファイルのサイズを短縮するため，本稿では凸多面体による次元縮小法を適用したよりコンパクトなビットファイル作成手法 CVA-file（Compact VA-file）を提案する．

d 次元データセットに対し，$m(\leq d)$ を与えられた場合，3 章で述べたように，m 次元超面をベースとした凸多面体分割が可能である．各凸多面体には同じ本数のベース軸があるが，超立方体を同じ本数のベース軸で分割する必要はないので，データごとに異なるベース軸の本数を決定するアプローチを採用する．このとき，ベース軸の本数 m は標高がパラメータとして与えられた標高閾値 e より小さい軸の数とする．本稿ではデータ v に対し，m 本のベース軸をそのデータの**有効次元**と定義する．b_i はビットファイルの各次元 $i = 1, 2, \ldots, d$ に与えられたビット数とする．VA-file はデータ v に対し，索引ファイルの各エントリにすべての次元の座標値の量子化したビットを格納する．一

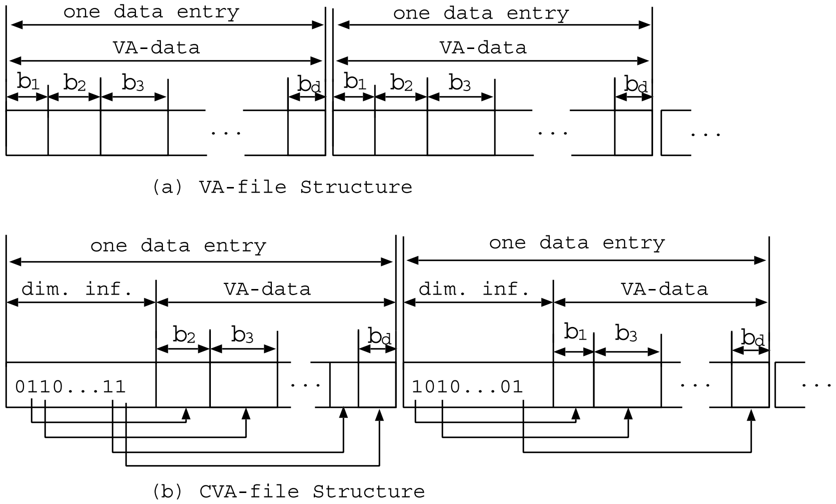

図 5 VA-file と CVA-file 構造
Fig. 5 Structure of VA-file and CVA-file.

表 1 記号と定義
Table 1 Notations and definitions.

e	縮小される軸の座標値の閾値
m	有効次元数
d	次元数
N	データの個数
i	データ番号，$i \in \{1, \ldots, N\}$
v_i	i 番目データ
b	近似データのビット数
$p_{j_t}[s]$	j_t 番目次元の s 番目の区切りの座標値
q	質問点
l_i, u_i	境界：$l_i \leq L_p(q, v_i) \leq u_i$
$v_i.j_t$	v_i の j 番目の座標値
b_{j_t}	j_t 次元の近似ビット数
$r_i.j_t$	データ v_i の j_t 番目の次元の区切り番号
n	検索結果の個数
L_p	距離定義 $L_p(q, v_i)$
$l_i.j_s, u_i.j_s$	l_i，u_i の有効次元 j_s $(1 \leq s \leq m)$ の値
$l_i.j_t{}', u_i.j_t{}'$	l_i，u_i の非有効次元 j_t $(m+1 \leq t \leq d)$ の値

方 CVA-file の各エントリは，**図 5** (b) に示したように，有効次元情報を保存するためのヘッダと有効軸の座標値の量子ビットの 2 つの部分から構成される．

