D-003

関係的相関ルール導出のための事例の性質の抽出

Construction of properties of examples for relational association rules mining

元山 純一[*] **Jun-ichi Motoyama**
中野 智文[†] **Tomofumi Nakano**
犬塚 信博[*] **Nobuhiro Inuzuka**

1．はじめに

データマイニングとは，大量のデータから隠された知識や新しい規則を発見するプロセスである．

複数の関係表からなるデータベースを扱う手法として，**帰納論理プログラミング**(Inductive Logic Programming : ILP) が注目される．これは論理的な記述によって高い可読性を持ち，データマイニングの有力な手法と考えられている．

ILP の枠組みで関係的知識のデータマイニングを行う手法がいくつか提案されてきた．その中の一つに WARMR がある [1, 2]．これは，関係データベース内のユーザの着目点であるキー (key) と呼ばれる述語とそれ以外の複数の述語で形成されたクエリーを生成し，それらがデータベース内で満たされるかを調べる．そしてクエリーの中でよく満たされるものを取り出し，相関ルールを生成する手法である．このとき，候補となるクエリーは言語バイアスであるモードやタイプを使った候補の制限や，論理的冗長性を使った制限を行っている．しかしながら，それでも生成される候補は多大にあり，実際にはありえないようなクエリーを含むかもしれない．

本論文では ILP の枠組みで行う新しい手法として，事例から性質を抽出してこれをデータマイニングに用いる MAPIX(Mining Algorithm by Property Item eXtraction) アルゴリズムを提案する．これは引数のモードなどで制限を行いながら，事例に関係する事実を取り出し，それらをアイテムとして APRIORI アルゴリズム [3, 4] と同様の方法で興味深いルールを導出する手法である．本手法では事例に関する事実を取り出すときに，すべての事例を使用しなくても全体を反映したルール導出を行えることを実験で示した．

2．方法のアイディア

簡単な例として図 1 の家族関係について考える．これは，親子の関係を表す parent と性別を表す male，female で構成されている．*が付けられた名前は female を満足するものである．ここで関係 grandfather に注目する．注目する関係は，その定義が未知である場合と，定義は既知でもその具体例に潜む性質に興味がある場合を考える．こうした注目する関係の例が表 1 のように正/負事例として与えられているとする．

grandfather の一つの例である `koji` に注目すると，`koji` について例えば以下のような事実が成り立つ．

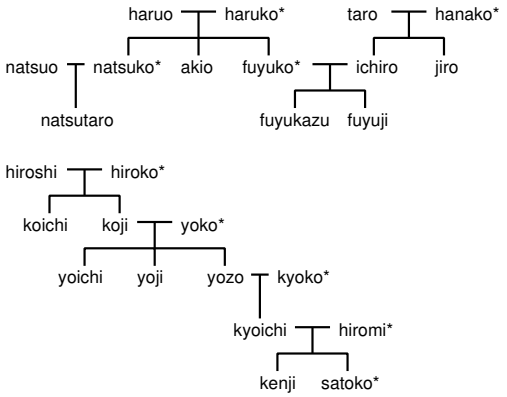


図 1　家族関係

表 1　grand-father についての正事例と負事例

正事例	負事例
grandfather	grandfather
haruo	haruko
hiroshi	natsuo
koji	natsuko
yozo	akio
taro	fuyuko
	. . .

$$\mathsf{parent}(\mathtt{koji},\ \mathtt{yoji}) \wedge \mathsf{male}(\mathtt{yoji}). \quad (1)$$

$$\mathsf{parent}(\mathtt{koji},\ \mathtt{yozo}) \wedge \mathsf{parent}(\mathtt{yozo},\ \mathtt{kyoichi}) \wedge \mathsf{male}(\mathtt{kyoichi}). \quad (2)$$

式 (1) は `koji` は `yoji` という息子を持つこと，(2) は `kyoichi` という孫を持つことを表している．このような，対象 `koji` がもつ事実を表わす述語の組を `koji` の**性質**と考えることができる．こうした性質を事例から抽出し，これらの性質についてのルールマイニングを行いたい．

3．準備

性質を定義するために述語のモードを導入する．モードとは，述語の各引数について各々が入力引数 (入力モード) なのか出力引数 (出力モード) なのかを表し，各々 $+$，$-$ で表わす．例えば家族の例では，それぞれの述語のモードは $\mathsf{parent}(+,\ -)$，

