
quantil

Introducción a Aprendizaje no Supervisado

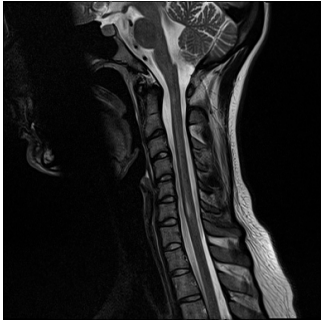
Felipe Suárez, Álvaro Riascos

25 de abril de 2017

Contenido

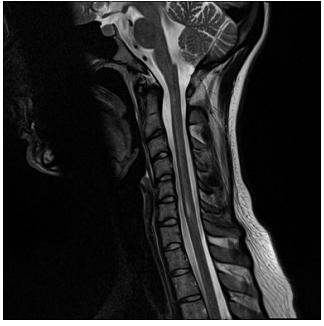
Problema

Considere la siguiente imagen médica.



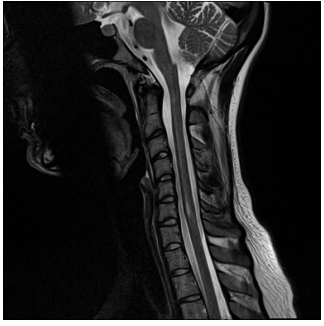
Problema

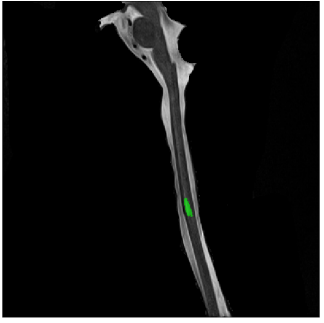
Considere la siguiente imagen médica.



Problema

Considere la siguiente imagen médica.





Historia

Hugo Steinhaus, 1887 - 1972

Historia

Hugo Steinhaus, 1887 - 1972

BULLETIN DE L'ACADÉMIE
POLONAISE DES SCIENCES
CL. III — VOL. IV, NO. 12, 1956

MATHÉMATIQUE

Sur la division des corps matériels en parties [1]

par
H. STEINHAUS

Présenté le 19 Octobre 1956

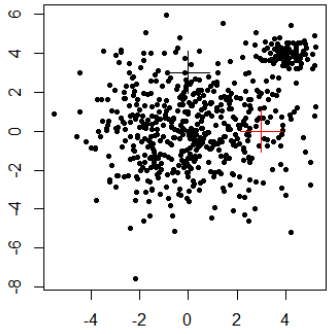
Un corps Q est, par définition, une répartition de matière dans l'espace, donnée par une fonction $f(P)$; on appelle cette fonction la *densité* du corps en question ; elle est définie pour tous les points P de l'espace ; elle est non-négative et mesurable. On suppose que *l'ensemble caractéristique* du corps $E = \underset{P}{E}\{f(P) > 0\}$ est borné et de mesure positive ; on suppose aussi que l'intégrale de $f(P)$ sur E est finie : c'est la *masse* du corps Q. On considère comme identiques deux corps dont les densités sont égales à un ensemble de mesure nulle près.

Definición

El método de agrupamiento (clustering) $k-$medias es un procedimiento de clasificación no supervisado basado en centroides. Las observaciones $x_1, \ldots, x_n$ son asignadas a la clase del centroide más cercano $\mu_1, \ldots, \mu_k$ a él.

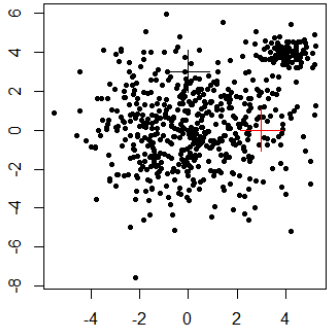
Definición

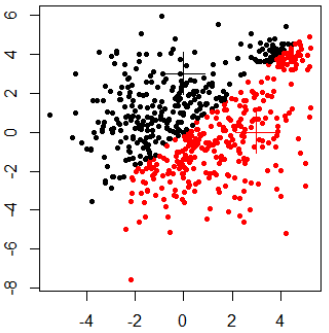
El método de agrupamiento (clustering) $k-$medias es un procedimiento de clasificación no supervisado basado en centroides. Las observaciones $x_1, \ldots, x_n$ son asignadas a la clase del centroide más cercano $\mu_1, \ldots, \mu_k$ a él.



