データベースシステム 119−58
（1999. 7. 23）

一般化相関ルールマイニングの並列処理方式における統計情報を用いた候補分割負荷分散手法の評価

新谷隆彦† 喜連川優†

データマイニングで得られる情報の代表的なものに相関ルールがある．我々は一般化相関ルールと呼ばれるデータの分類階層を考慮した相関ルールの抽出処理性能向上を目的とした並列処理方式の研究を進めて来た．

本稿では，データベース読み出し時に得られるデータの統計情報を利用することにより探索候補の支持度を予測し負荷の均衡化を図る探索候補割当手法を研究室で構築した大規模PCラスタ上に実装し，性能評価を行った．このPCクラスタは100台のパーソナルコンピュータをATMネットワークで接続したシステムである．本性能測定により，提案する負荷分散手法により大規模システム上でも負荷の偏りの影響の低減が可能であることを示す．

Performance evaluations of load balancing algorithms for mining generalized association rules

Takahiko SHINTANI† Masaru KITSUREGAWA†

One of the most important problems in data mining is discovery of association rules in large database. We have proposed parallel algorithms and the candidate duplication based load balancing algorithms for mining generalized association rules with classification hierarchy.

In this paper, we present the candidate partition based load balancing algorithm and examine the effectiveness of our algorithms on large scale PC cluster which consists of one hundred PCs interconnected by an ATM switch. Performance evaluations show that our load balancing algorithms are effective for handling skew on large scale parallel system.

1 はじめに

データマイニングで得られる情報の代表的なものに相関ルール (association rule) がある．我々は相関ルールに分類階層を導入した一般化相関ルール (generalized association rule) 抽出の並列処理方式の研究を進めてきた [4]．[4] で提案した並列処理方式では，データベースの分割によるディスク入出力処理の並列化だけでなく，探索候補もノード間に分割することにより，システム全体の主記憶空間を効果的に利用している．更に，データの分類階層を考慮して探索候補をハッシュ分割することにより，通信量の削減を図っている．しかし，データマイニングで扱うデータには偏りが存在するため，単純に探索候補の数をノード間に均等に割り当てることではCPU処理負荷の偏りが生じ，負荷の均衡化を実現することが困難となる．[4] では，数多くのトランザクションに含まれると予想される探索候補を全ノードに複製して個々のノードで処理することにより，処理負荷の偏りを低減する手法を提案してきた．本稿では，データベースの読み出し時に得られるデータの統計情報を利用することにより探索候補の支持度の予測を行い，ノード間の探索候補の検索回数が均等となるように探索候補を分割する手法を検討する．さらに，一般化相関ルール抽出の並列処理方式の負荷分散手法を我々の研究室で構築した大規模PCクラスタ上に実装し，各方式の性能評価を行う．

大規模な並列システムによる相関ルールの抽出処理方式として，[1] が発表されている．[1] では128台のプロセッサによる並列システムを用いているが，データをディスクから読み出さず，主記憶上に小さなサイズのデータを保持し，これを繰り返し用いることで仮想的に大容量のデータを用いた実験を行っている．さらに，[1] で用いられているデータは最大でも50MBytesである．一方，本稿では100台のPCによるPCクラスタにおいて約1GBytesからなる大容量のデータを用いた性能測定を行う．また，データを各ノードの固有なディスクに分割して割り当て，実際にディスクから読み出して実行する．

