

# Groceries に関する R 言語を使用した頻出パターン抽出及び相関ルール分析

文理学部情報科学科

5419045 高林 秀

2021 年 7 月 24 日

## 概要

本稿では、今年度データ科学 2 で学習した「頻出パターン抽出」及び「相関ルール分析」の手法を使用して、R 言語のライブラリである `arules` に付属しているデータ `Groceries` を対象とした頻出パターン抽出、相関ルール分析を行うものである。

## 1 目的

本稿では実際に、R 言語を使用しライブラリ `arules` 付属のデータである `Groceries` の頻出パターン抽出、相関ルール分析を行うことで、本年度データ科学 2 で学習した頻出パターン、相関ルール分析の手法への理解を深め、その定着を図ることを目的とする。また、1 年次に学習した `latex` を用いた PDF 作成の復習も兼ねるものである。

## 2 理論説明

今回の実験で用いた、計算理論をそれぞれ説明する。

### 2.1 バスケット分析

初めに、頻出パターン抽出、相関ルール分析を説明する前に「バスケット分析」について説明する。

バスケット分析とは、データマイニングにおける代表的な手法の 1 つで、「顧客の購買記録をデータ化し分析を行うことで、顧客に共通するルールや傾向を導く」データ分析のことである。すなわち「一緒に買われやすい商品の組み合わせを見つける」ということである。

顧客の買い物データを分析しその結果を企業の販促活動などのマーケティングに関わる施策に適用するのが目的である。なお、バスケット分析はアソシエーション分析<sup>\*1</sup>の一つとされ、マーケットバスケット分析とも呼ばれる。

---

<sup>\*1</sup> データマイニングにおけるデータ間の関連性を見つける手法のこと。「もし A ならば B である」といった法則を見つけ出し、主に購買記録などから顧客の購買行動の関連性を見つけ出すのに利用される。

## 2.2 頻出パターン・相関ルール分析の概要

身の回りで頻出パターン抽出と相関ルール分析が使われている代表例としては、先に述べたバスケット分析をはじめ、オンラインショッピングサイト等のレコメンドシステム等が挙げられるだろう。頻出パターン抽出と相関ルール分析はこれらのシステムの基本的な原理である。この章では、頻出パターン抽出と相関ルール分析とはなにか概要を説明する。

### 2.2.1 頻出パターン

まず、頻出パターン (頻出アイテム集合) とはなにか説明する。頻出パターンとは「データベース中に高頻度で現れる組み合わせ、集合のこと」であり、頻出パターン抽出 (頻出パターンマイニング) とはその集合を発見するための手法である。またこの集合のことを頻出アイテム集合と呼ぶ。頻出アイテム集合か、そうでないかを判断するための基準として後述する支持度 (同時確率) と呼ばれる数値を計算し、その数値が、あらかじめ設定した閾値を超えるかどうかで判定する。頻出パターン抽出は「どの商品と一緒に購入されているか」を見るので、得られた結果は、商品の陳列場所の改善や、販促キャンペーン、店舗レイアウト等を考える際に利用することができる。

### 2.2.2 相関ルール

次に、相関ルールとはなにか説明する。相関ルールとは「頻出パターン間の関係性」のことで、相関ルール分析はこの関係性すなわちルールを見つける目的で行われる。例えば、「あるアイテム集合  $I1$  が生起するとき、別のアイテム集合である  $I2$  も同時に生起する」といったようなものが相関ルールとなる。このとき、記号で「 $\{I1\} \Rightarrow \{I2\}$ 」といった形で記述する。導いた相関ルールを評価する評価基準として、後述する確信度と呼ばれるものが存在する。具体的な計算法は後述するが、確信度とは一言で言えば「ルールの強さ」を示す指標で、左辺のアイテム集合が生起したときの右辺のアイテム集合の生起確率である。加えて、支持度も利用される。相関ルールにおける評価指標としての支持度は「ルールの汎用性」を示すものとして利用される。

