Estadística y computación para metagenómica

Victor Muñiz Sánchez

victor_m@cimat.mx

Centro de Investigación en Matemáticas.
Unidad Monterrey.

Junio 2023

Sobre ésta parte del curso...

Sobre el curso

Temario:

1. Introducción y conceptos generales
   - Machine learning (ML) supervisado y no supervisado
   - Teoría de decisión estadística
2. Métodos de aprendizaje supervisado
   1. Regresión logística
   2. Redes neuronales
   3. Hiperplanos separadores óptimos y Máquinas de Soporte Vectorial
   4. Selección de modelos y regularización
   5. Modelos aditivos y métodos relacionados
      - Arboles de decisión
      - Boosting
      - Random forest

Introducción y conceptos generales

Introducción

- En esta parte del curso mostraremos algunos métodos y conceptos básicos de aprendizaje máquina (ML: machine learning) y reconocimiento estadístico de patrones para el análisis de datos multivariados en general, y con aplicaciones en datos metagenómicos en particular.
- Los pre-requiitos: conocimientos básicos de modelos estadísticos (inferencia y regresión), álgebra lineal, cálculo de varias variables, conocimientos de programación.

Introducción

Software:

Introducción

Software:

Python y la infraestructura para cómputo científico y ciencia de datos

Introducción

Instalación local.

- La opción más óptima: instalar python `https://www.python.org/` y usar el editor de tu preferencia (vi, emacs, pycharm, spyder, jupyter-notebook, etc).
- La opción más rápida (recomendada para iniciar), instalar la suite Anaconda: `https://www.anaconda.com/`
- En cualquier caso, recomiendo ampliamente crear **virtual environments** para el curso y/o proyectos específicos que requieran a su vez, librerías específicas.

Instalar librerías: En Anaconda, se puede hacer directamente en el framework. También puedes hacerlo en consola con los comandos `pip` y `conda` (si tienes Anaconda).

Introducción

Ejecución de código

- The Python interpreter
- The IPython interpreter
- Self-contained Python scripts
- Jupyter notebook

Introducción

Jupyter + Google Colab

- Colaboratory permite escribir y ejecutar código de Python en un navegador
- Sin configuración requerida
- Acceso gratuito a GPU y TPU
- Facilidad para compartir

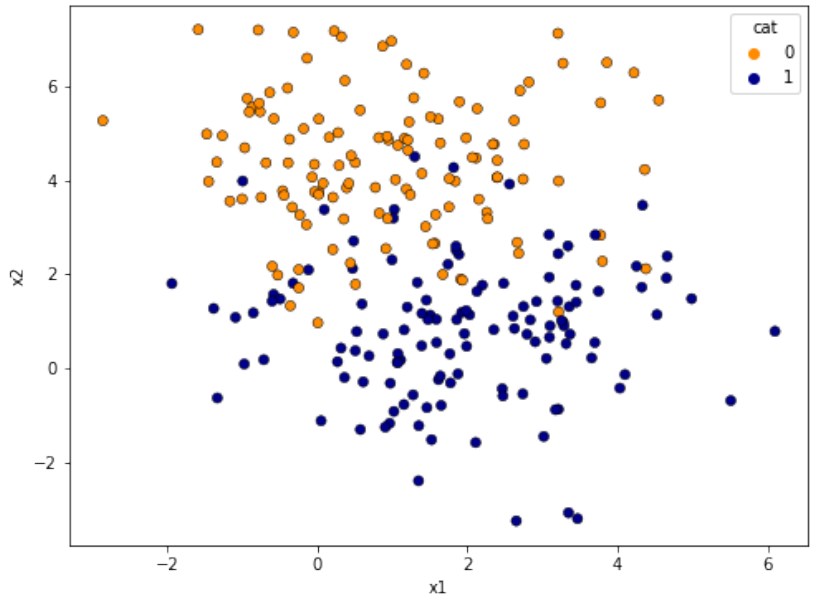
Introducción

`1-intro.ipynb`

Métodos de aprendizaje supervisado

Introducción

Considera un esquema clásico de clasificación.



Donde tenemos un conjunto de datos de entrenamiento

$$\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}; \quad \mathbf{x} \in \mathbb{R}^d, y \in \{0, 1\}$$

Introducción

Nuestro objetivo es obtener una función

$$f : \mathbb{R}^d \mapsto \{0, 1\}.$$

Introducción

Nuestro objetivo es obtener una función

$$f : \mathbb{R}^d \mapsto \{0, 1\}.$$

Bajo el esquema de aprendizaje máquina (ML), esperamos que ésta función se "aprenda" a partir de un conjunto de datos de entrenamiento, y generalmente depende de ciertos parámetros:

$$y = f(\mathbf{x}; \boldsymbol{\theta}).$$

Introducción

Nuestro objetivo es obtener una función

$$f : \mathbb{R}^d \mapsto \{0, 1\}.$$

Bajo el esquema de aprendizaje máquina (ML), esperamos que ésta función se "aprenda" a partir de un conjunto de datos de entrenamiento, y generalmente depende de ciertos parámetros:

$$y = f(\mathbf{x}; \boldsymbol{\theta}).$$

¿Cómo nos gustaría que fuera f?

Introducción

Nuestro objetivo es obtener una función

$$f : \mathbb{R}^d \mapsto \{0, 1\}.$$

Bajo el esquema de aprendizaje máquina (ML), esperamos que ésta función se "aprenda" a partir de un conjunto de datos de entrenamiento, y generalmente depende de ciertos parámetros:

$$y = f(\mathbf{x}; \boldsymbol{\theta}).$$

¿Cómo nos gustaría que fuera f?
Para ésta pregunta, veamos dos modelos para resolverlo.

