情報意味論（13） 相関規則

櫻井彰人
慶應義塾大学理工学部

本日の予定

- 相関規則
- 相関規則発見のアルゴリズム
  - large/frequent item set（頻出アイテム集合）
  - support（支持度）
  - confidence（信頼度）

相関規則(association rule)

- R. Agrawal, T. Imielinski, and A. Swami, Mining Association Rules between Sets of Items in Large Databases, SIGMOD Conference 1993: 207-216.
- R. Agrawal and R. Srikant , Fast Algorithms for Mining Association Rules, VLDB 1994:487-499.

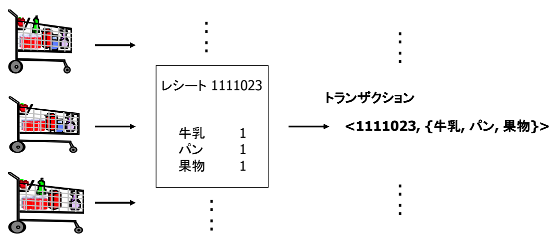
バスケット データ

小売店（デパート、スーパー、コンビニ等）での売上データをこのように呼ぶ。何故か？
一個のデータ（レコード）は
- 日時
- 顧客属性
- 購入品の単価、個数

類似の構造をもったものをバスケットデータと呼ぶ
一回ごとの取引（売上、購入、預入れ、引出し等）をトランザクションと呼ぶ

バスケット分析

- バスケット＝買い物かご
- バスケットの中（購入した商品の組合せ）を知って、どのような組合せで商品が購入されるかを知る



相関規則

- 複数種の製品（サービスでもよい）がどのような組合せで同時に購買されやすいかを表現する
- 理解が容易
  - {a, b, c, d,,,} も {a, b,,,,} も非常に頻繁に現れれば、{a, b} が購入されるときは {c, d} も購入されると言える
- 行動に結び付けられる
  - {a, b} の近くに {c, d} を置く

相関規則の例

パンとバターを含むトランザクションの90%は、牛乳を含む（パンとバターを買うと、90%の確からしさで、その客は牛乳を買う）

前件(antecedent): パンとバター

後件(consequent): 牛乳

信頼度(confidence factor): 90%

前件は前提、後件は結論などと呼ぶ

問合せ(query)の例

- 結論に「即席麺」を含む全ての規則を見つけよ
- 前提に「缶コーヒー」を含む全ての規則を見出せ
- 前提に「パン」、結論に「ジュース」を含む全ての規則をみつけよ
- 店内の棚Aと棚Bにある品目に関係する全ての規則を見出せ
- 結論に「即席麺」を含む規則のなかで「最良の」（信頼性が最も高い）k 個の規則を見出せ

記法

- アイテム – I = $\{i_1, i_2, \ldots, i_m\}$
- トランザクション – アイテムの集合 $T \subseteq I$
  - 通常、アイテムは辞書式順序で整列
- TID – トランザクションの一意名

記法

- 相関規則 – X → Y

$$X \subseteq I, Y \subseteq I \text{ かつ } X \cap Y = \phi$$

例

- *I*: アイテムの集合

{きゅうり，パセリ，玉ねぎ，トマト，塩，パン，ほうれん草，卵，バター}

- *D*: トランザクション集合

1 {{きゅうり，パセリ，玉ねぎ，トマト，塩，パン}，
2 {トマト，きゅうり，パセリ}，
3 {トマト，きゅうり，ほうれん草，玉ねぎ，パセリ}，
4 {トマト，きゅうり，玉ねぎ，パン}，
5 {トマト，塩，玉ねぎ}，
6 {パン，卵}
7 {トマト，卵，きゅうり}
8 {パン，バター}}

Confidence と Support

- 相関規則 X→Y の 信頼度confidence が c であるとは、
  D 中のトランザクションで X を含むものの 100 c% は、また、Y をも含む.
- 相関規則 X→Y の 支持度 support が s であるとは、
  D 中のトランザクションの 100 s% が X と Y とを含む.
- アイテムセット X の 支持度 support も同様に定義する。すなわち
  D 中のトランザクションの 100 s% が X を含む.

