

Reglas de asociación

Dra. Amparo López Gaona

Fac. Ciencias, UNAM
Abril 2018

Vivimos la era del **grandes cantidades de datos**. Ejemplos:

- Datos en transacciones: Minoristas (sistemas en puntos de venta, programas de lealtad) y del comercio electrónico.
- Datos de navegación Web: Analítica de la Web, motores de búsqueda, bibliotecas digitales, Wikis, etc.

Vivimos la era del **grandes cantidades de datos**. Ejemplos:

- Datos en transacciones: Minoristas (sistemas en puntos de venta, programas de lealtad) y del comercio electrónico.
- Datos de navegación Web: Analítica de la Web, motores de búsqueda, bibliotecas digitales, Wikis, etc.

Tamaño típico de los data sets:

- Minoristas: entre 10-500 grupos de productos y entre 500 y 10,000 productos.
- Amazon: más de 200 millones de productos (2013)
- Wikipedia: casi 5 millones de artículos (2015)
- Google: estimó más de 47 mil millones de páginas en índices (2015)

Vivimos la era del **grandes cantidades de datos**. Ejemplos:

- Datos en transacciones: Minoristas (sistemas en puntos de venta, programas de lealtad) y del comercio electrónico.
- Datos de navegación Web: Analítica de la Web, motores de búsqueda, bibliotecas digitales, Wikis, etc.

Tamaño típico de los data sets:

- Minoristas: entre 10-500 grupos de productos y entre 500 y 10,000 productos.
- Amazon: más de 200 millones de productos (2013)
- Wikipedia: casi 5 millones de artículos (2015)
- Google: estimó más de 47 mil millones de páginas en índices (2015)
- Típicamente entre 10,000 y 10 millones de transacciones (canasta de compra, sesiones de usuario, observaciones, pacientes, etc.)

- El objetivo del análisis de asociación es encontrar relaciones **interesantes** entre artículos (productos, documentos, etc.).
- Ejemplo: relación entre compras:
 - **leche, harina y huevos**, frecuentemente se compran juntos. O bien,
 - Si alguien compra **leche y harina** entonces a menudo también compran **huevos**.
- ¿Después de comprar una PC, qué más compran?
- ¿Qué tipos de ADN son sensibles a cierto medicamento nuevo?
- ¿Cómo se pueden clasificar automáticamente documentos web?

- Aplicaciones de encontrar relaciones:
 - Minoristas: Ubicación de productos, campañas de promoción, decisiones de surtido de mercancía, etc.
[Análisis de cesta de compra](#)
 - Comercio electrónico: bibliotecas digitales, motores de búsqueda: Personalización, personalización masiva.
[sistemas de recomendación, filtros colaborativos basados en artículos.](#)
- Puede ayudar en el proceso de toma de decisiones para procesos tales como diseño de catálogos, análisis de campañas de ventas, análisis de web log (flujos de click), análisis de secuencias de ADN, clustering, clasificación, etc.

- Técnica propuesta por Agrawal en 1993 y ampliamente estudiada por la comunidad de BD.
- Importante porque:
 - Revela propiedades importantes e intrínsecas de los data sets.
 - Es básica para varias tareas esenciales de la minería de datos como son:
 - Análisis de asociación, correlación y causalidad.
 - Patrones secuenciales, estructurales (sub-gráficas).
 - Análisis de patrones espaciotemporal, multimedia, series de tiempo y flujos de datos.
 - Clasificación: Clasificación asociativa.
 - Análisis de clusters: clusters basados en patrones frecuentes.

... Introducción

- Trabaja sobre datos categóricos.
- Ejemplo típico es análisis de canasta de compra. (market basket)



... Introducción

- Trabaja sobre datos categóricos.
- Ejemplo típico es análisis de canasta de compra. (market basket)



- ¿Cómo puedo encontrar estas reglas de asociación?
- ¿Cuáles reglas de asociación son más interesantes?
- ¿Cómo se pueden descubrir éstas?

Formato de los datos

Para hacer un análisis de canasta de compra y obtener reglas de asociación es necesario que trabajar con bases de datos transaccionales:

TID	Artículos
1	leche, pan
2	pan, mantequilla
3	cerveza
4	leche, pan, mantequilla
5	pan, mantequilla

Formato de los datos

Para hacer un análisis de canasta de compra y obtener reglas de asociación es necesario que trabajar con bases de datos transaccionales:

TID	Artículos
1	leche, pan
2	pan, mantequilla
3	cerveza
4	leche, pan, mantequilla
5	pan, mantequilla

	leche	pan	mant.	cerveza
1	1	1	0	0
2	0	1	1	0
3	0	0	0	1
4	1	1	1	0
5	0	1	1	0

Sean

- $I = \{i_1, i_2, \dots, i_n\}$ el conjunto de n atributos binarios llamados **items**.
- $D = \{t_1, t_2, \dots, t_m\}$ el conjunto de **transacciones** llamado la **base de datos**.
- Cada transacción en D tiene un identificador único TID y contiene un subconjunto de artículos (items) de I .

- **itemset**: Conjunto de uno o más elementos.
 - {leche, pan, cereal}
- **k-itemset**: Conjunto de exactamente k elementos $X = \{x_1, \dots, x_k\}$
- Una **regla de asociación**: es la representación de un patrón. Tiene la forma $X \Rightarrow Y$ con:
 - $X, Y \subseteq I$ **itemsets**
 - $X \cap Y = \emptyset$
 - X es el **antecedente** (lado izquierdo).
 - Y es el **consecuente** (lado derecho).

- **itemset**: Conjunto de uno o más elementos.
 - {leche, pan, cereal}
- **k-itemset**: Conjunto de exactamente k elementos $X = \{x_1, \dots, x_k\}$
- Una **regla de asociación**: es la representación de un patrón. Tiene la forma $X \Rightarrow Y$ con:
 - $X, Y \subseteq I$ **itemsets**
 - $X \cap Y = \emptyset$
 - X es el **antecedente** (lado izquierdo).
 - Y es el **consecuente** (lado derecho).
- Ejemplos:
 - computadora \Rightarrow softwareDWH
 - {Leche, pañales} \Rightarrow cerveza

... Conceptos básicos

Métricas de evaluación para una regla de la forma $X \Rightarrow Y$:

- **Soporte** de un itemset Z es la cantidad de transacciones que lo contienen.
- **Soporte de una regla** $X \Rightarrow Y$

$$\text{soporte}(X \Rightarrow Y) = \frac{\text{soporte}(X \cup Y)}{N}$$

- **Confianza** de una regla $X \Rightarrow Y$ se define como frecuencia con que los elementos Y aparecen en transacciones que contienen a X .

$$\text{Confianza}(X \Rightarrow Y) = \frac{\text{soporte}(X \cup Y)}{\text{soporte}(X)}$$

- Reflejan la utilidad y confianza de las reglas descubiertas.
 - Ejemplo: computadora \Rightarrow softwareDWH
[soporte = 2 %, confianza = 60 %]
 - El soporte significa que de las transacciones bajo análisis, sólo el 2 % muestran que ambas cosas se compran juntas y la confianza significa que el 60 % de los clientes que compraron una computadora también coon software.
- Se considera que la regla es interesante/fuerte si satisface tanto el mínimo de soporte como de confianza.
- Los umbrales los proporciona el usuario y/o experto en el dominio. Pueden expresarse en el rango de 0 a 1.0 o bien de 0 % a 100 %
- Dado un conjunto de transacciones T , el problema de minado de reglas de asociación se define como la tarea de encontrar todas las reglas que tengan un soporte $\geq minsop$ y confianza $\geq minconf$.

