社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

マージン最大化によるメトリック空間分割手法

倉沢　央†　深川　大路††　高須　淳宏††　安達　淳††

† 東京大学大学院　〒 101-8430 東京都千代田区一ツ橋 2 丁目 1-2
†† 国立情報学研究所　〒 101-8430 東京都千代田区一ツ橋 2 丁目 1-2
E-mail: †{kurasawa,daiji,takasu,adachi}@nii.ac.jp

あらまし　メトリック空間を対象とした類似検索索引では，検索処理の枝刈りのために空間を効果的に分割することが求められる．本稿は，最大マージン化によるメトリック空間分割手法，Maximal Metric Margin Partitioning (MMMP) を提案する．MMMP ではデータの分布，特にクラスタ形状にもとづいて空間を分割する．クラスタの境界間の距離が最大になる分割面となるように工夫されている点が本手法の特徴である．本稿はさらに，MMMP を用いた類似検索索引の一例として MMMP-Index も提案する．実験により，MMMP-Index は MMMP の枝刈り効果により検索処理コストを削減できることを示した．特にクラスタ化したデータに対して効果を発揮し，比較手法の 3 分の 2 程度にまで距離計算コストを削減できた．

キーワード　類似検索，検索索引，メトリック空間，マージン

Maximal Metric Margin Partitioning for Similarity Search Index

Hisashi KURASAWA†, Daiji FUKAGAWA††, Atsuhiro TAKASU††, and Jun ADACHI††

† The University of Tokyo　2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, 101-8430 Japan
†† National Institute of Informatics　2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, 101-8430 Japan
E-mail: †{kurasawa,daiji,takasu,adachi}@nii.ac.jp

Abstract　A fundamental issue that confronts the development of an index for similarity searches in metric spaces is how to divide the data effectively for search pruning. We propose Maximal Metric Margin Partitioning (MMMP), a partitioning scheme for similarity search indexes. MMMP divides the data space based on its distribution patterns, especially the boundaries of clusters. A partitioning surface created by MMMP is at maximum distances from the two cluster boundaries. MMMP is the first similarity search index approach to focus on the partitioning surfaces and data distribution patterns. We also present an indexing scheme, named the MMMP-Index, that uses MMMP and pivot filtering. The MMMP-Index discards many objects that are not relevant to a query by MMMP, and it reduces the query execution cost. Experimental results show that MMMP effectively indexes clustered data and reduces the search cost. For clustered vector data, the MMMP-Index reduces the computational cost to less than two thirds that of the compared schemes.

Key words　Similarity Search, Index, Metric Space, Margin

1. はじめに

類似検索は，膨大なオブジェクトの中からクエリに似たオブジェクトを効率的に探す技術である [1]．メトリック空間を対象とした類似検索は，オブジェクト間の距離（類似度）が距離の公理を満たすものであれば，いかなるオブジェクトも扱える．そのため，類似検索はベクトルや文字列，グラフ，タグ，画像などを大量に管理するシステムにおいて広く使われている技術である．我々は，類似検索の問い合わせ処理コストを大幅に削減することを目標に，検索索引構造の改善を試みている．

類似検索の索引付け手法は大きく，*pivot partitioning* と *pivot filtering* の 2 つに分類される．*pivot* は，索引において参照のために使われるオブジェクトを指す．pivot partitioning は，問い合わせ処理中の枝刈りのために，pivot からの距離をもとに空間を複数の部分空間に分割する手法である．一方，pivot filtering は，問い合わせ処理中の不適合オブジェクトの除去のために，pivot とオブジェクト間の距離を蓄積する手法である．両者の大きな違いは pivot とオブジェクトとの距離を蓄積するか否か

という点である．索引サイズで比較すると pivot partitioning のほうが小さくすむ．それゆえ，多くの索引では，まず pivot partitioning で空間を小さな部分空間に分割し，その後にそれぞれの部分空間に対して pivot filtering を適用している．