[*] 名古屋工業大学 大学院 工学研究科 情報工学専攻
[†] 名古屋工業大学 情報メディア教育センター

male(+)，female(+) と仮定する．また，述語 q の各引数のモードを与える関数 (**モード関数**) を mode_q と書くこととし，$\mathrm{mode}_q(i)$ で述語 q の i 番目の引数のモードを表わす．例えば，$\mathrm{mode}_{\mathsf{parent}}(1) = +$ ，$\mathrm{mode}_{\mathsf{parent}}(2) = -$ である．

モードに注目すると，述語は二つのクラスに分けられる．第 1 のクラスは，すべての引数が入力モードである述語のクラスで**判定述語**(check predicate) という．この述語はすべての引数に値の定まった形で呼ばれ，その真偽を決定する．もう一つのクラスの述語は，入力モードと出力モードを持つ述語のクラスで**経路述語**(path predicate) という．これは，入力モードである引数の値が束縛されて呼びだされ，出力モードの引数に値を返す関数的な使い方をする．

例えば male，female は判定述語，parent は経路述語である．また各述語からなるリテラルをそれぞれ**判定リテラル**(check literal)，**経路リテラル**(path literal) という．

ここで，先ほどの例 parent(koji，yozo) ∧ parent(yozo, kyoichi) ∧ male(kyoichi) は，引数 koji を用いて parent から yozo という値が呼び出され，次の parent に渡している．そして，同じように parent から kyoichi が呼び出され，次の male に渡している．最後に male に kyoichi という値を入れることで一つの事実を表している．このように 1 つの判定リテラルといくつかの経路リテラルで構成され，引数が注目する対象から判定述語の引数まで鎖状につながっている述語の組が性質である．

4．性質の抽出

与えられた事例，例えば koji の飽和節を与える．飽和節 [5, 6] とは，ある目標の述語についてのすべての説明し得る背景知識を本体に加えたものである．grandfather(koji) の飽和節は以下のようになる．

grandfather(koji) ← parent(koji, yozo) ∧ male(koji)∧
parent(koji, yoichi) ∧ parent(koji, yoji) ∧
male(yoichi) ∧ male(yoji) ∧ male(yozo) ∧
male(kyoichi) ∧ parent(yozo, kyoichi) ∧
parent(kyoichi, satoko) ∧ parent(kyoichi, kenji) ∧
male(kenji) ∧ female(satoko).

この中には，前節で見た二つの性質が含まれている．本手法では，飽和節を利用して事例から性質を抽出することを考える．

定義　事例 $e = p(t_1, \cdots, t_m)$ と e の飽和節の本体リテラルの集合 S_e に対し，性質候補および性質を以下のように定義する．

- 判定リテラル $c = q(t^c_1, \cdots, t^c_{m_c}) \in S_e$ に対し，$\mathsf{prop} = \{c\}$ は未解決項 $U = \{t^c_1, \cdots, t^c_{m_c}\} \backslash \{t_1, \cdots, t_m\}$ と解決項 $S = \{t_1, \cdots, t_m\}$ を持つ**性質候補**である．
- 解決項 S と未解決項 U を持つ性質候補 prop と，経路リテラル $d \in S_e$ に対し，d の入力引数の集合を I_d，出力引数の集合を O_d としたとする．このとき $U \cap O_d \neq \emptyset$ であれば，$\mathsf{prop} \cup \{d\}$ は未解決項 $(U \backslash O_d) \cup (I_d \backslash S)$ と解決項 $S \cup O_d$ を持つ**性質候補**である．
- 未解決項 $U = \emptyset$ を持つ性質候補を事例 e の**性質**という．

□

この定義にしたがって，事例からすべての性質を取り出す PIX(Property Item eXtraction) アルゴリズムを表 2 に示す．

例えば，先ほどの grandfather(koji) についての飽和節からは，7 つの性質が取り出される．

{ male(koji) }
{ parent(koji, yoichi), male(yoichi) }
{ parent(koji, yoji), male(yoji) }
{ parent(koji, yozo), male(yozo) }
{ parent(koji, yozo), parent(yozo, kyoichi),
male(kyoichi) }
{ parent(koji, yozo), parent(yozo, kyoichi),
parent(kyoichi, satoko), female(satoko) }
{ parent(koji, yozo), parent(yozo, kyoichi),
parent(kyoichi, kenji), male(kenji) }