Introducción

Mínimos cuadrados:

$$f(\mathbf{x}) = \beta_o + \boldsymbol{\beta}'\mathbf{x}$$

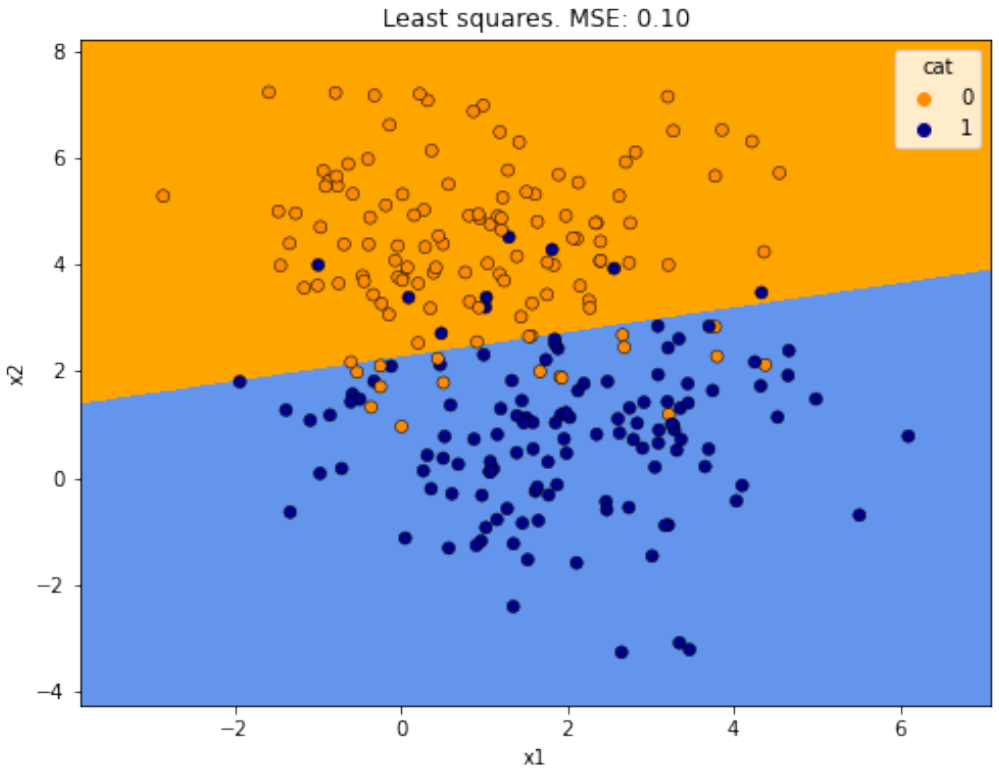
$$\hat{y} = \begin{cases} 0 & \text{si } f(\mathbf{x}) \leq 0.5 \\ 1 & \text{si } f(\mathbf{x}) > 0.5 \end{cases}$$

Introducción

Mínimos cuadrados:

$$f(\mathbf{x}) = \beta_o + \boldsymbol{\beta}'\mathbf{x}$$

$$\hat{y} = \begin{cases} 0 & \text{si } f(\mathbf{x}) \leq 0.5 \\ 1 & \text{si } f(\mathbf{x}) > 0.5 \end{cases}$$



Introducción

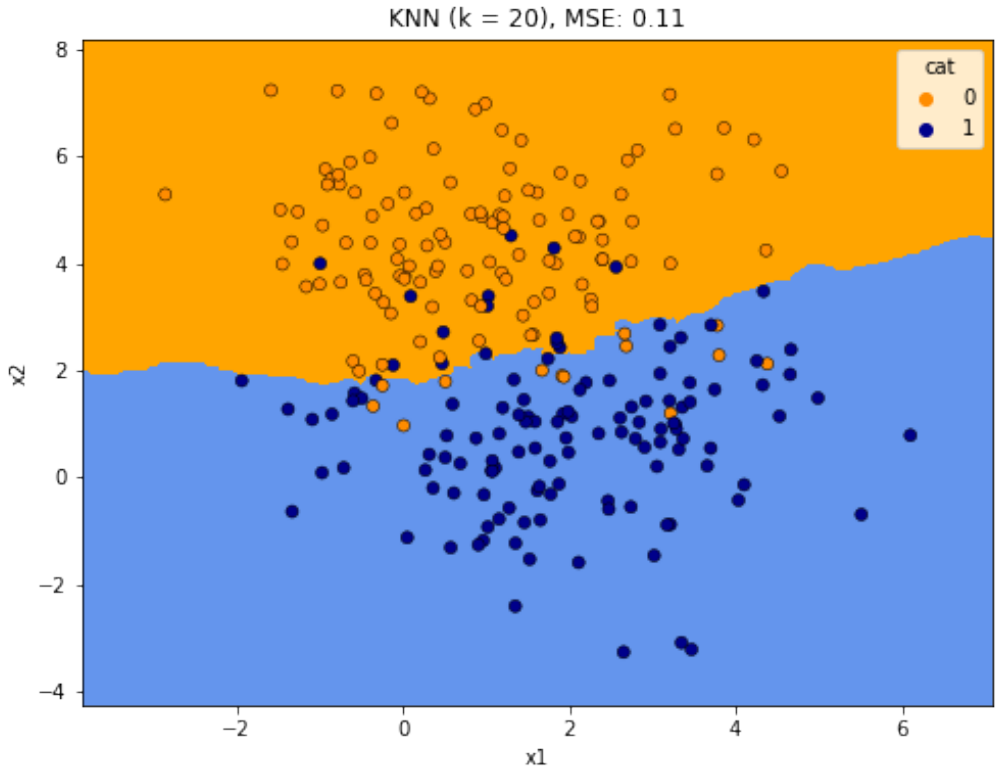
$k-$vecinos cercanos:

$$\hat{y} = f(\mathbf{x}) = \frac{1}{k} \sum_{\mathbf{x}_i \in \mathcal{N}_k(\mathbf{x})} y_i$$

