2A3-4

クラス付きアイテム集合からの頻出パターンの発見

Discovery of Frequent Patterns from Item Sets with their Classes

櫻井 茂明[*1]
Shigeaki Sakurai

[*1](株) 東芝 研究開発センター
Corporate Research & Development Center, Toshiba Corporation

This paper proposes a method that reflects the difference of conditions in the data collection. Here, the conditions are the selection of goods, the release time of goods, the access path of goods, and so on. The method regards the difference of conditions as classes. It efficiently discovers frequent patterns from item sets with their classes by using two kinds of redefined supports. Also, this paper applies the method to 3 data sets in UCI Machine Learning Repository and verifies the effect of the method by evaluating the difference of discovered patterns.

1. はじめに

コンピュータ及びネットワーク環境が発展した現代においては、多数のデータを簡便に収集・蓄積することが可能になっている。これらデータの中には、人の意思決定を支援可能な有用な知識が埋め込まれており、それら知識を発見するデータマイニング研究が、大きな研究分野を形成している。

これら分野のうち、同時に購入される傾向にある商品 (アイテム) を分析する、バスケット分析から発展した頻出パターンの発見法は、高速にすべての頻出パターンを発見することを可能としている [Han et al. 00]。また、バスケット分析で本来扱っていた購入・未購入に対応する 2 値のアイテムではなく、表構造データからの頻出パターンの発見法及び、表構造データに内在する欠損値を考慮したパターンの発見法も提案されている [Calders et al. 07][Sakurai and Mori 10]。

このような頻出パターンの発見問題は、現在も多様な進化を遂げており、精力的に研究活動が行われている。本論文では、データ収集時における収集条件の違いに着目し、このような収集条件の違いをクラスとして捉えることにする。これにより、パターン発見時においてその違いを区別できるようにし、このクラスを考慮した効率的なパターンの発見法を提案する。最終的には、UCI Machine Learning Repository(http://archive.ics.uci.edu/ml/) にて提供されている実データに適用し、提案法の効果を検証する。

2. 頻出パターンの発見法

2.1 クラス付きアイテム集合

本論文で対象とするクラス付きアイテム集合 t を、式 (1) により定義する。式 (1) においては、I がアイテムの全体集合、C がクラスの全体集合、n がアイテム集合に含まれるアイテムの個数、C_{it_i} がアイテム it_i が所属するクラスの集合 (以下、アイテムクラス) であるとする。本定義から分かるように、クラス付きアイテム集合は、重複を持たない n 個のアイテムとひとつのクラスから構成されている。ただし、当該クラスは、各アイテムのアイテムクラスに含まれるクラスであるとする。

$$\begin{aligned} &t = (cl|it_1, it_2, \cdots, it_n), it_i \in I \\ &\text{if } k \neq l, \text{then } it_k \neq it_l \\ &cl \in C_{it_i} \subseteq C \end{aligned} \tag{1}$$

2.2 発見基準

各クラスに属するクラス付きアイテム集合の数は必ずしも一定ではなく、多くのクラス付きアイテム集合が存在するものと、少しのクラス付きアイテム集合しか存在しないものがある。多くのクラス付きアイテム集合が存在するクラスによく現れるパターンと、クラス付きアイテム集合自体が少ないクラスによく現れるパターンが一致している場合には問題ないが、一致しない場合には、多くのクラス付きアイテム集合が存在するクラスによく現れるパターンの方が、その頻度が高い傾向にある。このため、頻度の少ないクラスと関連の深いパターンは、相対的にその頻度が低くなる傾向にあり、頻出パターンとして発見されにくくなる。そこで、このようなパターンの見逃しを回避するために、式 (2) によって、パターンの支持度を再定義する。以下においては、本支持度を特徴支持度と呼ぶこととする。

