UBI-tree: ユビキタスデータのためのインデキシング技術

荒川 豊　中村 隆幸　中村 元紀　松尾 真人
日本電信電話株式会社　NTT未来ねっと研究所

1. はじめに

我々はセンサ等デバイスへのデータ送受信を抽象化してアプリケーション構築・運用を支援する実世界データ共有機構 uTupleSpace(uTS)を提案し，実装評価を進めてきた[1]．様々なデバイスが uTS に接続し，多様なデータを蓄積する．また一方で様々なアプリケーションが uTS に接続し，これらの蓄積データに対して多様なアクセスを行う．我々が想定するこのような状況においては，データ種別を跨る，横断的な多次元検索が有用である．例えば，x, y 方向の 2 次元加速度センサから得られる 2 次元データと，x, y, z 方向の 3 次元加速度センサから得られる 3 次元データが混在して蓄積されている場合，x, y 方向の加速度データを必要とするアプリケーションはどちらの種類のデータも検索できると良い．

データベースの分野においては，多次元における範囲検索や近傍検索等を効率化するインデキシング技術の研究が盛んに行われてきた．しかしながら，上記のような，次元の数や種類が異なるデータ集合から横断的な多次元検索を効率的に行う方式は未だ確立されていない．XML-DB や，スキーマレスを謳った DB[2]もいくつか存在するが，我々の知る限り全文検索技術を利用した完全一致検索や 1 次元の範囲検索に対応するものばかりである．

そこで本稿では，最もシンプルな多次元インデキシング技術の一つである R-tree[3]を拡張し，このような次元の数や種類の異なるデータ集合に対する横断的多次元範囲検索を可能とする，新しいインデキシング方式を提案する．

2. 従来技術（R-tree）における課題

2.1. R-tree 概要

R-tree は，多次元データ集合に対する多次元インデキシング技術であり，格納すべき多次元データ集合を最小包囲方形（MBR）と呼ばれる方形領域へ再帰的に分割した平衡木である．各ノードには，データないし子ノードへのポインタと，それ以下の部分木に格納されている多次元データ集合に対する MBR 情報とを対にして格納する．

検索時には，ルートノードから，検索条件範囲と重なる MBR をもつ子ノードのみを辿っていくことで，検索条件に合致するデータを効率的に発見することができる．データを挿入する際には，ルートノードから，データ挿入による MBR の拡大（具体的には，MBR の各次元の幅を掛け合わせた量（V）の増加）がなるべく少ない子ノードを辿り，辿りついたリーフノードへデータを挿入する．これにより，検索実行時のアクセスノード数を削減し，検索効率を高めることができる．ノードが含む子ノードへのポインタ数ないしデータ数には閾値として最小値 E_{min}，最大値 E_{max} が設けられており，データ挿入により E_{max} を超えた場合には，ノード分割を行う．この分割も，分割後の 2 つのノードの V が小さくなるよう行う．

2.2. R-tree の問題

我々が想定する状況においては，下記 2 点の理由から R-tree では必ずしも効率的検索ができない．

・各データの次元は数・種類ともに異なる

このような状況では，子ノード間の MBR の次元が異なるため，その V の値を単純に比較することができず，データ挿入先子ノードの正しい選択や，正しいノード分割ができない．

・全次元が等しく検索条件に用いられるとは限らない

前述のような x, y 方向の 2 次元加速度データを必要とするアプリケーションからの検索頻度が高い場合，R-tree のように単純に全次元の幅を掛け合わせた V を小さくするのではなく，x, y を重視し，x, y 方向の幅をなるべく短くするようインデキシングを行った方が，アクセスノード数を削減できる．

以上の問題を解決するインデキシング技術として，UBI-tree を提案する．

3. 提案技術：UBI-tree

3.1. ユビキタス環境における問合せ分布見積り

新しいデバイスやアプリケーションが動的に発生するユビキタス環境において，どのような検索が行われるか，すなわち問合せ分布を正しく予測することは不可能である．しかしながら，例えばある次元をもつデータが 1 万個，別の次元をもつデータが数個蓄積されている場合であれば，前者の次元が検索条件に含まれる確率が高いと予測される．そこで，ある次元集合 S を用いた検索条件で検索される確率 Pc(S)を次式で見積もる．ただし，T_S は次元集合 S を含む蓄積データ，D_{T_S} はデータ T_S が含む次元数，M は蓄積データ総数を示す．

$$P_c(S) = \sum_{T_s} \frac{1}{M(2^{D_{T_S}} - 1)} \quad (1)$$

UBI-tree: Indexing Method for Ubiquitous Data
Yutaka ARAKAWA, Takayuki NAKAMURA, Motonori NAKAMURA, Masato MATSUO
NTT Network Innovation Laboratories, NTT Corporation