... Conceptos básicos (Ejemplo)

TID	DataSets Comprados
10	Cerveza, botana, pañales
20	Cerveza, café, pañales
30	Cerveza, pañales, huevos
40	Botana, huevos, leche
50	Botana, café, pañales, huevos, leche
60	Leche, huevos

- Soporte: Cerveza =

... Conceptos básicos (Ejemplo)

TID	DataSets Comprados
10	Cerveza, botana, pañales
20	Cerveza, café, pañales
30	Cerveza, pañales, huevos
40	Botana, huevos, leche
50	Botana, café, pañales, huevos, leche
60	Leche, huevos

- Soporte: $\text{Cerveza} = 3/6=0.5$, $\text{Botana} =$

... Conceptos básicos (Ejemplo)

TID	DataSets Comprados
10	Cerveza, botana, pañales
20	Cerveza, café, pañales
30	Cerveza, pañales, huevos
40	Botana, huevos, leche
50	Botana, café, pañales, huevos, leche
60	Leche, huevos

- Soporte: Cerveza = $3/6=0.5$, Botana = $3/6 = 0.5$, Pañales =

... Conceptos básicos (Ejemplo)

TID	DataSets Comprados
10	Cerveza, botana, pañales
20	Cerveza, café, pañales
30	Cerveza, pañales, huevos
40	Botana, huevos, leche
50	Botana, café, pañales, huevos, leche
60	Leche, huevos

- Soporte: Cerveza = $3/6=0.5$, Botana = $3/6 = 0.5$, Pañales = $4/6=0.66$, {Cerveza, Pañales} =

... Conceptos básicos (Ejemplo)

TID	DataSets Comprados
10	Cerveza, botana, pañales
20	Cerveza, café, pañales
30	Cerveza, pañales, huevos
40	Botana, huevos, leche
50	Botana, café, pañales, huevos, leche
60	Leche, huevos

- Soporte: Cerveza = $3/6=0.5$, Botana = $3/6 = 0.5$, Pañales = $4/6=0.66$, {Cerveza, Pañales} = $3 = 50\%$.
- Sean soporte mínimo = 50% y confianza mínima = 50% , entonces

... Conceptos básicos (Ejemplo)

TID	DataSets Comprados
10	Cerveza, botana, pañales
20	Cerveza, café, pañales
30	Cerveza, pañales, huevos
40	Botana, huevos, leche
50	Botana, café, pañales, huevos, leche
60	Leche, huevos

- Soporte: Cerveza = $3/6=0.5$, Botana = $3/6 = 0.5$, Pañales = $4/6=0.66$, {Cerveza, Pañales} = $3 = 50\%$.
- Sean soporte mínimo = 50% y confianza mínima = 50% , entonces
 - Cerveza \Rightarrow Pañales [soporte = 50% , confianza = 100%]
 - Pañales \Rightarrow Cerveza (50% , 75%)
 - Huevos \Rightarrow Leche (50% , 75%)
 - Cerveza \Rightarrow botana (16.6% , 30%)
 - etc.

Reglas de asociación

- Dado un conjunto de transacciones, encontrar reglas que ayuden a predecir la ocurrencia de un elemento basada en la ocurrencia de otros elementos en la transacción.
- Ejemplo, con la siguiente BD transaccional:

TID	Artículos
1	Pan, leche
2	Pan, pañales, cerveza, huevos
3	Leche, pañales, cerveza, coca
4	Pan, leche, pañales, cerveza
5	pan, leche, pañales, coca

se tiene

Reglas de asociación

- Dado un conjunto de transacciones, encontrar reglas que ayuden a predecir la ocurrencia de un elemento basada en la ocurrencia de otros elementos en la transacción.
- Ejemplo, con la siguiente BD transaccional:

TID	Artículos
1	Pan, leche
2	Pan, pañales, cerveza, huevos
3	Leche, pañales, cerveza, coca
4	Pan, leche, pañales, cerveza
5	pan, leche, pañales, coca

se tiene

- $\{\text{pañales}\} \Rightarrow \{\text{cerveza}\}$
- $\{\text{leche, pan}\} \Rightarrow \{\text{pañales, coca}\}$
- $\{\text{cerveza, pan}\} \Rightarrow \{\text{leche}\}$
- Implicación significa co-ocurrencia pero no causalidad.

... Minando reglas de asociación

En general, el proceso de minar reglas de asociación consta de dos pasos:

- Encontrar todos los itemsets frecuentes. (soporte mayor al mínimo)
- Generar reglas de asociación fuertes de los itemsets frecuentes. (confianza mayor al mínimo).

Métodos:

... Minando reglas de asociación

En general, el proceso de minar reglas de asociación consta de dos pasos:

- Encontrar todos los itemsets frecuentes. (soporte mayor al mínimo)
- Generar reglas de asociación fuertes de los itemsets frecuentes. (confianza mayor al mínimo).

Métodos:

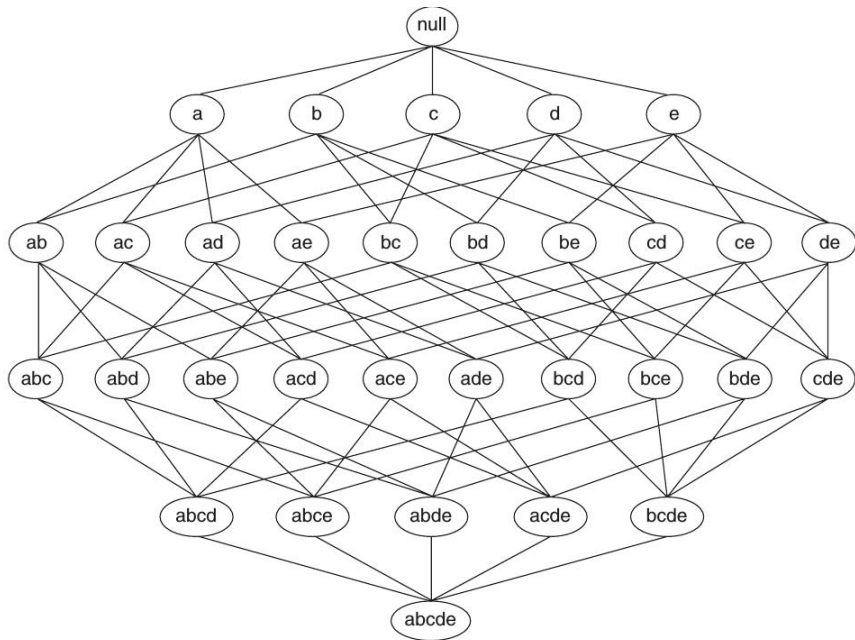
- Fuerza bruta.
- Apriori: enfoque de generación y prueba de un candidato.
- Mejora la eficiencia de Apriori.
- FPGrowth: Un enfoque de crecimiento de patrón frecuente.
- ECLAT: Minado de patrones frecuentes con formato de datos vertical.

Minando reglas de asociación

Fuerza bruta:

- Listar todas las posibles reglas de asociación.
- Calcular el soporte y la confianza para cada una.
- Eliminar las reglas que no satisfagan los umbrales *minsop* y *minconf*.

