CARATULA

Resumen

todo

Agradecimientos

todo

Índice general

1

Probabilidad y Estadística

Las expectativas sobre los objetivos que la inteligencia artificial podrá alcanzar en los próximos años tienden a ser muy ambiciosos. Avances constantes y no circunstanciales, solo serán posibles con el entendimiento absoluto en la materia, alejándose del enfoque de prueba y error.

El boom de la *inteligencia artificial* llegó para quedarse. Cada vez más, las decisiones asociadas a actividades comerciales y/o tecnológicas son tomadas en base a resultados algorítmicos. Dichas decisiones, lejos de basarse en reglas rígidas creadas por programadores, son tomadas en base al reconocimiento de patrones observados en experiencias estadísticas previas, definiendo lo que se conoce como el *aprendizaje estadístico*. Aunque la inteligencia artificial nos ha llevado a conseguir logros notables en las últimas décadas, las expectativas por lo que este campo será capaz de alcanzar en los próximos años tienden a exagerarse más de lo que podría ser posible. Para evitar todo tipo de decepción, es imprescindible entonces lograr avances permanentes que eviten todo tipo de estancamiento. Avances constantes y no circunstanciales, solo serán posibles con el entendimiento absoluto en la materia, alejándose del enfoque de *prueba y error*. El estudio de la matemática detrás de estos métodos es indispensable para transitar este camino. Quizás la única manera de comprender a la máquina sea con matemática.

1.1. Teoría de Probabilidad

La probabilidad es una teoría matemática que estudia el azar y la incertidumbre por medio de experimentos aleatorios [1]. La misma, se desarrolla en un marco conocido como **espacio de probabilidad**. Un espacio de probabilidad $(\Omega, \mathcal{A}, \mathbb{P})$ consta de una terna formada por un **espacio muestral** Ω que posee todos los resultados posibles de un experimento aleatorio, una **sigma-álgebra** $\mathcal{A}$ que contiene todos los conjuntos medibles, y una **medida de probabilidad** $\mathbb{P}$ que justamente mide que tan probable es un evento; cada uno con ciertas características. Las dos fórmulas más características para una $\mathbb{P} : \mathcal{A} \to [0, 1]$ son la **fórmula de probabilidades totales** y la **probabilidad condicional**.

Propiedades 1.1 *1. Sean $A_1, \cdots A_m \in \mathcal{A}$ una partición de Ω (es decir $A_i \cap A_j = \emptyset \quad \forall\, i \neq j$ y $\bigcup_{i=1}^m A_i = \Omega$), entonces para todo $B \in \mathcal{A}$:*

$$\mathbb{P}(B) = \sum_{i=1}^{m} \mathbb{P}(A_i \cap B) \tag{1.1}$$

2. Sea $B \in \mathcal{A}$ un evento con $\mathbb{P}(B) > 0$, entonces

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)} \tag{1.2}$$

Combinando estas dos fórmulas surge la famosa **regla de bayes**:

$$\mathbb{P}(A_k|B) = \frac{\mathbb{P}(B|A_k)\mathbb{P}(A_k)}{\sum_{i=1}^{m} \mathbb{P}(B|A_i)\mathbb{P}(A_i)} \tag{1.3}$$

Un concepto relacionado con estas expresiones matemáticas es lo que se conoce como **independencia estadística**.

Definición 1.1 *Dos eventos $A, B \in \mathcal{A}$ son independientes si $\mathbb{P}(A \cap B) = \mathbb{P}(A)\mathbb{P}(B)$.*

1.1.1. Variables Aleatorias

Distribución	Notación	$P_X(x)$	Soporte	$\mathbb{E}[X]$	var(X)
Bernoulli	Ber(p)	$p^x(1-p)^{1-x}$	$\{0,1\}$	p	$p(1-p)$
Binomial	Bin(n,p)	$\binom{n}{x}p^x(1-p)^{n-x}$	$\{0,\cdots,n\}$	np	$np(1-p)$
Geométrica	Geo(p)	$(1-p)^{x-1}p$	$\mathbb{N}$	$\frac{1}{p}$	$\frac{1-p}{p^2}$
Pascal	Pas(k,p)	$\binom{x-1}{k-1}(1-p)^{x-k}p^k$	$\{k, k+1, \cdots\}$	$\frac{k}{p}$	$k\frac{1-p}{p^2}$
Poisson	Poi(μ)	$\frac{\mu^x e^{-\mu}}{x!}$	$\mathbb{N}_0$	μ	μ

Cuadro 1.1: Algunas de las variables discretas más habituales.

Trabajar con eventos genéricos puede ser tedioso. Las **variables aleatorias** son funciones $X : \Omega \to \mathbb{R}$ que permiten codificar los resultados de un experimento aleatorio en valores numéricos. La medida de probabilidad sobre una variable aleatoria se caracteriza con la **función de distribución** $F_X(x) = \mathbb{P}(X \leq x)$. Se denomina **átomo** a todo $x \in \mathbb{R}$ tal que $\mathbb{P}(X = x) > 0$ (discontinuidades de la función de distribución). Si la suma de las probabilidades de los átomos vale 1 la variable se denomina discreta. Si en cambio, la función de distribución es continua, la variable se denomina continua. En caso de no ser ni continua ni discreta se denomina variable mixta.

El uso de la función de distribución para efectuar cálculos sigue siendo un poco incómodo, por eso surgen la **función de masa de probabilidad** $P_X(x) = \mathbb{P}(X = x)$ para

variables discretas y la **función de densidad de probabilidad** $p_X(x) = F'_X(x)$ para las variables continuas[1]. Cuando se habla de **distribución** a secas, se hace referencia a la medida de probabilidad asociada, siendo indiferente si la variable se representa con la función de distribución, la masa de probabilidad o la densidad. Cuando se quiere hablar en general, y no se sabe si la variable es discreta o continua, se usará la notación p_X la cuál debe interpretarse como masa o densidad según corresponda. En este sentido, $p_X(x)$ debe ser una función no negativa que integre 1[2]. Se denomina **soporte** a $\mathcal{X} = \{x \in \mathbb{R} : \ p_X(x) > 0\}$[3]. Algunas de las variables más conocidas pueden verse en el Cuadro 1.1 para el caso de las variables aleatorias discretas y en el Cuadro 1.2 para las variables continuas[4].

Distribución	Notación	$p_X(x)$	Soporte	$\mathbb{E}[X]$	$\text{var}(X)$
Uniforme	$\mathcal{U}(a,b)$	$\frac{1}{b-a}$	$[a,b]$	$\frac{a+b}{2}$	$\frac{(b-a)^2}{12}$
Normal	$\mathcal{N}(\mu,\sigma^2)$	$\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$	$\mathbb{R}$	μ	σ^2
Exponencial	$\exp(\lambda)$	$\lambda e^{-\lambda x}$	$[0,\infty)$	$\frac{1}{\lambda}$	$\frac{1}{\lambda^2}$
Gamma	$\Gamma(\nu,\beta)$	$\frac{\beta^\nu}{\Gamma(\nu)}x^{\nu-1}e^{-\beta x}$	$[0,\infty)$	$\frac{\nu}{\beta}$	$\frac{\nu}{\beta^2}$
Beta	$\beta(a,b)$	$\frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}x^{a-1}(1-x)^{b-1}$	$[0,1]$	$\frac{a}{a+b}$	$\frac{ab}{(a+b)^2(a+b+1)}$
Chi cuadrado	χ^2_k	$\frac{1}{2^{\frac{k}{2}}\Gamma\left(\frac{k}{2}\right)}x^{\frac{k}{2}-1}e^{-\frac{x}{2}}$	$[0,\infty)$	k	$2k$
t-student	t_ν	$\frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\Gamma\left(\frac{\nu}{2}\right)}\left(1+\frac{t^2}{\nu}\right)^{-\frac{\nu+1}{2}}$	$\mathbb{R}$	0	$\frac{\nu}{\nu-2}$
Lomax	$\text{Lomax}(\alpha,\beta)$	$\frac{\alpha\beta^\alpha}{(x+\beta)^{\alpha+1}}$	$[0,\infty)$	$\frac{\beta}{\alpha-1}$	$\frac{\beta^2\alpha}{(\alpha-1)^2(\alpha-2)}$

Cuadro 1.2: Algunas de las variables discretas más habituales.

Así como existen las variables aleatorias, se pueden definir los vectores aleatorios. En este contexto, las distribuciones involucradas reciben el nombre de conjunta (por ejemplo $p_{XY}(x,y)$), marginal ($p_X(x)$ y $p_Y(y)$) y condicional ($p_{X|Y=y}(x)$ y $p_{Y|X=x}(y)$). Las Props. 1.1 tendrán su contraparte con las funciones masa o las densidades.

Propiedades 1.2 *1. Las distribuciones marginales pueden calcularse como*

$$p_Y(y) = \int_{\mathcal{X}} p_{XY}(x,y)dx \tag{1.4}$$

[1]Siendo rigurosos, las variables con densidad son un subgrupo dentro de las variables continuas llamadas *absolutamente continuas*.

[2]En el caso de variables discretas, deben intercambiarse las integrales asociadas a dichas variables por sumas o series según corresponda.

[3]Siendo rigurosos, el soporte es la *clausura* de este conjunto.

[4]La varianza de la t-student solo existe para $\nu > 2$. La media de la Lomax solo existe para $\alpha > 1$ y su varianza para $\alpha > 2$.

2. Las distribuciones condicionales pueden calcularse como

$$p_{Y|X=x}(y) = \frac{p_{XY}(x,y)}{p_X(x)} \tag{1.5}$$

De igual manera, dos variables aleatorias (X, Y) son independientes si y solo si su distribución conjunta se puede factorizar como $p_{XY}(x,y) = p_X(x)p_Y(y)$. Algunos de los vectores aleatorios más conocidos pueden verse a continuación.

Definición 1.2 *El vector aleatorio $X = (X_1, \cdots, X_d)$ tiene distribución normal multivariada $X \sim \mathcal{N}(\mu, \Sigma)$ si su densidad conjunta es de la forma:*

$$p(X_1, \cdots, X_d) = \frac{1}{(2\pi)^{\frac{d}{2}}\sqrt{|\Sigma|}} e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)} \tag{1.6}$$

De igual manera, el vector aleatorio $X = (X_1, \cdots, X_d)$ tiene distribución multinomial $X \sim \mathcal{M}_n(n, [p_1, \cdots, p_d])$ si su función de probabilidad conjunta es de la forma:

$$P(X_1, \cdots, X_d) = \frac{n!}{\prod_{i=1}^d x_i!} \left(\prod_{i=1}^d p_i^{x_i}\right) \mathbf{1}\left\{\sum_{i=1}^d x_i = n, x \in \mathbb{N}_0^d\right\} \tag{1.7}$$

La probabilidad conjunta, siguiendo (1.5), siempre se puede factorizar de la forma $p_{XY}(x,y) = p_{Y|X=x}(y)p_X(x)$. Dado que en probabilidad este tipo de descomposiciones es única, cuando la conjunta es factorizada inmediatamente se conoce la marginal y la condicional. El siguiente ejemplo muestra como se puede factorizar analíticamente este tipo de conjunta.