問題の定義

トランザクション集合 D が与えられたとき，支持度と信頼度が、ユーザが指定する最小支持度と最小信頼度より大きくなるような 相関規則全部 を求めよ.

なお、最小支持度より大きな支持度をもつアイテムセットを頻出アイテム集合と呼ぶ

例

T ID	アイテム
1	乳製品,果物
2	乳製品,果物, 野菜
3	乳製品
4	果物, シリアル

support({乳製品}) = 3/4
support({果物}) = 3/4
support({乳製品, 果物}) = 2/4

もし最小支持度 = 3/4 ならば
{乳製品} と {果物} は頻出アイテム集合，{乳製品,果物} は違う.

注

- X → A は X∪Y→A を意味しない
  - 最小支持度に達しないかもしれない
- X → A と A → Z から X → Z が得られるわけではない
  - 最小信頼度に達しないかもしれない

全相関規則を見つけること

- 頻出アイテム集合 全てを見出せ
  - 最小支持度より大きな支持度をもつアイテムセット.
- 頻出アイテム集合を用いて、規則を生成する.

アイデアの基本

- 仮に ABCD と AB が頻出アイテム集合とする
- 次を計算する

  conf = support(ABCD) / support(AB)
- もし conf >= minconf ならば

  AB → CD が成立する.

頻出アイテム集合の発見

- データを複数回スキャンする
- 最初のスキャン – 個々のアイテムの支持度を数える.
- 以降のスキャン
  - 以前のスキャンで得た頻出アイテム集合を用いて候補アイテム集合 を生成する.
  - データをスキャンして。当該候補の 本当の 支持度を計算する.
- もし、新しい頻出アイテム集合が得られなくなれば、停止.
- 定義. *k*-itemset: *k* 個のアイテムをもつ頻出アイテム集合.

トリック

Apriori property

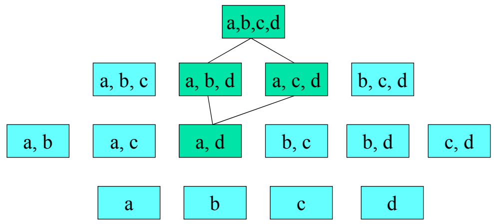
頻出アイテム集合のどんな部分集合も 頻出.

従って

頻出k-アイテム集合 k-itemset を見つけるには

- 頻出 k-1 アイテム集合を組み合わせて 候補を作る.
- 頻出でない部分集合を含む候補を削除する.

頻出アイテム集合の枝狩り



{a,d} は頻出ではないとする. そうすると 3-アイテム集合 {a,b,d}, {a,c,d} および 4-アイテム集合 {a,b,c,d}は頻出でなく、生成されない.

Apriori Algorithm

$L_1 = \{$頻出 1-アイテム集合$\}$ ← アイテム生起回数の算出

for ($k = 2$; $L_{k-1} \neq \phi$; $k{+}{+}$) do begin

  $C_k = \boxed{\text{apriori-gen}(L_{k-1});}$ ← 新しい k-アイテム集合の候補の生成

  for 全トランザクション $t \in D$ do begin

    $C_t = \boxed{\text{subset}(C_k, t)}$

    for 全候補 $c \in C_t$ do

      $c.count{+}{+};$

    end

  end

  $L_k = \{ c \in C_k | c.count \geq minsup \}$ ← minsup 以上の支持度をもつ候補のみ選び出す

end

$Answer = \bigcup_k L_k;$

全候補の支持度の計算

候補の生成

- Join step

insert into C_k
select $p.item_1, p.item_2, \ldots, p.item_{k-1}, q.item_{k-1}$
from L_{k-1} as p, L_{k-1} as q
where $p.item_1 = q.item_1, \ldots, p.item_{k-2} = q.item_{k-2}$, $p.item_{k-1} < q.item_{k-1}$

p と q は2つとも k-1 頻出アイテム集合で、先頭の k-2 アイテムが同一のもの

先頭だけで十分 何故か？

q の最後のアイテムを p に付加することによる

- Prune step

for 全アイテム集合 $c \in C_k$ do
  for cの全 (k-1)-部分集合 s do
    if ($s \notin L_{k-1}$) then
      C_k から c を削除

