Clasificación Lineal

Aprenentatge Automàtic

APA/GEI/FIB/UPC - 2025/2026 1Q

/ Javier Béjar

Introducción

- En clasificación, la respuesta de un ejemplo es un valor discreto
- El objetivo es identificar, de un conjunto finito de clases, la clase $\mathcal{C}_k$ a la que pertenecen los ejemplos
- Los ejemplos deben dividirse en áreas separadas (un ejemplo solo pertenecerá a una clase) denominadas regiones de decisión
- Los límites entre las clases se denominan fronteras de decisión (o superficies de decisión)
- Para la clasificación lineal, las fronteras son funciones lineales de los ejemplos definidas como hiperplanos de $D - 1$ dimensiones
- Las clases que se pueden separar perfectamente de esta manera se denominan Linealmente separables

Modelos probabilísticos

- Los problemas binarios representan la salida como valores $y \in \{0, 1\}$
- Los problemas no binarios representan la salida como un vector utilizando la codificación 1-of-K (también conocida como codificación 1-hot)

$$y = (0, 0, 1, 0)$$

- Las salidas se pueden interpretar como probabilidades (probabilidad de ser de la clase $\mathcal{C}_k$)

Modelos no probabilísticos

- Se pueden usar diferentes codificaciones dependiendo del modelo
- Para problemas binarios $\{0, 1\}$ puede ser útil, pero también $\{-1, 1\}$

Funciones discriminativas

- Se aprende una función que calcula directamente un valor para un ejemplo que lo asigna a una clase
- Desde una perspectiva práctica, son los modelos más simples porque requieren menos suposiciones sobre los datos, pero para algunos problemas no es realista suponer que la salida es completamente precisa

Modelos discriminativos probabilísticos

- Se aprende la probabilidad condicional de las clases $p(\mathcal{C}_k|x)$ como funciones de probabilidad parametrizadas optimizando sus parámetros
- Son un compromiso, con más escalabilidad, capaces de modelar la incertidumbre y más tolerantes a suposiciones erróneas sobre las distribuciones de probabilidad detrás de los datos

Modelos generativos probabilísticos

- Se aprenden las densidades de los datos condicionadas a las clases $p(x|\mathcal{C}_k)$ y las probabilidades a priori de las clases $p(\mathcal{C}_k)$
- Se aplica el teorema de Bayes para obtener las probabilidades de las clases dadas las observaciones

$$p(\mathcal{C}_k|x) \propto p(x|\mathcal{C}_k)p(\mathcal{C}_k)$$

- Son los más precisos si las suposiciones sobre las distribuciones de probabilidad detrás del problema son correctas, pero pueden ser difíciles de estimar y su número de parámetros crece cuadráticamente con el número de atributos

Funciones discriminativas

- Un enfoque simple para la clasificación es encontrar un conjunto de planos/superficies de decisión que definen la frontera entre las clases
- Para un problema binario, la decisión simplemente se toma como:

$$w^\top x + w_0 \geq 0 \Rightarrow \mathcal{C}_1$$
$$w^\top x + w_0 < 0 \Rightarrow \mathcal{C}_2$$

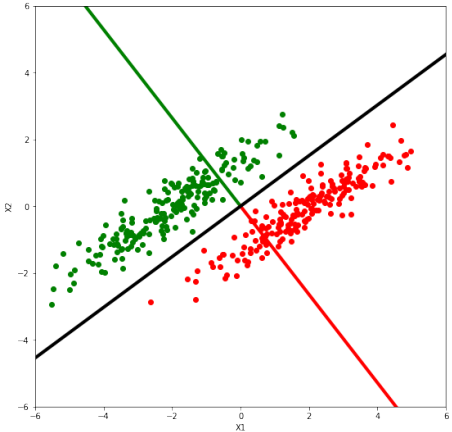
- Para múltiples clases podemos tener múltiples superficies de decisión para separar una clase del resto o que separen cada par de clases
- Hay múltiples formas de calcular esas superficies de decisión

- Discriminante lineal de Fisher: Proyectar los datos a menos dimensiones de tal manera que se maximice la separabilidad
- Perceptrón: Una función discriminante lineal que calcula una transformación no lineal de las entradas que define el modelo lineal generalizado:

$$f(x) = signo(w^\top \phi(x))$$

- Máquinas de soporte vectorial lineales: Función discriminante lineal que corresponde al separador óptimo entre clases

- El discriminante lineal de Fisher aborda el problema de clasificación como un problema de reducción de dimensionalidad
- Los datos se proyectan a vectores unidimensionales donde la separación entre clases se define como un umbral
- Los vectores se calculan para que la proyección maximice la separabilidad entre las clases



- Queremos obtener una proyección de los ejemplos a una dimensión como:

$$f(x) = w^\top x$$

- La proyección obviamente perderá mucha información y no hay garantía de que las clases sean separables linealmente

