頻出パターン発見法における欠損値の取扱い法

A Process Method of Missing Values for a Discovery Method of Frequent Patterns

櫻井 茂明[*1] 森 紘一郎[*1]
Shigeaki Sakurai Kouichirou Mori

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

In the case of dealing with tabular structured data, the discovery methods of frequent patterns require that each item is composed of the combination of an attribute and an attribute value. Some attribute values in the data can be missing due to the constraints of data collection. Thus, this paper proposes a new method that deals with the missing values based on the two kinds of supports. This method predicts the number of examples not including missing values in order to reduce the number of candidate patterns. This paper also verifies the effectiveness of the method by comparing with existing process methods of missing values.

1. はじめに

データを収集する際の制約条件などの問題によって、表構造データの特定の事例の特定の属性には、値が設定できない場合がある。そのような値は欠損値と呼ばれており、クラス付きの表構造データを対象とする帰納学習法では、以下に示すような前処理を実施し、欠損値を取り除くことが行われている。

1. 欠損値を含む事例そのものを削除する。
2. 現存する属性値の分布に基づいて欠損値を補完する。
3. 欠損値を表す特殊な属性値を設定する。

しかしながら、1. の方法の場合、欠損値を含む事例の他の属性に関する情報をも捨てることになるため、与えられているデータを必ずしも有効に利用していないという問題があった。また、2. の方法の場合、補完された値は推定値に過ぎないため、その後の学習において、推定値に依存した誤った学習が行われる危険性があった。加えて、3. の方法の場合、本来は意味の異なるものをひとつにまとめて扱うことになるため、学習されるモデルの中に、解釈が困難な部分が含まれる危険性があった。

一方、頻出パターンから抽出される相関ルールの発見法に基づいた欠損値の処理法として、[Ragel 98a] は、欠損値を含む事例から抽出された相関ルールを利用することにより、欠損値を補完する方法を提案している。また、[Shintani 06] は、欠損値を含むデータの属性に基づいて、与えられているデータベースを複数のデータベースに分割し、分割したデータベースにおいて、指定された最小事例数、最小支持度、最小信頼度以上となる相関ルールを発見する方法を提案している。

これら相関ルールの発見法における欠損値の処理法に対して、我々のグループでは、[Sakurai et al 09] で、2 種類の支持度に基づいて、現存する属性値を有効に活用した欠損値を含んだ表構造データからのパターンの発見法を提案した。しかしながら、欠損値の分布によっては、パターン発見の途中段階で保持される候補パターンの数が多くなる問題が指摘されていた。

そこで、本論文では、全属性において欠損値を含まない事例の数を精度よく見積もることにより、途中段階で生成される候補パターンの数を少なくする方法を提案する。また、提案法を組み込んだ欠損値を含むデータからの頻出パターンの発見法を提案する。最終的には、提案法の効果を、人工データ及び UCI サイトから収集したデータに基づいて検証する。

連絡先: 櫻井 茂明, (株) 東芝 研究開発センター, 〒 212-8582
神奈川県川崎市幸区小向東芝町 1, Tel:044-549-2397,
Fax:044-520-1308, E-mail:shigeaki.sakurai@toshiba.co.jp

2. アイテム表現と支持度

2.1 アイテム表現

表構造データを構成する各事例 t が n 個の属性の属性値によって、式 (1) に示すように与えられているとする。このとき、本事例は式 (2) に示すアイテムの集合と解釈することができる。ただし、A_i, $(i = 1, 2, \cdots, n)$ が属性、a_{ix_i} が属性 A_i の属性値を表すとし、属性と属性値を組にした $A_i : a_{ix_i}$ がアイテムを表すとする。

$$(a_{1x_1}, a_{2x_2}, \cdots, a_{nx_n}) \tag{1}$$

$$(A_1 : a_{1x_1}, A_2 : a_{2x_2}, \cdots, A_n : a_{nx_n}) \tag{2}$$

2.2 支持度

欠損値を含む事例が与えられた場合に、指定したパターン s における特徴支持度 $supp_{char}(s)$ 及び可能性支持度 $supp_{pos}(s)$ を、式 (3)、式 (4) によりそれぞれ定義する。ただし、属性パターンは指定された属性の集合を表すとし、属性パターンの無欠損事例数とは、当該属性パターンに含まれる属性だけから構成される事例を考えた場合に、欠損値を含む事例を除去した後に残存する事例の数を表すとする。以下においては、属性パターンを構成する属性の個数に着目する必要がある場合に、k 個の属性からなる属性パターンを、k 次属性パターンと呼ぶことにする。