Ejemplo 1.1 *Sea $p_{XY}(x,y) = e^{-x}\mathbf{1}\{0 < y < x\}$, hallar $p_X(x)$ y $p_{Y|X=x}(y)$.*

En la factorización $p_{XY}(x,y) = p_{Y|X=x}(y)p_X(x)$, la variable y aparece solamente en una de las distribuciones, por lo que el primer paso es separar todos los lugares donde aparezca dicha variable $p_{XY}(x,y) = \mathbf{1}\{0 < y < x\} \cdot e^{-x}\mathbf{1}\{x > 0\}$. Luego, completando la densidad condicional para que integren 1 en y (ayudándose con el Cuadro 1.2), puede verse que

$$p_{XY}(x,y) = \underbrace{\frac{1}{x}\mathbf{1}\left\{0 < y < x\right\}}_{p_{Y|X=x}(y)} \cdot \underbrace{xe^{-x}\mathbf{1}\left\{x > 0\right\}}_{p_X(x)} \tag{1.8}$$

En este caso, del Cuadro 1.2, se deduce que $Y|X = x \sim \mathcal{U}(0,x)$ y $X \sim \Gamma(2,1)$.

1.1.2. Momentos

En muchas circunstancias, tener toda una función que caracterice un experimento aleatorio es abrumador. Se denominan **momentos** a las magnitudes más relevantes que caracterizan a las variables aleatorias, caracterizadas por el operador esperanza $\mathbb{E}[\cdot]$. Los más representativos son la media $\mathbb{E}[X] = \int_{\mathcal{X}} x \cdot p_X(x)dx$, que representa a la variable aleatoria como una constante, y la varianza $\text{var}(X) = \mathbb{E}[(X - \mathbb{E}[X])^2]$, que representa el error que

se comente al aproximar a la variable aleatoria con su media. Algunas de las propiedades básicas de los momentos se presentan a continuación.

Propiedades 1.3 *Propiedades de la esperanza.*

1. *Cuando se aplican funciones sobre variables aleatorias, no es necesario conocer la distribución de la nueva variable* $\mathbb{E}[g(X)] = \int_{\mathcal{X}} g(x)p_X(x)dx$.
2. *La esperanza es lineal* $\mathbb{E}[aX + b] = a \cdot \mathbb{E}[X] + b$.
3. *Las probabilidades son un caso particular de las esperanzas* $\mathbb{E}[\mathbf{1}\{X \in A\}] = \mathbb{P}(X \in A)$.
4. *La varianza puede simplificar su cálculo como* $var(X) = \mathbb{E}[X^2] - \mathbb{E}[X]^2$.
5. *El comportamiento de la varianza frente a transformaciones afines es* $var(aX + b) = a^2 \cdot var(X)$.

Además, el siguiente teorema muestra en que sentido la media es la mejor aproximación constante de una variable aleatoria y como la viarainza nos da una idea de su error. Se basa en utilizar como criterio el **error cuadrático medio**.

Propiedades 1.4 $\mathbb{E}[(Y - c)^2] \geq var(Y)$ *con igualdad si y solo si* $c = \mathbb{E}[Y]$.

Como demostración, notar que la función a minimizar se puede escribir como una parábola convexa en función de c (usando linealidad de la esperanza): $\mathbb{E}[Y^2] - 2c\mathbb{E}[Y] + c^2$. Con lo cuál, su vértice se alcanza en $c = \mathbb{E}[Y]$, y para dicho valor el mínimo vale $\mathbb{E}[(Y - \mathbb{E}[Y])^2] = \text{var}(Y)$.

Los vectores aleatorios también tienen sus momentos asociados, definidos a través del operador esperanza. El momento más representativo de un vector (X, Y) es su **covarianza** $\text{cov}(X, Y) = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])]$. La esperanza aplicadas sobre vectores aleatorios también tiene sus propiedades, algunas de las cuales pueden verse a continuación,

Propiedades 1.5

1. *Al igual que en el caso escalar, sigue valiendo que* $\mathbb{E}[g(X, Y)] = \int_{\mathcal{Y}} \int_{\mathcal{X}} g(x, y)p_{XY}(x, y)dxdy$.
2. *No es necesario conocer la distribución marginal para calcular la media* $\mathbb{E}[X] = \int_{\mathcal{Y}} \int_{\mathcal{X}} x \cdot p_{XY}(x, y)dxdy$.
3. *La linealidad sigue valiendo* $\mathbb{E}[aX + bY + c] = a \cdot \mathbb{E}[X] + b \cdot \mathbb{E}[Y] + c$
4. *La covarianza también puede simplificar su cálculo como* $cov(X, Y) = \mathbb{E}[XY] -$

$\mathbb{E}[X] \cdot \mathbb{E}[Y]$.

5. *La varainza es un caso particular de la covarianza* $cov(X, X) = var(X)$
6. *La covarianza es simétrica* $cov(X, Y) = cov(Y, X)$.
7. *La covarianza es bilineal* $cov(aX + bY + c, \alpha X + \beta Y + \gamma) = a \cdot \alpha \cdot var(X) + b \cdot \beta \cdot var(Y) + (b \cdot \alpha + a \cdot \beta) \cdot cov(X, Y)$.
8. *La varianza de una suma puede calcularse utilizando la covarianza* $var(X+Y) = var(X) + var(Y) + 2 \cdot cov(X, Y)$.

Así como la independencia factoriza las distribuciones, también factoriza las esperanzas. Pero la recíproca no es válida. Es decir, si dos variables X e Y son independientes, luego $\mathbb{E}[g(X)h(Y)] = \mathbb{E}[g(X)]\mathbb{E}[h(Y)]$ y por lo tanto $\text{cov}(X, Y) = 0$. Pero descorrelación (covarianza nula) no implica independencia.

Los momentos caracterizan las vectores aleatorios. Así como la Prop. 1.4 demuestra que la constante que mejor aproxima a la variable aleatoria en términos del error cuadrático medio es la media, la recta que mejor aproxima recibe el nombre de **recta de regresión**. La misma es definida a partir de los momentos como se muestra en el siguiente teorema.

Propiedades 1.6 $\mathbb{E}[(Y - (aX + b))^2] \geq var(Y) - \frac{cov(X,Y)^2}{var(X)}$ *con igualdad si y solo si*

$$aX + b = \frac{cov(X, Y)}{var(X)} (X - \mathbb{E}[X]) + \mathbb{E}[Y] \tag{1.9}$$

Para demostrarlo, notar que la función a minimizar es

$$\mathbb{E}[Y^2] + a^2\mathbb{E}[X^2] + b^2 - 2a\mathbb{E}[XY] - 2b\mathbb{E}[Y] + 2ab\mathbb{E}[X] \tag{1.10}$$

Para buscar el mínimo, se puede igualar a cero las derivadas respecto de a y de b:

$$2a\mathbb{E}[X^2] - 2\mathbb{E}[XY] + 2b\mathbb{E}[X] = 2b - 2\mathbb{E}[Y] + 2a\mathbb{E}[X] = 0 \tag{1.11}$$

Luego $b = \mathbb{E}[Y] - a\mathbb{E}[X]$ y reemplazando

$$a\mathbb{E}[X^2] - \mathbb{E}[XY] + \mathbb{E}[X]\mathbb{E}[Y] - a\mathbb{E}[X]^2 = 0 \quad \rightarrow \quad a = \frac{\text{cov}(X, Y)}{\text{var}(X)} \tag{1.12}$$

para esos valores el error cuadrático medio es

$$\mathbb{E}\left[\left(Y - \mathbb{E}[Y] - \frac{\text{cov}(X, Y)}{\text{var}(X)}(X - \mathbb{E}[X])\right)^2\right]$$

$$= \text{var}(Y) + \frac{\text{cov}(X, Y)^2}{\text{var}(X)} - 2\frac{\text{cov}(X, Y)^2}{\text{var}(X)} \tag{1.13}$$

$$= \text{var}(Y) - \frac{\text{cov}(X, Y)^2}{\text{var}(X)} \tag{1.14}$$

Los momentos condicionales llevan al máximo el potencial de la esperanza. Sea $\varphi(x) = \mathbb{E}[Y|X = x]$ la media de la distribución de $Y|_{X=x}$ (como función de x) y $\mathbb{E}[Y|X] = \varphi(X)$

a la variable aleatoria construida evaluando a $\varphi(x)$ en la variable aleatoria X. Algunas de las propiedades más importantes pueden verse a continuación.

Propiedades 1.7 *La esperanza y la varianza condicional poseen las siguientes propiedades.*

1. $\mathbb{E}[Y] = \mathbb{E}[\mathbb{E}[Y|X]]$
2. $\mathbb{E}[Yg(X)] = \mathbb{E}[\mathbb{E}[Y|X]g(X)]$
3. $\mathbb{P}(Y \in \mathcal{R}) = \mathbb{E}[\mathbb{P}(Y \in \mathcal{R}|X)]$
4. $var(Y) = \mathbb{E}[var(Y|X)] + var\,(\mathbb{E}[Y|X])$

La característica principal de la esperanza condicional es que es el mejor predictor a la hora de estimar una variable aleatoria como puede verse en el siguiente teorema.

Propiedades 1.8 $\mathbb{E}[(Y - \varphi(X))^2] \geq \mathbb{E}[var(Y|X)]$ *con igualdad si y solo si* $\varphi(x) = \mathbb{E}[Y|X = x]$.

La demostración es sencilla: sumando y restando la esperanza condicional en la expresión a minimizar se observa que,

$$\mathbb{E}[(Y - \varphi(X))^2] = \mathbb{E}[(Y - \mathbb{E}[Y|X] + \mathbb{E}[Y|X] - \varphi(X))^2] \tag{1.15}$$

$$= \mathbb{E}[(Y - \mathbb{E}[Y|X])^2] + \mathbb{E}[(\mathbb{E}[Y|X] - \varphi(X))^2] + 2\mathbb{E}[(Y - \mathbb{E}[Y|X])(\mathbb{E}[Y|X] - \varphi(X))] \tag{1.16}$$

El primer sumando puede simplificando utilizando las propiedades de la esperanza condicional $\mathbb{E}[(Y - \mathbb{E}[Y|X])^2] = \mathbb{E}[\mathbb{E}[(Y - \mathbb{E}[Y|X])^2|X]] = \mathbb{E}[\text{var}(Y|X)]$. El segundo sumando simplemente se acota con $\mathbb{E}[(\mathbb{E}[Y|X] - \varphi(X))^2] \geq 0$ y la igualdad se da si y solo si $\varphi(x) = \mathbb{E}[Y|X = x]$. El tercer sumando se anula usando las propiedades de la esperanza condicional $\mathbb{E}[(Y - \mathbb{E}[Y|X])(\mathbb{E}[Y|X] - \varphi(X))] = \mathbb{E}[\mathbb{E}[Y - \mathbb{E}[Y|X]|X](\mathbb{E}[Y|X] - \varphi(X))] = 0$. Juntando los tres términos el teorema es probado: $\mathbb{E}[(Y - \varphi(X))^2] \geq \mathbb{E}[\text{var}(Y|X)]$ con igualdad se da si y solo si $\varphi(x) = \mathbb{E}[Y|X = x]$.

