
Clasificadores Generativos Básicos

Aprendizaje Automático II-2018

Prof. Ricardo Ñanculef

UTFSM Campus San Joaquín



1 Motivación

2 Clasificadores Bayesianos Ingenuos

3 Análisis de Discriminantes Gausianos (GDA)



Motivación



Recordemos que (por Bayes)

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)} \propto p(x|y)p(y).$$

A partir de esta observación aparecen dos tipos de clasificadores:

- **Clasificadores Discriminativos.** Aprenden directamente $p(y|x)$ o una transformación monótona de $p(y|x)$ que permite identificar la clase más probable *a-posteriori*.
- **Clasificadores Generativos.** Aprenden separadamente el *a-priori* $p(y)$ y la *verosimilitud* $p(x|y)$, para luego explotar la regla de Bayes.

Si bien un enfoque discriminativo tiende a tener mejores resultados **predictivos**, un enfoque generativo puede ser usado para “fabricar” data sintética vía simulación.



Es útil entender un **modelo probabilístico** como una descripción matemática acerca de cómo se podría haber generado un conjunto de datos. Por ejemplo, si necesitamos explicar las calificaciones que observamos en un certamen de Machine Learning a lo largo de los últimos años, podríamos formular el modelo:

$$D \sim \text{Ber}(0.5) \quad (\text{Certamen difícil?})$$

$$C|D \sim D \cdot \mathcal{N}(50, 5) + (1 - D) \cdot \mathcal{N}(70, 10)$$

Simular el modelo consiste en implementar un algoritmo capaz de producir datos que "luzcan como" los datos que el modelo intentaba explicar. Dado que el modelo es probabilístico, tal y como la realidad que se intenta explicar (no vemos los mismos datos siempre), esto requiere ser capaces de generar números aleatorios.



Algorithm 1: Genera “notas sintéticas” X_1, X_2, \dots, X_n que se distribuyen de acuerdo al modelo del ejemplo anterior.

1 Lanzar una moneda para determinar la dificultad del Quiz.

2 **for** $i = 1, \dots, n$ **do**

3 **if** Quiz difícil **then**

4 Simular una nota desde $\mathcal{N}(50, 5)$

5 **else**

6 Simular una nota desde $\mathcal{N}(70, 10)$

Naturalmente, esto requiere poder simular el modelo (más simple) $\mathcal{N}(\mu, \sigma^2)$.



Asumiendo que ya se dispone de un método para simular el modelo $U(0, 1)$, un método muy sencillo para simular modelos cuya CDF (función de distribución acumulada) $F(x)$ es analíticamente simple, consiste en el método de la transformación inversa.

Teorema

Sea $U \sim U(0, 1)$, $X \sim \mathcal{D}$, y $F_x(x) = P(X \leq x)$ la CDF correspondiente a la distribución \mathcal{D} . Entonces $Z = F_x^{-1}(U) \sim \mathcal{D}$.

Demostración

Como $F(\cdot)$ es no decreciente, $F(a) \leq F(b)$ si y sólo si $a \leq b$. Por lo tanto,

$$\begin{aligned} P(Z \leq a) &= P(F_x^{-1}(U) \leq a) = P(F_x(F_x^{-1}(U)) \leq F_x(a)) \\ &= P(U \leq F_x(a)) = F_u(F_x(a)) = F_x(a) \end{aligned}$$

donde el último paso se debe a que la CDF de la distribución $U(0, 1)$ es $F_u(u) = u$.



Ejemplo: Simular el modelo $\text{Exp}(\lambda)$ es muy sencillo, ya que la CDF en este caso viene dada por la expresión $F(x) = 1 - \exp(-\lambda x)$, que se puede invertir fácilmente,

$$F_x^{-1}(u) = -\frac{\log(1-u)}{\lambda}$$

Algorithm 2: Simulación de n datos $\text{Exp}(\lambda)$.

-
- 1 **for** $i = 1, \dots, n$ **do**
 - 2 Generar un número $U \sim U(0, 1)$.
 - 3 Entregar el número $X_i = -\log(1-u)/\lambda$.
-



Para simular una normal estándar podríamos invocar el siguiente resultado:

Teorema

X e Y siguen distribuciones normales $\mathcal{N}(0, 1)$ independientes si y sólo si $R^2 = X^2 + Y^2$ y $\theta = \sin^{-1}(Y/R)$ siguen distribuciones $\text{Exp}(1/2)$ y $U(0, 1)$ respectivamente.

Algorithm 3: Simulación de $2n$ datos $\mathcal{N}(0, 1)$.

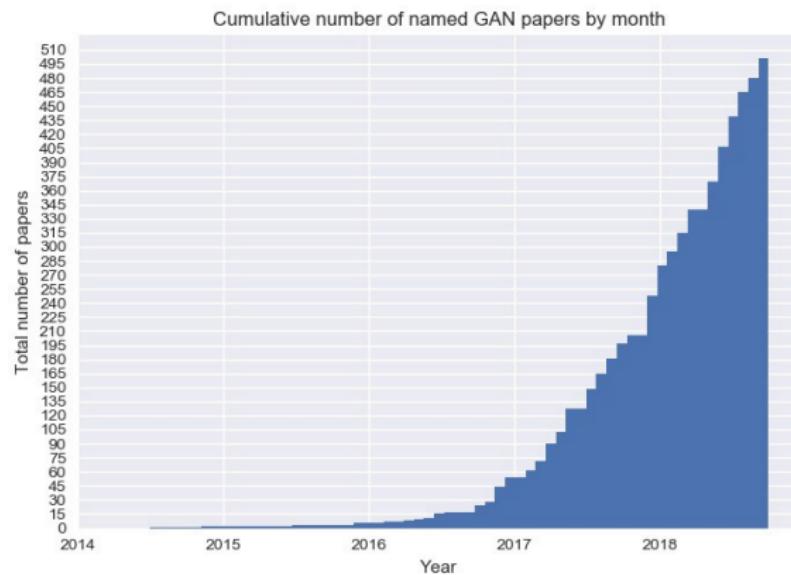
```
1 for  $i = 1, \dots, n$  do
2   Generar un número  $U \sim U(0, 1)$ .
3   Generar un número  $R_i^2 = -2 \log(1 - u)$ .
4   Generar un número  $\theta \sim U(0, 1)$ .
5   Entregar  $X_i = R_i \cos(\theta)$ 
6   Entregar  $Y_i = R_i \sin(\theta)$ 
```

La simulación de normales $\mathcal{N}(\mu, \sigma)$ a partir de normales $\mathcal{N}(0, 1)$ es muy simple.

