# Reglas de Asociación

Diplomatura en Ciencia de Datos,
Aprendizaje Automático y sus Aplicaciones
FaMAF-UNC
agosto 2018

# Contexto

- El algoritmo más popular es Apriori (Agrawal et al 1993)
- Todos los datos tienen que ser categóricos
- Inicialmente se usó para Análisis del Carrito de la Compra (Market Basket Analysis)

Pan ➔ Leche [sop = 5%, conf = 100%]

# Terminología

I = {i1, i2, ..., im}: un conjunto de items.

Transacción t :

t un conjunto de items, y t $\subseteq$ I.

Base de datos de transacciones: un conjunto de transacciones T = {t1, t2, ..., tn}.

# Ejemplo

Transacciones de compra de mercado:

```
t1: {pan, queso, leche}
t2: {manzana, huevos, sal, yogur}
…       …
tn: {bizcocho, huevos, leche}
```

Conceptos:

- Un item: un item/artículo en el carrito de la compra
- I: todos los items que se venden en el negocio
- transacción: items comprados en un carrito

# Ejemplo

Un dataset de documentos de texto. Cada documento es una bolsa de palabras

doc1: Estudiante, Enseñar, Escuela
doc2: Estudiante, Escuela
doc3: Enseñar, Escuela, Ciudad, Partido
doc4: Beisbol, Basket
doc5: Basket, Player, Espectador
doc6: Beisbol, Entrenador, Partido, Equipo
doc7: Basket, Equipo, Ciudad, Partido

Una transacción t contiene X, un conjunto de items (itemset) en I, si $X \subseteq t$.

Una regla de asociación es una implicación:

$X \rightarrow Y$, donde $X, Y \subset I$, y $X \cap Y = \varnothing$

Un itemset es un conjunto de items.

`X = {leche, pan, cereal}`

Un k-itemset es un itemset con k items.

`{leche, pan, cereal}` es un 3-itemset

# Métricas

**Soporte**: La regla tiene Soporte sup en T (el dataset de transacciones) si sup% de las transacciones contienen X ∪ Y.

$$sup = Pr(X \cup Y).$$

**Confianza**: La regla tiene Confianza conf en T si conf% de las transacciones que contienen X también contienen Y.

$$conf = Pr(Y \mid X)$$

Un regla de asociación es un patrón que dice que cuando ocurre X, ocurre Y con una cierta probabilidad.

# Objetivo de las reglas de asociación

Encontrar todas las reglas que satisfacen un soporte mínimo y confianza mínimo

- Todas las reglas
- No hay items objetivo

Una visión simplista de los datos, porque no incluye:

- cantidad
- precio
- promociones

# Algoritmos de reglas

- Hay muchos!
- Usan diferentes estrategias y estructuras de datos
- Pero los conjuntos de reglas resultantes son todos los mismos: dado un dataset, un soporte mínimo y una confianza mínima, el conjunto de reglas de asociación en T es determinístico.

Vamos a ver Apriori (Agrawal et al. 1983)

# Algoritmo Apriori

# Pasos

1. Encontrar todos los itemsets con soporte mínimo (itemsets frecuentes)

   `{pollo, ropa, leche}` [sop = 3/7]

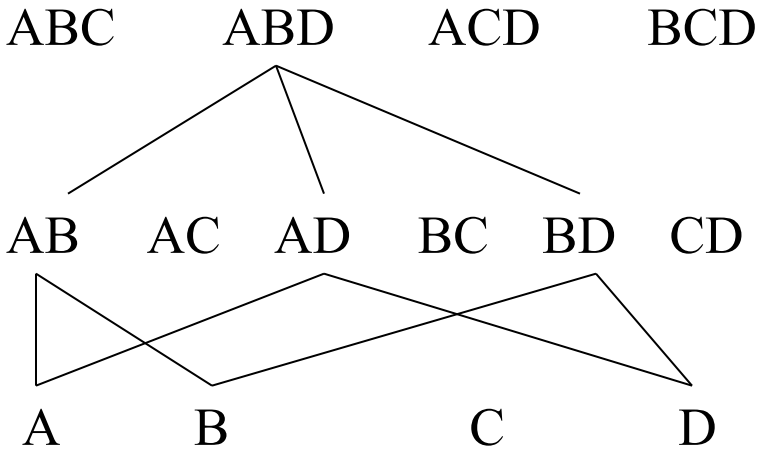
2. Usar los itemsets para generar reglas

   `ropa → leche, pollo` [sop = 3/7, conf = 3/3]

# Encontrar itemsets frecuentes

Itemset frecuente ➜ Soporte ≥ minsup

propiedad apriori (downward closure): todos los subconjuntos de un itemset frecuente también son itemsets frecuentes



# Encontrar itemsets frecuentes

Iterativo (por niveles)

Encontrar todos los itemsets frecuentes de 1 item, entonces todos los itemsets frecuentes de 2 items, y así sucesivamente

➜ en cada iteración k, considerar solamente los itemsets que contienen un itemset frecuente k-1

- Los items están ordenados, para evitar repeticiones

# Encontrar confianza

Para cada itemset frecuente X,

Para cada subconjunto no vacío A de X,

Sea B = X - A

Soporte(A → B) = Soporte(A ∪ B) = Soporte(X)