$$supp_{char}(p) = \frac{n(p)}{N(C_p)} \tag{2}$$

式 (2) においては、p がアイテム集合で構成されたパターン、$n(p)$ が p を含むクラス付きアイテム集合の数、C_p が p に対応するアイテム集合が所属する可能性のあるクラスの集合、$N(C_p)$ が C_p に含まれるクラスのいずれかが割り当てられたクラス付きアイテム集合の数とする。なお、C_p は式 (3) によって定義されるとし、パターンクラスと呼ぶこととする。

$$C_p = \bigcap_{it_i \in p} C_{it_i} \tag{3}$$

ここで定義した特徴支持度が指定したしきい値以上となるものを抽出することにより、クラスを考慮した特徴的なパターンを発見することが期待できる。しかしながら、パターンを抽出する元となるクラス付きアイテム集合の個数が著しく少なくなる可能性がある。このような場合、特徴支持度が高いといっても、偶然現れたとも考えられ、そのパターンの信頼性はそれ程高いものにはならないと考えられる。そこで、発見されるパターンの頻度に関する指標も新たに導入することとする。この頻度は、[Calders et al. 07] によって定義された代表度 (representativity) と等価なものである。

連絡先: 〒 212-8582 神奈川県川崎市幸区小向東芝町 1
(株) 東芝 研究開発センター 知識メディアラボラトリー
Tel:044-549-2397 E-mail:shigeaki.sakurai@toshiba.co.jp
櫻井 茂明

次に、特徴支持度と頻度のパターンの成長に対する性質について着目してみることにする。あるパターンを含むより大きなパターンにおける頻度は、元のあるパターンの頻度と同程度以下であり、頻度に関しては、単調に減少するといった性質が成立する。これに対して、特徴支持度に関しては、このような単調性は必ずしも成立しておらず、パターンが大きくなった場合に、特徴支持度が大きくなることがある。

頻出パターンの発見問題においては、パターンの単調性を利用することにより、効率的に頻出パターンを発見している。このため、特徴支持度を直接利用した場合には、効率的に特徴支持度を発見することは困難である。

そこで、[Sakurai and Mori 10] によって提案された、可能性支持度と同様の支持度を、新たな可能性支持度として式 (4) により定義する。ただし、$T(cl_i)$ は cl_i が割り当てられたクラス付きアイテム集合の数とする。

$$supp_{pos}(p) = \frac{n(p)}{\min\limits_{cl_i \in C_p} \{T(cl_i)\}} \tag{4}$$

定義した可能性支持度の場合、分子の値であるパターンに対応したクラス付きアイテム集合の数は、パターンが大きくなるに連れて単調に減少する。一方、分母の値はパターンクラスに割り当てられたクラスにおけるクラス付きアイテム集合の数の最小値である。当該パターン p を含むパターン $p'(\supseteq p)$ の場合、明らかに $C_{p'} \subseteq C_p$ の関係が成立している。このため、$\min\limits_{cl_i \in C_{p'}} \{T(cl_i)\} \geq \min\limits_{cl_i \in C_p} \{T(cl_i)\}$ といった関係が成立する。従って、分子が単調に減少する一方、分母が単調に増大するため、可能性支持度は単調に減少することになり、単調性が成立している。また、特徴支持度と可能性支持度の大小関係を比較してみると、分子の値が同一である一方、特徴支持度を算出する可能性のあるパターン p に関しては、$N(C_p) \geq \min\limits_{cl_i \in C_p} T(cl_i)$ の関係が成立している。このため、特徴支持度を算出する可能性のあるパターンに対して、式 (5) に示す関係が成立している。

$$supp_{pos}(p) \geq supp_{char}(p) \tag{5}$$

式 (5) に示す大小関係により、可能性支持度は、パターン p を含むパターンの特徴支持度の上限を与えている。この性質に基づいて、特徴支持度が最小支持度より小さくなるパターンであっても、可能性支持度以上となるようなパターンは、より大きなパターンを生成するための種として保存しておくことにより、特徴支持度が最小支持度以上となるすべてのパターンを効率的に発見することができる。なお、パターンの頻度が最小頻度以上であり、その可能性支持度が最小支持度以上であるパターンを、以下においては可能性パターンと呼ぶことにする。

