

Cap. 3. Técnicas de clasificación no paramétricas.

3.1. Introducción

En el capítulo 3 ahora consideraremos algunas técnicas para aproximar densidades de probabilidad de manera "no paramétrica", en este caso se hacen pocas consideraciones o suposiciones sobre la "forma" de la distribución. Y pueden ser extendidas a un marco Bayesiano como en los trabajos de Walker, Neal, Müller y Quintana. En el contexto de reconocimiento de patrones, las técnicas no paramétricas se han popularizado en los últimos 20 años. Procedimientos para estimar

$$P(x|w) \rightarrow \text{Verosimilitud}$$

$$P(w|x) \rightarrow \text{probabilidades a posteriori}$$

a partir de patrones muestra.

+ Idea básica es estimar la forma de la densidad de probabilidad.

Se comienza con la noción de histograma y a partir de dichos histogramas se

estima la densidad de probabilidad, la cual puede ser una densidad marginal o una distribución condicional las cuales deben cumplir con el teorema del límite central. Las propiedades de una densidad basada en un histograma las discutimos a continuación:

Sea una v.a. continua x , su histograma es una simple partición de x en distintos bins de ancho Δ_i y luego se cuenta el número n_i de observaciones de x que caen en el i -ésimo bin. Con la finalidad de normalizar dicho conteo y aproximar una densidad de probabilidad, simplemente dividimos entre la cantidad total de observaciones de x , la denominada por N y entonces tenemos que la "distribución empírica" estará dada por

$$P_i = \frac{n_i}{N \Delta_i}$$

para la cual es fácil ver que $\int p(x)dx = 1$.

Por lo que la ecuación anterior brinda entonces un modelo para aproximar $p(x)$ la cual es

constante sobre el ancho de cada bin, y los bins son elegidos de tal manera $\Delta_i = \Delta$ para cada n_i .

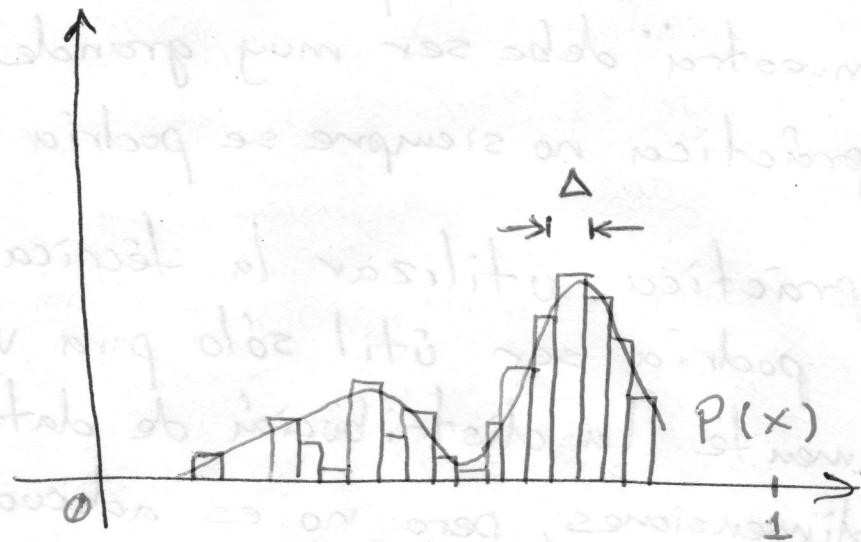


Figura 3.1. Histograma

Pueden tenerse 3 casos de ancho Δ :

- Δ es muy pequeño, la densidad empírica resultante está muy picuda y gran parte de la estructura no corresponde a $p(x)$,
- Δ es muy grande, la densidad empírica está bastante suavizada y los dos modos de $p(x)$ no están bien representados,
- Δ es de tamaño mediano, los resultados de aproximación a $p(x)$ son mejores como el caso de la figura 3.1, sin embargo aún así

la distribución requiere de ser suavizada, además como ya se vió el histograma depende del ancho de los bins y de su localización. Además, para tener una buena representación, la longitud de la muestra debe ser muy grande, lo cual en la práctica no siempre se podría tener.