VA-data に座標値を量子化した結果は**セル**（$\lfloor x_1 \times 2^{b_1} \rfloor, \lfloor x_2 \times 2^{b_2} \rfloor, \ldots, \lfloor x_d \times 2^{b_d} \rfloor$）として格納されている．図 5 (b) は CVA-file の構造を示している．ここで，dim. inf. は長さ d のビット列であり，有効軸に対応したビットは “1”，そうでなければ “0” にセットされている．VA-data には有効次元の座標値を量子化した結果がセル（$\lfloor x_{t_1} \times 2^{b_{t_1}} \rfloor, \lfloor x_{t_2} \times 2^{b_{t_2}} \rfloor, \ldots, \lfloor x_{t_d} \times 2^{b_{t_d}} \rfloor$）として格納されている．例として，$d = 5$，$e = 0.2$，$v = (0.9, 0.2, 0.6, 0.3, 0.1)$ の場合について考えてみる．3 章の式 (1) により標高は $v' = (0.1, 0.2, 0.4, 0.3, 0.1)$ となる，標高が閾値 e より大きい第 3 軸および第 4 軸がベース軸（=有効軸）である．このとき，$b_2 = b_3 = 3$，すなわち，2 つの有効軸とも量子化に 3 ビットを割り当てられたとすれば，CVA-file のデータ v のエントリは $(00110, 100, 010)_2$ となる．ここで，ヘッダ “00110” は 3，4 軸が有効軸であることを表し，また，“100” は $\lfloor 0.6 \times 2^3 \rfloor$ から得られ，“010” は $\lfloor 0.3 \times 2^3 \rfloor$ から得られる．

4.3 境界の算出

本稿の提案する次元縮小法は，既存の他の次元縮小法とは異なり，縮小された軸の座標値を見積もることを可能とする．以下の説明で用いる記号の定義を**表 1** に示す．

図 6 はある距離 L_p に対し，質問点 q とデータ v_i の下界 l_i と上界 u_i を示している．

l_i は質問点と v_i が所属するセルの各軸の最短距離の和である．同様に，u_i は質問点と v_i の所属するセルの各軸の最長距離の和である．凸多面体のベースを m 次元とし，有効次元が $X_{j_1}, X_{j_2}, \ldots, X_{j_m}$，非有効次元が $X_{j_{m+1}}, X_{j_{m+2}}, \ldots, X_{j_d}$ とする．l_i と u_i は次の式から得られる：

$$l_i = \left(\sum_{t=1}^{m} l_i.j_t{}^p + \sum_{t=m+1}^{d} l'{}_i.j_t{}^p \right)^{\frac{1}{p}}$$

$$u_i = \left(\sum_{t=1}^{m} u_i.j_t{}^p + \sum_{t=m+1}^{d} u'{}_i.j_t{}^p \right)^{\frac{1}{p}}$$

$l_i.j_t$ と $u_i.j_t (1 \le t \le m)$ は有効次元の下界と上界である．有効次元の座標値（の量子化ビット）は索引ファイルに格納されているので，VA-file のように各次元の $l_i.j_t$ と $u_i.j_t$ を以下のように直接算出できる．

$$l_i.j_t = \begin{cases} q.j_t - p_{j_t}[r_i.j_t+1] & (\text{if} \quad r_i.j_t < r_q.j_t) \\ 0 & (\text{if} \quad r_{i,j_t} = r_{q,j_t}) \\ p_{j_t}[r_{i,j_t}] - v_q.j_t & (\text{if} \quad r_i.j_t > r_q.j_t) \end{cases}$$

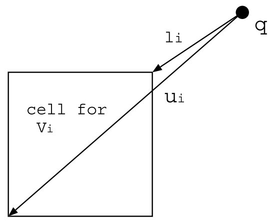

図 6 VA-file の下界と上界
Fig. 6 Lower bound and upper bound in VA-file.

$$u_i.j_t = \begin{cases} q.j_t - p_{j_t}[r_i.j_t] & (\text{if} \quad r_i.j_t < r_q.j_t) \\ \max(q.j_t - p_{j_t}[r_i.j_t], p_{j_t}[r_i.j_t+1] - q.j_t) & \\ & (\text{if} \quad r_i.j_t = r_q.j_t) \\ p_{j_t}[r_i.j_t+1] - q.j_t & (\text{if} \quad r_i.j_t > r_q.j_t) \end{cases}$$

