Teoría Bayesiana de la Decisión

MIAX-11, noviembre 2023

Repaso, conceptos previos

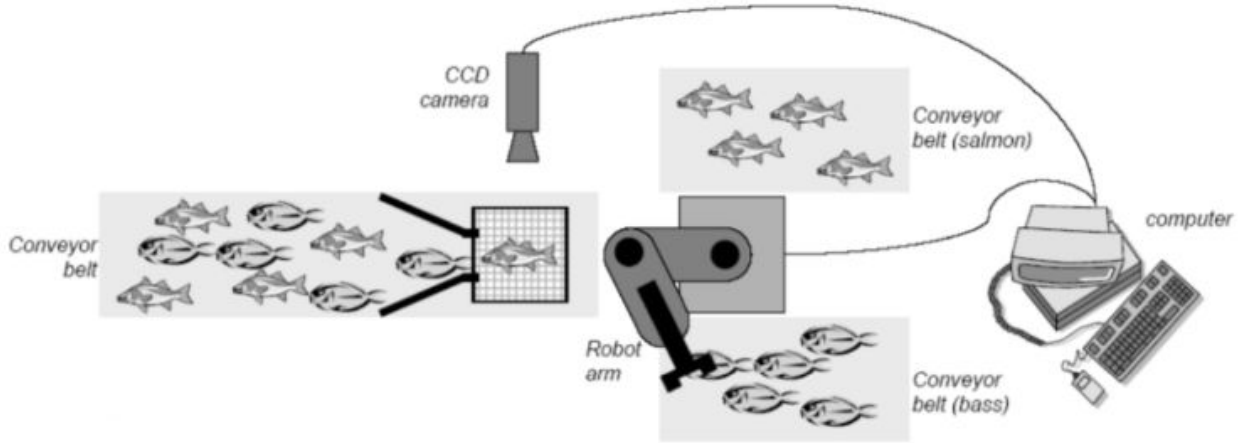
- **Aprendizaje automático**
  - Supervisado
  - No supervisado
  - Otros paradigmas
- **Aprendizaje supervisado** (pattern recognition)
  - Clasificación
  - Regresión
- Modelos **paramétricos** vs **no paramétricos**

Aprendizaje supervisado

- **Problema:** $\{(\boldsymbol{x}_1, t_1), (\boldsymbol{x}_2, t_2), \ldots, (\boldsymbol{x}_N, t_N)\}$
  - $\boldsymbol{x}_i \equiv$ vector de atributos
  - $t_i \equiv$ variable objetivo (target)
  - $N \equiv$ número de ejemplos/patrones
- **Objetivo:** predecir t a partir de $\boldsymbol{x}$
- Un **clasificador** (regresor) es una función que proporciona una estimación del objetivo: $y = f(\boldsymbol{x}) \approx t$
- Modelo **paramétrico**: la función f depende de unos parámetros ajustables (**entrenamiento**)

Ejemplo: salmones vs lubinas

- A fishing company wants to automate the process of separation of fish (salmon vs sea bass), using images recorded by a CCD camera



(From Duda, Hart and Stork, *Pattern Classification*, 2001)

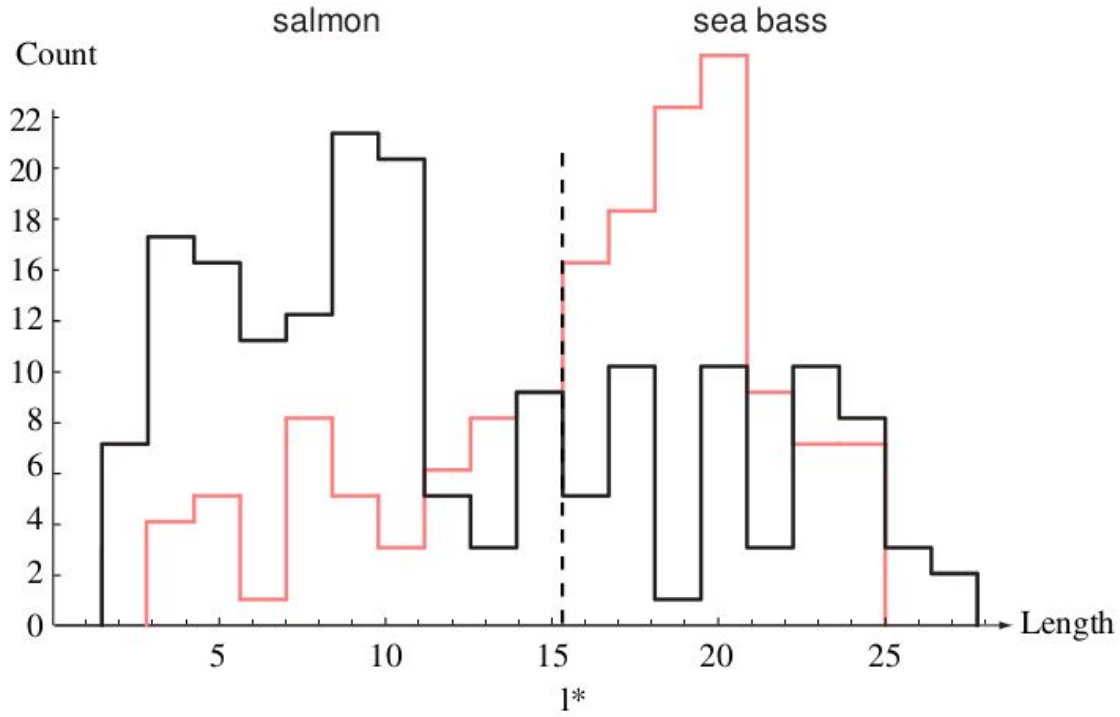
Ejemplo: salmones vs lubinas

Objetivo: ¿Salmón o lubina?

