Tutorial Sobre Regras de Associação

Ronaldo Ramos

1 de dezembro de 2016

Resumo

Neste texto mostramos alguns conceitos associados as chamadas Regras de Associação (RA) e também como construí-las a partir de um conjunto de dados (dataset) utilizando o algoritmo "A Priori". RA(s) são utilizadas em sistemas de recomendação, definição de layouts de supermercados e lojas, recomendação de serviços bancários e comerciais, identificação de perfis de usuários de cartão de crédito, etc. Será mostrado um caso específico de aplicação do algoritmo na solução do problema da cesta do supermercado (market basket analysis) que cobre desde a descoberta dos chamados conjuntos de itens frequentes até a saída final das regras que foram "mineradas".

(Palavras-Chave: Mineração de Dados, Regras de Associação, Aprendizagem de Máquina.)

1 Regras de Associação

1.1 Introdução

Uma regra é uma entidade que relaciona dois elementos chamados de antecedente e consequente na forma de uma implicação material. Normalmente representada na forma: $A \rightarrow B$ onde o termo A representa o conjunto de antecedentes e B o conjunto de consequentes da regra.

Em mineração de dados (*data mining*), as regras são classificadas em dois tipos: regras de classificação e regras de associação. As regras de classificação costumam levar, em seus antecedentes, elementos relacionados aos atributos de objetos dentro de um domínio de estudos enquanto o consequente relaciona as classes desse domínio. Tratam-se portanto de mecanismos de inferência que classificam ou categorizam um objeto a partir de seus atributos.

Pegando os exemplos do dataset das flores[1], poderíamos construir algumas regras tais quais as que seguem. Estas

> **Regra 1**:

SE (4,40 < SL < 5,70) E (2,92 < SW < 4,18) E
(1,12 < PL < 1,86) E (0,10 < PW < 0,48)
ENTÃO CLASSE = SETOSA.

> **Regra 2**:

SE (5,00 < SL < 6,88) E (2,20 < SW < 3,30) E
(3,08 < PL < 4,98) E (1,00 < PW < 1,68)
ENTÃO CLASSE = VERSICOLOR.

> **Regra 3**:

SE (5,62 < SL < 7,70) E (2,50 < SW < 3,76) E
(4,80 < PL < 6,70) E (1,50 < PW < 2,50)
ENTÃO CLASSE = VIRGINICA.

regras possuem em seus antecedentes, também chamados de lado esquerdo da regra, as condições de testes sobre os atributos das íris. No caso em análise, os atributos a serem testados são o comprimento da sépala (SL), comprimento da pétala (PL), largura da sépala (SW) e largura da pétala (PW). Um objeto em análise (flor) que possua atributos que casem com os valores colocados nos antecedentes destas regras, poderá ser classificado em uma das seguintes categorias: SETOSA, VERSICOLOR e VIRGÍNICA.

[1] Iris dataset - Disponível na UCI. Muito usado em cursos e treinamentos na área

As regras acima, em conjunto, funcionam como um classificador das iris de acordo com os atributos, que no caso são as dimensões das pétalas e sépalas. Daí receberem o nome de regras de classificação (RC).

Regras de Associação são mecanismos de representação do conhecimento que diferem sutilmente das chamadas regras de classificação. Elas também representam uma relação lógica entre os elementos dispostos do lado direito e do lado esquerdo da regra em uma forma semelhante a implicação material da lógica clássica, porém, ao contrário das regras de classificação, tanto o lado esquerdo quanto o direito da regra possuem apenas atributos.
Ex. **SE** casca = "RUGOSA" **E** frutaCítrica ="SIM" **ENTÃO** cheiroForte = "SIM"

Vamos ver agora alguns conceitos relacionados ao confronto entre um conjunto de regras de associação e um *dataset* específico.

Dada uma Regra **SE** *ESQUERDA* **ENTÂO** *DIREITA* e um dataset com **N** instâncias. Seja também:

N_{esq}	O número de instâncias que casam com a parte esquerda da regra
N_{dir}	O número de instâncias que casam com a parte direita da regra
N_{amb}	O número de instâncias que casam com ambos os lados da regra

Podemos definir os seguintes termos que serão úteis mais na frente quando tratarmos de medida de qualidade ou de interesse das regras de associação.