$$supp_{char}(s) = \frac{s \text{ を含む事例数}}{s \text{ に対応する属性パターンの無欠損事例数}} \tag{3}$$

$$supp_{pos}(s) = \frac{s \text{ を含む事例数}}{\text{全属性を含む属性パターンの無欠損事例数}} \tag{4}$$

このとき、任意のパターン s に対して、特徴支持度と可能性支持度との間には、式 (5) に示す関係が成立する。

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

3. 無欠損事例数の推定

[Sakurai et al 09] による提案法では、可能性支持度が最小支持度以上となるパターン (可能性パターン) を記憶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パターン (特徴パターン) を、効率的にすべて発見することができる。しかしながら、n 次属性パターン無欠損事例数が小さい場合、可能性支持度が著しく小さくなるため、特徴パターンの数に比べて、多くの可能性パターンを記憶する必要があった。

ここで、抽出される特徴パターンに着目してみれば、すべての属性から構成されるような特徴パターンは稀にしか存在しておらず、特徴パターンは通常一部の属性だけから構成されている。このため、n 次属性パターン無欠損事例数に基づいた可能性支持度は、すべての特徴パターンを発見するために、過度に多くの可能性パターンを記憶していることになる。一方、n 次属性パターン無欠損事例数が 0 になる場合を考えてみることにする。このとき、全属性からなる特徴パターンは、明らかに存在しないものの、一部属性だけからなる特徴パターンまでも存在しないと結論付けることはできない。しかしながら、n 次属性パターン無欠損事例数が 0 と与えられており、可能性支持度を算出することができないため、提案法では、一部属性だけからなる特徴パターンまでも発見することはできない。そこで、提案法によるこれら問題を回避するために、全属性からなるような特徴パターンの発見を目指すのではなく、指定された属性数 $q \in [2, n)$ 以下の長さとなる特徴パターンだけを発見することを考えてみることにする。

このとき、すべての特徴パターンを発見することを目指すとすれば、q 次属性パターンのそれぞれに対して、q 次属性パターン無欠損事例数を算出し、その事例数の最小値を n 次属性パターン無欠損事例数と置き換えて、可能性支持度を計算する必要がある。しかしながら、q 次属性パターンの種類は、すべての属性の中から指定した属性数の属性を取り出す組み合わせだけ存在するため、その種類はかなり多くなることが予想される。すなわち、q 次属性パターン無欠損事例数の最小値を計算するには、長い計算時間が必要となる。一方、欠損値を多く含む属性の場合、特徴パターンを抽出するのに参照する事例の数自体が少なくなるため、特徴支持度を超えたとしても、そのような特徴パターンはやや信頼性が劣ったものになる。このため、このような属性に関連した特徴パターンを若干見逃したとしても、それ程大きな問題にはならないと考えられる。そこで、本論文では、q 次属性パターン無欠損事例数の最小値を算出するのではなく、平均的な q 次属性パターンにおける無欠損事例数を推定することにより、当該推定値に基づいて可能性支持度を再定義することにする。具体的には、各属性における属性値の存在確率を平均した、平均存在確率 p_{avg} を式 (6) を用いて算出し、平均 q 次属性パターン無欠損事例数を式 (7) を用いて算出する。最終的には、可能性支持度 $supp_{pos}$ を式 (8) を用いて再定義する。

$$p_{avg} = \frac{1}{n} \cdot \sum_{i=1}^{n} \frac{A_i\text{の属性パターン無欠損事例数}}{\text{事例数}} \quad (6)$$

$$\text{平均 } q \text{ 次属性パターン無欠損事例数} = p_{avg}^{q} \cdot \text{事例数} \quad (7)$$

$$supp_{pos}(s) = \frac{s \text{ を含む事例数}}{\text{平均 } q \text{ 次属性パターン無欠損事例数}} \quad (8)$$