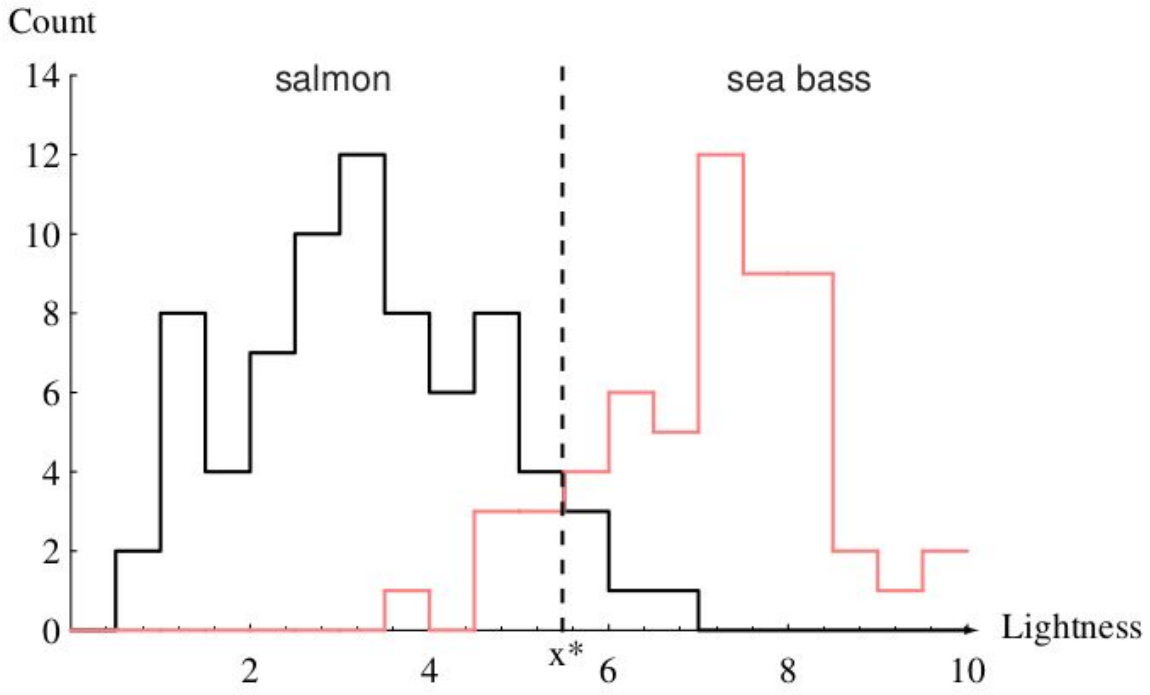
Atributos:

- Longitud
- Brillo
- Posición de las aletas
- …

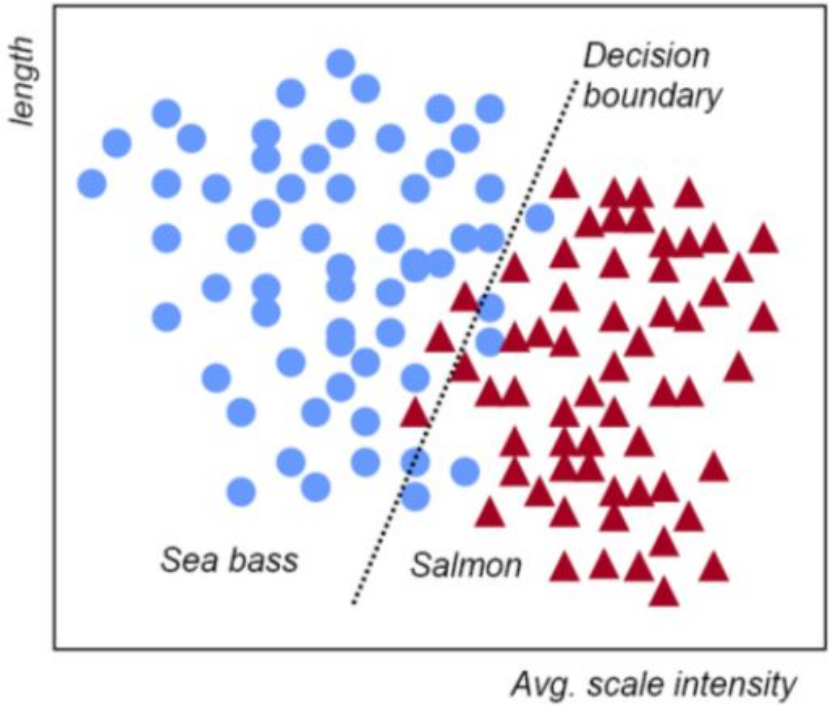
Distribución de clases de acuerdo a la longitud



Distribución de clases de acuerdo al brillo



Longitud y brillo conjuntamente



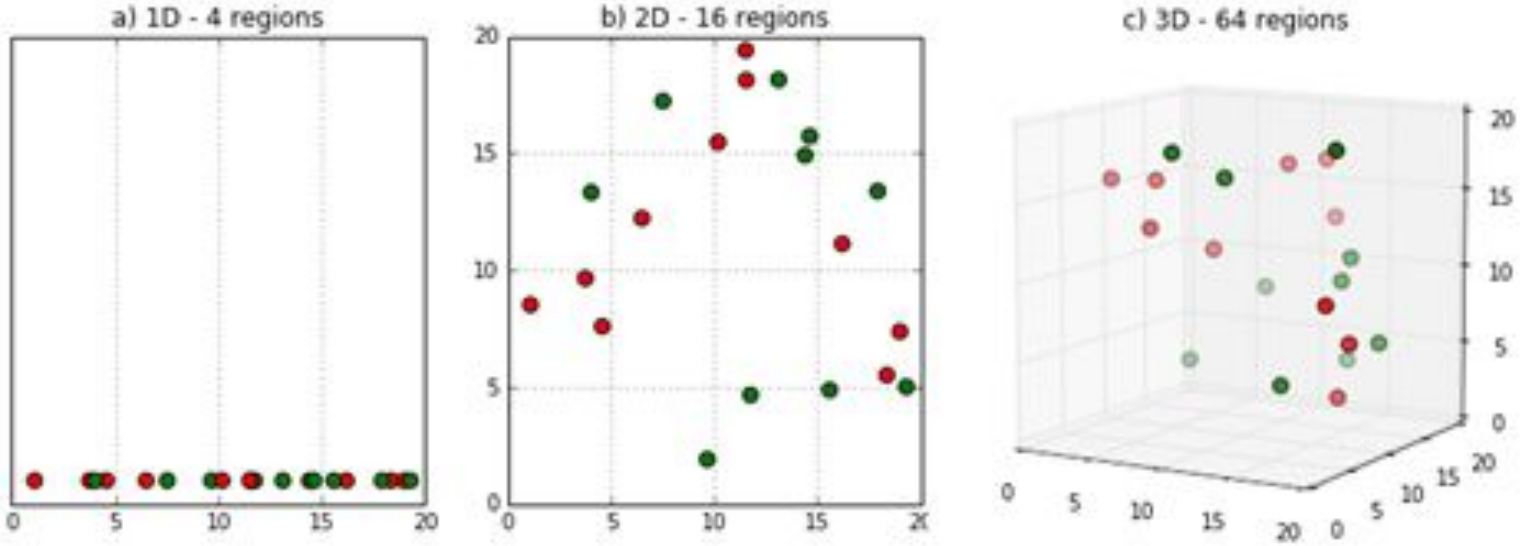
(From Duda, Hart and Stork, *Pattern Classification*, 2001)