† 東京大学 生産技術研究所
Institute of Industrial Science, The University of Tokyo

2 分類階層を考慮した相関ルール

データの分類階層構造$\mathcal{T}$を図1に示す木構造とし，その要素をアイテム$\mathcal{I}$と呼び，$\mathcal{T}$の辺はアイテム間の階層を示している．トランザクションデータベースを$\mathcal{D}=\{t_1,t_2,\ldots,t_n\}(t_i\subseteq\mathcal{I})$とし，各要素$t_i$をトランザクションと呼ぶ．長さ$k$のアイテム集合とは$k$個のアイテムの組合せを指す．アイテム集合$X$の支持度$Sup(X)$は$\mathcal{D}$全体に対して$X$を含むトランザクションの割合を表す．また，アイテム集合Xがトランザクションtのアイテムまたはそれらの上位アイテムで構成される場合，tはXを含むと表現する．分類階層を考慮した相関ルール(一般化相関ルール)は$X\Rightarrow Y$で表現され，$X,Y\subset\mathcal{I}$, $X\cap Y=\phi$, YはXの上位アイテムを含まない．一般化相関ルールは支持度，確信度の2つの値によりルールの有意性を示す．ここで，YがXの上位アイテムであるルール$X\Rightarrow ancestor(X)$の確信度は常に100%であり，冗長なルールとなる．一般化相関ルール$X\Rightarrow Y$の支持度$Sup(X\Rightarrow Y)$は$\mathcal{D}$全体に対しXとYを共に含むトランザクションの割合$Sup(X\cup Y)$により，また，確信度$Conf(X\Rightarrow Y)$は$\mathcal{D}$の中でXを含むトランザクションのうち，XとYを共に含むトランザクションの割合$Sup(X\cup Y)/Sup(X)$により定義される．一般化相関ルール抽出問題はユーザにより指定された最小支持度と最小確信度を満足する全てのルールを見出すことに相当する．その処理は，まず最小支持度を満足するアイテム集合(ラージアイテム集合)を生成し，それらを用いて最小確信度を満足するルールを導出する．ラージアイテム集合抽出の逐次処理方式としてCumulateアルゴリズム[3]がある．Cumlateアルゴリズムでは，はじめにデータベースを検索して各アイテムの出現回数(支持回数)を数え上げ，最小支持度を満たすものを取り出し，アイテム数1のラージアイテム集合L_1とする．次に，L_1から2つのアイテムの組合せを作成する．これをアイテム数2の候補アイテム集合C_2と呼ぶ．ここで，階層の上下関係となる候補アイテム集合は除去される．再びデータベースを検索して各C_2の支持回数を数え上げ，ラージアイテム集合L_2を取り出す．以降，アイテム数$k-1$のラージアイテム集合(L_{k-1})からアイテム数kの候補アイテム集合(C_k)を作成し，データベースを検索することによりアイテム数kのラージアイテム集合(L_k)を取り出す処理を繰り返す．このアイテム数kのラージアイテム集合を求める処理をパスkと呼ぶ．

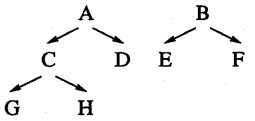

図 1: データの階層構造

3 並列処理方式

本節ではcitesigmodで提案した一般化相関ルール抽出の並列処理方式について簡単に述べる．

3.1 NPGM方式: Non Partitioned Generalized association rule Mining

NPGM方式では候補アイテム集合を全ノードに複製して処理する．パスkの処理は，

1. L_{k-1}を用いてC_kを作成し，分類階層の上下関係となるアイテムの組合せを含む候補を除去する．
2. ローカルディスクからトランザクションを読み出し，各C_kの支持回数を数え上げる．
3. 全トランザクションに対する処理が終了した時点で，各C_kの全ノードでの支持回数の総和を求め，最小支持度を満たすものを取り出しL_kとする．

となる．候補アイテム集合数が多く，単一ノードの主記憶から溢れる場合，主記憶に入り切る大きさに分けて処理を行う．

3.2 H-HPGM(hash)方式: Hierarchical Hash Partitioned Generalized association rule Mining

H-HPGM(hash)方式では候補アイテム集合をハッシュ関数を用いてノード間に分割して割り当て，トランザクションデータを相互通信することにより処理を進める．データの分類階層を考慮して，階層の上下関係となる候補アイテム集合を同一のノードに配置することにより，最下位のアイテムの通信のみで支持度の数え上げが可能となる．また，送信するアイテムの組合せの中で同一トランザクションから作成されたものをまとめ，短いトランザクションの形にして送信することにより，更なる通信量の削減を図っている．各ノードでのパスkでの処理は，

1. NPGM方式と同様にC_kを作成し，アイテムをその最上位アイテムに置き換えたものにハッシュ関数を適用し，ハッシュ値に対応するノードの識別子を求め，自分のノードの識別子と等し

いものを保持する．

2. ローカルディスクから読み出したトランザクション t の各アイテムを最下位のラージアイテムに置き換え，新しいトランザクション t'' を作成する．k 個のアイテムの組合せを作成し，"1" と同様にハッシュ関数を適用して，対応するノードを求め，t'' から各ノードに対応する最下位アイテムを取り出して送信する．
   同時に他ノードから送信された部分トランザクションに対応する候補アイテム集合とその上位候補アイテム集合の支持回数を数え上げる．
3. 全トランザクションの処理が終了した時点で，ノード毎にラージアイテム集合を決定し，他ノードへ放送する．