Las propiedades 1.4, 1.6 y 1.8 son esenciales conceptualmente para la inteligencia artificial. Los resultados de estos teoremas implican que:

- En términos del error cuadrático medio, el mejor predictor es la esperanza condicional. Si nos restringimos a las rectas, el predictor óptimo es la recta de regresión. Si en cambio buscamos un predictor constante, la mejor opción es la esperanza.
- Si la esperanza condicional es una recta, necesariamente debe coincidir con la recta de regresión. Si la recta de regresión es una constante debe coincidir con la esperanza $\mathbb{E}[Y]$.

- En ningún caso, el error cuadrático medio puede ser inferior a $\mathbb{E}[\text{var}(Y|X)]$. Esto lo convierte en un límite fundamental.

A continuación se presentará un ejemplo de cómputo de este tipo de magnitudes.

Ejemplo 1.2 *En el mercado de smartphones, los dispositivos con mayor capacidad de almacenamiento suelen tener baterías más duraderas. Modelar estos datos podría ayudar a estimar la duración de la batería en función de su capacidad de almacenamiento, algo útil para los consumidores a la hora de elegir un nuevo dispositivo. Sea X la capacidad de almacenamiento de los smartphones (en TB) e Y la duración de su batería (en días), con densidad de probabilidad conjunta de la forma:*

$$p_{XY}(x,y) = \frac{3}{4} \cdot \mathbf{1}\left\{0 < y < 1 + x^2, 0 < x < 1\right\} \tag{1.17}$$

Calcular la duración media de las baterías, la recta de regresión y la esperanza condicional. Indicar cuál es el error cuadrático medio asociado a cada aproximación.

Lo primero a determinar es la factorización de la distribución en la condicional $Y|X = x$ (para caracterizar el comportamiento de Y como función de x) y la marginal X (para medir correctamente cuanto se penalizan los errores). Al igual que en el Ej. 1.1, se separará todos los factores donde aparezca y y se completará la densidad para que integre 1 (utilizando la lista de distribuciones conocidas del Cuadro 1.2).

$$p_{XY}(x,y) = \underbrace{\frac{1}{1+x^2}\mathbf{1}\left\{0 < y < 1 + x^2\right\}}_{p_{Y|X=x}(y)} \cdot \underbrace{\frac{3(1+x^2)}{4}\mathbf{1}\left\{0 < x < 1\right\}}_{p_X(x)} \tag{1.18}$$

donde puede verse $Y|_{X=x} \sim \mathcal{U}(0, 1 + x^2)$ y X es una variable aleatoria continua con densidad bien determinada, pero sin ser ninguna de las mencionadas en el Cuadro 1.2. La esperanza y varianza condicional pueden observarse en el mencionado cuadro y determinar que $\mathbb{E}[Y|X = x] = \frac{1+x^2}{2}$ y $\text{var}(Y|X = x) = \frac{(1+x^2)^2}{12}$. El resto del problema es simplemente calcular todos los momentos asociados al problema.

Los momentos de X pueden obtenerse simplemente integrando polinomios

$$\mathbb{E}[X^k] = \int_0^1 x^k \frac{3(1+x^2)}{4} dx = \frac{3}{4}\left(\frac{1}{k+1} + \frac{1}{k+3}\right) \tag{1.19}$$

de esta manera $\mathbb{E}[X] = \frac{9}{16}$, $\mathbb{E}[X^2] = \frac{2}{5}$, $\mathbb{E}[X^3] = \frac{5}{16}$ y $\mathbb{E}[X^4] = \frac{9}{35}$. La varianza será entonces $\text{var}(X) = \mathbb{E}[X^2] - \mathbb{E}[X]^2 = \frac{107}{1280}$. La media de Y se pueden calcular utilizando las propiedades de la esperanza condicional.

$$\mathbb{E}[Y] = \mathbb{E}[\mathbb{E}[Y|X]] = \frac{1 + \mathbb{E}[X^2]}{2} = \frac{7}{10} \tag{1.20}$$

Para el caso de la varianza, puede hacerse el mismo tipo de análisis con $\text{var}(Y) =$

$\mathbb{E}[\text{var}(Y|X)] + \text{var}(\mathbb{E}[Y|X])$. En este caso

$$\mathbb{E}[\text{var}(Y|X)] = \frac{1 + 2\mathbb{E}[X^2] + \mathbb{E}[X^4]}{12} = \frac{6}{35} \tag{1.21}$$

$$\text{var}(\mathbb{E}[Y|X]) = \frac{\text{var}(X^2)}{4} = \frac{\mathbb{E}[X^4] - \mathbb{E}[X^2]^2}{4} = \frac{17}{700} \tag{1.22}$$

y por lo tanto $\text{var}(Y) = \frac{137}{700}$.

Por el lado de la covarianza $\text{cov}(X, Y) = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]$, resta calcular la esperanza del producto. La misma puede resolverse utilizando las Prop 1.7.

$$\mathbb{E}[XY] = \mathbb{E}[X\mathbb{E}[Y|X]] = \frac{\mathbb{E}[X] + \mathbb{E}[X^3]}{2} = \frac{3}{16} \tag{1.23}$$

y por lo tanto $\text{cov}(X, Y) = \frac{7}{160}$. Finalmente utilicemos los resultados 1.4, 1.6 y 1.8, para analizar las diferentes aproximaciones en términos del error cuadrático medio.

- La constante que mejor aproxima a Y es su media $\mathbb{E}[Y] = \frac{7}{10}$, y el error cuadrático medio es $\text{var}(Y) = \frac{137}{700} \approx 0.196$.
- La recta que mejor aproxima a Y es la recta de regresión $\frac{\text{cov}(X,Y)}{\text{var}(X)}(x - \mathbb{E}[X]) + \mathbb{E}[Y] = \frac{56}{107}x + \frac{217}{535}$, y el error cuadrático medio es $\text{var}(Y) - \frac{\text{cov}(X,Y)^2}{\text{var}(X)} = \frac{3236}{18725} \approx 0.173$.
- La función que mejor aproxima a Y es la asociada a la esperanza condicional $\mathbb{E}[Y|X = x] = \frac{1+x^2}{2}$, y el error cuadrático medio es $\mathbb{E}[\text{var}(Y|X)] = \frac{6}{35} \approx 0.171$.

1.2. Estadística

En la mayoría de las aplicaciones, rara vez se conoce la distribución exacta de las variables aleatorias. La estadística soluciona este inconveniente por medio de datos. Si bien el aprendizaje estadístico busca reconocer patrones por medio de ejemplos, su análisis no debe reducirse al conjunto de datos con los que cuenta sino que dichos patrones deben poder generalizarse a nuevas muestras. En ese sentido, la inteligencia artificial es solamente una gran máquina de estadística con suficiente automatización, cálculo computacional y un poco de marketing. Pero no nos pisemos la manguera entre bomberos.

1.2.1. Distribución Empírica e Histograma

La primera pregunta que analizaremos en esta sección es como aproximar una distribución por medio de datos. Uno de los métodos más simples para efectuar dicha aproximación se conoce como **distribución empírica** [2, Capítulo 11]. La distribución empírica asume una distribución discreta, donde la probabilidad de cada átomo corresponde a su frecuencia de aparición $\hat{P}(x) = \frac{K}{n}$ donde n es la cantidad de muestras totales, K la cantidad

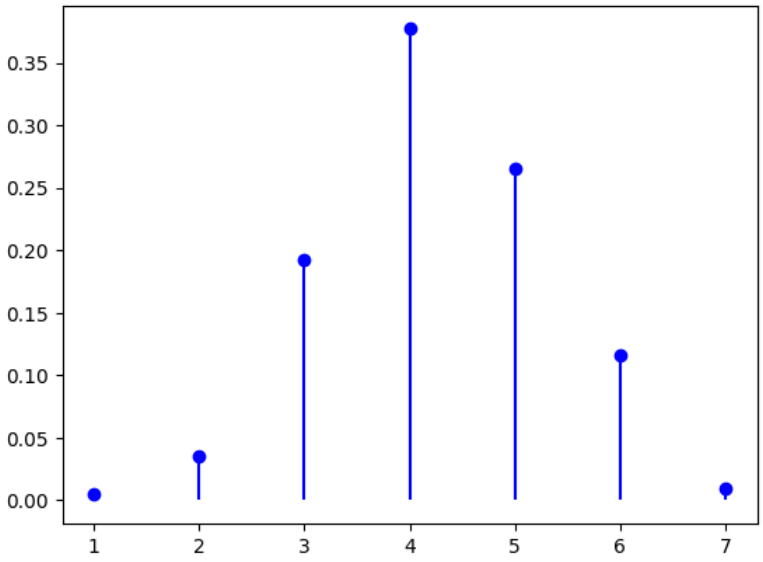


Figura 1.1: Ejemplo de función de masa de probabilidad empírica.

de muestras con valor x. Un ejemplo de función de masa de probabilidad empírica puede verse en la Fig. 1.1. Su función de distribución asociada será entonces del tipo *escalera*.

Otro enfoque para enfrentar esta problemática es la estimación de densidad por técnicas no paramétricas [3, Capítulo 4], siendo el **histograma** su variante más sencilla. El histograma modela la variable como continua y asume una densidad constante por regiones. En cada región asigna $\hat{p}(x) = \frac{K}{n \cdot V}$ donde n es la cantidad de muestras totales, K la cantidad de muestras en dicha región y V el volumen de la región (en el caso escalar el ancho del intervalo). Este método garantiza que la probabilidad de cada región sea proporcional a la cantidad de muestras que pertenecen a ella. Un ejemplo de función histograma puede verse en la Fig. 1.2.

1.2.2. Simulación

La generación de datos sintéticos es una parte esencial de la estadística. Ya sea para validar los modelos utilizados o para generar nueva información sobre una tarea, la simulación de datos es un área sumamente importante en la temática. En la práctica, cualquier software afín suele tener desarrollados algunos algoritmos generadores de números *pseudo-aleatorios*, pero es razonable que no cuente con todas las distribuciones necesarias que el usuario necesite. El ejemplo más simple es la $\mathcal{U}(0, 1)$, que cualquier calculadora científica puede simular. A partir de esos algoritmos, es necesario poder transformar las variables aleatorias generadas para que tengan cualquier distribución deseada. El siguiente teorema muestra algunos de los resultados más importantes en el tema.

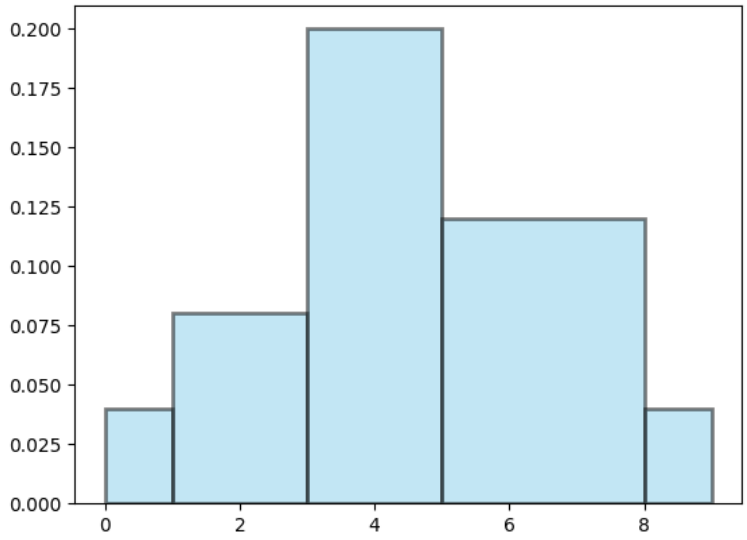


Figura 1.2: Ejemplo de función histograma (densidad de probabilidad).