En la práctica, utilizar la técnica de histogramas podría ser útil sólo para visualizar rápidamente la distribución de datos de 1 o 2 dimensiones, pero no es adecuada para la estimación de distribuciones y sus aplicaciones.

+ De lo anterior obtenemos dos lecciones importantes:

1: Estimación de la densidad de probabilidad en localidades particulares,

2: El valor de un parámetro de suavizado podría ser ni muy grande, ni muy pequeño de manera que la figura o contorno de la distribución sea muy cercana a la de $p(x)$.

IV

Tratando de resolver la problemática de los histogramas se propone utilizar estimadores

III

de la densidad basados en Kernels o núcleos.

3.2. Estimación por Kernels (Parzen)

Supongamos que contamos con observaciones x obtenidas a partir de una distribución de probabilidad $p(x)$ desconocida y en espacio D -dimensional, el cual podría ser un espacio de Euclides. Deseamos entonces estimar $p(x)$, en una pequeña región R la cual contiene a x . La masa de probabilidad asociada con dicha región, está dada por

$$P = \int_R p(x) dx.$$

Sea R una región que corresponde a un hiper cubo centrado en el punto x en el cual deseamos determinar la densidad de probabilidad, entonces existe una función

$$k(u) = \begin{cases} 1, & |u_i| \leq 1/2, \quad i = 1, 2, \dots, D \\ 0, & \text{de otra manera,} \end{cases}$$

la cual representa un cubo unitario centrado en el origen. La función $k(u)$ es un ejemplo

de función Kernel, y en nuestro contexto, también es conocida como ventana de Parzen. La cantidad $K((x - x_n)/h)$ tomará un valor 1 si x_n cae dentro del cubo de lado h centrado en x , de lo contrario tomará un valor de 0. Entonces, el número total de puntos que caen dentro del cubo estarán dados por

$$K = K(x) = \sum_{n=1}^N k\left(\frac{x - x_n}{h}\right).$$

Entonces, podemos aproximar $p(x)$ a partir de la ecuación anterior, de modo que

$$p(x) = \frac{1}{N} \sum_{n=1}^N \frac{1}{h^D} K\left(\frac{x - x_n}{h}\right),$$

en donde $V = h^D$ que corresponde al volumen del hipercubo de lado h en D -dimensiones.

La función $K(u)$ es simétrica, utilizando dicha propiedad, podemos re-interpretar la ecuación anterior no como un simple cubo centrado en x , si no como una suma sobre N cubos centrados en los N puntos x_n .

Dos problemas a resolver en este caso son:

- Una correcta selección de $h = ?$
- Selección de $k(u) = ?$

$k(u)$ debe cumplir las dos condiciones siguientes:

$$1.- \quad k(u) \geq 0,$$

$$2.- \quad \int k(u) du = 1.$$

Ver. figura 2.25 del libro de Bishop.

10 / Abril / 2019

3.3. Estimación del k -ésimo Vecino más cercano

(k -Nearest Neighbour) – k medias

Una de las dificultades que se tienen con la aproximación por Kernels para la estimación de densidades es que el parámetro h gobierna el ancho del Kernel y es de valor fijo para todos los Kernels. En regiones de alta densidad de datos un valor grande para h podría acarrear un sobre suavizado y por tanto no brindar una representación correcta. Por otro lado, si reducimos el valor de h podría llevarnos a estimaciones con picos o ruido que tampoco corresponden a una representación correcta, por lo que la estimación por Kernels requiere de una selección óptima de h la cual es dependiente de la localización en el espacio de los datos.

Para resolver este problema, los métodos basados en el vecino más cercano pueden ayudar para la estimación de la densidad de probabilidad. Nuevamente recordamos que deseamos aproximar

$$p(x) = \frac{k}{N \Delta}$$

en donde V es el volumen de R , K el número de puntos que caen en R . Ahora, en lugar de fijar el valor de V para la estimación local de la densidad y determinar el valor de K apropiado a partir de los datos, consideramos fijar el valor de K y utilizamos los datos para encontrar un valor apropiado para V .

