Regras de Associação

Paulo J Azevedo

DI - Universidade do Minho
2019

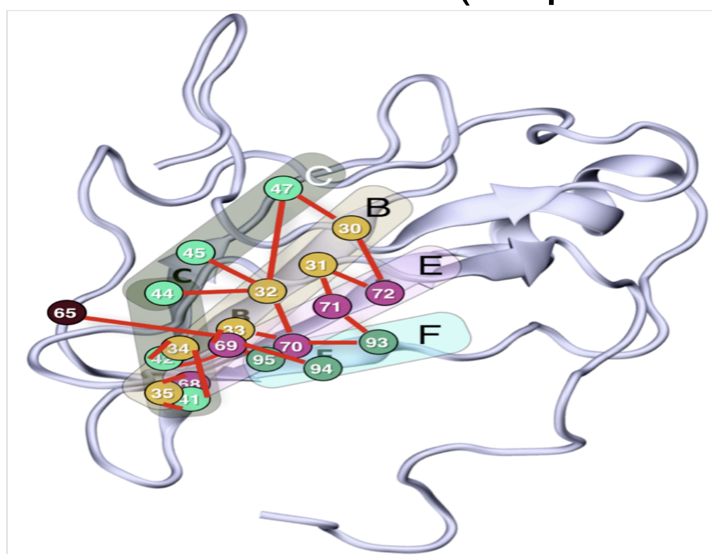
Detecção de associações nos dados

Sumário

- Motivação
- Introdução às regras de associação
- Algoritmos para cálculo de termos frequentes
  - Apriori e outras variantes Breath-first
- Representações Verticais
  - Algoritmos Depth-first com representações verticais
- Medidas de interesse
- Selecção e pruning de regras
- Atributos contínuos em regras de associação
- Subgroup Mining
  - Estudo de propriedades numéricas com regras de associação
  - Contrast Set Mining

Pattern Mining

- Identificar padrões interessantes nos dados.
  Padrões formados por uma configuração específica de uma colecções de elementos atómicos dos dados.
  - {maças, laranjas, iogurte} (termos frequentes)
  - ATGCTTCGGCAA (sequência de DNA)
  - Grafos:



  - etc…
  - O que significa *interessante* ?

    e.g. que ocorre um número significativo de vezes…

Problema

- Base de Dados de
  Ticket Data
- *Ex:*

1 1901,1881,199,901

2 901,1661

3 676,199,177,100

…..

…

120099 78,1881,199,8

item

Número da transacção

- O marketing da cadeia de Hipermercados pretende fazer um estudo de comportamento de compras.
- Tem acesso aos dados representativos dos "cestos de compras" (basket data)
- Exemplo de perguntas a responder:
- Que produtos estão associadas ao consumo de cerveja X ?
- Como podemos descrever a população consumidora de amendoins?
- Onde devem estar localizadas os produtos de limpeza doméstica ?
- Como se relacionam os produtos 1661 e 199 ?

Como expressar a informação extraída ?

- Regras que relacionam produtos (items),

Qualidade das regras expressa por medidas estatísticas.

901 & 1661 $\rightarrow$ 67

Há um número explosivo de potenciais regras que podem ser derivadas!

Todas as regras ?

Qual o procedimento eficiente a aplicar?

Como obter ?

Como seleccionar ?

Como discriminar regras “boas” de “más” ?

Como organizar ?

Medidas de Interesse

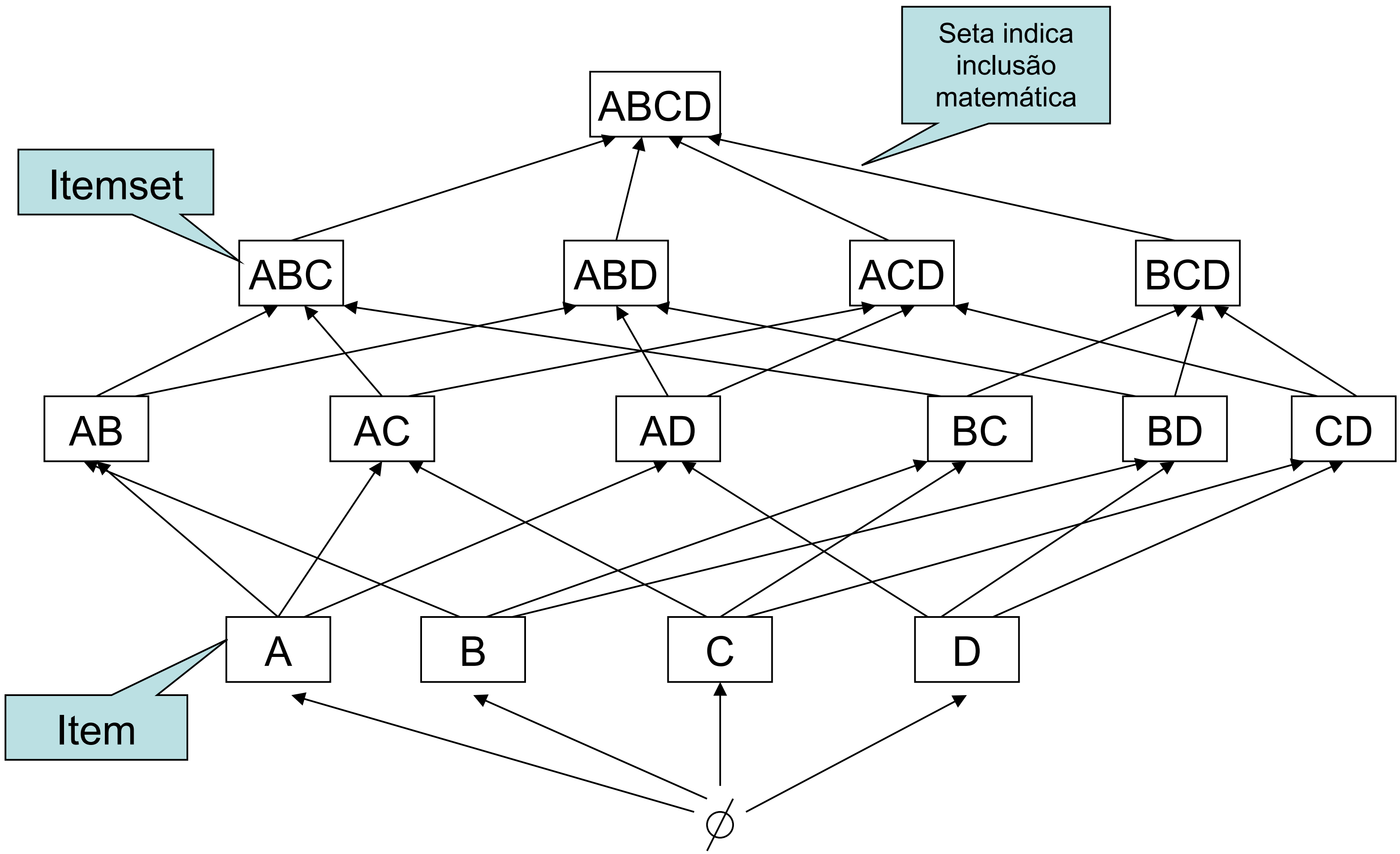
- Tipicamente recorre-se a uma métrica de incidência para definir quais as associações significantes. O utilizador define a noção de raridade de uma associação!
- A mais popular é o *suporte (contagem)* dos itemsets.
- As regras são qualificadas por uma métrica de interesse (previsibilidade, solidez ou força da regra).
- Normalmente é usada a *confiança* (probabilidade condicional)
  - $conf(A \rightarrow C) = sup(AC) / sup(A)$

- Assim, a regra de associação:

901 & 707 → 1088   (s=0.3, conf=0.9)

Deve ser lida como: *a compra conjunta dos produtos **901**, **707** e **1088** ocorre em 30% das transacções. Por outro lado, verifica-se que 90% das transacções que contêm **901** e **707** também contêm o produto **1088**.*

Podemos ver o problema pela perspectiva do espaço de pesquisa a explorar



Geração de Regras

- Cálculo da confiança: conf(A→C) = s(A C) / s(A).
- Noção de thresholds de conf e sup (minsup e minconf)
- Algoritmo “trivial” e.g:

  Tendo ABC (verificar a regra AB → C),

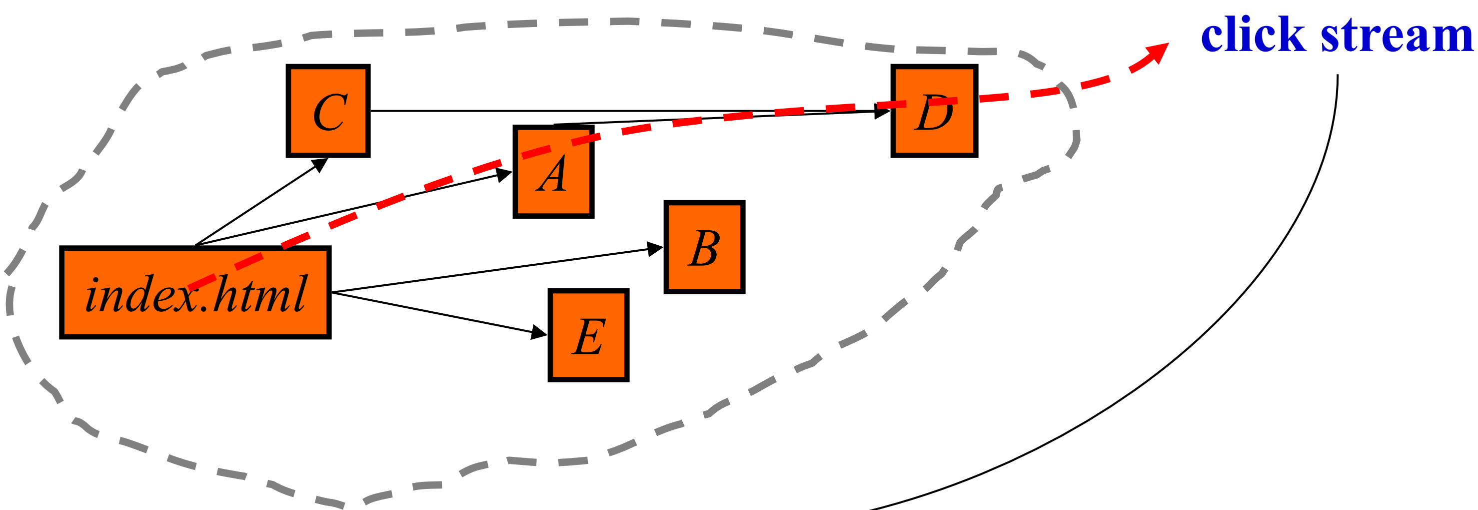
    testar, sabendo s(AB) e s(ABC),

    se s(ABC) / s(AB) ≥ minconf

  Fazer este procedimento para todos os

  itemsets ∈ Power_set({A,B,C}) em que #itemset > 1.

Aplicações: Sistema de Recomendações com ARs



Obs.: *A* *D*

Rules:

- *A* *B* *F* → *X* (conf: 0,8)
- *A* *E* → *R* (conf: 0,7)
- *A* *D* → *F* (conf: 0,6)
- *A* → *D* (conf: 0,5)
- *D* → *X* (conf: 0,4)

Recommendations (top 2):

- *F* (0,6)
- *X* (0,4)

Cálculo de Termos Frequentes (frequent itemsets)

- Algoritmo naif:

Se sup(termo) ≥ minsup dizemos que o termo é frequente.