Definición

El método de agrupamiento (clustering) $k-$medias es un procedimiento de clasificación no supervisado basado en centroides. Las observaciones $x_1, \ldots, x_n$ son asignadas a la clase del centroide más cercano $\mu_1, \ldots, \mu_k$ a él.





Definición

Definition

Sea $X = \{x_1, \ldots, x_n\} \subseteq \mathbb{R}^d$ un conjunto de n observaciones **numéricas**. El agrupamiento por $k-$medias consiste de particionar X en k subconjuntos $X = S_1 \cup \cdots \cup S_k$ por medio de k **centroides** $\mu_1, \ldots, \mu_k$ que minimizan la varianza:

$$\sum_{i=1}^{k} \sum_{x \in S_i} ||x - \mu_i||^2, \qquad \text{donde } \mu_i = \frac{1}{|S_i|} \sum_{x \in S_i} x.$$

Definición

Definition

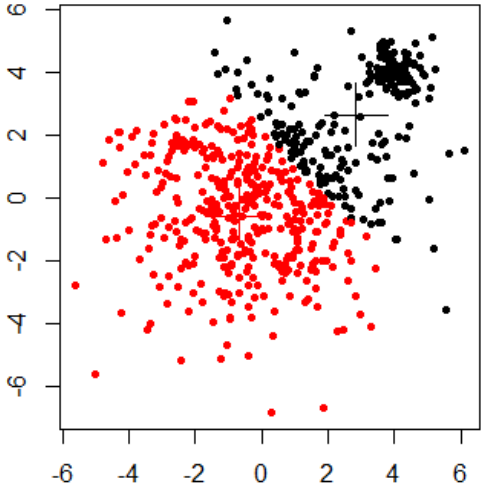
Sea $X = \{x_1, \ldots, x_n\} \subseteq \mathbb{R}^d$ un conjunto de n observaciones **numéricas**. El agrupamiento por $k-$medias consiste de particionar X en k subconjuntos $X = S_1 \cup \cdots \cup S_k$ por medio de k **centroides** $\mu_1, \ldots, \mu_k$ que minimizan la varianza:

$$\sum_{i=1}^{k} \sum_{x \in S_i} ||x - \mu_i||^2, \qquad \text{donde } \mu_i = \frac{1}{|S_i|} \sum_{x \in S_i} x.$$

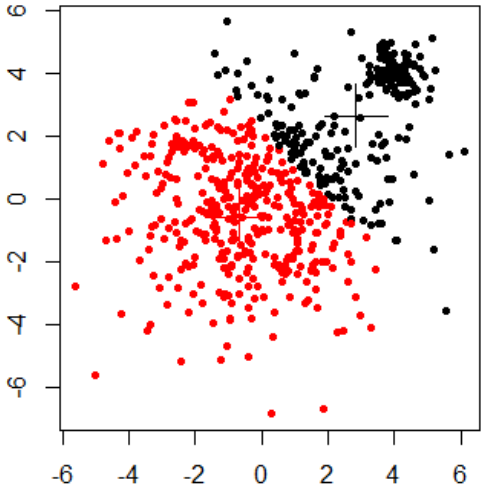
Propiedades

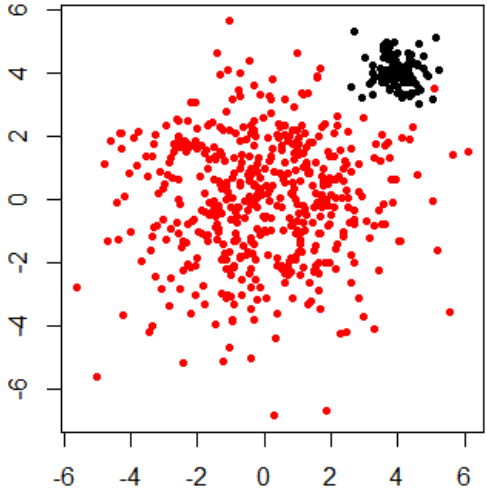
- Estimación de grupos con **simetría esférica**.
- Las variables tienen la **misma varianza**.
- Se asume apriori que todos los grupos son de **igual tamaño**.

