1B1-3

時系列パターン発見における制約パターンの効果

Effect of Constraint Patterns for Sequential Pattern Mining

櫻井 茂明[*1] 北原 洋一[*1] 折原 良平[*1]
Shigeaki Sakurai Youichi Kitahara Ryohei Orihara

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

This paper introduces a discovery method of sequential patterns from sequential data composed of rows of item sets. This method notes items classified into attributes and the items are described by both attributes and their attribute values. This method expresses the interests of analysts based on relationships among the attributes. The relationships are constraint patterns. Also, this method divides the constraint patterns into constraint sub-patterns, applies the sub-patterns to the generation of candidate sequential patterns, and evaluates whether the candidate sequential patterns satisfy the sub-patterns. This method can efficiently discover frequent sequential patterns corresponding to the interests. This paper uses various types of relationships and verifies the effect of the method through numerical experiments.

1. はじめに

営業日報、Web ログ、生体情報等の時系列的に与えられるデータを簡便に収集できるようになり、これらデータは着実に増加している。また、ユビキタス環境の進展に伴い、これらデータは今後益々増えることが予想されている。このため、このような時系列データを分析したいとのニーズが高まっている。論文 [Srikant and Agrawal 96] では、その分析法のひとつとして、離散的なアイテムが時系列的に並べられた時系列データから頻出する時系列パターンを、効率よく発見する方法を提案している。発見されたパターンは、与えられた時系列データを特徴付ける一種の知識と考えられる。しかしながら、頻出するパターンはありふれたパターンであることも多く、分析者が求める特徴的なパターンには、必ずしもなっていないことが指摘されている。

この問題に対して、分析者の背景知識を利用することにより、利用者が求める時系列パターンを発見する方法が提案されている [Garofalakis et al 99] [Pei et al 02]。また、我々のグループでも、アイテム間に明示的な関係性が与えられるような時系列データに着目し、アイテムに属性という概念を導入することにより、直感的に背景知識を記述可能な手法を提案している [櫻井 他 07]。本論文では、提案する属性に基づいた制約の導入法をより詳細に検証するために、多様な制約条件を設定した場合における提案法の効果を調査し、その効果を検証する。

2. 時系列パターンの発見法

2.1 時系列パターンの定義

はじめに、本論文が対象とする時系列データを定義する。時系列データは、複数のアイテム集合が時系列的に並べられたアイテム集合の系列のことである。このとき、各アイテム集合には、同一のアイテムはせいぜいひとつしか含まれていないことが仮定されており、時間的には同時刻に起こったアイテムを集めたものである。このような時系列データ s_1 は形式的には、$s_1 = (l_{11}, l_{12}, \cdots, l_{1n_1})$ と記述することができる。ただし、l_{1i} はアイテム集合であり、$i < j$ の関係がある場合には、l_{1i} は l_{1j} よりも時間的に先に起こることを表している。また、n_1 は時系列パターンを構成するアイテム集合の個数を表しているとし、この個数を系列データの長さと呼ぶことにする。各アイテム集合はアイテム v_{1ik} によって、$l_{1i} = \{v_{1i1}, v_{1i2}, \cdots, v_{1im_{1i}}\}$ と記述することができる。ただし、アイテム集合におけるアイテムに対する仮定により、$k \neq l$ の場合には、$v_{1ik} \neq v_{1il}$ といった条件が成立している。また、m_{1i} は i 番目のアイテム集合を構成するアイテムの個数とし、この個数をアイテム数と呼ぶことにする。一方、このようなアイテムは属性 A と属性値 a の対によって構成されており、$v_{1ik} = A_{1i} : a_{1ik}$ と表現することができる。例えば、({A 社:上昇, B 社:上昇, C 社:下降 }, {A 社:上昇, B 社:上昇, C 社:上昇 }) といった時系列データは、ある営業日とその前日の営業日の株価が、A 社、B 社において上昇する一方、C 社において下降した翌日の営業日において、A 社、B 社、C 社の株価がともに上昇したことを示している。

上述した時系列データが多数あった場合に、その中に特徴的に出現する部分系列が、本論文での発見対象となる時系列パターンになる。なお、以下においては、時系列パターンの長さが n の場合に、n 次時系列パターンと呼ぶことにし、時系列パターンを構成する頻出アイテム集合に含まれるアイテム数が m の場合に、m 次頻出アイテム集合と呼ぶことにする。