- Calculando el w adecuado podemos seleccionar una proyección que maximiza la separación entre clases

- Consideraremos dos clases con N_1 y N_2 ejemplos respectivamente
- Los centroides de las clases corresponden a:

$$m_1 = \frac{1}{N_1} \sum_{n \ en \mathcal{C}_1} x_n \quad m_2 = \frac{1}{N_2} \sum_{n \in \mathcal{C}_2} x_n$$

- El objetivo es obtener una proyección que maximice la distancia entre las medias proyectadas de las clases
- Añadimos también la restricción de que la varianza de la proyección de los datos sea minimizada para que el solapamiento entre las clases se reduzca
- Para obtener una única solución imponemos la condición $||w||_2^2 = 1$

- Definimos la covarianza entre clases (*between class*) como la matriz

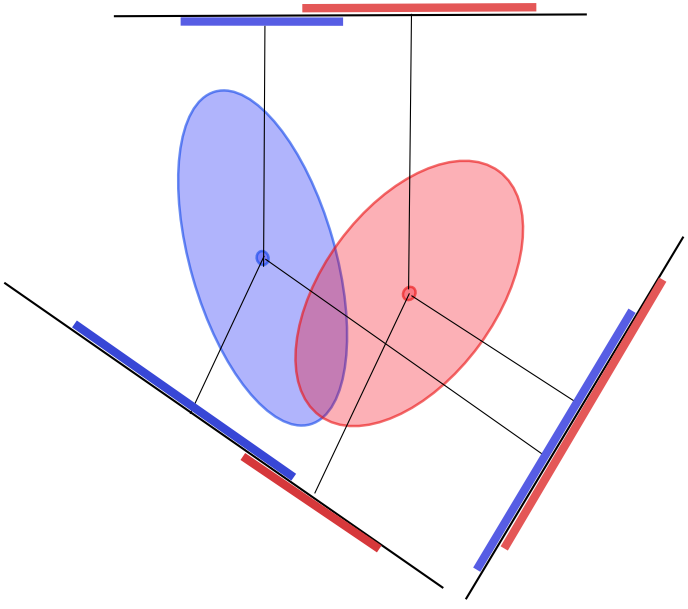
$$S_B = (m_2 - m_1)(m_2 - m_1)^\top$$

- y la covarianza interna de clase (*within class*) como la matriz

$$S_w = \sum_{n \in \mathcal{C}_1} (x_n - m_1)(x_n - m_1)^\top + \sum_{n \in \mathcal{C}_2} (x_n - m_2)(x_n - m_2)^\top$$

- Tenemos que encontrar los pesos que maximicen la relación entre estas dos matrices:

$$\max_w \frac{w^\top S_B w}{w^\top S_W w}$$



- La solución de este problema se puede obtener mediante la descomposición en valores propios de la matriz $S_W^{-1} S_B$ y tomando el vector propio con el valor propio más grande
- Una vez que tenemos la proyección podemos obtener el umbral que permite la clasificación con los datos proyectados (por ejemplo suponiendo que son gaussianos)
- El caso multiclase es el mismo pero con las matrices de covarianzas calculadas para todas las clases y tomando los vectores propios $K - 1$ con los valores propios más grandes

Modelos generativos probabilísticos

- La ventaja de un modelo generativo es que obtenemos una *estimación* de las funciones de distribución de probabilidad que generan los datos de las clases
  - Esto se puede usar para inferir otra información de las distribuciones y, en particular, permite generar nuevos datos usando muestreo
- Un modelo generativo aprende dos distribuciones:
  - Las densidades condicionales de la clase $p(x|\mathcal{C}_k)$
  - Las probabilidades a priori de la clase $p(\mathcal{C}_k)$
- Las probabilidades de las clases dado un ejemplo $p(\mathcal{C}_k|x)$ se pueden obtener aplicando el teorema de Bayes

$$p(\mathcal{C}_k|x) \propto p(x|\mathcal{C}_k)p(\mathcal{C}_k) = p(\mathcal{C}_k, x)$$

- A diferencia de la regresión, los modelos de clasificación deben mapear el resultado de las funciones lineales a una probabilidad $y = f(x) \in (0, 1)$
- Podemos generalizar el modelo lineal aplicando una transformación no lineal sobre la función que realiza el mapeo

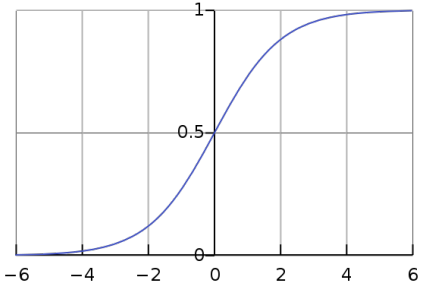
$$f(x) = g(w^\top x + w_0)$$

- g se denomina función activación (la inversa g^{-1} se denomina función de enlace (*link*) en estadística), debe ser continua, derivable y monótonamente creciente
- Las superficies de decisión corresponden a funciones lineales
- Estos modelos se denominan **Modelos Lineales Generalizados**