pivot partitioning の性能は，どのオブジェクトを pivot に設定するか，そしてどの距離で分割するか，という 2 つの要素に大きく左右される．pivot partitioning による理想的な分割状態というのは，少数の pivot でより多くのオブジェクトを問い合わせ処理中に枝刈れるときである．初期の pivot の選択手法は，pivot から各オブジェクトまでの距離が広く分散する pivot が有効であるという経験則に基づいていた [2]．これらの選択方法は，平均値や分散値といった簡素な統計が用いられているのが特徴である．しかしながら，これらの選択方法は無作為に選んだオブジェクトと比較して大した差がないのが実情であった．これに対して，近年の研究ではデータの分布の特徴にもとづいて pivot を選択する手法が提案されている．例えば，iDistance では k-means クラスタリングでクラスタの重心に位置するオブジェクトを探し，それらを pivot として扱う [3]．しかし，クラスタの重心オブジェクトを選択手法は，超球形状で密に分布しているオブジェクトの枝刈りには効果を発揮するが，どの分布にも効果を発揮するわけではない．つまり，これらの手法はクラスタ形状を考慮できないため，クラスタが複数の pivot によって分割されて枝刈りを困難にしうる．

そこで本稿は，新しい pivot partitioning 手法，Maximal Metric Margin Partitioning (MMMP) を提案する．MMMP はクラスタ形状に応じて空間を分割する．まず，MMMP は Density-based Clustering によって任意形状のクラスタから成る階層クラスタを構築する．そして，階層クラスタの上位階層から順に，クラスタの境界にもとづいて空間を分割する．MMMP は pivot による分割面が，分類したいクラスタの境界の間に位置し，かつ，それぞれのクラスタの最も近い端までの距離を最大化するように，pivot と分割距離を選択する．本研究の貢献は，最大マージンという概念を類似検索の索引手法に導入し，偏った分布のデータに効果的な索引を提案したことである．

本稿では，MMMP が類似検索索引のためどのように空間を分割するか説明する．さらに，MMMP を pivot filtering と組み合わせた MMMP-Index という検索索引も併せて紹介する．提案手法の評価として，MMMP による分割性能と，MMMP-Index の索引性能の 2 つの実験結果を紹介する．

本稿の構成は，以下の通りである．第 2. 章では，研究の背景と関連研究を紹介する．第 3. 章と第 4. 章では MMMP とその検索索引手法について説明する．第 5. 章にて評価実験の結果を示し，最後の第 6. 章にてまとめと今後の研究計画を述べる．

2. 背景と関連研究

本章では，メトリック空間の概要と距離関数の例，そして類似検索のための索引手法の関連研究を紹介する．

2.1 メトリック空間と距離関数

本稿の扱う類似検索索引は，距離の公理を満たす関数でオブジェクト間の類似性を定義できるすべてのデータタイプを対象とする．まず，メトリック空間の定義を以下に示す．

オブジェクト集合 D と距離関数 $d : D \times D \mapsto \mathbb{R}$ から成る空間 $M = (D, d)$ が，

non-negativity: $\forall x, y \in D, d(x, y) \geqq 0$
symmetry: $\forall x, y \in D, d(x, y) = d(y, x)$
identity: $\forall x, y \in D, x = y \leftrightarrow d(x, y) = 0$
triangle inequality: $\forall x, y, z \in D, d(x, z) \leqq d(x, y) + d(y, z)$

の 4 つの公理を満たすとき，メトリック空間という．

距離関数 d は，データタイプに適した類似度を表現するため，数多く提案されている．例えば，多次元のベクトルのための Minkowski distance，集合データのための Jaccard's coefficient や Hausdorff distance，色のヒストグラムのような相関する多次元ベクトルのための Quadratic form distance，書誌情報のような文字列のための Edit distance，などが挙げられる．

類似検索で扱われるクエリは，範囲検索 (Range query) や近傍検索 (Nearest neighbor query)，Similarity join などが提案されている．本研究では特に範囲検索を扱う．

2.2 関 連 研 究

類似検索索引は，ページアクセスコストや距離計算コストといった問い合わせ処理コストを削減するための技術である．先に述べたように，多くの関連研究では pivot partitioning と pivot filtering を用いる．本節では，pivot partitioning のための pivot 選択手法に焦点を当てて紹介する．

関連研究は共通して，少数の pivot でより多くのオブジェクトを問い合わせ処理中に枝刈れるような pivot partitioning を目標に研究している．初期の研究では，過去に選んだ pivot やオブジェクトから遠く離れたオブジェクトを pivot としていた [2], [4]．これらは pivot とオブジェクトとの距離が分散し，必要のない pivot を減らすことを念頭においた手法である．これに加え，いくつかの手法では pivot とオブジェクトとの距離が密な部分空間を取り除く試みをしている [5], [6]．いずれの手法も単純な統計値をもとにしており，平均値 [5], [7] や分散値 [2], [6]，合計値 [8]，比 [9] などが用いられている．