一方，非有効次元について，軸の区切り番号 $r_i.j_t$ $(m+1 \le t \le d)$ は索引ファイル（CVA-file）に格納されていないが，3 章で述べた凸多面体における標高の性質 (3) を利用すれば，**図 7** で示すようにデータ v_i の非有効次元について境界が見積もることが可能である．式 (3) により非有効次元の標高は有効次元より小さいため，図の縦軸を最小標高の有効次元とすると，非有効次元の座標値境界は薄い網掛け部分に入ると断定できる．

一般に，有効次元の最小標高 $p_i.min$ は以下のように求められる．

$$p_i.min = \min(p'_{j_t}[r_i.j_t]) \quad (1 \le t \le m)$$

$$p'_{j_t}[r_i.j_t] = \begin{cases} p_{j_t}[r_i.j_t+1] & (\text{if} \quad p_{j_t}[r_i.j_t] < 0.5) \\ 1.0 - p_{j_t}[r_i.j_t] & (\text{if} \quad p_{j_t}[r_i.j_t] \ge 0.5) \end{cases} \tag{4}$$

また，非有効次元の下界と上界は質問点の位置によって次のように算出可能である．

$$l'_i.j_t = \begin{cases} 0 & (\text{if} \quad 1 - p_i.min < q.j_t) \\ \min(q.j_t - p_i.min, (1 - p_i.min) - q.j_t) & \\ & (\text{if} \quad p_i.min \le q.j_t \le 1 - p_i.min) \\ 0 & (\text{if} \quad v_q.j_t < p_i.min) \end{cases} \tag{5}$$

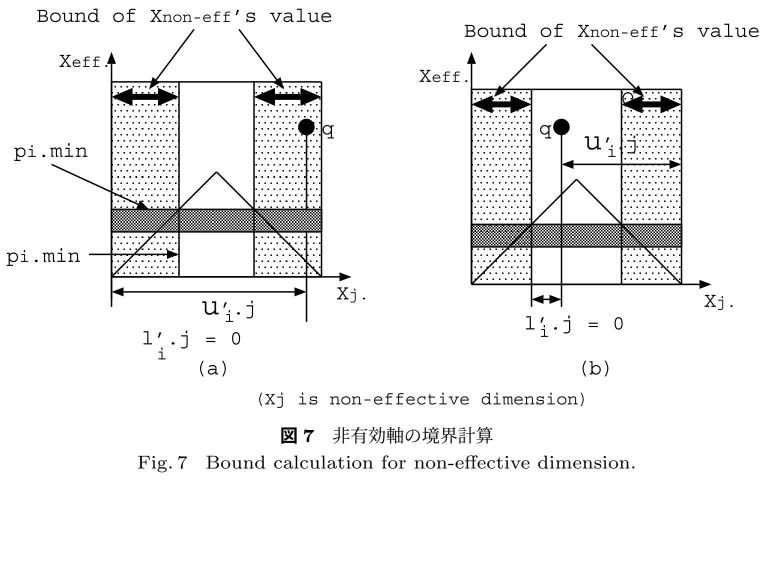

図 7 非有効軸の境界計算
Fig. 7 Bound calculation for non-effective dimension.

$$u'_i.j_t = \begin{cases} 1-v_q.j_t & (\text{if } 1-p_i.min < v_q.j_t) \\ \max(1-v_q.j_t, v_q.j_t) & \\ & (\text{if } p_i.min \le v_q.j_t \le 1-p_i.min) \\ q.j_t & (\text{if } v_q.j_t < p_i.min) \end{cases} \quad (6)$$

このように求めた下界と上界を利用することにより，CVA-file の検索速度を向上させることが可能となる．

5. 実験と評価

CVA-file の索引構造の効果を検証するために，アルゴリズムを実装し，VA-file との比較実験を行った．図 5 からも分かるとおり，CVA-file においては各データごとに量子化する次元数は異なるので，それぞれの VA-data は可変長になる．しかし，固定長のヘッダ $dim.inf$ に各データの量子化される軸が記録されているため，VA-file 同様，単純なデータ構造で実装できる．