- Vamos a definir el problema de clasificación utilizando un modelo generativo como un modelo lineal generalizado donde la función de decisión se calculará como:

$$f(x) = g(w^\top x)$$

- La función sigmoide (logística) será la función g que transformará la combinación lineal al rango $[0, 1]$, por lo que de ella obtenemos probabilidades



- La función sigmoide tiene buenas propiedades que facilitan las cosas

$$\sigma(a) = \frac{1}{1 - e^{-a}} = \frac{e^a}{e^a + 1}$$

- Tiene la siguiente propiedad de simetría

$$\sigma(-a) = 1 - \sigma(a)$$

- Tiene como función inversa (función logit)

$$a = \log\left(\frac{\sigma}{1 - \sigma}\right)$$

- Su derivada es

$$\sigma'(a) = \sigma(a)(1 - \sigma(a))$$

- Para el caso de dos clases podemos definir la probabilidad a posteriori para la clase $\mathcal{C}_1$ como:

$$p(\mathcal{C}_1|x) = \frac{p(x|\mathcal{C}_1)p(\mathcal{C}_1)}{p(x|\mathcal{C}_1)p(\mathcal{C}_1) + p(x|\mathcal{C}_2)p(\mathcal{C}_2)}$$

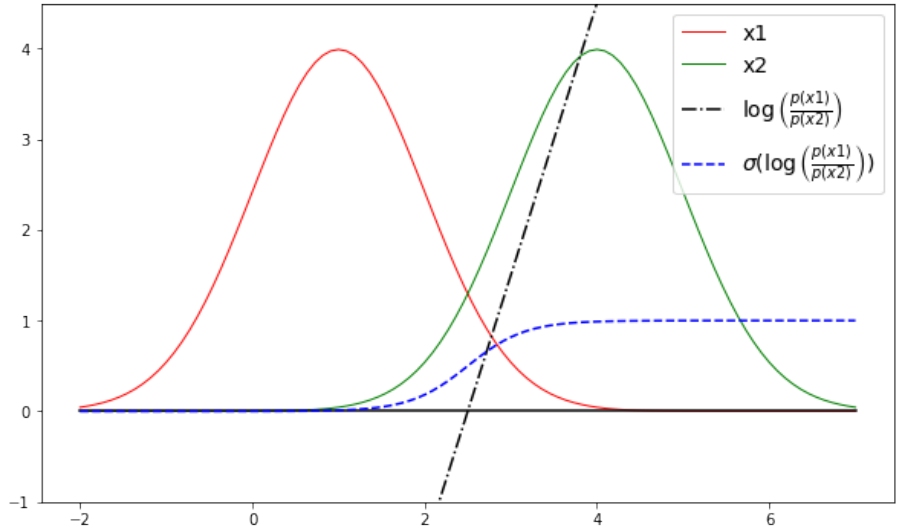
- Si definimos a como

$$a = \log \frac{p(x|\mathcal{C}_1)p(\mathcal{C}_1)}{p(x|\mathcal{C}_2)p(\mathcal{C}_2)}$$

- Obtenemos

$$p(\mathcal{C}_1|x) = \frac{1}{1 + \exp(-a)} = \sigma(a)$$

- La función a se llama log odds, el cociente entre las probabilidades son los odds


x1
x2
$\log\left(\frac{p(x1)}{p(x2)}\right)$
$\sigma(\log\left(\frac{p(x1)}{p(x2)}\right))$
4
3
2
1
0
−1
−2
0
2
4
6

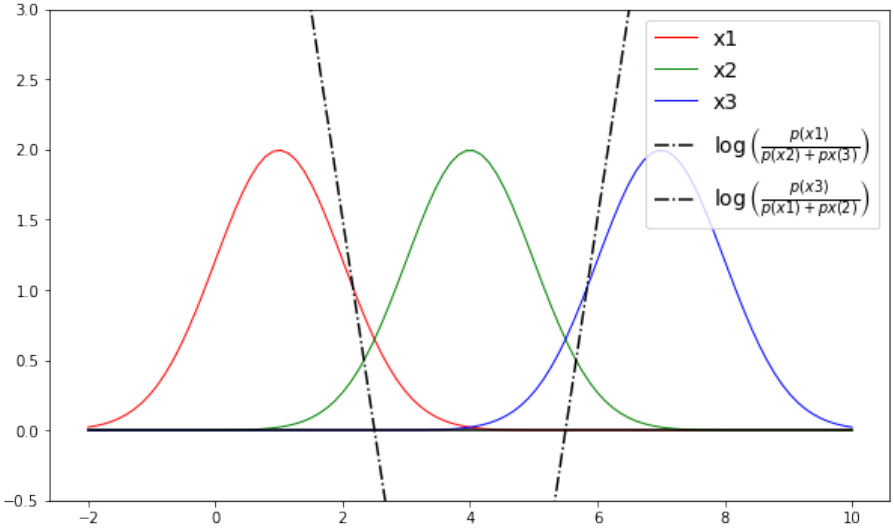
- Para el caso de más de dos clases tenemos

