LTI Cinvestav

Clasificador bayesiano

Reconocimiento de patrones

Profesor: Dr. Wilfrido Gómez Flores

Estudiante: Rafael Pérez Torres

1. Introducción

El problema de la clasificación supervisada consiste en asignar una clase a un patrón anónimo, basándose en información obtenida previamente a través del estudio de características presentes en patrones que se sabe pertenecen a alguna clase en particular. Existen diferentes familias de clasificadores basándose en el tipo de enfoque que siguen para estudiar a los datos y realizar la clasificación.

La familia de clasificadores que se basan en aspectos probabilísticos tienen su origen en el teorema de Bayes, enunciado por el matemático Thomas Bayes en 1763, el cual vincula la probabilidad de que suceda un evento A dado otro evento B. A partir de este sencillo concepto, se construye una serie de elementos que permiten obtener la probabilidad de que un nuevo objeto pertenezca a alguna de las clases existentes en nuestro modelo.

El presente documento muestra de forma breve la descripción de la implementación y prueba del clasificador Bayes, empleando la *PDF* gaussiana y utilizando como criterio de decisión la regla *MAP*.

2. Marco teórico

En 1763 Thomas Bayes enuncia su *teorema de Bayes* que vincula la probabilidad de que suceda un evento A dado B con la probabilidad de B dado A. Si se sabe que los elementos de un conjunto de sucesos$\Omega = \{A_1, A_2, \cdots, A_n\}$tienen una probabilidad mayor a cero y son mutuamente excluyentes, y además se conoce de un suceso B todas las probabilidades condicionales $P(B \mid A_i)$, entonces la probabilidad de que alguno de los eventos $A_i$suceda dado B puede ser determinada utilizando la regla de Bayes:

$$P(A_i \mid B) = \frac{P(B \mid A_i)\, P(A_i)}{P(B)} \tag{1}$$

donde $P(A_i)$ son todas las probabilidades *a priori*, $P(B \mid A_i)$es la probabilidad de B en la hipótesis A_i y $P(A_i \mid B)$ son las probabilidades *a posteriori*.

Este enfoque puede ser trasladado al contexto de la clasificación. Suponiendo que se puede clasificar en c posibles clases $\omega_1, \omega_2, \ldots, \omega_c$a un patrón desconocido que es representado por un vector de características $\mathbf{x}$, entonces se pueden construir las probabilidades condicionales de que el patrón $(\mathbf{x})$ pertenezca a algunas de las clases $(\omega_1, \omega_2, \ldots, \omega_c)$ mediante:

$$P(\omega_i \mid x), \forall i = 1, 2, \ldots, c$$

y calcularse tal como se indica en la Ecuación 1. Dado que se está migrando la regla de Bayes al contexto de la clasificación, el valor de las probabilidades condicionales será representado por el valor obtenido por alguna función que represente la probabilidad de las clases.

Con el fin de encontrar aproximaciones a las probabilidades descritas por los datos del mundo real y alcanzar la *máxima verosimilitud*, se utiliza una *PDF*[1] que permita modelarlos y representarlos. En la práctica, la *PDF* gaussiana (o función de densidad normal) es la más utilizada por su similitud y debido a que sigue al *TLC*[2]. Este teorema permite describir las características de la población de medias creadas a partir de una cantidad infinita de muestras aleatorias de tamaño N tomadas de una población padre.

La *PDF* gaussiana para l-dimensiones puede ser calculada mediante:

$$P(\mathbf{x} \mid \omega_i) = \frac{1}{(2\pi)^{l/2} |\Sigma_i|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu_i)^T \Sigma_i^{-1} (\mathbf{x} - \mu_i)\right)$$

[1]*PDF*, función de densidad de probabilidad, por sus siglas en inglés.

[2]*TLC*, teorema del límite central

donde $\mu_i = E[x]$ es el valor medio de la clase ω_i, Σ_i es la matriz de covarianza de tamaño lxl, y $|\Sigma_i|$y Σ_i^{-1} denotan el determinante e inversa de Σ_i, respectivamente.