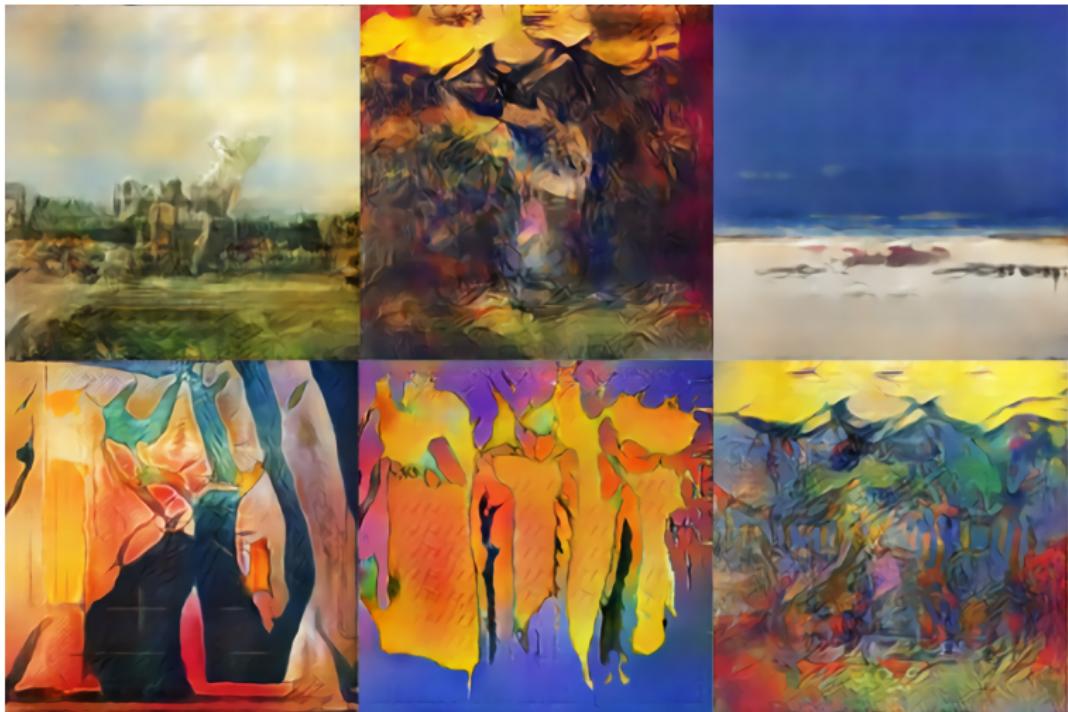


Métodos Generativos Actuales

Durante los últimos años, el área de los métodos generativos en aprendizaje automático es una de las más activas y de mayor crecimiento.