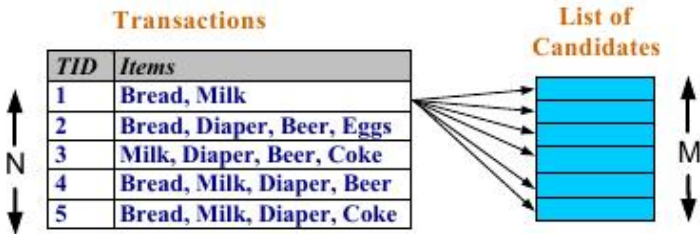
Computacionalmente es prohibitivo...



... Minando reglas de asociación

Enfoque con fuerza bruta:

- Cada itemset en la gráfica es un candidato a itemset frecuente.

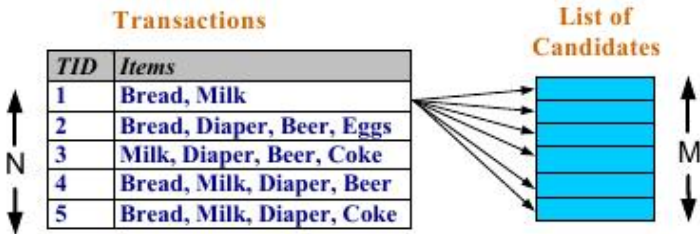


- Calcular el soporte para cada candidato.
 - Comparar cada transacción contra cada candidato.
 - Complejidad:

... Minando reglas de asociación

Enfoque con fuerza bruta:

- Cada itemset en la gráfica es un candidato a itemset frecuente.

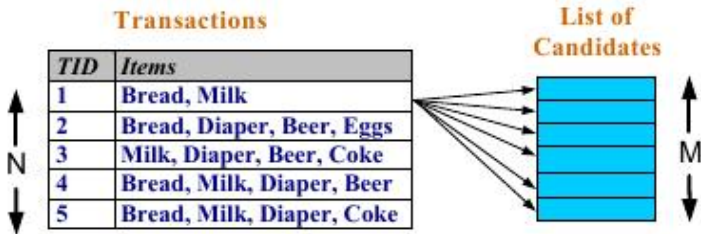


- Calcular el soporte para cada candidato.
 - Comparar cada transacción contra cada candidato.
 - Complejidad: $O(NM) \Rightarrow$

... Minando reglas de asociación

Enfoque con fuerza bruta:

- Cada itemset en la gráfica es un candidato a itemset frecuente.



- Calcular el soporte para cada candidato.
 - Comparar cada transacción contra cada candidato.
 - Complejidad: $O(NM) \Rightarrow$ Es caro ya que $M = 2^n$!!!
- R = número de posibles reglas de asociación:

$$\sum_{k=1}^{n-1} \left[\binom{n}{k} \times \sum_{j=1}^{n-k} \binom{n-k}{j} \right] = 3^n - 2^{n+1} + 1$$

Si $n=6$ hay 602 reglas.

... Minando reglas de asociación

Estrategias para la generación de itemsets frecuentes:

- Reducir el número de candidatos. (M)
 - La búsqueda completa: $M = 2^n$
 - Usar técnicas de poda para reducir M.
- Reducir el número de transacciones.
 - Reducir el tamaño de N conforme se incrementa el tamaño de los itemsets.
- Reducir el número de comparaciones. (NM)
 - Usar estructuras de datos eficientes para almacenar los candidatos o las transacciones.
 - No se necesita comparar cada candidato contra cada transacción

... Minando reglas de asociación

TID	Artículos
1	Pan, leche
2	Pan, pañales, cerveza, huevos
3	Leche, pañales, cerveza, coca
4	Pan, leche, pañales, cerveza
5	Pan, leche, pañales, coca

Ejemplos de reglas:

$\{\text{leche, pañales}\} \Rightarrow \{\text{cerveza}\}$

... Minando reglas de asociación

TID	Artículos
1	Pan, leche
2	Pan, pañales, cerveza, huevos
3	Leche, pañales, cerveza, coca
4	Pan, leche, pañales, cerveza
5	Pan, leche, pañales, coca

Ejemplos de reglas:

$\{\text{leche, pañales}\} \Rightarrow \{\text{cerveza}\} \quad (40\%, 67\%)$

$\{\text{leche, cerveza}\} \Rightarrow \{\text{pañales}\}$

... Minando reglas de asociación

TID	Artículos
1	Pan, leche
2	Pan, pañales, cerveza, huevos
3	Leche, pañales, cerveza, coca
4	Pan, leche, pañales, cerveza
5	Pan, leche, pañales, coca

Ejemplos de reglas:

$\{\text{leche, pañales}\} \Rightarrow \{\text{cerveza}\} \quad (40\%, 67\%)$

$\{\text{leche, cerveza}\} \Rightarrow \{\text{pañales}\} \quad (40\%, 100\%)$

$\{\text{pañales, cerveza}\} \Rightarrow \{\text{leche}\}$

... Minando reglas de asociación

TID	Artículos
1	Pan, leche
2	Pan, pañales, cerveza, huevos
3	Leche, pañales, cerveza, coca
4	Pan, leche, pañales, cerveza
5	Pan, leche, pañales, coca

Ejemplos de reglas:

$\{\text{leche, pañales}\} \Rightarrow \{\text{cerveza}\}$ (40 %, 67 %)

$\{\text{leche, cerveza}\} \Rightarrow \{\text{pañales}\}$ (40 %, 100 %)

$\{\text{pañales, cerveza}\} \Rightarrow \{\text{leche}\}$ (40 %, 67 %)

$\{\text{cerveza}\} \Rightarrow \{\text{leche, pañales}\}$ (40 %, 67 %)

$\{\text{pañales}\} \Rightarrow \{\text{leche, cerveza}\}$ (40 %, 50 %)

$\{\text{leche}\} \Rightarrow \{\text{pañales, cerveza}\}$ (40 %, 50 %)

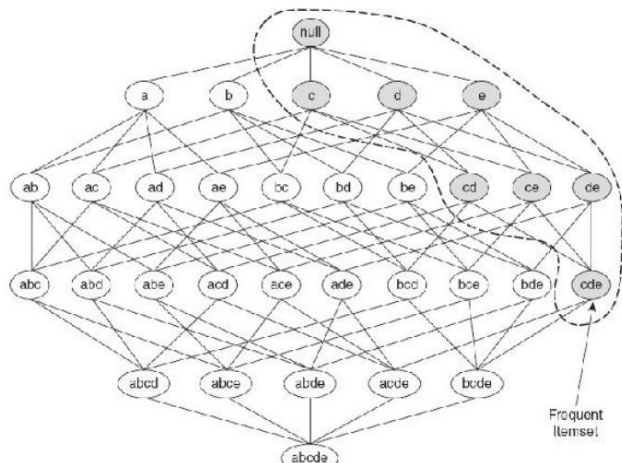
... Minando reglas de asociación

Observaciones:

- Todas las reglas son particiones binarias del mismo itemset: {leche, pañales, cerveza}.
- Las reglas generadas de un itemset tienen igual soporte pero pueden tener diferente confianza.
- Así que se pueden desacoplar los requerimientos de soporte y confianza.