Confianza(A → B) = Soporte(A ∪ B) / Soporte(A)

A → B es una regla de asociación si

Confianza(A → B) ≥ minconf,

Esta información ya se obtuvo en el momento de generación de itemsets, no hay que recorrer el dataset de vuelta

# Ejemplo

Supongamos {2,3,4} es frecuente, con sop=50%

Subconjuntos propios no vacíos: {2,3}, {2,4}, {3,4}, {2}, {3}, {4}, con sop=50%, 50%, 75%, 75%, 75%, 75% respectivamente

Generan estas reglas de asociación:

2,3 → 4, Confianza=100%
2,4 → 3, Confianza=100%
3,4 → 2, Confianza=67%
2 → 3,4, Confianza=67%
3 → 2,4, Confianza=67%
4 → 2,3, Confianza=67%

# Consideraciones sobre Apriori

Parece muy caro pero...

- Búsqueda por niveles, explotando la propiedad de downward closure
- El parámetro k (tamaño del itemset más grande) limita el coste
- Escalable!

- El espacio de todas las reglas de asociación es exponencial, O(2m), donde m es el número de items en I.
- Explota la sparseness de los datos, los valores altos de Soporte y Confianza.
- Igualmente: un número enorme de reglas!!!

# Diferentes soportes mínimos

# Diferentes soportes mínimos

- El soporte mínimo genérico asume que todos los items se distribuyen igual
- En muchas aplicaciones, algunos items son muy frecuentes y otros no
- Si el soporte mínimo es muy alto, no encontramos reglas para items poco frecuentes
- Si el soporte mínimo es muy bajo, hay demasiadas reglas

Solución:

- Especificar diferentes soportes mínimos para diferentes items
- Propagar a reglas

  Sea MIS(i) el valor MIS del item i. El soporte mínimo de una regla R es el valor MIS más bajo de todos los items de la regla

# Ejemplo

`pan, zapatos, ropa`

Los valores MIS especificados por el usuario son:

MIS(pan) = 2% MIS(zapatos) = 0.1% MIS(ropa) = 0.2%

Esta regla no supera el soporte mínimo:

`ropa → pan` [sup=0.15%,conf =70%]

Esta regla sí supera el soporte mínimo:

`ropa → zapatos` [sup=0.15%,conf =70%]

# Downward closure

Este modelo no preserva downward closure!

Ejemplo: consideramos los cuatro items 1, 2, 3 y 4 en una base de datos. Sus soportes mínimos son

MIS(1) = 10%  MIS(2) = 20%

MIS(3) = 5%  MIS(4) = 6%

{1, 2} con Soporte 9% es infrecuente, pero {1, 2, 3} y {1, 2, 4} podrían ser frecuentes.

# Valoración diferentes soportes mínimos

- Contiene al modelo con soporte mínimo genérico
- Es un modelo más realista para aplicaciones prácticas
- Ayuda a encontrar reglas para items raros sin producir un montón de reglas inútiles con items frecuentes
- Podemos forzar a hacer reglas solamente con esos items

# Reglas de asociación con clase

# Reglas de asociación con clase

- Las reglas de asociación no tienen objetivo: encuentran todas las reglas que existen en los datos, cualquier item puede aparecer como consecuente o condición de una regla
- En algunas aplicaciones nos interesan algunos objetivos concretos

  Ejemplo: encontrar palabras asociadas a algún tema

# Reglas de asociación con clase

Sea un dataset de transacciones T con n transacciones.

Cada transacción también se etiqueta con una clase y.

Sea I el conjunto de todos los items en T, Y las etiquetas de clase y $I \cap Y = \varnothing$.

Una regla de asociación con clase es una implicación de la forma

$X \rightarrow y$, donde $X \subseteq I$, $y \in Y$.

Las definiciones de Soporte y Confianza son igual que en las reglas de asociación normales.

# Ejemplo

doc 1: Estudiante, Enseñar, Escuela : Educación
doc 2: Estudiante, Escuela : Educación
doc 3: Enseñar, Escuela, Ciudad, Partido : Educación
doc 4: Beisbol, Basket : Deporte
doc 5: Basket, Player, Espectador : Deporte
doc 6: Beisbol, Entrenador, Partido, Equipo : Deporte
doc 7: Basket, Equipo, Ciudad, Partido : Deporte

minsup = 20% y minconf = 60%

Estudiante, Escuela ➔ Educación [sup= 2/7, conf = 2/2]

Partido ➔ Deporte [sup= 2/7, conf = 2/3]

# Algoritmo

Se pueden minar en un solo paso!

Encontrar todos los ruleitems que tienen soporte > minsup, con forma:

(condset, y), y representa una regla condset ➔ y

Donde condset es un conjunto de items de I (i.e., condset $\subseteq$ I), y $\in$ Y es una etiqueta de clase.

El algoritmo apriori se puede modificar para generar reglas con clase

# Clase y diferentes soportes mínimos

El usuario puede especificar diferentes soportes mínimos para diferentes clases

Ejemplo:

- tenemos la clase Sí y la clase No
- Queremos soporte 5% para la clase Sí y Soporte 10% para la clase No

Si especificamos soporte mínimo de 100% para una clase, no se generan reglas para esa clase