性質アイテム

以上のように作られた事例 e の性質 prop に対し，節 $c = e' \leftarrow \mathsf{prop}$ を考える．ここで e' は e と同じ引数と新しい述語名を持つ述語である．さらに c に現れる，すべての項を別の変数に置き換えたものを**性質節** pclause という．例えば $e =$ grandfather(koji) の性質 (1), (2) から得られる性質節は，

$\mathsf{item1}(A) \leftarrow \mathsf{parent}(A,\ B) \wedge \mathsf{male}(B).$
$\mathsf{item2}(A) \leftarrow \mathsf{parent}(A,\ B) \wedge \mathsf{parent}(B,\ C) \wedge \mathsf{male}(C).$

となる．さらに，この性質節の頭部リテラルの述語名（すなわち，新しく用意した述語名）を**性質アイテム**という．ここで性質アイテム item に対し，$\mathsf{item}(e)$ によって item の引数を e と同じものにしたものを表わすことにする．例えば，$\mathsf{item1}(e) = \mathsf{item}(\mathsf{grandfather}(\mathtt{koji})) = \mathsf{item1}(\mathtt{koji})$，$\mathsf{item2}(e) = \mathsf{item2}(\mathtt{koji})$ である．

このとき背景知識 $\mathcal{B}$ について，$\mathcal{B} \cup \{\mathsf{pclause}\} \models \mathsf{item}(e)$ を満たすとき，e は **性質アイテム** item **を持つ**という．また，性質アイテムの集合を**性質アイテムセット**という．例えば，

$$\mathcal{B} \cup \{\mathsf{item1}(A) \leftarrow \mathsf{parent}(A,\ B) \wedge \mathsf{male}(B)\} \models \mathsf{item1}(e) = \mathsf{item1}(\mathtt{koji})$$

である．ここで $\mathcal{B}$ は parent(koji,yoji)，male(yoji) を含んでいる図 1 を表わす背景知識とする．

また，性質アイテムの集合 I とそれらに対応する性質節 C について，事例 e が持つ性質アイテムの集合は次のように表現でき，t_e と表わす．

$$t_e = \{\mathsf{item} \in I \mid B \cup C \models \mathsf{item}(e)\} \subseteq I$$

性質アイテムを用いたデータマイニング

性質アイテムの集合 $\{\mathsf{item1},\ \cdots,\ \mathsf{item}n\}$ を使って，相関ルール $R =$ ‘$\mathsf{item}s_1,\ \cdots,\ \mathsf{item}s_m \Longrightarrow \mathsf{positive}$’. を導出する．この規則は，性質アイテム ‘$\mathsf{item}s_1,\ \cdots,\ \mathsf{item}s_m$’ を持てば正事例であるということを意味する．

PIX(e):

input e : 正事例 $e = p(t_1, \cdots, t_k)$;
output I : e から取り出された性質アイテムの集合;
C : I に対応した性質節の集合;

1. $I := \emptyset$; $C := \emptyset$; $P := \emptyset$;
2. $S_e :=$ e の飽和節の本体リテラルの集合;
3. $T_{\rm head} := \{t_1, \cdots, t_k\}$;
4. $C := \{\ell \in S_e \mid \ell$ は判定リテラル $\}$; $P := \{\ell \in S_e \mid \ell$ は経路リテラル $\}$;
5. **For each** $\ell \in C$ **do**
6. prop $:= \{\ell\}$;
7. $U_{\rm now} := \{t_i \mid \ell = q(t_1, \cdots, t_k) \wedge\ m_q(i) =$ '+'$\}$;
8. $P' := P$;
9. **While** $U_{\rm now} \backslash T_{\rm head} \neq \emptyset$ **do**
10. $ll = \{q(t_1, \cdots, ts) \in P' \mid \exists i \in \{1, \cdots, s\}, {\rm mode}_q(i) =$ '$-$' $\wedge\ t_i \in U_{\rm now} \backslash T_{\rm head}\}$;
11. prop $:=$ prop $\cup\ ll$;
12. $P' := P' \backslash$ prop;
13. $U_{\rm now} := \{t_i \mid q(t_1, \cdots, t_k) \in ll\ \wedge\ m_q(i) =$ '+'$\}$;
14 pclause $:= e' \leftarrow$ prop, ここでの e' は，e の述語名を新しい名前に置き換える;
15 pclause のすべての項を別の変数に置き換える;
16 item $:=$ pclause の頭部の述語名;
17. $I := I \cup \{$item$\}$; $C := C \cup \{$pclause$\}$;
18. **return** (C, I);