4. 特徴パターンの発見法

[Sakurai et al 09] で提案した特徴パターンの発見法を図 1 に示すように改良する。図 1 においては、表構造データで与えられる M 個の事例からなる事例集合 DB、最小支持度 $MinSp$、発見する特徴パターンの最大長 q を入力とすることにより、長さ q 以下となる特徴パターンを発見する。ただし、St は属性パターン及び属性値パターンに関する情報を格納する構造体、$calFq()$ は構造体に設定されている属性パターンに対応するアイテム集合の頻度を算出する関数、$output()$ は構造体に含まれる特徴パターンを出力する関数、$jdgPt()$ は構造体に可能性パターンが存在するかどうかを判定する関数、$addQ()$ は構造体をキューに格納する関数、$pkQ()$ はキューから構造体を取り出す関数、$delSt()$ は構造体を削除する関数、$crtSt()$ は構造体を生成する関数、$genAtPt()$ はふたつの構造体に含まれる属性パターンを組み合わせてより高次の属性パターンを生成する関数、$genVlPt()$ は構造体に含まれる属性パターンと属性値パターンからアイテム集合を生成する関数を表すとする。

$A = \cup_{i}^{n} A_i$;
//1 次特徴パターン発見
for(each attribute $A_i \in A$){
  $crtSt(\&St)$;
  $St.AtPtAry = A_i$;
  $St.VlPtAry = \phi$;
  for(each attribute value $a_{ij} \in A_i$){
    add a_{ij} to $St.VlPtAry$;
  }
  $calFq(St, DB, \&FqAry, \&ev_{A_i})$;
  $p_{A_i} = \frac{ev_{A_i}}{n}$;
}
$p_{avg} = \frac{1}{n} \cdot \sum_{i=1}^{n} p_{A_i}$;
$Ev = p_{avg}^{q} \cdot M$;
for(each attribute $A_i \in A$){
  $k = 0$;
  for(each attribute value pattern $VlPt_k \in St.VlPtAry$){
    $supp_{char} = \frac{FqAry[k]}{ev_{A_i}}$;
    IF $supp_{char} \geq MinSp$;
      Then $output(St, k)$;
      Else $supp_{pos} = \frac{FqAry[k]}{Ev}$;
        IF $supp_{pos} < MinSp$;
          Then delete $VlPt_k$ from $St.AtvPtAry$;
    $k = k + 1$;
  }
  IF $jdgPt(St) == true$;
    Then $addQ(St, \&Q_1)$;
    Else $delSt(St)$;
}
//高次特徴パターン発見
$i = 1$;
while($true$){
  while($(tSt1 = pkQ(Q_i))! = NULL$){
    for(each attribute pattern $tSt2 \in Q_i$){
      $crtSt(\&St)$;
      $genAtPt(tSt1.AtPtAry, tSt2.AtPtAry, \&St.AtPtAry)$;
      $genVlPt(tSt1.VlPtAry, tSt2.VlPtAry, \&St.VlPtAry)$;
      $calFq(St, DB, \&FqAry, \&ev)$;
      $k = 0$;
      for(each attribute value pattern $VlPt_k \in St.VlPtAry$){
        $supp_{char} = \frac{FqAry[k]}{ev}$;
        IF $supp_{char} \geq MinSp$;
          Then $output(St, k)$;
          Else $supp_{pos} = \frac{FqAry[k]}{Ev}$;
            IF $supp_{pos} < MinSp$;
              Then delete $VlPt_k$ from $St.VlPtAry$;
        $k = k + 1$;
      }
      IF $i + 1 < q$;
        Then IF $jdgPt(St) == true$;
                Then $addQ(St, \&Q_{i+1})$;
                Else $delSt(St)$;
        Else $delSt(St)$;
    }
    $i = i + 1$;
  }
  $i = i - 1$;
  IF $i == 0$; Then break;
}