Ejemplos



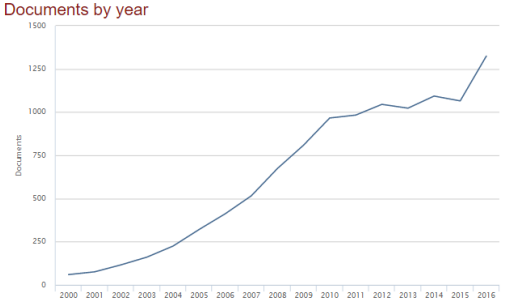
Ejemplos

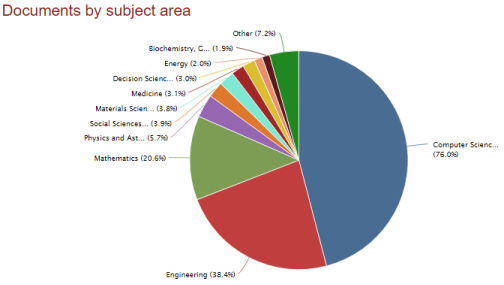




Estado del Arte

Quién lo utiliza? Todavía se investiga?





Algoritmo

El problema de optimización no es computacionalmente fácil de resolver: es NP-Dificil! Existen heurísticas que aproximan la solución a un mínimo local muy rápido.

Algoritmo

El problema de optimización no es computacionalmente fácil de resolver: es NP-Dificil! Existen heurísticas que aproximan la solución a un mínimo local muy rápido.

Definition

Input: Datos numéricos $X = \{x_1, \ldots, x_n\} \subseteq \mathbb{R}^d$, Número de grupos $k > 0$.
Output: Subgrupos $(x_1, y_1), \ldots, (x_n, y_n)$, con $y_i \in \{1, \ldots, k\}$.
Algoritmo:

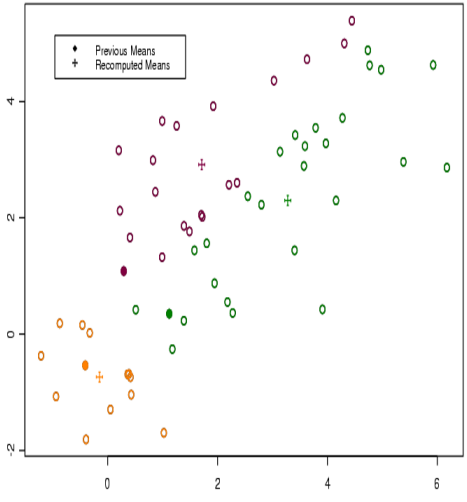
1. Inicialice los centros $\mu_1, \ldots, \mu_k$ aleatoriamente.
2. Repita
   2.1 Calcula el subgrupo de cada observación:

$$y_i = \arg\min_j ||x_i - \mu_j||$$

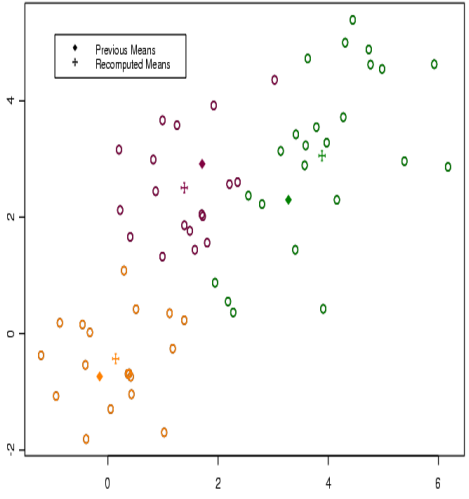
   2.2 Actualiza los centros:

$$\mu_i = \frac{1}{|S_i|} \sum_{x \in S_i} x.$$

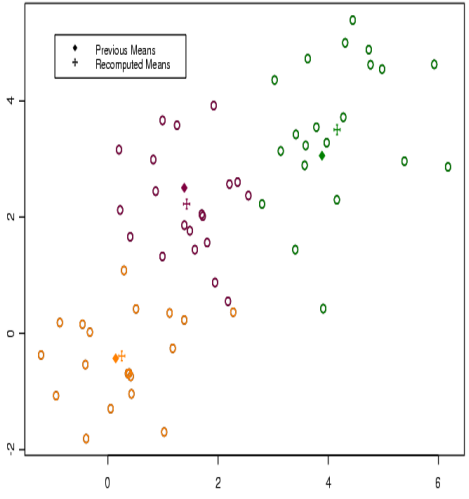
Ejemplo



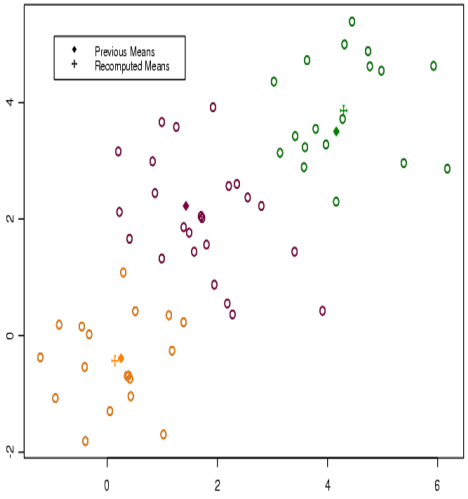
Ejemplo



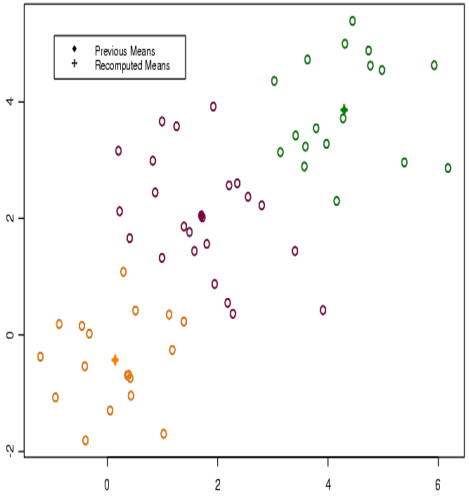
Ejemplo



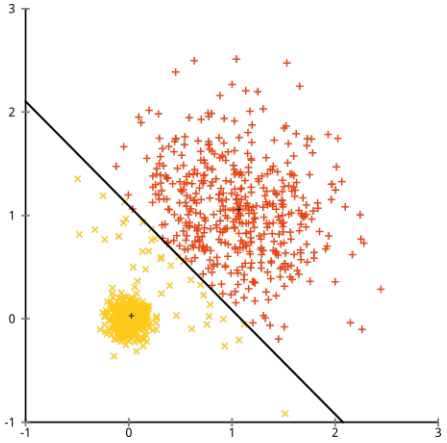
Ejemplo



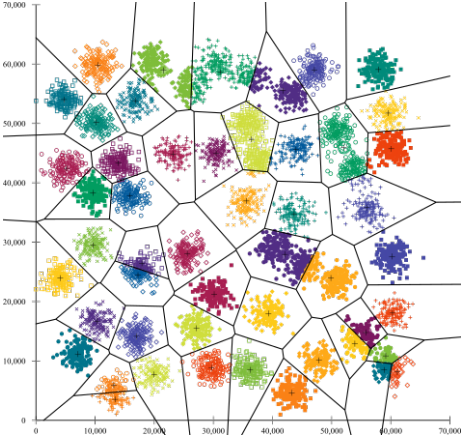
Ejemplo



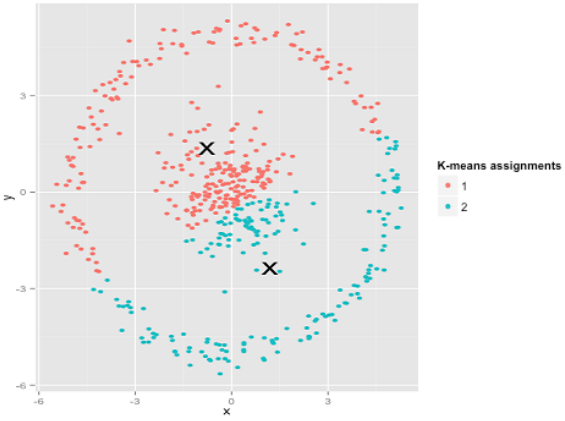
Casos Problemáticos



Casos Problemáticos



Casos Problemáticos



Alternativas

Existen alternativas al algoritmo heurístico. Las alternativas ayudan a evitar la agrupación discontinua y la simetría esférica.