2.2 制約パターンの定義

頻出する時系列パターンの場合、既知の時系列パターンが多数出力される傾向にあるため、発見された時系列パターンの中から分析者にとって興味のある時系列パターンを抽出することが必要となる。一方、興味のある時系列パターンはタスクに依存してはいるものの、分析者は時系列的な変化に興味があると考えられる。また、分析者はどのような属性値の変化に興味があるかをある程度は指定できると考えられる。そこで、このような考えの下、属性値の共通するアイテムの時系列的な変化に着目した制約条件として、制約パターンを導入した [櫻井 他 07]。本制約パターンは、一般に式 (1) のように記述することができる。

$$\begin{aligned} &(C_1, C_2, \cdots C_m), \\ &C_i = \{x_1 : a_{i1}, x_2 : a_{i2}, \cdots, x_n : a_{in}\} \end{aligned} \tag{1}$$

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:044-549-2406,
Fax:044-520-1268, E-mail:shigeaki.sakurai@toshiba.co.jp

ただし、x_j, $(j = 1, 2, \cdots, n)$ は任意の属性を表す変数とし、異なる x_j に対しては、異なる属性が割り当てられるとする。また、a_{jk} は x_j に対応する属性の属性値を表しているとする。本制約パターンによって、時系列パターンを構成する各頻出アイテム集合が、n 種類の異なる属性に対応する n 個の指定された属性値からなり、時系列パターンの長さが m となる時系列パターンだけを抽出することができる。このとき、$m = 1$ の場合には、制約アイテム集合と呼ぶことにする。

例えば、「A 社:上昇」、「A 社:下降」、「B 社:上昇」、「B 社:下降」といったアイテムが与えられており、(C_1, C_2), C_1={x_1 : 上昇, x_2 : 下降 }, C_2={x_1 : 上昇, x_2 : 下降 } が制約パターンとして与えられているとする。このとき、x_1, x_2 に対応する属性は {A 社, B 社 } であり、その属性値が { 上昇, 下降 } となる。このため、本制約条件によって、以下の 2 種類の系列だけが時系列パターンの候補として取り出されることになる。

({A 社:上昇, B 社:下降 }, {A 社:上昇, B 社:下降 })、
({B 社:上昇, A 社:下降 }, {B 社:上昇, A 社:下降 })

2.3 制約パターンの分割法

与えられた制約パターンを満たす時系列パターンだけを発見するために、時系列パターンの発見アルゴリズムにおいて、図 1 に示す分割アルゴリズムに従って、制約パターンを部分制約パターンに分解する。図 1 においては、制約パターンの集合 ConsDB を入力として与えることにより、i 次部分制約アイテム集合を $R_{1,i}$、k 次部分制約パターンを R_k に出力している。また、図 1 においては、calc_Size() を与えられた制約パターン (p) の長さ (ptlen) 及び制約パターンを構成するアイテム集合のアイテム数 (itnum) を計算する関数、divide_PtConst() を入力された部分制約パターンをふたつの部分制約パターンに分解する関数、get_PtConst() を入力された部分制約パターンと一致する部分制約パターンを取り出す関数、divide_ItConst() を入力された部分制約パターンをふたつの部分制約アイテム集合に分解する関数、check_ItConst() を入力された部分制約アイテム集合と一致する部分制約アイテム集合を取り出す関数とする。

