UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Facultad de Estudios Superiores
Acatlán

Practica 3:

Algoritmo A Priori

Aprendizaje de maquina y minería de datos avanzados

Integrantes:

Gustavo Adolfo Alvarez Hernández

Profesor:

Eduardo Eloy Loza Pacheco

FES Acatlán, 19 de febrero 2023

Objetivo

Que el alumno entienda los vínculos y reglas asociación entre elementos, por medio del algoritmo a priori. Esto con el fin de encontrar patrones que sirvan para generalizar el comportamiento del usuario.

Objetivo

Que el alumno utilice el algoritmo a priori sobre datos de canasta de supermercado, para encontrar reglas de asociaciones sobre los artículos que compran juntos. Esto variando los límites de aceptación de confidencia, lift o soporte.

Antecedentes

El algoritmo a priori consiste en encontrar subconjuntos de elementos frecuentes entre todos los conjuntos posibles que se puede realizar, este trabajo puede llegar a ser extensivo y por tanto costoso computacionalmente, por lo cual se sigue este procedimiento en general.

1. Pasa sobre todas las canastas o transacciones y calcula el soporte de cada elemento del conjunto de datos, lo cual es:

$$Soporte(X) = \frac{Ocurrencias(X)}{Total\ de\ transacciones}$$

2. Se establece un umbral del soporte, los elementos que superen el umbral se consideran frecuentes y ahora se buscan subconjuntos de dos elementos, con la restricción de solo admitir subconjuntos de elementos frecuentes.
3. Se calcula el soporte de los subconjuntos de dos elementos.
4. Se repite el proceso ahora para 3 elementos y así sucesivamente hasta que se determine la máxima cardinalidad del subconjunto o ya no se puedan realizar subconjuntos.

Al añadir el elemento de umbral eliminamos el numero de posibles subconjuntos y por tanto se hace una búsqueda más eficiente, basándonos en la idea de “Un subconjunto frecuente no contiene elementos no frecuentes”.

Otras métricas que se utilizan es la confianza que es la probabilidad condicional de escoger un elemento dado que ya se tiene el otro:

$$Confianza(X \rightarrow Y) = \frac{soporte(X \cap Y)}{soporte(X)}$$

Y la otra es Lift que nos habla de la relación que tienen los elementos pero tomando en cuenta que tan probable es que pase la consecuencia:

$$Lift(X \rightarrow Y) = \frac{Confianza(X \rightarrow Y)}{Soporte(Y)}$$

Desarrollo

1. Cargamos los módulos

```
import pandas as pd
from apyori import apriori
```

2. Cargamos los datos donde se tiene una lista de artículos de supermercado.

```
data= pd.read_csv("Market_Basket_Optimisation.csv",header=None)
```

3. Procedemos a separar los valores para tener las transacciones en el formato en el que podemos contar los elementos uno por uno de manera eficiente.

```
transacciones=[]
for i in range(7501):
    transacciones.append([str(data.values[i,j]) for j in range(0,20)])
```

4. Aplicamos el algoritmo a priori y establecemos los hiper parámetros, para obtener los elementos frecuentes.

```
reglas=apriori(transacciones,min_support=0.003,min_confidence=0.6,
min_lift=3,min_lenght=3)
```

5. Obtenemos los subconjuntos con su respectivo soporte

```
[RelationRecord(items=frozenset({'ground beef', 'spaghetti', 'cereals'}), support=0.00306
62578322890282, ordered_statistics=[OrderedStatistic(items_base=frozenset({'ground beef',
'cereals'}), items_add=frozenset({'spaghetti'}), confidence=0.6764705882352942, lift=3.88
53031258445188)]), RelationRecord(items=frozenset({'olive oil', 'spaghetti', 'tomatoe
s'}), support=0.004399413411545127, ordered_statistics=[OrderedStatistic(items_base=froze
nset({'olive oil', 'tomatoes'}), items_add=frozenset({'spaghetti'}), confidence=0.6111111
111111112, lift=3.5099115194827295)]), RelationRecord(items=frozenset({'ground beef', 'na
n', 'spaghetti', 'cereals'}), support=0.0030662578322890282, ordered_statistics=[OrderedS
tatistic(items_base=frozenset({'ground beef', 'nan', 'cereals'}), items_add=frozenset({'s
paghetti'}), confidence=0.6764705882352942, lift=3.8853031258445188)]), RelationRecord(it
ems=frozenset({'soup', 'milk', 'mineral water', 'frozen vegetables'}), support=0.00306625
78322890282, ordered_statistics=[OrderedStatistic(items_base=frozenset({'soup', 'milk',
'frozen vegetables'}), items_add=frozenset({'mineral water'}), confidence=0.7666666666666
666, lift=3.21631245339299), OrderedStatistic(items_base=frozenset({'soup', 'mineral wate
r', 'frozen vegetables'}), items_add=frozenset({'milk'}), confidence=0.6052631578947368,
lift=4.670863114576565)]), RelationRecord(items=frozenset({'olive oil', 'nan', 'spaghett
i', 'tomatoes'}), support=0.004399413411545127, ordered_statistics=[OrderedStatistic(item
s_base=frozenset({'olive oil', 'nan', 'tomatoes'}), items_add=frozenset({'spaghetti'}), c
onfidence=0.6111111111111112, lift=3.5099115194827295)]), RelationRecord(items=frozenset
({'nan', 'milk', 'frozen vegetables', 'soup', 'mineral water'}), support=0.00306625783228
```

Conclusiones

Por medio de la variación de los soporte, confidencia, el mínimo de elementos por conjunto vamos ir obteniendo mayor o menor número de relaciones, aunque en general a mayor cantidad no se tienen reglas de asociaciones fuertes puesto que bajamos los criterios de selecciones. Por ejemplo, tenemos carne, pasta y cereal pero este subconjunto sucede en el 0.306% de las veces, por lo cual son muy pocas veces en general.

Por eso mismo de la holgura en nuestros umbrales, obtenemos canastas que no nos suenen coherentes como lo son sopa, agua mineral y leche. Por lo cual también representa un problema encontrar los hiper parámetros adecuados para tener relaciones coherentes y generalizables.

Referencias

Korstanje, J. (22 de Sep de 2021). *Towards Data Science* . Obtenido de The Apriori algorithm: https://towardsdatascience.com/the-apriori-algorithm-5da3db9aea95