近年の研究では，積極的にデータの分布を pivot の選択に利用する試みがなされている．iDistance [3] では，オブジェクトに対して k-means クラスタリングを行い，この重心オブジェクトを pivot とし，空間を Voronoi 分割する．iDistance は各クラスタの位置情報を添付した索引 [10] など現在まで数多くの改善がなされている．List of Clusters (LC) [11] はクラスタリングが意味をなさないようなデータに対して，compact partitioning という pivot partitioning の改善手法を示した．LC はデータ空間を密度にもとづいた小さな部分空間に再帰的に pivot partitioning して索引付けする．

我々は，データの分布を利用することでさらに効果的な索引付けができると考えた．従来手法では，クラスタ形状についての考察が不十分であるがゆえ，クラスタ内のオブジェクトが複数の pivot に分割して管理されることが起こりうる．本稿では，この点を解決する新たな分割手法を提案する．

3. MMMP

本章では，我々が提案するデータのクラスタ形状にもとづいた pivot 選択手法，Maximal Metric Margin Partitioning (MMMP) について紹介する．

まず，MMMP の基本概念を説明する．メトリック空間 $M = (D, d)$ において，pivot p によってドメイン D の中のオブジェクトセット S が 2 つの部分空間 S_1 と S_2 に分割されているときを考える．それぞれの部分空間は，

$$S_1 = \{o \in S \mid d(o,p) \leqq r_p\}, \qquad S_2 = \{o \in S \mid d(o,p) > r_p\}$$

で表現できる．ここで r_p は p による分割距離を表す．このとき，クエリオブジェクト q から距離 r_q の範囲検索を問い合わせたとき，

$$d(q,p) + r_q \leqq r_p \ , \tag{1}$$

を満たすならば，三角不等式の関係から，問い合わせに適合するオブジェクトは S_1 のみを探索すれば良い．同様に，

$$d(q,p) - r_q > r_p \ , \tag{2}$$

を満たすならば，S_2 のみを探索すれば良い．仮に S_1 と S_2 の境界に挟まれるマージンが大きいとき，式 (1) もしくは式 (2) は高い確率で成立する．つまり，S_1 と S_2 のどちらかを高い確率で問い合わせ処理中に枝刈りできることを意味する．一方で，たとえ S_1 と S_2 それぞれのクラスタの重心が離れていたとしても，クラスタの境界が互いに近い場合，枝刈りは成立しにくい．以上の考察をもとに，我々はクラスタ間のマージンを考慮した pivot 選択手法を研究するに至った．

部分空間間のマージンを最大化するような pivot を探すため，MMMP では事前に空間を階層型クラスタリングによって 2 分木構造を生成する．我々はこの過程に Density-based Clustering の OPTICS [12] を用いている．MMMP の処理過程は以下の通りである．まず，巨大なデータセットを索引対象にするため，データセットからランダムに選び出したサブセットに対して OPTICS を実行する．そして，MMMP はクラスタリング結果のルートから順に各階層について，最大マージンでクラスタを分割できるような pivot を選び出す．MMMP による分割例を図 1 に示す．

3.1 Density-based Clustering

Density-based Clustering はオブジェクト空間の密度にもとづいてクラスタに分類する手法である．さらに，このクラスタリングはいずれのクラスタにも分類が難しいノイズオブジェクトも見つけることができる．OPTICS は Density-based Clustering の 1 手法であり，任意形状の階層クラスタを抽出する [12]．OPTICS は各オブジェクトにつき core-distance と reachability-distance という 2 つの値を算出する．OPTICS は過去に選んだオブジェクトから最も近い距離のオブジェクトを選び，直前に選んだオブジェクトからの reachability-distance を出力していく．出力された reachability-distance から階層構造のクラスタを抽出できる．OPTICS の計算量は通常 $O(n^2)$ 必要とする．近年改善手法も提案されているが [13]，依然として大きい．それゆえ，我々はランダムに選んだオブジェクトセットに対してクラスタリング処理することにした．

