Mineração de Dados

Aula 13: Regras de Associação

Rafael Izbicki

O Problema

Imagine que temos um banco de dados em que cada linha representa a ida de uma pessoa a um supermercado, e cada coluna representa se ela comprou ou não determinado produto.

O objetivo de regras de associação (*market basket*) é descobrir regras do tipo

"Quem compra leite em geral também compra pão",

"Quem compra cerveja e refrigerante em geral também compra carne".

Queremos fazer isso eficientemente pois os bancos de dados relacionados a isso são comumente muito grandes.

Outro exemplo: Amazon

O Problema

Imagine que temos um banco de dados em que cada linha representa a ida de uma pessoa a um supermercado, e cada coluna representa se ela comprou ou não determinado produto.

O objetivo de regras de associação (*market basket*) é descobrir regras do tipo

"Quem compra leite em geral também compra pão",

"Quem compra cerveja e refrigerante em geral também compra carne".

Queremos fazer isso eficientemente pois os bancos de dados relacionados a isso são comumente muito grandes.

Outro exemplo: Amazon

O Problema

Imagine que temos um banco de dados em que cada linha representa a ida de uma pessoa a um supermercado, e cada coluna representa se ela comprou ou não determinado produto.

O objetivo de regras de associação (*market basket*) é descobrir regras do tipo

"Quem compra leite em geral também compra pão",

"Quem compra cerveja e refrigerante em geral também compra carne".

Queremos fazer isso eficientemente pois os bancos de dados relacionados a isso são comumente muito grandes.

Outro exemplo: Amazon

O Problema

Imagine que temos um banco de dados em que cada linha representa a ida de uma pessoa a um supermercado, e cada coluna representa se ela comprou ou não determinado produto.

O objetivo de regras de associação (*market basket*) é descobrir regras do tipo

"Quem compra leite em geral também compra pão",

"Quem compra cerveja e refrigerante em geral também compra carne".

Queremos fazer isso eficientemente pois os bancos de dados relacionados a isso são comumente muito grandes.

Outro exemplo: Amazon

Vamos assumir que cada variável X_i é binária (qualquer variável categórica pode ser transformada em um conjunto de variáveis binárias; variáveis contínuas sempre podem ser discretizadas).

Nosso objetivo pode ser traduzido como

Encontrar subconjuntos S das d variáveis (ex: $S = \{2, 4, 10\}$) tais que

$$\mathbb{P}\left(\bigcap_{i \in S} X_i = 1\right)$$

é alto

Estimamos $\mathbb{P}\left(\bigcap_{i \in S} X_i = 1\right)$ usando $\frac{1}{n}\sum_{k=1}^{n} \mathbb{I}(X_{k,i} = 1 \; \forall i \in S)$

Tal probabilidade é chamada de suporte do conjunto de itens S.

Assim, buscamos todos os subconjuntos de itens S com suporte maior ou igual a um corte t.

Vamos assumir que cada variável X_i é binária (qualquer variável categórica pode ser transformada em um conjunto de variáveis binárias; variáveis contínuas sempre podem ser discretizadas).

Nosso objetivo pode ser traduzido como

Encontrar subconjuntos S das d variáveis (ex: $S = \{2, 4, 10\}$) tais que

$$\mathbb{P}\left(\bigcap_{i \in S} X_i = 1\right)$$

é alto

Estimamos $\mathbb{P}\left(\bigcap_{i \in S} X_i = 1\right)$ usando $\frac{1}{n}\sum_{k=1}^{n} \mathbb{I}(X_{k,i} = 1 \; \forall i \in S)$

Tal probabilidade é chamada de suporte do conjunto de itens S.

Assim, buscamos todos os subconjuntos de itens S com suporte maior ou igual a um corte t.

Vamos assumir que cada variável X_i é binária (qualquer variável categórica pode ser transformada em um conjunto de variáveis binárias; variáveis contínuas sempre podem ser discretizadas).

Nosso objetivo pode ser traduzido como

Encontrar subconjuntos S das d variáveis (ex: $S = \{2, 4, 10\}$) tais que

$$\mathbb{P}\left(\bigcap_{i \in S} X_i = 1\right)$$

é alto

Estimamos $\mathbb{P}\left(\bigcap_{i \in S} X_i = 1\right)$ usando $\frac{1}{n}\sum_{k=1}^{n} \mathbb{I}(X_{k,i} = 1 \; \forall i \in S)$

Tal probabilidade é chamada de suporte do conjunto de itens S.

Assim, buscamos todos os subconjuntos de itens S com suporte maior ou igual a um corte t.

Vamos assumir que cada variável X_i é binária (qualquer variável categórica pode ser transformada em um conjunto de variáveis binárias; variáveis contínuas sempre podem ser discretizadas).

Nosso objetivo pode ser traduzido como

Encontrar subconjuntos S das d variáveis (ex: $S = \{2, 4, 10\}$) tais que

$$\mathbb{P}\left(\bigcap_{i \in S} X_i = 1\right)$$

é alto

Estimamos $\mathbb{P}\left(\bigcap_{i \in S} X_i = 1\right)$ usando $\frac{1}{n}\sum_{k=1}^{n} \mathbb{I}(X_{k,i} = 1 \ \forall i \in S)$

Tal probabilidade é chamada de suporte do conjunto de itens S.

Assim, buscamos todos os subconjuntos de itens S com suporte maior ou igual a um corte t.

Vamos assumir que cada variável X_i é binária (qualquer variável categórica pode ser transformada em um conjunto de variáveis binárias; variáveis contínuas sempre podem ser discretizadas).

