データマイニング

第４回授業

＜相関ルール抽出問題＞ （復習）

【相関ルール抽出問題】
・確信度の閾値（最小確信度）
・サポートの閾値（最小サポート）
の２つを入力したとき，それぞれの閾値以上の確信度・サポートを有する相関ルールを全て発見せよ.

条件を満たす相関ルールの抽出問題･･･下記の２つの部分問題に分解された.
(1) 頻出アイテム集合を全て見出し，サポートを求める（アプリオリ・アルゴリズム）
(2) 上記(1)で求めた頻出アイテム集合を使って，最小確信度以上の相関ルールを求める

・ 相関ルールの導出

これまでは相関ルール抽出問題の（１）の段階の処理について説明した.
引き続き，（２）の段階の頻出アイテム集合を使った最小確信度以上の相関ルールの導出法を紹介する.
まず，効率の良い導出の為に，以下で最小確信度に関する性質を導く.

既習の通り，アイテム集合 a の任意の部分集合 $\tilde{a} \subset a$ について，必ず

support($\tilde{a}$)≧support(a)　　　　(i)

が成り立つ.

＜具体例＞

$a = \{$牛乳，パン，チョコレート$\}$
$\tilde{a} = \{$牛乳$\}$　　　(記号～はチルダまたはチルドと読みます)

とすると,

$\tilde{a} \subset a$

が成り立っている.
このとき,「少なくとも牛乳を買う客」の数のほうが,「少なくとも牛乳とパンとチョコレートの３つを買う客」より明らかに多い.
よって,

「全部の客に対する，少なくとも牛乳を買う客の割合」

≧ 「全部の客に対する，少なくとも牛乳とパンとチョコレートの 3 つを買う客の割合」

が成り立つ．この不等式が(i)式に対応する．

ここで，lを頻出アイテム集合の１つとし， $a \subset l$ とする．

$a \Rightarrow (l - a)$

の確信度 conf($a \Rightarrow (l - a)$) は

conf($a \Rightarrow (l - a)$) = support(l) / support(a) (ii)

となる．

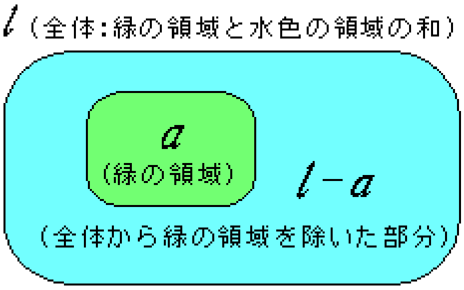


図１： 集合l，a，$l - a$の関係

<(ii)式の成立の理由>

第 3 回授業より，conf(X⇒Y) = support(X∪Y) / support(X)だった．

この式に X としてaを，Y として$(l - a)$を当てはめて考えたとき，

conf($a \Rightarrow (l - a)$) = support($a \cup (l - a)$) / support(a)

ここで，

$a \cup (l - a) = l$

となるから，(ii)式

conf($a \Rightarrow (l - a)$) = support(l) / support(a)

が成り立つ．

(ii)式で表される確信度 conf($a \Rightarrow (l - a)$) = support(l) / **support(a)** は，(i)より support($\tilde{a}$)≧ support(a)なので，

$\tilde{a} \Rightarrow (l - \tilde{a})$

の確信度

conf($\tilde{a} \Rightarrow (l - \tilde{a})$) = support(l) / **support($\tilde{a}$)** (iii)

と等しいか，あるいは大きい．

(↑(ii), (iii)両式の右辺の分母に(i)を適用して考えてみると，明らかに成り立つことがわかる)

すなわち，

「$a \Rightarrow (l - a)$ 」の確信度 ≧ 「$\tilde{a} \Rightarrow (l - \tilde{a})$」の確信度 ・・・ ☆

となる．

<具体例>

$l=\{$牛乳，パン，チョコレート，カップラーメン$\}$

$a=\{$牛乳，パン$\}$

$\tilde{a}=\{$牛乳$\}$

とすると，

$\tilde{a}\subset a\subset l$

が成り立っている．

このとき，$a \Rightarrow (l-a)$ の確信度とは，$(l-a)=\{$チョコレート，カップラーメン$\}$ なので，

「少なくとも牛乳とパンを買う客が，どれだけチョコレートとカップラーメンも一緒に買うか」

を示す．

いっぽう，$\tilde{a} \Rightarrow (l-\tilde{a})$ の確信度とは，$(l-\tilde{a})=\{$パン，チョコレート，カップラーメン$\}$ なので，