... Minando reglas de asociación

- Reducción del número de candidatos:
 - Si un itemset es frecuente entonces todos sus subconjuntos, no vacíos, también son frecuentes. **Principio Apriori.**



... Principio Apriori

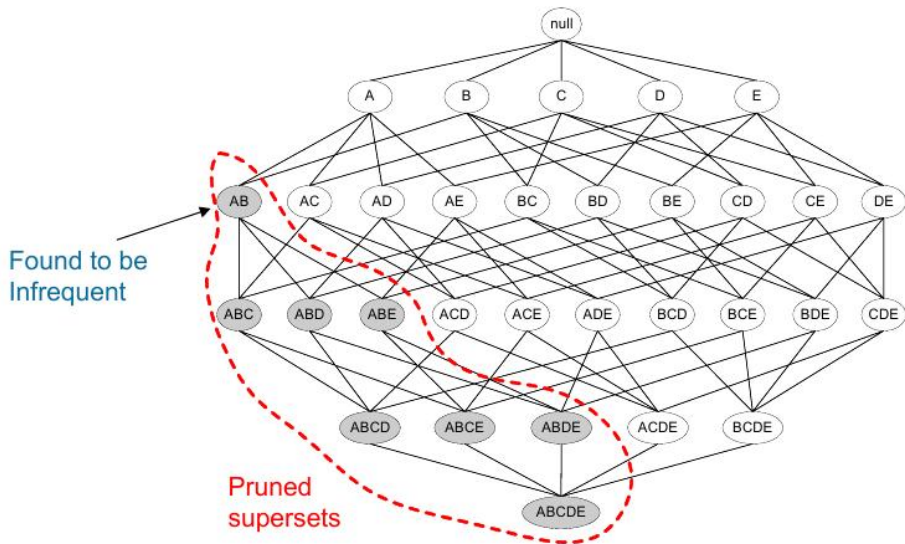
El principio Apriori lleva a la siguiente propiedad del soporte:

- El soporte de un itemset nunca excede el soporte de sus subconjuntos.

$$\forall X, Y : (X \subseteq Y) \Rightarrow \text{soporte}(X) \leq \text{soporte}(Y)$$

- Si un itemset es infrecuente, todos sus superconjuntos lo son también.
- Se puede podar el árbol una vez que se encuentra un conjunto infrecuente.

... Principio Apriori



... Minando reglas de asociación usando apriori

Con soporte mínimo de $60\% = 3$

TID	Artículos
1	Pan, leche
2	Pan, pañales, cerveza, huevos
3	Leche, pañales, cerveza, coca
4	Pan, leche, pañales, cerveza
5	Pan, leche, pañales, coca



Item	Frec
Pan	4
Coca	2
Leche	4
Cerveza	3
Pañales	4
Huevos	1



... Minando reglas de asociación usando apriori

Con soporte mínimo de $60\% = 3$

TID	Artículos
1	Pan, leche
2	Pan, pañales, cerveza, huevos
3	Leche, pañales, cerveza, coca
4	Pan, leche, pañales, cerveza
5	Pan, leche, pañales, coca



Item	Frec
Pan	4
Coca	2
Leche	4
Cerveza	3
Pañales	4
Huevos	1



Item	Frec
{Pan, Leche}	3
{ Pan, Cerveza }	2
{Pan, Pañales}	3
{ Leche, Cerveza }	2
{Leche, Pañales}	3
{Cerveza, Pañales}	3



... Minando reglas de asociación usando apriori

Con soporte mínimo de $60\% = 3$

TID	Artículos
1	Pan, leche
2	Pan, pañales, cerveza, huevos
3	Leche, pañales, cerveza, coca
4	Pan, leche, pañales, cerveza
5	Pan, leche, pañales, coca

Item	Frec
Pan	4
Coca	2
Leche	4
Cerveza	3
Pañales	4
Huevos	1

Item	Frec
{Pan, Leche}	3
{ Pan, Cerveza }	2
{Pan, Pañales}	3
{ Leche, Cerveza }	2
{Leche, Pañales}	3
{Cerveza, Pañales}	3

Ítem	Frec
{Pan, Leche, Pañales}	3

• Con todos los conjuntos considerados = $C_1^6 + C_2^6 + C_3^6 = 41$

• Podando tomando como base el soporte = $6+6+1 = 13$

Algoritmo Apriori (Pseudo código)

- Sean:
 C_k : conjunto de k-itemsets candidatos
 F_k : conjunto de k-itemsets frecuentes
- $K = 1$,
- Generar $F_1 = \{1\text{-itemset frecuentes}\}$
 $F_1 = \{i | i \in I \wedge \sigma(\{i\}) \geq \text{minsup}\}$
- Repetir hasta que no se identifiquen itemsets frecuentes:
 - Generar C_{k+1} , a partir de F_k .
 - Calcular el soporte para cada candidato en C_{k+1} recorriendo la BD.
 - Eliminar candidatos que son infrecuentes, dejando en F_{k+1} sólo los frecuentes.
 - $k = k+1$
- Regresar F_k

Detalles del algoritmo Apriori

¿Cómo generar los candidatos?

Detalles del algoritmo Apriori

¿Cómo generar los candidatos?

- Haciendo $F_{k-1} \times F_1$:

Detalles del algoritmo Apriori

¿Cómo generar los candidatos?

- Haciendo $F_{k-1} \times F_1$:

Problemas:

- Genera una cantidad enorme de candidatos.
- Genera el mismo candidato varias veces.
 - Ejemplo: $\{abc\}$ puede generarse de $\{ab\}$ con $\{c\}$, de $\{ac\}$ con $\{b\}$ o de $\{bc\}$ con $\{d\}$.

Solución:

Detalles del algoritmo Apriori

¿Cómo generar los candidatos?

- Haciendo $F_{k-1} \times F_1$:

Problemas:

- Genera una cantidad enorme de candidatos.
- Genera el mismo candidato varias veces.
 - Ejemplo: $\{abc\}$ puede generarse de $\{ab\}$ con $\{c\}$, de $\{ac\}$ con $\{b\}$ o de $\{bc\}$ con $\{d\}$.

Solución:

- Tener los elementos del conjunto en orden lexicográfico.
- Hacer un auto-join F_k , suponiendo que los artículos en F_k están en orden lexicográfico.

Detalles del algoritmo Apriori

¿Cómo generar los candidatos?

- Haciendo $F_{k-1} \times F_1$:

Problemas:

- Genera una cantidad enorme de candidatos.
- Genera el mismo candidato varias veces.
 - Ejemplo: {abc} puede generarse de {ab} con {c}, de {ac} con {b} o de {bc} con {d}.

Solución:

- Tener los elementos del conjunto en orden lexicográfico.
- Hacer un auto-join F_k , suponiendo que los artículos en F_k están en orden lexicográfico.

```
INSERT INTO Ck
```

```
SELECT p.item1, p.item2, ..., p.itemk-1, q.itemk-1
FROM Fk-1 p, Fk-1 q
WHERE p.item1=q.item1, ..., p.itemk-2=q.itemk-2,
      p.itemk-1 < q.itemk-1
```


... Detalles del algoritmo Apriori

En cualquier caso, es necesario podar:

- Para todos los itemsets c in C_k hacer
 - Para todos los $(k-1)$ -subsets s de c hacer
Si (s no está en F_{k-1}) borrar c de C_k

... Detalles del algoritmo Apriori

- Ejemplo de generación de candidatos:
 - Sea $F_3 = \{abc, abd, acd, ace, bcd\}$
 - Auto-join $F_3 * F_3$
 - $abcd$ de abc y abd
 - $acde$ de acd y ace
 - Poda:
 - $acde$ se elimina porque ade no está en F_3
 - $C_4 = \{abcd\}$

... Detalles del algoritmo Apriori

- Cálculo del soporte: Proceso de determinar la frecuencia de ocurrencia de cada itemset candidato que sobrevive al paso de poda.