A partir del teorema de Bayes y de la caracterización de las probabilidades a través de la PDF gaussiana, es posible aplicar dos reglas de decisión al clasificar patrones:

- Máxima verosimilitud (*ML*): en la que se asume que todas las hipótesis (clases) son *equiprobables*, por lo que solamente se requiere conocer $P(\mathbf{x} \mid \omega_i)$, es decir la probabilidad de que aparezca el dato $\mathbf{x}$ cuando se cumple la clase ω_i. Dadas todas las probabilidades, se elige la mayor como la clase a la que el patrón pertenece:

$$\hat{\omega}_{ML}(x) = \arg \max_{\omega_i \epsilon \Omega} P(\mathbf{x} \mid \omega_i)$$

- Máximo a posteriori (*MAP*): en la probabilidad $P(\mathbf{x} \mid \omega_i)$es ponderada con la probabilidad *a priori* de cada clase, $P(\omega_i)$, dando mayor peso a las clases más probables:

$$\hat{\omega}_{ML}(x) = \arg \max_{\omega_i \epsilon \Omega} P(\mathbf{x} \mid \omega_i) P(\omega_i)$$

La siguiente sección describe la metodología seguida para la implementación del clasificador Bayes en *Matlab*, utilizando la *PDF* gaussiana y la regla *MAP*.

3. Metodología

La actividad fue dividida en tres módulos, dos de ellos principales, enfocados en las tareas de entrenamiento y clasificación, y uno complementario dedicado a la visualización de los resultados de la clasificación.

El flujo de tareas (indicado en la Figura 1) consistió en las siguientes etapas:

1. Selección del dataset con los datos a procesar. El dataset incluye las clases *reales* de cada instancia.

2. División de los datos en dos grupos, uno de entrenamiento (70 %) y uno de prueba (30 %).

3. Ejecución del proceso de entrenamiento, otorgando como entrada la porción del dataset destinada para ello.

4. Ejecución del proceso de prueba, empleando la porción del dataset destinada para ello. Dentro de este proceso se puede identificar la tasa de errores generada en la clasificación, gracias a la etiqueta *real* que existe en el dataset.

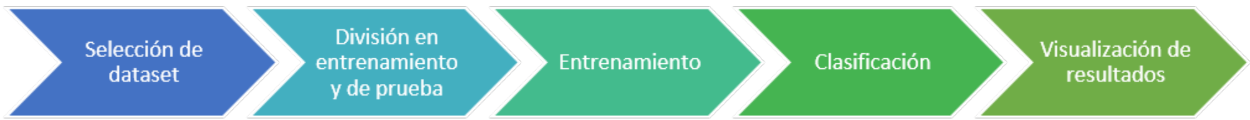


Figura 1: Flujo de actividades

5. Visualización de los resultados y la evaluación del clasificador utilizando la tasa de errores.

Las siguientes subsecciones describen los procesos implementados para el entrenamiento, clasificación y visualización de datos.

3.1. Entrenamiento

La etapa de entrenamiento analiza el conjunto de datos y obtiene un modelo o representación de los mismos para descubrir la probabilidad que tendrá un nuevo patrón de pertenecer a las clases del dataset. Recordando que el tipo de clasificación que se realiza es supervisada, los datos de entrada para la etapa de entrenamiento deben incluir la etiqueta, o *clase*, a la que pertenece cada instancia o patrón.

El Algoritmo 1 describe los pasos seguidos para la generación de los parámetros que describen al modelo (dataset) y que son requeridos en la etapa de clasificación. Los valores de estos parámetros describen a:

- μ_{ω_i}: Matriz de las medias de los atributos (dimensiones) para cada clase.
- Σ_{ω_i}: Vector de matrices (*hipermatriz*) de covarianza para los atributos (dimensiones) de cada clase.
- $P(\omega_i)$: Vector con las probabilidades de cada clase.

Este módulo únicamente recibe como argumento un conjunto de datos de entrenamiento con sus respectivas etiquetas

3.2. Clasificación

Una vez que el modelo que describe a los datos ha sido generado, la etapa de clasificación permite asignar una clase a un patrón anónimo que se desee identificar. Como se

Algoritmo 1 Algoritmo de entrenamiento

Entrada: *Xtr*, *Ytr*

Salida: *mu_wi*, *S_wi*, *P_wi*

```
para cada clase c en Ytr hacer
    Obtener instancias c_instances de c
    Actualizar mu_ui con las medias de cada dimensión en c_instances
    Actualizar S_wi con la matriz de covarianza para c
    Actualizar P_wi con la probabilidad de c
fin para
```

ha mencionado, los parámetros requeridos por la etapa de clasificación son aquellos generados por la etapa de entrenamiento, en conjunto con los datos de prueba que se desea clasificar.