Para llevar a cabo esto, ahora consideramos una pequeña esfera centrada en el punto x para el cual se desea estimar $p(x)$, y luego permitimos que el radio de la esfera crezca hasta que este contenga de manera precisa a los K puntos de los datos. La estimación de la densidad $p(x)$ mediante estas técnicas es mejor conocida como K -ésimo Vecino más cercano (K -NN), y justamente ahora la aproximación está gobernada por K , la selección del valor para K es ahora nuestro problema a resolver, y la suavidad de la estimación de $p(x)$ ahora depende de K .

El modelo producido por K -NN no es tan aproximado debido a que presenta picos debido

a que la integral sobre todos los espacios diverge (Fig. 2.26 del libro de Bishop).

A pesar de que la aproximación por Kernels es más precisa en forma que la aproximación k-NN, esta última técnica también se suele utilizar en tareas de clasificación. Si se aplica dicha técnica a cada clase de manera separada y si se utiliza el teorema de Bayes.

Ahora, supongamos que contamos con un conjunto de datos que tiene N_k puntos dentro de la clase C_k , de modo que tenemos un total de puntos N , por lo que $\sum_k N_k = N$. Si deseamos clasificar un nuevo punto x , entonces consideramos una esfera centrada en x que contiene de manera precisa K puntos para su clase.

Suponga que dicha esfera tiene un volumen V y contiene K_k puntos para la clase C_k , entonces la densidad estimada asociada a cada clase se puede escribir como

$$p(x|C_k) = \frac{K_k}{N_k V},$$

De manera similar la densidad incondicional estará dada por

$$p(x) = \frac{K}{NV},$$

mientras que las distribuciones a priori de la clase serán

$$p(C_k) = \frac{N_k}{N}.$$

Entonces, combinando las tres ecuaciones anteriores en la regla de Bayes, tenemos que la probabilidad a posteriori se puede escribir como

$$p(C_k | x) = \frac{p(x | C_k) p(C_k)}{p(x)} \\ = \frac{K_k}{K}.$$

IV Si deseamos minimizar la probabilidad de una mala clasificación, lo podremos realizar ~~aceudiendo~~ asignando un punto test x a la clase que tiene la probabilidad posterior más

grande, que corresponde al valor más grande de la relación K_k/K . Entonces, para clasificar un punto nuevo, identificaremos los K -ésimos puntos más cercanos del conjunto de puntos de entrenamiento y en seguida asignamos el nuevo punto a la clase que tiene el número más grande de representantes de entre todo el conjunto.

El caso particular en que $K=1$ es denominado regla del vecino más cercano, debido a que el punto test es simplemente asignado a la misma clase que la del punto más cercano que pertenece al conjunto de entrenamiento. (Ver figura 2.27 de Bishop)

Un valor pequeño para K produce que se tengan muchas regiones pequeñas para cada clase (muchas clases), mientras un valor grande para K

④ produce una poca cantidad de grandes regiones para cada clase (pocas clases). (Ver figura 2.28 de Bishop).

Una propiedad interesante del clasificador NN cuando $K=1$, es que en el límite cuando $N \rightarrow \infty$, el error de clasificación (error rate) no es mayor que al doble del error mínimo alcanzable para un clasificador óptimo.

Como ya se discutió, tanto los estimadores de Kernels, como los estimadores K-NN necesitan contar con conjuntos de datos de entrenamiento y prueba, los cuales estarán previamente almacenados, y tradicionalmente los tiempos de cómputo son bastante costosos.

- Caso particular de método K-medias
- Métodos no paramétricos tienen limitaciones
- Métodos paramétricos limitan en la forma de $p(x)$.

VI

V

3.4. Regla de decisión de los vecinos más cercanos - Cuantificación Vectorial (CV).

Para construir un cuantificador vectorial o libro de códigos e implementar así la CV, el procedimiento a seguir es el siguiente:

- a) Se debe contar con un conjunto de una gran cantidad de vectores de características $\{x_1, x_2, \dots, x_L\}$ que formarán un conjunto de "entrenamiento". Dicho conjunto es utilizado para generar el "conjunto óptimo" de vectores del libro de códigos que represente la variabilidad espectral observada en el conjunto de entrenamiento.