Métodos Generativos Actuales



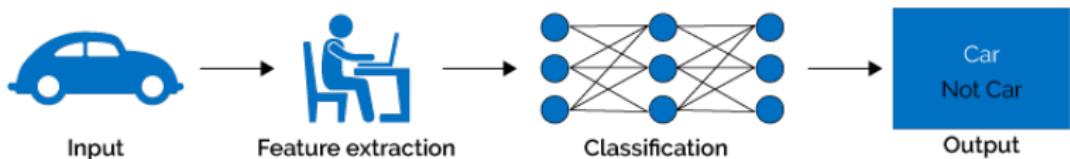
INPUT



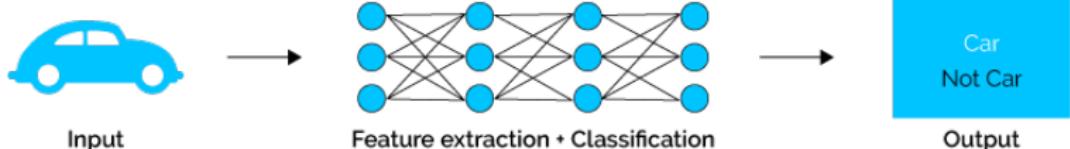
OUTPUT



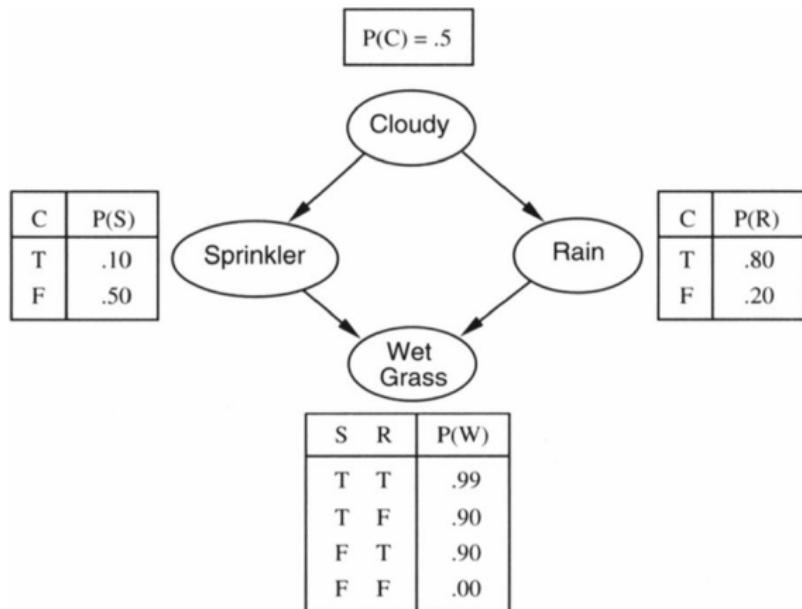
Machine Learning

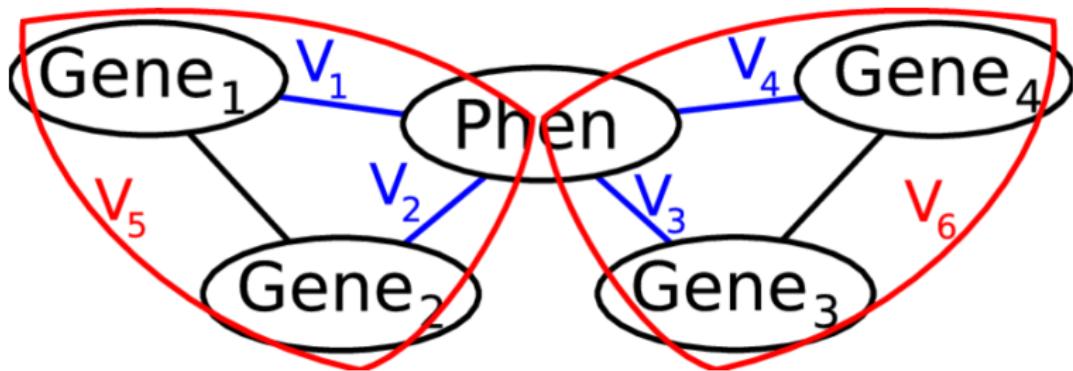


Deep Learning



Representación del Modelo





Clasificadores Bayesianos Ingenuos



- Supongamos entonces que decidimos aproximar $p(y|x)$ de modo generativo, es decir explotando la descomposición

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)} \propto p(x|y)p(y).$$

- Debemos definir un modo para aproximar $p(x|y)$ y $p(y)$ a partir de un conjunto de ejemplos etiquetados $S = \{(x^{(1)}, y^{(1)}), (x^{(1)}, y^{(2)}), \dots, (x^{(n)}, y^{(n)})\}$.
- Estimar $p(y)$ suele ser fácil porque, en clasificación, $y \in \{c_1, c_2, c_K\}$, de modo que sólo debemos estimar $K - 1$ probabilidades (porqué no K ?).
- Recordando la Ley de los Grandes Números (LGN) y definiendo n_j como el número de ejemplos de la clase c_j , podríamos considerar en efecto los estimadores

$$\theta_j = \frac{\sum_{i=1}^n I(y^{(i)} = c_j)}{n} = \frac{n_j}{n} \quad \forall j. \tag{1}$$



Estimación de $p(y)$

- Gracias a las LGN sabes que estos estimadores son consistentes (bajo ciertos supuestos)

$$\theta_j = \frac{n_j}{n} \xrightarrow[n \rightarrow \infty]{a.s.} p(y = c_j).$$

- En efecto, de la desigualdad de Hoeffding

$$P\left(\left|\frac{n_j}{n} - p(y = c_j)\right| > \epsilon\right) \leq 2 \exp\left(-2m\epsilon^2\right),$$

concluimos que para estimar $p(y = c_j)$, $j = 1, 2, \dots, K$, con error de a lo más ϵ y confianza de a lo menos δ , necesitamos del orden de $m = \mathcal{O}(-K \log(\delta)/\epsilon^2)$ datos.

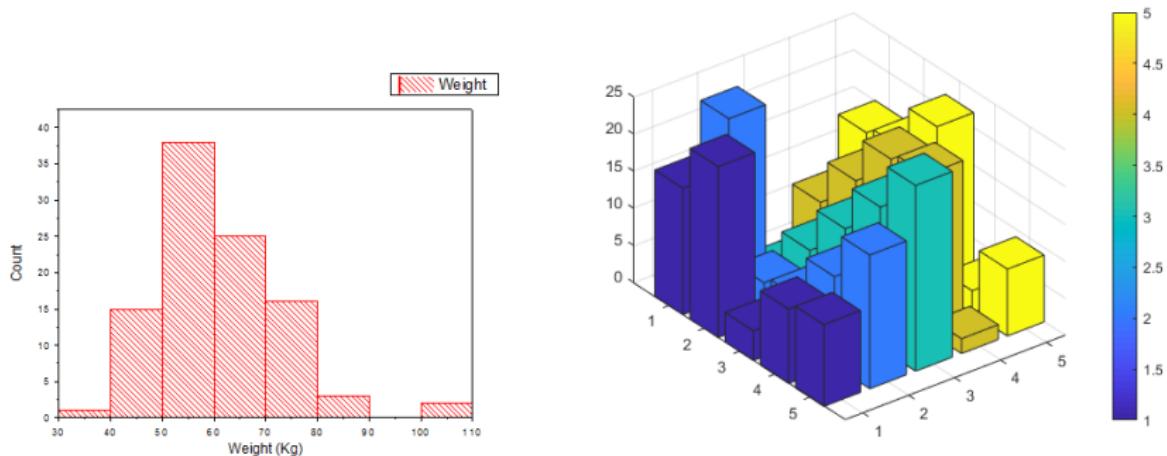
- Ahora, ¿Cómo estimamos $p(x|y)$ para implementar la regla de decisión

$$\arg \max_j p(y = c_j|x) = \arg \max_j p(x|y = c_j)p(y = c_j)?$$

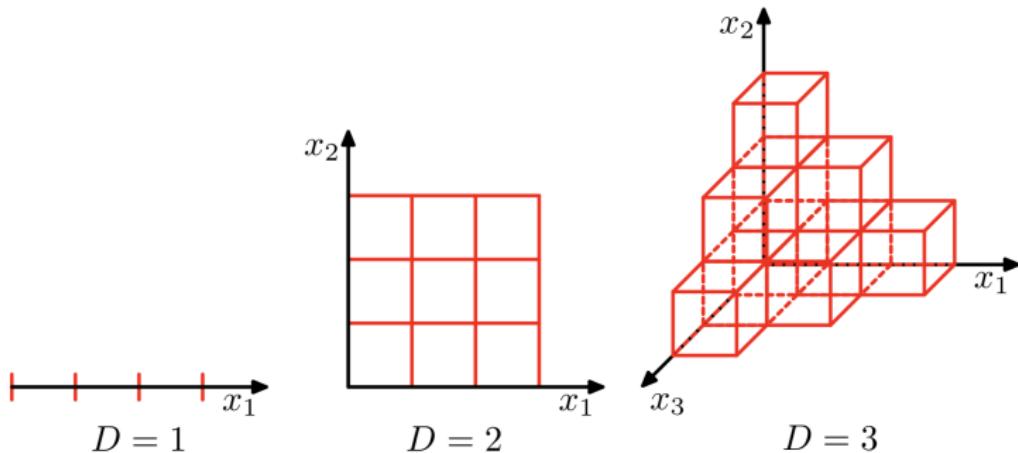


Estimación de $p(x|y)$

- La diferencia fundamental acá es que $x \in \mathbb{R}^d$, y c.s. $d \gg 1$, situación que desata el problema denominado “maldición de la dimensionalidad”.
- Para visualizar el problema, imaginemos que decidimos estimar $p(x|y)$ usando un histograma d -dimensional con B “bins” por dimensión $1, 2, \dots, d$.