... Detalles del algoritmo Apriori

- Cálculo del soporte: Proceso de determinar la frecuencia de ocurrencia de cada itemset candidato que sobrevive al paso de poda.
- Comparar cada transacción contra cada itemset candidato.
 - Problema:

... Detalles del algoritmo Apriori

- Cálculo del soporte: Proceso de determinar la frecuencia de ocurrencia de cada itemset candidato que sobrevive al paso de poda.
- Comparar cada transacción contra cada itemset candidato.
 - Problema: varias pasadas a la BD.
 - Soluciones:

... Detalles del algoritmo Apriori

- Cálculo del soporte: Proceso de determinar la frecuencia de ocurrencia de cada itemset candidato que sobrevive al paso de poda.
- Comparar cada transacción contra cada itemset candidato.
 - Problema: varias pasadas a la BD.
 - Soluciones:
 - Usar una estructura hash.

... Detalles del algoritmo Apriori

- Cálculo del soporte: Proceso de determinar la frecuencia de ocurrencia de cada itemset candidato que sobrevive al paso de poda.
- Comparar cada transacción contra cada itemset candidato.
 - Problema: varias pasadas a la BD.
 - Soluciones:
 - Usar una estructura hash.
 - Reducción de transacciones. Transacciones que no contienen ningún k -itemset frecuente no contendrán ningún $(k-1)$ -itemset frecuente, así que esa transacción puede eliminarse en siguientes pasos del ciclo.

- Umbral para el soporte:
 - a menor umbral mayor cantidad de itemsets candidatos.
 - a menor umbral itemsets con mayor cantidad de elementos.
- Tamaño de la base de datos:
 - Ya que se hacen varias pasadas a la BD, el tiempo de ejecución del algoritmo puede incrementarse con la cantidad de transacciones.
- Tamaño de las transacciones.
 - El tamaño máximo de itemsets frecuentes tiende a incrementarse.
 - Las transacciones contendrán más itemsets.

Generación de reglas de asociación

Generación de reglas de asociación

Con mínimo soporte = 2:

TID	id_ítem
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3



Itemset	cont
{I1}	

Generación de reglas de asociación

Con mínimo soporte = 2:

TID	id_ítem
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

→

Itemset	cont
{I1}	6
{I2}	

Generación de reglas de asociación

Con mínimo soporte = 2:

TID	id_ítem
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3



Itemset	cont
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Generación de reglas de asociación

Con mínimo soporte = 2:

TID	id_ítem
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3



Itemset	cont
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2



Itemset	cont
{I1,I2}	4

Generación de reglas de asociación

Con mínimo soporte = 2:

TID	id_ítem
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3



Itemset	cont
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2



Itemset	cont
{I1,I2}	4
{I1,I3}	4
{I1,I5}	2
{I2,I3}	4
{I2,I4}	2
{I2,I5}	2

Generación de reglas de asociación

Con mínimo soporte = 2:

TID	id_ítem
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3



Itemset	cont
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2



Itemset	cont
{I1,I2}	4
{I1,I3}	4
{I1,I5}	2
{I2,I3}	4
{I2,I4}	2
{I2,I5}	2



Itemset	cont
{I1,I2,I3}	2
{I1,I2,I5}	2

Reglas:

Generación de reglas de asociación

Con mínimo soporte = 2:

TID	id_ítem
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3



Itemset	cont
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2



Itemset	cont
{I1,I2}	4
{I1,I3}	4
{I1,I5}	2
{I2,I3}	4
{I2,I4}	2
{I2,I5}	2



$I1 \wedge I2 \Rightarrow I5$,

confianza =

Itemset	cont
{I1,I2,I3}	2
{I1,I2,I5}	2

Reglas:

Generación de reglas de asociación

Con mínimo soporte = 2:

TID	id_ítem
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3



Itemset	cont
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2



Itemset	cont
{I1,I2}	4
{I1,I3}	4
{I1,I5}	2
{I2,I3}	4
{I2,I4}	2
{I2,I5}	2



Itemset	cont
{I1,I2,I3}	2
{I1,I2,I5}	2

Reglas:

$I1 \wedge I2 \Rightarrow I5$,

$I1 \wedge I5 \Rightarrow I2$,

confianza = $2/4 = 50\%$

confianza =

Generación de reglas de asociación

Con mínimo soporte = 2:

TID	id_ítem
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3



Itemset	cont
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2



Itemset	cont
{I1,I2}	4
{I1,I3}	4
{I1,I5}	2
{I2,I3}	4
{I2,I4}	2
{I2,I5}	2



Itemset	cont
{I1,I2,I3}	2
{I1,I2,I5}	2

Reglas:

$$I1 \wedge I2 \Rightarrow I5,$$

$$I1 \wedge I5 \Rightarrow I2,$$

$$I2 \wedge I5 \Rightarrow I1,$$

$$I1 \Rightarrow I2 \wedge I5,$$

$$I2 \Rightarrow I1 \wedge I5,$$

$$I5 \Rightarrow I1 \wedge I2,$$

$$\text{confianza} = 2/4 = 50\%$$

$$\text{confianza} = 2/2 = 100\%$$

$$\text{confianza} = 2/2 = 100\%$$

$$\text{confianza} = 2/6 = 33\%$$

$$\text{confianza} = 2/7 = 29\%$$

$$\text{confianza} = 2/2 = 100\%$$

... Generación de reglas de asociación

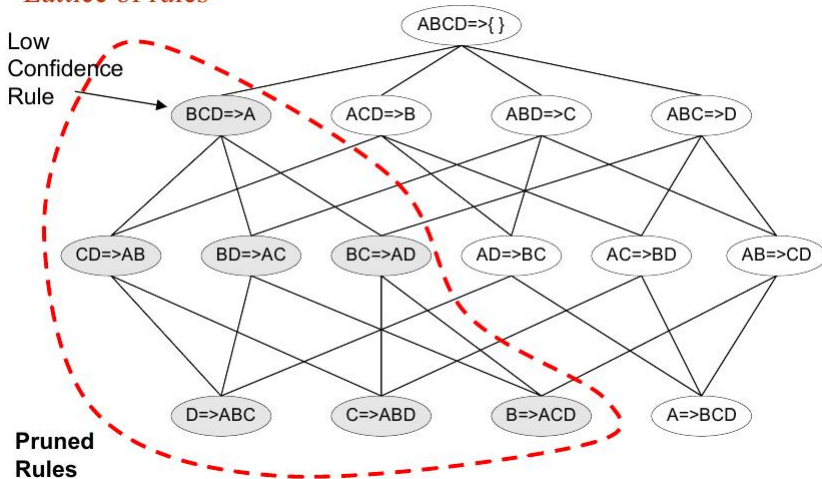
- De cada k-itemset frecuente Y , se pueden producir $2^k - 2$ reglas de asociación.
- Cada regla se genera dividiendo el conjunto en dos subconjuntos: X y $Y - X$, tales que $X \rightarrow Y - X$ tiene confianza $\geq \text{minconf}$.
- El soporte

... Generación de reglas de asociación

- De cada k-itemset frecuente Y , se pueden producir $2^k - 2$ reglas de asociación.
- Cada regla se genera dividiendo el conjunto en dos subconjuntos: X y $Y - X$, tales que $X \rightarrow Y - X$ tiene confianza $\geq \text{minconf}$.
- El soporte ya sabemos que está por arriba del solicitado pues es frecuente.
- Al principio todas las reglas con alta confianza tienen sólo un artículo en el consecuente.
- Para generar las reglas se va incrementando la cantidad de elementos del consecuente.
- Si una regla $X \rightarrow (Y - X)$ no satisface el umbral de confianza, entonces tampoco lo satisface ninguna regla $X' \rightarrow Y - X'$, con X' subconjunto de X .

