3N-3

分散関係データベースからの飽和パターンマイニングにおけるマージ演算の効率化

谷本　翔一 †　神谷　洋平 ††　世木　博久 †

† 名古屋工業大学大学院 工学研究科 情報工学専攻　†† 名古屋工業大学 情報工学科

1　はじめに

関係データマイニング (MRDM)[2] は効率性が課題である [1]. したがって, 分散されたデータベース (DB) に対してマイニングを行うことは有益である. そのため, アイテム集合マイニングにおける分散化手法を MRDM に適用し, 分散環境基盤 Hadoop[5] を用いて実験を行った [4]. 結果, 分散化によって必要となった処理にかかる時間が, 分散 DB の数が増えると全体の処理時間に影響を与えることが分かった. 本編では, 分散化による処理を効率化する手法を提案, 実験し, 処理時間が短縮されたことを確認する.

2　関係データマイニング

MRDM とは複数の関係表からなる DB から, 述語論理式で表現されるパターン (連言) を発見する手法である. 例えば, 図 1 の DB_{ex} は 5 つの関係表を持ち, 連言　$customer(X)$,$buys(X,Y)$,$parent(X,Z)$,$male(Z)$ がマイニングされる. この連言は「顧客 X は Y を買い, X は子供 Z を持ち, Z は男性である」を意味する.

データマイニングにおいてこのようなパターンの多くは冗長であるため, 同一の出現集合を持つパターンから代表たる, すなわち最大長のパターン (飽和パターン) のみを求める方が効率が良い.

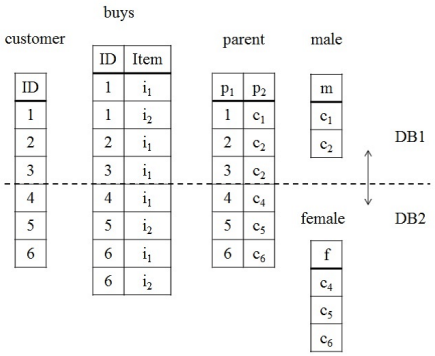

図 1: 顧客に関するデータベースの例 DB_{ex}

3　マージ演算

全体 DB が 2 つの部分 DB (DB_1, DB_2) に分割されるとする. この時, 全体 DB の飽和パターン集合 $\mathcal{C}$ は, 各 DB_i の飽和パターン集合 $\mathcal{C}_i$ から次式で定義されるマージ演算 $\oplus$ により求められる [3].

$$\mathcal{C} = \mathcal{C}_1 \oplus \mathcal{C}_2$$
$$= (\mathcal{C}_1 \cup \mathcal{C}_2) \cup \{C_1 \cap C_2 \mid (C_1, C_2) \in (\mathcal{C}_1 \times \mathcal{C}_2)\}$$

上式は分割 DB の数が n (≥ 2) の場合にも成り立つ.

マージ演算には共通部分演算が必要となるが, 演算回数が飽和パターンの数に依存し, 処理時間に負担となる. また, マージ演算の回数は最低 $n-1$ 回必要となる. そのため, n が多くなればなるほどマージ演算に処理時間がかかるようになり, マイニングの分散化による効果が損なわれてしまう. したがって, マージフェーズに対して, その計算コストを小さくする手法が必要となる.

その手法として, マージ演算の分散実行を行った. 図 2 の 0^{th} $step$ における四角で囲まれた 2 つのマージ演算は互いに独立であるため, これを分散実行することで, 本来マージ 3 回分の時間を 2 回分に抑えることができる. その結果, n が多くなるごとに効率が良くなることが判明したが, 飽和パターン数が増え, 処理時間のかかる, 後半の $step$ のマージ演算が効率化されていないため, 予想より効果が上がらなかった.

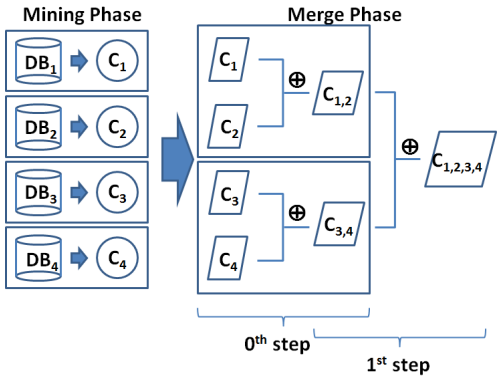

図 2: 分散 MRDM(DB は分散データベース, C は飽和パターン集合を示す)

On Closed Pattern Mining from Distributed Relational Databases

† Shoich Tanimoto (cjk17569@stn.nitech.ac.jp)
† Hirohisa Seki (seki@nitech.ac.jp)
†† Youhei Kamiya(cin15038@stn.nitech.ac.jp)
Dept. of Computer Science, Nagoya Inst. of Technology (†)
Showa-ku, Nagoya, 466-8555 Japan