となる．

4 負荷分散手法

H-HPGM(hash) 方式はノード間の通信負荷を十分に低減することが可能であるが，候補アイテム集合を数え上げる回数 (候補アイテム集合の検索回数) に関しては，ノード間の候補アイテム集合の分割の粒度が粗いため，プロセッサ間で負荷の偏りが生じる．

本節ではノード間の負荷の偏りを低減する手法として，データの統計情報を用いた候補分割による負荷分散手法と [4] で提案した候補複製による負荷分散手法と候補分割による負荷分散手法を組み合わせた手法を述べる．

4.1 統計情報を用いた候補分割負荷分散手法

データの統計情報を用いて候補アイテム集合の支持度を予測し，これを元にして候補アイテム集合の分割を最適化する手法 (統計情報を用いた候補分割負荷分散手法) について検討する．

アイテム集合 X, Y が $Y \subset X$ ならば，支持度は $sup(X) \leq sup(Y)$ である．したがって，X をパス k における候補アイテム集合とすると，X の支持度の上限値 $max\_sup(X)$ は，

$$max\_sup(X) = min\{sup(Y) \mid Y \subset X,\ and\ |Y| = k-1\} \quad (1)$$

となる．例として，アイテム A と B の支持度がそれぞれ 5 と 2 である場合，アイテム集合 $\{A, B\}$ の支持度は最大でも 2 となる．パス k ではパス $k-1$ までに求まったラージアイテム集合の支持度を統計情報として活用することが可能であることから，それぞれの候補アイテム集合の支持度の上限値を求めることができる．この値を評価値としてノード間の検索回数を均等にするように候補アイテム集合の分割の最適化を試みる．

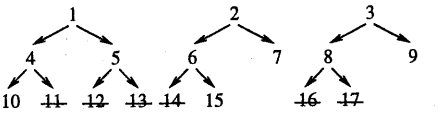

図 2: パス 2 でのデータの階層構造

4.1.1 H-HPGM(stat) 方式:H-HPGM using Statistic information

H-HPGM(hash) 方式での候補アイテム集合の分割単位が木の組合せであることから，木の組合せの評価値をその木に含まれるすべての候補アイテム集合の支持度の予測値の総和とする．木の組合せ X はルートアイテムの組合せで示すことが可能である．X の評価値 $W(X)$ は，

$$W(X) = \sum_{y \in Y} max\_sup(y) \quad (2)$$

となる．ここで，Y は X の下位となる候補アイテム集合 (descendant of X) である．

図 2 に示すデータの階層構造の場合，木の組合せ $\{1,1\}, \{1,2\}$ の評価値は，

$$\begin{aligned}
W(\{1,1\}) &= min\{sup(\{4\}), sup(\{5\})\} \\
&+ min\{sup(\{5\}), sup(\{10\})\} \\
W(\{1,2\}) &= min\{sup(\{1\}), sup(\{2\})\} \\
&+ min\{sup(\{1\}), sup(\{6\})\} \\
&+ min\{sup(\{1\}), sup(\{7\})\} \\
&\cdots \\
&+ min\{sup(\{7\}), sup(\{10\})\} \\
&+ min\{sup(\{10\}), sup(\{15\})\}
\end{aligned}$$

となる．

パス k での評価値の計算および候補アイテム集合の分割の手順を以下に示す．

1. ラージアイテムから最上位アイテム (ルートアイテム) を取り出し，k 個のルートアイテムの組合せ (木の組合せ) を作成する．ここでは，同一のアイテムからなる組合せも作成する．
2. 式 (2) を用いて木の組合せの評価値を計算する．
3. 木の組合せを評価値の高い順にソートする．
4. 既に割り当てられている木の組合せの評価値の和が最低であるノードに木の組合せを割り当てる処理をそれぞれの木の組合せに対して順次行う (図 3)．

foreach size-k combination of root items x **do**
Select the minimum node n whose weight is smallest of all the nodes
$C_k^n := x$ and all descendant candidates of x
(C_k^n means the set of candidates allocated to n-th node)
$CW(n) += $ weight of x
($CW(n)$ means the sum of weight of allocated candidates to n-th node)
end