Confiança, Acuracidade ou Confiabilidade. *Proporção entre o número de instâncias que casam ambos os lados em relação aos que casam apenas o lado esquerdo.* N_{amb}/N_{esq}

Suporte. *Proporção entre as instâncias que casam ambos os lados com o total.* N_{amb}/N. *Tecnicamente o suporte é uma proporção entre um conjunto de regras consideradas e o total. Ex.* $Sup(A) = N_A/N$ *ou* $Sup(A \rightarrow B) = N_{A \rightarrow B}/N$

Completude. *Proporção entre as instâncias que casam ambos os lados com e as regras que casam o lado direito.* N_{amb}/N_{dir}

Lift (Elevação ou Aperfeiçoamento) . *Mede a correlação entre ambos os lados de um conjunto de regras.*

Seja a regra **A** : **(X)** $\rightarrow$ **(Y)** onde **(X)** e **(Y)** são conjuntos de atributos. **X** : (a,b,c,d...) e **Y** : (x,y,z,k,w,...)

$$lift((\mathbf{X}) \rightarrow (\mathbf{Y})) = \frac{Sup((\mathbf{X}) \rightarrow (\mathbf{Y}))}{Sup(\mathbf{X}) * Sup(\mathbf{Y})}$$

Obs. $Lift > 1$ implica em correlação positiva.

Em problemas práticos costuma-se limitar o uso de regras que possuam: Suporte > 0,01 (1%) Confiança > 0,80 (80%) e lift > 1. Logicamente estes valores podem variar de acordo com a vontade do analista de dados.

1.2 Cálculo das Propriedades de Interesse

A tabela 1 nos mostra um dataset hipotético que usaremos como referência para a realização dos cálculos das propriedades definidas na seção 1. O conjunto de Itens (I) é composto de {Maizena,Fralda,Sabão ,Cerveja,Arroz,Suco} e portando toda transação deve ser suconjunto deste conjunto de itens (I).

Vamos realizar os seguintes cálculos:

1. Calcular o Suporte de (Sabão $\rightarrow$ Cerveja).
2. Calcular a Confiança de (Sabão $\rightarrow$ Cerveja)
3. Calcular o Lift de (Sabão $\rightarrow$ Cerveja)
4. Calcular o Suporte de (Sabão e Suco $\rightarrow$ Arroz)

5. Calcular a Confiança de (Sabão e Suco → Arroz)
6. Calcular o Lift de (Sabão e Suco → Arroz)

Tr.	Maizena	Fralda	Sabão	Cerveja	Arroz	Suco
1	S	S	N	N	N	N
2	N	N	S	S	N	N
3	N	N	S	N	S	S
4	S	S	S	N	N	S

Tabela 1: Dataset Hipotético

Solução do problema 1. A partir dos dados da tabela 1 verificamos primeiramente o número de transações que corresponde exatamente à quantidade de linhas da tabela e que se costuma chamar, no jargão da mineração de dados, de instância. Temos, portanto, quatro instâncias (N=4). A regra Sabão → Cerveja implica que existe uma associação entre esses dois itens. Classicamente significa que quem compra sabão também adquire cerveja. Suportando essa regra nós temos apenas uma instância que é exatamente a transação de número dois, ou seja, está registrado na transação de número dois que o cliente comprou sabão e também cerveja (S=SIM, N=Não).

Logo,

$N = 4$

$N_{amb} = N(Sabao \rightarrow \text{Cerveja}) = 1$

$$Sup(Sabao \rightarrow Cerveja) = \frac{N(Sabao \rightarrow Cerveja)}{N} = \frac{1}{4} \tag{1}$$

Solução do problema 2. A cálculo anterior nos deu o valor do suporte da regra que estamos analisando, mas precisamos também do número de instâncias que suportam o lado esquerdo da regra. No caso, quantas vezes o item (Sabão [2]) foi comprado. Observando o dataset verificamos que este item foi adquirido 3 vezes (instâncias 2,3 e 4).

Logo,

$N = 4$

$N_{amb} = N(Sabao \rightarrow Cerveja) = 1$

$N_{esq} = N(Sabao) = 3$

$$Conf(Sabao \rightarrow Cerveja) = \frac{N(Sabao \rightarrow Cerveja)}{N(Sabao)} = \frac{1}{3} \tag{2}$$

Solução do problema 3. O cálculo do lift requer o cálculo de três valores de suporte que são o suporte da regra, o suporte do lado direito e o do lado esquerdo.