Seja K = { items em DB},

Derivar o P(K) (Power_set),

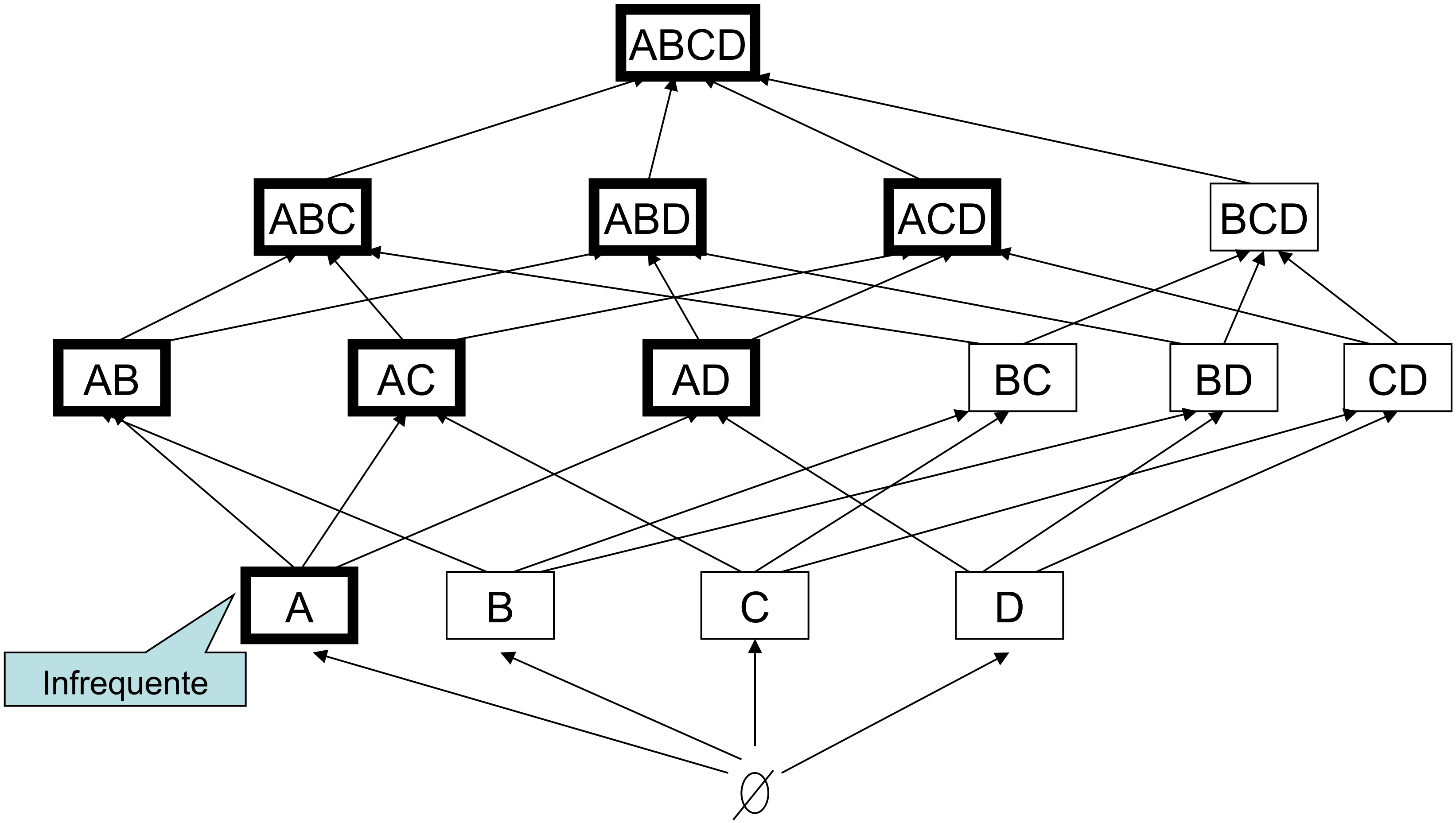
Percorrer DB para contar as ocorrências de P(K)

Filtrar os itemset em P(K) que não verificam minsup.

- Intractável!!!!!!!!
- Melhor: fazer uso da propriedade *downward closure* do suporte

$$\text{Se } X \subseteq Y \text{ então } s(X) \geq s(Y)$$

Aplicação da Propriedade Anti-monótona



Algoritmo Apriori [Agrawal & Srikant 94]

- L_1 = { 1-items frequentes}
- For(k=2;L_k-1 ≠ {}; k++) do

  C_{k} = apriori_gen(L_{k-1});

  forall transacções t ∈ D do

  - C_{t} = subsets(C_{k},t)
  - Forall candidatos c ∈ C_{t} do
    - c.count++;

  End

  L_{k} = {c ∈ C_{k} | c.count ≥ minsup}

End

Answer= ∪ L_{k};

Algoritmo Bottom-up e breath-first. Em **apriori_gen** é gerado os candidatos a contar. Só são considerados os candidatos que obedecem à propriedade anti-monótona
(é candidato se todos os seus subconjuntos são frequentes!)

Apriori “in action…”

DataBase DB

TID	ITEM–SET
100	1 2 4 5
200	1 3 5
300	1 2 3 5 6
400	1 2 5

Scan DB

C_1

ITEM–SET	SUPORTE
{1}	4
{2}	3
{3}	2
{4}	1
{5}	4
{6}	1

L_1

ITEM–SET	SUPORTE
{1}	4
{2}	3
{3}	2
{5}	4

C_2

ITEM–SET
{12}
{13}
{15}
{23}
{25}
{35}

Scan DB

C_2

ITEM–SET	SUPORTE
{12}	3
{13}	2
{15}	4
{23}	1
{25}	3
{35}	2

L_2

ITEM–SET	SUPORTE
{12}	3
{13}	2
{15}	4
{25}	3
{35}	2

C_3

ITEM–SET
{125}
{135}

Scan DB

C_3

ITEM–SET	SUPORTE
{125}	3
{135}	2

Não gerado pois {123}, {235}, não são frequentes!

L_4

ITEM–SET	SUPORTE
{1235}	1

Package R "arules"

Dataset de documentos descarregados de um site da UViena

```
> library(arules)
> data("Epub")
> summary(Epub)
transactions as itemMatrix in sparse format with
 15729 rows (elements/itemsets/transactions) and
 936 columns (items) and a density of 0.001758755

most frequent items:
doc_11d doc_813 doc_4c6 doc_955 doc_698 (Other)
   356     329     288     282     245   24393
> rules <- apriori(Epub,parameter = list(support = 0.001, confidence = 0.3))
> write(rules)
"rules" "support" "confidence" "lift" "count"
"1" "{doc_506} => {doc_507}" 0.0012079598194418 0.655172413793103
303.094320486815 19
"2" "{doc_507} => {doc_506}" 0.0012079598194418 0.558823529411765
303.094320486815 19
"3" "{doc_714} => {doc_574}" 0.0010808061542374 0.369565217391304
113.978260869565 17
……….
```

Algoritmo FP-Growth [Han 2000]

- Um dos algoritmos mais populares para cálculo de termos frequentes

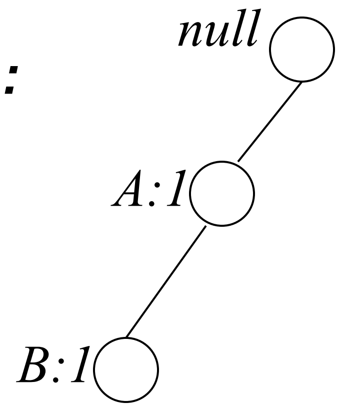
- Usa uma representação eficiente da base de dados na forma de uma estrutura em árvore - FP-tree.
- Dois scans na BD: 1º para contar items frequentes, 2º para construir a FP-tree.
- Uma vez a FP-tree construida, o algoritmo usa uma aproximação recursiva *divide-and-conquer* para obter os itemsets frequentes.

Construção da estrutura FP-Tree

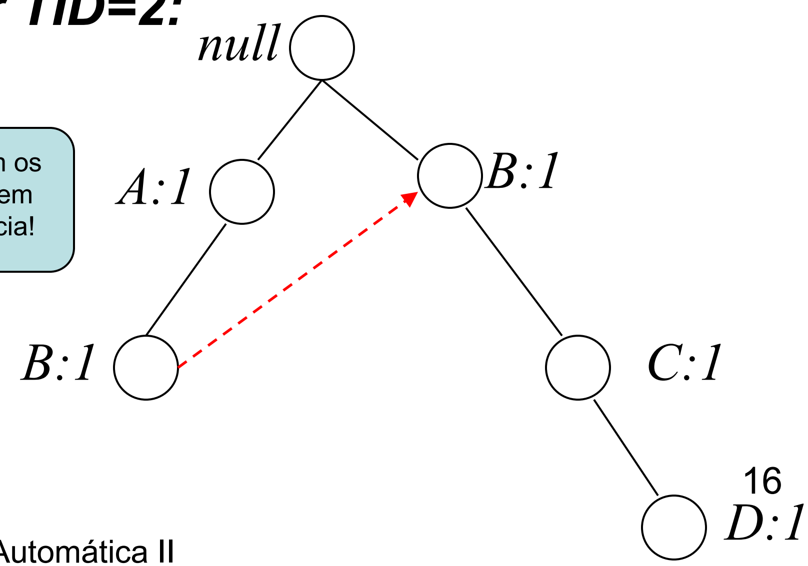
TID	Items
1	{A,B}
2	{B,C,D}
3	{A,C,D,E}
4	{A,D,E}
5	{A,B,C}
6	{A,B,C,D}
7	{B,C}
8	{A,B,C}
9	{A,B,D}
10	{B,C,E}

As transações estão com os items ordenados por ordem descendente de frequência!

Depois de ler TID=1:



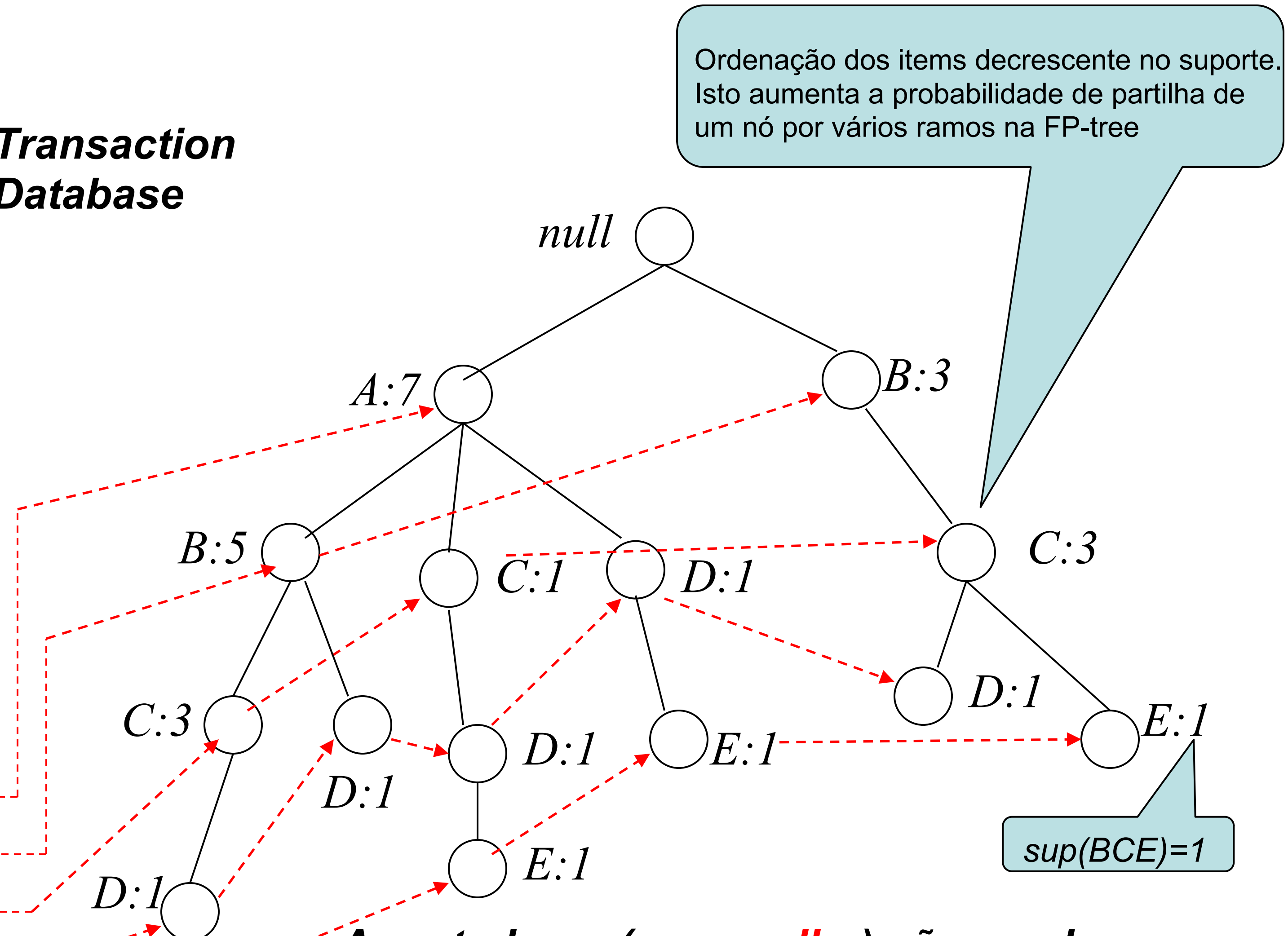
Depois de ler TID=2:



Construção da FP-Tree (2)

TID	Items
1	{A,B}
2	{B,C,D}
3	{A,C,D,E}
4	{A,D,E}
5	{A,B,C}
6	{A,B,C,D}
7	{B,C}
8	{A,B,C}
9	{A,B,D}
10	{B,C,E}

Transaction Database



Header table

Item	Pointer
A	
B	
C	
D	
E	

Apontadores (a vermelho) são usadas para facilitar a geração dos termos frequentes.