$$p(\mathcal{C}_k|x) = \frac{p(x|\mathcal{C}_k)p(\mathcal{C}_k)}{\sum_j p(x|\mathcal{C}_j)p(\mathcal{C}_j)} = \frac{\exp(a_k)}{\sum_j \exp(a_j)}$$

  Donde $a_k = \log p(x|\mathcal{C}_k)p(\mathcal{C}_k)$

- Esta es la exponencial normalizada también conocida como función softmax

- Esta función hace que la probabilidad de una clase k sea cercana a 1 cuando el valor de a_k es mucho mayor que el resto


x1
x2
x3
$\log\left(\frac{p(x1)}{p(x2)+px(3)}\right)$
$\log\left(\frac{p(x3)}{p(x1)+px(2)}\right)$
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−2
0
2
4
6
8
10

Modelos generativos gaussianos

- Haciendo diferentes suposiciones sobre las distribuciones de probabilidad condicional de clase podemos obtener diferentes clasificadores
- Podemos suponer que las distribuciones $p(x|\mathcal{C}_k)$ son gaussianas y las distribuciones a priori $p(\mathcal{C}_k) = \pi_k$ (con $\sum_k \pi_k = 1$)
- Para una clase k

$$p(x_n, \mathcal{C}_k) = \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)$$

- La función de verosimilitud se define como:

$$p(y|\pi, \mu_1, \ldots, \mu_k, \Sigma_1 \ldots, \Sigma_k) = \prod_{n=1}^{N} \prod_{k=1}^{K} [\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)]^{y_{nk}}$$

- Donde y es un vector 1-de-K que tiene un valor 1 en la posición k si x_n es de la clase $\mathcal{C}_k$ y 0 para el resto

- Para el caso particular de matrices de covarianza comunes $\Sigma = \Sigma_1 = \cdots = \Sigma_k$ obtenemos un conjunto de funciones discriminantes lineales[1]:

$$a_k(x) = x^\top \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^\top \Sigma^{-1} \mu_k + \log \pi_k$$

- Podemos ver que tiene la forma $w^\top x + w_0$

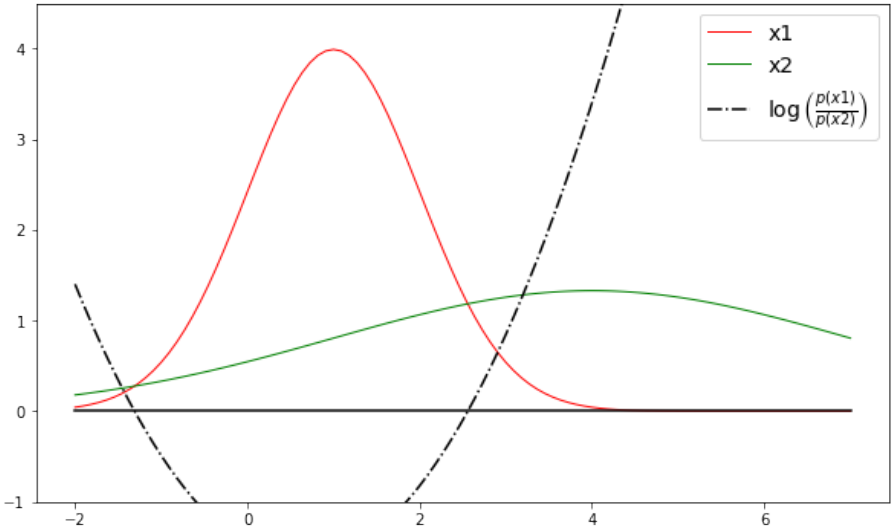
$$w = \Sigma^{-1} \mu_k$$
$$w_0 = -\frac{1}{2} \mu_k^\top \Sigma^{-1} \mu_k + \log \pi_k$$

[1] Ver (Hastie et al, pag 108) para la derivación

- Si asumimos una matriz de covarianza específica para cada clase, en su lugar tenemos una superficie de decisión cuadrática

$$a_k(x) = -\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(x-\mu_k)^\top \Sigma_k^{-1}(x-\mu_k) + \log \pi_k$$