$M = 2^B$, B : Bit, $L \gg M$, en la práctica

$10M = L$ es un buen compromiso.

- b) Una medida de similitud o distancia d , entre un par de vectores y un conjunto de vectores de entrenamiento, tales que pueden ser "clasificados" arbitrariamente por los vectores del libro de códigos.

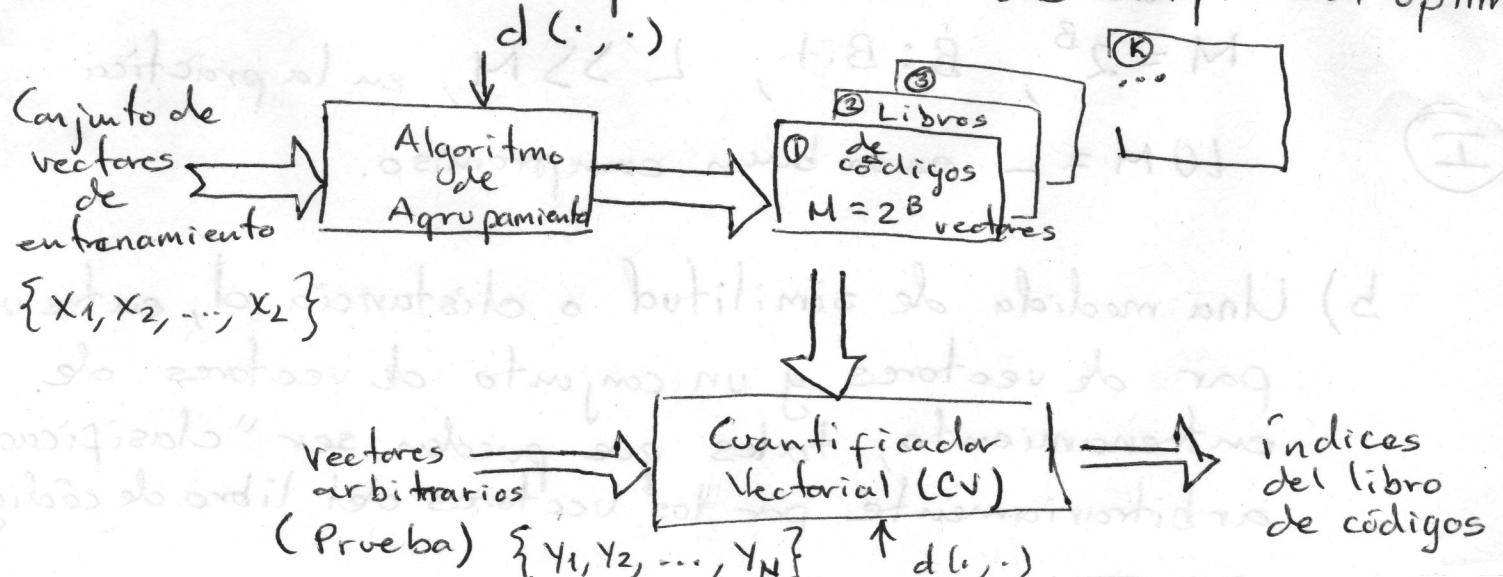
(I)

c) Un procedimiento para la obtención de "centroïdes", el cual trabajará sobre la base de que se realiza una partición del espacio de L vectores de entrenamiento para "agruparlos" en M subconjuntos.

Las componentes del libro de códigos son seleccionadas como los centros o centroïdes de cada uno de los M subespacios.

d) Finalmente, para la utilización del CV es necesario definir un proceso que clasifique vectores arbitrarios, tal que elija el vector del libro de códigos más cercano al vector de entrada y utilice el índice de (B -bits) como su representación resultante. Este procedimiento es referido como regla del vecino más cercano (RVMC ó Nearest Neighbor) o bien, también conocido como procedimiento de codificación óptima.