FP-growth: Termos frequentes

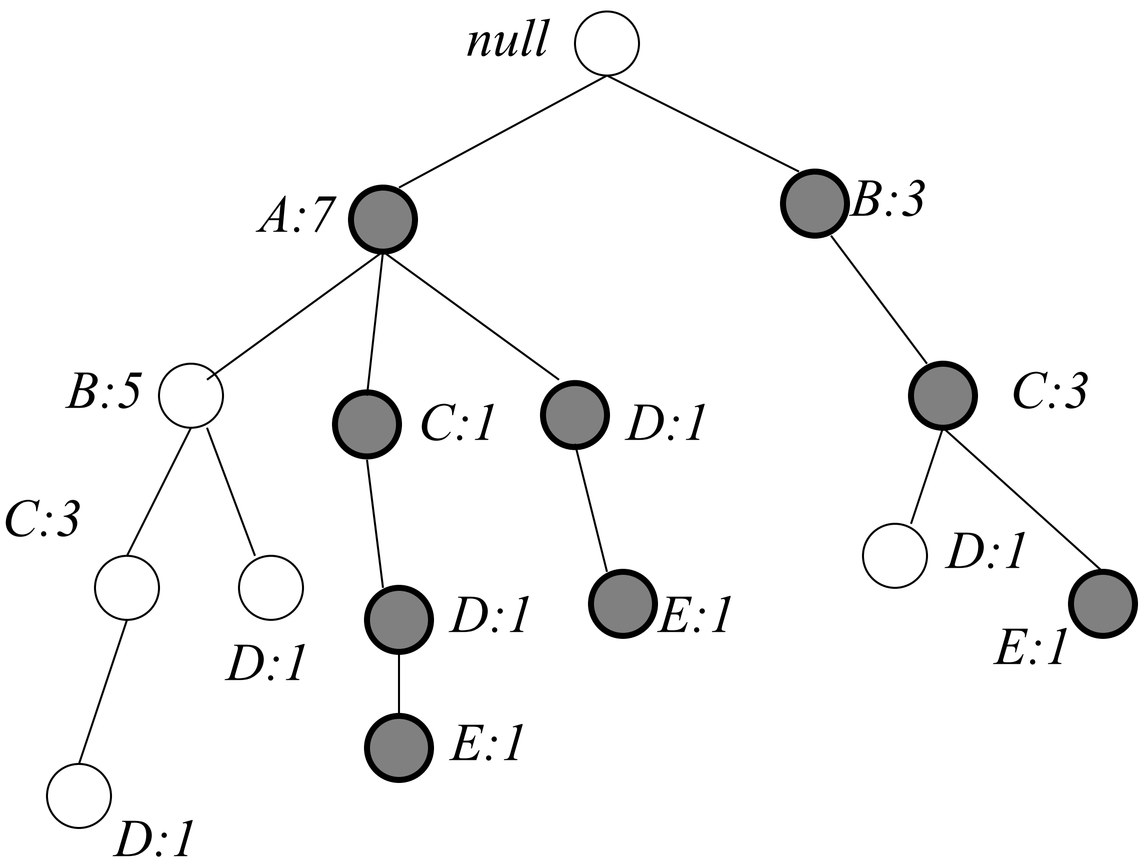
- Todos os termos frequentes podem ser obtidos da FP-tree,

- Estes são obtidos seguindo os links dos nós que iniciam na tabela (header table).

- Começar pelos items <u>menos frequentes</u>.

- Podemos resumir o algoritmo:
  1. Para cada item A
  2. Obter os itemsets que contêm A (conditional pattern base)
  3. Obter FP-tree para esses itemsets (conditional FP-tree)
  4. Obter itemsets tamanho 2 da conditional FP-tree.
  5. Para cada itemset tamanho 2 repetir passo 2).

FP-growth – derivar itemsets frequentes



Construir a** <u>conditional pattern base</u> **para E:

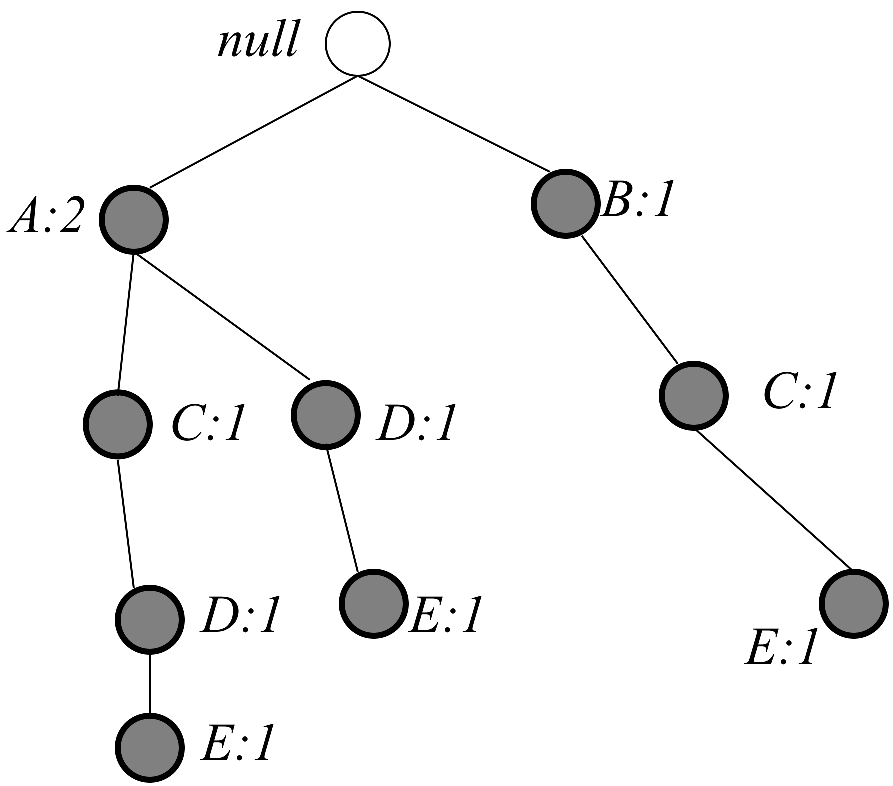
P = {(A:1,C:1,D:1),
(A:1,D:1),
(B:1,C:1)}

Aplicar FP-growth recursivamente em P

NOTA: minsup(abs) = 2

FP-growth

Conditional tree para E:



Conditional Pattern base para E:

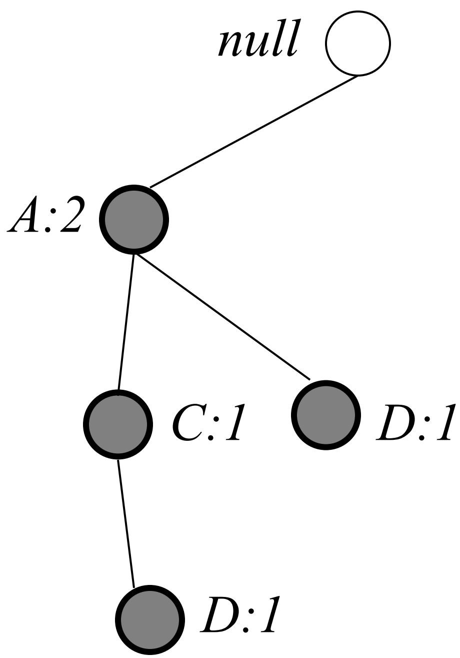
P = {(A:1,C:1,D:1,E:1),
(A:1,D:1,E:1),
(B:1,C:1,E:1)}

 # de E =3: {E} é frequente

Aplicar recursivamente FP-growth em P

FP-growth

<u>Conditional tree para D dentro da conditional tree de E:</u>



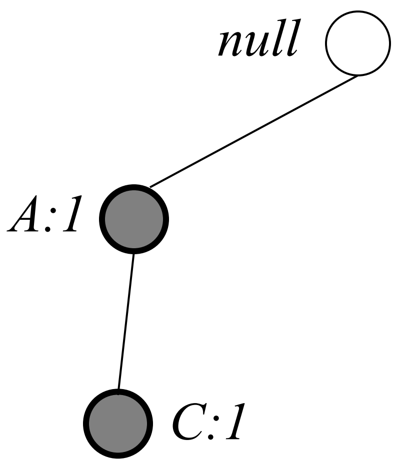
Conditional pattern base de D dentro da conditional base de E:
P = {(A:1,C:1,D:1), (A:1,D:1)}

 # de D =2: {D,E} é um itemset frequente

Aplicar recursivamente FP-growth em P

FP-growth

<u>Conditional tree para C dentro de D, esta dentro de E:</u>



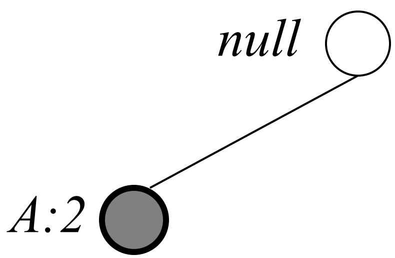
Conditional pattern base para C dentro de D dentro de E:

P = {(A:1,C:1)}

 # de C = 1: {C,D,E} não é frequente!

FP-growth

Conditional tree para A dentro de D dentro de E:



 # de A = 2: {A,D,E} é frequente

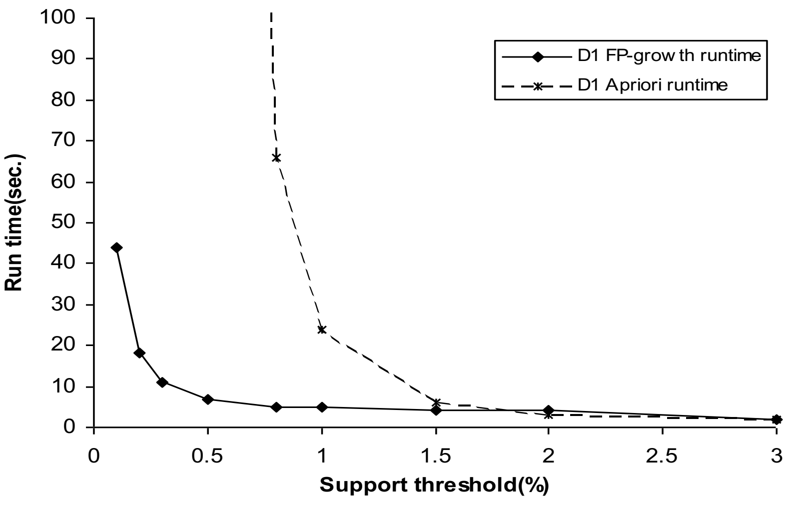
Próximo passo:

Construir conditional tree de C dentro da conditional tree de E

Continuar até explorar a conditional tree para A (que tem só o nó A)

Benefícios da estrutura FP-tree

- Desempenho mostra que
  - FP-growth é uma ordem de magnitude mais rápido que o Apriori.
- Príncipios:
  - No candidate generation, no candidate test
  - Uso de uma estrutura compacta
  - Elimina a necessidade de sucessivos database scans
  - Operação básicas são contagem e construção da FP-tree.



Algoritmos: Representações

- Horizontais
  - Transacções são listas de items. Ex:

    t12: 1,4,6,7,12,129,929

    t15: 2,4,5,6,14,189,901

- Verticais
  - Representar a cobertura de cada item nas transacções. Ex:

    Tidlist(6) = [t12,t15,t24,t123,t300,…]

    Tidlist(14) = [t15,t120,t541,…]

    Tidlist(129)= [t12,t18,t45,…]

Representações Verticais

- Cover Lists
  - Ideal para “sparse” data
  - Tidlist(I) = [t4,t9,t12,t45,t312,…]
  - s(I) = #coverlist(I)
  - Tidlist(A U B) = tidlist(A) ∩ tidlist(B)
- BitMaps
  - Melhores resultados com “dense” data
  - bitmap(I)= “0010011100011000”
  - s(I) = bitcount(bitmap(I)) *Contar bits ligados*
  - bitmap(A U B) = bitmap(A) & bitmap(B) *Bitwise logical* **and**

Exemplos de regras

Association Rules (geradas pelo Caren) ...

Sup = 0.01500	Conf = 0.37500	oranges	←	bananas & peaches
Sup = 0.03900	Conf = 0.30000	oranges	←	peaches
Sup = 0.01000	Conf = 0.28571	oranges	←	bananas & potatoes
Sup = 0.01000	Conf = 0.28571	oranges	←	peaches & potatoes

- Que informação é possível tirar deste tipo de estrutura ?
- Leitura das regras…
- Capacidade de previsão?
- Interpretação das métricas
- Característica da população descrita...
- Redundância

Exemplo execução CAREN para o dataset *hepatitis*

```
> caren hepatitis.data  0.1  0.5  -s,  -Att  -Hclass  -classclass  -ovrt  -fisher  -null?  -Discfia1,a18,a17,a16,a15,a14

Sup = 0.10968   Conf = 0.73913    class=1c   <--   a17=]-oo : 3.8500]  &  a6=1  &  a19=2
Sup = 0.10323   Conf = 0.69565    class=1c   <--   a18=]-oo : 50.5000]  &  a17=]-oo : 3.8500]  &  a8=2  &  a5=1
Sup = 0.11613   Conf = 0.69231    class=1c   <--   a6=1  &  a19=2  &  a8=2
Sup = 0.11613   Conf = 0.69231    class=1c   <--   a18=]-oo : 50.5000]  &  a17=]-oo : 3.8500]  &  a2=1
Sup = 0.14194   Conf = 0.68750    class=1c   <--   a17=]-oo : 3.8500]  &  a19=2  &  a8=2
Sup = 0.13548   Conf = 0.67742    class=1c   <--   a17=]-oo : 3.8500]  &  a19=2  &  a5=1  &  a2=1
Sup = 0.12903   Conf = 0.66667    class=1c   <--   a17=]-oo : 3.8500]  &  a19=2  &  a5=1  &  a4=2
Sup = 0.10323   Conf = 0.66667    class=1c   <--   a14=]1.6500 : +oo[  &  a17=]-oo : 3.8500]
Sup = 0.10323   Conf = 0.66667    class=1c   <--   a18=]-oo : 50.5000]  &  a17=]-oo : 3.8500]  &  a4=2  &  a5=1
Sup = 0.10323   Conf = 0.64000    class=1c   <--   a11=1  &  a17=]-oo : 3.8500]  &  a4=2  &  a2=1
Sup = 0.13548   Conf = 0.63636    class=1c   <--   a17=]-oo : 3.8500]  &  a19=2  &  a4=2  &  a2=1
Sup = 0.11613   Conf = 0.62069    class=1c   <--   a18=]-oo : 50.5000]  &  a17=]-oo : 3.8500]
Sup = 0.11613   Conf = 0.62069    class=1c   <--   a18=]-oo : 50.5000]  &  a5=1  &  a2=1
Sup = 0.11613   Conf = 0.60000    class=1c   <--   a6=1  &  a19=2  &  a4=2  &  a2=1
Sup = 0.14194   Conf = 0.59459    class=1c   <--   a17=]-oo : 3.8500]  &  a19=2
Sup = 0.13548   Conf = 0.58333    class=1c   <--   a19=2  &  a5=1  &  a8=2  &  a4=2  &  a2=1
Sup = 0.12258   Conf = 0.57576    class=1c   <--   a6=1  &  a19=2
.
.
.
.
```

Regras Default

O "mundo complementar" onde não há tomates corresponde a 70% das transações!