Introducción

$k-$vecinos cercanos:

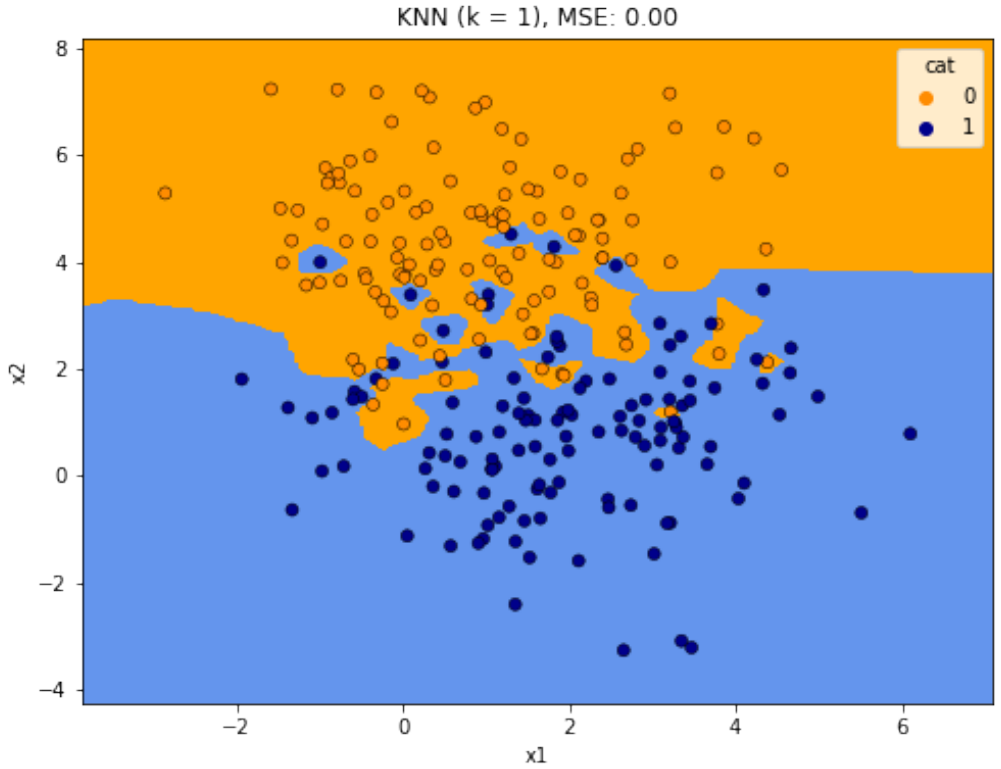
$$\hat{y} = f(\mathbf{x}) = \frac{1}{k} \sum_{\mathbf{x}_i \in \mathcal{N}_k(\mathbf{x})} y_i$$



Introducción

$k-$vecinos cercanos:

$$\hat{y} = f(\mathbf{x}) = \frac{1}{k} \sum_{\mathbf{x}_i \in \mathcal{N}_k(\mathbf{x})} y_i$$



Introducción

¿Cuál prefieres y porqué?

Introducción

Lo que vamos a ver en ésta última parte del curso, son diferentes propuestas para construir un clasificador

$$f : \mathbb{R}^d \mapsto \{1, 2, \ldots, K\},$$

incluyendo aspectos muy importantes, tales como:

- supuestos
- ajuste
- eficiencia
- complejidad
- generalización estadística
- regularización

entre otros...

Teoría de decisión estadística

Teoría de decisión estadística

Regresemos al problema de clasificación tratado anteriormente. Consideremos de momento, el caso de clasificación binaria:

$$\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}; \quad \mathbf{x} \in \mathbb{R}^d, y \in \{0, 1\}$$

Teoría de decisión estadística

Supongamos que podemos obtener **de alguna forma** éstas probabilidades:

$$P(y = 0), \qquad P(y = 1).$$

Si sólo contaramos con ésta información, lo más lógico (óptimo) es asignar

$$y_i = 1 \quad \text{si} \quad P(y = 1) > P(y = 2)$$

Teoría de decisión estadística

Sin embargo, nosotros contamos con información valiosa sobre la clase de las observaciones a través de las covariables $\mathbf{x}$.
Si una covariable $\mathbf{x}$ es discrimitiva, esperariamos que su distribución esté asociada a y, entonces nos interesa conocer

$$P(\mathbf{x}|y)$$

Entonces, con toda ésta información, podemos modelar

$$P(y, \mathbf{x}) = P(y|\mathbf{x})P(\mathbf{x}) = P(\mathbf{x}|y)P(y),$$

Teoría de decisión estadística

entonces (Bayes):

$$P(y=1|\mathbf{x}) = \frac{P(\mathbf{x}|y=1)P(y=1)}{P(\mathbf{x})},$$

con el factor de normalización

$$P(\mathbf{x}) = \sum_{i=1}^{2} P(\mathbf{x}|y=i)P(y=i).$$

En palabras:

$$\text{posterior} = \frac{\text{verosimilitud} \times \text{apriori}}{\text{evidencia}}$$