VA-file と比較して CVA-file では追加的な計算が必要となるが，**図 8** の結果から分かるように，両者の CPU 時間の差は次元によらず無視できる．したがって，以下ではページアクセス数を基準として CVA-file を VA-file，KLT 空間の VA-file および SR-tree と比較した．

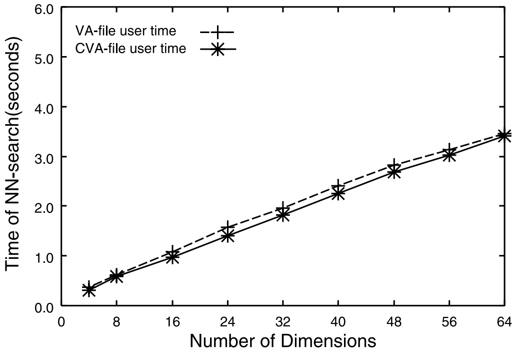

図 8 CPU 時間の比較
Fig. 8 Comparison of CPU time.

VA-file，CVA-file は索引ファイルをメモリにロードして線形走査を行うため，小さいファイルであることが望ましい．VA-file と CVA-file のサイズは理論的な計算により比較できる．N をデータセットのサイズすなわちデータ数とし，b を索引ファイルの 1 つのエントリのビット数とすると，VA-file の大きさは bN となる．これに対して，CVA-file の平均有効次元を $\bar{m}$ とし，m 本の有効軸近似座標値のビット数の平均値 $\bar{b_j}$ をとすると，CVA-file の大きさは $(\bar{m}\bar{b_j}+d)N$ となる．下記の条件を満たせば，CVA-file は VA-file より小さい．

$$bN/(\bar{m}\bar{b_j}+d)N = b/(\bar{m}\bar{b_j}+d) > 1$$

一番簡単なケースとして，すべての次元に同じビット数が与えられた場合を考えると，近似的に $\bar{b_j} \doteq b/d$ が成り立ち，上の式は

$$\bar{m} < d(1-1/\bar{b_j})$$

となる．図 2 で示したように，実データでは $\bar{m}$ が d の 1 割であること，また下の実験で示される $\bar{b_j}$ がほとんどの場合 5 以上であることを考えれば，上の式は簡単に満たされることが分かる．

まず，合成データを用いて，VA-file との比較を行った．2 章の実データに対する分析より，次元が高くなると，0 に近い座標値が急激に増える．本稿では Zipf 分布を用いて，この現象を反映させ，4〜64 合成データを作成した．座標値は n 個の区間 $[0.0, a_1), [a_1, a_2), \ldots, [a_i, a_{i+1}), \ldots, [a_{n-1}, 1.0)$ に分ける．各区間の座標値の数の分布は**図 9** の Zipf 分布で示される．Zipf 分布は式で表すと

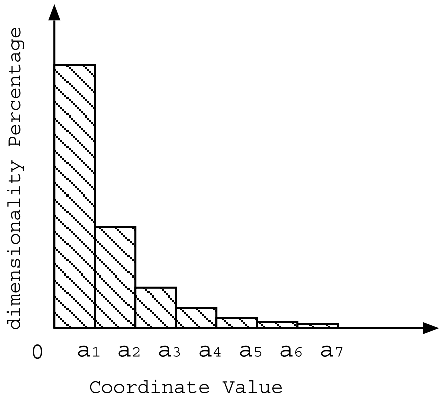

図 9 Zipf 分布
Fig. 9 Zipf distribution.

$$P[X > x] \propto x^{-k}$$

となる．ここで，x は区間の下界値に相当し，k は Zipf の係数である．k が大きいほど，0 に近い座標値の数の比例が大きくなる．本研究では，$k = 2.5$ を採用し，区間は等分に分けた．区間の長さは l で表す．

$$l = a_1 - 0.0 = a_2 - a_1 = \ldots = 1.0 - a_{n-1} = 1/100 \quad (4, 8, 16, \ldots, 32 \text{ 次元の場合})$$

$$l = a_1 - 0.0 = a_2 - a_1 = \ldots = 1.0 - a_{n-1} = 1/200 \quad (40, 48, 56, 64 \text{ 次元の場合})$$