図 1: 特徴パターン発見法

5. 数値実験

5.1 実験方法

欠損値を考慮した特徴パターンの発見法の効果を検証するために、乱数を用いて欠損値を含む事例を生成した人工データ

と、UCI サイト (http://archive.ics.uci.edu/ml/) の実データを用いることにする。

乱数を用いた人工データにおいては、各属性が一定の個数の属性値を持った属性で構成された事例に対して、各属性値を指定した欠損値率 ($=\frac{\text{欠損値数}}{\text{事例数} \times \text{属性数}}$) に基づいて、欠損値を挿入した事例を表 2 の生成法に従って生成する。ただし、trnum を事例数、atnum を属性数、atvnum を属性値数、mrate を欠損値率として与えることにする。また、乱数の初期値により生成される事例集合が異なるため、乱数の初期値を 10 種類設定することにする。

1. trnum, atnum, atvnum, mrate の読込み。
2. trcnt=0, atcnt=0
3. trcnt が trnum 以上ならば、終了。
4. atcnt が atnum 以上ならば、ステップ 9. へ進む。
5. atcnt+=1
6. $[0, 1)$ の範囲の乱数を生成。
7. 乱数値が mrate 以下ならば、当該事例の属性値の値を欠損値に設定。乱数値が mrate より小さいならば、$[0, atvnum)$ の範囲の乱数を生成して、設定する属性値を決定。
8. ステップ 4. に復帰。
9. trcnt+=1, atcnt=0 とし、ステップ 3. に復帰。

図 2: 人工データ生成法

一方、UCI サイトのデータとして、欠損値を含む離散値から構成され、比較的広範囲に欠損値が散らばっている Congressional Voting Records を利用する。本データはアメリカ議会における投票行動をデータ化したものである。本データは、16 の属性とひとつのクラスからなる 435 個の事例から構成されている。ただし、特徴パターンの発見においては、属性とクラスを区別する必要がないため、クラスも属性とみなすことにする。

上記 2 種類のデータを用いて、[Sakurai et al 09] で提案した特徴パターンの発見法を用いて特徴パターンを発見するとともに、提案する無欠損事例数の推定法を導入した発見法を用いて特徴パターンを発見する。また、従来の欠損値の処理法と比較するために、欠損値を含む事例をすべて除いた事例集合から特徴パターン (欠損値が存在しないので、頻出パターンと同じ) を発見するとともに、各属性における属性値の出現確率を用いて、欠損値を補完した事例集合から特徴パターン (欠損値が存在しないので、頻出パターンと同じ) を発見する。ただし、欠損値の補完法においては、乱数の初期値に依存して補完される属性値が異なるため、乱数の初期値を 10 種類設定して、補完することにする。

5.2 実験結果

図 3、図 4 に実験結果の一部を示す。図 3 が人工データにおいて、データ数を 10,000、属性数を 15、属性値数を 0.3、最小支持度を 0.01、無欠損事例数の推定法における最大長を 10 とした場合の結果を示している。本人工データの場合においては、初期値を変えて 10 回実施した実験のうちのひとつの結果を示している。また、図 4 が UCI の Congressional Voting Records を利用した場合において、無欠損事例数の推定法における最大長を 10 とした場合の結果を示している。

各図においては、(a) が [Sakurai et al 09] に提案した欠損値を考慮した発見法と、無欠損事例数の推定法を利用した発見法において生成される可能性パターンの数の変化を示している。(a) においては、可能性パターンの長さが x 軸、可能性パターンの数が y 軸に対応している。また、Original が [Sakurai et al 09] で提案した発見法の結果を示しており、Refined が無欠損事例数の推定法を利用した発見法の結果を示している。

(b) は提案法と削除法に基づいて発見される特徴パターンの違いが変化する様子を示してる。(b) においては、特徴パターンの長さが x 軸、発見された特徴パターンを各手法で比較した場合の特徴パターンの数が y 軸に対応している。また、Common が従来の提案法及び削除法で共通に発見される特徴パターンの数、Original only が従来の提案法で発見される一方、削除法では発見されない特徴パターンの数、Deletion only が削除法で発見される一方、従来の提案法では発見されない特徴パターンの数を示している。