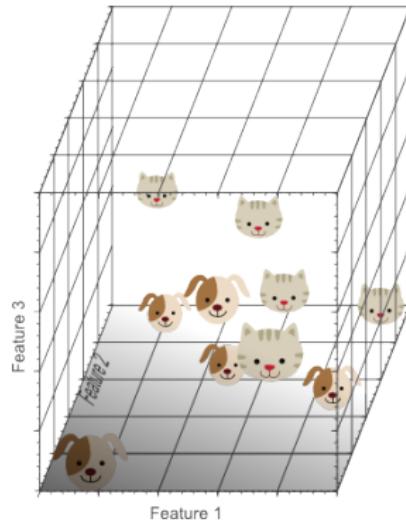
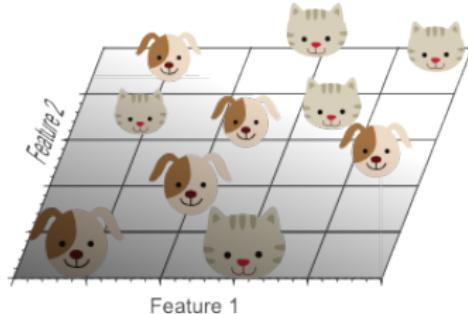


- Si en cada dimensión tenemos B “bins”, en d dimensiones, tenemos B^d diferentes bins!



Maldición de la Dimensionalidad

- Si para estimar $p(x_i \in B_{ik})$ (con B_{ik} el bin k de la dimensión i) requerimos $\mathcal{O}(m)$ datos, para estimar las probabilidades correspondientes a los B^d bins, necesitamos $\mathcal{O}(B^d m)$ datos. Para $B = 10$, $d = 10$, $m = 50$, esto implica disponer de alrededor de 500.000 millones de ejemplos, por cada clase.



- Un modelo Bayesiano ingenuo aborda este problema considerando la siguiente simplificación

$$p(x|y) \approx p(x_1|y) \cdot p(x_2|y) \cdots p(x_d|y), \quad (2)$$

que permite estimar $p(x_i|y)$ independientemente de $p(x_j|y)$, $\forall j \neq i$.

- Si para estimar $p(x_i \in B_{ik})$ requerimos $\mathcal{O}(m)$ datos, para estimar el histograma correspondiente a una dimensión necesitamos $\mathcal{O}(Bm)$ datos. Gracias al supuesto ingenuo, podemos estimar los d histogramas con $\mathcal{O}(dBm)$ datos en vez de $\mathcal{O}(B^d m)$. En el ejemplo anterior: 5000 ejemplos, en vez de 500.000 millones.
- Técnicamente, el supuesto de la Eqn. (2) corresponde a asumir independencia condicional de los atributos a la clase de cada dato.



- En la práctica, el supuesto Bayesiano ingenuo suele aplicarse al caso de atributos categóricos, i.e. $x_i \in \{v_{i1}, v_{i2}, \dots, v_{iB}\}$, de modo que para cada clase y cada dimensión x_i debemos estimar $B - 1$ probabilidades correspondientes a

$$p_{ij|k} = P(x_i = v_{ij} | y = c_k), j = 1, 2, \dots, B - 1. \quad (3)$$

En adelante, el término clasificador Bayesiano ingenuo (NBC) se reservará a este caso.

- Para estimar $p_{ij|k}$ podríamos considerar los estimadores “frequentistas”

$$\phi_{ij|k} = \frac{n_{ijk}}{n_k}, \quad (4)$$

donde n_{ijk} es el número de datos de la clase c_k para los cuales el atributo x_i toma el valor v_j . La LGN garantiza la consistencia de estos estimadores.



Entrenamiento del Clasificador Bayesiano Ingenuo

- Es fácil mostrar los estimadores que hemos definido ($\Theta = \{\theta_k, \phi_{ij|k}\}_{ijk}$) corresponden a los estimadores máximo verosímiles del modelo, es decir, aquellos que resultan si entrenamos el modelo para maximizar la log-verosimilitud condicional.
- En efecto, si $\underline{X} = \{x^{(\ell)}\}_{i=1}^n$ y $\underline{Y} = \{y^{(\ell)}\}_{i=1}^n$, tenemos que

$$\begin{aligned}\ell(\Theta) &= \log P(\underline{Y}|\underline{X}, \Theta) = \sum_{\ell} \log p(y^{(\ell)}|x^{(\ell)}, \Theta) \\ &= \sum_{\ell} \log p(x^{(\ell)}|y^{(\ell)}, \Theta) + \log p(y^{(\ell)}|\Theta) + \text{const.},\end{aligned}\tag{5}$$

desde donde, usando el supuesto ingenuo, obtenemos

$$\begin{aligned}\ell(\Theta) &= \sum_{\ell} \sum_i \log p(x_i^{(\ell)}|y^{(\ell)}, \Theta) + \log p(y^{(\ell)}|\Theta) + \text{const.} \\ &= \sum_{\ell} \sum_i \sum_k \log \phi_{ij|k} I(y^{(\ell)} = c_k, x_i^{(\ell)} = v_j) + \theta_k I(y^{(\ell)} = c_k) + \text{const.},\end{aligned}\tag{6}$$

- Recordando que $\phi_{iB|k} = 1 - \phi_{i1|k} - \phi_{i2|k} - \cdots - \phi_{iB-1|k}$, y que $\theta_K = 1 - \theta_1 - \theta_2 - \cdots - \theta_{K-1}$, podemos obtener las derivadas de $\ell(\Theta)$ con respecto a los parámetros libres.