「少なくとも牛乳を買う客が，どれだけパンとチョコレートとカップラーメンも一緒に買うか」

を示す．

上述の黄色の☆で示される関係を適用すると，

「少なくとも牛乳とパンを買う客のうち，チョコレートとカップラーメンも一緒に買う客の割合」

のほうが，

「少なくとも牛乳を買う客のうち，パンとチョコレートとカップラーメンも一緒に買う客の割合」

より**大きい**ことになる．

上記より，次の命題が成り立つ．

もし

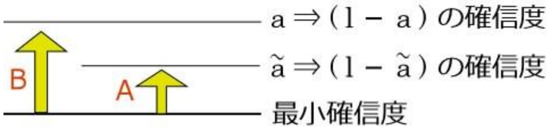
$\tilde{a} \Rightarrow (l - \tilde{a})$ の確信度 ≧ 最小確信度

ならば，

$a \Rightarrow (l - a)$ の確信度 ≧ 最小確信度

となる． （※）

上記は，確信度の低いほうのルールが最小確信度以上であれば，確信度の高いほうのルールも必然的に最小確信度以上になることを示す．



「AならばB」が成り立つ

この(※)の命題の対偶をとる（「A ならば B」の対偶は「not B ならば not A」．ある命題が真であれば，その対偶も真になる）と，次のようになる．

もし

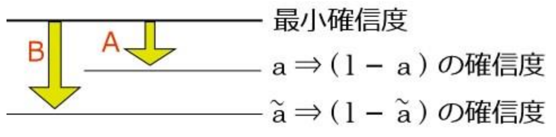
$a \Rightarrow (l - a)$ の確信度 ≦ 最小確信度

ならば,

$\tilde{a} \Rightarrow (l - \tilde{a})$ の確信度 ≦ 最小確信度

となる.　　　　　　　　　　　　　　　　（※※）

これは確信度の高いほうのルールが最小確信度以下であれば，確信度の低いほうのルールも最小確信度以下になることを示す.



「AならばB」が成り立つ

ここで,

$c = l - \tilde{a}$

$\tilde{c} = l - a$

のようにおく.

このとき $\tilde{c} \subset c$ が成り立つ（図 2 参照).

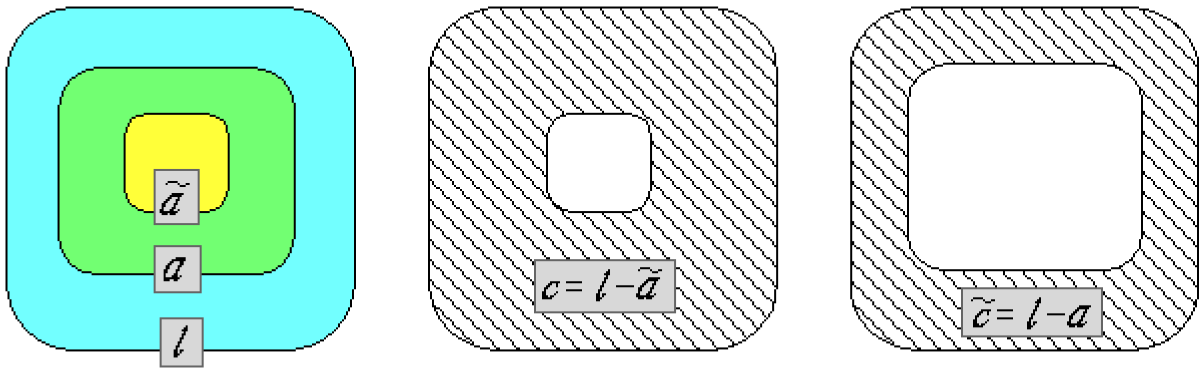


図 2： $l - a \subset l - \tilde{a}$ の関係の図解

上で定義した c，$\tilde{c}$ を用いると,

$a = l - \tilde{c}$

$\tilde{a} = l - c$

$l - a = \tilde{c}$

$l - \tilde{a} = c$

となるので，上述の命題（※※）は次のように表せる.

もし

$(l - \tilde{c}) \Rightarrow \tilde{c}$ の確信度 ≦ 最小確信度

ならば,

$(l - c) \Rightarrow c$ の確信度 ≦ 最小確信度

となる.　　　　　　　　　　　　　　　　（※※※）

以上より，**あるルールが最小確信度未満であれば，そのルールの帰結部の集合を包含するような集合を帰結部に持つ一連のルールも必ず最小確信度未満になる**ので，最小確信度以上のルールの導出の際にはそれらを却下できる．

> <具体例>
>
> 上の(※※※)の命題にて，l = {A, B, C, D} ，c = {C, D}，$\tilde{c}$ = {D} $\subset$ c とすると，
>
> {A, B, C} ⇒ {D} の確信度が最小確信度未満
>
> であるとき，
>
> {A, B} ⇒ {C, D} の確信度は最小確信度未満
>
> となる．

上記の性質を利用して最小確信度(*minconf*)以上の確信度を持つ相関ルールを効率良く導出するアルゴリズムを以下に紹介する．

0) Algorithm Generate-Rules() {

1) foreach 頻出アイテム集合 l_k （要素数 k ＞2） {

2) H_1 := { { h ∈ l_k } | conf((l_k−{h}) ⇒ {h}) ≧ *minconf* };