O mesmo que s(tomates)=0.3

- Do tipo { } → C
- e.g. { } → tomates s=0.3, conf=0.3

✓ Neste caso Sup == Conf, o que indica a incidência desta subpopulação.
✓ Pode ser usada para medir distância para independência

✓ Algumas medidas de interesse referem estas regras
✓ Ajuda a identificar redundância de certas regras

✓ Usadas em classificação para controlar previsão por omissão

Medidas de Interesse

Confiança:

• mede probabilidade condicional P(C) dado A
•Tende a dar ênfase a regras não correlacionadas (spurious rules).

$$conf\,(A \rightarrow C) = \frac{s(A \cup C)}{s(A)}$$

Laplace:

• estimador da confiança que tem em conta o suporte
• torna-se mais pessimista com o valores de s(A) mais pequenos
• sofre dos mesmos problemas da confiança

$$lapl(A \rightarrow C) = \frac{s(A \cup C) + 1}{s(A) + 2}$$

Lift:

• Mede a distância para a independência entre A e C
• varia entre [0, +oo[
• Valor 1 → independência,
• Valores longe de 1 → indicam que a evidencia de A fornece informação sobre C.
• mede co-ocorrência (não implicação)
• é simétrica!

$$Lift(A \rightarrow C) = \frac{conf\,(A \rightarrow C)}{s(C)}$$

Medidas de Interesse (2)

Mundo complementar à regra { } → *C*.

Conviction:

- motivada pelas fraquezas de conf e lift
- varia entre [0.5, +oo[
- tenta capturar o grau de implicação entre A e C
- é directional i.e. conv(A →C) ≠ conv(C → A)
- valor 1 indica independência
- motivação (implicação lógica): A → C ⇔ ~A ∨ C ⇔ ~(A ∧ ~C)
- medir quanto (A ∧ ~C) se desvia da independência.
- inverto o rácio entre s(A ∨ ~C*) e s(A) x s(~C) para lidar com negação*
- excelente medida para classificação.
- rácio entre a frequência esperada de previsões erradas (assumindo independência entre *A* e *C)*, e a frequência observada de previsões erradas e.g. ***conv = 1.2*** significa que a regra estaria errada 1.2 mais vezes se associação entre *A* e *C* fosse fruto do acaso.

$$conv(A \rightarrow C) = \frac{\boxed{1 - s(C)}}{1 - conf(A \rightarrow C)}$$

Frequência esperada no caso de independência

Leverage:

- varia entre]-0.25, 0.25[
- mede o número de casos extra obtidos

em relação ao esperado (à independência)

$$leve(A \rightarrow C) = s(A \cup C) - \boxed{s(A) \times s(C)}$$

Teste do χ²:

•Mede independência estatística entre antecedente e consequente

- não captura a força da correlação entre A e C
- Apenas suporta a decisão de independência

$$X^2 = \sum_{r_{i \varepsilon R}}^{n} \frac{(O(r) - E[r])^2}{E[r]}$$

Problemas da métrica *Confiança*

A confiança pode não detectar independência. A regra *ovos* → *leite* pode ter conf=80% mas podemos saber que o consumo de *ovos* é independente de *leite*.

Independência entre A e C:

$$s(A \cup C) = s(A) \times s(C)$$

Noutros casos podemos ter dependência positiva/negativa. Podemos usar uma medida de X^2 para medir correlação entre antecedente e consequente.

$$X^2 = \sum_{r_{i \varepsilon R}}^{n} \frac{(O(r) - E[r])^2}{E[r]}$$

Ver switch –chi no Caren para filtrar regras independentes.

Aplicar teste de X^2 com um valor de conf=95% e 1 grau de liberdade, Se X^2 >= 3.84 rejeita-se a hipótese de independência, (na tabela, para α=0.05 e 1 grau o valor é 3.84)

Fraquezas do framework suporte - confiança

- Pode ser difícil definir um suporte mínimo ideal
- Certos problemas podem exigir suporte mínimos extremamente baixos e.g. caviar → champagne

- Suporte e confiança mínimas altas podem perder regras interessantes
- Confiança pode atribuir alto interesse a regras não correlacionadas (como vimos!)

- Outras medidas sofrem de problemas similares

Selecção e Pruning de Regras

- Um algoritmo de FIM (mesmo com filtragem de suporte confiança mínima) pode gerar milhões de regras. Podemos ter #{regras} >> #{transacções} !!!
- Maioria das regras são geradas fruto do acaso (no sentido estatístico). Noção de *false discoveries*
- Regras não correlacionadas (em que o antecedente e o consequente são independentes)

Regras Redundantes

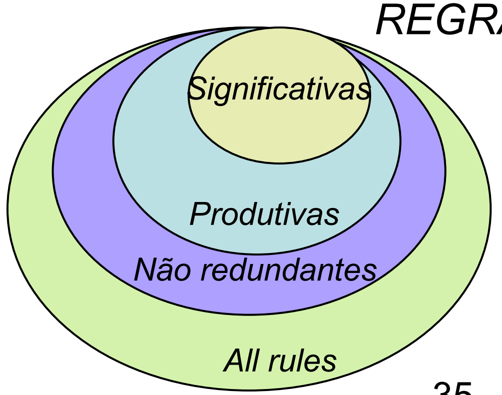
- Aparecimento de *regras redundantes (Zaki00)*. Regras contêm items no antecedente que são explicados por outros items também no antecedente.

- Ex (grávida → mulher):
    - Grávida & mulher → retenção_de_liquidos

  – Descartar *regra redundante* $x \rightarrow y$ se:
    – $\exists z \in x : s(x \rightarrow y) = s(x - z \rightarrow y)$

Notar preservação do suporte!!

REGRAS



Regras Produtivas

- Problema de *improvement* nas regras

Conf = 0.300 oranges $\leftarrow$ bananas & peaches
Conf = 0.315 oranges $\leftarrow$ peaches

- Noção de improvement:
  - uma regra mais especifica tem de produzir uma mais valia em termos de valor de medida de interesse.

$$imp(A \to C) = \min(\forall A' \subset A : met(A \to C) - met(A' \to C))$$

met pode ser = {conf, lift, conv, leve, etc}

- Se improvement > 0 dizemos que são *regras produtivas*.

Significância Estatística

- Em vez de definir um improvement mínimo, aplicar um teste de significância estatística: eliminar *regras não significativas* (Webb, Magnum Opus, Webb[2007])
- Uma regra $x \rightarrow y$ é *insignificante* se
  - Existir outra regra $x - z \rightarrow y$ em que valor

    $\mathrm{met}(x \rightarrow y) - \mathrm{met}(x - z \rightarrow y)$

    não é significativamente alto (sendo $\mathrm{met}(x \rightarrow y) > \mathrm{met}(x\text{-}z \rightarrow y)$)

- Usa-se um teste estatístico frequentista de hipóteses para determinar significância.

Teste para Regras Significativas

H_0: $p(y|x) \leq p(y|x\text{-}z)$

- *Fisher exact Test*,
  - p-value(x → y, x–z → y):

Calcula a probabilidade de observar os valores obtidos de ocorrência de x&y (ou valores maiores) dado o número de ocorrências de x-z&y se P(y|x)==P(y|x-z). Assume amostragem sem reposição

  - Aceitar x → y se <u>todos</u> os p-value ≤ α
- *Webb* aplica este teste <u>somente</u> entre cada regra x → y e as suas imediatas generalizações. Isto é, regras:
  - { } → y e
  - x–z → y tal que $|x\text{-}z| = n - 1$, sendo $|x| = n$.
  - Notar que x → y tem $2^{|x|} - 1$ generalizações!!

Teste de Fisher para Regras Significativas

	y	¬y	
x	*a*	*b*	a+b
x-z	*c*	*d*	c+d
	a+c	b+d	

a = *s(x ∪ y)*
b = *s(x ∪ ¬y)*
c = *s(x-z ∪ ¬z ∪ y)*
d = *s(x-z ∪ ¬z ∪ ¬y)*

Sensível ao tamanho de ambas as populações testadas (assume não reposição!)

Usa o *Fisher exact Test*, p-value(x → y, x–z → y):

$$p = \sum_{i=0}^{\min(b,c)} \frac{(a+b)!(c+d)!(a+c)!(b+d)!}{(a+b+c+d)!(a+i)!(b-i)!(c-i)!(d+i)!}$$

p (p-value) é a probabilidade de encontrar os valores (ou valores mais extremos) observados na tabela de contingência i.e. ao longo da diagonal *a, d.*

Significant Patterns (3)
(Múltiplas Hipóteses)

- Problema das Multiplas Comparações. Risco de erro tipo I é não mais do que *α*.
- Probabilidade de ocorrer um erro de tipo I aumenta com o número de testes. Para n testes $\alpha_{real} = 1 - (1 - \alpha)^n$
- Usar Ajustamento de *Bonferroni* (corrigir α para n testes como sendo $\kappa = \alpha/n$) – crivo demasiado fino!
- Usar Ajustamento de *Holm* (k em vez de α).
  - Requer ordenação crescente dos p-values e ter disponíveis todos estes valores antes de determinar valor de ajustamento (k).
  - Para n testes,

$$k = \max(p_i : \forall_{1 \leq j \leq i} p_j \leq \frac{\alpha}{n - j + 1})$$

Significant Patterns (4)
(Implementação Caren)

- Usar Ajustamento de *Bonferroni* (corrigir α para n testes como sendo $\kappa = \alpha/n$).
- Usar layered critical values,
- Em vezes de um cutoff global que corrige o *α* inicial, obter vários α'_L para cada nível L.

$$\alpha'_L = \frac{\alpha}{(L_{\max} \times S_L)}$$

Onde S_L é o nº de regras possíveis de gerar no dataset dado com L items no antecedente, L_{max} é o nº máximo de items permitido no antecedente de uma regra. Temos a garantia que:

$$\alpha \geq \sum_{L=1}^{L_{\max}} \alpha'_L \times S_L$$

Pruning no CAREN

```
> caren student_courses.bas  0.1  0.5  -s,  -ovrt1  -hCC412
Sup = 0.12796   Conf = 1.00000    CC412    <--    CC447  &  CC410
Sup = 0.12796   Conf = 1.00000    CC412    <--    CC447  &  CC410  &  CC411
Sup = 0.11374   Conf = 1.00000    CC412    <--    CC447  &  CC410  &  CC420
Sup = 0.11374   Conf = 1.00000    CC412    <--    CC447  &  CC410  &  CC413
Sup = 0.11374   Conf = 1.00000    CC412    <--    CC447  &  CC410  &  DIP463
Sup = 0.11374   Conf = 1.00000    CC412    <--    CC447  &  CC410  &  DIP463  &  CC411
Sup = 0.11374   Conf = 1.00000    CC412    <--    CC447  &  CC410  &  CC413  &  CC411
Sup = 0.11374   Conf = 1.00000    CC412    <--    CC447  &  CC410  &  CC420  &  CC411.
.
.

> caren student_courses.bas  0.1  0.5  -s,  -ovrt2  -hCC412  -imp0.0000000000000001
Sup = 0.12796   Conf = 1.00000    CC412    <--    CC447  &  CC410
Sup = 0.10427   Conf = 1.00000    CC412    <--    CC450  &  CC442  &  CC411
Sup = 0.10427   Conf = 1.00000    CC412    <--    CC583  &  CC432  &  CC442  &  CC411
Sup = 0.27962   Conf = 0.98333    CC412    <--    CC421  &  CC442  &  CC411  &  CC413
.
.
```

Pruning no CAREN

```
> caren student_courses.bas  0.1  0.5  -s,  -ovrt3  -hCC412  -fisher
Sup = 0.12796   Conf = 1.00000    CC412   <--   CC447  &  CC410
Sup = 0.20853   Conf = 0.97778    CC412   <--   CC447  &  CC411  &  CC422  &  CC420
Sup = 0.26540   Conf = 0.96552    CC412   <--   CC410  &  DIP461
Sup = 0.25118   Conf = 0.96364    CC412   <--   CC447  &  CC411
Sup = 0.23697   Conf = 0.96154    CC412   <--   CC410  &  CC421
Sup = 0.30806   Conf = 0.95588    CC412   <--   CC410  &  CC413
.
.
.
```

Dados não Categóricos

(tratamento durante a geração dos itemsets)

- Em formato atributo/valor os atributos numéricos (ou de uma grandeza não categórica, como ex: hierarquias) podem dar origem a inúmeros items.
- Tende a gerar muitas regras e demasiado especificas, muitas vezes sem valor de interesse. Em vez de:

class=1 $\leftarrow$ colesterol = high & age=29

class=1 $\leftarrow$ colesterol = high & age=32

class=1 $\leftarrow$ colesterol = high & age=41

Deviamos ter:

class=1 $\leftarrow$ colesterol = high & age $\in$ [29,41]

- *Catch 22 situation*: items de intervalos pequenos implica suportes baixos o que leva à não geração de certas regras.
- Por outro lado, intervalos grandes implica regras de confiança baixa. Juntar valores de um atributo num intervalo leva à perda de informação!