2.4 制約パターンに基づいた発見法

時系列パターンをひとつの頻出するアイテムから開始して、順次時系列パターンを大きくしていくことにより、すべての時系列パターンを発見するアルゴリズム [Srikant and Agrawal 96] において、制約パターンを考慮する場合、候補頻出アイテム、候補頻出アイテム集合、候補時系列パターンを生成した直後に、対応する部分制約を評価することになる。このため、部分制約パターンを満たす候補に対してだけ、頻度計算が実施され、頻出するかどうかの判断が行われる。最終的には、頻出する時系列パターンのうち、制約パターンに一致する時系列パターンだけを結果として出力する。

```
//制約パターンの設定;
maxptlen = 1;
maxitnum = 1;
R_k = φ;
R_{1,i} = φ;
For each constraint pattern p ∈ ConsDB;
  calc_Size(p, &ptlen, &itnum);
  If ptlen == 1;
  Then add p to R_{ptlen};
       If maxptlen < ptlen;
       Then maxptlen = ptlen;
       If maxitnum < itnum;
       Then maxitnum = itnum;
  Else  add p to R_{1,itnum};
       If maxitnum < itnum;
       Then maxitnum = itnum;
//制約パターンの分解;
For(k = maxptlen; k > 1; k − −);
  For each constraint sub-pattern p ∈ R_k;
    divide_PtConst(p, k, &p_1, &p_2);
    If (r=get_PtConst(p_1, R_{k−1}))==φ;
    Then add p_1 to R_{k−1};
    If (r =get_PtConst(p_2, R_{k−1}))==φ;
    Then add p_2 to R_{k−1};
//1 次制約パターンの振り分け;
For each 1st constrain sub-pattern p ∈ R_1;
  calc_Size(p, &ptlen, &itnum);
  add p to R_{1,itnum};
//制約アイテム集合の分解;
For(i = maxitnum; i > 1; i − −);
  For each constraint item sub-set p ∈ R_{1,i};
    divide_ItConst(p, i, &p_1, &p_2);
    If (r=get_ItConst(p_1, R_{1,i−1}))==φ;
    Then add p_1 to R_{1,i−1};
    If (r=get_ItConst(p_2, R_{1,i−1}))==φ;
    Then add p_2 to R_{1,i−1};
```

図 1: 制約パターン分解アルゴリズム

3. 数値実験

3.1 実験データ

本論文では、http://homepage1.nifty.com/hdatelier/data.htm によって提供されている株価データを利用した数値実験を実施することにより、制約パターンに基づいた方法の効果を検証する。本データは 2005 年 9 月から 2007 年 9 月までの日々の株価に関連したデータであり、企業コード、営業日、始値、高値、安値、終値、出来高からなる 5 つの株価指標から構成されている。本データの特定のひとつの株価指標に対して、同一企業ごとにその営業日で並べることにより、各企業に対して 5 種類の時系列データを生成する。また、連続する営業日における各指標の値を比較し、その値が変化したレベルを決定することにより、レベル変化に基づいた時系列データを生成する。ただし、レベルの決定においては、前の結果で後の結果を割った値から 1 を引いた値である変化率に応じてレベルを決定する。すなわち、2d+1(d は自然数) 個のレベルに分ける場合には、$x < d\%$, $d\% \leq x < -(d-1)\%$, $\cdots$, $-2\% \leq x < -1\%$, $-1\% \leq x \leq 1\%$, $1\% < x \leq 2\%$, $\cdots$, $(d-1)\% < x \leq d\%$, $d\% < x$ に対応して、$0 \sim 2d$ のレベル変化を決定する。ただし、x が変化率を表しているとし、d は利用者によって与えられるレベルの個数を決定する値とする。株価データを用いた時系列パターン分析では、ある企業における指標の変化が他の企業における同一の指標の変化にどのように関係しているかに興味があるため、企業コードを属性、当該企業の特定指標のレベル変化を属性値とみなしたアイテムを生成する。例えば、ある企業 (企業コード $CODE1$) における始値の値が「100」→「100」→「98.5」と変化し、$d = 2$ と与えられているとする。このとき、変化率は 0.0, -1.5 と与えられるので、「$CODE1 : 2$」→「$CODE1 : 1$」といったレベル変化に基づいた時系列データが生成される。また、本レベル変化に基づいた時系列データは対象期間における長大な時系列データが、各指標ごとにひとつ生成されるだけである。本実験では、長大な時系列データの

中に含まれる時系列パターンを取り出すために、6 回のレベル変化を単位として、時系列データを取り出すたびに、4 回のレベル変化だけ進めることで、長大な時系列データから複数の時系列データを切り出すこととする。なお、実験で利用する時系列データにおいては、企業コード 1000 から 1500 までの企業の始値及び、JASDAQ、日経平均、TOPIX の始値に関する 28 種の時系列データとする。従って、時系列データを構成する各アイテム集合は 28 個のアイテムから構成されている。