候補の (k-1)-部分集合を全部調べ、頻出でない部分集合をもつような候補を削除する

例

L_3 = { {1 2 3}, {1 2 4}, {1 3 4}, {1 3 5}, {2 3 4} }

join のあと

{ {1 2 3 4}, {1 3 4 5} }

prune のあと

{1 4 5} と {3 4 5} は L_3 に含まれていない

{1 2 3 4}

正しさ

$C_k \subseteq L_k$ であることを示せ

頻出アイテム集合の部分集合は頻出でなければならない

このjoinは、 L_{k-1} に任意のアイテムを付け加えて拡張し、次に、その(k-1) 部分集合がL$_{k-1}$ にないものを削除することと等価である

insert into C_k
select $p.item_1, p.item_2, \ldots, p.item_{k-1}, q.item_{k-1}$
from L_{k-1} as p, L_{k-1} as q
where $p.item_1 = q.item_1, \ldots, p.item_{k-2} = q.item_{k-2}$, $\boxed{p.item_{k-1} < q.item_{k-1}}$

for 全アイテム集合 $c \in C_k$ do
  for cの全 (k-1)-部分集合 s do
    if ($s \notin L_{k-1}$) then
      C_k から c を削除

重複を防ぐ

Subset 関数

- 候補アイテム集合 - C_k は、ハッシュ木に格納
- 大きさ k の候補アイテム集合がトランザクション t に含まれているかどうかを O(k) の時間で調べる.
- 最大時間 O(max(k,size(t))

```
L_1 = {頻出 1-アイテム集合}
for ( k = 2; L_{k-1} ≠ φ; k++) do begin
    C_k = apriori-gen(L_{k-1});
    for 全トランザクション t ∈ D do begin
        C_t = subset(C_k, t)
        for 全候補 c ∈ C_t do
            c.count++;
        end
    end
    L_k = { c ∈ C_k | c.count ≥ minsup}
end
Answer = ∪_k L_k;
```

問題?

- 全てのスキャンが全データに対して行われている.

```
L_1 = {頻出 1-アイテム集合}
for ( k = 2; L_{k-1} ≠ φ; k++) do begin
    C_k = apriori-gen(L_{k-1});
    for 全トランザクション t ∈ D do begin
        C_t = subset(C_k, t)
        for 全候補 c ∈ C_t do
            c.count++;
        end
    end
    L_k = { c ∈ C_k | c.count ≥ minsup}
end
Answer = ∪_k L_k;
```

簡単な例:

Trans-ID	Items
1	A C D
2	B C E
3	A B C E
4	B E
5	A B C E

簡単な例:

TID	アイテム集合
1	ACD
2	BCE
3	ABCE
4	BE
5	ABCE

最小支持度 60%
最小信頼度 75%

頻出アイテム集合	支持度
{BCE},{AC}	60%
{BC},{CE},{A}	60%
{BE},{B},{C},{E}	80%

相関規則 : $X => Y$

信頼度$(X => Y)$ = support$(X \cup Y)$/ support(X)

支持度$(X => Y)$ = support$(X \cup Y)$

規則 $\{BC\} => \{E\}$に対し:

支持度 = support$(\{BCE\})$ = 60%

信頼度 = support$(\{BCE\})$/support$(\{BC\})$ = 100%

簡単な例:

TID	アイテム
1	ACD
2	BCE
3	ABCE
4	BE
5	ABCE

最小支持度 60%
最小信頼度 75%

頻出アイテム集合	支持度
{BCE},{AC}	60%
{BC},{CE},{A}	60%
{BE},{B},{C},{E}	80%

相関規則	信頼度
{BC} =>{E}	100%
{BE} =>{C}	75%
{CE} =>{B}	100%
{B} =>{CE}	75%
{C} =>{BE}	75%
{E} =>{BC}	75%

支持度$(X => Y)$ = support$(X \cup Y)$

信頼度$(X => Y)$

= support$(X \cup Y)$/ support(X)

簡単な例 minsup = 40%

Database D

TID	Items
1	A C D
2	B C E
3	A B C E
4	B E
5	A B C E

Scan D

C_1

itemset	sup.
{A}	3
{B}	4
{C}	4
{D}	1
{E}	4

L_1

itemset	sup.
{A}	3
{B}	4
{C}	4
{E}	4

C_2

itemset
{AB}
{AC}
{AE}
{BC}
{BE}
{CE}

Scan D

C_2

itemset	sup
{AB}	2
{AC}	3
{AE}	2
{BC}	3
{BE}	4
{CE}	3

L_2

itemset	sup
{AB}	2
{AC}	3
{AE}	2
{BC}	3
{BE}	4
{CE}	3

簡単な例　minsup = 40%

Database D

TID	Items
1	A C D
2	B C E
3	A B C E
4	B E
5	A B C E

itemset	sup
{AB}	2
{AC}	3
{AE}	2
{BC}	3
{BE}	4
{CE}	3

itemset
{ABC}
{ABE}
{ACE}
{BCE}

Scan D

C_3

itemset	sup
{ABC}	2
{ABE}	2
{ACE}	2
{BCE}	3

L_3

itemset	sup
{ABC}	2
{ABE}	2
{ACE}	2
{BCE}	3

C_4

itemset
{ABCE}

Scan D

C_4

itemset	sup
{ABCE}	2

L_4

itemset	sup
{ABCE}	2

別の例

Database D

TID	items
100	A C D
200	B C E
300	A B C E
400	B E

Scan D

itemset	sup.
{A}	2
{B}	3
{C}	3
{D}	1
{E}	3

Remove

itemset	sup.
{A}	2
{B}	3
{C}	3
{E}	3

itemset
{A B]
{A C}
{A E}
{B C}
{B E}
{C E}

Scan D

Itemset	sup
{A B}	1
{A C}	2
{A E}	1
{B C}	2
{B E}	3
{C E}	2

Remove

itemset	sup
{A C}	2
{B C}	2
{C E}	3
{B E}	2

itemset
{B C E}

Scan D

itemset	sup
{B C E}	2

Remove

itemset	sup
{B C E}	2

Aprioriアルゴリズム —　例3

データベース D

TID	Items
100	A C D
200	B C E
300	A B C E
400	A B E

Scan D

C_1

itemset	sup.
{A}	3
{B}	3
{C}	3
{D}	1
{E}	3

L_1

itemset	sup.
{A}	3
{B}	3
{C}	3
{E}	3

C_2

itemset
{A B}
{A C}
{A E}
{B C}
{B E}
{C E}

Scan D

C_2

itemset	sup
{A B}	2
{A C}	2
{A E}	2
{B C}	2
{B E}	3
{C E}	2

C_3

itemset
{A B C}
{A B E}
{A C E}
{B C E}

Scan D and Remove

L_3

itemset	sup
{A B E}	2
{B C E}	2

Aprioriアルゴリズム —　例4

データベース D

TID	Items
100	A C E
200	B C E
300	A B C
400	B E

Scan D

C_1

itemset	sup.
{A}	2
{B}	3
{C}	3
{D}	1
{E}	2

L_1

itemset	sup.
{A}	2