ここでは，100,000 データに対する評価を行った結果を示す．**図 10** は第 2 段階のページアクセス数が同じであるときの第 1 段階のページアクセス数を示してい

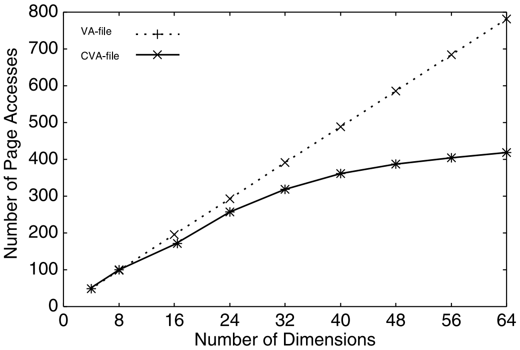

図 10 第 1 段階のページアクセス数
Fig. 10 Number of page accesses in Phase 1.

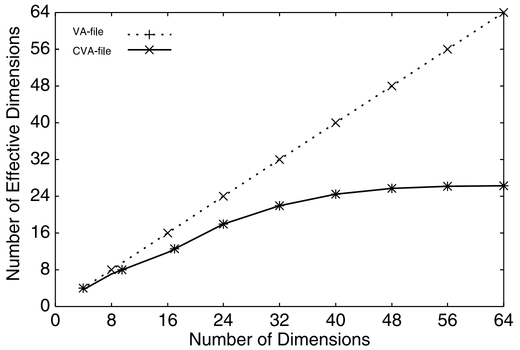

図 11 有効次元数（合成データ）
Fig. 11 Number of effective dimensions (synthetic data).

る．**図 11** はそのときの有効次元数である．

次に，実データを用いて，VA-file，KLT [8]空間に適用した VA-file（本稿ではこれを VA-file/KLT と呼ぶことにする）との比較を行った．KLT 空間の構成手法は付録に示す．

データセット Corel Database☆から抽出した 70,000 枚の画像のカラーヒストグラム☆☆の $4, 8, 16, \ldots, 64$ 次元の特徴ベクトルに対して，ユークリッド距離で 10-NN 近傍検索を行った．ページサイズは 8 K バイトで，VA-file に対して最も良い索引効果となるビット数をテストした．その結果，本稿で用いた実データにおいても，4〜24 次元の場合には，b_j の値を 8 に設定して構成した VA-file が最も良い結果を示した．また，32〜64 次元では，b_j の値を 7 にしたときに，最も良い結果となった．これは，文献 10) に示された結果と一致する．以下の CVA-file との比較ではそれぞれの次元数において最も良い値を b_j に設定したときの結果

☆ http://corel.digitalriver.com/
☆☆ http://kdd.ics.uci.edu/databases/CorelFeatures/CorelFeatures.html

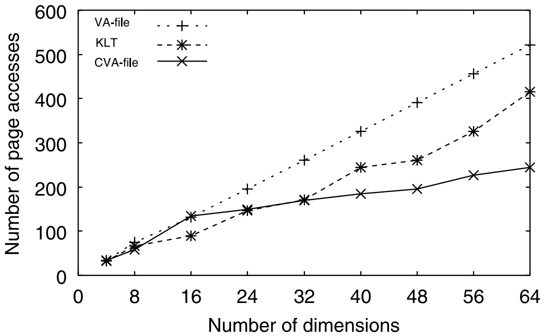

図 12 第 1 段階のページアクセス数
Fig. 12 Number of page accesses in Phase 1.

表 2 第 2 段階のページアクセス数
Table 2 Number of page accesses in Phase 2.