Aprox. 95% acierto

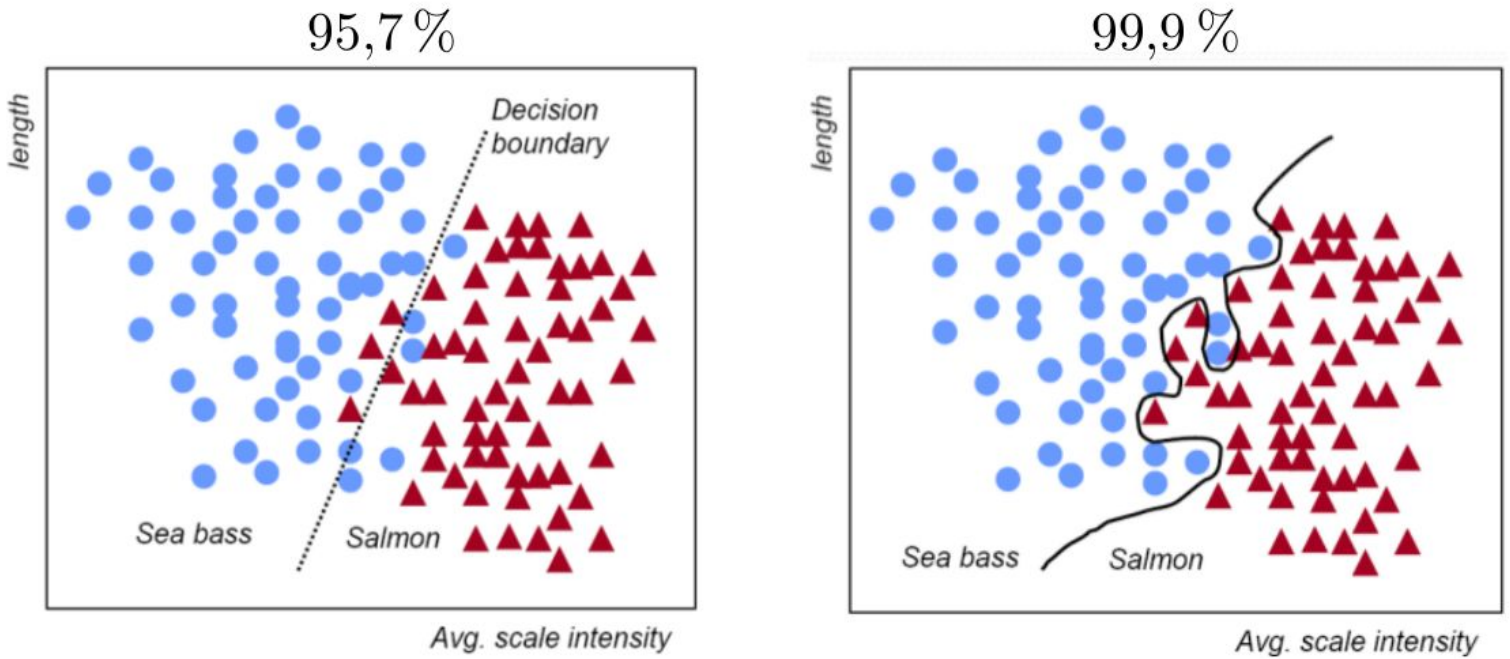
Longitud y brillo conjuntamente

- En este problema, añadir un nuevo atributo parece mejorar los resultados
- ¿Tendría sentido añadir un tercer atributo? ¿Y un cuarto? ¿Y …?
- ¿Hay un máximo de atributos?
- ¿Cuáles son los mejores?

La maldición de la dimensión (curse of dimensionality)



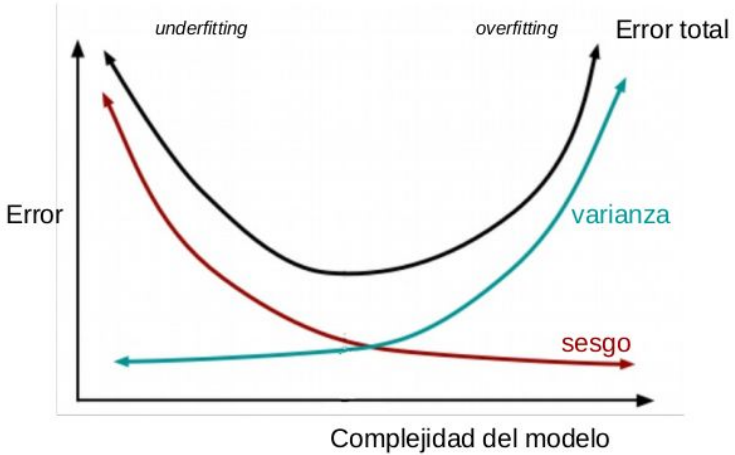
¿Cuál es el mejor modelo?



(From Duda, Hart and Stork, *Pattern Classification*, 2001)

Complejidad, generalización, sobreajuste

- Modelos más complejos se adaptan mejor a los datos
- Pero generalizan peor
- Sobreajuste / Overfitting
- Dilema sesgo - varianza



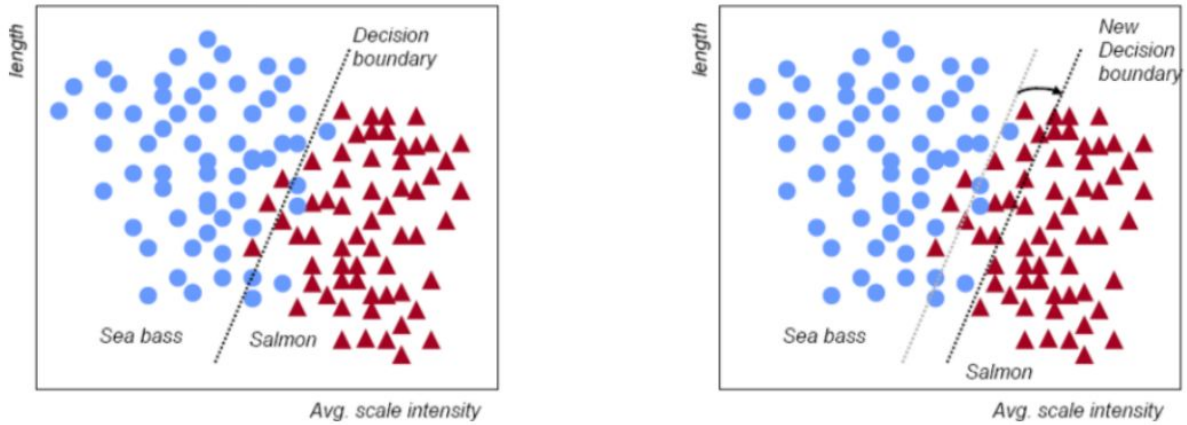
Entonces… ¿cuál es el mejor modelo?