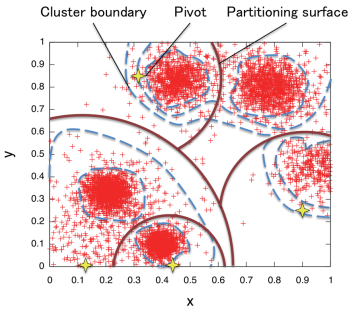


図 1　Partitioning by MMMP

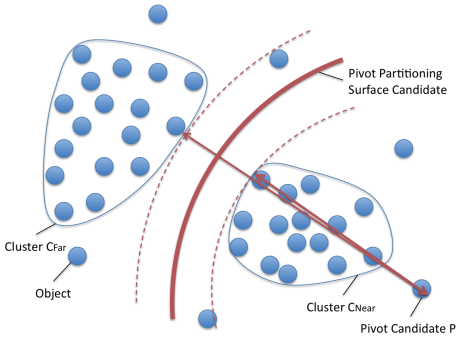


図 2　Pivot Selection in the MMMP

我々が OPTICS をクラスタリング手法として選んだ理由は以下の通りである．

- クラスタ数を事前に設定する必要がない
- 階層型クラスタリングである
- クラスタの重心を抽出することなく，任意形状のクラスタを抽出できる
- オブジェクトが属するクラスタにおける位置が大まかにわかる

3.2 Pivot 選択手法

事前に OPTICS によってクラスタリングされ，階層 2 分木でクラスタが抽出されているとする．木構造の分岐 v において，以下のように pivot を選ぶ．任意のオブジェクト p を用いて，v で分離されるクラスタ 2 つを p からの距離が近い方を $C_{\mathrm{Near}(p)}$，遠い方を $C_{\mathrm{Far}(p)}$ とラベル付けする．このとき，pivot 候補は，

$$\mathrm{pivot} = \underset{p \in S}{\mathrm{argmax}} \left(\min_{o_f \in C_{\mathrm{Far}(p)}} d(p, o_f) - \max_{o_n \in C_{\mathrm{Near}(p)}} d(p, o_n) \right) . \tag{3}$$

によって求まる．式中の第 1 項は $C_{\mathrm{Far}(p)}$ における最も p に近いオブジェクトまでの距離，第 2 項は $C_{\mathrm{Near}(p)}$ における最も遠いオブジェクトまでの距離を示す．つまり，式 (3) はマージンの幅を表していると言える．

MMMP がクラスタ 2 つを $C_{\mathrm{Near}(p)}$ と $C_{\mathrm{Far}(p)}$ にラベル付けするのをたった 1 つのオブジェクトを基準に行う理由として，2 つ挙げられる．1 つめは，pivot 選択手法で生じる計算量を削減するためである．もう 1 つの理由としては，ラベル付けが不適当であっても，そのときの基準となったオブジェクトはほぼ

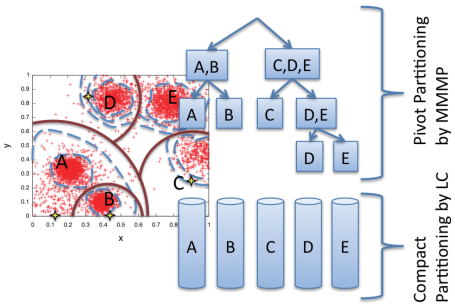


図 3　MMMP-Index

確実に式 (3) で除外されるためである．もちろん，もし pivot として適切なオブジェクトが存在しない場合は，本手法はうまく働かず，式 (3) の右辺の値は負になる．つまり，一部のオブジェクトは OPTICS と MMMP で異なるクラスタに属することもありうる．pivot による分割距離は以下の式にもとづいて設定される．

$$\mathrm{Distance}(p) = \frac{1}{2}\left(\min_{o_f \in C_{\mathrm{Far}(p)}} d(p, o_f) + \max_{o_n \in C_{\mathrm{Near}(p)}} d(p, o_n)\right). \quad (4)$$

図 2 は MMMP による pivot 選択手法を表す．

4. MMMP-Index

本章では，MMMP の適用事例として，MMMP と pivot filtering を組み合わせた類似検索索引 MMMP-INdex を紹介する．前章にて説明した通り，MMMP はクラスタ化している偏ったデータに対して，問い合わせ内容と不適合な部分空間を効果的に処理対象から除くように設計されている．しかしながら，クラスタ数が極端に少なかったり，クラスタ 1 つの大きさが極端に大きいと MMMP の効果は低い．pivot が管理するオブジェクト数は索引の性能を大きく左右する．そこで我々は，図 3 のように MMMP でクラスタの数の部分空間に分割した後に，各部分空間を LC [11] のアルゴリズムで十分に小さい部分空間に pivot を使って分割する，2 段階構造の類似検索索引 MMMP-Index を提案する．