El Algoritmo 2 describe los pasos seguidos para clasificar cada patrón y asignarle la etiqueta de clase correspondiente. Puede identificarse que en este proceso se evalúa la *PDF* gaussiana generada durante el entrenamiento y cómo se determina la asignación de la clase a través de la regla *MAP*.

El proceso finaliza obteniendo un vector con las etiquetas que el clasificador ha asignado, así como con un valor escalar que indica el porcentaje de error generado. Este porcentaje puede ser calculado gracias a que se cuenta con la etiqueta (clase) *real* que cada patrón debería observar.

3.3. Visualización de resultados

Aunque las etapas principales y necesarias para el clasificador son el entrenamiento y la clasificación, resulta de gran utilidad el contar con una estrategia para representar visualmente los datos y observar tendencias, outliers e información adicional que los datos numéricos no pueden entregar de forma directa.

Se han utilizado las facilidades aportadas por Matlab para representar el espacio de datos particionado en áreas según su clase, graficar los datos originales y para ubicar los puntos ubicados de forma errónea por el clasificador. El Algoritmo 3 muestra de forma breve los pasos seguidos para realizar la visualización de los datos. La idea general consiste en recoger el resultado de una ejecución del clasificador y generar un set de puntos sintéticos dentro del rango de valores de los datos clasificados. Estos puntos sintéticos son también clasificados para recoger las etiquetas asignadas por Bayes y con ellas *pintar* el fondo de la gráfica. Finalmente, se realiza la graficación de los puntos y los errores.

Algoritmo 2 Algoritmo de clasificación

Entrada: Xtt, Ytt, mu_wi, S_wi, P_wi

Salida: $Ypred$,err

para cada instancia x_i en Xtt **hacer**
    Calcular la probabilidad de que pertenezca a cada clase:
    para cada clase c_i en P_wi **hacer**
        Calcular el valor de la pdf gaussiana para esta clase, mediante:
        $pdf = \frac{1}{(2\pi)^{l/2}|\Sigma_i|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu_i)^T \Sigma_i^{-1} (\mathbf{x}-\mu_i)\right)$
        $Probabilidad(c\_i)$ = pdf $*P\_wi$(c_i)
    fin para
    Elegir la clase con probabilidad más alta en el vector $Probabilidad$
    Colocar la clase elegida en $Ypred$
fin para
Actualizar err con la discrepancia entre $Ypred$ y Ytr

Adicionalmente, se ha generado también una gráfica con el porcentaje de errores obtenido en cada iteración independiente, que por su simplicidad no es descrita en este documento.

Algoritmo 3 Algoritmo de visualización

Entrada: Xtt,Ytt,$Ypred$,mu_wi,S_wi,P_wi

Salida: Una gráfica con el espacio particionado

$rango_x$ = crear 100 puntos entre los valores min y max en X ($Xtt(1,:)$)
$rango_y$ = crear 100 puntos entre los valores min y max en Y ($Xtt(2,:)$)
Crear una matriz xy con los puntos sintéticos
Cambiar la forma de xy a vectores para ser clasificados por Bayes, utilizando
    $labels$ = classifyBayes(xy,mu_wi,S_wi,P_wi)
Cambiar la forma del vector de etiquetas $labels$ a matriz (reshape)
Escalar y mostrar la matriz como el fondo (regiones particionadas) de la gráfica (imagesc)
Graficar puntos clasificados
Graficar los errores (puntos mal clasificados) según la diferencia entre Ytt y $Ypred$

Dataset	Runtime	% Error min	% Error max	% Error media
clouds01	16.70 s	1.72	2.72	2.22
clouds02	11.72 s	0.00	0.53	0.24
halfkernel	1.91 s	1.33	11.33	4.82
overlapped	7.30 s	3.38	10.53	7.03
twospirals	1.94 s	39.67	29.33	35.78