II



Sea $\mathbb{C} = \{z_i\}_{i=1}^M$ una colección de vectores de reproducción o libro de códigos y sea $d(x_t, z_i)$ la medida de distorsión que se utilizará entre x_t y z_i . El objetivo de un CV es encontrar el mejor código de \mathbb{C} que alcanza una esperanza de distorsión mínima $E\{d(x_t, z_i)\}$ en donde x_t es considerado como un vector arbitrario de las mismas dimensiones que z_i .

El libro de códigos se diseña para minimizar

$$D = \frac{1}{T} \sum_{t=1}^T d(x_t, \hat{x}_t),$$

en donde

$$\hat{x}_t = \underset{z_i \in \mathbb{C}}{\operatorname{argmin}} \{d(x_t, z_i)\}.$$

Considere un conjunto de vectores $\{x_i\}_{i=1}^L$ y una medida $d(x, y)$, suponemos que estos vectores se asignan a una misma etiqueta de grupo. Entonces, el centroide de $\{x_i\}_{i=1}^L$ se define como el vector \bar{y} que minimiza la distorsión promedio

$$\bar{y} = \underset{y}{\operatorname{argmin}} \frac{1}{L} \sum_{i=1}^L d(x_i, y).$$

La solución, depende bastante de la selección de la medida de distorsión. Cuando x_i e y son vectores de la forma: $x_i = [x_{i1}, x_{i2}, \dots, x_{ik}]$ e $y = [y_1, y_2, \dots, y_k]$ medidos en un espacio K -dimensional con norma L_2 (distancia Euclídea), entonces el centroide es el valor medio del conjunto, esto es

$$\bar{y} = \frac{1}{L} \sum_{i=1}^L x_i,$$

Otra distancia con robustez intrínseca, es la de valor absoluto o L_1 , en donde

$$d(x_i, y) = \sum_{k=1}^K |x_{ik} - y_k|,$$

en este caso el centroide \bar{y} corresponde a la mediana del vector $\{x_i\}_{i=1}^L$.

ALGORITMO GENERALIZADO DE LLOYD o K-MEDIAS

Paso 1: Inicialización: Elección de M vectores, tal que $M \ll L$ y será el conjunto inicial de centroides de C .

Paso 2: Búsqueda del vecino más cercano: Para cada uno de los vectores de entrenamiento hallar el centroide más cercano en el libro de códigos actual (de acuerdo a una distancia), y asignando cada vector a la celda o región correspondiente (asociar con centroide más cercano)

06/Mayo/2019

Paso 3º: Actualización de centroides: actualizar los centroides en cada región o subconjunto usando los vectores de entrenamiento asignados a cada región.

Paso 4º: Iteración: Repetir los pasos 2 y 3 hasta que se alcance una distancia menor a ϵ . (ϵ es un umbral).

ALGORITMO DE LLOYD MODIFICADO POR LINDE-BUZO-GRAY (LBG).

Una manera eficiente de implementar el algoritmo de Lloyd, es hacer un diseño por etapas, para ello se utilizan conjuntamente algunos pasos del algoritmo anterior y la técnica de división binaria para los centroides, se parte de un libro de tamaño 2 y se sigue el proceso de división hasta alcanzar el tamaño deseado del libro (M).

Paso 1º: Diseño de un libro de tamaño uno: El primer vector del conjunto de entrenamiento es asignado como el centroide del libro C (no hay iteración).

Paso 2º: Duplicar el tamaño del libro: De acuerdo a la división de cada centroide dentro del libro de códigos actual Y_n , se sigue la siguiente regla

$$Y_n^+ = Y_n(1+\epsilon)$$

$$Y_n^- = Y_n(1-\epsilon)$$

en donde n varía desde 1 hasta el tamaño deseado del libro M , y ϵ es un parámetro de división ($0.01 \leq \epsilon \leq 0.05$).

Paso 3: Uso del algoritmo iterativo K-medias, para elegir el mejor conjunto de centroides.

Paso 4: Iteración: Repetir los pasos 2 y 3 hasta alcanzar el tamaño M del libro de códigos deseado.