Propiedades 1.9 *Para encontrar una transformación que satisfaga una determinada distribución pueden utilizarse los siguientes resultados*

1. *Sea* $U \sim \mathcal{U}(0,1)$*, luego la variable aleatoria* $Y = F_Y^{-1}(U)$ *posee función de distribución* F_Y*, donde* F_Y^{-1} *es la* inversa generalizada*:*
$$F_Y^{-1}(u) = \text{mín}\{y \in \mathbb{R} : \ u \leq F_Y(y)\} \tag{1.24}$$

2. *Toda variable aleatoria* X*, con función distribución estrictamente creciente en un intervalo, evaluada en su propia función de distribución* $U = F_X(X)$ *posee distribución uniforme* $U \sim \mathcal{U}(0,1)$*.*

3. *Sea* X *una variable aleatoria con función de distribución estrictamente creciente en un intervalo, luego* $Y = F_Y^{-1}(F_X(X))$ *posee función de distribución* F_Y*.*

La primera propiedad indica como poder transformar variables $\mathcal{U}(0,1)$ en variables con cualquier distribución que se necesite, la segunda explica como poder transformar variables con una determinada distribución en $\mathcal{U}(0,1)$, y la tercera combina los dos resultados anteriores.

En el caso de vectores, si se desea simular un vector aleatorio $(X,Y) \sim p_{XY}$ basta con generar una muestra de $X \sim p_X$, para luego usar dicho valor observado x para generar $Y|X = x \sim p_{Y|X=x}$. En el Ej. 1.1, primero se generarían muestras $X \sim \Gamma(2,1)$ para luego generar muestras $Y|_{X=x} \sim \mathcal{U}(0,x)$ (una y para cada x).

Para simular una variable truncada $X|X \in A$, basta con generar datos de $X \sim p_X$, para luego usar solamente con los que cumplen $x \in A$ (descartando el resto). Para simular

un vector truncado $(X,Y)|(X,Y) \in A$, basta con generar datos de $X \sim p_X$, generar sus correspondientes $Y|X = x \sim p_{Y|X=x}$ y luego quedarme con los pares $(x,y) \in A$.

1.2.3. Estadística Frecuentista

El objetivo de reconocer y estimar toda la distribución de una variable aleatoria por medio de datos es sumamente ambicioso. En la mayoría de los casos no se cuenta con una cantidad de datos suficiente para tal objetivo. Es entonces cuando surge la necesidad de incorporar supuestos previos sobre las variables involucradas, **modelando** el problema estadístico. No es necesario que dichos supuestos sean totalmente ciertos, sino que basta con que balanceen el desempeño esperado, la complejidad del modelo y la cantidad de datos con las que se cuentan.

El modelado consiste en asumir información parcial en el conocimiento de la distribución de la variable $p(x|\theta)$ [4]. Es decir que se conoce dicha distribución exceptuando un conjunto de parámetros $\theta \in \Theta$. La estadística frecuentista asume que las muestras son independientes e idénticamente distribuidas para cada posible parámetro, y a la distribución conjunta del set de datos observados $\mathbf{x} = (x_1, \cdots, x_n)$ se la conoce como **verosimilitud**:

$$\mathcal{L}(\theta) = p(\mathcal{D}_n|\theta) = \prod_{i=1}^{n} p(x_i|\theta) \tag{1.25}$$

La **bondad de un estimador** está dada por la relación de compromiso sesgo/varianza, como se puede ver en el siguiente teorema.

Propiedades 1.10 $\mathbb{E}[(\hat{\theta} - \theta)^2|\theta] = var(\hat{\theta}|\theta) + \mathcal{B}^2(\theta)$, *donde* $\mathcal{B}(\theta) = \theta - \mathbb{E}[\hat{\theta}|\theta]$ *se denomina sesgo.*

La demostración puede verse al final de la sección. La relación de compromiso sesgo/varianza explica que un buen estimador necesita tener simultáneamente bajo sesgo y baja varianza. Se habla de relación de compromiso porque muchas de las soluciones más utilizadas para mejorar uno de los términos termina perjudicando al otro.

Uno de los estimadores más utilizados en la estadística frecuentista, debido a su consistencia, es el **estimador de máxima verosimilitud** $\hat{\theta}_{\text{MV}} = \arg\max_{\theta \in \Theta} \mathcal{L}(\theta)$, es decir elegir los parámetros que maximicen la verosimilitud (los estimadores son funciones de la muestra observada). Bajo ciertas condiciones de regularidad dicha estimación puede efectuarse igualando a cero la derivada del logaritmo[5] de la verosimiltud:

$$\sum_{i=1}^{n} \frac{\partial}{\partial \theta} \log p(x_i|\hat{\theta}_{\text{MV}}) = 0 \tag{1.26}$$

Una vez caracterizados los parámetros, el modelo es capaz de predecir el comportamiento de nuevas muestras no observadas. Sea x_{test} una muestra no observada en $\mathbf{x}$, las

[5]Por ser una función monótona, no modifica la ubicación de los máximos.

predicciones se efectuan a través del **principio de invarianza**: la estimación por máxima verosimilitud de cualquier función de θ puede calcularse evaluando dicha función en $\hat{\theta}_{\text{MV}}$ [5].

$$\widehat{p}_{\text{MV}}(x_{\text{test}}) = p(x_{\text{test}}|\hat{\theta}_{\text{MV}}) \tag{1.27}$$

Demostración 1.1 (Prop. 1.10) *El error cuadrático medio puede descomponerse como:*

$$\mathbb{E}[(\hat{\theta} - \theta)^2|\theta] = \mathbb{E}[(\hat{\theta} - \mathbb{E}[\hat{\theta}|\theta] + \mathbb{E}[\hat{\theta}|\theta] - \theta)^2|\theta] \tag{1.28}$$

$$= \mathbb{E}[(\hat{\theta} - \mathbb{E}[\hat{\theta}|\theta])^2|\theta] + (\mathbb{E}[\hat{\theta}|\theta] - \theta)^2 + 2\mathbb{E}[(\hat{\theta} - \mathbb{E}[\hat{\theta}|\theta])|\theta](\mathbb{E}[\hat{\theta}|\theta] - \theta) \tag{1.29}$$

donde cada uno de los sumandos se puede simplificar como:

- $\mathbb{E}[(\hat{\theta} - \mathbb{E}[\hat{\theta}|\theta])^2|\theta] = var(\hat{\theta}|\theta)$
- $(\mathbb{E}[\hat{\theta}|\theta] - \theta)^2 = \mathcal{B}^2(\theta)$
- $\mathbb{E}[(\hat{\theta} - \mathbb{E}[\hat{\theta}|\theta])|\theta](\mathbb{E}[\hat{\theta}|\theta] - \theta) = 0$

Reemplazando en la expresión correspondiente el teorema fue demostrado.

1.2.4. Estadística Bayesiana

La estadística bayesiana busca verdades en contexto de incertidumbre, interpretando la probabilidad como una medida de credibilidad en un evento [6, Capítulo 1]. El modelo no solo representa el fenómeno a predecir, sino también nuestra propia ignorancia sobre el mismo. Esto no quiere decir que las ciencias empíricas están condenadas a decir "no se" a todas las hipótesis que uno desea contrastar, sino que busca evitar mentir maximizando incertidumbre (no afirmar más de lo que se sabe) dada la información disponible (sin ocultar lo que efectivamente se sabe).

A nivel técnico, la estadística bayesiana representa los parámetros del modelo por medio de una variable aleatoria T con distribución *a priori* $p_T(\theta)$ [7]. En este tipo de modelos, la hipótesis de independencia es válida *cuando se conoce el parámetro*. Es decir que la verosimilitud de una muestra puede escribirse como $p_{\mathbf{X}|T=\theta}(\mathbf{x}) = \prod_{i=1}^{n} p_{X|T=\theta}(x_i)$. No se pierde generalidad en asumir que las variables son idénticamente distribuidas[6].

El corazón de la estadística bayesiana es la *distribución a posteriori*, la cuál se deduce por medio de la **regla de bayes** combinando la distribución *a priori* con la verosimilitud.

$$p_{T|\mathbf{X}=\mathbf{x}}(\theta) \propto p_T(\theta) \cdot \prod_{i=1}^{n} p_{X|T=\theta}(x_i) \tag{1.30}$$

[6]Se podría haber escrito $p_{X_i|T=\theta}(x_i)$ en su lugar, pero no es necesario

La distribución *a posteriori* nos permite definir estimadores puntuales a partir de ella. En el caso de buscar parámetros dentro de un conjunto Θ discreto, se suele elegir como estimador el **máximo a posteriori** $\hat{\theta}_{\text{MAP}} = \arg\max_{\theta \in \Theta} p_{T|\mathbf{X}=\mathbf{x}}(\theta)$. En el caso de Θ continuo, la elección habitual suele ser la **media a posteriori** $\hat{\theta}_{\text{BAY}} = \mathbb{E}[T|\mathbf{X} = \mathbf{x}]$.

Sin embargo, el verdadero potencial de la estadística bayesiana radica en hacer predicciones sin necesidad de estimadores puntuales. En este sentido, este tipo de estadística no solo puede resolver los mismos problemas que la frecuentista, sino que también pueden intentar resolver problemas donde la estadística clásica es insuficiente o iluminar el sistema subyacente con un modelado más flexible. Es entonces que se define la **distribución predictiva**.

$$p_{X_{\text{test}}|\mathbf{X}=\mathbf{x}}(x_{\text{test}}) = \int_{\Theta} p_{X|T=\theta}(x_{\text{test}}) p_{T|\mathbf{X}=\mathbf{x}}(\theta) d\theta \tag{1.31}$$

donde X_{test} es una variable aleatoria no vista en el conjunto de entrenamiento $\mathbf{X}$. A continuación se analizará un ejemplo mostrando como trabajar con este tipo de estadística analíticamente.