Teoría de decisión estadística

Para el caso de clasificación binaria, la probabilidad de error está dada por

$$P(\text{error}|\mathbf{x}) = \begin{cases} P(y=1|\mathbf{x}) & \text{si decidimos } y=2 \\ P(y=2|\mathbf{x}) & \text{si decidimos } y=1 \end{cases},$$

y el error promedio es:

$$P(\text{error}) = \int_{-\infty}^{\infty} P(\text{error}|\mathbf{x})P(\mathbf{x})dx.$$

Teoría de decisión estadística

Si para cada $\mathbf{x}$, nos aseguramos que el error que cometemos es muy pequeño, entonces la integral debe ser muy pequeña, y eso se logra al usar la **regla de decisión Bayesiana**:

$$\text{decide } y = 1 \text{ si } P(y = 1|\mathbf{x}) > P(y = 2|\mathbf{x}),$$

o de forma equivalente:

$$\text{decide } y = 1 \text{ si } P(\mathbf{x}|y = 1)P(y = 1) > P(\mathbf{x}|y = 2)P(y = 2).$$

En ambos casos

$$P(\text{error}|\mathbf{x}) = \text{mín}\{P(y = 1|\mathbf{x}), P(y = 2|\mathbf{x})\},$$

ya que la categoría correcta se asigna de acuerdo a la probabilidad máxima, y bajo este supuesto, el error es asignar la probabilidad minima.

Teoría de decisión estadística

En forma general, el **clasificador óptimo Bayesiano** está dado por :

$$\hat{y} = \arg\max_{y_k} P(y = y_k|\mathbf{x}),$$

donde $\mathbf{x} \in \mathbb{R}^d$, $y_k \in \{1, 2, \ldots, K\}$, y las probabilidades posteriores están dadas por la fórmula de Bayes:

$$P(y = y_k|\mathbf{x}) = \frac{P(\mathbf{x}|y = y_k)P(y = y_k)}{P(\mathbf{x})}.$$

Otra forma de cuantificar la consecuencia de mi decisión es a través del riesgo y el costo asociado a ella.

Teoría de decisión estadística

Considera ésta matriz de costos para clasificación binaria

$$\mathbf{\Lambda} = \begin{pmatrix} 0 & \lambda_{-1,1} \\ \lambda_{1,-1} & 0 \end{pmatrix},$$

donde $\lambda_{i,j}$ es el costo de clasificar un objeto de la clase j como clase i. El costo promedio asociado a clasificar un dato $\mathbf{x}$ en la clase y_i, llamado función de Riesgo de Bayes, está dado por

$$R(y = y_i|\mathbf{x}) = \sum_{j \in \{-1,1\}} \lambda_{i,j} P(y = y_j|\mathbf{x}),$$

en nuestro caso, $R(y = -1|\mathbf{x}) = \lambda_{-1,1} P(y = 1|\mathbf{x})$ y $R(y = 1|\mathbf{x}) = \lambda_{1,-1} P(y = -1|\mathbf{x})$.

Teoría de decisión estadística

La decisión óptima es elegir aquella clase que minimice el riesgo, es decir

$$\hat{y} = \arg\min_{y_k} R(y = y_k|\mathbf{x}).$$

Es fácil ver que el clasificador de Bayes corresponde con esta regla de decisión, ya que ésta implica elegir $\hat{y} = -1$ si

$$R(y = -1|\mathbf{x}) < R(y = 1|\mathbf{x}),$$

o, en términos de probabilidades posteriores, si

$$\lambda_{1,-1} P(y = -1|\mathbf{x}) > \lambda_{-1,1} P(y = 1|\mathbf{x}),$$

que es equivalente al clasificador óptimo de Bayes.

Teoría de decisión estadística

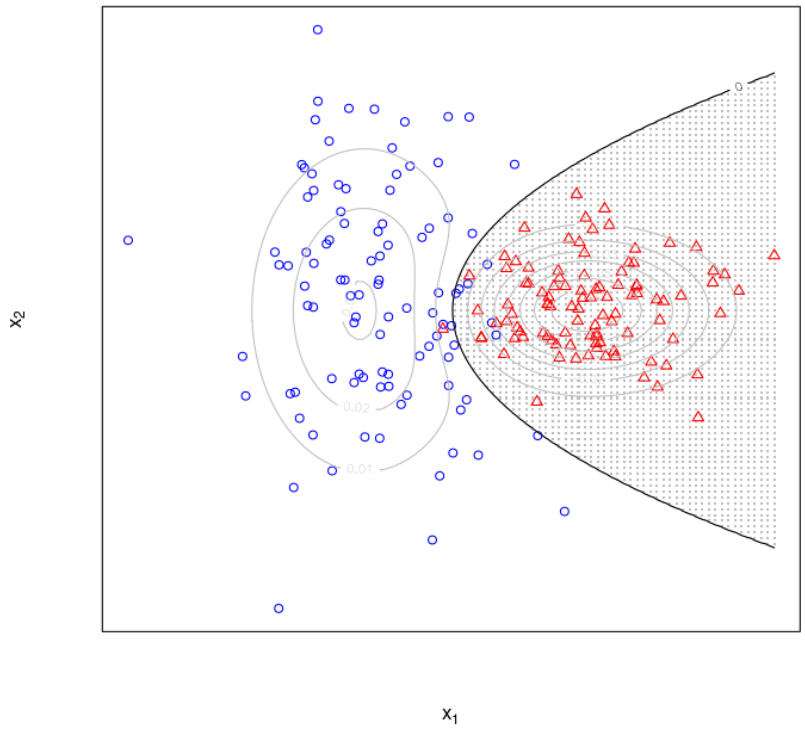
En este caso, las probabilidades posteriores estarán dadas por