以下の節では，MMMP-Index の索引の構築方法と範囲問い合わせ処理の詳細について述べる．

4.1 索 引 付 け

4.1.1 索引の構築

MMMP-Index の検索索引は以下の 2 段階で構築される．

- 1 段階目の Pivot Partitioning

（1） データセットからランダムに k 個のオブジェクトを選ぶ

（2） k 個のオブジェクトから OPTICS で階層型のクラスタリング構造を抽出する

（3） クラスタ構造のルートの分岐から順に，クラスタ形状にもとづいて MMMP で pivot と分割距離を選択する．

（4） 選んだ pivot から暫定的な木構造の索引をつくる

- 2 段階目の Compact Partitioning

（1） 索引付け対象のすべてのオブジェクトをそれぞれ適切な木構造の葉ノードに割り当てる

（2） x 個以上のオブジェクトが割り当てられる葉ノードについて，LC と同様の手順で新しい pivot によって小さい部分空間に再帰的に分割して，木構造にノードを追加していく．このとき，分割距離は部分空間内のオブジェクト数によって定まる．また，過去に選んだ pivot から最も遠いオブジェクトが pivot として用いられる．

4.1.2 オブジェクトの挿入と削除

オブジェクトの挿入と削除は，まず，オブジェクトがどの pivot によって管理されるべきか，または管理されているかを，pivot の木構造から探索する．そして，オブジェクトが管理されるべき葉ノードに到達した後，挿入または削除を実行する．pivot によって管理されるオブジェクト数が上限値を超えた場合は，LC と同様に部分空間の再分割処理をする．

4.2 範 囲 検 索

MMMP-Index にてクエリオブジェクト q から距離 qr の範囲検索を問い合わせたときの処理過程は，以下の通りである．

（1） pivot の木構造のルートから順に，q から pivot までの距離と三角不等式を組み合わせた式 (1)，(2) で枝刈りする

（2） 問い合わせに適合するオブジェクトが存在しうる，MMMP によってつくられた部分空間すべてにたいして，

（a） 小さい部分空間にたいしても三角不等式で枝刈りをする．枝刈りできない小さい部分空間にたいして，

i. pivot と q との距離と qr からクエリに適合するオブジェクトの存在しうる範囲を計算し，オブジェクトを読み込む

ii. 読み込まれたオブジェクトそれぞれについて，q との距離を計算し，qr の範囲内のものを適合オブジェクトセットに追加する

5. 評　　価

我々は，MMMP-Index の索引付けに伴う計算コストについての評価と，MMMP による分割性能の評価，そして MMMP-Index による索引の評価を行った．

5.1 データセット

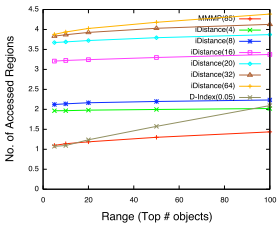
我々は，独自に生成した人工のベクトルデータと UCI KDD Archive [14] から取得した Corel Image Features のベクトルデータを評価実験に用いた．人工のベクトルデータとして，2，8，16 次元の分布を生成した．クラスタの中心は各次元 0 から 1 の範囲でランダムに選び，また，各クラスタの要素数も 1 以上の値にするという制約のもとランダムに選んだ．さらに，クラスタ数を 10，20，30 に変えた 3 種類のデータを生成した．クラスタ中のオブジェクトは正規分布にし，その標準偏差は 0 から 0.10 の範囲でランダムに選んだものと，0 から 0.20 の範囲でランダムに選んだものの 2 種類を生成した．各データセットのクエリは，索引対象のデータと同じ分布で重複のないものを準備した．データセットについての詳細を表 1 に示す．

5.2 索引付けコスト

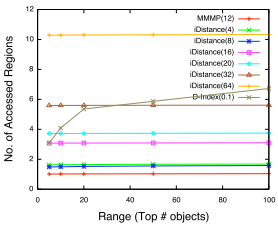
MMMP-Index は，(a) 階層型クラスタリング，(b) pivot の選択とクラスタにもとづいた分割，そして (c) オブジェクト数の多い空間に対する小さい部分空間への分割，の 3 つのステップからなる．ステップ (a) は，OPTICS [12] では $O(nlogn)$ から $O(n^2)$ の計算量となる．ステップ (b) では $O(n^2)$，ステップ (c) では $O(m \cdot logm)$ から $O(m^2)$ 必要とする．n は索引付け対象のオブジェクト数を，m は部分空間中のオブジェクト数を表す．一般的に m は n よりも非常に小さい値である．それゆえ，ステップ (a) と (b) が索引構築のコストの大部分を占める．一