2.3 発見アルゴリズム

本節では、クラス付きアイテム集合からの特徴パターンの発見法として、[Han et al. 00] に提案されている、FP-tree 及び FP-growth に基づいた方法を提案する。FP-tree は、パターンの頻度を高速にカウントできるように、アイテム集合を木構造の形式に格納したものである。クラス付きアイテム集合向けの FP-tree においては、そのヘッダーに、パターンに対応するパターンクラスを格納する領域とパターンが特徴パターンであるか可能性パターンであるかを識別するフラグが追加されている。本 FP-tree は、表 1 に示すアルゴリズムを実施することにより生成される。表 1 においては、従来の FP-tree の生成法と異なる部分が太字にて記載されている。

1. 入力されたアイテム集合をサーチして、各アイテムの頻度を算出する。
2. **初期パターン Its のパターンクラスと各アイテム it_i のアイテムクラスの積集合を求め、Its と it_i からなるパターン p_i のパターンクラス C_{p_i} とする。**
3. **C_{p_i} のクラスに対応するクラス付きアイテム集合の総数及び最小値を算出する。**
4. **p_i の特徴支持度、可能性支持度を算出する。**
5. **ステップ 2. からステップ 4. を、任意の i で繰り返す。**
6. **最小頻度以上の頻度かつ、最小支持度以上の可能性支持度を与えるアイテムの集合を求め、頻度を第 1 キー、特徴支持度を第 2 キー、可能性支持度を第 3 キーとして、降順にアイテムを並べたリスト $Flist$ を生成する。**
7. $Flist$ を FP-tree T のヘッダー H とする。**また、H の各アイテム it_i と C_{p_i} を関連付けて格納する。加えて、p_i の特徴支持度が最小支持度以上であるかどうかを示すフラグを it_i と関連付けて格納する。**
8. T のルートを生成し、ラベル「null」を割り当てる。
9. ルートをノード N として設定する。
10. クラス付きアイテム集合 t_i をひとつ取り出す。このとき、取り出すクラス付きアイテム集合が無ければ、アルゴリズムを終了する。
11. t_i を構成するアイテムの中から H に含まれるアイテムだけを取り出して、H の順に並び替えたアイテム集合 Its_{t_i} を生成する。
12. Its_{t_i} の先頭からアイテム it_{ij} を取り出す。このとき、取り出すアイテムが無ければ、ステップ 10. へ戻る。
13. N の子ノードの中に、it_{ij} に一致するアイテム名がラベル付けされたノード N_{child} があれば、N_{child} の頻度を 1 増やし、N_{child} を新たな N とする。一方、N_{child} がなければ、新たなノード N' を生成して、it_{ij} のアイテム名を当該ノードに割り当てて、その頻度を 1 とする。また、N と N' を結ぶリンクを生成し、N' を新たな N とする。加えて、N' を H の it_{ij} に登録する。
14. ステップ 12. へ戻る。

表 1: クラス付きアイテム集合向けの FP-tree の生成法

一方、クラス付きアイテム集合向けの FP-growth は、従来の FP-tree と同様に、アイテム集合から FP-tree を生成する。また、FP-tree から特定のアイテムで条件付けられたアイテム部分集合を生成する。この FP-tree の生成とアイテム部分集合の生成を再帰的に繰り返すことにより、すべての頻出するパターンを発見することができる。表 2 は本アルゴリズムを示しており、従来の FP-tree との違いが太字にて示されている。

以上に示した、FP-tree の生成法と FP-growth により、すべての特徴パターンを効率的に発見することができる。

3. 数値実験

3.1 実験方法

機械学習分野のベンチマークテストとして頻繁に利用される UCI のデータを、実験データとして利用する。UCI のデータには多数のものが存在するが、それらの中から、属性値が離散値のみで構成されており、クラスが付与されたデータ集合として、car, hayes, nursery のデータを利用する。ただし、ア