... Generación de reglas

Lattice of rules



¿El Apriori es suficientemente rápido?

- EL corazón del algoritmo Apriori:
 - Usa $(k-1)$ itemsets frecuentes para generar los candidatos a k -itemsets frecuentes.
 - Recorre la BD y compara patrones para calcular los contadores para los itemsets candidatos.
- El cuello de botella del Apriori:

¿El Apriori es suficientemente rápido?

- EL corazón del algoritmo Apriori:
 - Usa (k-1) itemsets frecuentes para generar los candidatos a k-itemsets frecuentes.
 - Recorre la BD y compara patrones para calcular los contadores para los itemsets candidatos.
- El cuello de botella del Apriori: la generación de candidatos.
 - Puede requerir generar grandes cantidades de conjuntos candidatos:
 - Para descubrir un patrón frecuente de tamaño 100, se necesitan generar $2^{100} \approx 10^{30}$ candidatos.
 - Múltiples barridos de la BD:
 - Se necesitan (n+1), con n la longitud del patrón más largo.

Mejoras a la eficiencia del Apriori

Mejoras a la eficiencia del Apriori

- Reducir la cantidad de k-itemsets candidatos para $k > 1$ utilizando una tabla de dispersión.
- Reducción de transacciones. Una transacción que no contiene ningún k-itemset frecuente no tendrá ninguno (k+1) frecuente, por lo tanto se podría marcar y eliminar para consideraciones futuras.

Métodos de evaluación de los patrones

- Medidas objetivas:

- soporte; y
- confianza.

- Medidas subjetivas:

Una regla (patrón) es interesante si:

- es *inesperado* o
- *accionable*.

... Métodos de evaluación de los patrones

¿Todas las reglas fuertes, descubiertas, son suficientemente interesantes para presentarlas al usuario?

- Ejemplo: Entre 5000 estudiantes se tiene que:
 - 3000 juegan fútbol.
 - 3750 comen cereal.
 - 2000 juegan fútbol y comen cereal.

	fútbol	no fútbol	total
cereal	2000	1750	3750
no cereal	1000	250	1250
total (col)	3000	2000	5000

- Umbrales: soporte = 30 % y confianza = 60 %
- `juegan fútbol` \Rightarrow `comenCereal` (40 %, 66.7 %) es engañosa porque el porcentaje total de estudiantes que comen cereal es 75 % que es mayor que el 66.7 %.
- `juegan fútbol` \Rightarrow `noComenCereal` (20 %, 33.3 %) es más precisa, aunque tiene menor soporte y confianza.
- La confianza de una regla puede ser engañosa

... Métodos de evaluación de los patrones

- Las reglas basadas en soporte-confianza son útiles para muchas aplicaciones pero si se desea que A implique no B, hay otras alternativas basadas en correlación.
- $A \Rightarrow B$ [soporte, confianza, correlación]
- **lift** es una medida de correlación:
 - La ocurrencia del itemset A es independiente de la ocurrencia del itemset B si $P(A \cup B) = P(A)P(B)$,
 - En otro caso, son dependientes y correlacionados.
- La correlación se mide aplicando:

$$lift_{A,B} = \frac{P(A \cup B)}{P(A)P(B)}$$

- Si < 1 A y B están correlacionados negativamente.
- Si > 1 A y B están correlacionados positivamente.
- Si $= 1$ A y B son eventos independientes.

... Métodos de evaluación de los patrones

- Entonces en el ejemplo:

	fútbol	no fútbol	total
cereal	2000	1750	3750
no cereal	1000	250	1250
total (col)	3000	2000	5000

- `jueganFútbol` \Rightarrow `comenCereal` (40%, 66.7%)

$$lift(F, C) = \frac{2000/5000}{3000/5000 * 3750/5000} = 0,89$$

Ejemplo: Trabajo con un conjunto de datos del Titanic

Suponer que se tiene información de los pasajeros que viajaban en el Titanic, como sigue:

```
> dim(titanic)
```

```
[1] 2201    4
```

```
> head(titanic)
```

	Class	Sex	Age	Survived
1	3rd	Male	Child	No
2	3rd	Male	Child	No
3	3rd	Male	Child	No
4	3rd	Male	Child	No
5	3rd	Male	Child	No
6	3rd	Male	Child	No

```
> summary(titanic)
```

Class	Sex	Age	Survived
1st :325	Female: 470	Adult:2092	No :1490
2nd :285	Male :1731	Child: 109	Yes: 711
3rd :706			

Generación de reglas (valores por omisión)

```
> rules.all <- apriori(titanic)
parameter specification:
  confidence minval  smax   arem  aval   originalSupport support minlen
  0.8          0.1     1     none FALSE TRUE              0.1     1
  maxlen target  ext
  10      rules  FALSE

....
```

```
apriori - find association rules with the apriori algorithm
version 4.21 (2004.05.09)      (c) 1996-2004 Christian Borgelt
set item appearances ... [0 item(s)] done [0.00s].
set transactions ... [10 item(s), 2201 transaction(s)] done [0.00s]
sorting and recoding items ... [9 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 done [0.00s].
writing ... [27 rule(s)] done [0.00s].
creating S4 object ... done [0.00s].
```

Mostrar reglas

```
> rules.all
```

```
set of 27 rules
```

```
> inspect(rules.all)
```

	lhs	rhs	support	confidence	lift
1	{}	=> {Age=Adult}	0.9504771	0.9504771	1.0000000
2	{Class=2nd}	=> {Age=Adult}	0.1185825	0.9157895	0.9635051
3	{Class=1st}	=> {Age=Adult}	0.1449341	0.9815385	1.0326798
4	{Sex=Female}	=> {Age=Adult}	0.1930940	0.9042553	0.9513700
5	{Class=3rd}	=> {Age=Adult}	0.2848705	0.8881020	0.9343750
6	{Survived=Yes}	=> {Age=Adult}	0.2971377	0.9198312	0.9677574
7	{Class=Crew}	=> {Sex=Male}	0.3916402	0.9740113	1.2384742
8	{Class=Crew}	=> {Age=Adult}	0.4020900	1.0000000	1.0521033
9	{Survived=No}	=> {Sex=Male}	0.6197183	0.9154362	1.1639949
10	{Survived=No}	=> {Age=Adult}	0.6533394	0.9651007	1.0153856
11	{Sex=Male}	=> {Age=Adult}	0.7573830	0.9630272	1.0132040
12	{Sex=Female, Survived=Yes}	=> {Age=Adult}	0.1435711	0.9186047	0.9664669

13 {Class=3rd, Sex=Male}	=> {Survived=No}	0.1917310	0.8274510	1.2222950
14 {Class=3rd, Survived=No}	=> {Age=Adult}	0.2162653	0.9015152	0.9484870
15 {Class=3rd, Sex=Male}	=> {Age=Adult}	0.2099046	0.9058824	0.9530818
16 {Sex=Male, Survived=Yes}	=> {Age=Adult}	0.1535666	0.9209809	0.9689670
17 {Class=Crew, Survived=No}	=> {Sex=Male}	0.3044071	0.9955423	1.2658514
18 {Class=Crew, Survived=No}	=> {Age=Adult}	0.3057701	1.0000000	1.0521033
19 {Class=Crew, Sex=Male}	=> {Age=Adult}	0.3916402	1.0000000	1.0521033
20 {Class=Crew, Age=Adult}	=> {Sex=Male}	0.3916402	0.9740113	1.2384742
21 {Sex=Male, Survived=No}	=> {Age=Adult}	0.6038164	0.9743402	1.0251065
22 {Age=Adult, Survived=No}	=> {Sex=Male}	0.6038164	0.9242003	1.1751385
23 {Class=3rd,				