$$\begin{aligned} \lambda_{-1,1} P(y=1|\mathbf{x}) &= \frac{P(\mathbf{x}|y{=}1)P(y{=}1)\lambda_{-1,1}}{P(\mathbf{x})}, \\ \lambda_{1,-1} P(y=-1|\mathbf{x}) &= \frac{P(\mathbf{x}|y{=}-1)P(y{=}-1)\lambda_{1,-1}}{P(\mathbf{x})}. \end{aligned}$$

De aquí podemos concluir que, considerar diferentes costos de mala clasificación es equivalente a **modificar las probabilidades apriori** $P(y = y_k)$**,** y en consecuencia, cambiar las fronteras de clasificación.

Teoría de decisión estadística

Ejemplo: Clase 0 (o). Clase 1 (+).

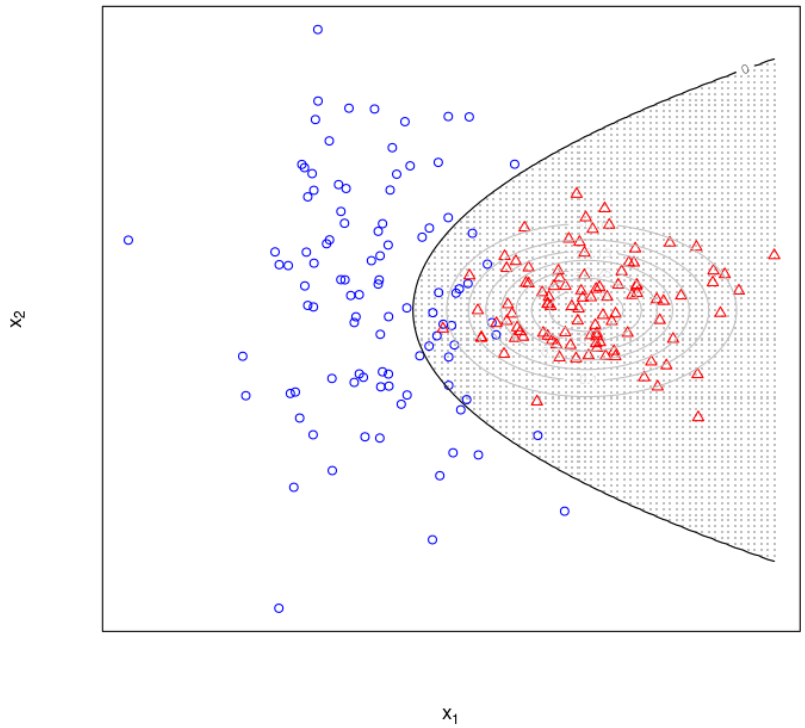


Clasificador Bayesiano óptimo.

$$P(y=1) = P(y=2) = 1/2, \qquad \mathbf{\Lambda} = \begin{pmatrix} 0 & 1 \\ 4 & 0 \end{pmatrix}$$

Teoría de decisión estadística

Ejemplo: Clase 0 (o). Clase 1 (+).

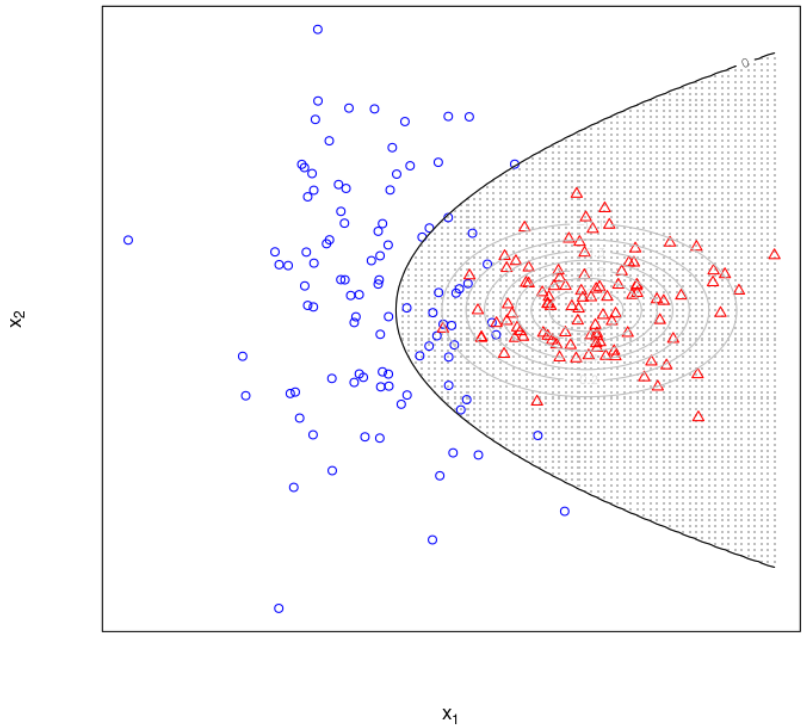


Clasificador Bayesiano óptimo.

$$P(y=1) = P(y=2) = 1/2, \qquad \mathbf{\Lambda} = \begin{pmatrix} 0 & 1 \\ 5 & 0 \end{pmatrix}$$

Teoría de decisión estadística

Ejemplo: Clase 0 (o). Clase 1 (+).