Entrenamiento del Clasificador Bayesiano Ingenuo

- En efecto, $\forall j = 1, \dots, B - 1$,

$$\begin{aligned}\frac{\partial \ell}{\partial \phi_{ij|k}} &= \sum_{\ell} \frac{I(y^{(\ell)} = c_k, x_i^{(\ell)} = v_j)}{\phi_{ij|k}} - \sum_{\ell} \frac{I(y^{(\ell)} = c_K, x_i^{(\ell)} = v_B)}{\phi_{iB|k}} \\ &= \frac{n_{ijk}}{\phi_{ij|k}} - \frac{n_{iBk}}{\phi_{iB|k}},\end{aligned}\tag{7}$$

- Trabajando sobre condición para un punto crítico, obtenemos

$$\begin{aligned}n_{ijk}\phi_{iB|k} &= \phi_{ij|k} n_{iBk} \Rightarrow \sum_{j=1}^{B-1} n_{ijk}\phi_{iB|k} = \sum_{j=1}^{B-1} \phi_{ij|k} n_{iBk} \\ &\Leftrightarrow (n_k - n_{iBk})\phi_{iB|k} = (1 - \phi_{iB|k})n_{iBk} \\ &\Leftrightarrow n_k\phi_{iB|k} = n_{iBk} \Leftrightarrow \phi_{iB|k} = n_{iBk}/n_k,\end{aligned}\tag{8}$$

- Sustituyendo en (7), obtenemos que, $\forall j = 1, \dots, B - 1$,

$$\phi_{ij|k} = n_{ijk}/n_k.\tag{9}$$

- Con un procedimiento análogo obtenemos,

$$\theta_k = n_k/n.\tag{10}$$



- Una vez que hemos entrenado el clasificador, la clasificación de un nuevo dato x con atributos $(v_{1j_1}, v_{1j_2}, \dots, v_{1j_B})$ se realiza calculando $p(y = c_k|x)$ y usando la regla de decisión ya mencionada

$$\begin{aligned}\arg \max_k p(y = c_k|x) &= \arg \max_j p(x|y = c_k)p(y = c_k) \\ &= \arg \max_k \theta_k \prod_i \phi_{ij_i|k}.\end{aligned}\tag{11}$$

- Por razones numéricas es conveniente modificar la regla anterior de la siguiente forma

$$\begin{aligned}\arg \max_k p(y = c_k|x) &= \arg \max_j p(x|y = c_k)p(y = c_k) \\ &= \arg \max_k \theta_k \prod_i \phi_{ij_i|k} \\ &= \arg \max_k \log \theta_k + \sum_i \log \phi_{ij_i|k}.\end{aligned}\tag{12}$$



- Una vez que hemos entrenado el clasificador, la clasificación de un nuevo dato x con atributos $(v_{1j_1}, v_{1j_2}, \dots, v_{1j_B})$ se realiza calculando $p(y = c_k|x)$ y usando la regla de decisión ya mencionada
- Notemos también que situaciones con muchos atributos y/o pocos datos, pueden existir muchos $\phi_{ij_i|k}$ nulos que producen $p(y = c_k|x) = 0$.
- Para manejar esta situación es conveniente aplicar la denominada **corección de Laplace** de los estimadores máximo verosímiles

$$\phi_{ij_i|k} = (n_{ijk} + 1)/(n_k + B) \quad (13)$$

$$\theta_k = (n_k + 1)/(n + K).$$

¿Porqué en el primer caso el denominador suma B (y no 1)?

¿Porqué en el primer caso el denominador suma K (y no 1)?



- Un escenario en que el NBC es muy efectivo es la clasificación de texto.
- En ese caso, el texto se puede representar como un vector (x_1, x_2, \dots, x_d) donde x_i indica la presencia o ausencia de una palabra w_i del "diccionario" de términos utilizado (es decir $B = 2$). En este caso $\phi_{i|k} = \phi_{i1|k}$ es la prob. de que la palabra w_i aparezca en un documento de la clase k .
- En la literatura especializada, este enfoque se denomina *Bernoulli Naive Bayes* (BNBC).
- Una desventaja de BNBC es que resulta muy sensible al largo de los documentos, ya que el modelo considera como aspectos discriminativos tanto la presencia como la ausencia de la palabra. Por otro lado, el número de veces que aparece una palabra es irrelevante mientras éste sea > 1 .
- Una variante de NBC específicamente pensada para análisis de texto, se denomina *Multinomial Naive Bayes* (MNBC).



- En MNBC, x se modela como una secuencia de palabras $x_1 x_2 \dots x_{n_x}$ de largo variable n_x en vez de un vector de largo fijo d . El principio ingenuo se aplica del siguiente modo

$$p(x|y) = p(x_1|y) \cdot p(x_2|y) \cdots p(x_{n_x}|y). \quad (14)$$

- Los parámetros del modelo siguen siendo las probabilidades $(\phi_{i|k})$ de que cada palabra (w_i) aparezca en un documento de la clase k . Sin embargo, la ausencia de un término no influye (explícitamente) en el cálculo de $p(x|y)$. Además, el número de veces que aparece una palabra influye en el cálculo de $p(x|y)$.
- Si x_i corresponde a la palabra w_{j_i} del diccionario,

$$p(x|y) = \phi_{i_1|k} \cdot \phi_{i_2|k} \cdots \phi_{i_{n_x}|k} \quad (15)$$



- El enfoque anterior cambia también el resultado del entrenamiento. En efecto, puede mostrarse (tarea) que los estimadores máximo verosímiles en este caso vienen dados por

$$\phi_{i|k} = \tilde{n}_{ik}/\tilde{n}_k , \quad (16)$$

donde \tilde{n}_{ik} es el número total de veces que la palabra w_i aparece en los documentos de la clase k (considerando repeticiones) y \tilde{n}_k es la suma de los largos de los documentos de la clase k .

- En BNBC en cambio se tiene que

$$\phi_{i|k} = n_{ik}/n_k , \quad (17)$$

con n_{ik} el número de documentos de la clase k en que la palabra w_i aparece y n_k es el número de documentos de la clase k .