No Free Lunch Theorem:

Si el objetivo es una buena generalización, no existen a priori motivos para decantarse por un clasificador frente a otro

Coste frente a precisión

- Usually misclassified patterns from different classes imply different costs
- In those cases we might want to adjust the decision boundary in order to minimize the cost associated to misclassifications



(From Duda, Hart and Stork, *Pattern Classification*, 2001)

En la clase de hoy

- Teoría Bayesiana de la decisión
- Estimación de densidades
- Clasificador Naive Bayes

Teoría Bayesiana de la decisión

- **Clasificación**
- Planteamiento **estadístico** del problema,
- Suposiciones:
  - El problema se puede plantear en términos de probabilidades y costes
  - Todas las probabilidades relevantes son conocidas

Volviendo al problema de los pescados

- La variable *w* representa la **clase**, $w \in \{salmón, lubina\}$
- **Probabilidad a priori:**
  - *p(salmón)*
  - *p(lubina)*
  - *p(salmón)* + *p(lubina)* = 1
  - Representa la probabilidad de observar un salmón/lubina en la cinta (sin información adicional)
- En general: $p(w_1) + p(w_2) + \ldots + p(w_c) = 1$, c = número de clases

Por ejemplo: *p(salmón)* = 0.6, *p(lubina)* = 0.4

Regla de decisión

Regla que prescribe la **acción a tomar** (*salmón*/*lubina*) basándose en la entrada (atributos) observada

Supongamos que:

- No hay datos de entrada (sólo conocemos el prior)
- El coste de cada clasificación errónea es el mismo (da igual confundir una lubina con un salmón que un salmón con una lubina)

Lo mejor que podemos hacer es:

$$\text{decisión} = \begin{cases} \textit{lubina} \text{ si } p(\textit{lubina}) > p(\textit{salmón}) \\ \textit{salmón} \text{ si } p(\textit{lubina}) < p(\textit{salmón}) \end{cases}$$

Decisión en base al prior

Elegir w_i tal que $p(w_i) \geq p(w_j)$, $j = 1, 2, \ldots, c$

- Elegimos siempre la **clase más probable**
- Regla **óptima** en ausencia de más información (minimiza la probabilidad de error)
- Pero siempre asigna la misma clase (la más probable) a todos los patrones

Por ejemplo, si *p(salmón)* = 0.6, *p(lubina)* = 0.4, esta regla diría que **todo son salmones**

Regla de decisión

Regla que prescribe la **acción a tomar** (*salmón*/*lubina*) basándose en la entrada (atributos) observada

Supongamos que:

- Conocemos un conjunto de variables que describen a cada patrón
- **Características/atributos** (supondremos continuas por simplicidad)

$$\mathbf{x} = (x_1, x_2, \ldots, x_d) \in R^d$$

- Longitud, anchura, brillo, posición de las aletas, …

¿Podemos mejorar la regla basada en el prior?

Probabilidad a posteriori

	L < 50 cm	**L ≥ 50 cm**	**Total**
Salmón	20	40	60
Lubina	30	10	40

Probabilidades **a priori:**

p(salmón) = **0.6**, *p(lubina)* = **0.4**

Probabilidades **a posteriori:**

p(salmón | L < 50 cm*)* = **0.4**, *p(lubina* | L < 50 cm*)* = **0.6**

p(salmón | L ≥ 50 cm*)* = **0.8**, *p(lubina* | L ≥ 50 cm*)* = **0.2**

Decisión MAP

Elegir w_i tal que $p(w_i \mid \boldsymbol{x}) \geq p(w_j \mid \boldsymbol{x})$, $j = 1, 2, \ldots, c$

- Elegimos siempre la **clase más probable dados los atributos observados**
- Estimación **MAP** (máximo a posteriori)
- La regla MAP también minimiza la probabilidad de error

En el ejemplo anterior, decidiremos *lubina* si L < 50 cm y *salmón* si L ≥ 50 cm

$p(\textit{salmón} \mid \text{L} < 50 \text{ cm}) = \mathbf{0.4}$, $p(\textit{lubina} \mid \text{L} < 50 \text{ cm}) = \mathbf{0.6}$

$p(\textit{salmón} \mid \text{L} \geq 50 \text{ cm}) = \mathbf{0.8}$, $p(\textit{lubina} \mid \text{L} \geq 50 \text{ cm}) = \mathbf{0.2}$

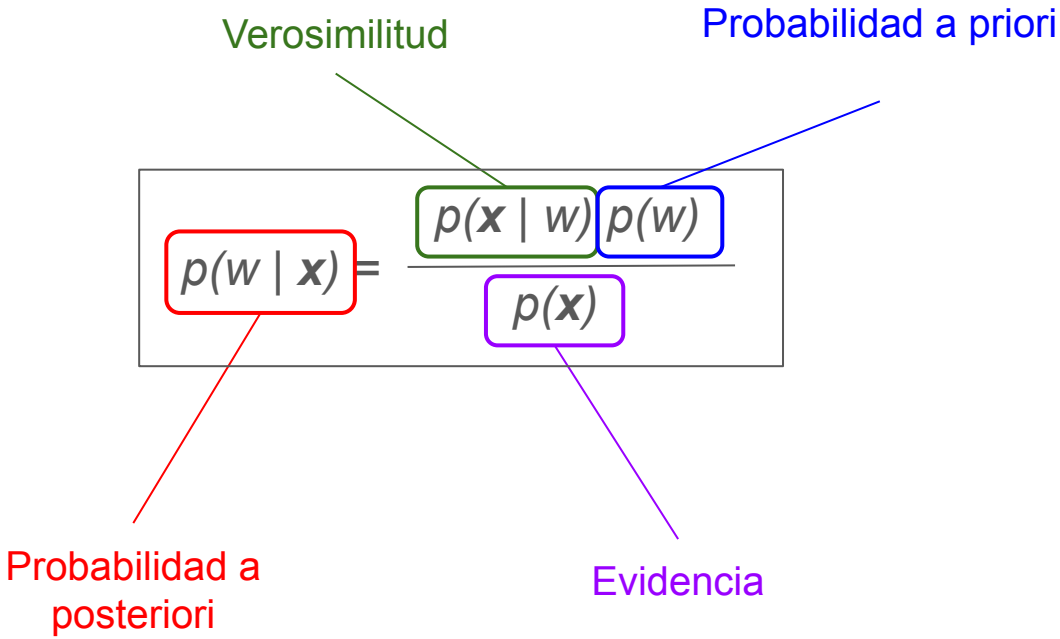
Teorema de Bayes

$$p(w \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid w)\, p(w)}{p(\boldsymbol{x})}$$