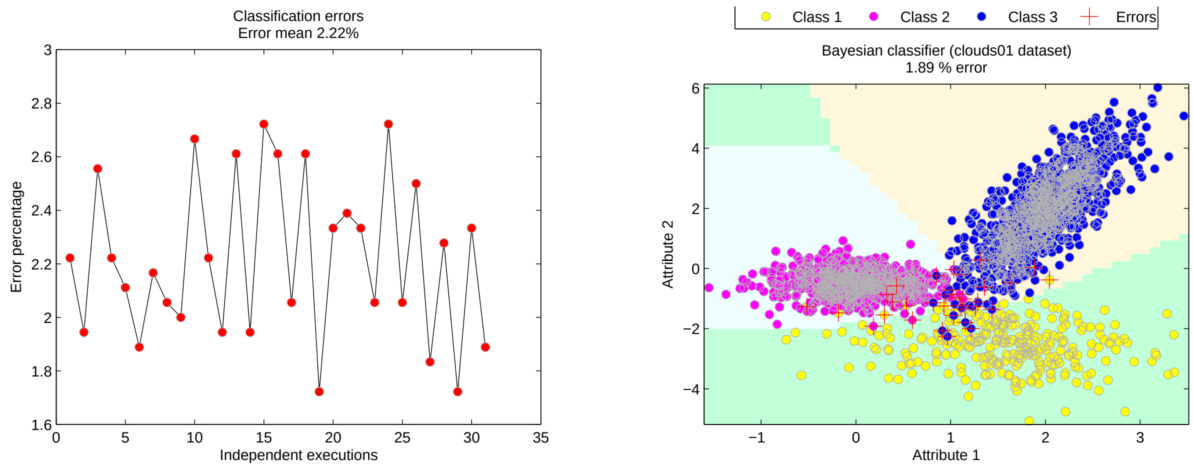
Cuadro 1: Resumen de los datos de errores obtenidos.

4. Resultados

Se ha ejecutado 31 veces el clasificador bayesiano para cada uno de los datasets, obteniendo porcentajes de error menores al 12 % a excepción del dataset de `twospirals`. En dicho dataset, al parecer el hecho de que los *envoltorios* de cada espiral se solapen hace que los valores en las matrices de medias y covarianzas sean también solapados, provocando una separación lineal del espacio de objetos. Esto puede ser notado en la Figura 6 en la que la línea que separa a las clases parte de los valores mínimos del atributo 1 y 2 de los objetos de la clase 1 hacia los valores máximos del atributo 1 y 2 de los objetos de la clase 2.

La Tabla 1 muestra el resumen de los datos de errores obtenidos durante la ejecución del clasificador para cada uno de los datasets. El tiempo reportado considera la carga del dataset y las etapas de clasificación y entrenamiento.

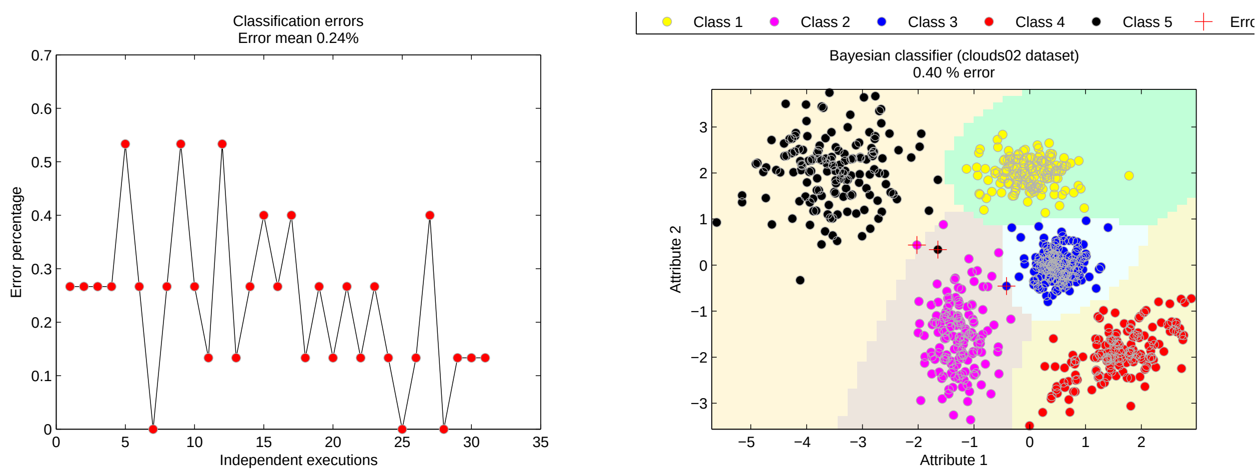
Las Figuras 2, 3, 4, 5 y 6 muestran las parejas de gráficas tanto de error como del espacio particionado, para cada uno de los datasets. La gráfica del espacio particionado corresponde a la última ejecución, donde pueden observarse también los puntos correctamente clasificados así como aquellos erróneamente etiquetados.