1. Estandarizar variables para evitar sesgos de escala.

Alternativas

Existen alternativas al algoritmo heurístico. Las alternativas ayudan a evitar la agrupación discontinua y la simetría esférica.

1. Estandarizar variables para evitar sesgos de escala.
2. Fuzzy k-means:

$$f(X,S) = \sum_{i=1}^{k} \sum_{x \in S_i} u_i(x) ||x - \mu_i||^2, \qquad u_i(x) = \frac{||x - \mu_i||^{-2/r}}{\sum_i ||x - \mu_i||^{-2/r}}$$

Alternativas

Existen alternativas al algoritmo heurístico. Las alternativas ayudan a evitar la agrupación discontinua y la simetría esférica.

1. Estandarizar variables para evitar sesgos de escala.
2. Fuzzy k-means:

$$f(X,S) = \sum_{i=1}^{k} \sum_{x \in S_i} u_i(x) ||x - \mu_i||^2, \qquad u_i(x) = \frac{||x-\mu_i||^{-2/r}}{\sum_i ||x-\mu_i||^{-2/r}}$$

3. Mixturas Gaussianas: Se asume que la muestra es una suma de normales y se actualizan μ, σ por subgrupos ponderado por probabilidad $\mathbb{P}(x_i|\mu_j, \sigma_j)$.

Alternativas

Existen alternativas al algoritmo heurístico. Las alternativas ayudan a evitar la agrupación discontinua y la simetría esférica.

1. Estandarizar variables para evitar sesgos de escala.
2. Fuzzy k-means:

$$f(X,S)=\sum_{i=1}^{k}\sum_{x\in S_i}u_i(x)||x-\mu_i||^2, \qquad u_i(x)=\frac{||x-\mu_i||^{-2/r}}{\sum_i||x-\mu_i||^{-2/r}}$$

3. Mixturas Gaussianas: Se asume que la muestra es una suma de normales y se actualizan μ,σ por subgrupos ponderado por probabilidad $\mathbb{P}(x_i|\mu_j,\sigma_j)$.
4. Kernel k-medias: Se utiliza una función de kernel $k(x_i-\mu_j,x_i-\mu_j)$ a cambio de $||x_i-\mu_j||^2$.

Alternativas

Existen alternativas al algoritmo heurístico. Las alternativas ayudan a evitar la agrupación discontinua y la simetría esférica.

1. Estandarizar variables para evitar sesgos de escala.
2. Fuzzy k-means:

$$f(X,S)=\sum_{i=1}^{k}\sum_{x\in S_i}u_i(x)||x-\mu_i||^2, \qquad u_i(x)=\frac{||x-\mu_i||^{-2/r}}{\sum_i||x-\mu_i||^{-2/r}}$$

3. Mixturas Gaussianas: Se asume que la muestra es una suma de normales y se actualizan μ,σ por subgrupos ponderado por probabilidad $\mathbb{P}(x_i|\mu_j,\sigma_j)$.
4. Kernel k-medias: Se utiliza una función de kernel $k(x_i-\mu_j,x_i-\mu_j)$ a cambio de $||x_i-\mu_j||^2$.
5. Metricas no euclidianas: l_1, l_∞, l_p.

Garantías

1. Complejidad? NP-Difícil, pero las heurísticas son O(nd).

Garantías

1. Complejidad? NP-Difícil, pero las heurísticas son O(nd).
2. Convergencia? El algortimo heurístico converge rápidamente a un óptimo local.