Tratamento de valores numéricos

- Em Pré processamento:
  – Discretização em intervalos de valores. Ex: criar intervalos onde é preservado o valor de classe.
  – Binarização; cada atributo é convertido em dois valores. Há a selecção de um valor de corte.
- Durante o processamento (árvores de decisão):
  – Binarização: Seleccionar um valor de corte entre os valores do conjunto associado à sub-árvore. O valor escolhido é aquele que maximiza ganho! (e é sempre um que está na transição entre valores de classe).
  – Aplicação recursiva deste princípio.

Dados Numéricos

Age	**Colest**	**Blood**	**Class**
23	High	2.33	A
24	Low	2.39	B
27	High	2.21	A
27	Low	1.09	A
29	Low	2.02	A
30	Low	2.98	C
31	Low	3.01	C
31	High	1.98	B
33	low	2.09	B

Discretização Supervisionada: Atributo especial comando o processo.
Ex: **Age**: [23-23],[24-24],[27-29],[30-31],[33-33]
Ou **Age** < 29, **Age** ≥ 29.

Não supervisionada: O processo é independente dos outros atributos.
Ex: **Age**:[23-27],[29-31],[33-33]

Discretização

- Supervisionada:
  - Fayyad & Irani: Entropy oriented
  - Class intervals (caren)
  - Chi-Merge
- Não supervisionada:
  - Equi-depth (intervalos de igual suporte)
  - Equi-width (intervalos de igual largura)
  - Srikant (caren)
  - K-means

SubGroup Mining

- Estudar uma propriedade dentro de uma população e.g. colesterol
- Identificar subgrupos em que os valores ou estatísticas dessa propriedade são desviantes, surpreendentes ou interessantes

- Propriedade pode ser numérica, categórica, descrita na forma de um contraste ou de uma restrição.

Framework para SubGroup Mining

- Derivar subgrupos usando algoritmos de regras de associação;
- Algoritmos são *rule-based.*
- Detetar desvios (*interest*) usando significância estatística;
- Controle de especialização (*overfitting*) usando o mesmo tipo de teste estatístico;
- Vários tipos de regra dependendo do contexto de aplicação.

Identifying Interesting Subgroups

- Derivar regras para identificar subpopulações *interessantes* que ocorrem nos dados estudados

Subgroup_describing_characteristics $\rightarrow$ poi

- A (poi) no consequente pode ser uma atributo categórico numérico, uma restrição ou um contraste! Várias estatísticas são calculadas para cada regra.

Análise de Propriedades Numéricas

- Por vezes é interessante analisar a distribuição dos valores de um atributo numérico.
- Queremos identificar subpopulações que se distinguem em relação à população geral por uma caraterística particular do atributo numérico e.g. distribuição.
- Aplicações naturais em dados médicos.

Geração de Regras de Associação para propriedades de interesse numéricas

Ideia geral: Ter regras em que o consequente é a representação de uma propriedade numérica.

Exemplos:

Sex=female ➔ Wage: mean=$7.9 (overall mean=$9.02)

non-smoker & wine-drinker ➔ life-expectancy=85 (overall=80)

Regras de Associação com propriedades numéricas (cont)

- Várias propostas
  - Quantitative Association Rules (Aumann & Lindell99)
  - Impact Rules (Webb 2001)
  - Distribution Rules (Jorge & Azevedo 2006)
- Ideia comum a todas as propostas:

  Gerar regras que representam o comportamento de uma propriedade numérica num sub população *interessante*. Diferentes propostas de noção de regra interessante.

Regras de Associação com propriedades numéricas (cont)

- Noção de sub população interessante.
  - QAR, usa um *z-test* para confirmar interesse (validade) da regra. z-test entre $mean_J(Tx)$ e mean(D-Tx) com $\alpha=0.05$.

Complemento de Tx

Regras do tipo: subset(X) → $Mean_J(Tx)$ onde

$Mean_J(Tx) \neq Mean(D-Tx)$

z.test(μ_0,*observ*,σ): Usado para encontrar diferenças significativas entre as médias μ_0 e média da amostra.

Calcula a probabilidade de a média de uma amostra obtida assumindo a média e desvio padrão da população (μ_0 e σ) seja maior do que a média observada - assume distribuição Normal!

Permite concluir se a amostra pertence à população em estudo.

Regras de Associação com propriedades numéricas (cont)

Impact Rules (Webb)

• Interesse refere-se à noção de impacto. Optimização pesquisando-se impact rules que maximizam a definição de *impacto*.

• Uso de *t-test* para avaliar significância: tende para o *z-test* com o aumento do número de graus de liberdade. Mais adaptado para amostra pequenas. Procedimento de comparação de médias.

Média da prop. de interesse

• Noção de Impacto:

$$\mathrm{Impact}(IR) = (Mean(IR) - \overline{poi}) \times \left|\mathrm{cover}(ant(IR))\right|$$

Distribution Rules

• O consequente é uma distribuição,

• Uso do *teste Kolmogorov-Smirnov*, para avaliar se regra é interessante.

• Noção de interesse: Regra é interessante se o *p-value* do

Uma Distribuição Referência e.g. Distribuição da população geral.

Distribuição do subgrupo da regra

ks-test(*apriori*,rules-dist) < α

for menor que o valor α dado pelo utilizador.

Ideia Geral

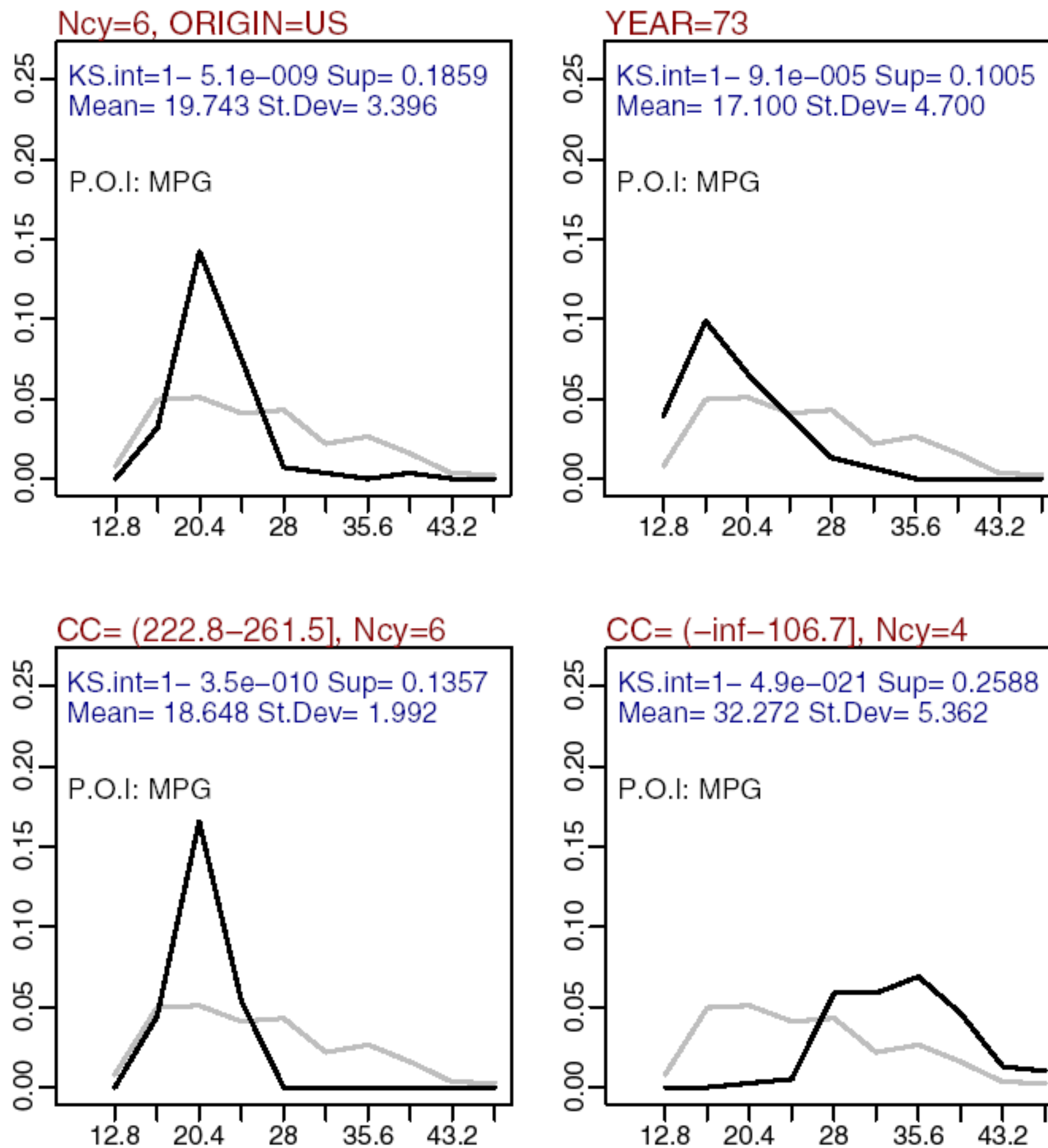
- Gerar regras de associação em que o consequente é a distribuição da propriedade numérica a estudar e o antecedente a descrição da sub população.
- Comparar distribuição *apriori* (da população referência) com a distribuição do sub grupo (via ks-test()).
- Ex: Ant-Sup = 0.14482

IDADE={46/1,48/1,51/2,52/2,54/1,55/1,57/2,58/1,59/3,60/2,61/2,62/2,63/3,64/4,65/4,66/4,67/3,68/4,69/2,70/6,72/6,73/4,75/3,76/7,77/5,78/3,79/1,80/2,81/1,82/4,83/2,84/3,86/3,90/1 } ← ***TAVC=1*** & ***DIAB=0***

Descreve a distribuição da *IDADE* para a sub população (que representa 14,4% dos indivíduos estudados) que teve o tipo de AVC 1 e não é diabética.

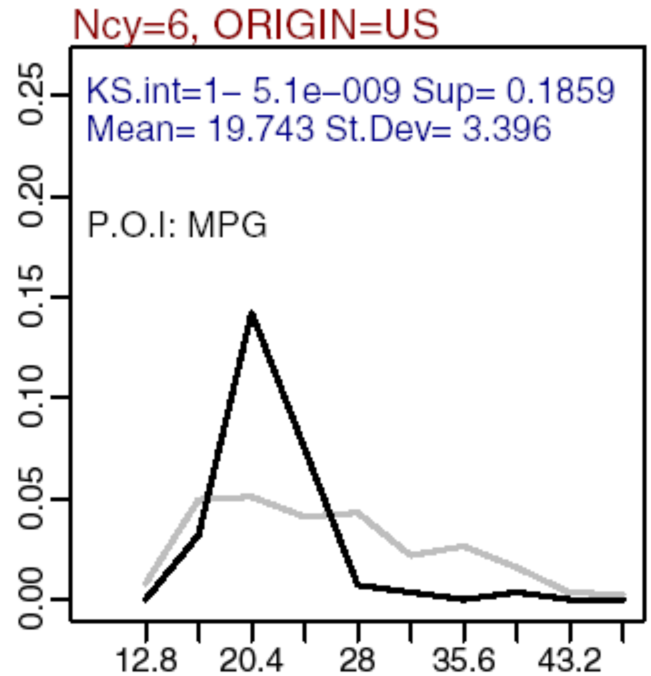
Distribution Rule presentation

- property of interest
- each DR is a plot
- distribution plot
  - frequency polygon
  - static binning
- distribution statistics
- comparison with default distribution



Medir o interesse de uma DR

- KS-interest:
  - Given a rule **A→y=D$_{y|A}$**, its KS-interest is 1-p,
  - p is the p-value of the KS test comparing **D$_{y|A}$** and **D$_{y|\varnothing}$**



- KS-improvement
  - value added by the refinements of a rule
  - imp(A→B) is

    min({KS-interest(A→B) - KS-interest(As→B) | As $\subseteq$A})

Há uma forma alternativa de controlar o overfitting via teste KS.