1. FP-tree T のヘッダーの後方からアイテム it を取り出す。このとき、取り出すアイテムが無ければ、アルゴリズムを終了する。
2. it をアイテム集合 Its に追加し、it **に関連付けられているフラグを参照する。このとき、当該フラグにより特徴支持度以上であると判別できれば、**Its を頻出パターンとして出力する。
3. T の中から、it のアイテム名と一致するノード N をひとつ取り出す。このとき、取り出すノードが無ければ、ステップ 6. へ進む。
4. N からルートまでのパスに存在するアイテム集合を取り出し、取り出したアイテム集合の頻度を N における頻度とするアイテム集合を生成する。
5. ステップ 3. へ戻る。
6. 生成したアイテム集合の集合 $Trans$ を表 1 のアルゴリズムに適用して、FP-tree T' を生成する。このとき、$T' = \phi$ であるならば、本アルゴリズムを終了する。さもなければ、T' 及び Its を引数として本アルゴリズムを呼び出す。
7. ステップ 1. へ戻る。

表 2: クラス付きアイテム集合向け FP-growth アルゴリズム

イテムクラスは与えられていないので、各事例における属性値とクラスとの間の関係を調査し、属性値と同時に現れるクラスの集合をアイテムクラスとして設定する。

また、最小支持度及び最小頻度を変化させて、提案法による実験データからの特徴パターンの発見を試みる。このとき、参考のために、特徴パターンを発見するのに必要となる可能性パターンも併せて出力する。一方、比較のために、クラスを考慮しない従来法において、頻出パターンを発見することも試みる。ただし、頻出パターンの発見においては、アイテム集合に付与されているクラスは無視することにする。また、特徴パターンの発見において設定する最小支持度及び最小頻度に基づいて、頻出パターンの発見における最小頻度を設定する。最終的には、このようにして発見したパターンの違いを比較することにより、提案法の効果を検証する。

3.2 実験結果

図 1 及び図 2 に結果の一部を示す。各図は、グラフ (a)∼ グラフ (c) の 3 つのグラフから構成されており、各グラフがデータセットに対応した結果を示している。図 1 は、提案法によって発見される特徴パターンと、従来法によって発見される頻出パターンとの違いを示している。図 1 においては、“Class only”が発見された特徴パターンと頻出パターンを比較した場合に、特徴パターンのみによって発見されるパターンの数を示しており、“Class common”が共通に発見されるパターンの数を示しており、“No class only”が従来法によってのみ発見されるパターンの数を示している。

次に、図 2 は、特徴パターンの発見に利用される可能性パターンと、頻出パターンとの違いを示している。図 2 においては、最小頻度の変化がどのように頻出パターンの発見に影響するかを確認するために、最小頻度を変化させた場合の頻出パターンと可能性パターンとの比較を行っている。また、“Class”が可能性パターンの数を示しており、“No class”が頻出パターンの数を示している。

これらグラフにおいては、水平軸が最小支持度を表しており、垂直軸がパターンの数を表している。ただし、従来法において設定される最小頻度の値は、最小支持度に換算されて表示されている。

3.3 考察

発見されるパターンの違い: UCI データにおける特徴パターンと頻出パターンの違いに着目してみると、提案法における最小頻度を小さくしていくことにより、従来法では発見できなかったパターンの発見に成功している。このようなパターンを従来法で発見するには、その最小頻度を、提案法で設定したレベルにまで低くする必要がある。しかしながら、このような設定を実施した場合、従来法では多数の頻出パターンが発見されることになる。すなわち、図 2 の各グラフの最左端に着目してみると、実に、car, hayes, nursery の場合で、135 倍、15.1 倍、85.2 倍におよぶ頻出パターンを発見している。このため、従来法では、特徴パターンが多数の頻出パターンの中に埋没する危険性がある。