Garantías

1. Complejidad? NP-Difícil, pero las heurísticas son O(nd).
2. Convergencia? El algortimo heurístico converge rápidamente a un óptimo local.
3. Robustez? No. De hecho es muy fácil que el algoritmo falle:

Garantías

1. Complejidad? NP-Difícil, pero las heurísticas son O(nd).
2. Convergencia? El algortimo heurístico converge rápidamente a un óptimo local.
3. Robustez? No. De hecho es muy fácil que el algoritmo falle:
   3.1 Muestra desbalanceada

Garantías

1. Complejidad? NP-Difícil, pero las heurísticas son O(nd).
2. Convergencia? El algortimo heurístico converge rápidamente a un óptimo local.
3. Robustez? No. De hecho es muy fácil que el algoritmo falle:
   - 3.1 Muestra desbalanceada
   - 3.2 Sensible a escala

Garantías

1. Complejidad? NP-Difícil, pero las heurísticas son O(nd).
2. Convergencia? El algortimo heurístico converge rápidamente a un óptimo local.
3. Robustez? No. De hecho es muy fácil que el algoritmo falle:
   - 3.1 Muestra desbalanceada
   - 3.2 Sensible a escala
   - 3.3 No acepta datos categórticos

Garantías

1. Complejidad? NP-Difícil, pero las heurísticas son O(nd).
2. Convergencia? El algortimo heurístico converge rápidamente a un óptimo local.
3. Robustez? No. De hecho es muy fácil que el algoritmo falle:
   - 3.1 Muestra desbalanceada
   - 3.2 Sensible a escala
   - 3.3 No acepta datos categórticos
   - 3.4 Qué pasa en dimensiones altas ($d > n$)?

Garantías

1. Complejidad? NP-Difícil, pero las heurísticas son O(nd).
2. Convergencia? El algortimo heurístico converge rápidamente a un óptimo local.
3. Robustez? No. De hecho es muy fácil que el algoritmo falle:
   - 3.1 Muestra desbalanceada
   - 3.2 Sensible a escala
   - 3.3 No acepta datos categórticos
   - 3.4 Qué pasa en dimensiones altas ($d > n$)?
   - 3.5 Convergencia a óptimo local

Garantías

1. Complejidad? NP-Difícil, pero las heurísticas son O(nd).
2. Convergencia? El algortimo heurístico converge rápidamente a un óptimo local.
3. Robustez? No. De hecho es muy fácil que el algoritmo falle:
   - 3.1 Muestra desbalanceada
   - 3.2 Sensible a escala
   - 3.3 No acepta datos categórticos
   - 3.4 Qué pasa en dimensiones altas ($d > n$)?
   - 3.5 Convergencia a óptimo local

Reglas de Asociación

Motivación

- Las reglas de asociación son muy usadas en bases de datos comerciales.
- Tenemos "canastas" de bienes que son vectores (muy grandes)
- ¿Podemos encontrar bienes que sean comprados juntos con mucha frecuencia?
- ¿Para qué podría servir algo así?

Motivación

- Un almacén puede organizar mejor sus repisas para mejorar sus ingresos.
- Hacer mercadeo cruzado de ciertas promociones que tengan que ver la una con la otra.
- Segmentar clientes de acuerdo a sus patrones de consumo.
- Diseñar experiencias para usuarios en páginas web.
- Recomendar películas en Netflix? Productos en Amazon?

Motivación

Aproximación al problema

- Quisiéramos encontrar algunas canastas $v_1, \ldots, v_n$ tales que su densidad de probabilidad $Pr(v_i)$ sea "alta".
- Este problema se llama *Mode Finding* o *Bump Hunting*.
- Sin embargo hay muchísimas canastas posibles (¿Cuántos productos tiene Éxito en sus repisas?)
- Una estimación de la densidad de estas canastas siempre dará algo muy pequeño como para ser confiable.
- Además calcular esas densidades es incomputable! ¿Cuántas canastas hay?

¿Cuántos productos vende Amazon?

Amazon.com	USA	253 million
Amazon.co.uk	UK	153 million
Amazon.de	Germany	141 million
Amazon.fr	France	119 million
Amazon.co.jp	Japan	108 million
Amazon.it	Italy	88 million
Amazon.es	Spain	86 million
Amazon.ca	Canada	56 million

Simplificar

- Podemos dejar de buscar canastas prototipos y buscar "regiones" en el consumo.
- Probablemente sea mucho más interesante y fácil de interpretar una relación entre dos o pocos productos.
- Podríamos agrupar tipos de productos en una sola variable dummy.
- Podemos pensar en la probabilidad de todas las canastas que contienen el artículo ω.