Se deriva a partir de la definición de la probabilidad conjunta:

$$p(w, \boldsymbol{x}) = p(w \mid \boldsymbol{x})\, p(\boldsymbol{x}) = p(\boldsymbol{x} \mid w)\, p(w)$$

Teorema de Bayes



Ejemplo

	L < 50 cm	**L ≥ 50 cm**	**Total**
Salmón	20	40	60
Lubina	30	10	40
Total	50	50	100

$$p(w \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid w)\, p(w)}{p(\boldsymbol{x})}$$

$$p(\text{salmón} \mid L < 50) = \frac{p(L < 50 \mid \text{salmón})\, p(\text{salmón})}{p(L < 50)}$$

Ejemplo

	L < 50 cm	L ≥ 50 cm	Total
Salmón	20	40	60
Lubina	30	10	40
Total	50	50	100

$$p(w \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid w)\, p(w)}{p(\boldsymbol{x})}$$

$$p(\text{salmón} \mid L < 50) = \frac{p(L < 50 \mid \text{salmón})\, p(\text{salmón})}{p(L < 50)}$$

20/50 = 0.4

Ejemplo

	L < 50 cm	L ≥ 50 cm	Total
Salmón	20	40	60
Lubina	30	10	40
Total	50	50	100

$$p(w \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid w)\, p(w)}{p(\boldsymbol{x})}$$

$$p(\text{salmón} \mid L < 50) = \frac{p(L < 50 \mid \text{salmón})\, p(\text{salmón})}{p(L < 50)}$$

$p(\text{salmón} \mid L < 50)$: 20/50 = 0.4

$p(L < 50 \mid \text{salmón})$: 20/60 = 0.33

Ejemplo

	L < 50 cm	L ≥ 50 cm	Total
Salmón	20	40	60
Lubina	30	10	40
Total	50	50	100

$$p(w \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid w)\, p(w)}{p(\boldsymbol{x})}$$

$$p(\text{salmón} \mid L < 50) = \frac{p(L < 50 \mid \text{salmón})\, p(\text{salmón})}{p(L < 50)}$$

$p(\text{salmón} \mid L < 50)$: 20/50 = 0.4

$p(L < 50 \mid \text{salmón})$: 20/60 = 0.33

$p(\text{salmón})$: 60/100 = 0.6

Ejemplo

	L < 50 cm	L ≥ 50 cm	Total
Salmón	20	40	60
Lubina	30	10	40
Total	50	50	100

$$p(w \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid w)\, p(w)}{p(\boldsymbol{x})}$$

$$p(\text{salmón} \mid L < 50) = \frac{p(L < 50 \mid \text{salmón})\, p(\text{salmón})}{p(L < 50)}$$

$p(\text{salmón} \mid L < 50)$: 20/50 = 0.4

$p(L < 50 \mid \text{salmón})$: 20/60 = 0.33

$p(\text{salmón})$: 60/100 = 0.6

$p(L < 50)$: 50/100 = 0.5

Decisión MAP

Elegir w_i tal que $p(w_i \mid \boldsymbol{x}) \geq p(w_j \mid \boldsymbol{x})$, $j = 1, 2, \ldots, c$

$$p(w \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid w)\, p(w)}{p(\boldsymbol{x})}$$

La evidencia $p(\boldsymbol{x})$ no depende de w

Elegir w_i tal que $p(\boldsymbol{x} \mid w_i)\, p(w_i) \geq p(\boldsymbol{x} \mid w_j)\, p(w_j)$, $j = 1, 2, \ldots, c$

$$w^* = \underset{i}{\operatorname{argmax}}\; p(\boldsymbol{x} \mid w_i)\, p(w_i)$$

Decisión MAP

$$w^* = \underset{i}{\operatorname{argmax}}\ p(\boldsymbol{x} \mid w_i)\, p(w_i)$$

Si conocemos $p(w_i)$ y $p(\boldsymbol{x} \mid w_i)$, esto es lo mejor que podemos hacer

El clasificador de Bayes es óptimo

La decisión depende sólo de:

- Los **prioris**, $p(w_i)$
- Las **verosimilitudes**, $p(\boldsymbol{x} \mid w_i)$

(Si los prioris son uniformes, la decisión depende únicamente de la verosimilitud)

Coste (no simétrico) asociado a cada tipo de error

$C(w_i \mid w_j)$ es el coste de elegir la clase w_i cuando la clase real es w_j

Por ejemplo:

Predicción

Clase real		**Salmón**	**Lubina**
	Salmón	0	100
	Lubina	50	0

Nos cuesta más clasificar un salmón como lubina que una lubina como salmón

Riesgo o coste esperado

$$R(w \mid \mathbf{x}) = C(w \mid w_1)\, p(w_1 \mid \mathbf{x}) + C(w \mid w_2)\, p(w_2 \mid \mathbf{x}) + \ldots + C(w \mid w_c)\, p(w_c \mid \mathbf{x})$$

La mejor decisión en este caso es:

$$w^* = \underset{i}{\operatorname{argmin}}\, R(w_i \mid \mathbf{x})$$

Que coincide con la regla de Bayes cuando el coste es:

$$C(w_i \mid w_j) = \begin{cases} 0 \text{ si } i = j \\ 1 \text{ si } i \neq j \end{cases}$$

Resumen

Clasificador de Bayes / regla MAP:

$$w^* = \underset{i}{\operatorname{argmax}}\ p(\boldsymbol{x} \mid w_i)\, p(w_i)$$

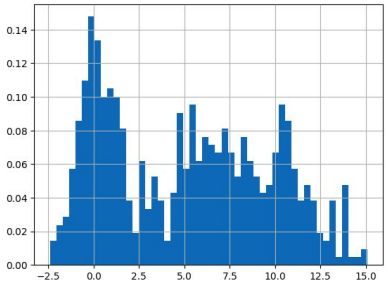
Necesitamos conocer:

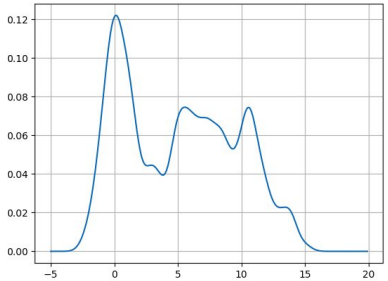
- Los **prioris**, $p(w_i)$
- Las **verosimilitudes**, $p(\boldsymbol{x} \mid w_i)$

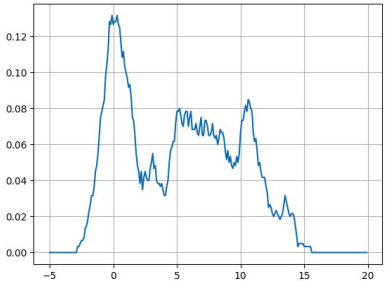
¿Cómo podemos estimarlos a partir de una muestra finita?