表 1　Data Sets

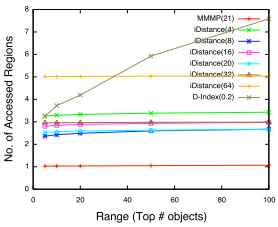
Data	Distance	No. of data	No. of queries	Data source	Dimension	Standard deviation range	No. of clusters
Vectors	Euclid distance	100,000	1,000	synthetic (see sec.5.1)	2	(0,0.10)	20
					8	(0,0.10)	10
					8	(0,0.10)	20
					8	(0,0.10)	30
					8	(0,0.20)	20
					16	(0,0.10)	20
Corel Image Features	Euclid distance	67,040	1,000	UCI KDD Archive	32	-	-



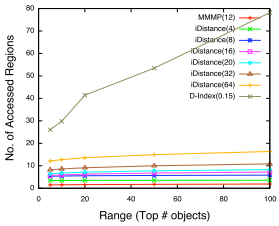
(a) 2-d, 20 clusters, σ:(0,0.10)



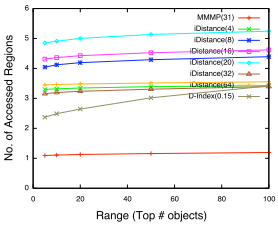
(b) 8-d, 10 clusters, σ:(0,0.10)



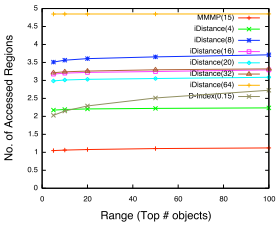
(c) 8-d, 20 clusters, σ:(0,0.10)



(d) 8-d, 20 clusters, σ:(0,0.20)

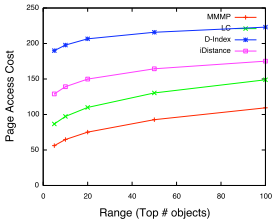


(e) 8-d, 30 clusters, σ:(0,0.10)

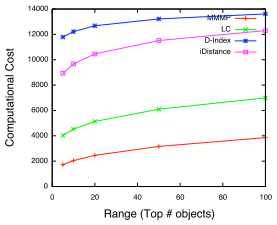


(f) 16-d, 20 clusters, σ:(0,0.10)

図 4　Partitioning Performance

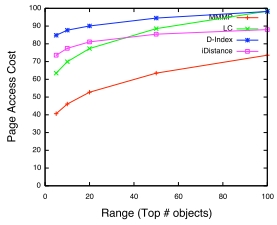


(a) Page Access Cost

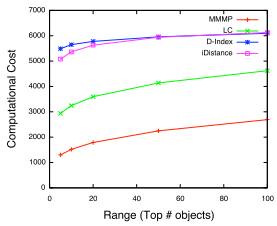


(b) Computational Cost

図 5　Index Performance
(8-d, 10 clusters, σ:(0,0.10))

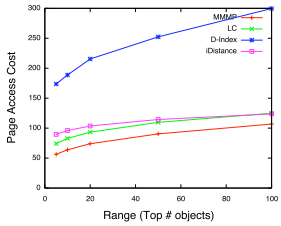


(a) Page Access Cost

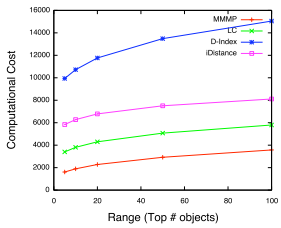


(b) Computational Cost

図 6　Index Performance
(8-d, 20 clusters, σ:(0,0.10))

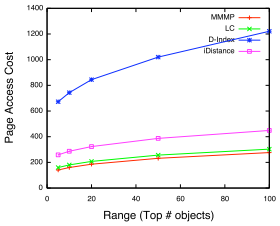


(a) Page Access Cost

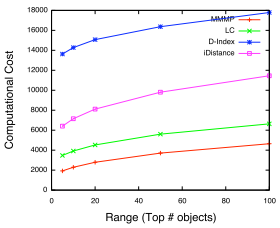


(b) Computational Cost

図 7　Index Performance
(8-d, 20 clusters, σ:(0,0.20))