3.2 実験方法

本論文では、制約条件として与える制約パターンの長さ、制約パターンの数、制約パターンに含まれるアイテムの数といった諸条件を変化させることにより、制約パターンの有効性をより詳細に検証する。制約パターンとしては、変化した値間の変化に着目したい一方、制約パターンの長さの違いや制約パターンに含まれるアイテムの数の違いを調査したいため、属性数が 2 の場合には系列長を 5 までとした場合のレベル変化に関する制約を導入する。また、属性数が 3 の場合には系列長を 3 までとした場合のレベル変化に関する制約を導入する。

また、本制約パターン集合を、対応する各データセットに適用した実験を実施する一方、制約パターンの数の違いを詳細に評価するために、制約パターン集合に含まれる制約パターンをひとつずつにして、時系列パターンを抽出することにする。この他、比較のために、制約パターンを利用せずに時系列パターンを抽出することにする。ただし、本時系列パターンの抽出においては、最小支持度として、0.01%, 0.5%, 1%, 2%, 3%, 5%, 10%, 20% といった値を利用することにし、制約パターンを利用しない場合には、アイテム集合に含まれるアイテム数を 2 あるいは 3 に制限することにする。

3.3 実験結果

実験結果の一部を図 2 に示す。各図におては、図 2(a) が始値を 5 分割した株価データに対して、制約パターンを個別に適用した場合とまとめて適用した場合における 2 次時系列パターンの抽出数の違いを示しており、図 2(b) が始値を 3 分割した株価データに対して、制約パターンを個別に適用した場合とまとめて適用した場合における 2 次時系列パターンの抽出数の違いを示しており、図 2(c) が始値を 5 分割した株価データに対して、適用する制約パターンの長さを変えた場合における時系列パターンの抽出数の違いを示しており、図 2(d) が始値を 5 分割した株価データに対して、アイテム集合内のアイテム数を 2 として、制約パターンを適用しない場合における時系列パターンの抽出数の違いを示している。また、実際に時系列パターンとして出力される時系列パターンの数が抽出数、より長い時系列パターンを生成するために必要となる時系列パターンを含めた時系列パターンの数が生成数として示されている。

各図においては、x 軸が最小支持度の値を示しており、y 軸が時系列パターンの数を示している。一方、図 2(a)、図 2(b) においては、a3l2.1e1∼6e1 が株価 (属性数:3, 系列長:2) の制約パターンをひとつずつ順に利用した場合の 1 次時系列パターンの抽出数を示しており、total がひとつずつ適用した場合の結果を合計した 1 次時系列パターンの抽出数を表している。図 2(c) においては、a2lx.0ex, (x=1∼5) が株価 (属性数:2, 系列長:1 ∼ 5) をそれぞれ適用した場合における抽出数を表している。図 2(d) においては、no.e1 ∼ 6 が制約無しで株価データから抽出される時系列パターンの系列長 1 ∼ 6 に応じた抽出数を表している。

3.4 考察

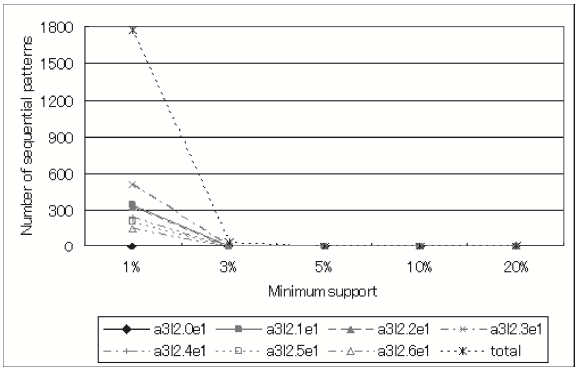
時系列パターンの数: 図 2(c) と図 2(d) との間の関係を比較することにより、制約パターンの導入が大幅な時系列パターンの削減に結びつくことが確認できる。また、制約パターンを利用しないアイテム数が 3 の場合の結果は示していないが、アイテム数が 3 の場合には、大部分の最小支持度において、途中段階で生成される時系列パターンの数が多く成り過ぎてしまい、利用している計算機環境の制限によって、すべての時系列パターンを発見できなくなっている。一方、制約パターンを利用した場合には、アイテム数が 3 の場合でも時系列パターンを抽出することが可能になっており、制約パターンを導入することにより、従来は解析が難しかった 3 要因間の関係を分析できるようになったといえる。