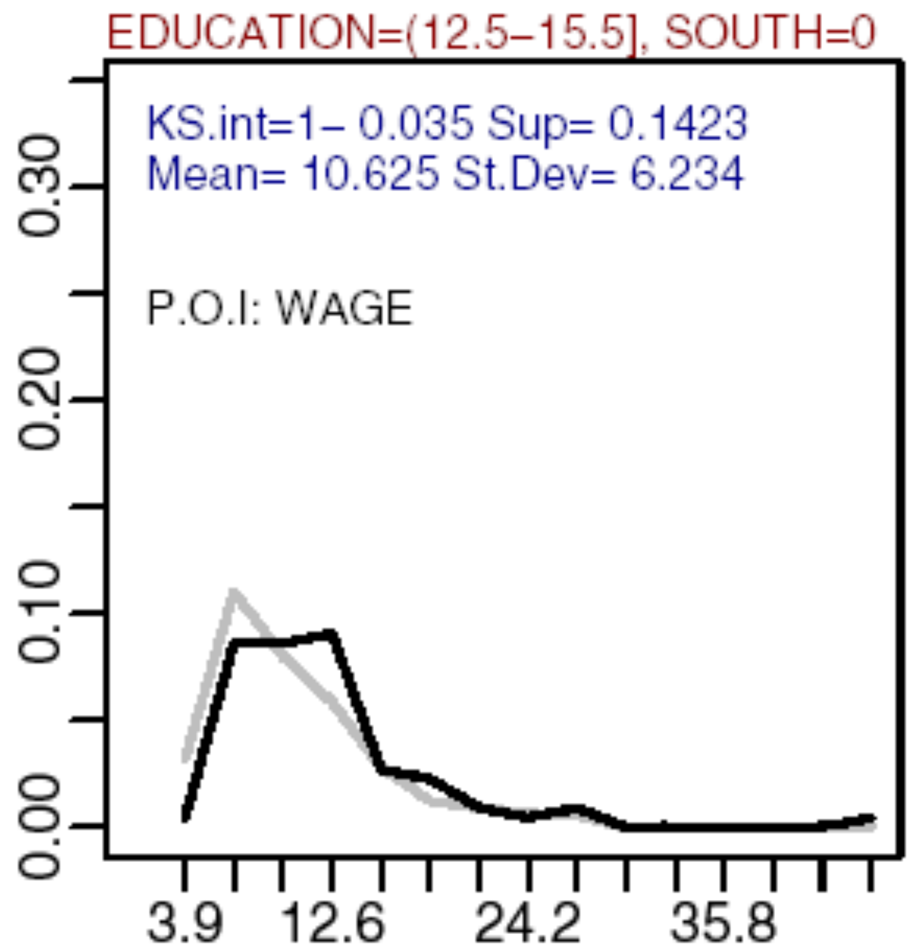
Aplicações de Regras de Distribuição

- Descripitive data mining
  - dataset: Determinants of Wages from the 1985 Current Population Survey in the United States, a.k.a. Wages
  - property of interest: WAGE
- Rule discovery
  - min-sup=0.1, KS-int=0.95
  - minimal KS-improvement of 0.01
  - numerical attributes in the antecedent were pre-discretized
  - compact internal representation of rules
  - rules can be output as text or graphically

```
Sup=0.118 KS.int=1-0.0085 Mean=10.982 St.Dev=6.333
EDUCATION=(12.5-15.5] & SOUTH=0 & RACE=3
  ->  WAGE={ 3.98/1,4.0/1,4.17/1,4.5/1,4.55/1,4.84/1,5.0/1,5.62/1,5.65/1,5.8/1,6.0/1,6.25/4,7.14/1,7.5/1,7.67/1,7.7/1,7.96/1,
           8.0/2,8.4/1,8.56/1,8.63/1,8.75/1,8.9/1,9.22/1,9.63/1,9.75/1,9.86/1,10.0/3,10.25/1,10.5/1,10.53/1,10.58/1,10.61/1,
           11.11/1,11.25/2,12.0/1,12.47/1,12.5/4,13.07/1,13.75/1,13.98/1,14.29/1,15.0/1,16.0/1,16.14/1,16.42/1,17.25/1,17.86/1,
           18.5/1,21.25/1,22.5/1,26.0/1,44.5/1 }
```

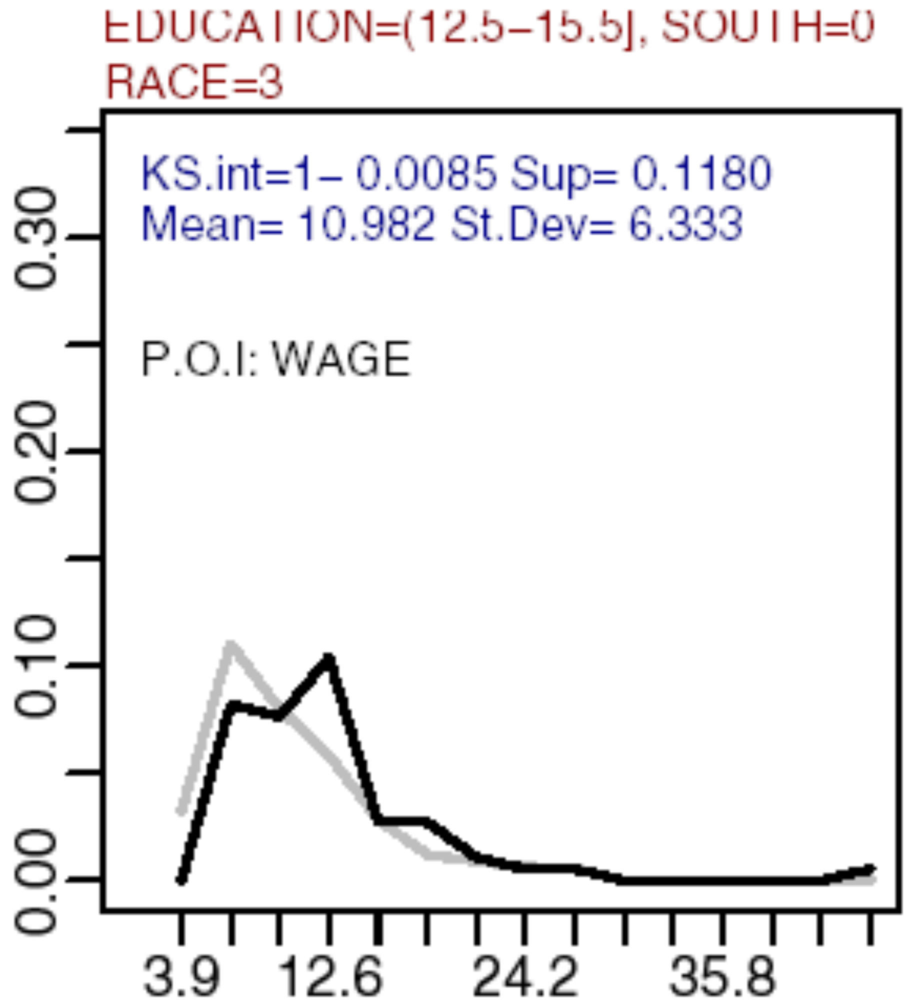
Using Distribution Rules

- antecedent
  - people with 13 to 15 years of education
  - not from the south
- consequent
  - wage distribution is better than the whole population but still concentrated on the same interval



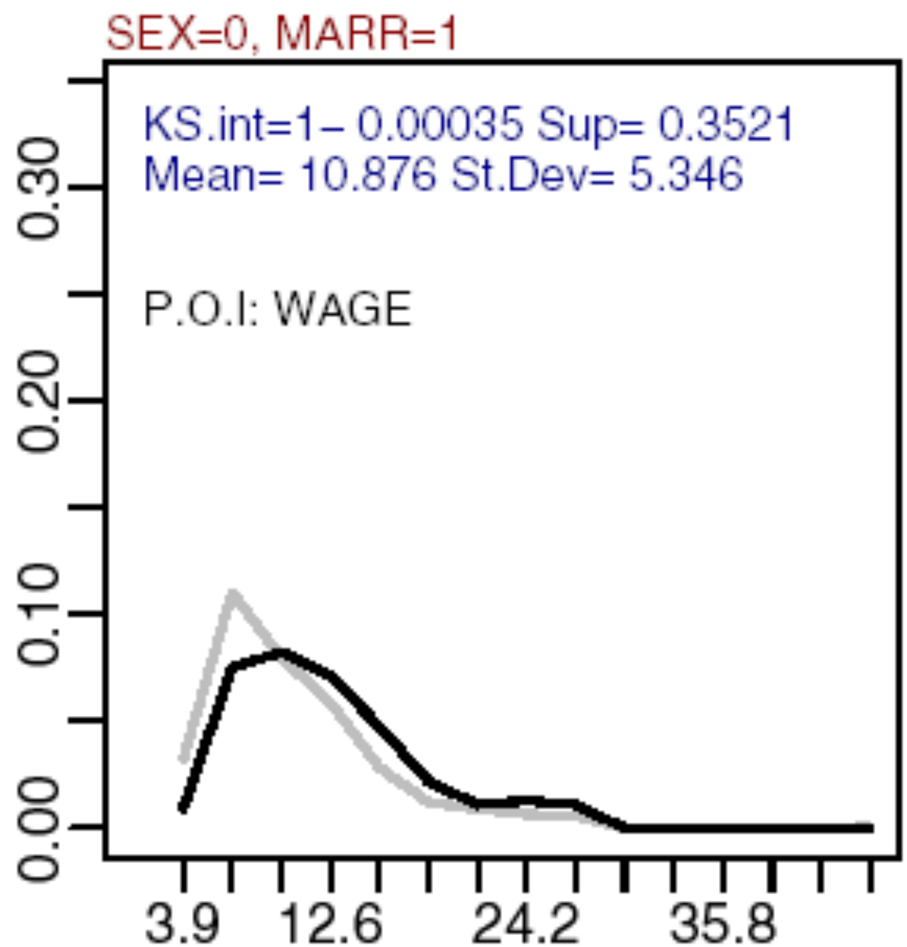
Using Distribution Rules

- antecedent
  - refinement of previous
  - race is white
- consequent
  - wage distribution is even better than before
  - KS-improvement is higher than 0.01
  - the wages still are concentrated on the same interval as before



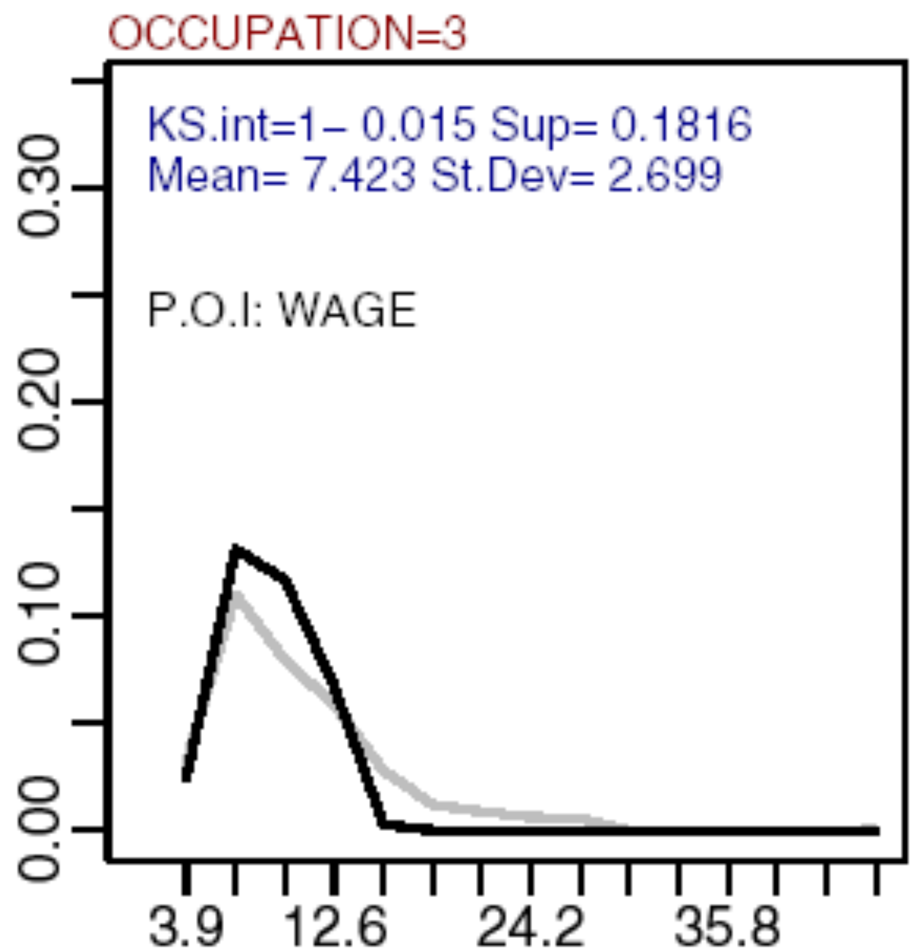
Using Distribution Rules

- antecedent
  - married males
- consequent
  - less interesting
  - still signif. different



