有限メモリ空間で相関ルールを抽出するマイニングアルゴリズム

2W－8　小幡 康，三石 彰純，山崎 高日子，田中 秀俊，白石 將†
†三菱電機（株）情報技術総合研究所

1 はじめに

データベースから情報を発掘するデータマイニングの技術の中でも、データ間の相関関係を発見しようとする相関ルールの分野では、その処理方式等についてさまざまな提案がなされている。現在開発中であるデータマイニングシステム Knodias の相関ルール抽出エンジンではχ^2適合度検定によるアルゴリズムを採用している[1]が、トランザクションを構成するアイテムが多い場合、アイテムセットを格納するハッシュ木が実メモリ空間に収まらないため、従来の探索方法では実用的な速度での実行ができないという問題があった。今回我々は、アイテムセットの集合を指定されたメモリに格納できる範囲だけ展開する手法を開発した。本稿では、その手法と性能測定について報告する。

2 アルゴリズムの改良

上記のハッシュ木のページングによる性能の問題を解決するための二つの手法について説明する。

2．1 ハッシュ木の分割による手法

この手法では、アイテムセットが格納されたハッシュ木は、ある一定量のメモリ領域内に収まる様な部分集合に分割される。この分割は図 1 の様に、アイテムセットを昇順に並べた先頭から順に部分集合にアイテムセットを追加し、必要なノードの容量の合計を集計し、設定したメモリ領域の容量を超える時点で区切るという方法で行われる。

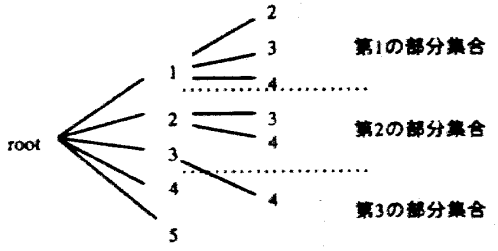

図1ハッシュ木の分割の例

支持度の集計は、まず第 1 の部分集合に存在するアイテムセットの支持度を集計し、次に、第 2、第 3 の部分集合、という具合に部分集合毎に行われる。

2．2 ファイルを利用した手法

この手法では最小支持度を越える一定長のアイテムセットをファイルに格納し、この読み込み用ファイルの先頭からアイテムセットを一件ずつ順に読み、ハッシュ木に追加し、枝伸ばしを行うという手順をハッシュ木がある一定のメモリ量に達するまで繰り返す。そして、この段階のハッシュ木について、支持度の集計、ルールの生成を行う。この処理で新たに生成された最小支持度を満たすアイテムセットは、読み込み用とは別の新規ファイルに格納される。次にそれまでのハッシュ木を破棄し、ファイル中の前回読み込みを終了した位置から、アイテムセットを順に読み、同様に支持度集計からルール生成までを実行する。

以上の処理が読み込み用ファイル中の全ての要素について終了すると、読み込み用ファイルを作成した新規ファイルにかえて、処理対象のアイテムセットの長さを増やし、ファイル読み込みからハッシュ木破棄までの処理を繰り返す。

この手法ではハッシュ木用に確保されるメモリ量が一定であるため、OS が用意するメモリ空間の制限により、プログラムが停止することがないという利点がある。

3 性能測定

3．1 測定方法

前章で説明した二つの手法を評価するため、χ^2適合度検定を利用したアルゴリズムによる正の相関ルール抽出エンジンに実装し、相関ルール抽出の実行時間、ハッシュ木のために確保されるメモリ量の

Algorithm for Mining Rules in Small Memory Space.
Yasushi OBATA, Akitoshi MITSUISHI, Takahiko YAMAZAKI, Hidetoshi TANAKA, Masashi SHIRAISHI †
† Mitsubishi Electric Corporation

測定を行った。測定したエンジンは3種類で、各々は2.1節の手法を採用したKnodias3、2.2節の手法を採用したKnodias5、いずれの手法も採用していないKnodias2である。対象として用いたデータは、ある健康診断の結果を記録する実データから相関がないと考えられるいくつか属性を除いて作成したもので、レコード数は5770件、属性数は337個、属性値数(アイテム数)は1133個、CSVファイルにして5.79MB（エンジンに直接入力するファイルは1.31MB)なる大きさである。エンジンの実行はWindowsNT4.0を搭載したPCサーバApricot FT8000(プロセッサ：Pentium Pro 200MHz)上で行った。

