

検索回数を考慮した相関ルール抽出 データマイニングアルゴリズム*

張 信華 渋沢 進 岡本 秀輔 鈴木 均†

茨城大学工学部‡

1 はじめに

大規模なデータから価値ある情報を取り出すデータ処理技術として、データマイニング技術が開発されてきた。データベースの個々の要素をアイテムとよび、アイテムの集合をトランザクションとよび、データベースはトランザクションの集合よりなる。データマイニングの相関ルールとは、複数のアイテムまたはアイテム集合間の相関関係を表すものであり、アイテム集合 X 、 Y ($X \cap Y = \phi$) に対して、 X が成り立つとき Y が起こることを表す規則 $X \Rightarrow Y$ である。相関ルールを求める方法として、Apriori アルゴリズムが良く知られているが、大きな検索コストがかかる [1],[2]。本研究では、検索回数を減少させることで、処理時間を減少させることを目的とした2つのアルゴリズムを考えて、これらの実行時間を測定した。

2 相関ルールのマイニング

2.1 相関ルール

トランザクションデータベース D から X と Y が一緒に出現する割合を、 $X \Rightarrow Y$ のサポート値とよび、 $\text{support}(X \Rightarrow Y)$ で表す。 D において、 X を含むトランザクションのうち Y を含むものの割合を $X \Rightarrow Y$ のコンフィデンス値とよび、 $\text{confidence}(X \Rightarrow Y)$ で表す。 $X \Rightarrow Y$ のコンフィデンス値は、 $\text{confidence}(X \Rightarrow Y) = \frac{\text{support}(X \cup Y)}{\text{support}(X)}$ である。相関ルールを発見する問題は、ユーザによって指定されたサポート値の最小値 (min_s) とコンフィデンス値の最小値 (min_c) を満たすすべての相関ルールを見つけ出すことに相当し、次の2段階の処理に分割できる。

1. アイテムの集合から min_s を満たすすべてのアイテム集合を見つけ出す。これらのアイテム集合を頻出アイテムセットとよび、要素数 k の頻出アイテム

集合を L_k で表す。このとき、 D, min_s を入力し、 $\{L_1, L_2, \dots, L_j \mid (L_j \supseteq I \text{ かつ } \text{support}(l_j) \geq \text{min}_s)\}$ を出力する。

2. 第1段階で求めた L_k から、 min_c を満たす相関ルールを導出する。

上の第1段階の処理は、全部のアイテムとアイテムセットの出現回数を求めるため莫大な時間がかかる。第1段階を逐次的に処理する方法として、Apriori アルゴリズムが良く知られている [1][2]。

2.2 Apriori アルゴリズム

長さ k の候補アイテムセットを C_k とするとき、Apriori アルゴリズムは次のような処理を行う。

1. トランザクションデータベースを検索して、各 C_k のサポート値を調べる。
2. C_k 中の最小サポート値を満たす頻出アイテムセット L_k から C_{k+1} を生成する。

C_k がなくなるまで上の処理を繰り返す。

Apriori アルゴリズムは候補アイテムセットを生成することで、最小サポート値を満たさないアイテムセットを削除していくため、出現回数を求めようとするアイテムセットの数はかなり減らすことができる。しかし、各アイテムセットのサポート値を求めるために、毎回トランザクションデータベース全体を検索するので、サポート値の数え上げを行う候補アイテムセットが多いと、大きな検索コストがかかる。

3 提案するアルゴリズム

本研究では、元のデータベースから、頻出アイテムセット L_k とトランザクション集合の対応を表す逆変換テーブルを作成する。各アイテムセットの出現回数を求める際に、逆変換テーブルから条件を満たすトランザクションを調べれば、トランザクションデータベース全体を検索する必要がなく、検索回数を減少させることができる。要素数 k に対する逆変換テーブル RT_k は、 L_k からそれらを含むトランザクションを検索できるようにしたテーブルである。

*Data mining algorithms of association rules based on the number of search operations