表 2　PIX アルゴリズム，事例からの性質の抽出

興味深いルールの導出を行うために，相関ルール導出で使用される支持度と確信度を使って生成されたアイテムセットの評価を行う．性質アイテムの集合と対応する性質節を C とすると，ルール $R \leftarrow {\rm item}r_1 \wedge \cdots \wedge {\rm item}r_x$ の支持度 ${\rm sup}(R)$，確信度 ${\rm conf}(R)$ は以下のように表される．

$$
{\rm sup}(R) = \frac{|\{e \in E^+ \cup E^- \mid B \cup C \models {\rm item}r_1(e) \wedge \cdots \wedge {\rm item}r_x(e)\}|}{|E^+ \cup E^-|}
$$

$$
{\rm conf}(R) = \frac{|\{e \in E^+ \mid B \cup C \models {\rm item}r_1(e) \wedge \cdots \wedge {\rm item}r_x(e)\}|}{|\{e \in E^+ \cup E^- \mid B \cup C \models {\rm item}r_1(e) \wedge \cdots \wedge {\rm item}r_x(e)\}|}
$$

ここで，${\rm sup}(R)$ は標準的な相関ルール導出における支持度と異なる．これは，本来の支持度の意味である全体に対する割合という評価値とするために変更を加えた．

マイニングのタスクを以下のように定式化する．

ルール導出手法：

Given E^+/E^- : 正/負事例
$\mathcal{B}$: 背景知識，関係データベース
${\rm sup}_{\rm min}$: 最低支持度
${\rm conf}_{\rm min}$: 最低確信度

Enumerate すべてのルール '$I' \Longrightarrow$ positive'

s.t. ${\rm sup}(I' \Longrightarrow {\rm positive}) \geq {\rm sup}_{\rm min}$
${\rm conf}(I' \Longrightarrow {\rm positive}) \geq {\rm conf}_{\rm min}$
ここで $I' \subseteq I$ と I は性質アイテムの集合

5．アルゴリズム

以上の原理を用いたアルゴリズムの概要を次に与える．

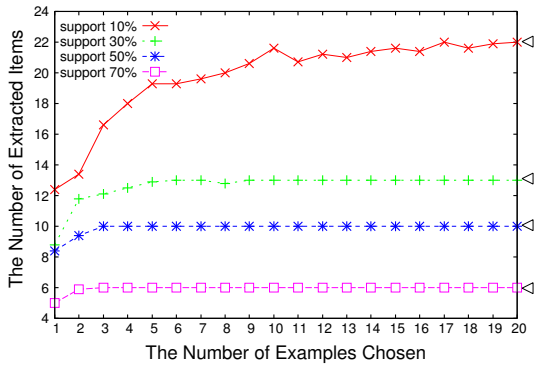
1. 与えられた正事例の集合からいくつかの事例を選択する．
2. PIX アルゴリズムを使って，選んだ正事例から性質アイテムを抽出する．
3. すべての性質アイテムを使って，興味深いルールを枚挙する．

ステップ 3 は APRIORI アルゴリズムと同様の方法を利用できる．APRIORI は頻出アイテムセットを求めた後，興味深い相関ルールを求めるのと同じように，ここでは頻出の性質アイテムセットを求める．

ただし，ここでは 100% の支持度を持つ性質アイテムを除いてアイテムセットの生成をする．100% の支持度を持つアイテムは，正負事例問わずに現れるため，興味深いアイテムとは言えない．マーケットデータベースでは支持度 100% のアイテムは考えられないが性質アイテムでは恒真な条件において 100% となりよく現れる．また，100% のアイテムを除くことによってアイテムセットの計算量を減らす効果も大きい．

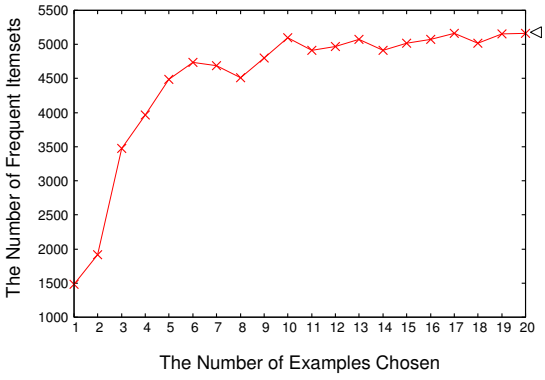
6．実験

前節の MAPIX アルゴリズムを SWI-prolog で実装し，これを用いてアルゴリズムの評価を行う．実験データには，East-West Challenge[7] を用いて，120 台の貨物列車のうち 57 台の正事例と 63 台の負事例を使用し，同じ条件で 20 回実験を