Ejemplo 1.3 *El tiempo de vida (en años) de un transistor es una variable aleatoria con distribución exponencial de parámetro θ. A priori se modela θ como una variable aleatoria con distribución $\Gamma(2, 3)$. Si en 20 transistores se observó una duración total $\sum_{i=1}^{20} x_i = 7$.*

1. *Hallar la distribución a posteriori del parámetro θ.*
2. *Hallar la distribución predictiva del tiempo de vida de un transistor.*

Como primer paso en un problema bayesiano, hay que comenzar planteando la distribución *a posteriori*. En este caso evitaremos las constantes de proporcionalidad:

$$p_{T|\mathbf{X}=\mathbf{x}}(\theta) \propto p_T(\theta) \cdot \prod_{i=1}^{n} p_{X|T=\theta}(x_i) \propto \theta e^{-3\theta} \mathbf{1}\{\theta > 0\} \cdot \prod_{i=1}^{20} \theta e^{-\theta x_i} = \theta^{21} e^{-10\theta} \mathbf{1}\{\theta > 0\} \tag{1.32}$$

Es decir, la variable se distribuye *a posteriori* como $T|_{\mathbf{X}=\mathbf{x}} \sim \Gamma(22, 10)$. La distribución predictiva es de la forma

$$p_{X_{\text{test}}|\mathbf{X}=\mathbf{x}}(x_{\text{test}}) = \int_{\Theta} p_{X|T=\theta}(x_{\text{test}}) p_{T|\mathbf{X}=\mathbf{x}}(\theta) d\theta \propto \int_{0}^{\infty} \theta e^{-\theta x_{\text{test}}} \mathbf{1}\{x_{\text{test}} > 0\} \cdot \theta^{21} e^{-10\theta} d\theta \tag{1.33}$$

Reconociendo el núcleo de la integral, se puede observar que el mismo es proporcional a la densidad de una $\Gamma(\nu, \lambda)$, es decir $p(x) = \frac{\lambda^\nu}{\Gamma(\nu)} x^{\nu-1} e^{-\lambda x} \mathbf{1}\{x > 0\}$. Sabiendo que por ser densidad debe integrar 1:

$$p_{X_{\text{test}}|\mathbf{X}=\mathbf{x}}(x_{\text{test}}) \propto \int_{0}^{\infty} \theta^{22} e^{-\theta(10+x_{\text{test}})} d\theta \cdot \mathbf{1}\{x_{\text{test}} > 0\} \propto \frac{1}{(10 + x_{\text{test}})^{23}} \mathbf{1}\{x_{\text{test}} > 0\} \tag{1.34}$$

donde se utilizó $\nu = 23$ y $\lambda = 10 + x_{\text{test}}$. Esta distribución se la conoce como Lomax (véase Cuadro 1.2) $X_{\text{test}}|_{\mathbf{X}=\mathbf{x}} \sim \text{Lomax}(22, 10)$.

Tanto *a priori* como *a posteriori*, la variable T es una Gamma. Este fenómeno de mantenerse dentro de una familia ocurre por cierta compatibilidad entre la distribución *a priori* y la verosimilitud (en este caso una exponencial, caso particular de la Gamma). Cuando se da este fenómeno se dice que la distribución *a priori* es una **conjugada a priori**. Las soluciones analíticas suelen proponer conjugadas, como distribución *a priori*, ya que así garantizan que la distribución *a posteriori* pertenezca a una familia conocida (la misma que la distribución *a priori*). Es simplemente una recomendación para hacer sencillos (o al menos factibles) los cálculos.

1.2.4.1. Estadísticos Suficientes

Un concepto muy útil a la hora de efectuar inferencia es el de **estadístico suficiente**. Un estadístico $S(\mathbf{X})$ se denomina suficiente para θ si la distribución de $\mathbf{X}|_{S(\mathbf{X})=s}$ no depende de θ. Es decir que toda la información que posee la muestra sobre θ se encuentra en el estadístico. Además, el teorema de Neyman-Fisher nos permite encontrar estadísticos suficientes de forma muy sencilla [4, Capítulo 6].

Propiedades 1.11 (Teorema de Neyman-Fisher) *El estadístico $S(\mathbf{X})$ es suficiente, si y solo si su verosimilitud se puede descomponer como:*

$$p_{\mathbf{X}|T=\theta}(\mathbf{x}) = g(\theta, S(\mathbf{x})) \cdot h(\mathbf{x}) \tag{1.35}$$

En términos bayesianos un estadístico suficiente se interpreta como una independencia condicional $\mathbf{X} \perp \theta|_{S(\mathbf{X})=s}$ (es decir que la muestra y los parámetros son independientes cuando se conoce el estadístico suficiente). Este resultado implica que la distribución *a posteriori* debe cumplir $p_{T|\mathbf{X}=\mathbf{x}}(\theta) = p_{T|S(\mathbf{X})=s(\mathbf{x})}(\theta)$, y por lo tanto nos permite intercambiar el conocimiento de toda la muestra por el del estadístico suficiente. En el ejemplo anterior la distribución solo dependía de la muestra a través de la suma, estadístico suficiente para θ en una distribución exponencial.

Esto nos permite hacer equivalencias sobre los datos de las variables observadas. En el Ej. 1.3, es equivalente pensar que se cuenta con 20 muestras $\exp(\theta)$ que con una sola muestra $\Gamma(20, \theta)$[7]. Otro ejemplo clásico donde se da este fenómeno es en las variables Bernoulli, donde también la suma es estadístico suficiente: es equivalente tener n muestras $\text{Ber}(p)$ que una muestra $\text{Bin}(n, p)$.

[7] La suma de 20 variables $\exp(\theta)$ independientes e idénticamente distribuidas se distribuye como una $\Gamma(20, \theta)$

1.2.4.2. Test de hipótesis

TODO

2

Regresión en Inteligencia Artificial

Tal vez el mayor desafío no sea aprender a usar la inteligencia artificial, sino redefinir nuestro aporte humano en un mundo que automatiza incluso la inteligencia.

El objetivo de la inteligencia artificial es resolver tareas de forma automatizada y con la mejor calidad posible. Ésta tecnología está modificando de forma acelerada la matriz laboral tal como la conocíamos. Tareas que antes requerían horas de redacción, análisis o asistencia técnica ahora pueden ser resueltas en minutos por una inteligencia artificial entrenada para comprender y generar material con una sorprendente fluidez. Esta transformación no se limita a sectores creativos o administrativos; también está empezando a influir en el desarrollo de software, la ingeniería de datos y la automatización de procesos, áreas donde muchos ingenieros se formaron creyendo que la demanda sería estable o creciente.

Esto plantea una pregunta incómoda pero necesaria: ¿qué lugar ocuparemos los profesionales en un entorno donde las máquinas no solo ejecutan, sino también piensan -al menos en términos funcionales-? El rol del ingeniero ya no se limita a diseñar sistemas eficientes, sino que debe integrar consideraciones éticas, adaptarse a herramientas inteligentes y desarrollar una visión crítica sobre la tecnología que crea. Tal vez el mayor desafío no sea aprender a usar estos modelos, sino redefinir nuestro aporte humano en un mundo que automatiza incluso la inteligencia.

En cualquier problema de **aprendizaje supervisado**, es decir inferir Y a partir de X[1], siempre hay algunas magnitudes que se pueden destacar. El objetivo siempre será minimizar el valor esperado de la llamada **función costo** $\ell(x, y)$, el cuál recibe el nombre de **riesgo esperado** $\mathbb{E}[\ell(X, Y)]$. La estimación que minimice dicho riesgo se conocerá como **solución óptima** y llamaremos **error bayesiano** al mínimo error posible capaz de ser alcanzado (asociado a la solución óptima). Estamos hablando de un límite fundamental para el error que nunca podrá ser mejorado independientemente de la tecnología utilizada.

En particular, en este capítulo estudiaremos el **problema de regresión**, es decir estimar $Y = \varphi(X)$ utilizando el error cuadrático como función costo $\ell(x, y) = (y - \varphi(x))^2$ (y

[1]Véase el Capítulo 4 para detalles precisos sobre el término.

por lo tanto el error cuadrático medio como riesgo esperado). En el capítulo anterior se demostró, Prop. 1.8, que la regresión óptima es $\mathbb{E}[Y|X=x]$ y con ésta el error alcanza el error bayesiano $\mathbb{E}[\text{var}(Y|X)]$. Todo este resultado es la base del aprendizaje estadístico: lo mejor que puedo hacer es utilizar la esperanza condicional como regresor y el menor error al que puedo aspirar es el bayesiano.

La inteligencia artificial lejos está de terminarse con este resultado. En la práctica, el problema radica en no conocer la distribución de los datos, y por lo tanto no poder calcular fidedignamente la esperanza condicional. El aprendizaje estadístico propone entonces **aprender** la esperanza condicional por medio de datos.

2.1. Relación de Compromiso Sesgo/Varianza

La solución inmediata que uno puede proponer al problema de regresión es la **minimización del riesgo empírico**. Es decir, encontrar la función $\varphi(x)$ que minimice $\frac{1}{n_{\text{tr}}}\sum_{i=1}^{n_{\text{tr}}}\ell(x_i,y_i)$ para un conjunto de datos observado $\{(x_i,y_i)\}_{i=1}^{n_{\text{tr}}}$. El problema está en que el verdadero objetivo es minimizar el riesgo esperado $\mathbb{E}[\ell(X,Y)]$ que no necesariamente va a coincidir con el empírico. En este sentido surge el **gap de generalización**: la capacidad de generalizar el comportamiento de los datos observados a datos desconocidos (representados por los valores esperados).

$$\underbrace{\mathbb{E}[\ell(X,Y)]}_{\text{Riesgo esperado}} = \underbrace{\frac{1}{n_{\text{tr}}}\sum_{i=1}^{n_{\text{tr}}}\ell(x_i,y_i)}_{\text{Riesgo empírico}} + \left(\underbrace{\mathbb{E}[\ell(X,Y)] - \frac{1}{n_{\text{tr}}}\sum_{i=1}^{n_{\text{tr}}}\ell(x_i,y_i)}_{\text{Gap de generalización}}\right) \tag{2.1}$$

En este sentido, para disminuir el riesgo esperado se necesita simultáneamente tratar de minimizar el riesgo empírico y el gap de generalización; dos magnitudes de características muy distinta [8, Sección 2.9]. Este problema es análogo al estimador paramétrico puntual estudiado en la Prop. 1.10: es una relación de compromiso entre el sesgo (representado por el riesgo empírico) y la varianza (representado por el gap).

Cuando hablamos de errores cuadráticos, hay que tener en consideración que este tipo de error es una magnitud difícil de interpretar por no tener valor máximo. Si bien tener un error nulo implica desempeño perfecto, otro valor de error requiere contextualizarlo para catalogarlo como insuficiente o satisfactorio. En este sentido, el gap de generalización propone un marco interpretativo al ser una diferencia: ¿Que tanto más grande es el riesgo esperado con respecto al empírico? En cambio el riesgo empírico por si solo no es interpretable. Para darle un sentido se lo compara con el error bayesiano, ya que este es el error al que todo algoritmo desea alcanzar. En la práctica el error bayesiano suele ser livianamente estimado con imaginación: ¿Que error creo que se puede llegar a alcanzar

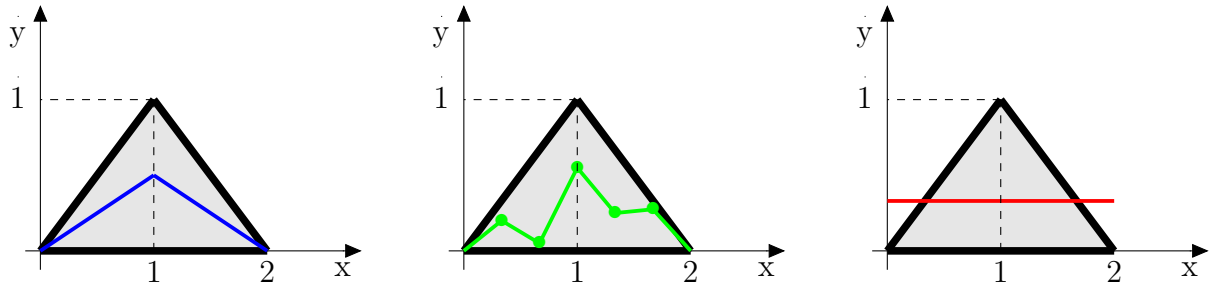


Figura 2.1: Regresores para el Ej. 2.1. (a) Solución $\mathbb{E}[Y|X=x]$ asociada a la esperanza condicional, (b) solución empírica y (c) recta de regresión.

en esta tarea? En muchos casos, el *error humano* es buen candidato.