次元数	4	8	16	32	40	48	56	64
ページ数	14	18	21	22	19	23	23	26

を用いている．同様に，CVA-file に対し，最も良い索引効果の有効次元数 m をテストした．文献 10) にも述べらているとおり，ランダムアクセスの効率がシーケンシャルアクセスの 1/5〜1/10 と考えると，第 1 段階で候補が十分に絞り込まれることが望ましい．我々の提案した次元縮小法の有効性を公平に比較するため，各方法が第 2 段階において，ランダムアクセスページ数を十分絞り込んだときの第 1 段階におけるシーケンシャルアクセスページ数を比較した．その結果を**図 12** に示す．なお，このとき各方法の第 2 段階のページアクセス数は同じになる（**表 2**）．10-NN 近傍検索に対し，第 2 段階のページアクセス数は近傍検索結果の 2 倍程度まで絞り込んだ．

KLT 手法は各データの射影の散らばり具合で射影空間軸を取り出す手法であり，最も分散の大きい軸を射影軸とする．図 12 から，次元が高くなると，相対的に分散の良い軸が得られなくなり，回答候補を絞り込むには各軸を量子化するためのビット数が多く必要になることが分かる．結果として全体の索引ファイルが大きくなり，第 1 段階のアクセスページ数が増加した．一方，CVA-file の場合では，第 1 段階のページアクセス数は緩やかな増加にとどまる．その理由としては，実データにおいては，図 2 に示したとおり，次元が増加しても大きい座標値を持つ軸数が増えず，有効次元が次元数に対して線形に増加しないことがあげられる．**図 13** は VA-file，VA-file/KLT，および，CVA-file の有効次元数を示している．

次に第 1 段階と第 2 段階のアクセスページ数の統計についての結果を示す．

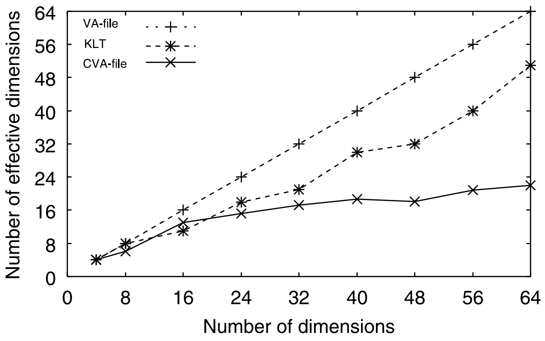

図 13　有効次元数（実データ）
Fig. 13　Number of effective dimensions (real data).

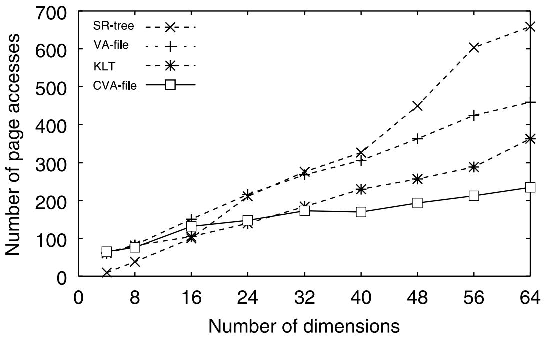

図 14　ページアクセス数
Fig. 14　Number of access pages.

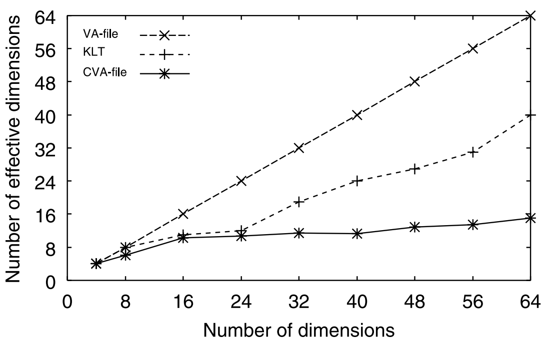

図 15　有効次元数
Fig. 15　Number of effective dimensions.