Logo, $N = 4$

$N_{amb} = N(Sabao \rightarrow Cerveja) = 1$

$N_{esq} = N(Sabao) = 3$

$N_{dir} = N(Cerveja) = 1$

$Sup(Sabao \rightarrow Cerveja) = \frac{1}{4}$ vide eq. (1)

$$Sup(Sabao) = \frac{N(Sabao)}{N} = \frac{3}{4} \tag{3}$$

[2] Os nomes dos itens aparecerão sem acentos nas fórmulas

$$Sup(Cerveja) = \frac{N(Cerveja)}{N} = \frac{1}{4} \tag{4}$$

$$lift(Sabao \rightarrow Cerveja) = \frac{Sup(Sabao \rightarrow Cerveja)}{Sup(Sabao) * Sup(Cerveja)} = \frac{\frac{1}{4}}{\frac{3}{4} * \frac{1}{4}} = \frac{4}{3} \tag{5}$$

Solução do problema 4. Neste caso temos dois antecedentes e um consequente, ou seja, dois itens do lado esquerdo da regra e um item do lado direito. O cálculo deve ocorrer da mesma forma. Vejamos que os dois itens do lado esquerdo aparecem juntos nas transações 3 e 4.

Logo,

$N = 4$ (Número de instâncias)

$N_{amb} = N(Sabao, Suco \rightarrow Arroz) = 1$ (Transação 3)

$$Sup(Sabao, Suco \rightarrow Arroz) = \frac{N(Sabao, Suco \rightarrow Arroz)}{N} = \frac{1}{4} \tag{6}$$

Solução do problema 5. Aqui vamos repetir os mesmos passos realizados anteriormente.

Logo,

$N = 4$ (Número de instâncias)

$N_{amb} = N(Sabao, Suco \rightarrow Arroz) = 1$ (Transação 3)

$N_{esq} = N(Sabao, Suco) = 2$ (Transações 3 e 4)

$$Conf(Sabao, Suco \rightarrow Arroz) = \frac{N(Sabao, Suco \rightarrow Arroz)}{N(Sabao, Suco)} = \frac{1}{2} \tag{7}$$

Solução do problema 6. O calculo do lift é sempre um pouco mais complexo.

Logo,

$N = 4$ (Número de instâncias)

$N_{amb} = N(Sabao, Suco \rightarrow Arroz) = 1$ (Transação 3)

$N_{esq} = N(Sabao, Suco) = 2$ (Transações 3 e 4)

$N_{dir} = N(Arroz) = 1$ (Transação 3)

$Sup(Sabao, Suco \rightarrow Arroz) = \frac{1}{4}$ vide eq. (6)

$$Sup(Sabao, Suco) = \frac{N(Sabao, Suco)}{N} = \frac{2}{4} = \frac{1}{2} \tag{8}$$

$$Sup(Arroz) = \frac{N(Arroz)}{N} = \frac{1}{4} \tag{9}$$

$$lift(Sabao, Suco \rightarrow Arroz) = \frac{Sup(Sabao, Suco \rightarrow Arroz)}{Sup(Sabao, Suco) * Sup(Arroz)} = \frac{\frac{1}{4}}{\frac{1}{2} * \frac{1}{4}} = 2 \tag{10}$$

Você pode bem observar que a única coisa a ser comprada em comum por quem compra Sabão e Suco é exatamente o Arroz.

2 Algoritmo Apriori

Digamos agora que temos um *dataset* disponível. A questão que se coloca é exatamente como encontrar o melhor conjunto de regras de associação entre seus elementos. Dá pra perceber que estes dados são muito importantes para uma série de aplicações. Antes disto, vamos ver um algoritmo que nos permite identificar o que é chamado de Conjunto de Itens Frequentes (CIF). Estes conjuntos de itens frequentes são subconjuntos dos conjuntos de itens que aparecem em diversas transações. Até aí não vamos construir as regras, mas vamos identificar os subconjuntos de itens frequentes (CIF).

O algoritmo 1 mostra a formalização do mesmo. Vamos agora executar este algoritmo sobre um certo conjunto hipotético de dados. Antes porém vamos ver como o mesmo funciona.

- Seja um conjunto de itens (de um supermercado, por exemplo): $I = i_1, i_2, i_3, i_n$
- Seja o conjunto de todas as transações $T = t_1, t_2, t_3, t_n$
- Cada item t_i é subconjunto de I. $t_n \subseteq I$
- Sejam também dados os valores mínimos de suporte e confiança que podem ser admitidos para as regras.

