データベースシステム 125－80
（2001．7．19）

凸多面体を用いた次元圧縮法とそれを利用した高次元索引機構

安際元[†] 古瀬一隆[‡] 陳漢雄[‡]
石川雅弘[§] 大保信夫[‡]

[†] 筑波大学 工学研究科
[‡] 筑波大学 電子・情報工学系
[§] 農業生物資源研究所

概 要

本稿では，高次元データの非均一性とデータの各次元の相関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，次元縮小手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 bound を計算することにより，精度を保ながら索引ファイル縮小することができる．実データを用いた実験では CVA-file は VA-file や SR-tree より良い結果となった．

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

Jiyuan An[†] Kazutaka Furuse[‡] Hanxiong Chen[‡]
Masahiro Ishikawa[§] Nobuo Ohbo[‡]

[†] Doctoral Program in Engineering, University of Tsukuba
[‡] Institute of Information Science and Electronics, University of Tsukuba
[§] National Institute of Agrobiological Sciences

Abstract

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 and dimensions are interrelated. 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 the VA-file 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, いわゆる"高次元の呪い"のため, 高次元データに対しては従来の木構造索引は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 [2][5].

このような問題に対して, 次元縮小手法は有効な手段の一つとして知られている [3][6][8]. Pyramid-tree 手法 [3] は d 次元空間データを一次元で表現し, B+-tree など索引構造で類似検索ができる. 図 1 は範囲

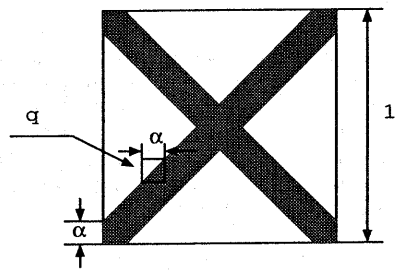

図 1: ピラミッドの範囲検索

検索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 を与えられた時, 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, 高次元データに対しては, 検索範囲は幾ら小さくても, 全てのピラミッドの検索が必要となる. これは, 白い三角形の面積が $=(1-2a)^d/(2d)$ であり, 次元 d が増えると, 面積が急激に 0 に近付くことからわかる.

FastMap[6] はユークリッド空間上の高次元データを低い次元に射影により, 次元を縮小する方法として提案された. [8] では FastMap 手法を L_1 距離空間に適用した.

本論文では, 高次元実データ間の非均一性と相関性に着目し, データ毎にローカルに次元を縮小する手法を提案する. 図 2 は 70,000 件の画像の 64 次元のカラーヒストグラムデータについて 1 に正規化した各座標値の分布を示している. 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

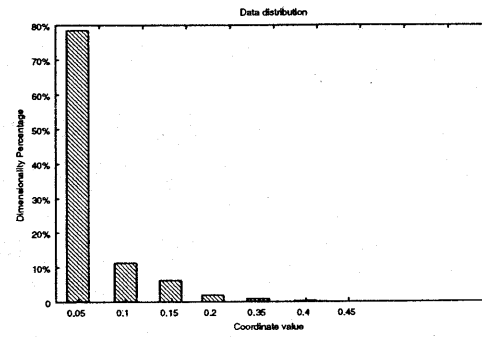

図 2: data distribution

入っていることが分かる. つまり, 0.1 以下の座標値は約 9 割りを占められている. 逆の言い方をすれば, 高次元データの 9 割は 1 割の座標値によって占められている. このことから, この 1 割の軸だけから構成したコンパクトな索引機構が考えられる. 同様の考えは射影クラスタリングのアルゴリズム PROCLUS[1] にもある. この手法は, クラスタ毎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, 元の座標系上次元によって部分区間によるクラスタを見つける点が挙げられる.

本稿は高次元データの非均一性に着目し, データ毎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. この次元縮小手法のもう一つメリットは, 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, 非有効軸即ち縮小された軸の座標値の bound 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できる. その一つの例として, この手法を高次元索引機構 VA-file[9] に適用し, 新しい索引機構 CVA-file(Compact VA-file) を提案する. 次元縮小手法を用いることにより, 索引ファイルを短縮させると同時に, lower bound と 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できる.

