Medidas de disimilitud y escalado multidimensional

Data Mining Tools

Prof. Ph.D Carlos Fernando Montoya Cubas

Introducción

Introducción

- En varios algoritmos necesitaremos una medida de (dis) similitud entre objetos.
  - Por ejemplo, en la estimación de densidad no paramétrica (la semana pasada), usamos una ventana con vecinos.
- Dependiendo del tipo de datos, son posibles varias mediciones.
  - Veremos solo los mas comunes
  - Algunos algoritmos asumen ciertos tipos de medidas, cuando corresponda, esto se resaltará.

Introducción

- Empezaremos considerando medidas de disimilitud
  - En la mayoría de los casos es trivial transformar los valores en similitud

$$s = -d, s = 1 - d, s = \frac{1}{1 + d}$$

Introducción

- Muchos de los algoritmos asumen una función de disimilitud / distancia entre un par de ejemplos, o una matriz de disimilitud / distancias entre todos los ejemplos.

$d(\mathbf{x}_i, \mathbf{x}_j)$ disimilitud entre el objeto $\mathbf{x}_i$ y $\mathbf{x}_j$

$$D_{N,N} = \begin{pmatrix} d(\mathbf{x}_1, \mathbf{x}_1) & \cdots & d(\mathbf{x}_1, \mathbf{x}_N) \\ \vdots & \vdots & \vdots \\ d(\mathbf{x}_N, \mathbf{x}_1) & \cdots & d(\mathbf{x}_N, \mathbf{x}_N) \end{pmatrix}$$

función de distancia

- Si $d(\mathbf{a}, \mathbf{b})$ es una métrica que calcula la distancia entre dos puntos **a** y **b**, tenemos que:
  - $d(\mathbf{a}, \mathbf{b}) \geq 0, \forall \mathbf{a}, \mathbf{b}$
  - $d(\mathbf{a}, \mathbf{b}) = 0$, apenas se $\mathbf{a} = \mathbf{b}$
  - $d(\mathbf{a}, \mathbf{b}) = d(\mathbf{b}, \mathbf{a})$
  - $d(\mathbf{a}, \mathbf{c}) \leq d(\mathbf{a}, \mathbf{b}) + d(\mathbf{b}, \mathbf{c})$
- Algunas medidas de distancia no obedecen todas las reglas, en este caso, no son correctamente métricas

Función de distancia euclidiana

- La función de distancia más utilizada es la distancia euclidiana:
  - Sí, el que ya conoces :)
  - Sean **a** y **b** dos vectores en $\mathbb{R}^M$:

$$d_{\mathrm{EUC}}(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_{m=1}^{M} (a_m - b_m)^2}$$

- A menudo solo necesitamos la relación de orden entre las distancias
- Por lo tanto, es común considerar la distancia euclidiana al cuadrado

Función de distancia euclidiana

- forma vectoriall

$$d^2_{\text{EUC}}(\mathbf{a}, \mathbf{b}) = (\mathbf{a} - \mathbf{b})^T (\mathbf{a} - \mathbf{b})$$

$$d^2_{\text{EUC}}(\mathbf{a}, \mathbf{b}) = \mathbf{a}^T \mathbf{a} + \mathbf{b}^T \mathbf{b} - 2\mathbf{a}^T \mathbf{b}$$

- En algunos contextos, esta forma de describir la distancia euclidiana es útil.
  - similitud de coseno
  - Escalamiento multidimensional

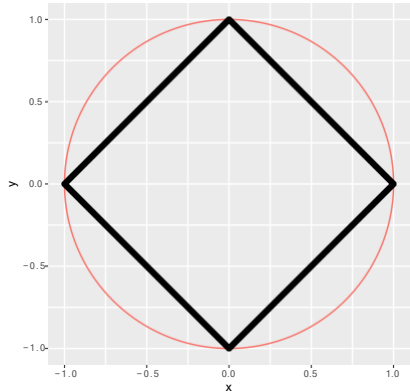
Función de distancia de Manhattan

- Tambien se llama *city-block* e *taxicab*

$$d_{\mathrm{MNH}}(\mathbf{a}, \mathbf{b}) = \sum_{m=1}^{M} |a_m - b_m|$$

- Pensar en las ciudades puede verse como caminar por las calles

Comparación entre la distancia euclidiana y Manhattan



Función de distancia entre atributos ordinales

- Un enfoque común es asumir que las clasificaciones están en una escala de intervalo.
  - De esta forma, la distancia euclidiana / Manhattan se puede aplicar en las clasificaciones.
- Por ejemplo:
  - Alumno 1 = [A, B, C, A, B]; Alumno 2 = [B, B, B, B, B]
  - Rankings = [F = 0, D = 1, C = 2, B = 3, A = 4]

$$d_{MNH}(\text{A1}, \text{A2}) = |4-3| + |3-3| + |2-3| + |4-3| + |3-3| = 3$$

Función de distancia entre atributos binarios

- Las mediciones para este tipo de atributos se basan en las siguientes cantidades:
  - f_{00} número de atributos en que **x** e **y** son iguales a 0
  - f_{11} número de atributos enque **x** e **y** son iguales a 1
  - f_{10} número de atributos en que **x** é 1 e **y** é 0
  - f_{01} número de atributos en que **x** é 0 e **y** é 1
- Coeficiente de emparejamiento simple:

$$s_{SMC} = \frac{f_{11} + f_{00}}{f_{01} + f_{10} + f_{11} + f_{00}}$$

$$d_{SMC} = 1 - s_{SMC}$$

Función de distancia entre atributos binarios

- ¿Y cuando 0 no es informativo?
  - Comparación entre matrículas de estudiantes
  - Comparación entre comprar en un supermercado
- Coeficiente de Jaccard

$$s_J = \frac{f_{11}}{f_{01} + f_{10} + f_{11}}$$

$$d_J = 1 - J$$

- cual es el valor de d_{SMC} e d_J para $\mathbf{x} = (1, 0, 0, 0, 0)$ e $\mathbf{y} = (0, 1, 0, 0, 1)$?

Función de distancia entre atributos nominales (caso general)

- La más utilizada es el emparejamiento sencillo.

$$s_{CS} = \sum_{m=1}^{M} \mathbb{K}\{x_m = y_m\}$$

$$d_{CS} = M - s_{CS}$$

- Para algunas aplicaciones hay medidas más útiles.
  - *Edit distance*
  - *Qwerty distance*

Funciones de distancia

- Hay varias medidas que se utilizan para calcular la distancia / disimilitud:
  - Diferencia entre distribuciones (divergencia KL)
  - Medida de coseno (de uso frecuente en minería de textos)
  - Medidas específicas para comparar imágenes (Similitud estructural)
  - ...
- Algunos algoritmos hacen ciertas suposiciones sobre distancias / disimilitudes
  - pueden ser necesarias adaptaciones en el algoritmo.
  - Puede ser necesario utilizar una medida específica para un determinado algoritmo.

Preprocesamiento

Preprocesamiento de atributos

- Considere la distancia euclidiana y los siguientes datos:
  - **x** = (23, 2500), **y** = (55, 3000)
  - Los atributos son: edad y salario
  - ¿Los atributos tienen el mismo peso al calcular la distancia?
  - ¿Como resolver?

Preprocesamiento de atributos

- Sea **z** el vector correspondiente a un atributo yz uno de sus valores
- Normalizar entre [0, 1]:

$$z' = \frac{z - min(\mathbf{z})}{max(\mathbf{z}) - min(\mathbf{z})}$$

- Transformar a una media igual a 0 y una desviación estándar igual a 1

$$z' = \frac{z - \bar{z}}{\sigma_z}$$

- ¿Cuándo usar uno u otro?

Preprocesamiento de atributos

- Muchos algoritmos basados en la distancia no aceptan atributos nominales.
  - Podemos convertir a un conjunto de atributos binarios (representación *one-of-K*)
- Ejemplos:
  - Atributo de función: {estudiante, técnico, profesor} -> {001, 010, 100}
- Hay otros enfoques.

Preprocesamiento de atributos

- Puede ser necesario el camino inverso (continuo -> discreto)
  - Los algoritmos que crean reglas pueden ser más eficientes con atributos discretos
  - Esta transformación se llama discretización.
  - hay varios enfoques, hoy hablaremos de dos simples:
    - ancho fijo
    - frecuencia fija

Discretización - Ancho fijo

- Separe el rango de datos ([*min*(**z**), *max*(**z**)]) en intervalos de igual tamaño especificados por el usuario
- Ejemplo:
  - separar en rangos de tamaño 5

$$\mathbf{z} = (32, 34, 43, 45, 51, 59, 62, 67, 68, 69, 70, 71, 72)$$

- El depósito en el que se encuentra el valor de la nota coincide con el nuevo valor del atributo.
  - El límite inferior del primer cubo y el límite superior del último se pueden $-\infty$ e $+\infty$

Discretización - Ancho fijo

- Ejemplo:
  - separar en rangos de tamaño 5

$$\mathbf{z} = (32, 34, 43, 45, 51, 59, 62, 67, 68, 69, 70, 71, 72)$$

$$[32, 37) = \{32, 34\} \quad [37, 42) = \{\} \quad [42, 47) = \{43, 45\}$$

$$[47, 52) = \{51\} \quad [52, 57) = \{\} \quad [57, 62) = \{\}$$

$$[62, 67) = \{62\} \quad [67, 72) = \{67, 68, 69, 70, 71\} \quad [72, 77) = \{72\}$$

Discretización - Frecuencia fija

- Separe el rango de datos ([*min*(**z**), *max*(**z**)]) en intervalos con aproximadamente el mismo número de objetos, y el usuario especificará el número de intervalos.
- Ejemplo, separar en 5 intervalos:
  - 13 itens, 5 intervalos 13/5 = 2.6, por lo que no todos los intervalos tendrán el mismo número de itens

  • **z** = (32, 34, 43, 45, 51, 59, 62, 67, 68, 69, 70, 71, 72)

- El depósito en el que se encuentra el valor de la nota coincide con el nuevo valor del atributo
- Evita que un cubo determinado tenga demasiados elementos mientras que otros están vacíos