†Zhang Xinhua, Susumu Shibusawa, Shusuke Okamoto, Hitoshi Suzuki

‡Ibaraki University, Hitachi, Ibaraki 316-8511, Japan

MS(Min Support) アルゴリズム

1. L_k とそれら L_k を含むトランザクションからなる逆変換テーブル RT_k を元のデータベースから作成する。
2. 組合せようとする1つのアイテムのみ異なる2つの L_k のうち、サポート値が小さい L_k を含むトランザクションを RT_k から調べ、また元のデータテーブルで、 RT_k での条件を満たしているトランザクションを探し出し、その中から異なるアイテムを調べ、サポート値を求める。
3. その中から最小サポート値を満たすアイテムセットを取り出し、 L_{k+1} とする。

L_{k+1} がなくなるまで1~3の処理を繰り返す。

L_k とトランザクションの表から直接候補アイテムセットを求めながら、次の頻出アイテムセットを求めることによって、さらに検索回数を減少させることができ、これは、次のような手順で実現できる。

PS(Product Set) アルゴリズム

1. L_k とそれら L_k を含むトランザクションからなる RT_k を元のデータベースから作成する。
2. RT_k から、組合せようとする2つの L_k に対応するトランザクションを調べ、その積集合を求め、新しい集合の長さから、サポート値を求める。
3. 最小サポート値を満たすアイテムセットを取り出し、 L_{k+1} とする。

L_{k+1} がなくなるまで1~3の処理を繰り返す。

トランザクションデータベースのトランザクション集合の大きさを $|D|$ 、要素数 k の候補アイテムセットの要素数を $|C_k|$ とすると、Apriori の検索回数は $O(|D| \sum_{i=1}^k |C_i|)$ である。MS アルゴリズムは D 全体を検索する必要がないので、Apriori アルゴリズムよりは検索回数が小さくなり、PS アルゴリズムの検索回数は $O(|D|)$ である。

4 アルゴリズムのシミュレーション

FreeBSD4.8 の動作している PC 上に、データベースサーバ PostgreSQL7.3.4 をインストールした。この上で、SQL 埋め込み関数と C 言語を用いて Apriori, MS, PS アルゴリズムのプログラムを作成した。用いた PC の CPU 速度は 3.2GHz、主メモリは 1Gbyte、HDD 容量は 120GByte である。データは、幾つかの特定のアイテムセットが、前もって決めた出現率で出現するようにランダムに生成した。これにより幾つかのアイテ

ムセットのサポート値が、必ず指定した割合以上の出現率をもつことができる。

アイテム数、トランザクション数、最小サポート値を変化させながら、各アルゴリズムを実行したときの実行時間の例を図1に示す。図1は、トランザクション数が 6×10^4 、アイテム数が 1000 の場合であり、横軸は最小サポート値、縦軸は実行時間を表す。

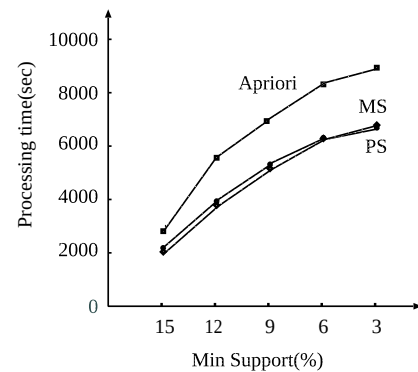


図 1: 実行時間の例

図1より、MS と PS アルゴリズムの実行時間はともに、Apriori アルゴリズムより小さい。図1の場合では、MS アルゴリズムと PS アルゴリズムがほぼ同じ性能を示しているが、最小サポート値が小さくなると PS の方が良い性能を示している。

5 まとめ

今回は、検索回数を減少させ、実行時間を小さくするための2つのアルゴリズムを提案した。また、よく知られている相関ルールアルゴリズム Apriori と提案したアルゴリズムを実装し、実行時間を測定してみた。その結果、MS と PS アルゴリズムはともに Apriori アルゴリズムより実行時間が小さくなり、頻出アイテムセット数が少ない場合 MS アルゴリズムが良い性能を示し、頻出アイテムセット数が多い場合 PS アルゴリズムが良い性能を示している。

謝辞 本研究についてご討論いただいた本学科鎌田賢助教授、米倉達広助教授、大瀧保広講師に感謝致します。

参考文献

- [1] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," *Proc. of the 20th Int'l Conf. on VLDB*, pp.487-499, Sept. 1994.
- [2] J. Han and M. Kamber, *Data Mining, Concepts and Techniques*, Morgan Kaufmann, 2001.