Clasificador Bayesiano óptimo.

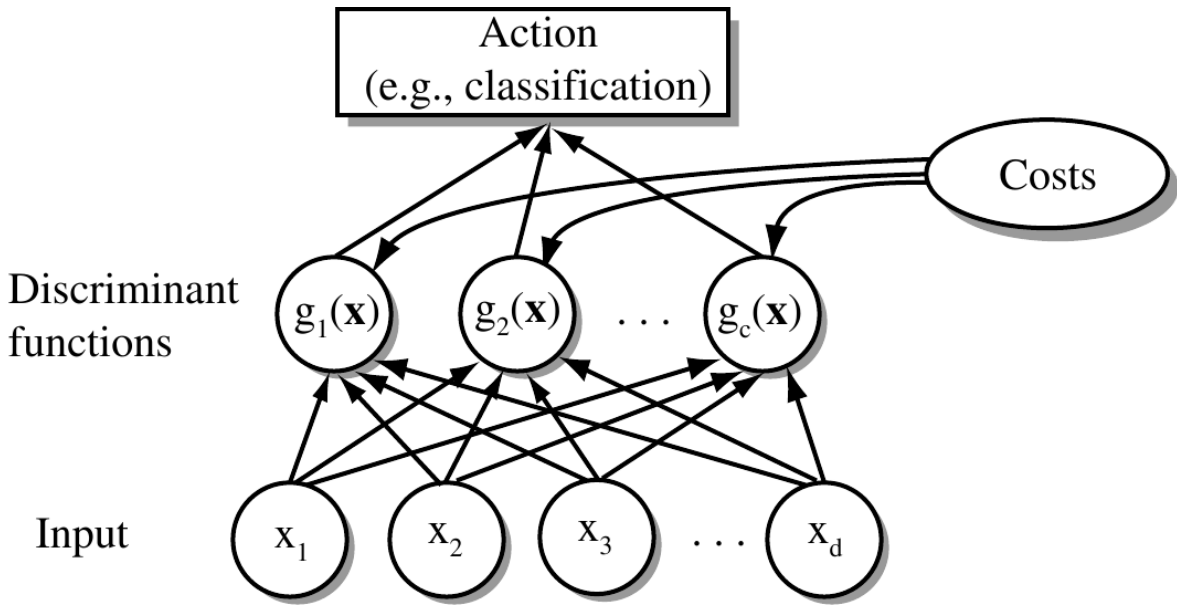
$$P(y=1) = P(y=2) = 1/2, \qquad \mathbf{\Lambda} = \begin{pmatrix} 0 & 1 \\ 10 & 0 \end{pmatrix}$$

Ejemplo: Funciones discriminantes para la distribución normal (LDA y QDA)

Ejemplo: LDA y QDA

Una forma de representar un clasificador es mediante funciones discriminantes $g(\mathbf{x})$. Para el caso general de clasificación en K clases, el clasificador **asigna** un objeto $\mathbf{x}$ a la clase k si

$$g_i(\mathbf{x}) > g_j(\mathbf{x}), \qquad \forall j \neq i, i = 1, 2, \ldots, K$$



Duda, Hart. Pattern Classification

Ejemplo: LDA y QDA

Según esta definición, podemos asignar $g_i(\mathbf{x}) = P(y = i|\mathbf{x})$, así la función discriminante máxima corresponderá a la probabilidad posterior máxima.
Las siguientes expresiones son equivalentes:

$$\begin{aligned} g_i(\mathbf{x}) &= P(y = i|\mathbf{x}) = \frac{P(\mathbf{x}|y=i)P(y=i)}{P(\mathbf{x})} \\ g_i(\mathbf{x}) &= P(\mathbf{x}|y = i)P(y = i) \\ g_i(\mathbf{x}) &= \log P(\mathbf{x}|y = i) + \log P(y = i) \end{aligned}$$

Las funciones discriminantes particionan el espacio de entrada (o espacio de características) en k regiones.

Ejemplo: LDA y QDA

Sin duda, el caso más estudiado es cuando se considera que nuestros datos provienen de una distribución normal: $X|y_i \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$. Es decir,

$$P(\mathbf{x}|y=i) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}_i|^{1/2}} \exp\left[\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)'\boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu_i})\right],$$

Simplificando la notación $y=i$ como y_i, las funciones discriminantes tienen la forma:

$$\begin{aligned} g_i(\mathbf{x}) &= \log P(\mathbf{x}|y_i) + \log P(y_i) \\ &= -\tfrac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)'\boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu_i}) - \tfrac{d}{2}\log 2\pi - \tfrac{1}{2}\log|\boldsymbol{\Sigma}_i| \\ &\quad +\log P(y_i) \end{aligned}$$

Ejemplo: LDA y QDA

3 casos principales:

- Varianzas iguales (LDA): $\boldsymbol{\Sigma}_i = \sigma^2 \mathbf{I}$.