以下, 2 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. 3 章では CVA-file の構造を索引機構

を述べ, 凸多面体の性質を利用して bound の計算を行う手法について説明する. 4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, 5章に結論と今後の課題を述べる.

2 高次元空間の凸多面体

d 次元の単位超立方体は $d-1$ 次元の超面に覆われている. 同時に $d-1$ 次元の超面から構成した $d-2$ 次元の超面に覆われている. 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. 例えば, 立方体は 6 個の面と 12 本の線と 8 つの点に覆われている. m 次元超面の個数は $2^{(d-m)} * \binom{d}{d-m}$ となる. 各超面をベース (底面) として, 超立方体の等分割ができる. 分割した幾何形状は凸多面体になる. 分割方法は凸多面体の一種 Pyramid 手法 [3] から説明する. 単位超立方体は, その中心 $(0.5, 0.5, \ldots, 0.5)$ を頂点とし, $(d-1)$ 次元超面をベースとした $2d$ 個のピラミッドによって分割できる. 図 3(a) は立方体が $x_2 = 0$ の面をベースとして分割したピラミッドを示している. ピラミッド内の点は全ての面の中でベースとの距離が一番短いことが分かる. ここではこれを**ピラミッドの性質**という.

性質: ピラミッド π に対し, π のベースは $x_i = 0$ *or* 1 になる. π 内のデータ $p(x_1, x_2, \ldots, x_d)$ に対し, 下記の式を満たす.

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

where

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

以降, x_k' はデータ p の k 軸の**標高**という.

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. 我々はこの手法を一般化し, m 次元の超平面をベースとした分割手法を提案する. ピラミッド手法は単位超立方体を $2d$ 個ピラミッドによって分けられる. 本稿の手法は $2^{(d-m)} * \binom{d}{d-m}$ 個の凸多面体によって等分割する. ピラミッド手法は $m = d-1$ の時の特例である.

図 3(b) は $d = 3, m = 1$ の場合の分割手法を示している. $m = 1$ であるため, 全てのベースは線である. 図の凸多面体のベースは下記によって定義できる.

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

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.

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

立方体には 12 本の線があるため, 本稿の手法は立方体を 12 等分する.

本稿では, ベースを構成する軸 $x_{j_s} (1 \leq s \leq m)$ 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.

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

ベースの個数は $2^{(d-m)} * \binom{d}{d-m}$ である. ピラミッドの性質と同様に, 以下の性質は**凸多面体の性質**という.

性質: 任意の点 $p(x_1, x_2, \ldots, x_d)$ に対し, ベース軸の標高は非ベース軸の標高より大きい. つまり, 下記の式を満たす:

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

x_{j_t}' と x_{j_s}' は軸 i, j の標高である.

3 VA-file に適用 CVA-file

凸多面体分割法による次元縮小法は, これまでに提案されたさまざまな高次元索引に適用できる. 本稿ではこの手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.