4　効率化方式

後半の *step* におけるマージ演算を効率的に処理する方法として, 今までタスクの最小単位としていたマージ演算の分散実行が挙げられる. マージ演算の分割とは $\oplus$ が分配則を満たすことから, 次式が成り立つ:

$$\begin{aligned}\mathcal{C}_1 \oplus \mathcal{C}_2 &= \mathcal{C}_1 \oplus (\mathcal{C}_{2-1} \cup \mathcal{C}_{2-2}) \quad //\mathcal{C}_2 = \mathcal{C}_{2-1} \cup \mathcal{C}_{2-2} \\ &= (\mathcal{C}_1 \oplus \mathcal{C}_{2-1}) \cup (\mathcal{C}_1 \oplus \mathcal{C}_{2-2}).\end{aligned}$$

このように飽和パターン集合 $\mathcal{C}_i$ を分割することでマージ演算をさらに分散実行する, パターン集合分割方式を提案する (図 3). マージフェーズではマージ演算が行われるごとに必要なマージ演算の数が減少し, 必要なマシンの数も減少する. そこで, タスクを割り振られなくなったマシンをマージ演算の分散実行に利用する.

例えば図 2 のマイニングフェーズではマシンを 4 つ使用しているのに対し, マージフェーズでは 0^{th} *step* で 2 つ, 1^{st} *step* では 1 つである. したがって, 0^{th} *step* では 1 つのマージ演算に対して 2 つ, 1^{st} *step* では 4 つのマシンを使用することが可能なため, 処理時間を抑えることが期待できる. また, マシン数が n と同数ある場合には, *step* が進むごとにマージ演算を多くのマシンで分散実行できることから, より処理時間の必要であった後半の *step* のマージ演算を効率化する手法として適していることがわかる.

この手法を採る場合には分散処理によって得られた結果の重複除去が必要である. この処理はマージ演算における和集合演算と同様の処理を行えばよい.

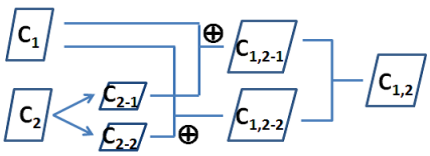

図 3: パターン集合分割方式によるマージ演算

5　実験

パターン集合分割方式について実装し, 非分散版, マージフェーズの分散化を行ったマージフェーズ分散版との全てのマージ演算に要した時間について比較実験を行った. DB の分割数 (飽和パターン集合の数) は 4, 8, 16 の 3 種とし, マシンの台数は n 台あると仮定. 入力として, 突然変異 DB および, 英文コーパスのマイニング結果を用いた.

図 4 から, パターン分割方式は分散を行わなかったマージと比べ, n が大きくなるほど計算時間が短くなり, 突然変異 DB, $n = 16$ の時に最大で約 80%の計算時間が短縮された. マージフェーズ分散と比較しても, 最大 58%計算時間が短縮される結果となった.

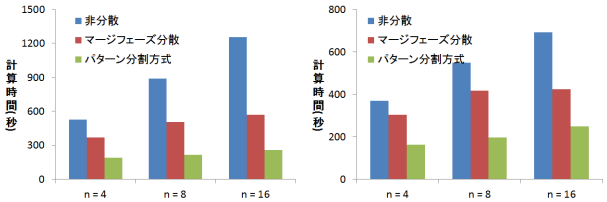

図 4: 各手法毎のマージ演算に要した計算時間 (左: 突然変異 DB, 右: 英文コーパス)

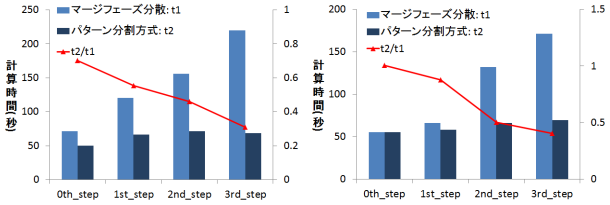

図 5: $n = 16$ での *step* 毎のマージ演算の計算時間比較

また, 図 5 より, s が大きくなるほどマージ演算の計算時間を短縮することができ, 突然変異 DB, $s = 4$ の時に最大で約 70%短縮という結果になった. しかし, 上条件では 1 つのマージ演算を 16 台で分散実行しているため, 理想では 1/16 の時間で処理が完了することを考えると, 思うような効果を上げられなかった.

6　まとめ

本研究により, パターン分割方式を用いてマージ演算を分散して処理することは, パターン数増加によるマージ演算の処理時間増やマシンの余剰という問題に対して効果があることが分かった. 今後の課題として, マージ演算の最適化や演算に適したデータ構造の構築, パターン集合分割方式における効果的なパターン分割といった他の効率化手法の提案, 実装が挙げられる.

参考文献

[1] Blockeel, H., Sebag, M. Scalability and efficiency in multi-relational data mining. SIGKDD, Vol.4, Issue 2, pp.1-14. 2003.

[2] Dzeroski, S., Lavrač, N. Relational Data Mining. Springer-Verlag. Inc. 2001.

[3] Lucchese, C., Orlando, S., and Pergo, R. Distributed Mining of Frequent Closed Itemsets: Some Preliminary Results. International Workshop on High Performance and Distributed Mining. 2005.

[4] Seki, H., Tanimoto, S. Distributed Closed Pattern Mining in Multi-Relational Data based on Iceberg Query Lattices: Some Preliminary Results. CLA, pp.115-126, 2012.

[5] White, T. (訳:玉川竜司, 兼田聖士), Hadoop. 2010.