{B}	3
{C}	3
{E}	2

C_2

itemset
{A B}
{A C}
{A E}
{B C}
{B E}
{C E}

Scan D

C_2

itemset	sup
{A B}	1
{A C}	2
{A E}	0
{B C}	2
{B E}	2
{C E}	1

L_2

itemset	sup
{A C}	2
{B C}	2
{B E}	2

興味度の尺度

- 客観的尺度には二つのよく知られた尺度：
  - 支持度
  - 信頼度
- 主観的尺度
  実際に、ルール(パターン)が興味深いのは、例えば、以下のような場合
  - それが 思いがけない時 (ユーザにとって驚くべき事実であるとき); and/or
  - 行動可能なとき (ユーザがそれによって何か意味のある行動がとれるとき)

支持度と信頼度に対する批判

- 例 1: (Agrawal & Yu, PODS98)
  - 5000人の学生の中で
    - 3000人がバスケットボールをする
    - 3750人がシリアルを食べる
    - 2000人がバスケットをし、かつシリアルを食べる
  - *バスケットボールをする ⇒ シリアルを食べる* [40%, 66.7%]　は誤解を招く。なぜなら、全学生の中でシリアルを食べる学生は75%で、それは 66.7%よりも大きいから。
  - *バスケットボールをする ⇒ シリアルを食べない* [20%, 33.3%]　の方がより正確だが、支持度と信頼度は、いずれもより低い。

支持度　信頼度

$$0.4 = \frac{2000}{5000},\ 0.667 = \frac{2000}{3000}$$

$$0.2 = \frac{1000}{5000},\ 0.333 = \frac{1000}{3000}$$

	basketball	not basketball	sum(row)
cereal	2000	1750	3750
not cereal	1000	250	1250
sum(col.)	3000	2000	5000

支持度と信頼度に対する批判2

- 例2:
  - XとY：正の相関を持つ（8ケのペア中、6ケが一致)
  - XとZ：負の相関を持つ（8ケのペア中、5ケが不一致)
  - X⇒Zの支持度と信頼度の方が大きくなる。

X	1	1	1	1	0	0	0	0
Y	1	1	0	0	0	0	0	0
Z	0	1	1	1	1	1	1	1

Rule	Support	Confidence
X⇒Y	25%	50%
X⇒Z	37.50%	75%

興味度の他の尺度：corr

- $\mathrm{corr}_{A,B} = \dfrac{P(A \wedge B)}{P(A)P(B)}$
  - $P(A)$と$P(B)$ を考える（A, Bを含まない場合を考えることに）
  - AとBとが独立のとき、$P(A \wedge B)=P(B)*P(A)$
  - この値が1より小さいとき，AとBは負の相関を持つ; そうでなければ、AとBは正の相関を持つ。

X	1	1	1	1	0	0	0	0
Y	1	1	0	0	0	0	0	0
Z	0	1	1	1	1	1	1	1

Itemset	Support	corr
X,Y	25%	2
X,Z	37.50%	0.9
Y,Z	12.50%	0.57

例：バスケットボールとシリアルの場合

	basketball	not basketball	sum(row)
cereal	2000	1750	3750
not cereal	1000	250	1250
sum(col.)	3000	2000	5000

バスケットボールをする：B　　シリアルを食べる：C

$P(B)=0.6$　$P(C)=0.75$　$P(\overline{C})=0.25$　$P(B \wedge C)=0.4$　$P(B \wedge \overline{C})=0.2$

$B \Rightarrow C$ [40%, 66.7%]　$\mathrm{corr}_{B,C} = \dfrac{P(B \wedge C)}{P(B)P(C)} = \dfrac{0.4}{0.6 \times 0.75} = \dfrac{0.4}{0.45}$

$B \Rightarrow \overline{C}$ [20%, 33.3%]　$\mathrm{corr}_{B,\overline{C}} = \dfrac{P(B \wedge \overline{C})}{P(B)P(\overline{C})} = \dfrac{0.2}{0.6 \times 0.25} = \dfrac{0.2}{0.15}$