(c) は提案法と補完法に基づいて発見される特徴パターンの違いが変化する様子を示してる。(c) においては、各軸は (b) と同様な意味を持っており、Common が従来の提案法及び補完法で共通に発見される特徴パターンの数、Original only が従来の提案法で発見される一方、補完法では発見されない特徴パターンの数、Completion only が補完法で発見される一方、従来の提案法では発見されない特徴パターンの数を示している。

5.3 考察

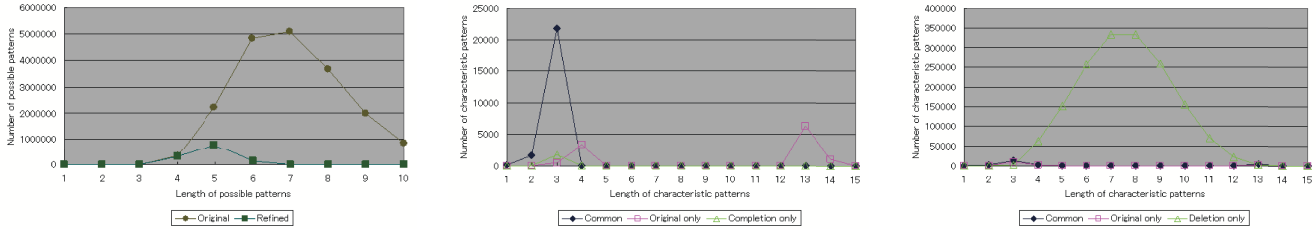
無欠損事例数の推定法: 無欠損事例数の推定法を導入した発見法は、ほとんどの実験において、指定した長さ以下の特徴パターンをすべて発見することに成功している。また、図 3、図 4 の (a) に結果を示すように、途中段階で発見される可能性パターンの数は少なくなっており、この傾向は、欠損値率が高い程顕著になっている。このため、提案する無欠損事例数の推定法は、欠損値率が高く、あまり長くない特徴パターンを発見する場合に有効であるといえる。

加えて、無欠損事例数の推定法は、全属性を対象とした無欠損事例数が 0 になるような事例集合からも、特徴パターンを発見することができる。このような場合、元々全属性を対象とする特徴パターンは存在していない。一方、欠損値でない属性値を最も多く含む事例における属性値数を、発見する特徴パターンの最大長として指定することにより、本最大長以下の長さを持つ特徴パターンを発見することができる。この面でも、無欠損事例数の推定法は有効であるといえる。

評価に利用する事例数: 提案法と補完法における評価に利用する事例数の違いに着目してみると、補完法においては、見掛け上すべての事例を利用して特徴パターンを発見している。しかしながら、その中の一部は補完された属性値である。これに対して、提案法では欠損値を除いたすべての属性値を利用した特徴パターンを発見している。このため、利用している事例数の差は補完された属性値の差とみなすことができる。図には示していないものの、特徴パターンの数が長くなるに従って、補完された属性値を利用する割合は高くなっており、補完法によって発見される特徴パターンは、その信頼性が劣ったものになると考えられる。

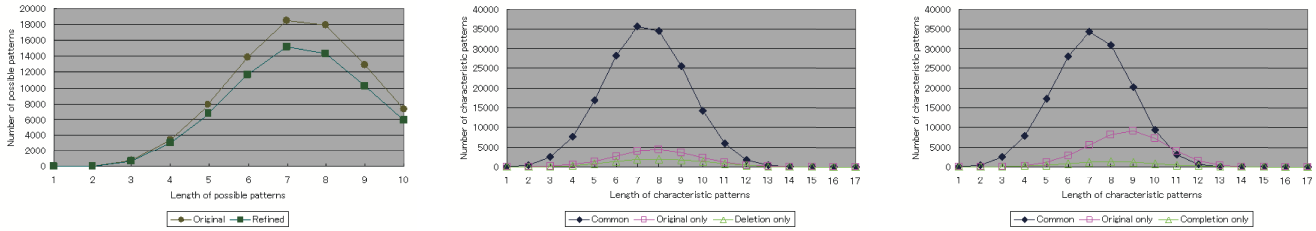
次に、提案法と削除法に着目してみることにする。削除法においては、短い特徴パターンを発見する際に、提案法よりもかなり少ない事例に基づいて特徴パターンを発見している。より多くの事例に基づいて発見した特徴パターンの方が、少ない事例に基づいて発見した特徴パターンよりも通常妥当なものと考えられる。このため、削除法に基づいた特徴パターンは、その長さが短い場合に信頼性が劣ったものになると考えられる。