// ↑ H_m (ここでは m=1) は最小確信度以上の相関ルールを作れる要素数 m の結論部の「集合」．

// h は l_k に含まれる 1 個のアイテム．

3) call AP-GenRule(l_k , H_1);

4) }

5) }

6) Procedure AP-GenRule(Itemset l_k , 結論部の集合 H_m) {

7) if (k > m +1) {

8) H_{m+1} = AprioriGen(H_m);

// ↑ H_m から要素数の 1 つ多い H_{m+1} を作り出す．前回授業で既出．

9) foreach h_{m+1} ∈ H_{m+1} {

// ↑ H_{m+1} に含まれる，要素数 m+1 のあらゆる部分集合 h_{m+1} について以下を実行．

10) *conf* = support(l_k) / support($l_k - h_{m+1}$);

// ↑ 相関ルール ($l_k - h_{m+1}$) ⇒ h_{m+1} の確信度を計算する．

11) if (*conf* ≧ *minconf*)

// ↑ 確信度が最小確信度以上か？

12) output ($l_k - h_{m+1}$) ⇒ h_{m+1} ;

// ↑ 相関ルール($l_k - h_{m+1}$) ⇒ h_{m+1} は条件を満たすので，このルールを出力．

else

$H_{m+1} = H_{m+1} - \{h_{m+1}\}$;

// ↑ 最小確信度以上にはならなかったから，h_{m+1} を候補から除く.

}

AP-GenRule(l_k , H_{m+1});

// ↑ 再帰呼び出しで，さらに要素数の多い結論部分について吟味.

}

}

＜処理の流れ＞

頻出アイテム集合 *l* について，まず要素数 1 の結論部分を持つ相関ルールを作成する.

次に，確信度が最小確信度より小さくなかった結論部分の集合に対して，AprioriGen()関数を適用して，要素数 2 の結論部分を生成する.

以上を繰り返して，徐々に大きな結論部分を有する相関ルールを求めてゆく.

実行例

第 3 回授業で扱った下記のデータベースより頻出アイテム集合が求まっているとする.

データベースD	
ＴＩＤ	アイテム
0001	A, C, D
0002	B, C, E
0003	A, B, C, E
0004	B, E

＜上記データベースより求まる頻出アイテム集合(最小サポート 50%)＞

{{A}, {B}, {C}, {E}, {A, C}, {B, C}, {B, E}, {C, E}, {B, C, E}}

これら頻出アイテム集合に対して，本ページにて説明してある相関ルール導出アルゴリズム“Generate-Rules()”を適用する.

要素数 2 より大きい各頻出アイテム集合 I_k に対して，H_1 を生成しつつ，手続き“AP-GenRule”を呼び出す.

ここでは I_3 の(唯一の)要素である{B, C, E}に対して上記ルールを適用してみる.

H_1={{B}, {C}, {E}}.

＜確認＞

conf({C, E}⇒{B}) = support{B, C, E} / support{C, E} = (2/4)/(2/4) = 1 > 0.5

conf({B, E}⇒{C}) = support{B, C, E} / support{B, E} = (2/4)/(3/4) = 2/3 > 0.5

conf({B, C}⇒{E}) = support{B, C, E} / support{B, C} = (2/4)/(2/4) = 1 > 0.5

よって，{B}, {C}, {E}は H_1 の要素になれる.

上記 H_1 より，AP-GenRule を呼び,

H_2={{B, C}, {B, E}, {C, E}}

が得られる．ここで，最小確信度は 0.5 とする．

以下，H_2 の各要素に対してルール($l_k - h_{m+1}$) ⇒ h_{m+1} を吟味する．

・h_2={B, C}に対して：

conf({E}⇒{B, C}) = support{B, C, E}/support{E} = (2/4) / (3/4) = 2/3 > 0.5

∴ {E} ⇒ {B, C} は求めるべきルールに含められる．　　　(i)

・h_2={B, E}に対して：

conf({C}⇒{B, E}) = support{B, C, E}/support{C} = (2/4) / (3/4) = 2/3 > 0.5

∴ {C} ⇒ {B, E} は求めるべきルールに含められる．　　　(ii)

・h_2={C, E}に対して：

conf({B}⇒{C, E}) = support{B, C, E}/support{B} = (2/4) / (3/4) = 2/3 > 0.5

∴ {B} ⇒ {C, E} は求めるべきルールに含められる．　　　(iii)

再度，AP-GenRule を呼ぶと，k=3 に対して m+1 も 3 になり，となり，条件の k > m+1 を満たさなくなるので，ここで終了.

よって，(i), (ii), (iii)が求まるルールとなる.

【以上】

～～～～～～～～～～

発展版として，アイテム同士の間に階層関係が成り立つ場合の相関ルール「一般化相関ルール」も提案されている.