制約パターンの分割: 提案する制約パターンに基づいた時系列パターンの発見法では、複数の制約パターンを同時に指定することができる。各制約パターンは OR の関係にあるため、制約パターンごとに時系列パターンの抽出を試みたとしても最終的に抽出される時系列パターンは同じものとなる。しかしながら、すべての制約パターンを同時に指定した場合には、途中段階で生成される時系列パターンが多くなり過ぎるため、時系列パターンを抽出することができない。これに対して、個々に指定した場合には時系列パターンを抽出することが可能となっている。このため、利用する計算機環境に応じて、どのくらいの制約パターンを一度に指定するべきかを調整することが必要である。現在のところ、時系列パターンの抽出に失敗した段階で、適用する制約パターンの個数を順次少なくすることにより時系列パターンの抽出を試みているが、計算機環境に合わせた調整を自動的に行う枠組みも今後検討する必要があると考えられる。

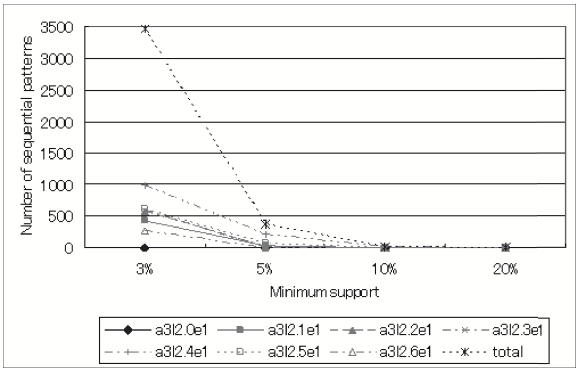
属性値数: 図 2(a)、図 2(b) に示すように、属性を構成する属性値数を増やすことにより、抽出される時系列パターンの数は少なくなっている。この現象は、属性値数を増やしたことにより、制約パターンの対象となる時系列データの数が減少したことにより、最小支持度を満たす時系列パターンの数が少なくなったためと考えられる。また、条件を満たす時系列パターンの数が少なくなったため、属性値数が少ない場合に、より小さい最小支持度となる時系列パターンを発見することが可能になっている。このように、属性値を適宜細かく設定し、設定した属性値に合わせた制約パターンを記述して、時系列パターンの抽出を実施するとすれば、制約パターンを利用することにより、従来は時系列パターンの生成に失敗していた時系列データからも、時系列パターンを発見できる可能性がある。

系列長: 図 2(c) に示すように、今回の実験では、時間軸方向のバリエーションをも考えて制約パターンを設定している。このため、系列長を 2 とした場合の制約パターン集合において抽出される時系列パターンが、最小支持度が小さい場合に多くなっている。しかしながら、個々に制約パターンを指定した場合には、時間軸方向に制約が追加されるため、最終的に抽出される時系列パターンの数は、系列長が長くなる程少なくなっている。多くの制約を満たす時系列データの数は、制約の数が増えるに従って急速に減少する傾向にあるため、今回の実験でも、最小支持度が大きくなるに従って、長い制約パターンを適用して抽出される時系列パターンの数は少なくなっている。アイテム集合に含まれるアイテムの数が 2 の場合においては、系列長が 4 になった段階で、変化に関連する時系列パターンが存在しないことを確認することができる。これに対して、制約を指定しない場合には、以降の長さに対しても時系列パターンの発見処理が実施されており、無駄な探索が行われている。この意味でも制約パターンの導入は効果的であると考えられる。