A continuación se analizará un ejemplo para entender las diferencias entre los riesgos empírico y esperado.

Ejemplo 2.1 *Sea (X,Y) un vector uniforme en el triángulo de vértices $(0,0)$, $(1,1)$ y $(2,0)$. Analizar posibles regresores para este problema.*

Comencemos analizando la solución óptima. Al factorizar la distribución conjunta uniforme, se observa una condicional uniforme (véase Ej. 1.2). En este caso,

$$Y|X=x \sim \begin{cases} \mathcal{U}(0,x) & \text{Si } 0<x<1 \\ \mathcal{U}(0,2-x) & \text{Si } 1<x<2 \end{cases} \tag{2.2}$$

y por lo tanto, solución $\mathbb{E}[Y|X=x]$ asociada a la esperanza condicional puede verse gráficamente en la Fig. 2.1a. Analíticamente, tanto la esperanza como la varianza condicional se definen a partir del Cuadro 1.2:

$$\mathbb{E}[Y|X=x] = \begin{cases} \frac{x}{2} & \text{Si } 0<x<1 \\ \frac{2-x}{2} & \text{Si } 1<x<2 \end{cases} \qquad \text{var}(Y|X=x) = \begin{cases} \frac{x^2}{12} & \text{Si } 0<x<1 \\ \frac{(2-x)^2}{12} & \text{Si } 1<x<2 \end{cases} \tag{2.3}$$

En este caso el error bayesiano se puede calcular utilizando que el triángulo tiene área unitaria:

$$\mathbb{E}[\text{var}(Y|X)] = \int_0^1 \int_0^x \frac{x^2}{12} dydx + \int_1^2 \int_0^{2-x} \frac{(2-x)^2}{12} dydx = \frac{1}{24} \tag{2.4}$$

Esas son la solución ideal y el mínimo error esperado al que se puede aspirar. Sin embargo, dado un conjunto de datos $\{(x_i,y_i)\}_{i=1}^{n_{\text{tr}}}$, cuando se elige una solución minimizando el riesgo empírico se encuentran regresores como el visto en la Fig. 2.1b. Esta solución no solo minimiza $\frac{1}{n_{\text{tr}}}\sum_{i=1}^{n_{\text{tr}}} \ell(x_i,y_i)$ sino que directamente alcanza error nulo. Sin embargo, soluciones de este tipo suelen poseer un algo gap de generalización debido a un exceso de

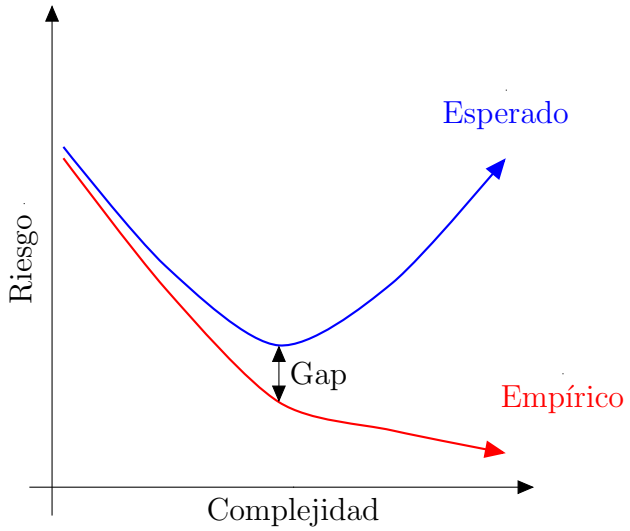


Figura 2.2: Relación de compromiso sesgo/varianza típica según la teoría clásica de generalización.

complejidad en el modelado, es decir un **problema de varianza**. En este caso, al darle libertad total a la elección del regresor, se eligió un regresor mucho más complejo que el óptimo $\mathbb{E}[Y|X = x]$. Las soluciones que poseen un excesivo problema de varianza se dice que tienen **overfitting**, ya que éstas se sobreajustan a los datos.

En cambio, si uno asigna una complejidad excesivamente baja al regresor se encuentra con un **problema de sesgo**. En la Fig. 2.1c se postula la recta de regresión como posible regresor con este tipo de problema. Dentro de los regresores de complejidad lineal, la recta de regresión es el óptimo (véase Prop 1.6). Posiblemente esta solución tenga un muy bajo gap de generalización (ni siquiera se calculó utilizando los datos), sin embargo al tener una complejidad mucho más baja que la solución óptima, el riesgo empírico será importante. Las soluciones que poseen un excesivo problema de sesgo se dice que tienen **underfitting**, ya que éstas se subajustan a los datos.

La Fig. 2.2 muestra como la teoría clásica de generalización caracteriza esta relación de compromiso. Los modelos de muy baja complejidad ven imposibilitado cualquier tipo de sobreajuste y tratarán de forma similar los datos conocidos a los desconocidos (alcanzando un bajo gap de generalización). En contraposición, los modelos muy complejos pueden reducir el riesgo empírico tanto como quieren pero corren el riesgo de sobreajustar. Encontrar el balance óptimo, lejos de ser trivial, es problema principal de la inteligencia artificial.

La cantidad de datos con la que se cuenta juega un rol vital en este análisis. Si se cuenta con la posibilidad de obtener más y más datos llegará un momento en que un modelo, de una complejidad determinada, no podrá sobreajustarlos. Bajo ciertas hipótesis

de consistencia, el gap de generalización deberá disminuir con el número de muestras (en la medida que los promedios tiendan a los valores esperados en (2.1))[2]. En este sentido, aumentar la cantidad de datos soluciona problemas de varianza, pero no así de sesgo (al tener más muestras, es más difícil representarlas a todas con una complejidad fija). En cualquier caso, se deberá adaptar la complejidad del modelo a la cantidad de datos con las que se cuenta, pudiendo permitirse modelos más complejos cuando se cuenta con grandes cantidades de datos y limitando la misma cuando la cantidad es escueta.

2.2. Regresión Lineal

El objetivo del aprendizaje estadístico es intentar disminuir simultáneamente el riesgo empírico y el gap de generalización para así reducir el riesgo esperado (2.1). Pero mientras que el gap de generalización no se conoce, el riesgo empírico es totalmente computable y por lo tanto podemos detectar fácilmente cuando estamos en presencia de un problema de sesgo. La idea de la regresión lineal es muy sencilla: limitar al máximo la complejidad del modelo (soluciones lineales) para tener un gap de generalización moderado (evitar problemas de varianza) y posteriormente verificar si hay problema de sesgo. En caso de que no los haya podemos concluir que tenemos un aceptable riesgo esperado. En este sentido, la regresión lineal busca aprender la recta de regresión (véase Prop. 1.6) en lugar de la esperanza condicional, básicamente porque es una tarea mucho más sencilla[3].

Sea $\{(x_i, y_i)\}_{i=1}^{n_{\text{tr}}}$ un conjunto de datos con $x_i \in \mathbb{R}^{d_x}$ e $y_i \in \mathbb{R}$. Se propone buscar soluciones de la forma $\hat{y}(x) = w^T \cdot x + b$ con $w \in \mathbb{R}^{d_x}$ y $b \in \mathbb{R}$ minimizando el riesgo empírico. Es decir, un algoritmo de inteligencia artificial tiene dos etapas bien diferenciadas: una primera llamada **entrenamiento** donde se calcularán los parámetros (w y b en este caso) y una segunda etapa llamada **predicción** donde a cada x se le asignará un $\hat{y}(x)$ (en este caso $\hat{y}(x) = w^T \cdot x + b$). En el caso de la regresión lineal, el entrenamiento entonces buscará calcular

$$(w, b) = \underset{w \in \mathbb{R}^{d_x}, b \in \mathbb{R}}{\arg\min} \frac{1}{n_{\text{tr}}} \sum_{i=1}^{n_{\text{tr}}} (w^T \cdot x_i + b - y_i)^2 \tag{2.5}$$

Esta ecuación define un problema de proyección ortogonal de álgebra lineal. Esto se puede analizar **vectorizando** la ecuación. Definiendo $\mathbf{X} \in \mathbb{R}^{n_{\text{tr}} \times (d_x+1)}$, $\mathbf{y} \in \mathbb{R}^{n_{\text{tr}}}$ y $\mathbf{w} \in$

[2]Dado que el regresor elegido depende del mismo conjunto de datos con el que se mide el riesgo empírico, el análisis es sofisticado (la *ley de los grandes números* no es suficiente para explicar el fenómeno). Para más información ver [7, Sección 6.5]

[3]Será una recta si X es escalar, un plano si tiene dos dimensiones, etc. En general será un hiperplano.

$\mathbb{R}^{d_x+1}$ como

$$\mathbf{X} = \begin{pmatrix} 1 & x_1^T \\ 1 & x_2^T \\ \vdots & \vdots \\ 1 & x_{n_{\text{tr}}}^T \end{pmatrix}, \qquad \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n_{\text{tr}}} \end{pmatrix}, \qquad \mathbf{w} = \begin{pmatrix} b \\ w \end{pmatrix} \tag{2.6}$$

el problema (2.5) puede reducirse a minimizar $J(\mathbf{w}) = \frac{1}{n_{\text{tr}}}\|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2$. Para ello basta con analizar la primera derivada (gradiente) y la segunda derivada (matriz Hessiana) respecto a $\mathbf{w}$ (para más información sobre derivadas respecto vectores/matrices ver [9]).

$$\nabla J(\mathbf{w}) = \frac{2}{n_{\text{tr}}}\mathbf{X}^T(\mathbf{X}\mathbf{w} - \mathbf{y}), \qquad \mathcal{H}_J(\mathbf{w}) = \frac{2}{n_{\text{tr}}}\mathbf{X}^T\mathbf{X} \tag{2.7}$$

Es habitual en los problemas de regresión que $n_{\text{tr}} \gg d_x$, lo cuál se suele traducir en una matriz Hessiana inversible (teniendo en cuenta que los x fueron elegidos de forma aleatoria). En ese caso, será también una matriz definida positiva y por lo tanto el problema será convexo. Es decir que el resultado obtenido de igualar a cero el gradiente de $J(\mathbf{w})$ será efectivamente un mínimo. Se puede despejar entonces para encontrar el procedimiento del entrenamiento:

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{2.8}$$

La solución es la **pseudoinversa** de la matriz $\mathbf{X}$ multiplicada por $\mathbf{y}$. A continuación se presenterá un ejemplo de como funciona la regresión lineal.