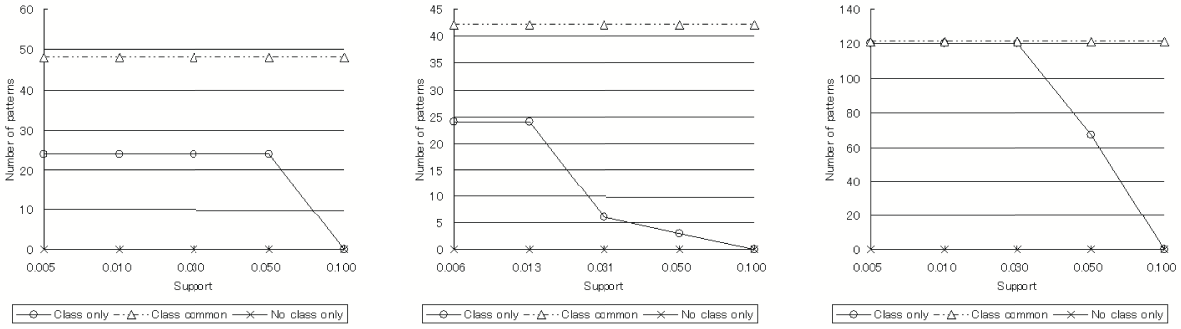
一方、従来法で頻出パターンを発見した後で、発見された頻出パターンに対して特徴パターンを発見するといった 2 段階の方法を実施することも可能である。このような 2 段階法により、特徴パターンの埋没を回避することが期待できる。しかしながら、ここで、可能性パターンの数に着目してみると、可能性パターンの数は、頻出パターンの数よりも少なくなっている。実に、car, hayes, nursery の場合で、0.376 倍、0.461 倍、0.404 倍である。すなわち、提案法は、2 段階法の半分以下のパターンをチェックするだけで、特徴パターンを発見することに成功している。提案法は、2 段階法に比べて、より処理負荷が軽い手法になっているといえる。

アイテム数の影響: 図としては記載していないが、car 及び nursery においては、パターンを構成する要素の数がひとつの場合に、特徴パターンと頻出パターンは完全に一致している。しかしながら、より多くの要素を含むパターンにおいて、特徴パターンのみによって発見されるパターンが観測されている。複数アイテムを組み合わせることにより、パターンクラスを構成するクラスの数が少なくなり、最小支持度以上となるのに必要となる頻度が少なくなったことが影響しているものと考えられる。必ずしも頻度がそれ程多くないパターンであったとしても、特定のクラスの組合せの中では、比較的よく現れるパターンを発見するといった、意図していた効果が実験的にも確認されたといえる。

前処理との組合せ: 従来法ではクラスを直接考慮できないとしても、従来法の前処理によって、クラスごとあるいはクラスの組合せごとにデータを分割し、その分割したデータに対して、頻出パターンの発見を試みることは可能である。このような前処理と従来法との組合せは、クラスの数が少ない場合には、ある程度有効に機能することが期待できる。しかしながら、クラスの数が多い場合には、クラスの組合せ自体が非常に多くなるといった問題がある。一方、多くの場合、どのようなクラスの組合せが有望であるかは、事前には分からないため、多数のクラスの組合せを逐一分析することが必要である。このため、前処理との組合せは、クラスの数が多い場合に、非常に労力がかかるものとなる。