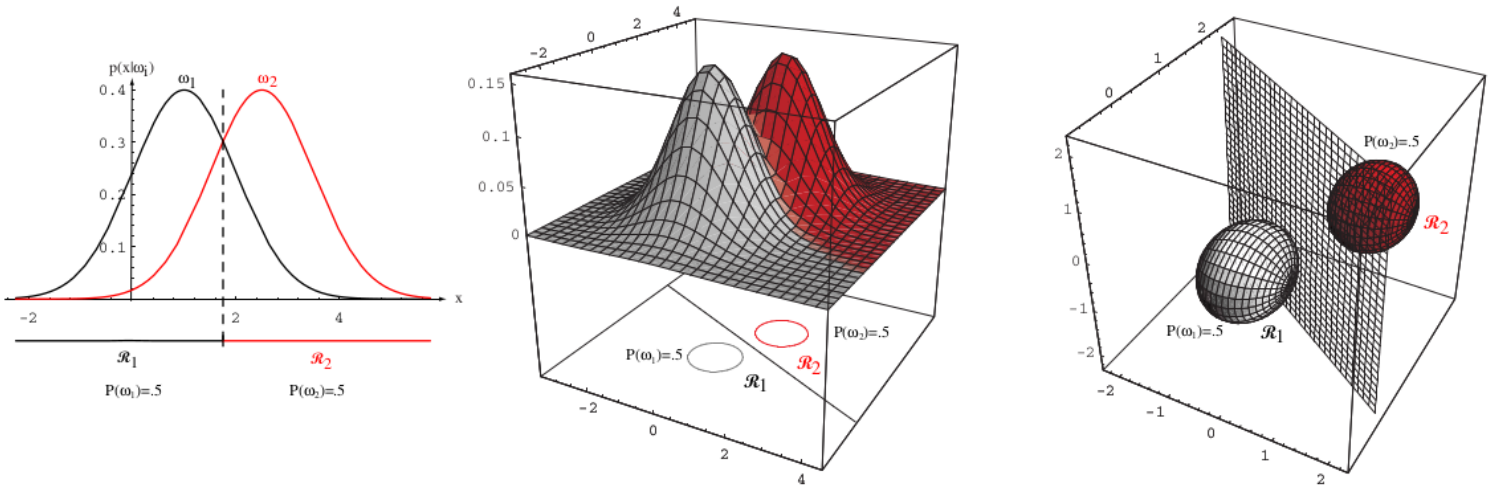
$$g_i(\mathbf{x}) = -\frac{\|\mathbf{x} - \boldsymbol{\mu}_i\|^2}{2\sigma^2} + \log P(y_i)$$

$$g_i(\mathbf{x}) = \mathbf{w}_i' \mathbf{x} + w_{i0},$$

con

$$\begin{aligned} \mathbf{w}_i &= \frac{1}{\sigma^2} \boldsymbol{\mu}_i \\ w_{i0} &= -\frac{1}{2\sigma^2} \boldsymbol{\mu}_i' \boldsymbol{\mu}_i + \log P(y_i) \end{aligned}$$

Ejemplo: LDA y QDA



Duda, Hart. Pattern Classification

Ejemplo: LDA y QDA

- Varianzas iguales (LDA): $\boldsymbol{\Sigma}_i = \boldsymbol{\Sigma}$.

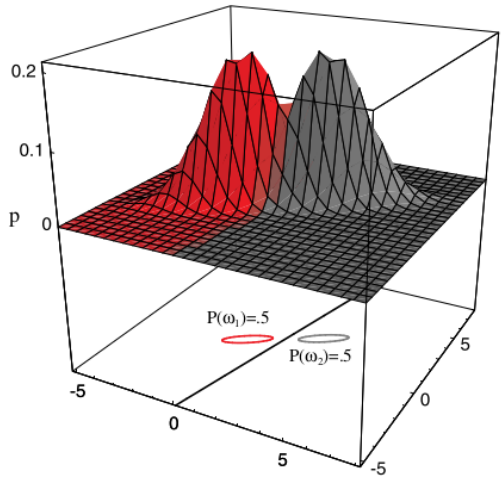
$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)'\boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}_i) + \log P(y_i).$$

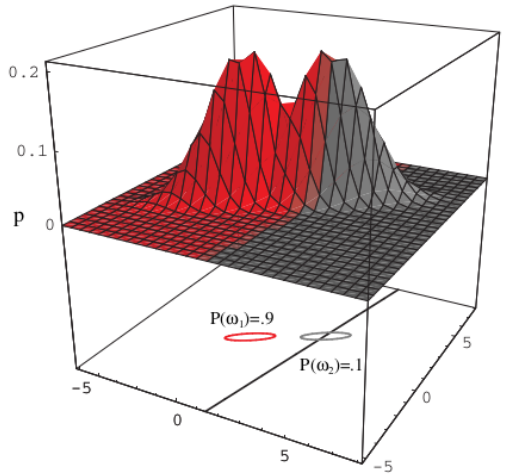
$$g_i(\mathbf{x}) = \mathbf{w}_i'\mathbf{x} + w_{i0},$$