図 14 は CVA-file，VA-file/KLT，VA-file，SR-tree の次元数と IO アクセスページ数の変化を示している．この図より，特徴次元が高くなると，CVA-file の索引効果が VA-file/KLT，VA-file，SR-tree より有効であることが分かる．たとえば，特徴ベクトルが 64 次元の場合，CVA-file は SR-tree に比べページアクセス数が大幅に減少し，VA-file との比較でもページアクセス数は半分程度になっていることが分かる．**図 15** は CVA-file，VA-file/KLT，VA-file の有効次元を示している．

6. 結　　論

本稿では実データの非一様性に着目した凸多面体分割による局所的次元縮小法を提案した．また，この手法を VA-file に適用した CVA-file 索引機構で実験を行い，実データに対する有効性を示した．今後は，局所的次元縮小法の木構造への適用に関して検討していく予定である．距離の定義は索引の性能に多く影響する．また，一般の距離の定義 L_p に対して p が大きくなると，有効次元が少なくなることが明らかになっており[1]，これはさらなる次元縮小を可能にすると予想される．しかし，文献 6) の結果により，類似検索の有意味性が低くなる恐れがあると指摘されていることをふまえ，今後実データについての距離の定義と検索の有意味性の関係を解明し，効率の良い索引機構の構築について研究を行う予定である．

謝辞　本研究について，国立情報学研究所の片山紀生先生から貴重なアドバイスをいただきました．深く感謝いたします．

参 考 文 献

1) Aggarwal, C., Hinneburg, A. and Keim, D.A.: On the Surprising Behavior of Distance Metrics in High Dimensional Spaces, *Proc. 8th Int. Conf. on Database Theory*, pp.420–434 (2001).
2) Aggarwal, C., Procopiuc, C., Wolf, J., Yu, P. and Park, J.: Fast Algorithms for Projected Clustering, *Proc. 1999 ACM SIGMOD International Conference on Management of Data*, pp.61–72 (1999).
3) 安　際元，古瀬一隆，陳　漢雄，石川雅弘，大保信夫：凸多面体を用いた次元縮小法とそれを利用した高次元索引機構，情報処理学会 DBS 研究報告，Vol.2001, No.71, 2001-DBS-125(II), pp.115–122 (2001).
4) Berchtold, S., Bohm, C., Keim, D. and Kriegel, H.-P.: A Cost Model For Nearest Neighbor Search in High-Dimensional Data Space, *ACM PODS Symposium on Principles of Database Systems*, pp.78–86 (1997).
5) Berchtold, S., Keim, D. and Kriegel, H.P.: The Pyramid-Technique: Towards Breaking the Curse of Dimensional Data Spaces, *Proc. 1998 ACM SIGMOD International Conference on Management of Data*, pp.142–153 (1998).
6) Beyer, K.S., Goldstein, J., Ramakrishnan, R. and Shaft, U.: When Is “Nearest Neighbor” Meaningful, *Proc. 7th Int. Conf. on Database Theory*, pp.217–235 (1999).
7) Faloutsos, C. and Lin, K.I.: FastMap: A

Fast Algorithm for Indexing, Data Mining and Visualization of Traditional and Multimedia Datasets, *Proc. 1995 ACM SIGMOD International Conference on Management of Data*, pp.163–174 (1995).
8) Fukunaga, K.: *Statistical Pattern Recognition*, Academic Press (1990).
9) Shinohara, T., An, J. and Ishizaka, H.: Approximate Retrieval of High-dimensional Data with L_1 Metric by Spatial Indexing, *New Generation Computing*, Vol.18, No.1, pp.39–47 (2000).
10) Weber, R., Schek, H.J. and Blott, S.: A Quantitative Analysis and Performance Study for Similarity-Search Methods in high-Dimensional Spaces, *Proc. 24th International Conference on Very Large Data Bases*, pp.194–205 (1998).
11) 安 際元，古瀬一隆，大保信夫：高次元空間における表面索引構造，第12回データ工学ワークショップ DEWS2001 7A-4 (2001).