Reglas interesadas en sobrevivientes

```
> rules <- apriori(titanic, control = list(verbose=F),  
  parameter = list(minlen=2, supp=0.005, conf=0.8),  
  appearance = list(rhs=c("Survived=No", "Survived=Yes"),  
    default="lhs"))  
> quality(rules) <- round(quality(rules), digits=3)  
> rules.sorted <- sort(rules, by="lift")  
> inspect(rules.sorted)
```

	lhs	rhs	support	confidence	lift
1	{Class=2nd, Age=Child}	=> {Survived=Yes}	0.011	1.000	3.096
2	{Class=2nd, Sex=Female, Age=Child}	=> {Survived=Yes}	0.006	1.000	3.096
3	{Class=1st, Sex=Female}	=> {Survived=Yes}	0.064	0.972	3.010
4	{Class=1st, Sex=Female, Age=Adult}	=> {Survived=Yes}	0.064	0.972	3.010

```
5 {Class=2nd,  
   Sex=Female} => {Survived=Yes} 0.042 0.877 2.716  
6 {Class=Crew,  
   Sex=Female} => {Survived=Yes} 0.009 0.870 2.692  
7 {Class=Crew,  
   Sex=Female,  
   Age=Adult} => {Survived=Yes} 0.009 0.870 2.692  
8 {Class=2nd,  
   Sex=Female,  
   Age=Adult} => {Survived=Yes} 0.036 0.860 2.663  
9 {Class=2nd,  
   Sex=Male,  
   Age=Adult} => {Survived=No} 0.070 0.917 1.354  
10 {Class=2nd,  
    Sex=Male} => {Survived=No} 0.070 0.860 1.271  
11 {Class=3rd,  
    Sex=Male,  
    Age=Adult} => {Survived=No} 0.176 0.838 1.237  
12 {Class=3rd,  
    Sex=Male} => {Survived=No} 0.192 0.827 1.222
```

Eliminación de redundancia

```
> rules <- sort(rules, by="confidence")
> inspect(rules)
#Vemos cuales son las reglas redundantes
> is.redundant(rules)
[1] FALSE TRUE FALSE TRUE FALSE FALSE FALSE TRUE FALSE TRUE FALSE
```

#Vemos las reglas redundantes

```
> quality(rules) <- round(quality(rules), digits=3)
> inspect(rules[is.redundant(rules)])
```

	lhs	rhs	support	conf
[1]	{Class=2nd,Sex=Female,Age=Child}	=> {Survived=Yes}	0.006	1.000
[2]	{Class=1st,Sex=Female,Age=Adult}	=> {Survived=Yes}	0.064	0.979
[3]	{Class=Crew,Sex=Female,Age=Adult}	=> {Survived=Yes}	0.009	0.833
[4]	{Class=2nd,Sex=Female,Age=Adult}	=> {Survived=Yes}	0.036	0.833

#Vemos las reglas redundantes

```
> quality(rules) <- round(quality(rules), digits=3)
> inspect(rules[!is.redundant(rules)])
```

... Eliminación de redundancia

```
> # elimina las reglas redundantes  
> rules <- rules[!is.redundant(rules)]
```

```
> inspect(rules)
```

	lhs	rhs	support	conf
1	{Class=2nd, Age=Child}	=> {Survived=Yes}	0.011	1.00
2	{Class=1st, Sex=Female}	=> {Survived=Yes}	0.064	0.97
3	{Class=2nd, Sex=Female}	=> {Survived=Yes}	0.042	0.87
4	{Class=Crew, Sex=Female}	=> {Survived=Yes}	0.009	0.87
5	{Class=2nd, Sex=Male, Age=Adult}	=> {Survived=No}	0.070	0.97
6	{Class=2nd, Sex=Male}	=> {Survived=No}	0.070	0.80
7	{Class=3rd, Sex=Male, Age=Adult}	=> {Survived=No}	0.176	0.83
8	{Class=3rd, Sex=Male}	=> {Survived=No}	0.192	0.83

Interpretación de las reglas

¿Qué podemos concluir de la regla:

```
1  {Class=2nd,  
    Age=Child}    => {Survived=Yes}  0.011      1.000      3.096
```

```
> rules <- apriori(titanic,  
+               parameter = list(minlen=3, supp=0.002, conf=0.2),  
+               appearance = list(rhs=c("Survived=Yes"),  
+                                 lhs=c("Class=1st", "Class=2nd",  
+                                       "Class=3rd",  
+                                       "Age=Child", "Age=Adult"),  
+                                 default="none"),  
+               control = list(verbose=F))  
> rules.sorted <- sort(rules, by="confidence")
```

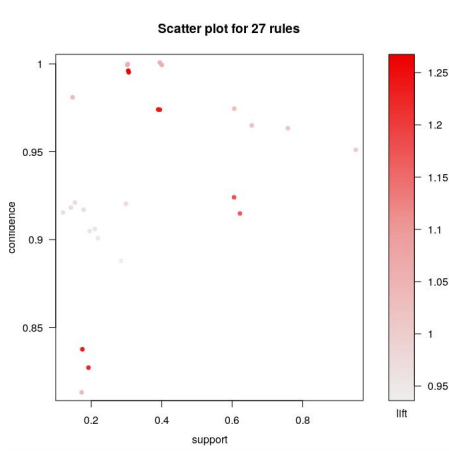
... Interpretación de las reglas

```
> inspect(rules.sorted)
```

	lhs	rhs	support	confidence	lift
1	{Class=2nd, Age=Child}	=> {Survived=Yes}	0.010904134	1.0000000	3.0956399
2	{Class=1st, Age=Child}	=> {Survived=Yes}	0.002726034	1.0000000	3.0956399
3	{Class=1st, Age=Adult}	=> {Survived=Yes}	0.089504771	0.6175549	1.9117275
4	{Class=2nd, Age=Adult}	=> {Survived=Yes}	0.042707860	0.3601533	1.1149048
5	{Class=3rd, Age=Child}	=> {Survived=Yes}	0.012267151	0.3417722	1.0580035
6	{Class=3rd, Age=Adult}	=> {Survived=Yes}	0.068605179	0.2408293	0.7455209

Visualización de reglas

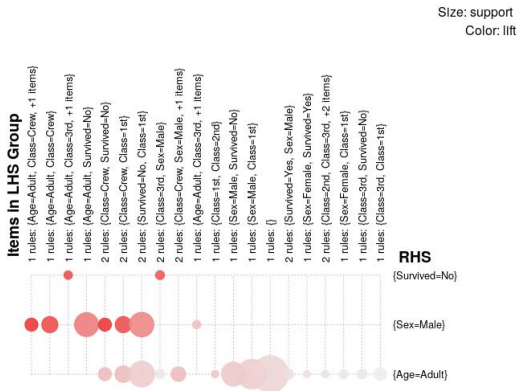
```
install.packages("arulesViz")  
library(arulesViz)  
plot(rules.all)
```