- Esto significa que es un clasificador no lineal ya que realiza una transformación cuadrática de los ejemplos de entrada


x1
x2
$\log\left(\frac{p(x1)}{p(x2)}\right)$
4
3
2
1
0
−1
−2
0
2
4
6

Podemos aprender el clasificador usando los parámetros que corresponden a los estimadores de máxima verosimilitud de la distribución gausiana:

$$\hat{\pi}_k = \frac{N_k}{N}$$

$$\hat{\mu}_k = \sum_{n \in \mathcal{C}_k} \frac{x_n}{N_k}$$

$$\hat{\Sigma}_k^{QDA} = \sum_{n \in \mathcal{C}_k} \frac{(x_n - \mu_k)(x_n - \mu_k)^\top}{N_k - 1}$$

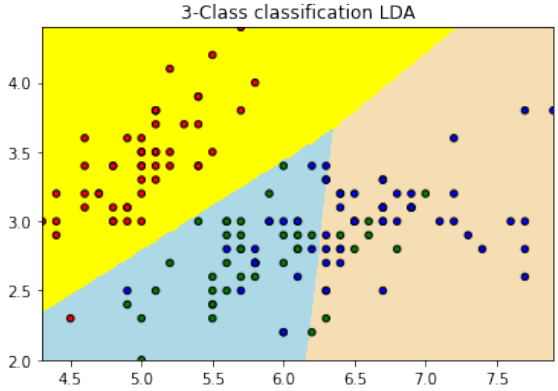
$$\hat{\Sigma}^{LDA} = \sum_{k=1}^{K} \sum_{n \in \mathcal{C}_k} \frac{(x_n - \mu_k)(x_n - \mu_k)^\top}{N - K}$$

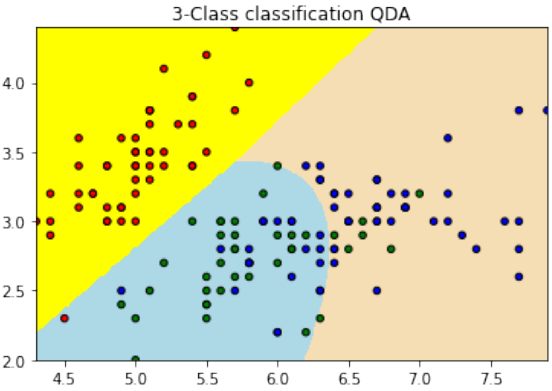
Básicamente la proporción de las clases, las medias muestrales y las covarianzas de los datos de cada clase para QDA o la covarianza común para LDA

- Como sucede en la regresión, el MLE para LDA/QDA tiende a sobreajustar los datos cuando la muestra es pequeña
- También podría haber problemas numéricos con matrices de covarianza que pueden ser singulares si algunas clases tienen pocos ejemplos.
- El análisis discriminante regularizado calcula las matrices de covarianza para las clases como:

$$\hat{\Sigma}_k(\alpha) = \alpha\hat{\Sigma}_k + (1-\alpha)\Sigma$$

- Con $\alpha \in [0, 1]$, eso permite un continuo entre QDA ($\alpha = 1$) y LDA ($\alpha = 0$)


3-Class classification LDA
4.0
3.5
3.0
2.5
2.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5


3-Class classification QDA
4.0
3.5
3.0
2.5
2.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5

Naïve Bayes

- El número de parámetros a estimar dada una distribución depende de las suposiciones sobre las relaciones entre las variables
- Para LDA y QDA usando distribuciones gaussianas, el número de parámetros crece cuadráticamente (D^2) con el número de atributos si asumimos matrices de covarianza completas

- Asumiendo atributos binarios representados como distribuciones de Bernoulli (valores binarios), el número de posibles resultados crece exponencialmente (2^D)
- Considerando distribuciones Multinomiales (n valores) tenemos $\prod_{d=1}^{D} V_d$ donde V_d es el número de valores del atributo d
- Si restringimos nuestras distribuciones de probabilidad a aquellas donde todos los atributos son independientes, el costo computacional de estimar probabilidades (y almacenarlas) se reduce considerablemente

- **Naïve Bayes** es un clasificador probabilístico generativo que asume que los atributos de las probabilidades condicionales de la clase son independientes

$$p(x|\mathcal{C}_k) = p(x|\Theta_k) = \prod_{d=1}^{D} p(x_d, \Theta_{kd})$$

- En general, esto no es cierto, pero puede ser una buena aproximación para muchos casos.
- Esto transforma la log-verosimilitud en

$$\log p(x|\mathcal{C}_k)p(C_k) = \sum_{d=1}^{D} \log p(x_d, \Theta_{kd}) + \log p(C_k)$$