Nosso objetivo pode ser traduzido como

Encontrar subconjuntos S das d variáveis (ex: $S = \{2, 4, 10\}$) tais que

$$\mathbb{P}\left(\bigcap_{i \in S} X_i = 1\right)$$

é alto

Estimamos $\mathbb{P}\left(\bigcap_{i \in S} X_i = 1\right)$ usando $\frac{1}{n}\sum_{k=1}^{n} \mathbb{I}(X_{k,i} = 1 \; \forall i \in S)$

Tal probabilidade é chamada de suporte do conjunto de itens S.

Assim, buscamos todos os subconjuntos de itens S com suporte maior ou igual a um corte t.

O algoritmo Apriori

É um algoritmo usado para achar conjuntos com suporte ao menos t eficientemente, mesmo em conjunto de dados gigantescos.

Ele explora o fato de que se $S_0 \subseteq S_1$, então o suporte de S_1 é necessariamente menor ou igual ao suporte de S_0.

Usa vários outros truques.

Após encontrar todos os subconjuntos S com suporte alto, ele busca regras do tipo

$$A \Rightarrow B$$

A é chamado de antecedente, e B de consequente

Ex: "Quem compra leite em geral também compra pão"

O algoritmo Apriori

É um algoritmo usado para achar conjuntos com suporte ao menos t eficientemente, mesmo em conjunto de dados gigantescos.

Ele explora o fato de que se $S_0 \subseteq S_1$, então o suporte de S_1 é necessariamente menor ou igual ao suporte de S_0.

Usa vários outros truques.

Após encontrar todos os subconjuntos S com suporte alto, ele busca regras do tipo

$$A \Rightarrow B$$

A é chamado de antecedente, e B de consequente

Ex: "Quem compra leite em geral também compra pão"

O algoritmo Apriori

É um algoritmo usado para achar conjuntos com suporte ao menos t eficientemente, mesmo em conjunto de dados gigantescos.

Ele explora o fato de que se $S_0 \subseteq S_1$, então o suporte de S_1 é necessariamente menor ou igual ao suporte de S_0.

Usa vários outros truques.

Após encontrar todos os subconjuntos S com suporte alto, ele busca regras do tipo

$$A \Rightarrow B$$

A é chamado de antecedente, e B de consequente

Ex: "Quem compra leite em geral também compra pão"

O algoritmo Apriori

É um algoritmo usado para achar conjuntos com suporte ao menos t eficientemente, mesmo em conjunto de dados gigantescos.

Ele explora o fato de que se $S_0 \subseteq S_1$, então o suporte de S_1 é necessariamente menor ou igual ao suporte de S_0.

Usa vários outros truques.

Após encontrar todos os subconjuntos S com suporte alto, ele busca regras do tipo

$$A \Rightarrow B$$

A é chamado de antecedente, e B de consequente

Ex: "Quem compra leite em geral também compra pão"

O algoritmo Apriori

É um algoritmo usado para achar conjuntos com suporte ao menos t eficientemente, mesmo em conjunto de dados gigantescos.

Ele explora o fato de que se $S_0 \subseteq S_1$, então o suporte de S_1 é necessariamente menor ou igual ao suporte de S_0.

Usa vários outros truques.

Após encontrar todos os subconjuntos S com suporte alto, ele busca regras do tipo

$$A \Rightarrow B$$

A é chamado de antecedente, e B de consequente

Ex: "Quem compra leite em geral também compra pão"

Dado uma regra do tipo $A \Rightarrow B$, define-se:

Confiança: Uma estimativa de $P(B|A)$ (entre os usuários que compraram A, quantos compraram B?)

Lift/Levantamento: Uma estimativa de $\frac{P(B|A)}{P(B)}$ (O quando o usuário ter comprado A aumenta a probabilidade dele comprar B)

Ex:

$\{\text{manteiga de amendoim, geléia}\} \Rightarrow \{\text{pão}\}$

Suporte=0.03; Confiança=0.82; Lift=1.95

Queremos descobrir regras de associação com suporte E confiança altos.

Dado uma regra do tipo $A \Rightarrow B$, define-se:

Confiança: Uma estimativa de $P(B|A)$ (entre os usuários que compraram A, quantos compraram B?)

Lift/Levantamento: Uma estimativa de $\frac{P(B|A)}{P(B)}$ (O quando o usuário ter comprado A aumenta a probabilidade dele comprar B)

Ex:

$\{\text{manteiga de amendoim, geléia}\} \Rightarrow \{\text{pão}\}$

Suporte=0.03; Confiança=0.82; Lift=1.95

Queremos descobrir regras de associação com suporte E confiança altos.

Dado uma regra do tipo $A \Rightarrow B$, define-se:

Confiança: Uma estimativa de $P(B|A)$ (entre os usuários que compraram A, quantos compraram B?)

Lift/Levantamento: Uma estimativa de $\frac{P(B|A)}{P(B)}$ (O quando o usuário ter comprado A aumenta a probabilidade dele comprar B)

Ex:

$\{\text{manteiga de amendoim, geléia}\} \Rightarrow \{\text{pão}\}$

Suporte=0.03; Confiança=0.82; Lift=1.95

Queremos descobrir regras de associação com suporte E confiança altos.

Dado uma regra do tipo $A \Rightarrow B$, define-se:

Confiança: Uma estimativa de $P(B|A)$ (entre os usuários que compraram A, quantos compraram B?)

Lift/Levantamento: Uma estimativa de $\frac{P(B|A)}{P(B)}$ (O quando o usuário ter comprado A aumenta a probabilidade dele comprar B)

Ex:

$\{$manteiga de amendoim, geléia$\} \Rightarrow \{$pão$\}$

Suporte=0.03; Confiança=0.82; Lift=1.95

Queremos descobrir regras de associação com suporte E confiança altos.

Demonstração no R.