3.1 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

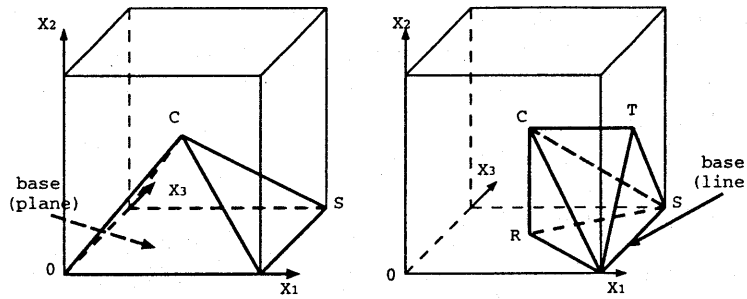

図 3: convex polyhedra

イルとデータファイルから構成されている．二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 で k-NN 検索 (k-最近近傍検索) 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 lower bound と upper bound 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 lower bound 順に並んでいる．従って，全ての候補の座標値がデータファイルから読み込む必要はなく，候補列に候補の lower bound が k-NN 検索の k 個目の距離より遠かったら，候補列に質問点と距離がもっと近いデータは存在しないので，検索が終る．実際の距離と lower bound が近ければ，少ない候補だけ正確な距離を計算すれば済むので，データファイルのアクセス回数を減らすことができる．図 4 は二つ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 次元の一様データで，ページサイズは $8k$ バイトである．シーケンシャルアクセスの第一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二段階のページアクセス数が莫大になることがわかる．第二段階のページアクセス数を減らすために lower bound と upper bound の幅を挟めるには，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，量子化ビット数が増えるとビットファイルが大きくなり．第一段階のアクセスページ数も増加する．

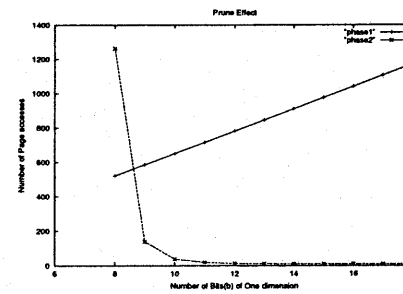

図 4: Effect of prune

3.2 凸多面体と用いた次元縮小

bound 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，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，パラメータ $m(\leq d)$ を与えられた場合，第 2 章で述べた $(d-m)$ 次元超面をベースとした凸多面体分割を行う．各凸多面体は m 本のベース軸がある．我々はデータが所属する凸多面体の m 本ベース軸をそのデータの**有効次元**とする．b_i はビットファイルの各次元 $i = 1, 2, \ldots, d$ のビット数とする．VA-file はデータ v に対する索引ファイルの各エントリの全て次元の座標値の量子ビットを詰める．CVA-file の各エントリには有効次元情報を保存するヘッダと有効軸の座標値の量子ビッ

トの二つの部分から構成される．図 5(a) は VA-file の機構を示している．VA-data に座標値を量子化し

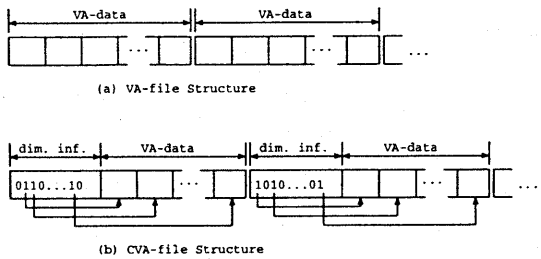

図 5: CVA-file structure

たビット数 ($\lfloor x_1 \times 2^{b_1} \rfloor, \lfloor x_2 \times 2^{b_2} \rfloor, \ldots, \lfloor x_d \times 2^{b_d} \rfloor$) が格納されている．図 5(b) は CVA-file の機構を示している．dim. inf. は有効軸であれば"1"，そうでなければ"0"を示す長さ d の二進数である．VA-data に有効次元の座標値を量子化したビット数 ($\lfloor x_{t_1} \times 2^{b_{t_1}} \rfloor, \lfloor x_{t_2} \times 2^{b_{t_2}} \rfloor, \ldots, \lfloor x_{t_d} \times 2^{b_{t_d}} \rfloor$) が軸の順に格納されている．例：$d = 5$, $m = 2$, $b_2 = b_3 = 3$, $v = (0.9, 0.2, 0.6, 0.3, 0.1)$, とする．第 2 章の式 1 により標高は $v' = (0.1, 0.2, 0.4, 0.3, 0.1)$ となる，標高が大きい 3, 4 軸はベース軸 (=有効軸) である．CVA-file のデータ v のエントリは $(00110, 100, 010)_2$ となる．"00110" は 3, 4 軸は有効軸であることを表す．"100" は $\lfloor 0.6 \times 2^3 \rfloor$ から得られる．"010" は $\lfloor 0.3 \times 2^3 \rfloor$ から得られる．