(a) Porcentajes de error obtenido

(b) Espacio de datos obtenido

Figura 2: Resultados de la clasificación para el dataset `clouds01`



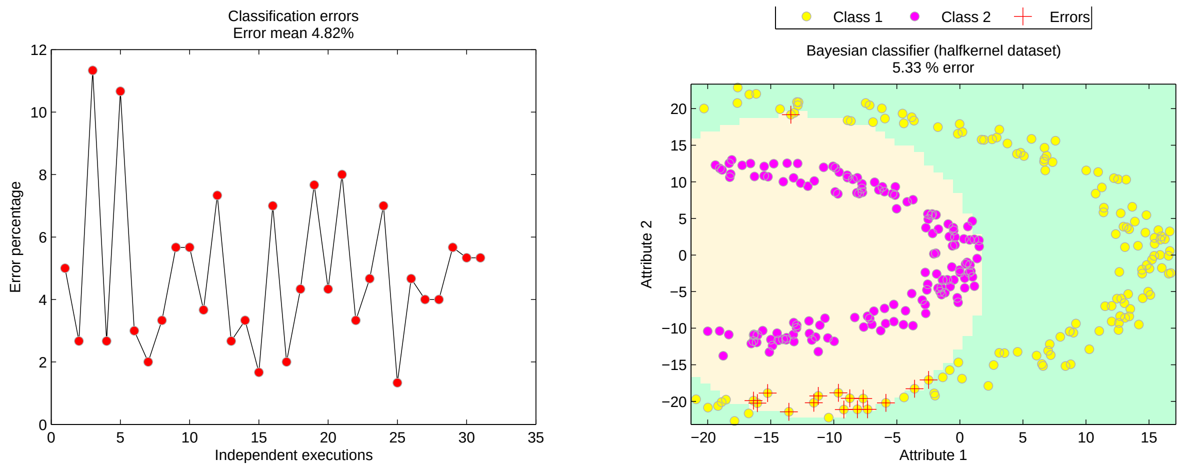
(a) Porcentajes de error obtenido

(b) Espacio de datos obtenido

Figura 3: Resultados de la clasificación para el dataset `clouds02`

5. Conclusiones

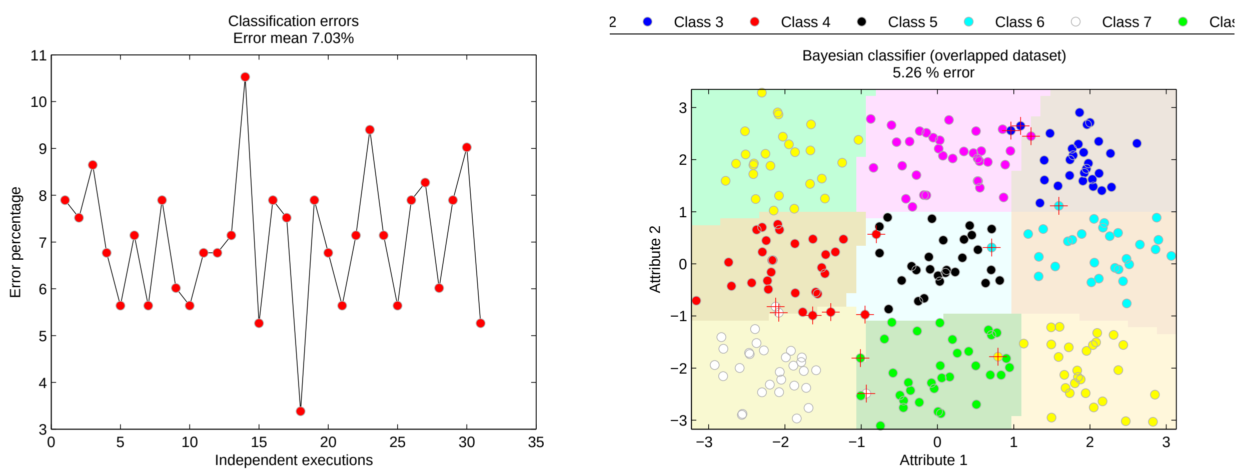
Los clasificadores con enfoque probabilístico intentan predecir la pertenencia de un patrón a una determinada clase basándose en probabilidades que son obtenidas a partir de eventos anteriores. Utilizando la idea básica de Thomas Bayes de la probabilidad condicional, el clasificador bayesiano intenta describir las características de los datos de un conjunto de entrenamiento a través de una función de densidad de probabilidad. Gracias



(a) Porcentajes de error obtenido

(b) Espacio de datos obtenido

Figura 4: Resultados de la clasificación para el dataset `halfkernel`



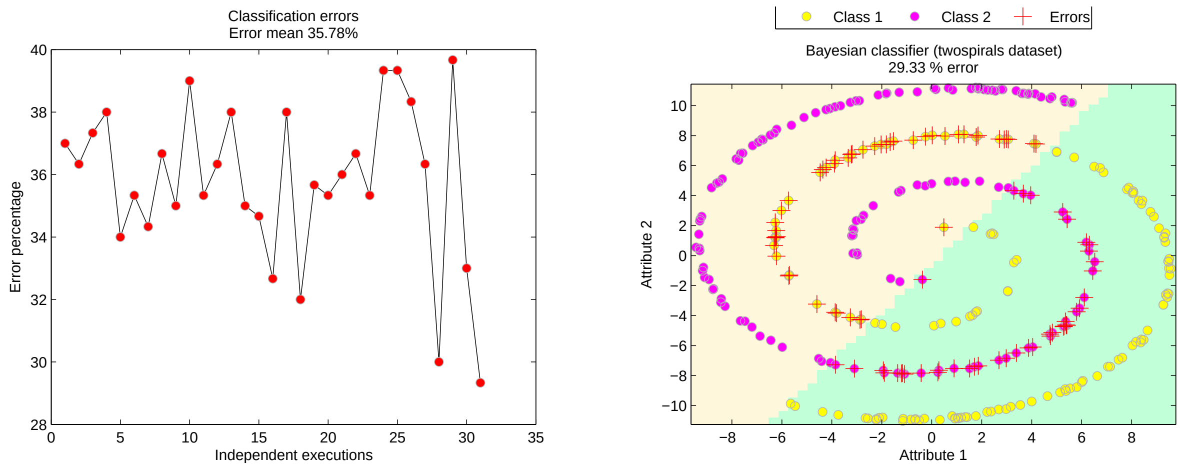
(a) Porcentajes de error obtenido

(b) Espacio de datos obtenido

Figura 5: Resultados de la clasificación para el dataset `overlapped`

a esta función, el clasificador puede obtener una *idea* sobre la distribución de los datos y hacer uso de ella para realizar la clasificación de patrones desconocidos.

El presente documento ha mostrado la implementación y prueba de un clasificador bayesiano que utiliza la *PDF* gaussiana para describir a los datos (mediante la media y covarianza) y la regla *MAP* para realizar la clasificación. Se ha encontrado que el clasificador bayesiano ofrece resultados con porcentajes de error menores al 10 % en datasets que no tengan regiones complejas, tal como ocurre con el dataset de *twospirals*. Dicho dataset



(a) Porcentajes de error obtenido

(b) Espacio de datos obtenido

Figura 6: Resultados de la clasificación para el dataset `overlapped`

contiene dos clases que definen áreas con demasiado traslape, lo cual se sospecha que ocasiona que los valores de medias y covarianzas de las clases sean también traslapados.