Ejemplo 2.2 *Se desea hacer una regresión lineal (sin normalizar) sobre el siguiente conjunto de datos:*

X	*0.2*	*1.4*	*-1.4*	*-0.2*
Y	*20.0*	*10.0*	*10.0*	*0.0*

- *Hallar los parámetros del modelo.*
- *Predecir y para $x = 0.1$.*

Basta con definir las matrices de (2.8):

$$\mathbf{w} = \left(\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0.2 & 1.4 & -1.4 & -0.2 \end{pmatrix} \begin{pmatrix} 1 & 0.2 \\ 1 & 1.4 \\ 1 & -1.4 \\ 1 & -0.2 \end{pmatrix} \right)^{-1} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0.2 & 1.4 & -1.4 & -0.2 \end{pmatrix} \begin{pmatrix} 20 \\ 10 \\ 10 \\ 0 \end{pmatrix}$$

$$= \begin{pmatrix} 4 & 0 \\ 0 & 4 \end{pmatrix}^{-1} \begin{pmatrix} 40 \\ 4 \end{pmatrix} = \begin{pmatrix} 10 \\ 1 \end{pmatrix} \tag{2.9}$$

con lo cual $b = 10$ y $w = 1$; y finalmente $\hat{y}(0.1) = 10.1$.

2.2.1. Codificación de variables categóricas

La regresión lineal, así como la mayor parte de los algoritmo de inteligencia artificial, requieren que los valores de sus entradas tengan valores numéricos. Pero hay determinados tipos de variables, donde no es posible encontrar una **relación de orden**, donde las mismas simplemente representa categorías (con una cantidad finita de opciones posibles)[4].

Por ejemplo, podemos contar con una base de datos donde una de sus columnas represente el color de un objeto. Supongamos que los resultados posibles son *rojo, verde, azul* y *negro*. Si asignamos respectivamente los valores 0, 1, 2 y 3 estaríamos diciendo que el rojo está mas cerca del verde que del negro, lo cuál sesgaría nuestro análisis por ser falso.

Para evitar este tipo de decisiones arbitrarias, se suele codificar a las variables categóricas sin relación de orden específica con representaciones **One-Hot**. Cada columna categórica, de K clases posibles, se convierte en K variables binarias donde siempre una y solamente una de ellas toma el valor 1. En nuestro ejemplo, codificaríamos el *rojo* con $(1, 0, 0, 0)$, el *verde* con $(0, 1, 0, 0)$, el *azul* con $(0, 0, 1, 0)$ y el *negro* con $(0, 0, 0, 1)$ (es decir 4 variables en total). Notar que dos colores cualesquiera distintos siempre están a una distancia geométrica de $\sqrt{2}$.

Es importante destacar que el proceso de codificación se define *durante el entrenamiento*. Es decir, que si al momento de efectuar una predicción se le solicita al algoritmo una categoría no vista anteriormente, se le suele asignar a todas las variables codificadas el valor 1. De esta manera simultáneamente tendrá una distancia constante al resto de las categorías ($\sqrt{K-1}$), y una categoría válida tendrá más cerca al resto de categorías válidas que a esta codificación particular. *One-Hot* permite procesar muy fácilmente las variables categóricas, pero puede aumentar considerablemente la cantidad de variables de

[4]En las variables que solo pueden tomar dos valores, la relación de orden es intrascendente. En esta sección nos referimos a variables con mayor cantidad de valores posibles.

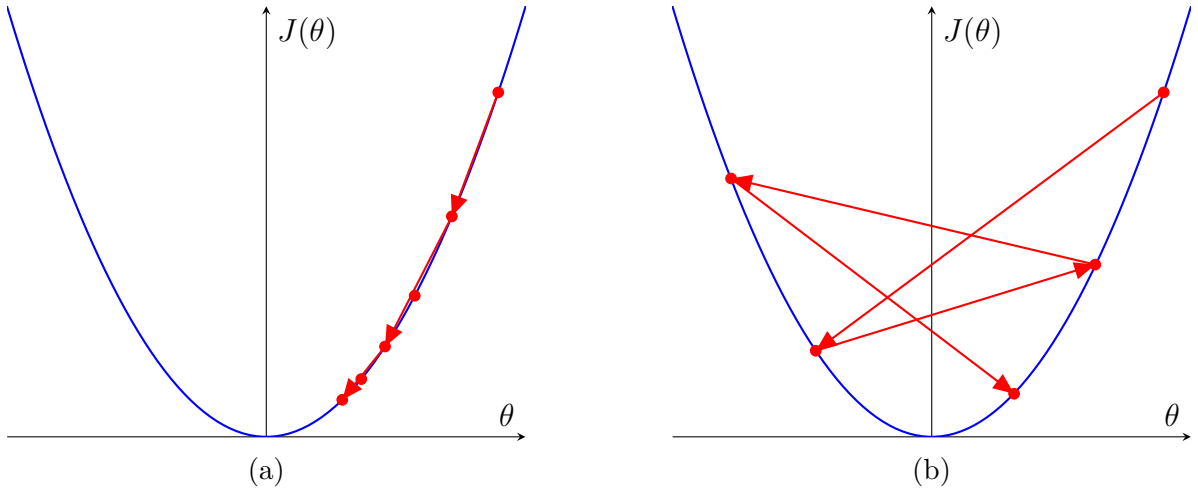


Figura 2.3: Comparación del comportamiento del gradiente descendente con distinto *learning rate*. (a) caso convergente y (b) caso divergente.

entrada o **predictores**.

2.3. Gradiente Descendente

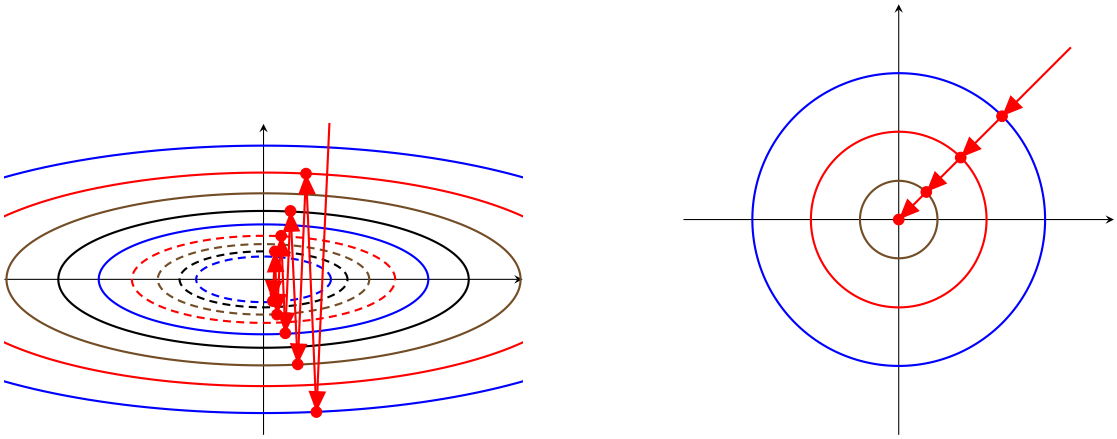
La regresión lineal tiene la ventaja de contar con solución analítica (2.8), pero la mayoría de las funciones a minimizar no poseen esa ventaja. Además si tenemos en cuenta que para un d_x muy grande la inversa de la matriz presente en (2.8) no es computable, nos damos cuenta de la necesidad de contar con un método numérico para minimizar funciones.

El método del **gradiente descendente** es un algoritmo numérico de optimización presentado por Cauchy muchos años atrás [10] y, sin embargo, es la esencia de la mayoría de los algoritmos modernos de inteligencia artificial. La idea es sencilla: igualar a cero la derivada de una función a minimizar $J(\theta)$ *numéricamente*. Es decir, avanzar *poco a poco* (de forma iterativa) en la dirección del máximo decrecimiento de la función.

$$\theta_{t+1} = \theta_t - \alpha \cdot \nabla J(\theta_{t+1}) \tag{2.10}$$

donde $\alpha > 0$ recibe el nombre de **learning rate** o tasa de aprendizaje. Este tipo de parámteros que no se deciden durante el entrenamiento reciben el nombre de **hiperparámetro**, para diferenciarlos de los parámetros entrenables. Según el valor de α, el comportamiento del algoritmo puede ser bien distinto. En la Fig. 2.3a se puede observar un ejemplo convergente del algoritmo. Paso a paso el algoritmo se va acercando al mínimo, aunque corre el riesgo de necesitar muchas iteraciones para alcanzar la convergencia. Sin embargo un *learning rate* muy grande, lejos de acelerar, puede generar comportamientos divergentes en el algoritmo como puede verse en la Fig. 2.3b.

Por desgracia no existe un optimizador universal que funcione para cualquier tarea y



(a) (b)

Figura 2.4: Comparación del gradiente descendente (a) sin y (b) con normalización de las variables de entrada a partir de las curvas de nivel.

conjunto de datos [11], dependerá de en cada problema el encontrar un valor de α adecuado. En la práctica se suele apuntar al valor más grande que genere un comportamiento convergente, eligiéndolo por *prueba y error*. A continuación analizaremos algunos detalles a tener en cuenta a la hora de utilizar este algoritmo.

2.3.1. Normalización como pre-procesamiento

El gradiente descendente, descripto en (2.10), tiene la característica en tener un mismo valor de α para todas las direcciones. El motivo de esto, tiene que ver con solamente tener que seleccionar un hiper-parámetro en lugar de muchos. Sin embargo, puede que no exista un valor que satisfaga a todas las direcciones. Es por esto que surge la necesidad de **normalizar**.

La normalización de cada componente permite poner a todas las variables en la misma unidad. Esta fuerza a todas las variables de entrada o *predcitores* a tener valor medio nulo y varianza unitaria (empíricamente hablando). Formalmente asigna los valores

$$(\mathbf{x})_k \leftarrow \frac{(\mathbf{x})_k - \mu_k}{\sigma_k} \tag{2.11}$$

donde las μ_k y σ_k son calculadas previo al entrenamiento como:

$$\mu_k = \frac{1}{n_{\text{tr}}} \sum_{i=1}^{n_{\text{tr}}} (\mathbf{x}_i)_k, \qquad \sigma_k = \sqrt{\frac{1}{n_{\text{tr}}} \sum_{i=1}^{n_{\text{tr}}} \left[(\mathbf{x}_i)_k - \mu_k\right]^2} \tag{2.12}$$

con n_{tr} la cantidad de muestras de entrenamiento. La Fig. 2.4a muestra el comportamiento de la minimización por gradiente descendente sobre una superficie representada por las curvas de nivel, denotando un comportamiento errático (recordar que el gradiente es ortogonal a las curvas de nivel). En contraste, la Fig. 2.4b muestra su contraparte nor-

malizada, donde con menos iteraciones se logró optimizar los parámetros. Cabe destacar que la normalización se define durante la *etapa de entrenamiento*, fijando los valores de μ_k y σ_k. A la hora de realizar una predicción, se utilizará la normalización ya calculada previamente.