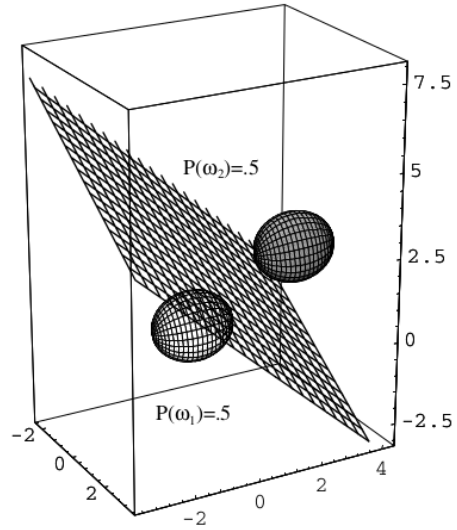
donde

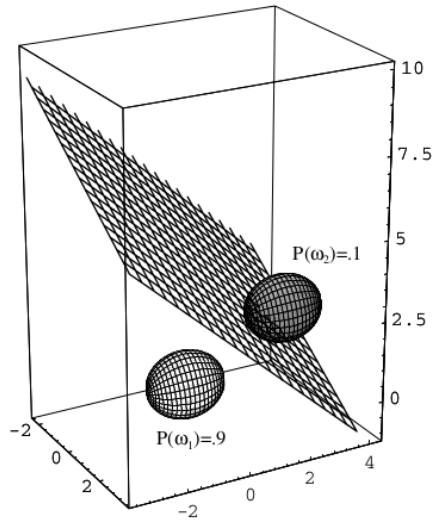
$$\begin{aligned} \mathbf{w}_i &= \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_i \\ w_{i0} &= -\tfrac{1}{2}\boldsymbol{\mu}_i'\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_i + \log P(y_i). \end{aligned}$$

Ejemplo: LDA y QDA









Duda, Hart. Pattern Classification

Ejemplo: LDA y QDA

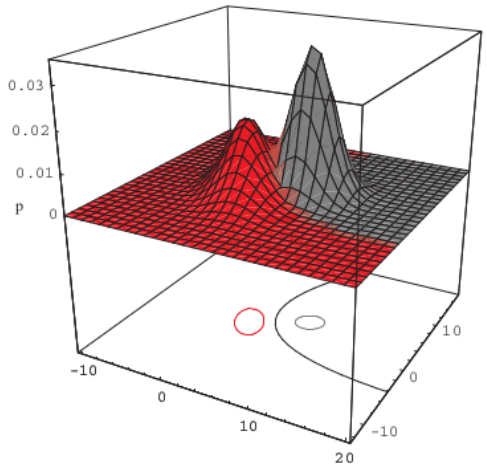
- Varianzas $\mathbf{\Sigma}_i$ son arbitrarias (QDA):

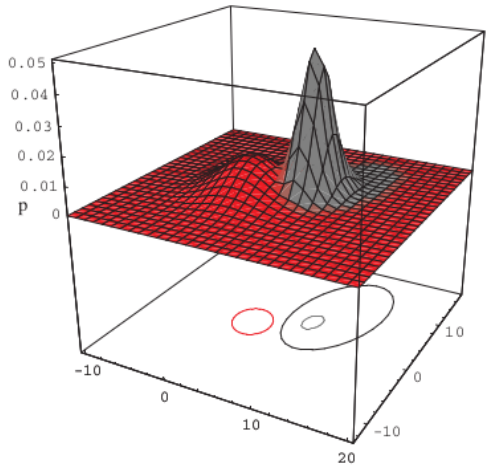
$$g_i(\mathbf{x}) = \mathbf{x}'\mathbf{W}_i'\mathbf{x} + \mathbf{w}_i'\mathbf{x} + w_{i0},$$

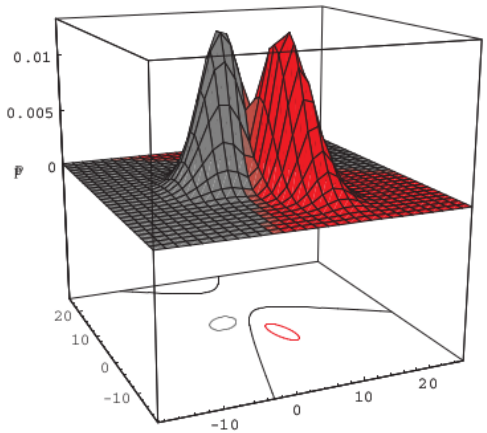
con

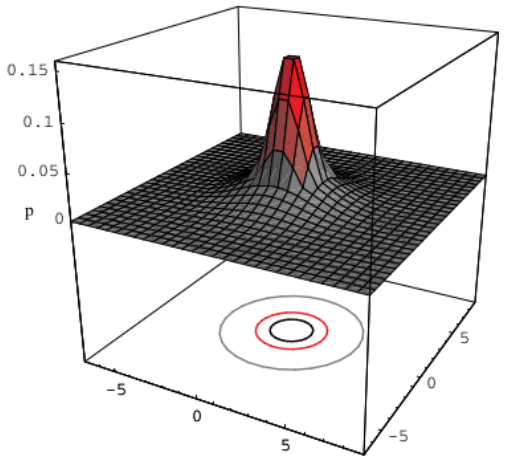
$$\begin{aligned} \mathbf{W}_i &= -\tfrac{1}{2}\mathbf{\Sigma}_i^{-1} \\ \mathbf{w}_i &= \mathbf{\Sigma}_i^{-1}\boldsymbol{\mu}_i \\ w_{i0} &= -\tfrac{1}{2}\boldsymbol{\mu}_i'\mathbf{\Sigma}_i^{-1}\boldsymbol{\mu}_i - \tfrac{1}{2}\log|\mathbf{\Sigma}_i| + \log P(y_i) \end{aligned}$$

Ejemplo: LDA y QDA









Duda, Hart. Pattern Classification