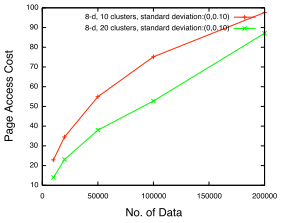


(a) Page Access Cost

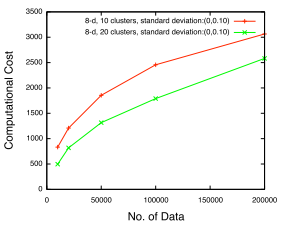


(b) Computational Cost

図 8　Index Performance
(Corel Image Features)



(a) Page Access Cost



(b) Computational Cost

図 9　Index Scalability

方，既存手法の LC [11] は，索引付けに $O(n \cdot logn)$ から $O(n^2)$ 必要とする．つまり，MMMP-Index の索引付けコストは，LC の最悪計算量とほぼ同値であると言える．両手法を実装し実験中に比較したところ，大差のない計算時間であった．

5.3　MMMP の分割性能

我々は既存手法の D-Index [5] と iDistance [3] と MMMP を比較した．D-Index は pivot とオブジェクトとの距離の平均値を用いて，密度の高い空間を Excluded Middle Partitioning で再帰的に分割する手法である．一方，iDistance は k-means でクラスタリングした結果をもとに Voronoi 分割し，クラスタの中心点を pivot に設定する手法である．

問い合わせ処理中に枝刈りできる空間をなるべく大きくするには，クエリの範囲が pivot によって分割された複数の部分空間のうちの 1 つに収まると都合が良い．それゆえ，我々はクエリがいくつの部分空間にアクセスするか測定することにした．実験には人工のベクトルデータを使用した．クエリ範囲は各クエリから 5 から 100 近傍オブジェクトまでの距離を設定した．

MMMP のクラスタリングはランダムに選んだ 20,000 オブ

ジェクトに対して行った．iDistance のクラスタ数のパラメータは 4 から 64 までを試した．D-Index の分割距離のパラメータは，予備実験にて最も検索性能の高かったものに定めた．実験結果は 1,000 クエリに対する結果の平均値で示す．

図 4 は実験結果である．縦軸は問い合わせ時にアクセスした部分空間の数を表す．横軸は問い合わせ範囲の大きさを示す．グラフ中の MMMP と iDistance のラベル横の数字はクラスタ数を，D-Index の横の数字は分割距離のパラメータ値を意味する．実験結果よりすべての実験データに対して，MMMP は最小の部分空間へのアクセスを実現していることがわかった．

本実験は分割性能を評価するのには十分ではあるが，枝刈りに MMMP が効果的か否かは明確でない．我々は部分空間の大きさが適切かどうか調べなければならない．例えば，もし仮に 1 つの部分空間が非常に大きく他の部分空間が非常に小さい場合，問い合わせ時に大きな部分空間のみにアクセスするという事例も起こりうる．しかしながら，クラスタの標準偏差やクラスタの要素数はランダムに選んでいるため，この点について評価することは難しい．それゆえ，我々は索引としての性能を評価する実験を行った．我々は MMMP の枝刈り効果によって，問い合わせ時にページアクセスコストと距離計算コストをどの程度削減できるか，第 5.4 節にて評価する．

5.4 MMMP-Index の索引性能

我々は，MMMP に加えて MMMP-Index の評価を行った．我々は MMMP-Index を iDistance [3]，D-Index [5]，そして LC [11] の 3 つの既存手法と比較した．

先行研究 [1] では，索引の性能はページアクセスコストと距離計算コストによって評価できるとある．我々も先行研究と同様の評価手法を用いた．各手法のパラメータは，予備実験にて我々の実験環境で検索応答時間が最短となったものに設定した．ページサイズは iDistance の論文と同様に 4096 bytes に設定した．実験結果は 1,000 クエリに対する結果の平均値で示す．

実験結果を図 5 から図 8 に示す．各グラフにおいて，赤線は MMMP-Index を，緑線は LC を，青線は D-Index を，そしてピンク線は iDistance を表す．クラスタ化したデータに対する索引として MMMP-Index が優れていることがわかる．8 次元データを例に挙げると，距離計算コストは他手法の $\frac{2}{3}$ 以下になっている．ページアクセスコストもまた他手法よりも下回っている．これら実験結果と第 5.3 節の結果から，MMMP はクラスタ化したデータに対して有効であると言える．図 6 と 7 を比べると，MMMP-Index は標準偏差が小さいデータに対するほうが効果的なことがわかる．この結果から，標準偏差の小さいデータほど偏り度合いが大きく，疎な空間ができやすいため，MMMP によって効果的に分割しやすいと推測される．