3．2 結果

各エンジンにおける支持度による実行時間とメモリ量の推移を、確信度120%、アイテムセットの長さが3までという条件で測定した結果を図2、3に示す。確信度については、抽出されたルールを結果ファイルに書き込む処理の部分を全体の実行時間から除くための設定である。Knodias3とKnodias5についてはハッシュ木のため確保する一定メモリ量を10MBとした。

実行時間については、(レコード数による)支持度100以降ではKnodias3、Knodias5、Knodias2の順に速い。Knodias5とKnodias3の差は、Knodias3ではハッシュ木が小さいためページングが起こらないのに対して、Knodias5ではアイテムセットを格納したファイルの入出力を行っていることが原因と考えられる。

支持度が100より小さい範囲では、ハッシュ木が実メモリ領域に収まらないため、Knodias5、Knodias3、Knodias2の順に実行時間が短くなっている。Knodias5とKnodias3の差はファイルの読みとページングによるディスクアクセスの違いによるものと考えられる。支持度100付近ではKnodias3の特性に急激な変化がみられるのに対し、Knodias5ではその特性を維持していることから、2.1節の手法ではハッシュ木の増大による実行時間の急激な上昇を押さえられないことが分かる。

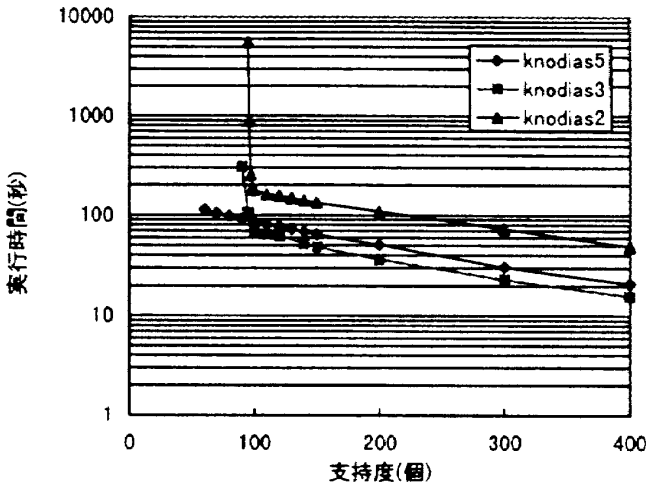

図2 各エンジンにおける実行時間の推移

メモリ量に関しては、Knodias2とKnodias3ではハッシュ木にアイテムセット全てを収めているため、支持度の減少に対して増加するのに対し、Knodias5では、一定値が維持されている。

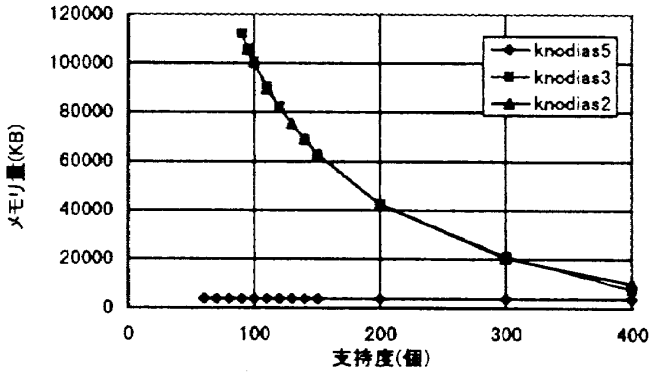

図3 各エンジンにおけるメモリ量の推移

4 おわりに

現在、このエンジンの更なる改良を行っている。今後は負の相関ルール抽出における各手法の性能評価と、メモリ量の見積もり方法等の問題の解決が課題である。

参考文献

[1]三石,他: Knodiasにおけるデータマイニング方式., 第56回情処全国大会1998.
[2]Agrawal, R., Srikant, R. : "Fast Algorithm for Mining Association Rules", Proc. VLDB '94.