相関ルール抽出問題とは、あらかじめ設定する「最小支持度」「最小確信度」を閾値として、この閾値を超える相関ルールをデータベース上から見つけることを目的とした問題である。

### 2.2.3 トランザクションデータベース

頻出パターン・相関ルール分析は後述するように、形式的な定義のもとで、入力を受けその出力として頻出パターン・相関ルールを返す。このとき、入力として「トランザクションデータベース」が与えられる。

トランザクション (英名: transaction) とは、商取引、議事録、売買等の意味があり、情報処理用語としては一連の処理をひとつにまとめたものという意味をもつ。トランザクションデータベースとは、データの更新処理を一つにまとめているようなデータベースのことである。

## 2.3 計算法

この章では、実際に頻出パターンや相関ルールがどのように計算されているのかについて説明する。その前に、概要の部分で登場した「支持度」と「確信度 (信頼度)」について説明する。以下の説明で使用する数式記号について、

$D = t_1, t_2, t_3, \dots, t_n : n$  個のトランザクションを含むデータベース  $D$

$I = \cup_{t_i \in D} t_i : \text{全アイテムの集合}$

$t_i \subseteq I : i$  番目のトランザクション

$\min\_sup(0 < \min\_sup \leq 1) : \text{最小支持度}$

$\min\_conf(0 \leq \min\_conf \leq 1) : \text{最小確信度}$

のように定義する。

### 2.3.1 支持度 (support)

支持度とは「ルールの汎用性 (一般性) の尺度」であり、集合  $I1, I2$  を例にしたベン図で示すと以下のようなになる。これは、同時確率とみなすことができる。つまり、あるパターン  $X$  の支持度とは  $X$  中のアイテムが同時に出現する確率ということができる。これを式で示すと以下のようなになる。

※  $|a|$  : 集合  $a$  の要素数

$$sup_D(X) = \frac{|t \in D | X \subseteq t|}{|D|}$$

- $t \in D$ : アイテム集合  $X$  を含むデータベース中のトランザクション

つまり、アイテム集合  $X$  の支持度は、 $X$  を含むトランザクションの割合、すなわち  $X$  中のアイテムがすべて出現するときの確率という意味で、その値は「 $X$  を含むデータベース上のトランザクション」を「データベース全体のトランザクション数」で除算した値である。

支持度が低いとは、そのパターンがごく少数の事例にのみ関係するパターンであり、いかに特徴的でもそのデータベースの一般的な傾向とはみなされていない、ということになる。反対に、支持度が高いとは、そのパターンがデータベース内で一般的であるということを意味している。

### 2.3.2 確信度 (confidence)

確信度とは、「そのルールの確からしさの尺度」であり、データマイニングにおける相関ルールの重要度を示す指標である。別名、信頼度とも呼ばれる。

前章の部分でも述べたが、あるアイテム集合  $I1$  が生起するとき同時に  $I2$  も生起するという現象、ルールは  $I1 \Rightarrow I2$  で表記される。確信度とは、 $I1 \Rightarrow I2$  のルールの強さを示す指標と言える。 $I1 \Rightarrow I2$  の確信度を  $conf_D(I1, I2)$  と示すと、確信度は以下のように計算される。

$$\begin{aligned} conf_D(I1, I2) &= \frac{|\{t \in D | I1 \subseteq t, I2 \subseteq t\}|}{|\{t \in D | I1 \subseteq t\}|} \\ &= \frac{|\{t \in D | (I1 \cup I2) \subseteq t\}|}{|\{t \in D | I1 \subseteq t\}|} \end{aligned}$$

このとき、分母の式  $|\{t \in D | I1 \subseteq t\}|$  は  $I1$  の出現回数を示している。また分子の式  $|\{t \in D | (I1 \cup I2) \subseteq t\}|$  は、 $I1, I2$  の同時出現回数を示している。つまり確信度とは、 $\frac{I1 \text{ の出現回数}}{I1 \text{ と } I2 \text{ が同時に出現する回数}}$  の値ということになり、これは条件付き確率と同じになる。