最後に，検索索引の規模拡張性について実験を行った．我々はクラスタ数の異なる 2 種類の 8 次元ベクトルデータに対して，データ数を 10,000 から 200,000 まで変化させたときの索引性能を調べた．図 9 は近傍 20 オブジェクトまでの距離を検索したときの検索コストを示している．検索コストは線形増加しているが，クラスタの数が多いほうがコストが小さいことがわかる．我々はこの結果を以下のように解釈している．図 4 からもわかるように，MMMP-Index はクエリが属する部分空間を効果的に見つけている．しかしながら，部分空間のオブジェクトにアクセスするには，LC の索引付け手法を用いているためほぼ線形のコストが必要となる．我々は LC よりも効果的なクラスタのない空間の索引付け手法が必要であると考えている．

6. まとめと今後の課題

本稿では，類似検索索引のための効果的な枝刈りを目指した空間分割手法 MMMP を提案し，評価した．MMMP では，OPTICS によって抽出したクラスタの形状にもとづいて，クラスタの境界間のマージンが最大になるような分割を実現する最適な pivot を選び，空間を分割する．我々はまた，MMMP-Index と呼ぶ MMMP を適用した類似検索索引を提案した．MMMP はクラスタ化したデータに対して効果を発揮する手法である．

我々は MMMP の課題として 2 つ考えている．1 つは，pivot 選択に必要なコストである．我々は pivot の候補を絞り込むことで pivot 選択コストを削減すべきだと考えている．もう 1 つは，頻繁に索引対象のデータが増えたり変更したりする環境への対応である．今後改良に努めていきたい．

文　献

[1] P. Zezula, G. Amato, V. Dohnal and M. Batko: “Similarity Search: The Metric Space Approach (Advances in Database Systems)”, Springer-Verlag New York, Inc. (2005).

[2] P. N. Yianilos: “Data structures and algorithms for nearest neighbor search in general metric spaces”, SODA (1993).

[3] H. V. Jagadish, B. C. Ooi, K. L. Tran, C. Yu and R. Zhang: “idistance: An adaptive b+-tree based indexing method for nearest neighbor earch”, ACM Trans. on Database Systems, **30**, 2, pp. 364–397 (2003).

[4] T. Bozkaya and Z. M. Ozsoyoglu: “Indexing large metric spaces for similarity search queries”, ACM Trans. on Database Systems, **24**, 3, pp. 361–404 (1999).

[5] V. Dohnal, C. Gennaro, P. Savino and P. Zezula: “D-index: Distance searching index for metric data sets”, Multimedia Tools and Applications, **21**, 1, pp. 9–33 (2003).

[6] J. Venkateswaran, D. Lachwani, T. Kahveci and C. Jermaine: “Reference-based indexing of sequence databases”, VLDB (2006).

[7] B. Bustos, G. Navarro and E. Chevez: “Pivot selection techniques for proximity searching in metric spaces”, Pattern Recognition Letters, **24**, 14, pp. 2357–2366 (2003).

[8] P. Ciaccia, M. Patella and P. Zezula: “M-tree: An efficient access method for similarity search in metric spaces”, VLDB (1997).

[9] O. Pedreira and N. R. Brisaboa: “Spatial selection of sparse pivots for similarity search in metric spaces”, SOFSEM (2007).

[10] Y. Zhuang, Y. Zhuang, Q. Li, L. Chen and Y. Yu: “Indexing high-dimensional data in dual distance spaces: a symmetrical encoding approach”, EDBT (2008).

[11] E. Chevez and G. Navarro: “A compact space decomposition for effective metric indexing”, Pattern Recognition Letters, **24**, 9, pp. 1363–1376 (2005).

[12] M. Ankerst, M. M. Breunig, H. Kriegel and J. Sander: “Optics: ordering points to identify the clustering structure”, SIGMOD (1999).

[13] S. Brecheisen, H. Kriegel and M. Pfeifle: “Multi-step density-based clustering”, Knowledge and Information Systems, **9**, 3, pp. 284–308 (2006).

[14] “Uci kdd archive, http://kdd.ics.uci.edu/”.