図 3: 統計情報を用いた候補分割負荷分散手法

4.2 候補複製による負荷分散手法

[4] では，支持度の高くなると予想される探索候補を見つけ出し，それらを NPGM 方式と同様に全ノードに複製して処理し，他の探索候補を H-HPGM(hash) 方式と同様に分割して処理する手法を述べた．本節では，H-HPGM(stat) に候補複製による負荷分散手法を組み合わせた手法を述べる．候補複製による負荷分散手法では複製される探索候補数が増加するに従い，負荷分散の効果が増大する．探索候補の検索回数と探索候補数の分布を共に考慮した探索候補分割のため，式 (2) の評価値の計算式を以下のようにする．

$$W(X) = \frac{\sum_{y \in Y} max\_sup(y)}{\#\ of\ transactions} + \frac{CN(X)}{CN_k - CN^D} \quad (3)$$

ここで，Y は X の下位となる探索候補の中で複製されないもの，$CN(X)$ は X の下位となる探索候補数，CN_kはパス k の探索候補数，CN^Dは複製される探索候補数を示す．式 (3) を用いた候補分割負荷分散手法を H-HPGM(stat+) と呼ぶ．

4.2.1 H-HPGM(stat+) with Tree Grain Duplicate: H-HPGM-TGD(stat+)

H-HPGM-TGD(stat+) はルートアイテムの支持度の高い木の組合せを見つけ出し，それらを全ノードに複製して NPGM と同様に処理する，パス k での探索候補の複製処理を以下に示す．

1. L_{k-1}を用いて探索候補を作成して候補数を数え上げ，主記憶の空きを計算する．
2. ルートアイテムから木の組合せを作成し，それぞれの下位となる探索候補数を数え上げる．
3. 木の組合せをルートアイテムの支持度の和でソートし，主記憶の空きに木の組合せに属する探索候補を複製する．
4. 残りの探索候補を H-HPGM(stat+) と同様に分割する．

4.2.2 H-HPGM(stat+) with Path Grain Duplicate: H-HPGM-PGD(stat+)

H-HPGM-PGD では，支持度の高い最下位アイテムで構成される候補アイテム集合とその全ての上位アイテムを全ノードに複製して処理する．ここで，最下位アイテムとそのすべての上位アイテムの組合せをパス (Path) と呼ぶ．パス k での探索候補の複製処理を以下に示す．

1. H-HPGM-TGD(stat+) と同様
2. 最下位アイテムを支持度の順にソートし，上から順に k 個のアイテムの組合せ (最下位候補アイテム集合) を作成し，主記憶の空きに最下位候補アイテム集合とその全上位候補アイテム集合を複製する．
3. H-HPGM-TGD(stat+) と同様

4.2.3 H-HPGM(stat+) with Fine Grain Duplicate: H-HPGM-FGD(stat+)

H-HPGM-FGD では支持度の高い中間アイテムからなる候補アイテム集合とその上位候補アイテム集合を全ノードに複製する．つまり，H-HPGM-PGD のパスから部分的に候補アイテム集合を選び出すことになるため，不必要な候補アイテム集合の複製を回避し，更に細粒度の負荷制御が可能となる．パス k での探索候補の複製処理を以下に示す．

1. H-HPGM-TGD(stat+) と同様
2. ラージアイテムを支持度の順にソートし，上から順に k 個のアイテムの組合せを作成し，主記憶の空きに複製する．
3. H-HPGM-TGD(stat+) と同様

5 性能評価

本稿で述べた並列処理方式および負荷分散手法を我々の研究室で構築した大規模 PC クラスタ [5] に実装し，性能評価を行った．この大規模 PC クラスタは 100 台のパーソナルコンピュータを 155Mbps の ATM ネットワークおよび 10Mbps のイーサネットの 2 系統のネットワークで接続したシステムであり，クラスタの各ノードには，200MHz の Pentium Pro PC が用いられている．PC の各構成要素を表 1 に示す．また，全体のシステムを図 4 に示す．