3.3 CVA-file の bound 算出

本稿の提案する次元縮小法は，今まで他の次元縮小とは異なり，縮小された軸の座標値を算出することができる．以下の説明に用いる記号定義を表 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 lower bound l_i と upper bound 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 所在のセルの各軸の最長距離の和である．凸多面体のベースは m 次元とし，有効次元は $x_{j_1}, x_{j_2}, \ldots, x_{j_m}$, 非有効次元は $X_{j_{m+1}}, X_{j_{m+2}}, \ldots, X_{j_d}$ とする．l_i と u_i は次の式から

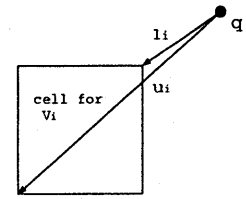

図 6: lower bound and upper bound in VA-file

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 lower bound と upper bound である．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 $l_i.j_t$ と $u_i.j_t$ を以下のように算出できる．

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

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

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，第 2 章で述べた凸多面体の性質 3 を利用すれば，図 7 示すようにデータ v_i の非有効次元の bound が測定できる．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 bound は薄い網掛け部分となる．有効次元の最小標高 $p_i.min$:

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

表 1: 記号と定義

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	bounds: $l_i \leq L_p(q, v_i) \leq u_i$
$v_i.j_t$	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

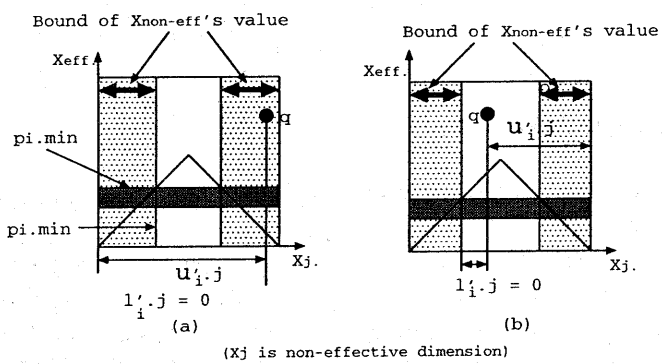

図 7: Estimating face axis bound

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

非有効次元の lower bound と upper bound は質問点の位置によって次のように算出できる.

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

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

4 実験と評価

類似検索においては, CPU 時間は IO アクセス時間と比べ無視できるため, 本研究は IO アクセスページ数だけを基準として VA-file, SR-tree と比較した. VA-file, CVA-file は索引ファイルとして線形走査を行うため, 全てメモリにロードされるため, 小さいファイルであることが望ましい. そこで,VA-file と CVA-file のサイズを計算する. N をデータセットの個数とし, b を索引ファイルの一つエントリのビット数とすると, VA-file の大きさは bN となる. CVA-file の有効次元を m とし, $\bar{b_j}$ を m 本の有効軸近似座標値のビット数の平均値とすると, CVA-file の大きさは $(m\bar{b_j} + d)N$ となる. 下記の条件を満たせば, CVA-file は VA-file より小さい.

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

$\bar{b_j} \doteq b/d$ を代入し

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

データセットは Corel Database Color Database(http://corel.digitalriver.com/) 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である. ページサイズは $8k$ バイトとした場合のユークリッド距離で $10 - NN$ 近傍検索を行った結果を図 8 で示している. VA-file は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,$4, 8, 16, 24$ 次元は $b_j = 8$ に対し, $32, 40, \ldots, 64$ 次元は $b_j = 7$ の時もっとも有効となった. 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. VA-file と比較するため, VA-file の有効次元数 (=元の次元数) も図 9 で示している. 次元が高くなると, 縮小した次元も増え, 特徴ベクトルの次元数が増えても, 有効次元はあまり増えないことが分かる.