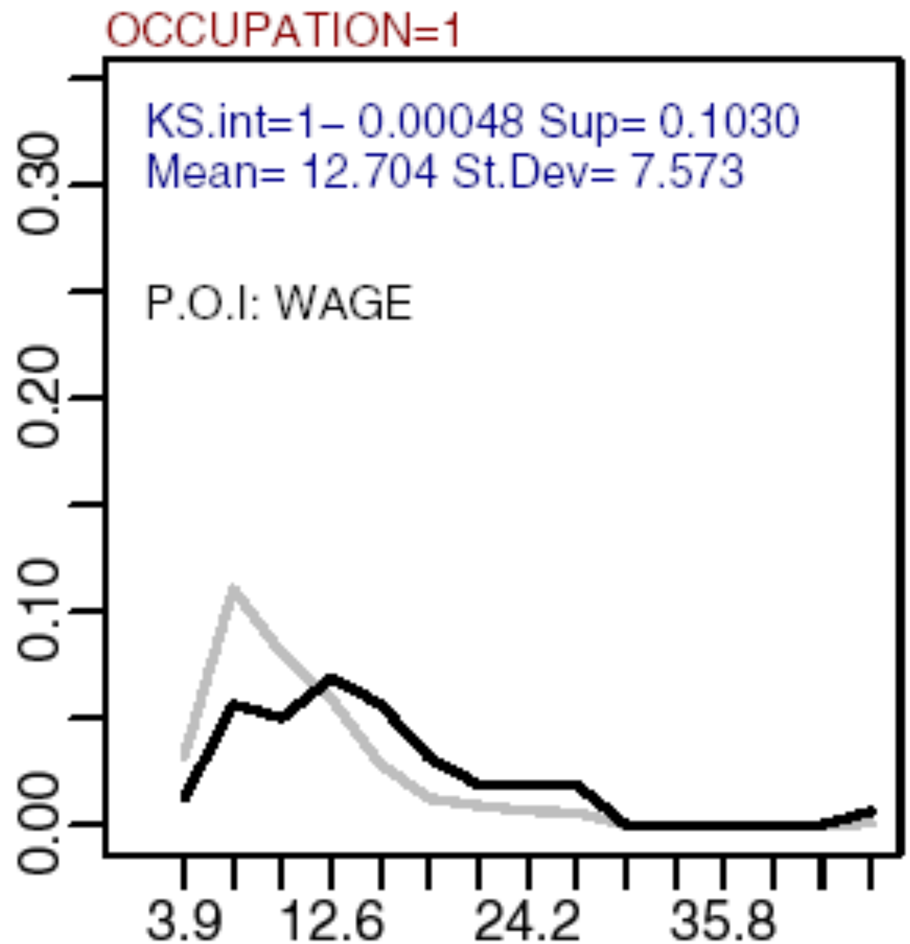
Using Distribution Rules

- antecedent
  - Occupation=Clerical
- consequent
  - concentrated on lower income



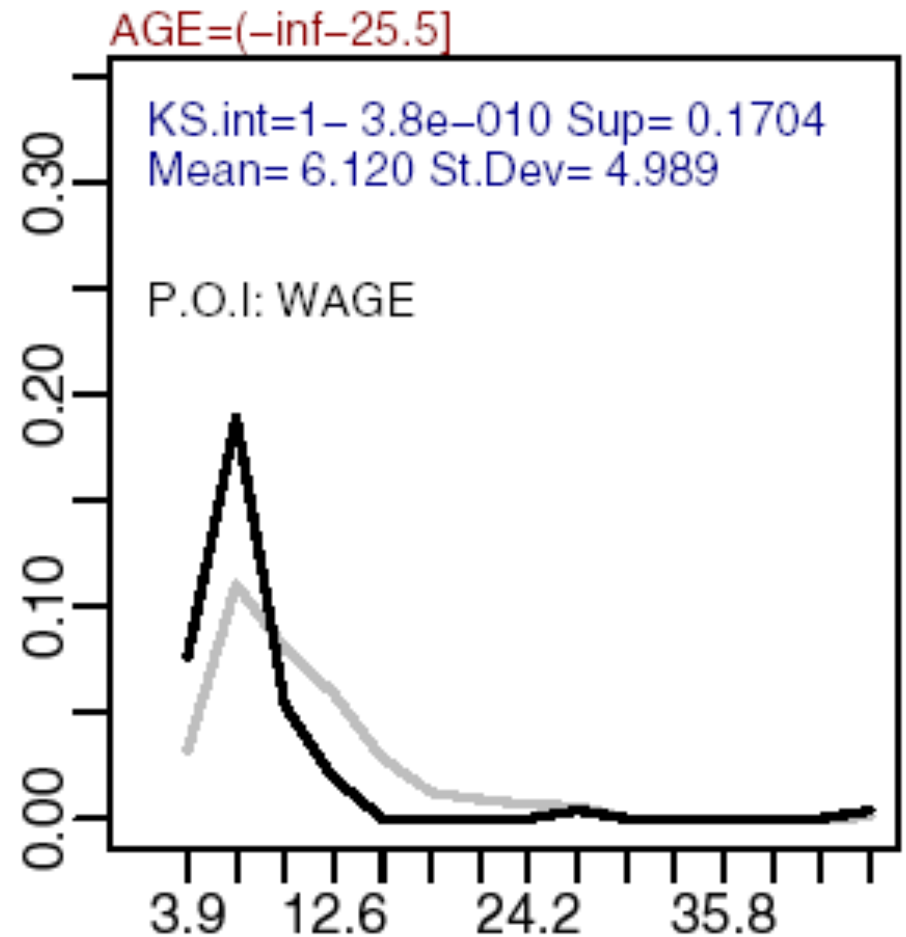
Using Distribution Rules

- antecedent
  - Occupation=Management
- consequent
  - clearly better wage distribution
  - we also observe a slightly lifted right tail

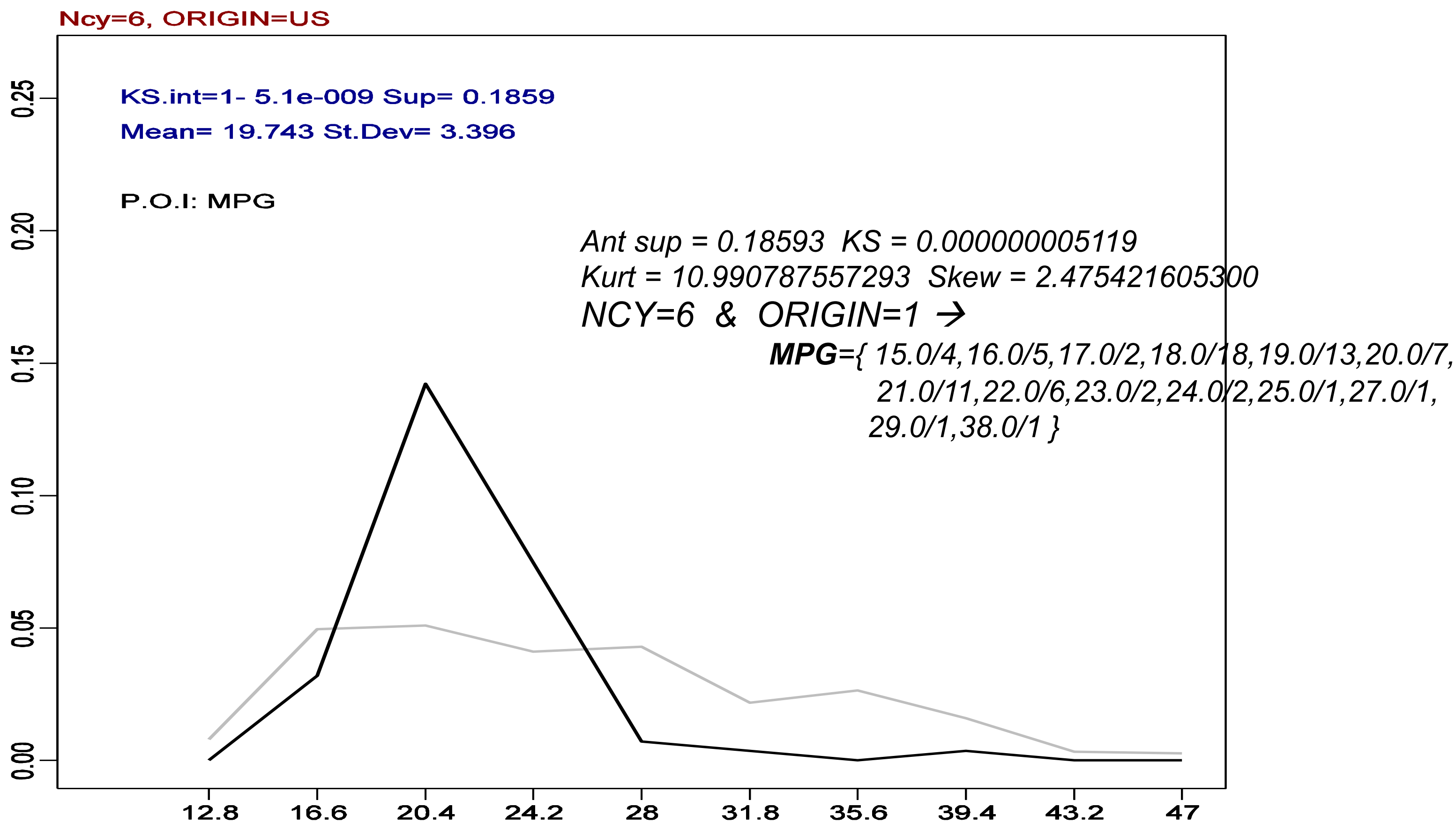


Using Distribution Rules

- antecedent
  - young people
- consequent
  - lower wages, very concentrated
  - some secondary modes are suggested

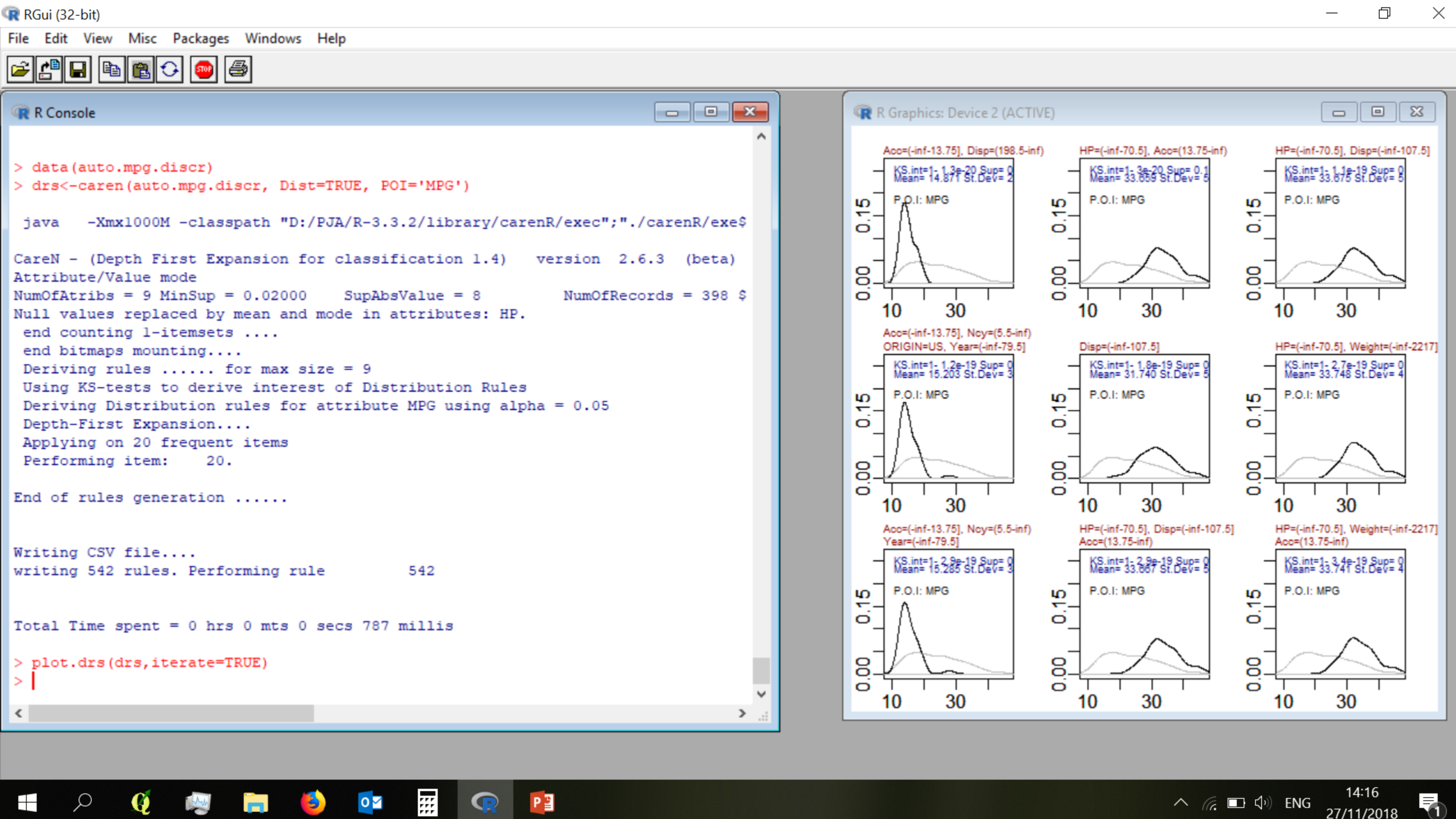


Análise de Distribuições



Aplicar testes de "*Goodness of Fit*". (e.g. Kolmogorov-Smirnov). Compara distribuições segundo parâmetros tipo *Skewness* (grau de assimetria) e *Kurtosis* (grau de afunilamento)

Geração e Visualização no carenR



Contrast Sets

- *Rules for Contrast Sets* [Azevedo2010]
- Descreve a diferença entre grupos de contraste.
- Um *contrast set* é um conjunto de características que descreve a subpopulação que ocorre com diferentes proporções ao longo dos diferentes grupos em estudo.
- Exemplos:
  - Diferentes instâncias temporais (vendas em 1998 versus 1999),
  - Diferentes localizações (encontrar distintas caraterísticas para a localização do gene *x* em DNA humano versus em DNA de ratos),
  - Ao longo de diferentes classes (diferenças entre loiras e morenas).