確信度が低いときは、そのルールが不正確であることを示している。反対に確信度が高いときは、そのルールが正確なルールであることを示していることになる。

### 2.3.3 頻出パターン抽出の計算法

頻出パターン抽出の際は、入力としてトランザクションデータベースを受け取り、すべての頻出アイテム集合を出力する。このとき、頻出パターン  $F$  は次の式で示すことができる。

$$F = \{X | X \subseteq I, X \neq \phi, \text{sup}_D(X) \geq \text{min\_sup}\}$$

■頻出パターン抽出の計算量 上記式で示すとおり、頻出パターンに選ばれた集合  $X$  の支持度は、予め定めた支持度の下限値すなわち最小支持度以上である必要がある。ということは、頻出パターンを見つけるには単純に、すべてのアイテム集合  $I$  に属する、すべての部分集合の支持度を1つずつ計算していき、その値が最小支持度を超えるかどうか判定すれば良いことになる。しかし、現実の場合ではそうもいかない。ご存知の通り、我々が普段使うコンピュータの計算資源は無限ではなく有限である。したがって、あまりにも計算量が大きすぎる問題に関してはそもそも計算リソースが足らず、答えを求めることができない。

現実の現場（小売店やその他店舗）等で使用されるデータベースは膨大な数のデータを扱うことがほとんどである。そうなれば当然すべてのアイテム集合  $I$  の数も膨大である。以下に、すべてのアイテム集合  $I$  の数を  $|I|$  としたときの冪集合\*2の個数を示す。

$$\text{冪集合の個数 } N = 2^{|I|} - 1$$

表1  $|I|$  の値による冪集合の個数

$ I $	冪集合の個数
$ I  = 10$	1024
$ I  = 16$	65536
$ I  = 50$	約 1.1 京 ※ 1 京 = $10^{16}$
$ I  = 100$	約 1267 穰 ※ 1 穰 = $10^{28}$
$ I  = 200$	約 1.6 那由他 ※ 1 那由他 = $10^{60}$ , 一説では $10^{72}$

上記の表からも分かるとおり、現実の現場ではすべての冪集合を求めることは不可能に近い。したがってより効率的に頻出パターンを計算するアルゴリズム・手法が必要となる。そのような代表例として、後述する「バックトラック法」や「アプリアリアルゴリズム」が存在する。

---

\*2 ある集合の部分集合全体の集合

## 2.4 パターン空間について

## 2.5 (補足) 深さ優先探索・幅優先探索

## 2.6 バックトラック法

## 2.7 アプリオリアルゴリズム

## 2.8 相関ルール抽出の計算法

## 2.9 相関ルール分析の評価基準

# 3 計算機実験

## 3.1 実験準備

### 3.1.1 実験環境

今回の実験は仮想マシン上で R 言語を起動し行った。下記に実験時の環境を示す。

- ホスト OS : Window10 Home Ver.20H2
- 仮想 OS : Ubuntu 20.04.2 LTS
- CPU : Intel(R)Core(TM)i7-9700K @ 3.6GHz
- GPU : Nvidia Geforce RTX2070 OC @ 8GB
- ホスト RAM : 16GB
- 仮想 RAM : 4GB

### 3.1.2 実験データ

### 3.1.3 R 言語での頻出パターン・相関ルール分析の手法

## 3.2 実験結果

## 3.3 結果の説明

# 4 考察

# 5 まとめ

## 参考文献

- [1] 集合について : <https://mathlandscape.com/power-set/#:~:text=%E4%B8%80%E8%A8%80%E3%81%A7%E3%81%84%E3%81%86%E3%81%A8,%E3%81%AE%E9%9B%86%E5%90%88%E3%82%92%E6%8C%87%E3%81%97%E3%81%BE%E3%81%99%E3%80%82>