- Aprender el clasificador Naïve Bayes se simplifica a:
  1. Calcular y almacenar las estimaciones de $p(C_k)$ como la frecuencia de las clases en la muestra
  2. Calcular por separado y almacenar los parámetros de la distribución elegida para cada clase y atributo de la muestra $p(x_d, \Theta_{kd})$
- Hacer predicciones implica calcular la log-verosimilitud para cada clase como:

$$a_k(x) = \sum_{d=1}^{D} \log p(x_d, \Theta_{kd}) + \ log(p(C_k))$$

- Como de costumbre, podemos usar la función sigmoide o la softmax para transformarla en probabilidades

- Los atributos binarios se pueden representar como distribuciones de Bernoulli, que tienen como pdf

$$p(x|\mu) = \mu^x(1-\mu)^{1-x}$$

con μ como la probabilidad del evento (x es 0 o 1)

- Asumiendo independencia, tenemos para una clase dada:

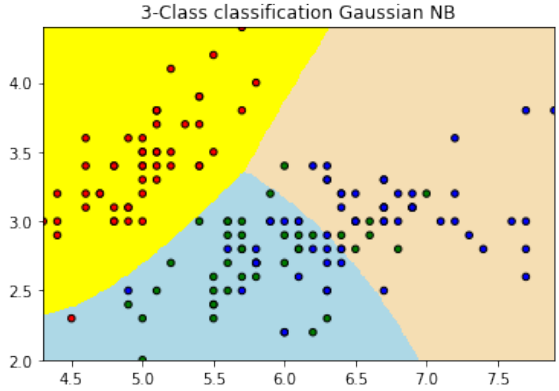
$$p(x|\mathcal{C}_k) = \prod_{d=1} \mu_{kd}^{x_d}(1-\mu_{kd})^{1-x_d}$$

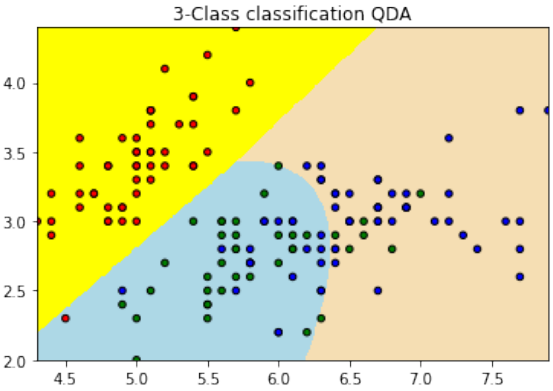
- Calculando la log-verosimilitud

$$a_k(x) = \sum_{d=1}^{D}[x_d \log \mu_{kd} + (1-x_d)\log(1-\mu_{kd})] + \log p(\mathcal{C}_k)$$

que es una función lineal de x

- Para atributos discretos (Bernoulli, Multinomial) podemos estimar la probabilidad de sus valores como su frecuencia muestral de la clase
- Para atributos continuos, asumiendo gausianidad, la estimación es como en LDA/QDA, pero las matrices de covarianza son diagonales (covarianza compartida = modelo lineal, covarianza por clase = modelo cuadrático)
- Si tenemos un problema con atributos mixtos podemos aprovechar que consideramos los atributos independientes, la probabilidad conjunta es el producto de probabilidades independientes


3-Class classification Gaussian NB
4.0
3.5
3.0
2.5
2.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5


3-Class classification QDA
4.0
3.5
3.0
2.5
2.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5

Este notebook muestra ejemplos de clasificación LDA, QDA y Naive bayes y cómo se pueden usar los modelos para generar nuevos datos.