Visualización de reglas

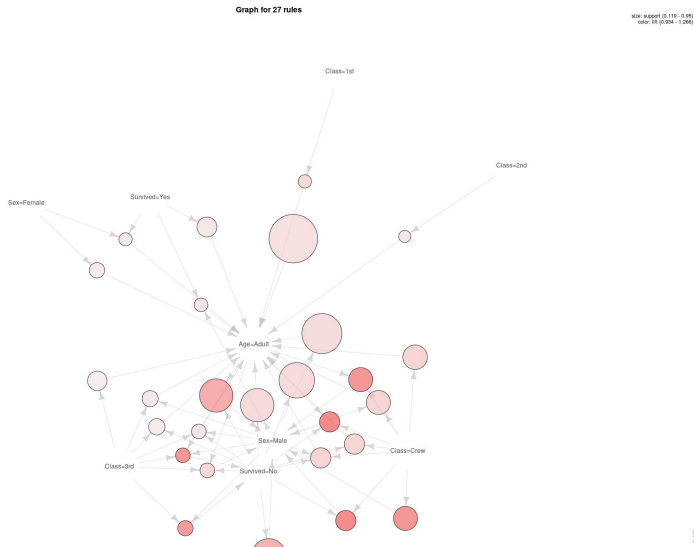
```
plot(rules.all, method="grouped")
```

Grouped Matrix for 27 Rules



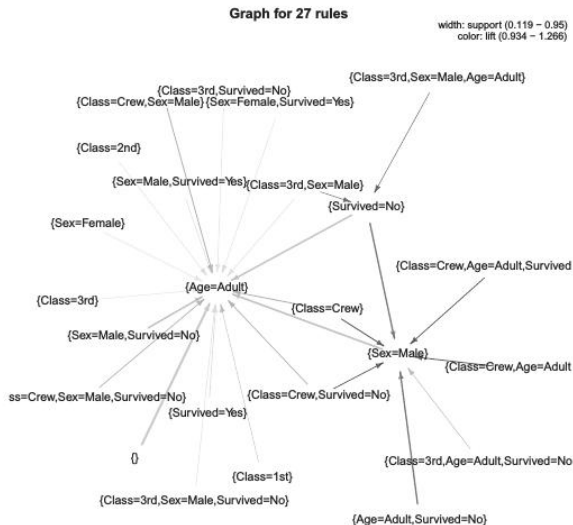
Visualización de reglas

```
plot(rules.all, method="graph")
```



Visualización de reglas

```
plot(rules, method="graph")
```



Minado de itemsets frecuentes en formato vertical

- Datos en formato horizontal:

TID	id_ítem
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

Minado de itemsets frecuentes en formato vertical

- Datos en formato horizontal:

TID	id_ítem
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

- Datos en formato vertical:

itemset	TID-set
I1	{T100, T400, T500, T700, T800, T900}
I2	{T100, T200, T300, T400, T600, T800, T900}
I3	{T300, T500, T600, T700, T800, T900 }
I4	{T200, T400}
I5	{T100, T800}

... Minado de itemsets frecuentes en formato vertical

- Algoritmo Eclat (*Equivalence Class Transformation*).
- El minado se hace intersectando los conjuntos de TID sets de cada par de artículos frecuentes.
- En el ejemplo, si el mínimo soporte es 2 se tiene que cada artículo es frecuente y por lo tanto hay 10 intersecciones en total que llevan a ocho 2-itemsets.

itemset	TID-set
{I1,I2}	{T100, T400, T800, T900}
{I1,I3}	{T500, T700, T800, T900}
{I1,I4}	{T400 }
{I1,I5}	{T100, T800}
{I2,I3}	{T300, T600, T800, T900}
{I2,I4}	{T200, T400}
{I2,I5}	{T100,T800}
{I3,I5}	{T800}

Minado de itemsets frecuentes en formato vertical

Basados en el principio Apriori se tiene un 3-itemset es candidato sólo si cada uno de sus 2-itemsets es frecuente, de ahí que los 3-itemsets posibles sólo son:

itemset	TID-set
{I1,I2,I3}	{T800, T900}
{I1,I2,I5}	{T100, T800}

Minado de itemsets frecuentes en formato vertical

En resumen:

- Transformar los datos de formato horizontal a vertical. (en una sola pasada a la BD)
- El soporte de cada itemset es la cardinalidad del TID-set.
- Empezando con $k=1$, se usan los k -itemsets frecuentes para construir $(k+1)$ -itemsets utilizando el principio Apriori.
 - Calcular la intersección de los TID-sets de los k -itemsets frecuentes.
- El proceso continua incrementando k en 1 hasta que no haya conjuntos frecuentes.

Ventajas:

- Se utiliza el principio Apriori para ir eliminando itemsets.
- No se recorre la BD varias veces.


```
fsets <-eclat(titanic)
```

parameter specification:

tidLists	support	minlen	maxlen	target	ext
FALSE	0.1	1	10	frequent itemsets	FALSE

algorithmic control:

sparse	sort	verbose
7	-2	TRUE

```
eclat - find frequent item sets with the eclat algorithm
version 2.6 (2004.08.16)          (c) 2002-2004  Christian Borgelt
create itemset ...
set transactions ...[10 item(s), 2201 transaction(s)] done [0.00s].
sorting and recoding items ... [9 item(s)] done [0.00s].
creating bit matrix ... [9 row(s), 2201 column(s)] done [0.00s].
writing ... [35 set(s)] done [0.00s].
Creating S4 object ... done [0.00s].
```

```
inspect(fsets)
  items support
1 {Class=2nd, Age=Adult} 0.1185825
2 {Class=1st, Age=Adult} 0.1449341
3 {Sex=Female, Age=Adult, Survived=Yes} 0.1435711
4 {Sex=Female, Age=Adult} 0.1930940
5 {Sex=Female, Survived=Yes} 0.1562926
6 {Class=3rd, Sex=Male, Age=Adult, Survived=No} 0.1758292
7 {Class=3rd, Age=Adult, Survived=No} 0.2162653
8 {Class=3rd, Sex=Male, Survived=No} 0.1917310
9 {Class=3rd, Sex=Male, Age=Adult} 0.2099046
10 {Class=3rd, Age=Adult} 0.2848705
11 {Class=3rd, Sex=Male} 0.2317129
12 {Class=3rd, Survived=No} 0.2398910
13 {Sex=Male, Age=Adult, Survived=Yes} 0.1535666
14 {Age=Adult, Survived=Yes} 0.2971377
15 {Sex=Male, Survived=Yes} 0.1667424
16 {Class=Crew, Sex=Male, Age=Adult, Survived=No} 0.3044071
```

```
fsets <- eclat(titanic, parameter = list(minlen=3), control = list(v  
fsets <- sort(fsets, by="support")
```

```
> inspect(fsets)
```

	items	support
11	{Sex=Male, Age=Adult, Survived=No}	0.6038164
10	{Class=Crew, Sex=Male, Age=Adult}	0.3916402
8	{Class=Crew, Age=Adult, Survived=No}	0.3057701
7	{Class=Crew, Sex=Male, Age=Adult, Survived=No}	0.3044071
9	{Class=Crew, Sex=Male, Survived=No}	0.3044071
3	{Class=3rd, Age=Adult, Survived=No}	0.2162653
5	{Class=3rd, Sex=Male, Age=Adult}	0.2099046
4	{Class=3rd, Sex=Male, Survived=No}	0.1917310
2	{Class=3rd, Sex=Male, Age=Adult, Survived=No}	0.1758292
6	{Sex=Male, Age=Adult, Survived=Yes}	0.1535666
1	{Sex=Female, Age=Adult, Survived=Yes}	0.1435711