Análisis de Discriminantes Gausianos (GDA)

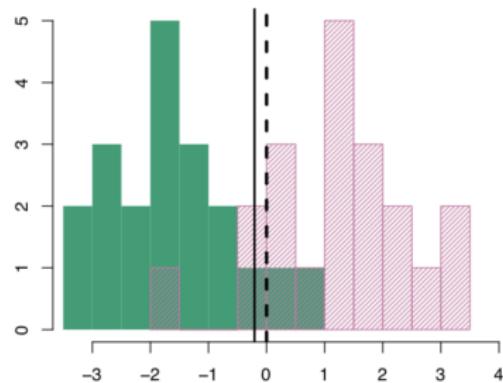
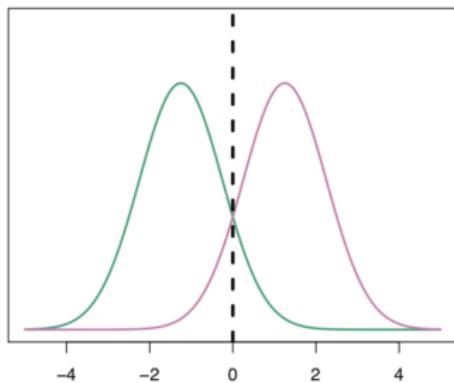


Idea General

- El análisis de discriminantes gausianos (GDA) aparece cuando, en la descomposición generativa

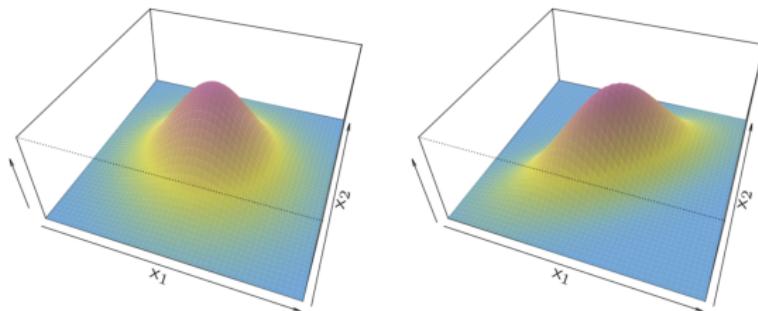
$$p(y|x) = \frac{p(x|y)p(y)}{p(x)} \propto p(x|y)p(y),$$

se decide modelar $p(x|y)$ usando una distribución normal $\mathcal{N}(\mu_y, \Sigma_y)$ (potencialmente diferente para cada clase).



Gaussianas Hyper-dimensionales

- Como, en general, $x \in \mathbb{R}^d$, con $d \gg 1$, tendremos que recurrir a gausianas multi-dimensionales



- Se dice que $z \in \mathbb{R}^d$ sigue una distribución normal $\mathcal{N}(\mu, \Sigma)$ ssi

$$p(z) = \frac{1}{\sqrt{2\pi|\Sigma|}} \exp\left(-\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu)\right),$$

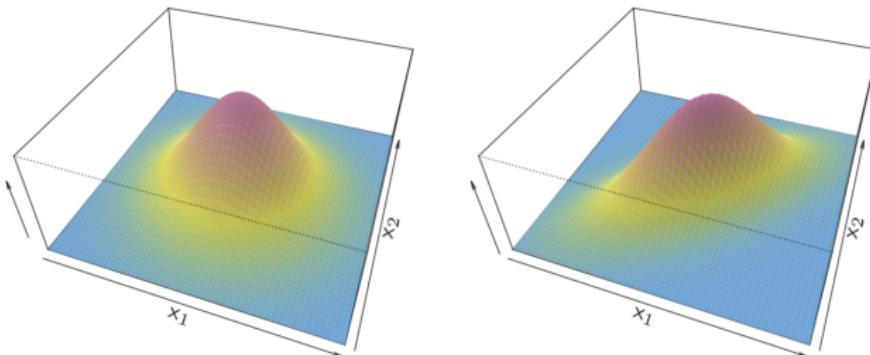
con $\mu \in \mathbb{R}^d$ y $\Sigma \in \mathbb{R}^{d \times d}$ $\succ 0$.



Gaussianas Hyper-dimensionales

- Como, en el caso unidimensional, μ establece la posición del centro (media, moda y mediana) de la gausiana en el espacio,

$$\mathbb{E}(z) = \mu \Leftrightarrow \mu_i = \mathbb{E}(z_i) \forall i.$$

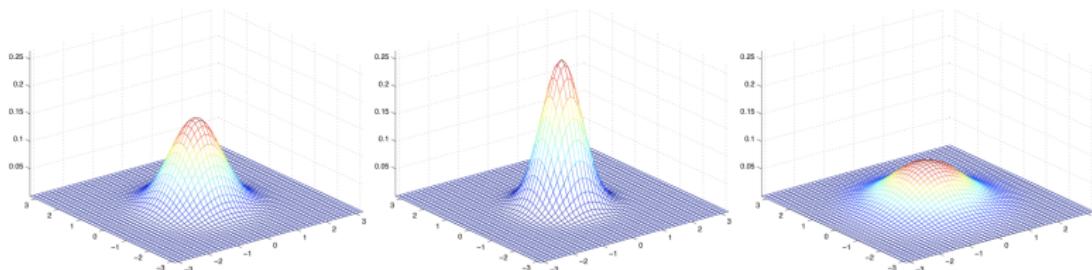


Gaussianas Hyper-dimensionales

- Como, en el caso unidimensional, la magnitud de Σ define el nivel de dispersión en torno a la media. Sin embargo, en d dimensiones, Σ determina también las correlaciones entre los distintos atributos que describen z . En efecto,

$$\begin{aligned}\Sigma &= \mathbb{E} \left((z - \mu)(z - \mu)^T \right) \Leftrightarrow \Sigma_{ij} = \mathbb{E} ((z_i - \mu_i)(z_j - \mu_j)) \\ &= \mathbb{E} ((z_i - \mathbb{E}[z_i])(z_j - \mathbb{E}[z_j])) ,\end{aligned}$$

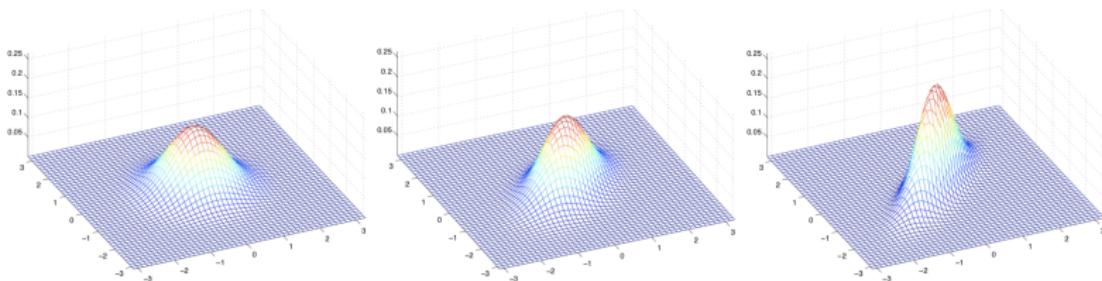
razón por la cual Σ se denomina **la matriz de covarianza**.