Con variables dummy

- Considerando un elemento de cada clase (posiblemente binaria) la ecuación se simplifica

$$P\left[\cap_{k\in\mathcal{K}}(z_k = 1)\right] = P\left[\prod_{k\in\mathcal{K}} z_k = 1\right] = T(\mathcal{K})$$

- Esto implica segmentar variables continuas que pueda haber en el análisis.
- El conjunto $\mathcal{K}$ se conoce como conjunto de bienes o *item set.*

Algoritmo Crecimiento de Árbol de patrones

Queremos encontrar reglas $X \to Y$ con buenas garantía probabilísticas (confianza). Estimamos las probabilidades de las canatas con la frecuencia en las base de transacciones:

$$supp(X) := \frac{T(X)}{T}, \qquad conf(X \to Y) = \frac{supp(X \text{ y } Y)}{supp(X)}.$$

Queremos tener en cuenta solo aquellas transacciones con un soporte mínimimo t y confianza mínima s. El algoritmo del árbol busca reglas aprovechando:

1. $X \to Y$ y $X \to Z$ implican $X \to Y \cup Z$
2. $X \subseteq Y$ implica $supp(X) \geq supp(Y)$

Algoritmo Crecimiento de Árbol de patrones

Queremos encontrar reglas $X \to Y$ con buenas garantía probabilísticas (confianza). Estimamos las probabilidades de las canatas con la frecuencia en las base de transacciones:

$$supp(X) := \frac{T(X)}{T}, \qquad conf(X \to Y) = \frac{supp(X \text{ y } Y)}{supp(X)}.$$

Queremos tener en cuenta solo aquellas transacciones con un soporte mínimimo t y confianza mínima s. El algoritmo del árbol busca reglas aprovechando:

1. $X \to Y$ y $X \to Z$ implican $X \to Y \cup Z$
2. $X \subseteq Y$ implica $supp(X) \geq supp(Y)$

El algoritmo realiza:

1. Saca las items con soporte $< t$.
2. Ordena los items descendientemente por soporte.
3. Orden las transacciones descendientemente por tamaño.
4. Construye un árbol con pesos en el que cada camino hacia la raíz es una transacción.
5. Para cada hoja del árbol, Y, la rama que la contiene es el antecedente que la implica X.

Ejemplo

- Se buscan reglas de asociación entre 9409 cuestionarios de compradores de un Mall en San Francisco.
- Se buscan relaciones entre las variables demográficas.
- Se parten las variables ordinales como "Dummies" por la mediana.
- El algoritmo encuentra 6288 reglas de asociación (difícil de interpretar) con soporte mayor a 10 %

Ejemplo

Association rule 2: Support 13.4%, confidence 80.8%, and lift 2.13.

$$\left[\begin{array}{rcl} \text{language in home} & = & \textit{English} \\ \text{householder status} & = & \textit{own} \\ \text{occupation} & = & \{\textit{professional/managerial}\} \end{array}\right] \Downarrow \text{income} \geq \textit{\$40,000}$$

Association rule 3: Support 26.5%, confidence 82.8% and lift 2.15.

$$\left[\begin{array}{rcl} \text{language in home} & = & \textit{English} \\ \text{income} & < & \textit{\$40,000} \\ \text{marital status} & = & \textit{not married} \\ \text{number of children} & = & \textit{0} \end{array}\right] \Downarrow \text{education} \notin \{\textit{college graduate, graduate study}\}$$

Limitaciones

- Está sesgado hacia observaciones más frecuentes que otras.
- Nunca encontraría Vodka $\Rightarrow$ Caviar. A pesar de tener un *lift* alto.
- Esto es porque intrínsicamente se está comparando con una **distribución de referencia uniforme**.
- Esto se puede solucionar en conjuntos pequeños con una estimación de la densidad basada en una distribución de referencia acorde a las probabilidades marginales.
- Se usa aprendizaje supervisado para ajustar una densidad relativa a esa distribución (por ejemplo una logística).


GRACIAS