付 録

A.1 KLT 空間の構成

n 次元空間から $m(m < n)$ 次元 KLT 空間への変換を考える．

$$\vec{x} \doteq \sum_{i=1}^{m} y_i \phi = \Phi \vec{y} \tag{7}$$

ここで，

$$\Phi = [\phi_1 \ldots \phi_m]$$

また

$$\vec{x} = [x_1 \ldots x_n]^T \qquad \vec{y} = [y_1 \ldots y_m]^T$$

である．$\vec{x}$ は n 次元空間のデータ（ベクトルで表す）とする．$\vec{y}$ は $\vec{x}$ の m 次元 KLT 空間の写像ベクトルを示す．ϕ_i はデータの共分散行列の m 番目までの最大固有値と対応する m 個の固有ベクトルから構成される．共分散行列は対称行列であるため，固有ベクトルは次の式を満たす．

$$\phi_i^T \phi_j = \begin{cases} 1 & for \quad i = j \\ 0 & for \quad i \neq j \end{cases}$$

ゆえに

$$y_i = \phi_i^T \vec{x} \tag{8}$$

である．

例として**図 16** の 2 つのデータセットに対する，1 次元の KLT 空間作成法を示す．

データセットの共分散行列 Σ_x は，

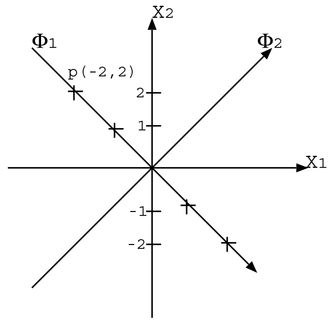

図 16 KLT 空間への変換
Fig. 16 KLT domain translation.

$$\begin{aligned}\Sigma_x &= \frac{1}{4}\sum_{i=1}^{4} \vec{x}_i \vec{x}_i^T \\ &= \frac{1}{4}\left\{ \begin{bmatrix} -1 \\ 1 \end{bmatrix} [\,-1 \;\; 1\,] + \begin{bmatrix} -2 \\ 2 \end{bmatrix} [\,-2 \;\; 2\,] \right. \\ &\quad \left. + \begin{bmatrix} 1 \\ -1 \end{bmatrix} [\,1 \;\; -1\,] + \begin{bmatrix} 2 \\ -2 \end{bmatrix} [\,2 \;\; -2\,] \right\} \\ &= \begin{bmatrix} 10/4 & -10/4 \\ -10/4 & 10/4 \end{bmatrix}\end{aligned}$$

Σ_x の固有値と固有ベクトルは

$$\lambda_1 = 5, \quad \lambda_2 = 0$$

$$\phi_1 = \begin{bmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{bmatrix}, \quad \phi_2 = \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}$$

である．式 (8) より，データ $p_1(-2, 2)$ の 1 次元の KLT 空間の座標値は

$$\phi_1^T \vec{p} = [\; 1/\sqrt{2} \quad -1/\sqrt{2} \;] \begin{bmatrix} -2 \\ 2 \end{bmatrix} = -2\sqrt{2}$$

となる．

(平成 13 年 9 月 20 日受付)
(平成 14 年 1 月 7 日採録)

（担当編集委員 河野 浩之）

安　　際元

1986 年中国合肥工業大学微機所工学科卒業．1998 年九州工業大学工学研究科修士課程修了．同年日立公共システムエンジニアリング（株）に入社．2000 年より筑波大学大学院工学研究科博士後期課程在学中．

古瀬　一隆（正会員）

1993 年筑波大学大学院工学研究科修了.（株）リコーソフトウェア研究所勤務，茨城大学工学部情報工学科助手を経て，1999 年筑波大学電子・情報工学系助手．博士（工学）．

陳　　漢雄（正会員）

1993 年筑波大学大学院工学研究科修了．同年同大学電子・情報工学系助手．1994 年つくば国際大学産業情報学科講師．2001 年筑波大学電子・情報工学系講師．博士（工学）．

石川　雅弘（正会員）

2001 年筑波大学大学院工学研究科修了．同年農業生物資源研究所研究員．博士（工学）．

大保　信夫（正会員）

1968 年東京大学大学院修士課程修了．同年同大学理学部助手．1980 年筑波大学電子・情報工学系講師．1995 年同大学電子・情報工学系教授．理学博士．