La compresión de datos usando CV ha recibido mucha atención en las últimas décadas pues se brindan tasas de compresión prometedoras y el algoritmo o algoritmos son relativamente simples.

VI

IV

V

En la fase de codificación, el centroide del libro de códigos más similar a cada vector de la señal de entrada puede ser encontrado: vector \rightarrow clu de los centroides del libro.

Otra fusión al algoritmo básico de Lloyd y de LBG (Linde, Buzo y Gray) es un algoritmo de ahorro en cálculos propuesto por Huang en 1992 para procesamiento de imágenes. Dicho algoritmo tiene la misma calidad y es equivalente al de búsqueda completa, pero decrece grandemente el tiempo de búsqueda.

EJEMPLO 1 DE LBG (Artículo 1980).

Centroides orden vect.

- Inicialización: $M=4$, $K=2$, $\epsilon = 0.001$ y $n=12$
 $\chi_j = (\chi_1, \chi_2, \dots, \chi_{12})$

SECUENCIA DE ENTRENAMIENTO

$$\begin{array}{ll}
 \chi_1 = (-0.37449, 0.98719) & \chi_7 = (-0.59161, 0.17968) \\
 \chi_2 = (0.63919, -0.11875) & \chi_8 = (0.14093, 1.76413) \\
 \chi_3 = (-0.83293, 0.60645) & \chi_9 = (0.70898, -0.35017) \\
 \chi_4 = (-0.70534, -1.21856) & \chi_{10} = (0.30038, 0.79836) \\
 \chi_5 = (-0.28952, -0.94821) & \chi_{11} = (0.30165, 1.06552) \\
 \chi_6 = (1.09924, 0.51600) & \chi_{12} = (-0.37801, -0.32708) \\
 \hat{A}_0 = \{(2,2), (2,-2), (-2,2), (-2,-2)\} = \{\chi_1, \chi_2, \chi_3, \chi_4\}
 \end{array}$$

$$m = \emptyset$$

$$D_{-1} = \infty \quad \text{o} \text{máximo valor de la Compu} \quad D_{-1} = 9.99 \times 10^{62}$$

- Busqueda del vecino más cercano: Encontrar $P(\hat{A}_0) = \{S_1, S_2, S_3, S_4\}$
 $\hat{A}_m = \{y_i; i=1, \dots, M\}$

Distorsión mínima $P(\hat{A}_m) = \{S_i; i=1, \dots, M\}$

para $x_j \in S_i \Leftrightarrow d(x_j, y_i) \leq d(x_j, y_m)$ para todo m

Luego calcular $D_m = D(\hat{A}_m, P(\hat{A}_m)) = \frac{1}{n} \sum_{j=0}^{n-1} \min_{y \in \hat{A}_m} d(x_j, y)$

$$S_1 = \{x_6, x_8, x_{10}, x_{11}\}$$

$$S_2 = \{x_2, x_9\}$$

$$S_3 = \{x_1, x_3, x_7\}$$

$$S_4 = \{x_4, x_5, x_{12}\}$$

$$d(x, \hat{x}) = \sum_{i=0}^{k-1} |x_i - \hat{x}_i|^2$$

$$\text{Calculando } D_0 = \frac{1}{12} \sum_{j=1}^{12} \min_{y \in \hat{A}_0} d(x_j, y) = 2.0172$$

Actualización del centroide: Si $(D_{m-1} - D_m) / D_m \leq \epsilon$ terminar con el cuantificador \hat{A}_m actual; si no continuar $(D_{m-1} - D_m) / D_0 = (9.99 \times 10^{-2} - 2.0172) / 2.0172 > 0.002$