RCS

- As caraterísticas da subpopulação a encontrar (contrast set) são *interessantes* (significantes) se as proporções das ocorrências individuais ao longo dos grupos estudados são distintas em valores significativos!
- i.e. subpopulação não é *independente de pertença a um grupo*. Significância é calculada usando um teste exato de Fisher.

No nosso caso definidos pelo atributo **education**

```
Gsup = 0.17191 | 0.04121 p = 1.1110878451E-017     education=Doctorate >> education=Masters
Gsup = 0.17191 | 0.01681 p = 3.0718399575E-040     education=Doctorate >> education=Bachelors
Sup(CS) = 0.03097                                    ← workclass=State-gov  &  class > 50K.
```

- Especialização do *contrast set* é também controlado por um teste de Fisher.

RCS (2)

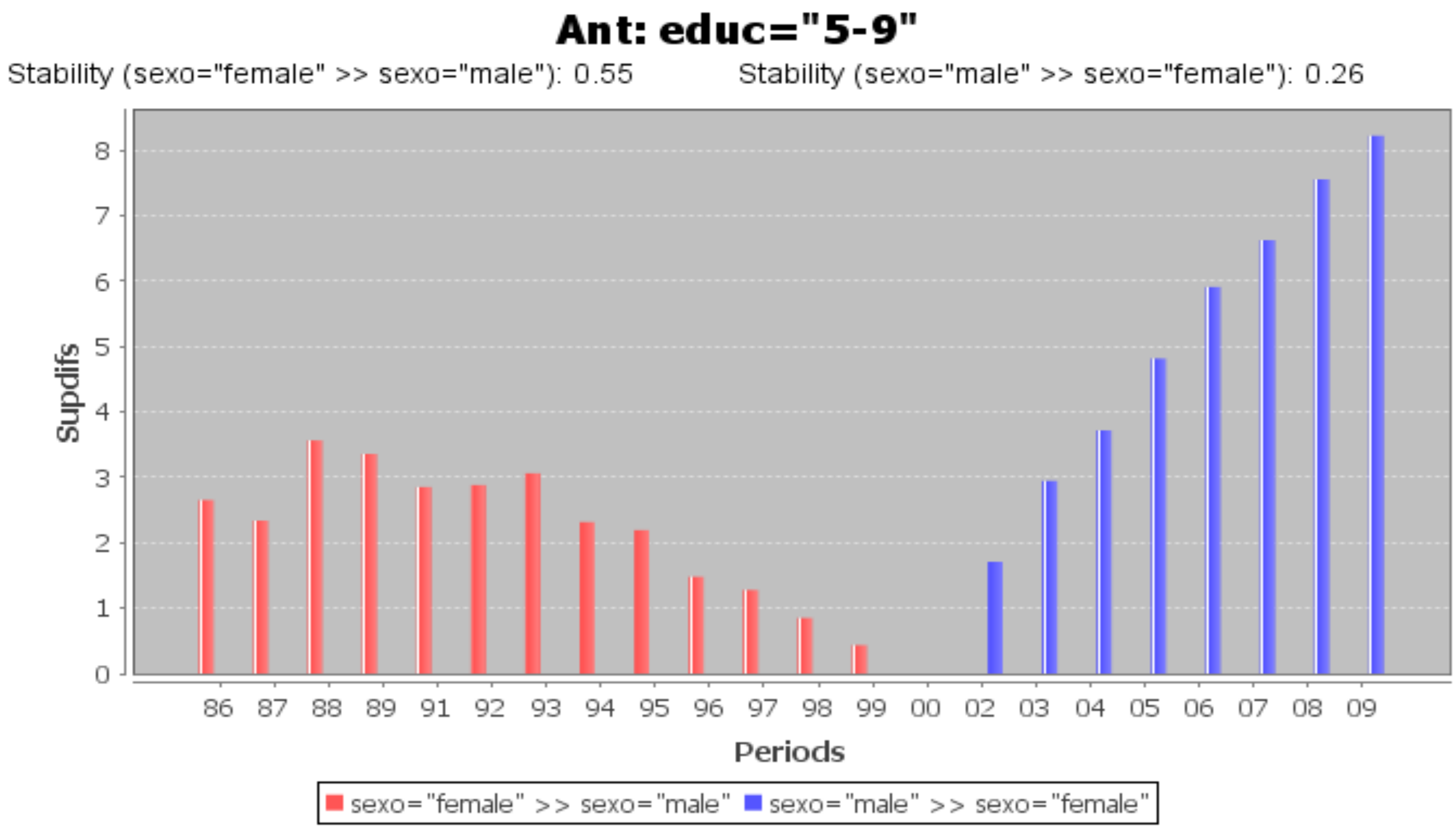
Contingency table for cs versus $\emptyset$

	G_i	G_j	$\sum row$
cs	$sup(cs, G_i)$	$sup(cs, G_j)$	
$\emptyset$	$n_i - sup(cs, G_i)$	$n_j - sup(cs, G_j)$	
$\sum column$	n_i	n_j	$n_i + n_j$

- Teste de Fisher sobre os dois universos do contraste nos dois grupos. n_i e n_j são as cardinalidades dos grupos.
- Mesmo teste usado para verificar significância entre um contraste cs e uma sua especialização $cs + \{d\}$.

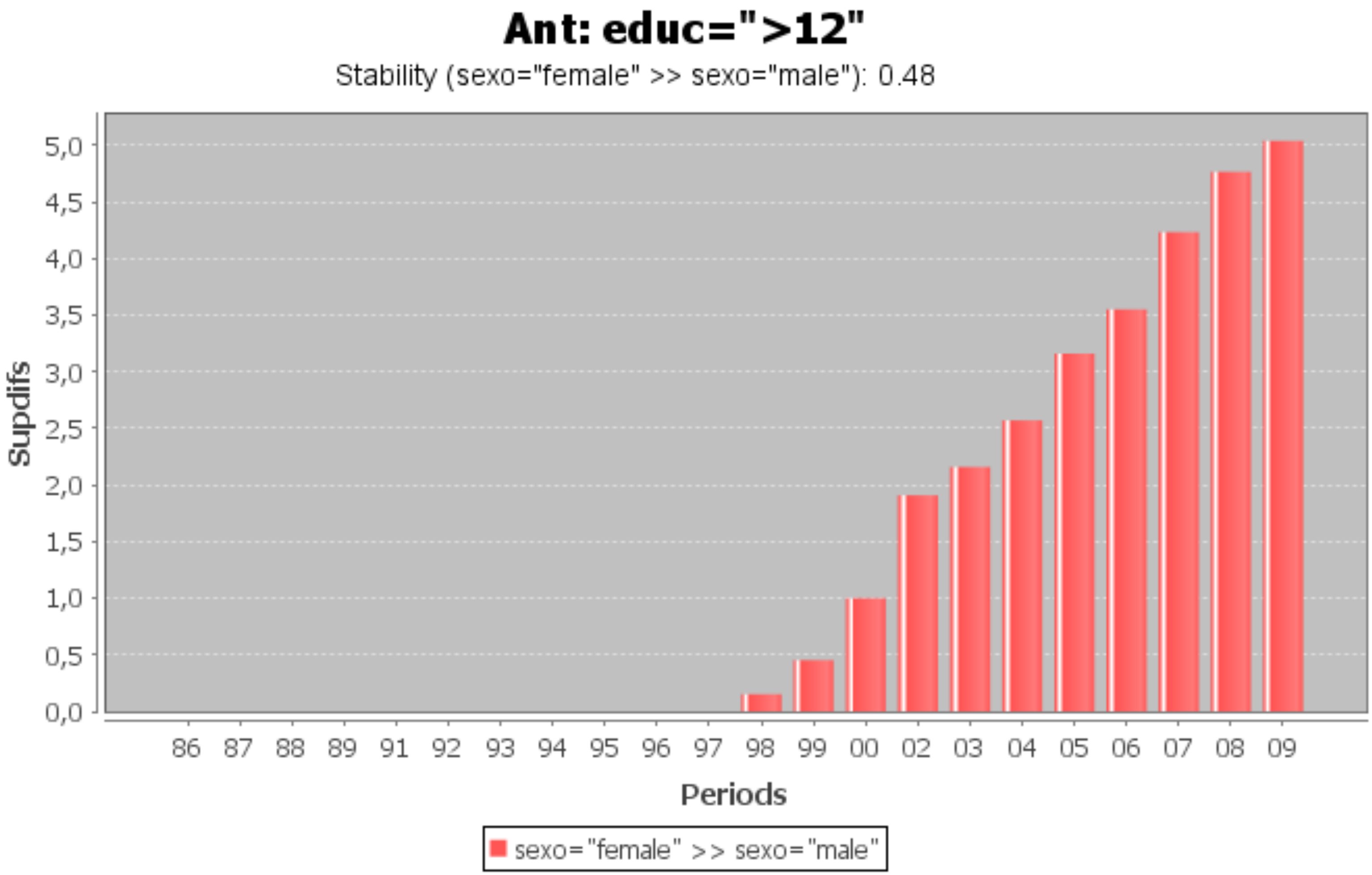
Case Study

Data representing employment from the Portuguese private sector between 1986 and 2009.



- *Contrast on individuals with basic (lower) education*

Case Study



- *Contrast found on individuals with higher education*

Exemplos (Caren)

D:\PJA\caren>java caren adult.data 0.01 0.5 -s, -Att -heducation=Masters,education=Bachelors,education=Doctorate -CS -ovrt -Discfieducation.num,age,hours_per_week -classclass

$$\Phi = \frac{(a\times c)-(b\times d)}{\sqrt{(a+c)(b+d)(a+b)(c+d)}}$$

Obs = 000271 | 000000 Gsup = 0.05061 | 0.00000 p = 1.0873415499E-009 phi = 0.06166 education=Bachelors >> education=Doctorate
Obs = 000271 | 000005 Gsup = 0.05061 | 0.00290 p = 3.7322820782E-027 phi = 0.10576 education=Bachelors >> education=Masters
Sup(CS) = 0.05043 <-- age=]21.5000 : 23.5000]

Mede correlação entre os grupos e o contrast set

Obs = 000089 | 000169 Gsup = 0.21550 | 0.09808 p = 5.1718928543E-010 phi = 0.14229 education=Doctorate >> education=Masters
Obs = 000089 | 000270 Gsup = 0.21550 | 0.05042 p = 9.3030477682E-028 phi = 0.17617 education=Doctorate >> education=Bachelors
Obs = 000169 | 000270 Gsup = 0.09808 | 0.05042 p = 7.1168654312E-012 phi = 0.08481 education=Masters >> education=Bachelors
Sup(CS) = 0.03986 <-- workclass=State-gov

Obs = 000233 | 000419 Gsup = 0.56416 | 0.24318 p = 1.0015025087E-034 phi = 0.27527 education=Doctorate >> education=Masters
Obs = 000233 | 000579 Gsup = 0.56416 | 0.10812 p = 1.9540669064E-100 phi = 0.33808 education=Doctorate >> education=Bachelors
Sup(CS) = 0.05709 <-- occupation=Prof-specialty & class=>50K

Obs = 001795 | 000073 Gsup = 0.33520 | 0.17676 p = 2.5752271157E-012 phi = 0.08730 education=Bachelors >> education=Doctorate
Obs = 001795 | 000404 Gsup = 0.33520 | 0.23447 p = 7.9263736218E-016 phi = 0.09341 education=Bachelors >> education=Masters
Sup(CS) = 0.32809 <--- marital.status=Never-married

Conclusões

- Algoritmos para calcular associações entre elementos atómicos nos dados (items);

- Geração de Regras que descrevem associação entre elementos atómicos dos dados;

- Seleção de regras interessantes e significativas;

- Tratamento de dados não categóricos;
- Análise de propriedades de interesse numéricas.