Discretización - Frecuencia fija

- Ejemplo, separar en 5 intervalos:
  - 13 itens, 5 intervalos 13/5 = 2.6, por lo que no todos los intervalos tendrán el mismo número de itens

- **z** = (32, 34, 43, 45, 51, 59, 62, 67, 68, 69, 70, 71, 72)
  - [32, 45) = { 32, 34, 43} [45, 62) = { 45, 51, 59}
  - [62, 69) = {62, 67, 68} [69, 72) = {69, 70, 71} [72, +∞) = {72}

- El depósito en el que se encuentra el valor de la nota coincide con el nuevo valor del atributo

- Evita que un cubo determinado tenga demasiados elementos mientras que otros están vacíos

Escalado multidimensional

Escalado Multidimensional

- Vimos cómo obtener una matriz de distancias a partir de un conjunto de datos.
- Y si solo tenemos la matriz de distancias y queremos visualizar los datos de forma aproximada
- ¿Por qué no tendríamos los datos?
  - confidencialidad
  - Datos intrínsecamente relacionales (distancias obtenidas subjetivamente)

Escalado Multidimensional

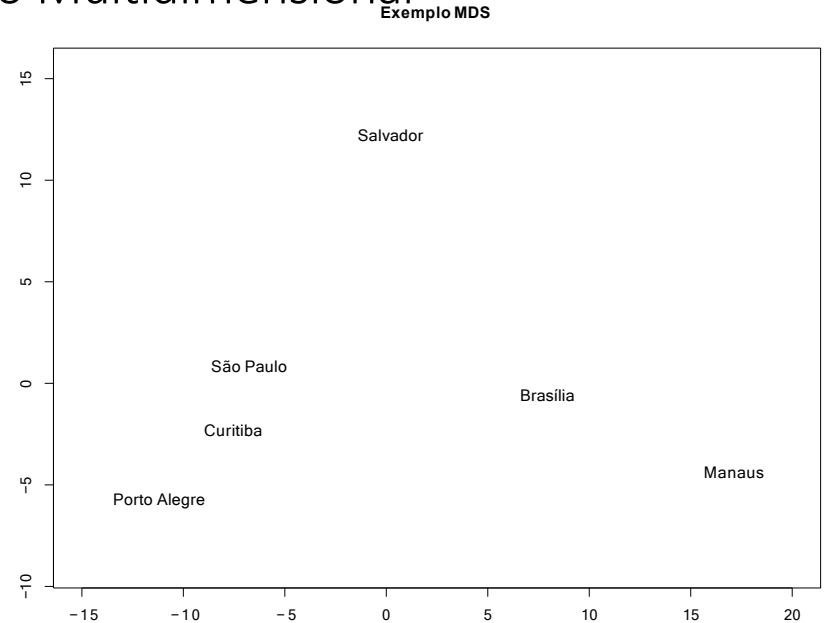
- Para este problema, las técnicas de *Multidimensional Scaling*
  - Hay varias técnicas posibles
  - Nos acercaremos al más tradicional, derivado de la distancia euclidiana.

Escalado Multidimensional

- Ejemplo:
  - Base de datos con la distancia (euclidiana) entre las coordenadas de los centros de 6 ciudades brasileñas

Curitiba	0.0	3.2	15.6	16.5	24.7	5.0
São Paulo	3.2	0.0	14.8	13.3	24.4	7.9
Brasília	15.6	14.8	0.0	15.0	9.9	19.8
Salvador	16.5	13.3	15.0	0.0	23.7	21.3
Manaus	24.7	24.4	9.9	23.7	0.0	28.3
Porto Alegre	5.0	7.9	19.8	21.3	28.3	0.0

Escalado Multidimensional



Escalado Multidimensional

- Minimiza el cuadrado de la diferencia entre distancias (par-a-par) y proyección (generalmente en un espacio más pequeño)

$$Stress(X1, \cdots, Xn) = \sqrt{\left(\sum_{i=1}^{n} \sum_{j=i, j \neq i}^{n} (||x_i, x_j|| - d_{i,j})^2\right)}$$

donde $d_{i,j}$ é es una métrica de disimilitud entre i e j en el espacio original y $||x_i, x_j||$ es una métrica de disimilitud en el espacio proyectado.

Escalado Multidimensional

- Método Clásico

1. Calcular matriz de distancias cuadradas $D^2_{N,N}$
2. Calcule a matriz $B = -\frac{1}{2} C D^2_{N,N} C$, en que C es una matriz central
3. Calcule los autovalores y autovectores de B
4. $\tilde{\mathbf{X}} = E_p \Lambda_p^{1/2}$, en que E_p é la matriz con los vectores propios y $\Lambda_p^{1/2}$ la matriz diagonal dos auto-valores

Escalado Multidimensional

- Existen métodos que utilizan técnicas de optimización interactivas para calcular la proyección.
- Se pueden utilizar diferentes métodos de optimización

Referencias

P. Tan, M. Steinbach e V. Kumar, Introduction to Data Mining. **Seção 2.4**

E. Alpaydin, Introduction to Machine Learning. **Seção 6.5**