Estimación de densidades

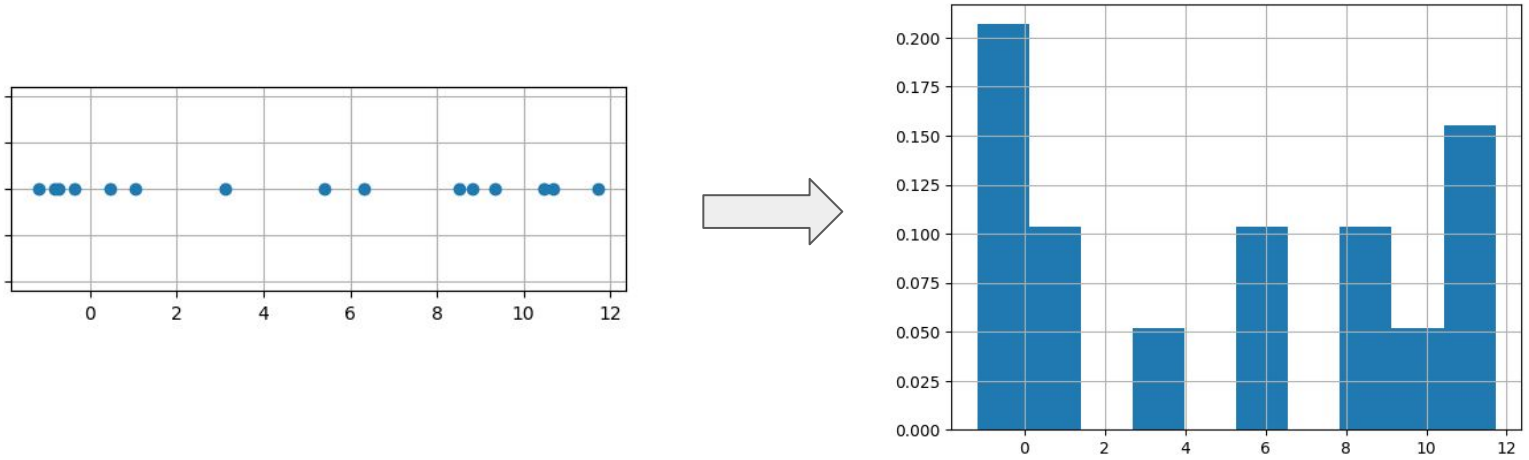
- Mediante histogramas
- Mediante kernels
- Estimación paramétrica de densidades (EM)
- Naive Bayes







Estimación mediante histogramas

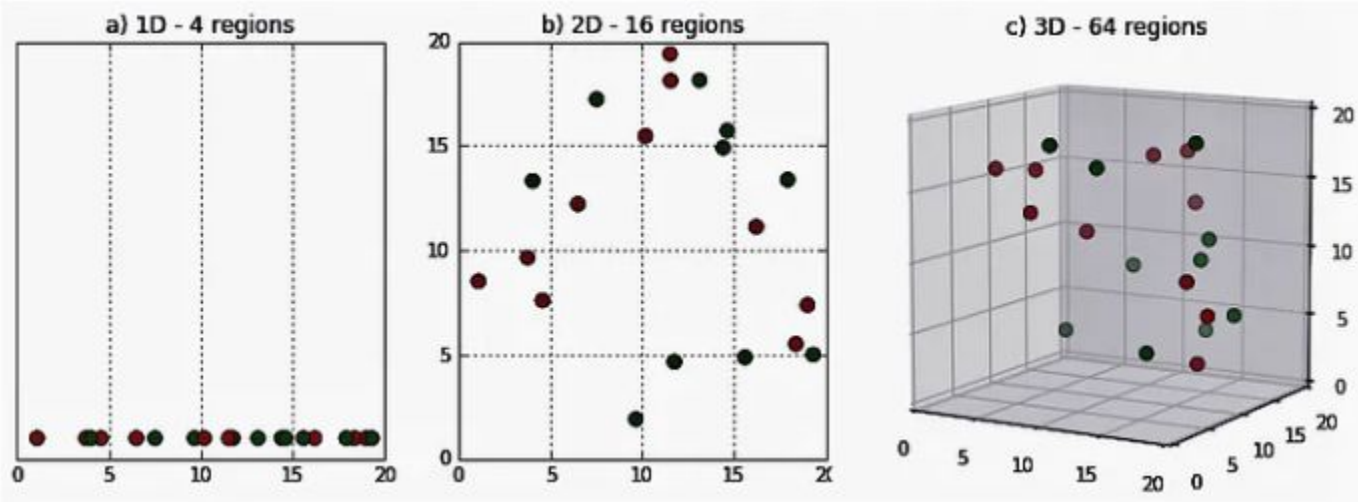


Problemas:

- Cómo elegir el tamaño del bin
- Difícil de aplicar en dimensión alta (**maldición de la dimensión**)

La maldición de la dimensión

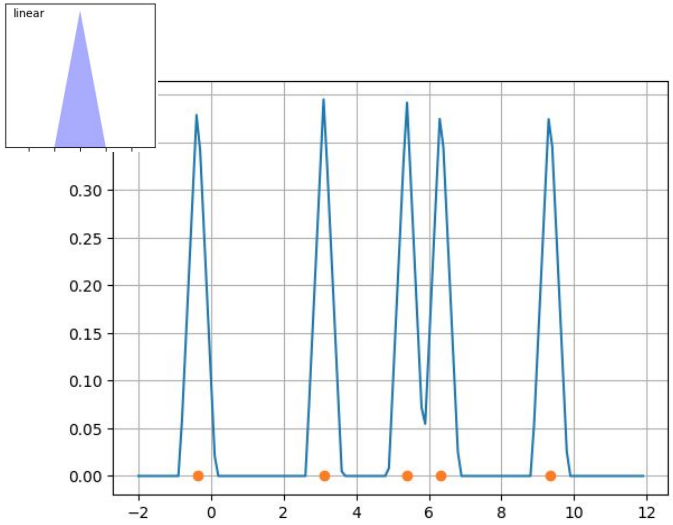
A medida que aumenta la dimensión del problema, el número de puntos necesarios para estimar correctamente la densidad crece exponencialmente