* Antes de hacer la iteración para $m=1$, encontrar el alfabeto óptimo de reproducción (act. de centroide):

$$\hat{A}_1 \triangleq \hat{x}(P(\hat{A}_0)) = \{\hat{x}(S_i); i=1, \dots, M\}$$

$$\hat{x}(S_1) = (x_6 + x_8 + x_{10} + x_{11}) / 4 = (0.46055, 1.0360)$$

$$\hat{x}(S_2) = (x_2 + x_9) / 2 = \text{[redacted]} (0.674085, -0.23446)$$

$$\hat{x}(S_3) = (x_1 + x_3 + x_7) / 3 = (-0.599676, 0.591106)$$

$$\hat{x}(S_4) = (x_4 + x_5 + x_{12}) / 3 = (-0.457623, -0.831283)$$

$$\hat{A}_1 = \{(0.46055, 1.0360), (0.674085, -0.23446), (-0.599676, 0.591106), (-0.457623, -0.831283)\}$$

y ahora nos regresamos a la búsqueda del vecino(s) nuevo(s) más cercano(s).

(21)

- $m=1$: Encontrar $P(\hat{A}_1) = \{S_1, S_2, S_3, S_4\}$
 evaluando $d(x_j, y_i) \leq d(x_j, y_m)$, quedan los mismos
 $x_j \in S_i \therefore P(\hat{A}_1) = P(\hat{A}_0)$

Calcular

$$D_1 = \frac{1}{12} \sum_{j=1}^{12} \min_{y \in \hat{A}_1} d(x_j, y) = 0.0997308$$

- Actualización del Centroide: $(D_0 - D_1) / D_1 = \frac{(2.0132 - 0.0997308)}{0.0997308}$
 $(D_0 - D_1) / D_1 = 19.2254 > 0.001$

* Antes de hacer la iteración para $m=2$, hacer la actualización de centroides:

$$\hat{A}_2 \hat{=} \hat{x}(P(\hat{A}_1)) = \hat{A}_1, \text{ puesto que } P(\hat{A}_1) = P(\hat{A}_0)$$

y se tiene que

$$\hat{x}(P(\hat{A}_1)) = \hat{x}(P(\hat{A}_0)), \text{ por lo que } \hat{A}_2 = \hat{A}_1$$

- $m=2$: Encontrar $P(\hat{A}_2) = \{S_1, S_2, S_3, S_4\}$
 puesto que $P(\hat{A}_1) = P(\hat{A}_0) = P(\hat{A}_2)$, entonces

$$D_2 = D_1$$

- Actualización del Centroide: $(D_1 - D_2) / D_2 = 0 < 0.001$
 y el cuantificador final es $\{\hat{A}_1, P(\hat{A}_1)\}$

El algoritmo Anillo - Esfera, es otro algoritmo basado en las ideas de Huang (1992) y se basa en los siguientes principios:

Sea $C = \{C_j; j=1, \dots, n\}$ el libro de códigos de tamaño n , $C_j = \{c_{j1}, \dots, c_{jk}\}$ un centroide

3.5. Edición del conjunto de entrenamiento

La clasificación consiste en el proceso de asignar a un objeto específico, el nombre de la clase a la que pertenece (supervizado). Las clases resultan de un problema de predicción, donde cada clase corresponde a la salida posible de una función que predice a partir de los atributos con los que describimos los elementos de la base de datos. Entonces, la necesidad de un clasificador surge por requerimientos de disponer de un procedimiento mecánico mucho más rápido que un supervisor humano y que a la vez pueda evitar sesgos y prejuicios adoptados por un experto.

I) Para la clasificación partimos de la existencia de un conjunto de entrenamiento de longitud $L \circ T$, de pares $\{x_i, y_i\}$ de datos. Se requiere de un mecanismo capaz de crear un modelo a partir de dicho conjunto de entrenamiento.

- Lineal Regresivo
- Redes Neuronales
- Árboles de decisión
- Máquinas de soporte vectorial (SVM)

- clasificadores probabilísticos o estocásticos.
- y clasificadores por vecindarios o no paramétricos como k-NN.

Los conjuntos de datos en su estado ~~original~~ original pueden traer algunas características que afecten el proceso de clasificación:

- presencia de ruido
- valores ausentes o datos aberrantes
- exceso de características que afectan la separabilidad de clases
- "ejemplos" o pares de datos repetidos
- "ejemplos" que no son una buena representación de su clase
- baja representatividad de uno de los conceptos a clasificar.