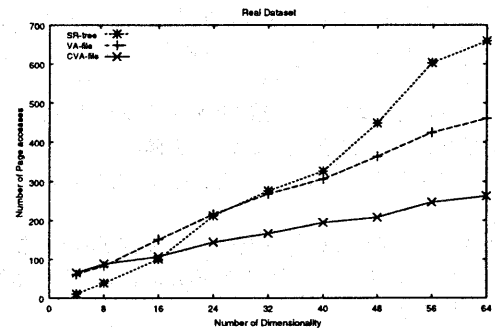

図 8: Comparison on number of page accesses

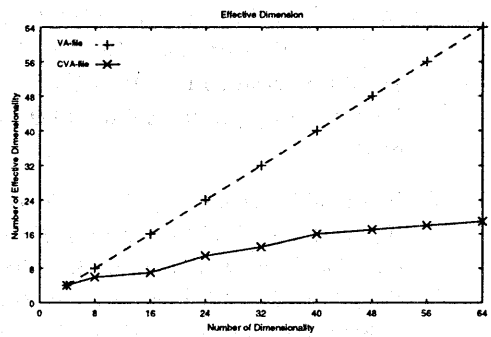

図 9: The effect of dimensionality reduction

図 8 は CVA-file, VA-file, SR-tree の各次元特徴ベ

クトルに対するIOアクセスページ数を示している．特徴次元が高くなると，CVA-fileの索引効果がVA-file, SR-treeより有効であることを示している．特徴ベクトルは64次元の場合，CVA-fileはSR-treeの2/3のページアクセス数に削減ができた．また，VA-fileと比較するとページアクセス数は1/2となっている．

5 結論

本稿では実データは非一様性に着目し，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性を検討していく予定である．距離の定義は索引性能に多く影響する．一般の距離の定義L_pのpが大きくなると，有効次元が少なくなり，更に次元縮小できることが分かる．しかし，[5]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．実データに対し，距離の定義と検索の有意味性の関係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頂きました．深く感謝いたします．

参考文献

[1] Aggarwal C., Procopiuc C., Wolf J., Yu P, Park J.: *'Fast Algorithms for Projected Clustering'*, Proc. ACM SIGMOD Int. Conf. Management of Data, 1999, pp. 61-72.

[2] Berchtold S., Bohm C., Keim D., Kriegel H.-P.: *'A Cost Model For Nearest Neighbor Search in High-Dimensional Data Space'*, ACM PODS Symposium on Principles of Database Systems, 1997,Tucson, Arizona, pp.78-86

[3] Berchtold S., Keim D., Kriegel H.-P.: *'The pyramid-Technique: Towards Breaking the Curse of Dimensional Data Spaces'*, Proc. ACM SIGMOD Int. Conf. Management of Data, Seattle, 1998, pp. 142-153.

[4] Beckmann N., Kriegel H.-P., Schneider R., Seeger B.: *'The R*-tree:An efficient and Robust Access Method for Points and Rectangles'*, Proc. ACM SIGMOD Int. Conf. Management of Data, NJ, 1990, pp. 322-331.

[5] Beyer K. S., Goldstein J., Ramakrishnan R., Shaft U.: *'When Is "Nearest Neighbor" Meaningful'*, Proc. of the 7th Int. Conf. on Database Theory, Jerusalem, Israel, 1999, pp. 217-235.

[6] Faloutsos C., Lin K. I.: *'FastMap: A Fast Algorithm for Indexing, Data Mining and Visualization of Traditional and Multimedia Datasets'*, Proc. ACM SIGMOD Int. Conf. Management of Data, 1995, pp. 163-174.

[7] Katayama N., Satoh S.: *'The SR-tree: An Index Structure for High-Dimensional Nearest Neighbor Queries.'*, Proc. ACM SIGMOD Int. Conf. Management of Data, Tucson, Arizona 1997, pp. 369-380.

[8] Shinohara T., An J., Ishizaka H.: *'Approximate Retrieval of High-dimensional Data with L1 Metric by Spatial Indexing'*, Journal of New Generation Computing, Vol. 18, No. 1(2000), pp. 39-47.

[9] Weber R., Schek H. J., Blott. S.: *'A Quantitative Analysis and Performance Study for Similarity-Search Methods in high-Dimensional Spaces.'*, Proc. of the VLDB conference, New York, 1998, pp. 194-205.