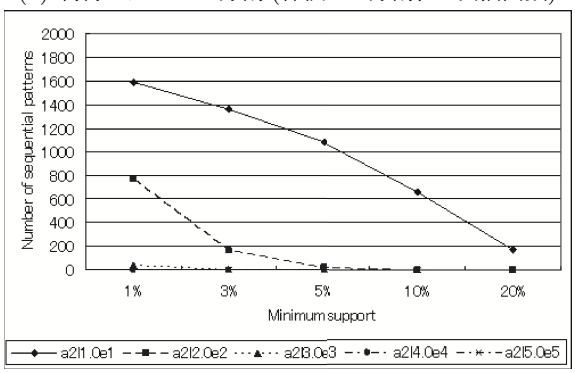
一方、制約パターンに基づいた方法では、予め系列の長さを



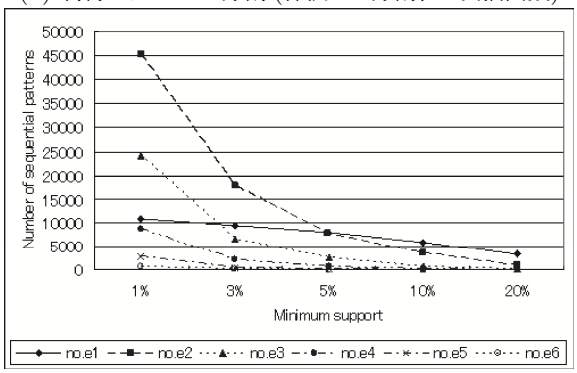
(a) 制約パターンの分割 (株価：5 分割、2 次抽出数)



(b) 制約パターンの分割 (株価：3 分割、2 次抽出数)



(c) 制約長変化 (株価：5 分割、抽出数)



(d) 制約長変化 (株価：5 分割、無制約、抽出数)

図 2: 実験結果

限定する必要があるといった欠点がある。しかしながら、系列長の異なる制約パターンを同時に指定することも可能であるため、制約パターンを上手く記述することにより、このような欠点をある程度回避することができると考えられる。

時系列パターンの妥当性: 今回の実験では、属性間の間にその値の変化に関連性があると考えられる時系列パターンを抽出するべく、制約パターンを指定している。本制約は、株価データにおいては、妥当な制約であり、抽出される時系列パターンに対する利用者の興味が高いものと推察される。しかしながら、抽出された時系列パターンの詳細な中身の検証は現在のところ行われていないため、妥当性をより詳細に検証するには、利用者による本時系列パターンの確認が必要と考えられる。

一方、他の時系列データを対象とした場合には、このような変化以外に興味があることも考えられる。どのような制約パターンを設定するかどうかは、タスクに依存したものになるものの、タスクに応じた制約パターンの集合をある程度準備することにより、利用者の分析における負荷を一層小さくすることができる。このため、タスクに応じた制約パターンに関しても、今後検討する必要がある。

上記の考察に従って、制約パターンに基づいた時系列パターンの発見法は、効率的に利用者にとって興味のある時系列パターンを発見することができると考えられる。

4. まとめと今後の課題

今回の論文では、提案している制約パターンに基づいた時系列パターンの発見法の効果をより詳細に検証するために、制約パターンにおける、系列の長さ、制約パターンの数、制約パターンに含まれるアイテムの数といったパラメータ相当の部分を変化させて、その効果を検証した。株価データといった一部の実データセットではあるが、本手法の効果を示すことができたと考えている。

今後の課題としては、考察のところでも記載しているが、同時に指定する制約パターンの数の自動調整、抽出された時系列パターンの利用者による妥当性の検証、タスクに応じた制約パターンの準備といったものが、まずは考えられる。また、より多様な時系列データを分析対象とするために、数値データと離散データをシームレスに扱う手法の検討、欠損値の扱いなども検討していきたいと考えている。

参考文献

[Garofalakis et al 99] M. N. Garofalakis, R. Rastogi and K. Shim: “SPIRIT: Sequential Pattern Mining with Regular Expression Constraints”, Proc. of the Very Large Data Bases Conf., 223-234 (1999).

[Pei et al 02] J. Pei, J. Han and W. Wang: “Mining Sequential Patterns with Constraints in Large Databases”, Proc. of the 11th ACM Int. Conf. on Information and Knowledge Management, 18-25 (2002).

[Srikant and Agrawal 96] R. Srikant and R. Agrawal, “Mining Sequential Patterns: Generalizations and Performance Improvements”, Proc. of the 5th Int. Conf. Extending Database Technology, 3-17 (1996).

[櫻井 他 07] 櫻井 茂明, 北原 洋一, 折原 良平: 「制約パターンに基づいた時系列パターンの発見法」, 第 66 回人工知能基本問題研究会, 35-40 (2007).