Dados los puntos previos, se plantea una etapa conocida como preprocessamiento o preparación, o edición de los datos de entrada a los clasificadores. La idea es "mejorar la calidad", de modo que los algoritmos de extracción de conocimiento puedan obtener mayor y mejor información.

Por ejemplo, actualmente uno de los grandes desafíos que enfrenta la minería de datos es el aprendizaje a partir de datos "no balanceados". La ocurrencia de sucesos poco frecuentes ha dado lugar a que exista una desproporción considerable entre el número de ejemplos en cada clase, la clase menos representativa suele ser la de mayor interés en el problema de clasificación, mientras que la más representada constituye simplemente contra ejemplos de la menos representativa.

- Desigual distribución de ejemplos por clases

III

97%
Ejemplos Clase 1
3% Ejemplos Clase 2

"no balance"

50%
Ejemplos Clase 1
50%
Ejemplos Clase 2

"balance"

VI

Técnicas para lidiar con el problema de "no balanceo" o desbalanceo de clases

- 1) Nivel de datos: - Remuestreo, - bajo-muestreo
- sobre muestreo

2) A nivel de algoritmos: - Diseño de algoritmos específicos que toman en cuenta dicha desproporción.

3) Aprendizaje sensible a la función de costo: - Este tipo de algoritmos de aprendizaje incorpora soluciones a nivel de los datos, a nivel de los algoritmos, o a la combinación. Se asigna mayor costo al error de fallar en un ejemplo de clase con muchos datos y menor costo al error de fallar en un ejemplo de clase con pocos datos.

4) Soluciones con multiclasicadores: - El uso de técnicas de multiclasicación para datos no balanceados, corresponde a la combinación de algoritmo de aprendizaje en conjunto con alguna técnica, de remuestreo, por ejemplo.

* Atención con el sobre aprendizaje *

Que las clases sean balanceadas o no balanceadas los datos deben brindar:

a) Información que pueda permitir la discriminación entre clases

b) Eliminar información redundante e irrelevante

c) se puede reducir la dimensionalidad de los datos (vectores de características).

+ Mejorar el desempeño del clasificador.

+ Establecer compromiso complejidad / rendimiento

+ Al conjunto de datos preprocesado se le puele denominar: Corpus

Es difícil poder establecer una relación exacta entre error de clasificación, N = número de patrones de entrenamiento, y d = número de características, se recomienda establecer el compromiso

$$N/d \geq 10.$$

Entonces, es importante llevar a cabo una selección de características relevantes, lo cual nos permite una mejora en la eficiencia de la clasificación.

- k-NN y árboles de decisión son sensibles a caracteristicas irrelevantes o ruido.
- Bayes es sensible a las características redundantes. (Método básico).

+ Si dos variables están perfectamente correlacionadas, entonces son redundantes y no se agrega información adicional

La reducción de la dimensionalidad nos permite entonces llevar a cabo una selección de características del espacio muestral

Dado un conjunto $\{x_1, x_2, \dots, x_p\} = X$

$$J(\tilde{X}_d) = \max_{X \in \tilde{X}_d} J(X),$$

en donde \tilde{X}_d es el conjunto de todos los subconjuntos de tamaño d , y se optimiza sobre todos los conjuntos posibles con d características, las cuales se tomaran de las p iniciales ($d \leq p$).

En el espacio transformado, podemos encontrar una representación de características de dimensión menor, en donde A es un conjunto de transformaciones posibles,

$$J(A^*) = \max_{A \in A} J(A(x)),$$

en donde se optimiza sobre las transformaciones posibles y se obtienen nuevas características $y = A^*(x)$

La selección de características puede realizarse entonces por varias metodologías agrupadas de acuerdo a tres enfoques

- i) Filtrado: de forma independiente al clasificador y con un criterio de relevancia.
- ii) Encapsulado o wrapping: en función al desempeño del ~~del~~ clasificador. Con una estrategia de búsqueda.
- iii) Intrínseco o embedding: la selección se lleva a cabo en el proceso de aprendizaje.

VII