3.2. ノード N がアクセスされる確率

次元 d_i を用いた検索条件で検索を行う際，ノード N がアクセスされる確率を N_{di}/W_{di} で見積もる．ただし，N_{di} はノード N の MBR における次元 d_i 方向の幅，W_{di} は次元 d_i の定義域である．すると，次元集合 S（$=d_1, d_2,\ldots$）を用いた検索条件でノード N がアクセスされる確率は次式で表される．

$$\prod_{d_i} N_{di}/W_{di} \qquad (2)$$

1 回の検索でノード N がアクセスされる確率は，（2）式の期待値である．ある次元集合 S が検索条件に用いられる確率が（1）式で得られることを考えると，（2）式の期待値は次式で表わされる．

$$\sum_S \left(\left(\prod_{d_i} N_{di}/W_{di} \right) \cdot P_c(S) \right) \qquad (3)$$

3.3. UBI-tree: R-tree の多種多次元データ対応化

以上の議論を元に，UBI-tree では R-tree の各アルゴリズムを次のように変更する．

・データ挿入時の子ノード選択アルゴリズム

データ挿入時，V の増加量が小さい子ノードを辿るのではなく，（3）式の増加量が小さい子ノードを辿る．

・ノード分割アルゴリズム

データ挿入により子ノードへのポインタ数ないしデータ数が最大値 E_{max} を超えたノードは，分割後の 2 つのノードに対する（3）式の和がなるべく小さくなるよう分割を行う．

このように，次元が異なりうる V でなく，予想される問合せ分布に基づいて算出した「アクセスされる確率」を統一的基準として用いることで，より効率的な検索を可能とするインデキシングができる．

4. 実験

提案する UBI-tree について，一次評価を行った．

多種多次元のデータに対してインデキシングを行うことは，R-tree ではできないが，B-tree のような 1 次元インデキシング技術を使い，次元毎にインデキシングを行うことは可能である．そこで，（1）UBI-tree を用いる，（2）次元毎に B-tree を用いる，という 2 つの方式で，疑似データを用いたインデキシングを行い，検索時のアクセスノード数を計測した．このとき，疑似データは T 個とし，T 個の中に出現する次元種類数 D は T に比例するものとした（T=100・D）．各データの次元数は平均 d の Poisson 分布に従うとした．また各次元のデータへの出現の仕方は互いに独立とし，1 つのデータに同じ次元が複数回登場しないものとした．各次元の出現頻度は Zipf 分布としたが，各データには同じ次元が複数回出現しないという条件により，正確な Zipf 分布よりややなだらかな分布とした．また各次元の定義域は 0 から 9999 とし，値はランダムに選出した．疑似範囲検索条件は 1000 件とし，各検索条件の出現の仕方は（1）式に従うものとした．また検索条件内の各次元について，0 から 9999 の 2 点をランダムに選び，範囲条件の始点・終点とした．

結果を図 1 に示す．ただし，ノードの E_{min}，E_{max} はそれぞれ 10，20 とした．また B-tree を使う場合には，データないし検索条件を 1 次元ずつに分け，それぞれ独立に挿入・検索した．

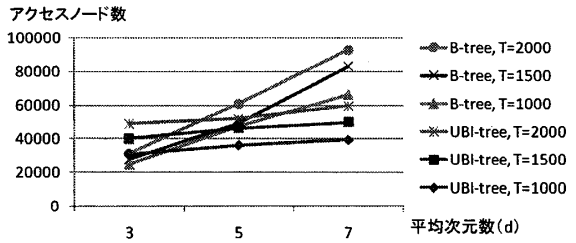

図 1．実験結果

図 1 より，平均次元数の増加に従い，双方ともアクセスノード数は増加するものの，その増加量は提案技術の方が小さくなることが分かる．次元数の大きい多次元検索では，各次元を独立にではなく一度に検索する方が効率良いことを示していると言える．また，この傾向はデータ数に依存しない．

UBI-tree を用いた検索は，各ノードにおける比較処理が複数次元での比較となるため B-tree に比べて大きくなるが，検索条件に含まれる次元数に比例するためそれほど大きくはならないと期待できる．各データの平均次元数がある程度大きい状況においては，UBI-tree は B-tree よりも検索効率の点で有利であると考えられる．

5. おわりに

本稿では，ユビキタス環境における多種多次元のデータに対し，多次元の範囲検索を効率的に行うためのインデキシング技術 UBI-tree を提案した．実験の結果，各データの平均次元数がある程度大きい場合には，従来技術に比べ優れた結果が得られることを確認した．

今後は，従来から R-tree に対して検討されてきた改善方式を提案方式に適用するなど，アルゴリズムの更なる改良を行う予定である．

参考文献

[1] Takayuki Nakamura, Motonori Nakamura, Atsushi Yamamoto, Keiichiro Kashiwagi, Yutaka Arakawa, Masato Matsuo and Hiroya Minami, "uTupleSpace: A Bi-Directional Shared Data Space for Wide-Area Sensor Network," 2nd International Workshop on Sensor Networks and Ambient Intelligence (SeNAmI 2009), December 2009.
[2] http://couchdb.apache.org/
[3] Antonin Guttman. R-trees: A dynamic index structure for spatial searching. In Proceedings of ACM SIGMOD Conference of Management of Data, pages 47-57, 1984.