特徴パターンの違い: 図 3、図 4 の (b) 及び (c) に結果を示



(a) 可能性パターンの数　(b) 特徴パターンの違い (欠損考慮 vs 欠損削除)　(c) 特徴パターンの違い (欠損考慮 vs 欠損補完)

図 3: 実験結果:人工データ (データ数:10,000、属性数:15、属性値数:5、欠損値率:0.3、最小支持度:0.01、最大長:10)



(a) 可能性パターンの数　(b) 特徴パターンの違い (欠損考慮 vs 欠損削除)　(c) 特徴パターンの違い (欠損考慮 vs 欠損補完)

図 4: 実験結果:Congressional Voting Records(データ数:435、属性数:17、属性値数:5、欠損値率:0.05、最小支持度:0.1、最大長:10)

すように、対象としている事例集合により、その形状は大きく異なっている。しかしながら、提案法で発見される特徴パターンと削除法及び補完法によって発見される特徴パターンには、かなりの違いがあることを確認することができる。評価に利用する事例数の考察から、提案法によって発見される特徴パターンは、削除法及び補完法によるものよりも与えられている事例集合を忠実に反映した特徴パターンと考えられる。このため、従来の提案法では、誤った特徴パターンが数多く算出されていたことを確認することができる。

以上の考察に基づいて、無欠損事例数の推定法を導入した発見法は、従来の提案法よりも現存する事例の属性値を有効活用した特徴パターンを発見することができる。特に、それ程長い特徴パターンを発見する必要がない場合に、無欠損事例数の推定法はより少ない計算機資源で特徴パターンを発見することができる。

6. まとめと今後の課題

本論文では、[Sakurai et al 09] に提案した欠損値を考慮した特徴パターンの発見法において、途中段階で生成する候補パターンを削減するために、無欠損事例数の推定法を導入した特徴パターンの発見法を提案した。また、人工データ及び UCI サイトのデータを利用した比較実験を通して、提案する特徴パターンの発見法の効果を検証するとともに、無欠損事例数の推定法が事例集合における欠損値率が高いとしても、特徴パターンを効率的に発見できることを検証した。

今後の課題としては、本論文では、人工データ及び比較的事例数の少ない事例集合を利用して提案法の有効性を検証しているが、より大規模な事例集合に本手法を導入し、その効果を詳細に検証していきたい。また、現在の枠組みは特定の時点に収集された事例を対象としており、時系列的な事例を対象とはしていない。頻出パターンの発見問題が時系列パターンの発見問題へと拡張されたように、本特徴パターンの発見問題は時系列的な特徴パターンの発見問題へと拡張することも可能である。このため、この方面での拡張を今後検討していきたい。加えて、現在の枠組みでは、各事例は特定のクラスによってクラス分けされていないものの、対象商品の集合とその商品集合が購入されたかどうかをクラスとして、対象商品の集合に内在する違いを分析することの必要性も感じている。このため、この方面での特徴パターンの発見法の検討も行っていきたい。

参考文献

[Ragel 98a] A. Ragel: “Preprocessing of Missing Values Using Robust Association Rules”, Proc. 2nd European Sympo. on Principles of Data Mining and Knowledge Discovery, 414-422 (1998).

[Sakurai et al 09] Shigeaki Sakurai, Kouichirou Mori, and Ryohei Orihara: “Discovery of Association Rules from Data including Missing Values”, Proc. Intl. Conf. on Complex, Intelligent and Software Intensive Systems, 67-74 (2009).

[Shintani 06] T. Shintani: “Mining Association Rules from Data with Missing Values by Database Partitioning and Merging”, Proc. 5th IEEE/ACIS Intl. Conf. on Computer and Information Science and 1st IEEE/ACIS International Workshop on Component-Based Software Engineering, Software Architecture and Reuse, 193-200 (2006).