Algoritmo 1: Descobrindo CIFs - Conjuntos de Itens Frequentes

Dados: Conjunto de Itens (D), Suporte Mínimo ou Frequência Mínima (smin), Conjunto de Transações (T)

Resultado: Conjuntos de Itens Frequentes (L)

```
1  início
2      k = 2
3      L_1 = combinações(D, 1)
4      enquanto L_{k-1}! = ∅ faça
5          C_k = combinações( L_{k-1}, k )
6          para cada E ∈ C_k faça
7              para cada S ∈ E faça
                   // Poda...
8                  se S ⊈ L_{k-1} então C_k = C_k − E
9              fim
10         fim
11         para cada t ∈ T faça
12             para cada E ∈ C_k faça
13                 se E ∈ t então  N(E) = N(E) + 1
14             fim
15         fim
16         para cada E ∈ C_k faça
17             se N(E) > smin então L_k = L_k + E
18         fim
19         k = k + 1
20     fim
21     retorna ⋃ L_k
22 fim
```

Aqui temos algumas características deste algoritmo

- O algoritmo realiza uma busca e geração de conjuntos de itens frequentes (CIF_{k+1}) de cardinalidade k+1 a partir conjuntos de itens frequentes de cardinalidade k (CIF_k).
- Cada subconjunto é expandido em um item por vez em um processo conhecido como geração de candidatos.
- Os grupos de candidatos são confrontados com os dados.
- O algoritmo identifica itens individuais frequentes no banco de dados estendendo-os a conjuntos de itens cada vez maiores a medida que estes itens aparecem com um certa frequência no banco de dados.

Transação	Itens
1	1 3 4
2	2 3 5
3	1 2 3 5
4	2 5
5	1 3 5

Tabela 2: Transações

Candidato.	Suporte.
{1}	3
{2}	3
{3}	4
{4}	**1**
{5}	4

Tabela 3: Candidatos k = 1

CIF (L)	Suporte
{1}	3
{2}	3
{3}	4
{5}	4

Tabela 4: Conjunto k = 1

- O Algoritmo tem como resultado um conjunto de itens frequentes que deverão ser usados para criar as regras de associação que representam tendências encontradas na base de dados.

É importante também saber que:

- O Algoritmo Apriori se vale do fato de que qualquer subconjunto de um conjunto de itens frequentes é um conjunto de itens frequentes. Isto implica que um item não frequente também torna não frequente um conjunto ao que o mesmo pertença.
- O Algoritmo reduz o número de candidatos explorando somente conjuntos cujo suporte seja maior que um valor mínimo preestabelecido.
- Todos os conjunto de itens não frequentes podem ser podados caso se verifique que seus subconjuntos são não frequentes.

Os passos básicos do algoritmo são:

1 Construímos uma lista de conjuntos candidatos com k itens (C_k) e extraímos daí os conjuntos de itens frequentes contendo k itens (L_k) usando o valor de suporte de cada conjunto.

2 A partir do conjunto criado na etapa 1, geramos um conjunto candidato de itens frequentes de k+1 itens.

3 Itens não frequentes são podados.

4 Repetimos o processo até que não haja mais candidatos ou conjuntos de itens (conjunto candidato vazio).

5 Retornamos uma lista de conjuntos itens frequentes de k-1 itens.

Vamos usar a tabela 2 como referencia para os cálculos que vamos realizar. Seja o conjunto $I = \{1, 2, 3, 4, 5\}$, o suporte mínimo de 2/5 e as transações conforme a tabela.

O primeiro conjunto candidato é gerado $L_{k=1}$. Trata-se de conjuntos contendo apenas um item. Colocamos ao lado da tabela os valores de suporte de cada item que podem ser facilmente calculados usando a fórmula fornecida na seção 1.

Observemos que o candidato {4} possui suporte de apenas um item que é inferior ao valor mínimo admissível. Esse elemento é excluído e gera-se então o conjunto $L_{k=1}$ que será como na tabela 4.

Agora podemos prosseguir para o próximo passo (k=2). A partir do conjunto anterior $L_{k=1}$ conseguimos o candidato $C_{k=2}$ pela obtenção de todas as combinações possíveis destes elementos (2 a 2). O Resultado está na tabela 5.