5.1 並列処理方式の実行

トランザクションデータは [2] で公開されているデータ作成ツールを用いた．これは，[3] で述べられた手順を元に小売業における購買トランザクション

表 1: PC クラスタの構成要素

CPU	Intel Pentium Pro 200MHz
Main memory	64MBytes
Disk drive	Seagate Barracuda 4.3GB
OS	Solaris2.5.1 for x86
ATM NIC	Interphase 5515 PCI ATM Adapter

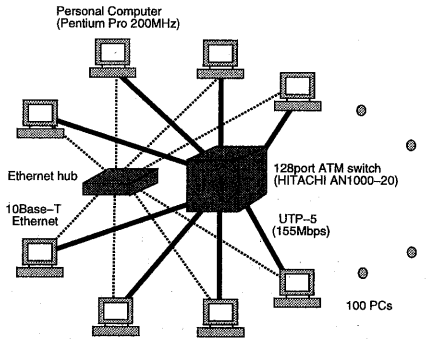

図 4: PC クラスタの構成図

を模倣して作成するものである．データセット作成に用いたパラメータを表 2 に示す．このデータセットのファイルサイズは約 1GB であり，各ノードに均等に分割して割り当ててある．

5.2 実験結果

図 5 に最小支持度を変化させた場合のパス 2 での処理時間を示す．各手法との比較のため，ノード間の負荷バランスが最も理想的となる場合の結果を Flat とした．Flat は他の手法で求めた結果から，木の組合せの評価値を設定したものである．また，本性能測定では探索候補数が最も多く，処理負荷の高いパス 2 の処理についてのみ示す．最小支持度が小さい場合，探索候補数が増大するため，候補の分割を行わない NPGM 方式の処理時間が長くなる．H-HPGM(stat)

表 2: データセットのパラメータ

Parameter	
Number of transactions	20,000,000 (約 1GB)
Average size of the transactions	5
Average size of the maximal potentially large itemsets	5
Number of maximal potentially large itemsets	10,000
Number of items	50,000
Number of roots	100
Number of levels	5–6
Fanout	5

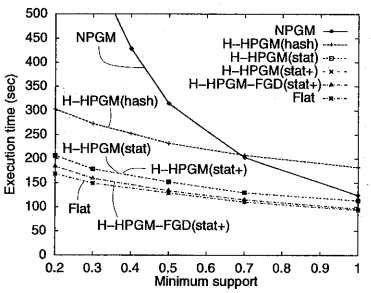

図 5: 最小支持度を変化させた場合の処理時間

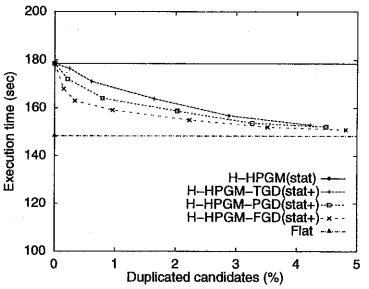

図 6: 複製サイズを変化させた場合の処理時間

方式と H-HPGM(stat+) 方式の処理性能はほぼ等しくなり，NPGM 方式および H-HPGM(hash) 方式よりも処理時間が短くなるが，H-HPGM-FGD(stat) よりも処理性能が劣る．H-HPGM(stat) 方式と H-HPGM(stat+) 方式は前パスでの支持度により求めた候補アイテム集合の支持度の最大値をもとにして探索候補の分割単位の評価値を設定するため，誤差が大きい．したがって，十分な負荷の均衡化を実現することが出来ない．H-HPGM-FGD(stat+) は探索候補の複製による負荷の均衡化も行うため，最も Flat に近い性能を実現することが出来る．

図 6 に複製する探索候補数を変化させた場合のパス 2 での処理時間，図 7 に複製サイズを変化させた場合のパス 2 での探索候補の検索回数の分布，図 8 にパス 2 での全探索候補の検索回数において複製された探索候補による検索回数の占める割合，図 9 に複製サイズを変化させた場合のパス 2 での各ノードの受信量の平均値を示す．これらの結果から，複製される探索候補数が増大するに従い，処理負荷の偏りと通信量が減少し，処理性能が向上することが分かる．図 8 から，複製された探索候補による検索回数が多いことが分かる．特に，H-HPGM-FGD(stat+) 方式では全探索候補の 1%の複製により，全検索回数の 25%を制御することが可能であることが分かる．