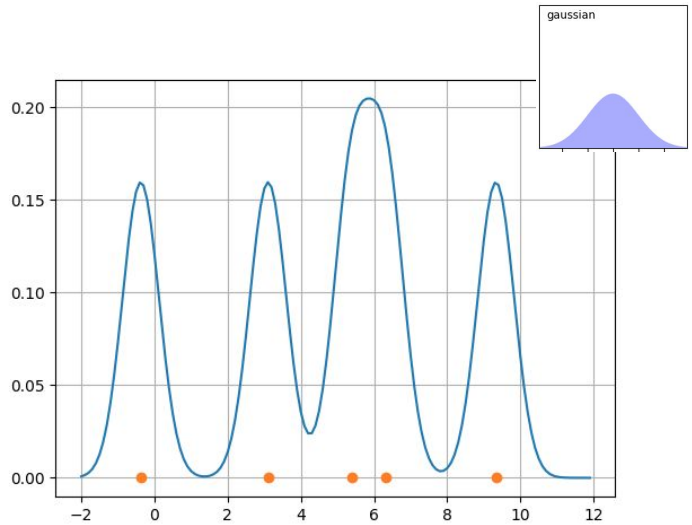


Estimación mediante kernels

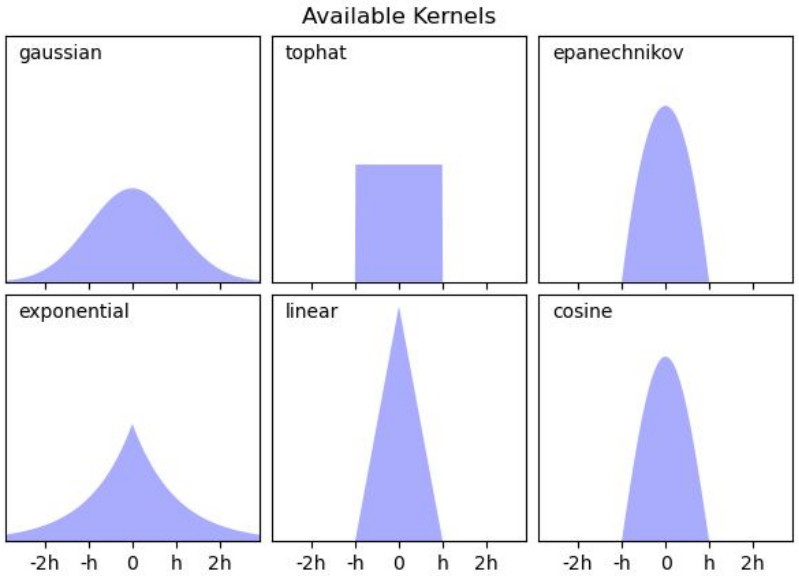
La densidad es la suma de una **función de kernel** aplicada sobre cada punto del problema:

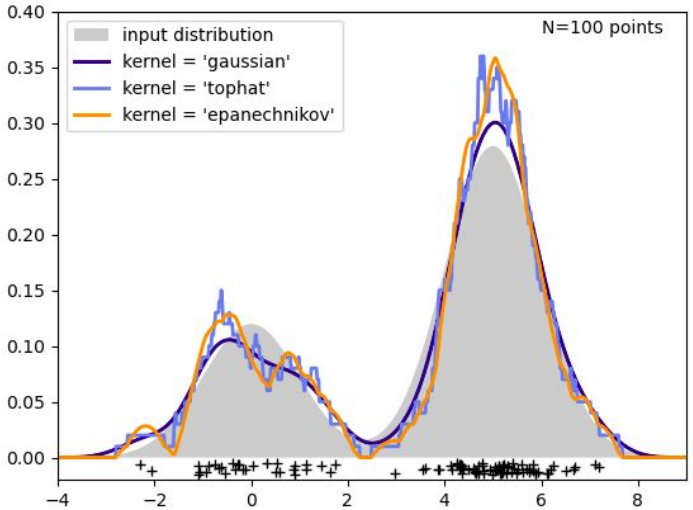
$$\rho(\boldsymbol{x}) = \sum_i k(\boldsymbol{x}-\boldsymbol{x}_i)$$