Gaussianas Hyper-dimensionales

- En efecto, la matriz de covarianza Σ determina también la orientación y asimetría de la gaussiana. Concretamente, la matriz de vectores propios de Σ corresponde a la solución del problema,

$$\max_{V \in \mathbb{R}^{d \times d}} \mathbb{E} \left(\|V^T x\| \right) \text{ s.t. } V^T V = I$$



Modelando con Gausianas

- El análisis de discriminantes gausianos (GDA) aparece entonces cuando se impone

$$p(x|y) = \frac{1}{\sqrt{2\pi|\Sigma_y|}} \exp\left(-\frac{1}{2} (x - \mu_y)^T \Sigma_y^{-1} (x - \mu_y)\right),$$

- Más explícitamente,

$$p(x|y = c_1) = \frac{1}{\sqrt{2\pi|\Sigma_1|}} \exp\left(-\frac{1}{2} (x - \mu_1)^T \Sigma_1^{-1} (x - \mu_1)\right)$$

$$p(x|y = c_2) = \frac{1}{\sqrt{2\pi|\Sigma_2|}} \exp\left(-\frac{1}{2} (x - \mu_2)^T \Sigma_2^{-1} (x - \mu_2)\right)$$

...

$$p(x|y = c_K) = \frac{1}{\sqrt{2\pi|\Sigma_K|}} \exp\left(-\frac{1}{2} (x - \mu_K)^T \Sigma_K^{-1} (x - \mu_K)\right).$$

- $p(y)$ se sigue modelando como antes, es decir, mediante parámetros $\theta_k = p(y = c_k)$.



- Como antes, los parámetros del modelo $\Theta = \{\mu_k, \Sigma_k, \theta_k\}$ pueden ajustarse para maximizar la log-verosimilitud de los datos.

$$\ell(\Theta) = \sum_{\ell} \log p(y^{(\ell)} | x^{(\ell)}) = \sum_{\ell} \log p(x^{(\ell)} | y^{(\ell)}) + \log p(y^{(\ell)}) + \text{const} \quad (18)$$

$$= \sum_k \sum_{\ell: y^{(\ell)}=k} \left(-\frac{1}{2} (x^{(\ell)} - \mu_k)^T \Sigma_k^{-1} (x^{(\ell)} - \mu_k) \right) - \log(|\Sigma_k|) + \log \theta_k + \text{const}, \quad (19)$$

- Este método lleva a las siguientes soluciones

$$\mu_k = \frac{1}{n_k} \sum_{\ell: y^{(\ell)}=k} x^{(\ell)}, \quad \Sigma_k = \frac{1}{n_k} \sum_{\ell: y^{(\ell)}=k} (x^{(\ell)} - \mu_k)(x^{(\ell)} - \mu_k)^T. \quad (20)$$



- La función de decisión para un dato nuevo x es la misma de antes,

$$\begin{aligned}\arg \max_k p(y = c_k | x) &= \arg \max_j p(x | y = c_k) p(y = c_k) \\&= \arg \max_k \theta_k \frac{1}{\sqrt{2\pi|\Sigma_k|}} \exp\left(-\frac{1}{2} (x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k)\right) \\&= \arg \max_k \log(\theta_k) - \left(\frac{1}{2} (x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k)\right) - \log(|\Sigma_k|).\end{aligned}\tag{21}$$

- Notemos que para dos clases i, j , $p(y = c_i | x) = p(y = c_j | x)$ ssi
 $\log p(y = c_i | x) - \log p(y = c_j | x) = 0$, es decir,ssi

$$\begin{aligned}\log(\theta_i) - \left(\frac{1}{2} (x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i)\right) - \log(|\Sigma_i|) \\- \log(\theta_j) + \left(\frac{1}{2} (x - \mu_j)^T \Sigma_j^{-1} (x - \mu_j)\right) + \log(|\Sigma_j|) = 0\end{aligned}$$