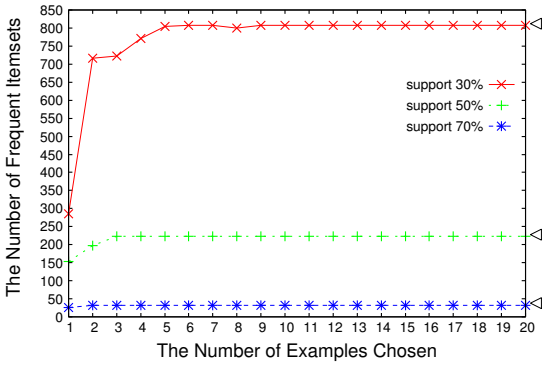


◁ は，全正事例を使用したときのアイテム数

図 2　抽出に使用した事例に対するアイテム数



(a) 支持度が 10% のとき



(b) 支持度が 30%, 50%, 70% のとき

◁ は，全正事例を使用したときのアイテムセット数

図 3　抽出に使用した事例に対するアイテムセット数

行いそれぞれ平均を実験値とした．

この実験の目的は，性質アイテムを取り出すためにすべての事例を用いなくてもすべての可能な興味深い規則の内，十分な割合が導出できることを確認することである．

性質を取り出す正事例の数を横軸とした頻出性質アイテムの数を図 2 に示す．この実験結果は，すべての性質アイテムを取り出すために必要な事例の数を考察するためのものである．図から，支持度 10% , 30% , 50% , 70% において，それぞれ 10 個，5 個，3 個，2 個の事例から性質を取り出すことですべての事例から取り出した性質アイテムの数の内，平均において 90% 以上の性質アイテムが取り出された．

また，同様に頻出性質アイテムセットの数についての実験結果を図 3 に示す．図から，支持度 10% , 30% , 50% , 70% において，それぞれ 20 個，9 個，4 個，2 個の事例から性質を取り出すことですべての事例から取り出した性質アイテムセットの数の内，平均において 90% 以上の性質アイテムセットが取り出された．

実験は Pentium-4 1.8GHz の CPU と 512MB のメインメモリを持つ Unix 上で行った．最も多くの頻出性質アイテムセットが生成される支持度が 10% で利用する正事例が 20 個のときの CPU 時間は約 14.8 秒であった．

7．おわりに

本論文は，関係データベースからの相関ルールの導出方法として，事例から性質を抽出する方法を提案した．データマイニングアルゴリズム MAPIX は，いくつかの事例から性質と呼ばれるものを取り出し，APRIORI と類似の方法によって興味深いルールのみを生成する．すべての事例から性質アイテムを取り出さなくても，一部の正事例のみから抽出した性質を用いて，すべての事例から生成されるルールの 90% 以上のルールを導出できることがわかる．よって，MAPIX アルゴリズムはデータベース全体を反映した数のルールを導出していると考える．

参考文献

[1] L. Dehaspe, L. De Raedt. “Mining association rules with multiple relations”, ILP-97, LNAI1297,pp.125–132, 1997.

[2] L. Dehaspe, H. Toivonen :“Discovery of Relational Association Rules”, in Relational Data Mining, pp. 189–212, Springer-Verlag, 2001.

[3] R. Agrawal, T. Imielinski N. A. Swami :“Mining association rules between sets of items in large database”, Proc. SIGMOD, pp. 207–216. ACM, 1993.

[4] R. Agrawal, R. Srikant : “Fast Algorithms for Mining Association Rules”, Proc. VLDB, pp. 487–499, 1994.

[5] P. Idestam-Almquist : “Efficient Induction of Recursive Definitions by Structural Analysis of Saturations”, in Advances in ILP, L. De Raedt (ed.), IOS Press Ohmsha, pp. 192–205, 1996

[6] M. Furusawa, N. Inuzuka, H. Seki, H, Itoh : “Induction of Logic Programs with More Than One Recursive Clause by Analysing Saturations”, ILP-97, pp. 165–172, LNAI 1297, Springer, 1997.

[7] “East-West Challenge”, ftp://ftp.mlnet.org/ml-archive/ILP/public/data/east_west/, at MLnet: Machine Learning Online Information Service, 2000.