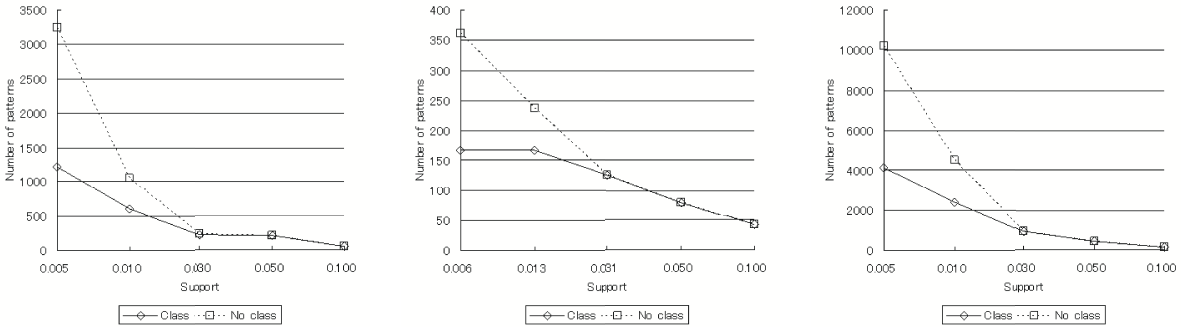
提案法においては、このような事前知識を仮定すること無しに、クラスを考慮したパターンを発見することができ、前処理と従来法との組合せよりも優れているといえる。

クラスの設定: 提案法におけるアイテム集合に付与するクラスとしては、多様なものを考えることができる。例えば、RFID データの分析では、店内における商品の動きをデータとして収



(a) car の場合　(b) hayes の場合　(c) nursery の場合

図 1: 特徴パターンと頻出パターン



(a) car の場合　(b) hayes の場合　(c) nursery の場合

図 2: 可能性パターンと頻出パターン

集することができる。このため、試着の有無などをクラスとみなすことにより、試着されて購入される商品と試着されずに購入される商品の特性を加味した分析を行うことができる。また、必ずしも RFID データではないとしても、店舗の違いをクラスとみなすことにより。店舗の特性を反映した分析を行うこともできる。さらには、購入時期の違いをクラスとみなすことにより、時期的な特性を反映した分析を行うこともできる。このように、多様なクラスを設定することにより、多様は観点でクラス付きアイテム集合を分析することができ、提案法は高い汎用性を備えているといえる。

上記議論に基づいて、クラス付きアイテム集合を多様な観点で効率よく分析する提案法は、有効な分析法であると考えられる。

4. まとめと今後の課題

本論文では、データ収集条件の違いをクラスと考え、多数のクラス付きアイテム集合から、特徴的なパターンを効率的に発見する方法を提案した。提案法においては、従来提案されていた特徴支持度及び可能性支持度を再定義することにより、特徴的なパターンを効率的に発見することを可能としている。また、提案する発見法の効果を、UCI データを用いた従来法との比較を通して検証した。

今後の課題としては、提案法のより高速化を検討していきたい。そのひとつの方向性としては、可能性支持度をより小さく見積もり、冗長な可能性パターンを生成しないといった方向性が考えられる。具体的には、現在までのパターンの成長状況によって、そのパターンによって拡張される可能性のあるパターンを限定し、拡張されるパターンのパターンクラスをより限定する方法を検討している。これにより、可能性支持度をより小さく評価することができるのではないかと考えている。また、別の方向性としては、今回の提案法では、単純な組合せにはならないため、本来の FP-growth では考慮している single prefix-path(ルートから続く、子ノードをひとつだけしか持たないノードからなるパス) を考慮していないものの、同様な方法の導入可能性も今後検討していきたい。この他、通常のアイテム集合からのパターン発見もより進化しているので、それら進化の適用可能性も検討していきたいと考えている。

参考文献

[Calders et al. 07] Calders, T., Goethals, B. and Mampaey, M., “Mining Itemsets in the Presence of Missing Values”, Proc. 2007 ACM Sympo. on Applied Computing, 404-408 (2007).

[Han et al. 00] Han, J., Pei, J. and Yin, Y., “Mining Frequent Patterns without Candidate Generation”, Proc. 2000 ACM SIGMOD Intl. Conf. on Management of Data, 1-12 (2000).

[Sakurai and Mori 10] Sakurai, S., and Mori, K. “Discovery of Characteristic Patterns from Tabular Structured Data including Missing Values”, Intl. J. of Business Intelligence and Data Mining, 5, 3, 213-230 (2010).