Observamos que o valor de suporte do conjunto {1,2} está abaixo do mínimo. Eliminamos este valor e chegamos ao conjunto $L_{k=2}$ que está na tabela 6

Agora passamos para o k=3. Geramos os candidatos e obtemos a tabela 7

Neste momento observamos que cada conjunto de (k=3) podem gerar combinações de (k=2) que não estão no conjunto anterior que deu origem ao mesmo. E, como um conjunto frequente tem subconjuntos frequentes e conjuntos não frequentes possuem subconjuntos não frequentes, verificamos se alguma combinação destes itens com k-1 elementos são elementos não frequentes por eventualmente terem sidos descartados ou por

Candidato	Suporte
{1,2 }	**1**
{1,3 }	3
{1,5 }	2
{2,3 }	2
{2,5}	3
{3,5}	3

Tabela 5: Candidatos k = 2

CIF (L).	Suporte
{1,3 }	3
{1,5 }	2
{2,3 }	2
{2,5}	3
{3,5}	3

Tabela 6: Conjunto final k = 2

Candidato.	Sup.
{1,2,3 }	
{1,2,5 }	
{1,3,5 }	
{2,3,5 }	

Tabela 7: Candidatos para k = 3

Candidato e Combs.	No Ant?.
{1,2,3 } : {1,2},{1,3},{2,3}	Não
{1,2,5 } : {1,2},{1,5},{2,5}	Não
{1,3,5 } : {1,3},{1,5},{3,5}	Sim
{2,3,5 } : {2,3},{2,5},{3,5}	Sim

Tabela 8: Subconjuntos a Checar

Cand.	Sup.
{1,3,5 }	2
{2,3,5 }	2

Tabela 9: k = 3

CIF(L).	Sup.
{1,3,5 }	2
{2,3,5 }	2

Tabela 10: Final

qualquer razão não estão presentes nos grupos anteriores. Isto se chama fazer a poda. Este passo é feito implicitamente nos passos anteriores.

Agora selecionamos somente aqueles cujos subconjuntos estão presentes no grupo anterior. ver na tabela 9.

Agora passamos para a próxima etapa (k=4). Pela combinação dos elementos anteriores podemos chegar apenas ao elemento {1,2,3,5} porém seu suporte na tabela de transções possui valor 1 o que faz com que o mesmo seja rejeitado e obtenhamos um conjunto vazio. Além disso as combinações {1,2,3} e {1,2,5} não estão presentes no conjunto (k=3). Daí não há mais como prosseguir. Parando o algoritmo no k = 4, retornamos como solução o conjunto (k=3) composto pelos subconjuntos {1,3,5} e {2,3,5} cujos valores de suporte são 2 para cada.

Com isto encerramos a etapa de determinação dos conjuntos frequentes. Uma vez determinados estes itens passamos para a etapa de geração de regras de associação.

3 Gerando as Regras de Associação

Dados os itens frequentes obtidos na etapa final do Apriori, passamos a geração das regras. Pegamos os itens frequentes obtidos no Apriori e geramos todas as combinações possíveis destes números. Os resultados são:

- para {1,3,5} temos: {1,3},{1,5}, {3,5}, {1},{3} e {5}
- para {2,3,5} temos: {2,3},{2,5}, {3,5}, {2},{3} e {5}

A seguir para cada subconjunto s não vazio do item I aplicar a regra: $s \rightarrow (I - s)$. Ficam assim geradas as regras, mas podemos submetê-las a uma etapa adicional que são a verificação da confiança e do lift.O objetivo obviamente é selecionar as regras que tiverem confiança maior que o valor mínimo admitido.

Vamos as regras geradas e os cálculos supondo que a confiança mímina é de 60

- R1: 1&3 $\rightarrow$ 5
  - $Confianca = sup(1,3,5)/sup(1,3) = 2/3 = 0.66$
  - R1 Selecionada
- R2: 1&5 $\rightarrow$ 3
  - $Confianca = sup(1,5,3)/sup(1,5) = 2/2 = 1$

