省メモリ版相関ルール抽出エンジンの開発

1T－3

三石 彰純，小幡 康†
†三菱電機（株）情報技術総合研究所

1 はじめに

大量のデータから相関ルールを抽出するデータマイニングにおいて、ルールを高速に抽出するアルゴリズムが提案されてきた[1,2]。しかしこれらのアルゴリズムは主記憶上で処理を行っており、マイニング対象データによっては膨大なメモリ空間を必要とするため、マイニング処理が不可能となる場合がある。我々が現在開発を進めているデータマイニングシステム Knodias では、一時ファイルを併用することで少ないメモリ空間の中でマイニングを行うアルゴリズムを開発してきた[3]。本稿では、従来方式の問題点、省メモリ版マイニングエンジン開発の狙い、目標、アルゴリズム、性能向上に関して行った種々の試みについて述べる。

2 開発の狙いと目標

Knodias は主として医療データや品質管理データから相関係数に関して有用な相関ルールを抽出することを目的として開発している。これらのデータ(以下、DB)は１つのレコードに多数(平均 50～100)のアイテムが存在し、アイテムの組み合わせ(以下、ＩＳ)数が膨大となるため、巨大なメモリ空間を必要とし、マイニング処理が不可能な場合が多い。このような場合、マイニングを行う対象のアイテムをあらかじめ絞り込み、マイニングの負荷を下げざるを得ないが、人為的にアイテムの絞り込みを行った場合は有用な相関ルールを抽出できない危険性がある。

そこで我々は、一時ファイルを利用することでレコード長が 50～100 の“長いレコード”に対しても実用的な処理速度が得られることを目標としてマイニングアルゴリズムの開発を行った。なお、我々は結論部のアイテム数が１のルールを重視しており、効率上の理由もあって結論部の長さを１に限定している。

A Data Mining Engine using Small Memory Space.
Akitoshi MITSUISHI, Yasushi OBATA †
† Mitsubishi Electric Corporation

3 従来方式の問題点

従来のマイニングアルゴリズムではマイニング処理に必要なＩＳをすべて主記憶上に木構造として展開していることが最大の問題点であった。レコード長が高々十数アイテムの場合には大きな問題とならないが、50～100 あるいはそれ以上になると、仮想メモリ空間でも足りない状況が発生する。

ここでレコード長の増加に伴うＩＳ数の増加を考えるため、レコード長が 10、レコード数が 10×ｎの DB1 とレコード長が 100、レコード数がｎの DB2 におけるＩＳ数を考える。簡単のため各 DB に含まれるアイテムはそれぞれ 100×ｎ種類、すなわち全てのアイテムが異なるとすると、DB1 に含まれるＩＳ数は約 $10^4\times n$ であるのに対して、DB2 では約 $2\times10^{13}\times n$ に達し、両者の比は約 2×10^9 になる。すなわち、DB1 と DB2 では総アイテム数は同一であるにも係わらず、ＩＳ数は９桁も違ってくる。現実にはこれほどの差は出ないが、レコード長の増加がアイテムセット数の増加に与える影響の大きさは容易に推測できる。

図１は apriori アルゴリズム[1]を元にしたマイニングアルゴリズムを示している。図において、L_k は長さｋのラージアイテムセット(DB 内の出現頻度が最小支持度以上のアイテムセット、以下ＬＩＳ)の集合、C_k は長さ k の候補ＩＳの集合を意味する。ここで、ステップ 4)～ステップ 7)において、C_k、L_k をメモリ上の木構造で表現することを前提としており、C_k、L_k の数が膨大になった時に処理が不可能となる。

1) L_1を作る
2) k=2
3) L_{k-1}が空なら終了
4) L_{k-1}からC_kを作る
5) DBを調べて、C_kの各要素の出現度数を数える
6) C_kの中から最小支持度を満たさないものを削除し、残りをL_kとする
7) L_k、L_{k-1}、L_1から長さ k のルールを抽出する
8) k++
9) 3)へ戻る