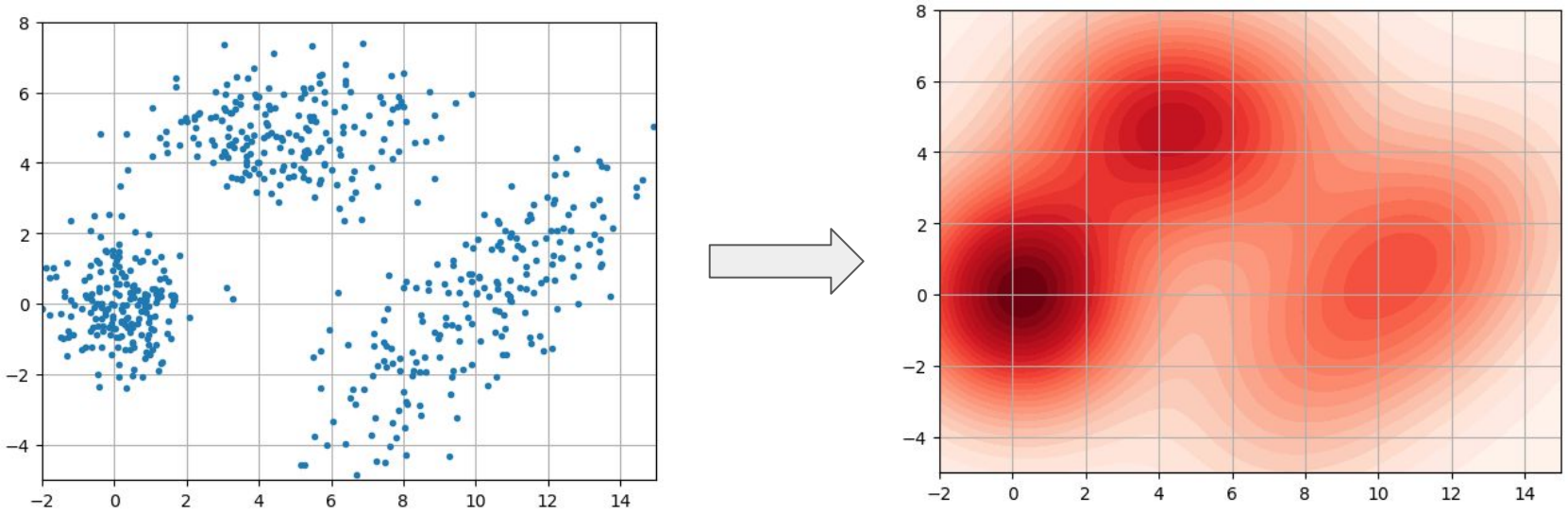
Tipos de kernel



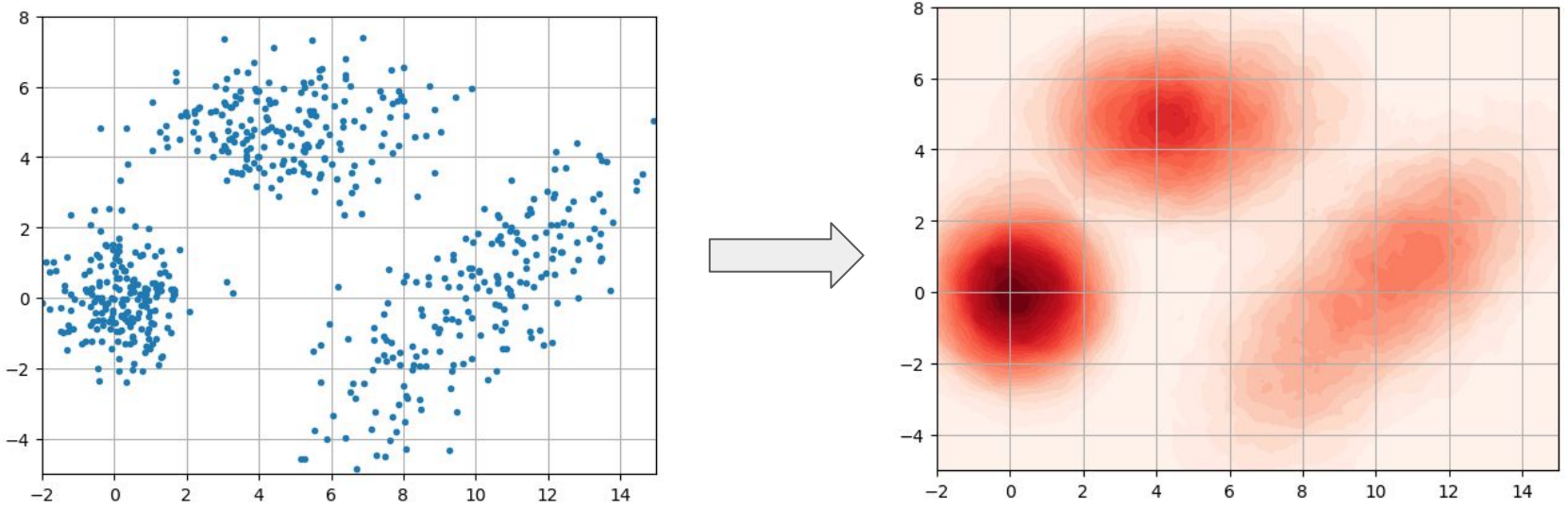


https://scikit-learn.org/stable/modules/density.html#kernel-density

Kernel gausiano



Kernel tophat

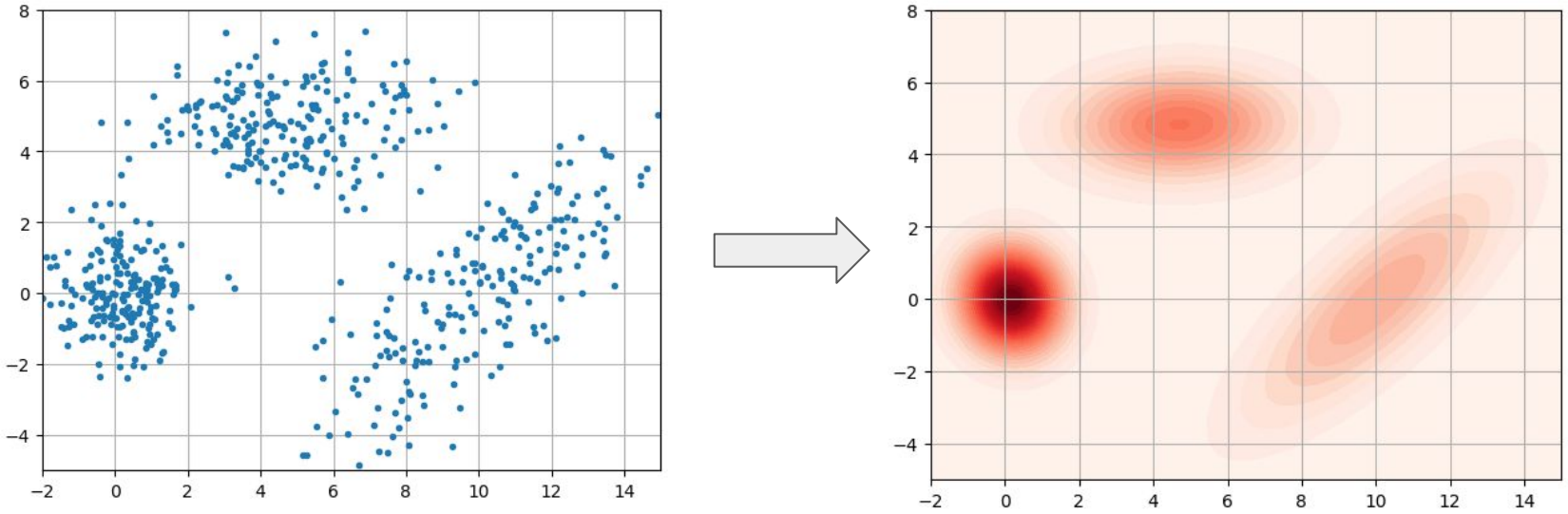


Estimación paramétrica (máxima verosimilitud)

- Suponemos forma funcional conocida, $f(\boldsymbol{x}, \boldsymbol{\theta})$
- Ajustamos los parámetros $\boldsymbol{\theta}$ para maximizar la probabilidad de las observaciones $\boldsymbol{x}$ (**máxima verosimilitud**)
- Puede usarse el **algoritmo EM**
- Es típico suponer que *f* es una **mezcla de distribuciones gausianas** cuyos parámetros (medias y covarianzas) se desconocen

https://scikit-learn.org/stable/modules/mixture.html#gmm

Mezcla de gausianas



Clasificador Naive Bayes

Clasificador que combina el teorema de Bayes con la suposición de que los atributos son **independientes** dada la clase:

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y)P(x_1, \ldots, x_n \mid y)}{P(x_1, \ldots, x_n)}$$

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i \mid y)}{P(x_1, \ldots, x_n)}$$

$$\hat{y} = \arg\max_y P(y) \prod_{i=1}^{n} P(x_i \mid y)$$

Es típico suponer además que las distribuciones son **gausianas**

Naive Bayes gausiano

1.9.1. Gaussian Naive Bayes

`GaussianNB` implements the Gaussian Naive Bayes algorithm for classification. The likelihood of the features is assumed to be Gaussian:

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

The parameters σ_y and μ_y are estimated using maximum likelihood.

Presupone independencia entre los atributos

https://scikit-learn.org/stable/modules/naive_bayes.html#gaussian-naive-bayes

Ejemplos

Notebook *8_1_estimacion_densidades.ipynb*