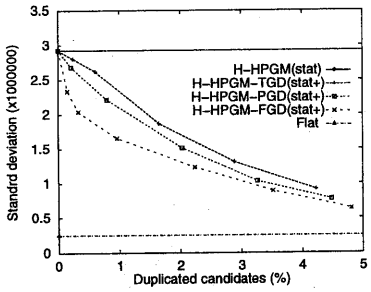

図 7: CPU 処理の分布

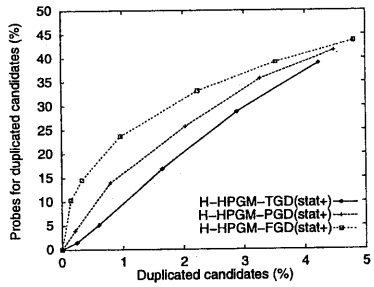

図 8: 複製候補の制御する CPU 処理負荷

図 10 にノード台数を 16 台から 100 台まで変化させた場合のパス 2 での台数効果を示す．結果から，H-HPGM-FGD(stat+) 方式は Flat に近い性能を実現していることが分かる．大規模な並列システムでは，H-HPGM(stat) 方式のように前パスの支持度による予測を用いた探索候補の分割による複製のみでは十分な性能を実現することが出来ない．H-HPGM-FGD(stat+) 方式のように探索候補の分割と複製を共に考慮することにより，良い性能向上を実現することが可能となる．

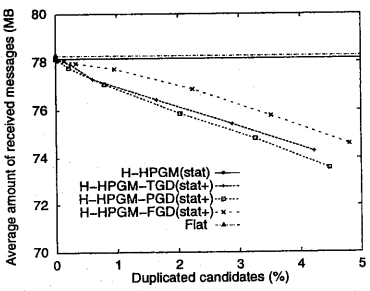

図 9: 受信量の平均値

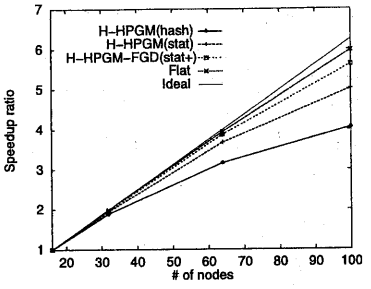

図 10: 台数効果

6 まとめ

本稿では一般化相関ルールマイニングの並列処理方式の負荷分散手法について，探索候補の分割による方式と探索候補の複製による手法を検討した．相関ルールマイニングではアイテム数 k の探索候補を調べる処理において，アイテム数 $k-1$ の支持度の情報を利用することが可能であり，これを用いた探索候補の支持度の予測による負荷分散手法として H-HPGM(stat) を提案した．H-HPGM(stat) 方式はハッシュ関数を用いる手法よりも良い性能を実現することが出来るが，前パスでの統計情報による予測では誤差が大きいため十分ではない．H-HPGM-FGD(stat+) 方式のように探索候補の分割と複製を共に考慮した負荷分散手法が必要となる．さらに，我々の研究室で構築した大規模 PC クラスタ上での性能評価により，大規模な並列システムでも H-HPGM-FGD(stat+) により良い性能向上を実現することが可能であることを示した．

参考文献

[1] E.-H.Han, G.Karypis, and Vipin Kumar. Scalable parallel data mining for association rules. In *Proc. of ACM SIGMOD*, pages 277–288, 1997.

[2] IBM. http://www.almaden.ibm.com/cs/quest/syndata.html. In *Quest Data Mining Project.*

[3] R.Srikant and R.Agrawal. Mining genrealized association rules. In *Proc. of VLDB*, pages 407–419, 1995.

[4] T.Shintani and M.Kitsuregawa. Parallel algorithms for mining generalized association rules with classification hierarchy. In *Proc. of ACM SOGMOD*, pages 25–36, June 1998.

[5] T.Tamura, M.Oguchi, and M.Kitsuregawa. Parallel database processing on a 100 node pc cluster: Cases for decision support query processing and data mining. In *Proc. of Supercomputing 97::High Performance Networking and Computing*, 1997.