図１ 従来方式

4 小ロット方式

図２に今回開発した省メモリ版マイニングアルゴリズム“小ロット方式”の基本アルゴリズムを示す。図において、F_kは長さｋのＬＩＳについて、そのアイテム番号と出現頻度を記録した一時ファイルを意味している。小ロット方式では、L_kをメモリではなく一時ファイルF_kに保持しておき、ユーザから指定されたメモリ領域に入る分だけＬＩＳをメモリに読み込んで木構造を作成する。すなわち、従来方式で展開される木構造の部分木のみを主記憶に展開し、その部分木に対してマイニング処理を行う。そのため従来方式と比較して DB と木構造の照合処理の回数は増加する。

上記の基本アルゴリズムに加えて、性能を向上させるために下記の工夫を行っている。

①兄弟ＬＩＳの一括処理

長さ k-1 のＬＩＳから長さ kの候補ＩＳを生成する時に、先頭からk-2個のアイテムが同じ“兄弟ＩＳ”を一括して処理することによって、不必要な候補ＩＳが生成されることを防いでいる。

② 不要候補ＩＳの削除

木構造を生成する過程において、中間的に作成し、最終的には不要であるノードが残る場合がある。この不要ノードを削除するステップを設けることで木構造とDBとの照合処理を高速化している。

③メモリ管理

木構造を作成するためのメモリ領域を処理の開始時に一括して確保し、その中をマイニングエンジンで管理することでメモリ管理を高速化している。

上記の他に、結果として採用しなかったが下記の改良案も検討した。

④複数レコード一括処理方式

複数のレコードを一括して読み込んで木構造との照合を行う。DBの性質によって効果にばらつきが多い。

⑤木構造の逆引き方式

DBと木構造との照合処理において、木構造側からレコード内のＩＳ有無を調べる。効果が大きくなく、逆効果の場合がある。

⑥Lk-1再照合方式

ルール生成に必要なLk-1の出現度数をDBと木構造を照合した時に調べる。一時ファイルのファイルサイズを縮小することができるが、処理速度が低下するため不採用。ただし、一時ファイルが大きくなる場合には有効であり、再検討する余地がある。

5 まとめ

限られたメモリ空間の中で相関ルールを抽出するマイニングアルゴリズムを開発し、実用的な性能を得ることができた[5]。今後は、より大規模なデータを対象とした評価を行い、改良を加えていく予定である。

参考文献

[1]Agrawal, R., Srikant, R. : “Fast Algorithm for Mining Association Rules” , Proc. VLDB '94, 1994.

[2]Park, J., and Chen M. : “Using a Hash-Based Method with Transaction Trimming for Mining Association Rules”, IEEE Trans. Knowledge and Data Engineering 9(5), 1997.

[3]小幡他: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, 第56回情処全国大会2W-8, 1998.

[4]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2W-6, 1998.

[5]小幡他:省メモリ版マイニングエンジンの評価, 第58回情全国大会1T-4, 1999.


```
1)  L1を作り、F1にセーブする
2)  k＝2
3)    空のFkを作成する
4)      Fk-1が空なら19)へ
5)        使用メモリ量が割当てメモリ量に達していれば9)へ
6)        Fk-1から１レコード読み出し、候補アイテムセットを作成してCkに加える
          Fk-1にレコードが残っていなければ9)へ
7)        作成した候補アイテムセットからルールを抽出するのに必要な長さ k-1 のアイテムセットを作り木構造に加える
8)        5)に戻る
9)      pos=Fk-1のファイル読み出し位置
10)     Ckの出現度数カウントに必要ない枝を隠す
11)     ＤＢを調べて、Ckの各要素の出現度数をカウントする
12)     Ckの中から最小支持度を満たすものをLkとする
13)     Lkのアイテム番号列とその出現度数をFkに出力する
14)     10)で隠蔽した枝を復活する
15)     Fk-1を最後まで読み込み、7)で追加したアイテムセットの出現度数を木構造に転記する
16)     Lk、Lk-1、L1から長さｋのルールを抽出する
17)     木構造を消去する
18)     Lk-1ファイルをposの位置まで巻き戻し、4)へ戻る
19)   Fk-1を削除、Fkが空ならば終了
20)   Fkを rewind してFk-1とする、k++、3)へ戻る
```

図２ 小ロット方式