También podéis ver un video que explica el contenido del notebook

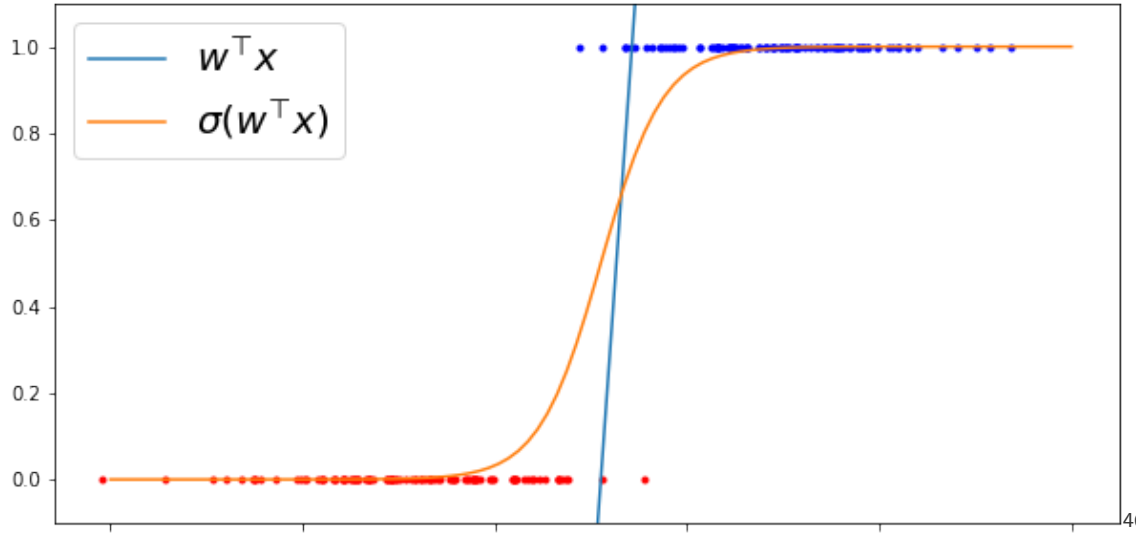
Modelos Probabilisticos Discriminativos

- En lugar de adoptar un enfoque complejo, modelando las distribuciones de probabilidad condicional y a priori de clase, podemos modelar directamente la distribución condicional $p(\mathcal{C}_k|x)$
- Esto tiene la ventaja de que el número de parámetros a estimar suele ser menor
- También se pueden obtener mejores resultados si la elección de la distribución de los modelos generativos está alejada de las distribuciones reales

- Como hemos visto, podemos definir la probabilidad a posteriori de una clase como:

$$p(\mathcal{C}_k|x) = f(x) = \sigma(w^\top x)$$

- Esto se conoce como regresión logística en estadística
- Los modelos generativos que asumen distribuciones gaussianas para las densidades condicionales de clase, necesitan un número cuadrático de parámetros (μ, Σ)
- Podemos estimar directamente la función lineal reduciendo la estimación a un número lineal de parámetros
- Para hacer eso, podemos usar la máxima verosimilitud para encontrar los estimadores para los parámetros


$w^\top x$
$\sigma(w^\top x)$
1.0
0.8
0.6
0.4
0.2
0.0
−2
0
2
4
6
8

- Dado un conjunto de datos $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ y sus etiquetas $\mathcal{Y} = \{y_1, y_2, \ldots, y_n\}$ asumiendo el caso de clasificación binaria $y_n \in \{0, 1\}$ con $\hat{y}_n = f(x_n, w)$, modelando la probabilidad de la respuesta y_n como una distribución de Bernoulli, tenemos como función de probabilidad

$$\mathcal{L}(w) = \prod_{n=1}^{N} \hat{y}_n^{y_n} (1 - \hat{y}_n)^{1-y_n}$$

- Podemos calcular la verosimilitud logarítmica negativa como

$$-\log \mathcal{L}(w) = E(w) = -\sum_{n=1}^{N} y_n \log \hat{y}_n + (1 - y_n) \log(1 - \hat{y}_n)$$

- Esto se conoce como error/pérdida de entropía cruzada

- Dado que $\hat{y}_n = f(x,w) = \sigma(w^\top x_n)$ podemos calcular la derivada de la log-verosimilitud negativa con respecto a los parámetros obteniendo[2]:

$$\nabla E(w) = \frac{\partial E(w)}{\partial w} = \sum_{n=1}^{N}(y_n - \hat{y}_n)x_n = \sum_{n=1}^{N}(y_n - \sigma(w^\top x_n))x_n$$

- Desafortunadamente, esto no da como resultado una solución analítica, por lo que debe resolverse usando optimización numérica
- Podemos usar un algoritmo de descenso de gradiente para obtener un óptimo

[2] recordad que $\sigma' = \sigma(1-\sigma)$

- Un algoritmo de este tipo para encontrar el óptimo para la regresión logística es Iterative Reweighted Least Squares (IRLS)
- Este algoritmo se basa en el método de optimización iterativo Newton-Raphson
- Newton-Raphson comienza con una suposición inicial x_0 y usa el cociente entre la función y su derivada como actualización, hasta la convergencia

$$x_{n+1} = x_n - \frac{f(x)}{f'(x)}$$

- En este caso, dado que tenemos la derivada del logaritmo de la verosimilitud, el cociente es entre su primera y su segunda derivada

$$w_{n+1} = w_n - H^{-1}(E(w))\nabla E(w)$$

  Donde H es la matriz hessiana, la segunda derivada de $E(w)$

- Si la función se comporta bien, la convergencia suele ser rápida

- Se pueden usar otros algoritmos más complejos para la optimización (L-BFGS, Gradiente Conjugado...)