  - R2 Selecionada
- R3: 3&5 → 1
  - $Confianca = sup(3,5,1)/sup(3,5) = 2/3 = 0.66$
  - R3 Selecionada
- R4: 1 → 3&5
  - $Confianca = sup(1,3,5)/sup(1) = 2/3 = 0.66$
  - R1 Selecionada
- R5: 3 → 1&5
  - $Confianca = sup(3,1,5)/sup(3) = 2/4 = 0.5$
  - R5 Rejeitada
- R6: 5 → 1&3
  - $Confianca = sup(5,1,3)/sup(5) = 2/4 = 0.5$
  - R6 Rejeitada
- R7: 2&3 → 5
  - $Confianca = sup(2,3,5)/sup(2,3) = 2/2 = 1$
  - R7 Selecionada
- R8: 2&5 → 3
  - $Confianca = sup(2,5,3)/sup(2,5) = 2/3 = 0.66$
  - R8 Selecionada
- R9: 3&5 → 2
  - $Confianca = sup(3,5,2)/sup(3,5) = 2/3 = 0.66$
  - R9 Selecionada
- R10: 2 → 3&5
  - $Confianca = sup(2,3,5)/sup(2) = 2/3 = 0.66$
  - R10 Selecionada
- R11: 3 → 2&5
  - $Confianca = sup(3,2,5)/sup(3) = 2/4 = 0.5$
  - R11 Rejeitada
- R12: 5 → 2&3
  - $Confianca = sup(5,2,3)/sup(5) = 2/4 = 0.5$
  - R12 Rejeitada

Agora podemos calcular o lift que pode classificar as regras por mais uma medida que denota a real correlação entre os atributos.

- R1: 1&3 → 5
  - $lift = \frac{sup(1,3,5)}{(sup(1,3) * sup(5))} = \frac{\frac{2}{5}}{\frac{3}{5} * \frac{4}{5}} = 0.83$
- R2: 1&5 → 3
  - $lift = \frac{sup(1,5,3)}{(sup(1,5) * sup(3))} = \frac{\frac{2}{5}}{\frac{2}{5} * \frac{4}{5}} = 1.25$
- R3: 3&5 → 1
  - $lift = \frac{sup(3,5,1)}{(sup(3,5) * sup(1))} = \frac{\frac{2}{5}}{\frac{3}{5} * \frac{3}{5}} = 1.1$
- R4: 1 → 3&5
  - $lift = \frac{sup(1,3,5)}{sup(1) * sup(3,5)} = \frac{\frac{2}{5}}{\frac{3}{5} * \frac{3}{5}} = 1.1$
- R7: 2&3 → 5
  - $lift = \frac{sup(2,3,5)}{sup(2,3) * sup(5)} = \frac{\frac{2}{5}}{\frac{2}{5} * \frac{4}{5}} = 1.25$

Tabela 11: Resultado Final

N	Regra	Litf
1	1&3 → 5	0.83
2	1&5 → 3	1.25
3	3&5 → 1	1.1
4	1 → 3&5	1.1
5	2&3 → 5	1.25
6	2&5 → 3	0.83
7	3&5 → 2	1.1
8	2 → 3&5	1.1

- R8: $2\&5 \rightarrow 3$
  - $lift = \frac{sup(2,5,3)}{sup(2,5) * sup(3)} = \frac{\frac{2}{5}}{\frac{3}{5} * \frac{4}{5}} = 0.83$
- R9: $3\&5 \rightarrow 2$
  - $lift = \frac{sup(3,5,2)}{sup(3,5) * sup(2)} = \frac{\frac{2}{5}}{\frac{3}{5} * \frac{3}{5}} = 1.1$
- R10: $2 \rightarrow 3\&5$
  - $lift = \frac{sup(2,3,5)}{sup(2) * sup(3,5)} = \frac{\frac{2}{5}}{\frac{3}{5} * \frac{3}{5}} = 1.1$

A tabela 11 a seguir ilustra o resultado final da aplicação do nosso algoritmo. A cor vermelha indica uma possível rejeição, a amarela uma possível aprovação e a cor verde marca as melhores regras. Pelo exposto acima podemos considerar que as melhores regras são: Quem compra 1 e 5 também compra 3 e quem compra 2 e 3 também compra 5.

4 Conclusão

Neste tutorial abordamos as regras de associação. Trata-se de uma tecnologia de extrema importância para a solução de diversos problemas práticos incluindo aí os chamados sistemas de recomendação. Mostramos como gerar as regras a partir de um dataset realizando, em primeiro lugar, uma seleção de conjuntos de dados que mais ocorrem em um certo número de transações dadas. Estes conjuntos são chamados de conjuntos frequentes e são obtidos através da execução de algoritmos tais como o algoritmo Apriori. Este algoritmo foi mostrado no texto e um exemplo prático foi desenvolvido a título de exercício.

Referências

[1] M. Bramer. *Principles of Data Mining.* Springer Verlag, 2013.

[2] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18, 2009.

[3] Saed Sayad. An introduction to data mining. www.saedsayad.com/data_mining_map.htm, Dec 2016.