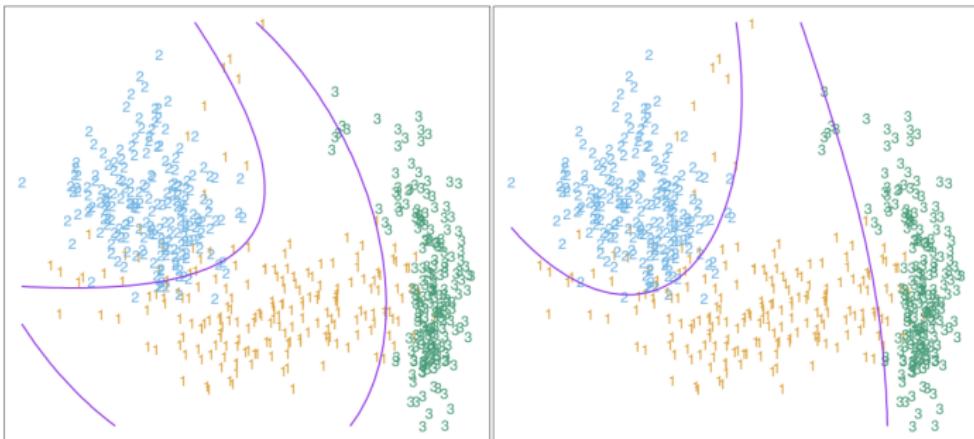


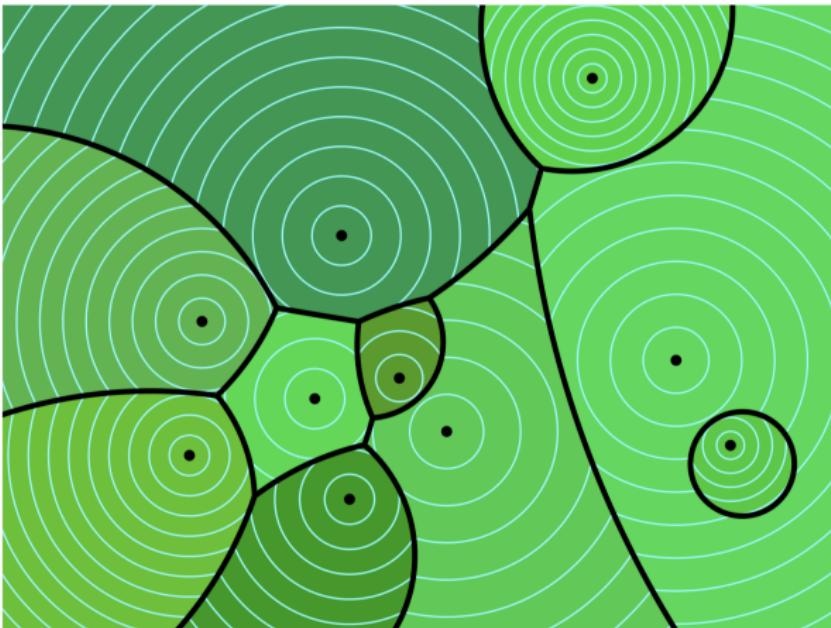
Función de Decisión y Fronteras

- La anterior, es una forma de segundo orden en x ,

$$x^T A x + x^T b + c = 0$$

lo que demuestra que las fronteras de decisión de GDA son cuadráticas.





- El número de parámetros libres de este modelo es $Kd^2 + Kd$, lo que sugiere que es necesario tener un número cuadrático de ejemplos para “aprender bien” el modelo.
- Es posible mostrar - ver (Ashtiani, 2018) - que estimar una gausiana en d -dimensiones con precisión ϵ requiere alrededor de $\Omega(d^2/\epsilon^2)$ datos. La complejidad viene de la necesidad de estimar Σ .
- En aplicaciones con pocos ejemplos y/o muchos atributos esto puede ser demasiado. Por ejemplo, si $d = 1000$ (e.g. imagen de 32×32), se necesitarían del orden de 1 millón de datos, por cada clase.
- **Idea:** restringir la “libertad” de las matrices Σ_k . Aparecen así 3 variantes,
 - **QDA.** No simplificar el modelo.
 - **LDA.** Asumir que $\Sigma_k = \Sigma \forall k$.
 - **NB LDA.** Asumir que $\Sigma_k = \text{diag}(\sigma_1^2, \sigma_2^2, \dots, \sigma_d^2) \forall k$.



Función de Decisión y Fronteras en LDA

- La simplificación adoptada por LDA reduce la complejidad de la función de decisión y de las fronteras

$$\arg \max_k p(y = c_k | x) = \arg \max_k \log(\theta_k) - \left(\frac{1}{2} (x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k) \right) - \log(|\Sigma_k|) \quad (22)$$

$$\arg \max_k p(y = c_k | x) = \arg \max_k \log(\theta_k) + \left(\mu_k^T \Sigma^{-1} x - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k \right)$$

$$\arg \max_k p(y = c_k | x) = \arg \max_k w_k^T x - b_k,$$

$$\text{con } w_k = \mu_k^T \Sigma^{-1} \text{ y } b_k = \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k - \log(\theta_k).$$

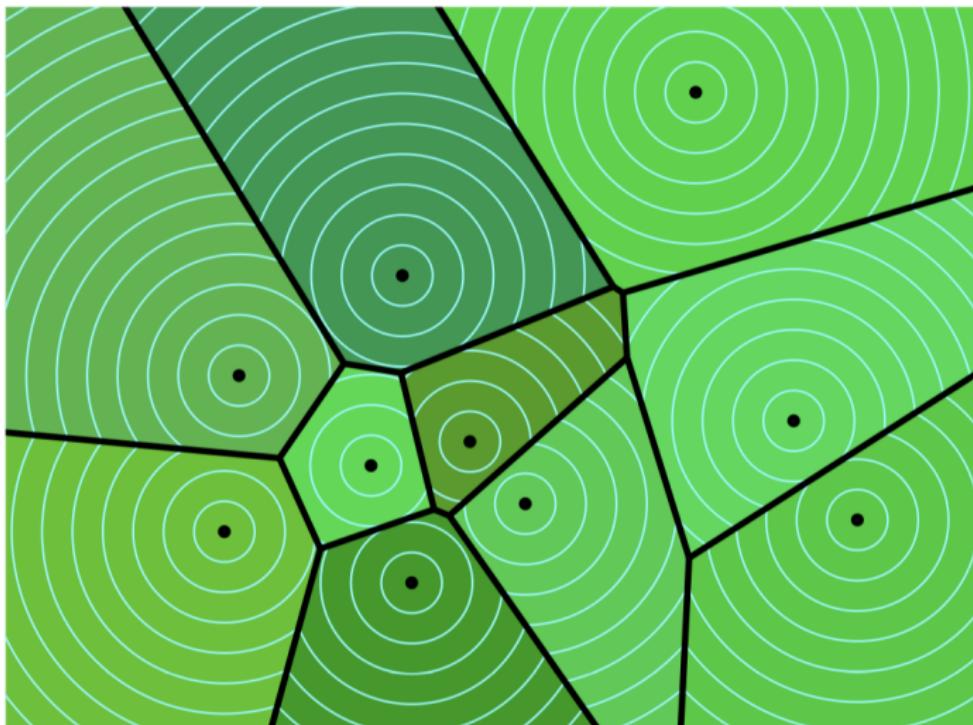
- La ecuación de la frontera entre las clases i, j toma entonces la forma

$$\begin{aligned} w_i^T x - b_i &= w_j^T x - b_j \\ \Leftrightarrow (w_i - w_j)^T x - (b_i - b_j) &= 0 \\ \Leftrightarrow w_{ij}^T x - b_{ij} &= 0 \end{aligned}$$



Función de Decisión y Fronteras en LDA

Lo anterior demuestra que las fronteras de decisión de LDA son lineales



Interpretación Geométrica