- Dado que estamos usando los estimadores MLE podemos tener problemas de sobreajuste, especialmente cuando las clases son separables linealmente (soluciones infinitas)

- La regularización se usa para resolver este problema
  - Regularizaciones L_2, L_1 o una combinación

  - Criterios de información como el criterio de información de Akaike (penalizado por el número de parámetros) o el criterio de información bayesiano (penalizado por el número de parámetros multiplicado por $\log N$)

- El problema de optimización es similar cuando tenemos K clases
- Usamos distribuciones multinomiales para modelar las clases y la salida usa una representación de 1-de-K $y_n = \{0, \ldots, 1, \ldots, 0\}$
- Tenemos un separador para cada clase con un conjunto de parámetros w_k para cada uno, y calculamos las probabilidades usando la función softmax
- La probabilidad se define como:

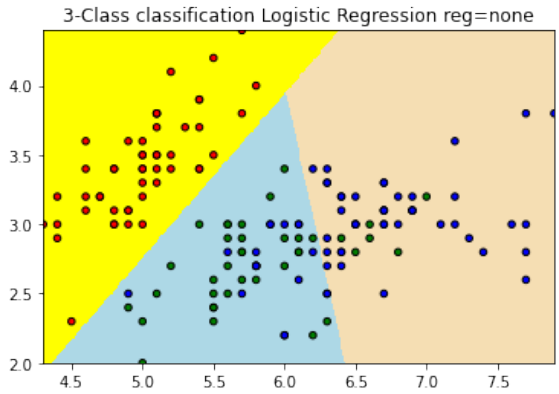
$$p(\mathcal{Y}|w_1, \ldots, w_K) = \prod_{n=1}^{N} \prod_{k=1}^{K} p(\mathcal{C}_k|x_n)^{y_{nk}} = \prod_{n=1}^{N} \prod_{k=1}^{K} \hat{y}_{nk}^{y_{nk}}$$

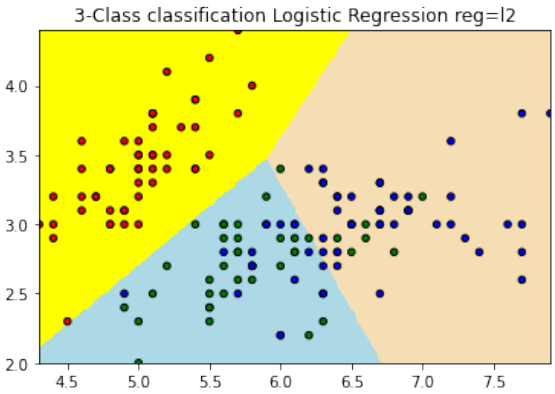
- La log-verosimilitud resultante es:

$$E(w_1, \ldots, w_n) = -\sum_{n=1}^{N}\sum_{k=1}^{K} y_{nk} \log \hat{y}_{nk}$$

- Esto se conoce como error/pérdida de entropía cruzada multiclase.

- Tomando derivadas, llegamos a un problema similar que en el caso de dos clases que tampoco tiene solución analítica y tiene que ser resuelto usando optimización iterativa


3-Class classification Logistic Regression reg=none
4.0
3.5
3.0
2.5
2.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5


3-Class classification Logistic Regression reg=l2
4.0
3.5
3.0
2.5
2.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5

Interpretabilidad/Explicabilidad

- Siempre que los modelos no incluyan interacciones de características o transformaciones no lineales, los pesos se pueden usar para la interpretación del modelo (esto descarta QDA, ya que usa las características al cuadrado)
- Mayores pesos para una característica significan mayor importancia
- El efecto de los pesos en la decisión ahora es multiplicativo

$$p(\mathcal{C}=1) = \sigma(w^\top x) = \frac{exp(w^\top x)}{1+\exp(w^\top x)} = \frac{\prod_i \exp(w_i x_i)}{1+\prod_i \exp(w_i x_i)}$$

- El modelo asume independencia entre variables, por lo que el efecto de cada característica depende de las probabilidades de sus valores
  - Para valores discretos, si la probabilidad de un valor de una característica en una clase es mayor, esto significa que es más importante para predecir la clase
- Las predicciones de los ejemplos se pueden explicar directamente a partir de las probabilidades asignadas al ejemplo por el modelo
  - Si la probabilidad de una característica es mayor para el ejemplo esto significa que tiene más influencia en la decisión

- Para la regresión logística y LDA, el efecto viene dado por los coeficientes de la combinación lineal
- La relevancia de las características depende directamente de la magnitud de los pesos (cuanto mayor sea el valor, más importante)
- La explicación del resultado para un ejemplo se hace directamente a partir de los coeficientes obtenidos, cuanto mayor sea el valor, mayor será el efecto sobre la respuesta
- Para la clasificación binaria, la probabilidad corresponde a la proporción entre los dos posibles resultados (a contra b)
- Para el caso multiclase, calculamos las probabilidades usando la función softmax que mide las proporciones de una clase contra el resto

Este notebook muestra un ejemplo de LDA y regresión logística con una explicación de los modelos