El uso de la normalización no se restringe solamente a algoritmos optimizados por gradiente descendente. Además de solucionar posibles problemas de convergencia, permite forzar media nula en los predictores hipótesis de muchos métodos basados en álgebra lineal del aprendizaje automático. También permite que sea pertinente la relación entre los predictores. Supongamos por ejemplo que contamos con un conjunto de datos que posee como variables la superficie de una vivienda a cotizar y como otra variable la cantidad de habitaciones. No debería ser distinto decir que una vivienda posee $36m^2$ y 3 habitaciones (12 veces más grande un número que el otro) que $600cm^2$ y 3 habitaciones (200 veces más grande). En ese sentido es muy útil cuando las magnitudes involucradas tienen diferentes unidades.

Sin embargo, vale resaltar que hay que tener muy claro el motivo de la normalización. Ya sea para ayudar la convergencia, para forzar media nula o para volver las magnitudes comparables, es necesario entender por que se utiliza. Normalizar preventivamente cualquier conjunto de datos suele traer muchos problemas.

2.3.2. Learning Rate óptimo

En esta sección se buscarán garantías teóricas de optimalidad para α. Esto es solamente un análisis teórico, en la práctica el *learning rate* se suele elegir intentando tomar el máximo valor posible convergente. Esto está relacionado con asociar el α con la velocidad de convergencia, lo cuál en rigor de verdad no es totalmente cierto. El presente análisis teórico nos permitirá entender las limitaciones de este tipo de asociaciones. Se estudiarán garantías sobre la velocidad de convergencia para un algoritmo de gradiente descendente sobre una función $J(\theta)$ genérica (no necesariamente regresión lineal, pero sin perderla de vista), aunque con un mínimo de hipótesis razonables de convexidad:

- Existe un único θ^* tal que $\nabla J(\theta^*) = 0$.
- La matriz Hessiana $\mathcal{H}_J(\theta)$ existe y es definida positiva para todo θ.

A continuación se utlizarán algunos resultados matemáticos conocidos. En primer lugar, se reescribirá $\nabla J(\theta_t)$ utilizando el **teorema de Taylor** de primer orden, alrededor del mencionado θ^* [12, Apéndice A.6]:

$$\nabla J(\theta_t) = \nabla J(\theta^*) + \mathcal{H}_J(\tilde{\theta}) \cdot (\theta_t - \theta^*) \tag{2.13}$$

para algún $\tilde{\theta}$ en el segmento que une θ_t y θ^*. Notar que $\mathcal{H}_J(\tilde{\theta})$ es una matriz real, cuadrada y simétrica. Por lo tanto, por el **teorema espectral** [13, Sección 4.1], puede escribirse

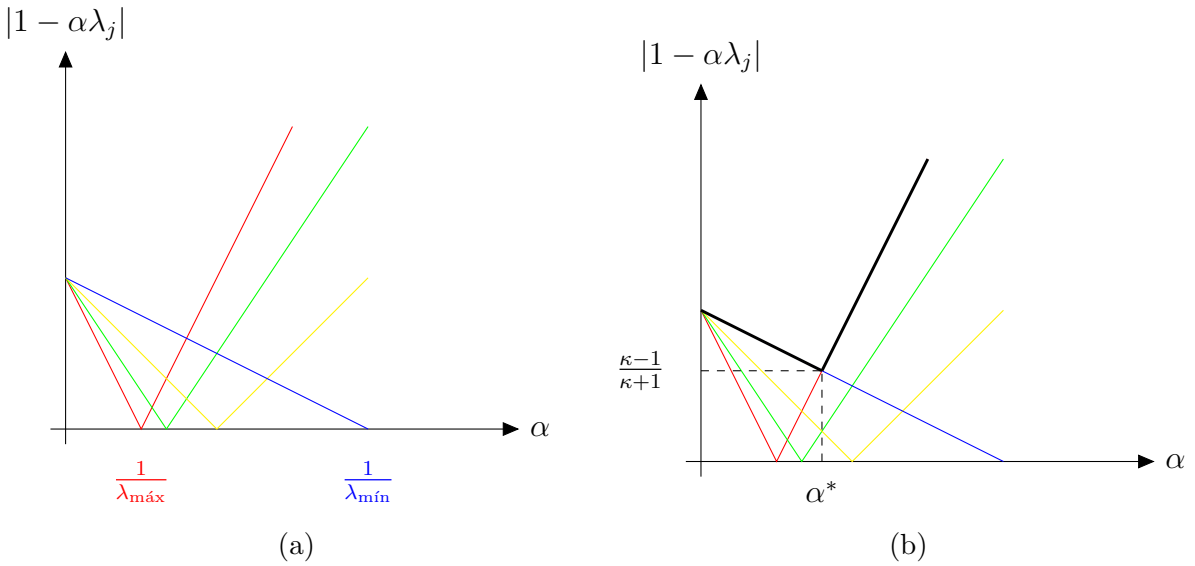


Figura 2.5: Resolución gráfica del problema (2.17). En (a) se resaltan los valores $|1-\alpha\lambda_j|$ para los diferentes autovalores λ_j; y en (b) se resalta la resolución del problema minmax.

como $\mathcal{H}_J(\tilde{\theta}) = Q^T\Lambda Q$ con una matriz de autovalores Λ diagonal y una de autovectores Q ortogonal $Q^TQ = QQ^T = I$. Utilizando este resultado y que $\nabla J(\theta^*) = 0$ se obtiene $\nabla J(\theta_t) = Q^T\Lambda Q(\theta_t - \theta^*)$. En general, Λ y Q podrán ser desconocidas (aunque en el caso de regresión lineal pueden calcularse ya que $\mathcal{H}_J(\tilde{\mathbf{w}}) = \frac{2}{n_{\text{tr}}}\mathbf{X}^T\mathbf{X}$ no depende de $\tilde{\mathbf{w}}$). Con el fin de estudiar la recurrencia, se reescribirá la diferencia $(\theta_{t+1} - \theta^*)$ utilizando la definición del gradiente descendente (2.10):

$$\theta_{t+1} - \theta^* = \theta_t - \theta^* - \alpha\nabla J(\theta_t) \tag{2.14}$$

$$= \left(I - \alpha Q^T\Lambda Q\right)(\theta_t - \theta^*) \tag{2.15}$$

$$= Q^T\left(I - \alpha\Lambda\right)Q\left(\theta_t - \theta^*\right) \tag{2.16}$$

Sea $v_t = Q\left(\theta_t - \theta^*\right)$, la relación de recurrencia (2.16) puede escribirse como $v_{t+1} = \left(I - \alpha\Lambda\right)v_t$ y por lo tanto $v_t = \left(I - \alpha\Lambda\right)^t v_0$. Hay una relación directa entre la convergencia en θ_t y la convergencia en v_t, por lo que para estudiar garantías, ésta última es suficiente. Como criterio de garantía sobre la velocidad de convergencia se elegirá un criterio de peor caso:

$$\min_{\alpha}\max_{j}|1-\alpha\lambda_j| \quad \text{s.t.} \quad |1-\alpha\lambda_j| < 1 \quad \forall\, j \tag{2.17}$$

donde λ_j son los elementos de la diagonal de Λ. Es un problema **minmax** con restricciones. Por el lado de las restricciones, la matriz $(I - \alpha\Lambda)$ es naturalmente diagonal y la convergencia estará dada cuando cada coeficiente sea menor que uno en valor absoluto (por estar elevada a la cantidad de iteraciones). Como criterio de garantía de velocidad se elige minimizar el peor caso de éstos (su máximo). Es importante notar que el α obtenido de esta manera no será el mejor posible en cada caso, sino el que me brinda garantías.

Este problema minmax se puede resolver gráficamente [14, Capítulo 9]. En la Fig. 2.5a se muestran los los diferentes valores $|1-\alpha\lambda_j|$ como función de α. Cada curva en V obtiene su mínimo cuando $\alpha = \frac{1}{\lambda_j}$, y habrá un valor mínimo y un valor máximo (serán positivos porque son autovalores de una matriz definida positiva). El máximo valor de estas curvas se puede ver en la Fig. 2.5b: la máxima curva comienza siendo la de menor autovalor ($\lambda_{\text{mín}}$), hasta que se cruza con la curva de mayor autovalor ($\lambda_{\text{máx}}$). En el vértice, o mínimo valor de esta nueva curva, se encuentra justamente el *learning rate* óptimo α^*. Para encontrar el valor hace falta intersectar las dos curvas: la semirrecta decreciente de la curva con mínimo autovalor con la semirrecta creciente de la curva con máximo autovalor.

$$1-\alpha^*\lambda_{\text{mín}} = \alpha^*\lambda_{\text{máx}} - 1 \quad \rightarrow \quad \alpha^* = \frac{2}{\lambda_{\text{mín}} + \lambda_{\text{máx}}} \tag{2.18}$$

La velocidad de convergencia estará asociada entonces al valor correspondiente a este *learning rate* en el eje de las ordenadas:

$$1 - \frac{2}{\lambda_{\text{mín}} + \lambda_{\text{máx}}}\lambda_{\text{mín}} = \frac{\lambda_{\text{máx}} - \lambda_{\text{mín}}}{\lambda_{\text{máx}} + \lambda_{\text{mín}}} = \frac{\kappa - 1}{\kappa + 1} \tag{2.19}$$

donde $\kappa = \frac{\lambda_{\text{máx}}}{\lambda_{\text{mín}}}$ se denomina **número de condición** de la matriz asociada $\mathcal{H}_J(\tilde{\theta})$. El *learning rate* óptimo no coincide con el máximo valor convergente; la condición $|1-\alpha\lambda_j| < 1$ para todo j se puede reescribir como $0 < \alpha < \frac{2}{\lambda_j}$ y por lo tanto la condición de convergencia es $\alpha < \frac{2}{\lambda_{\text{máx}}}$. Un α muy grande convergente, puede implicar rebotes a la hora de converger como es el caso del ejemplo de la Fig. 2.4a.

Vale la pena volver a mencionar que este análsis es teórico. En regresión lineal, donde se conoce la matriz $\mathcal{H}_J(\tilde{\mathbf{w}}) = \frac{2}{n_{\text{tr}}}\mathbf{X}^T\mathbf{X}$, no se suele utilizar el valor α^* principalmente porque calcular los autovalores es tan costoso como invertir la matriz para utilizar (2.8).

2.4. Regresión Polinómica

En la Sección 2.2 se presentó la *regresión lineal* como una solución que garantiza baja complejidad; en caso de alcanzar un riesgo empírico bajo hay ciertas garantías de buen desempeño. El problema surge cuando el riesgo empírico no alcanza un valor suficientemente satisfactorio, denotando que la complejidad del modelo es insuficiente.

El objetivo más ambicioso, en un problema de regresión, es estimar la regresión asociada a la esperanza condicional $\mathbb{E}[Y|X=x]$, ya que esta minimiza el *error cuadrático medio* (Prop. 1.8). Independientemente de la complejidad de la esperanza condicional, el **teorema de Taylor** indica que cualquier función se puede aproximar como una combinación lineal de coeficientes polinómicos. Es entonces que surge la **regresión polinómica**; utilizar una aproximación lineal sobre un **mapa polinómico** de los *predictores* vecto-

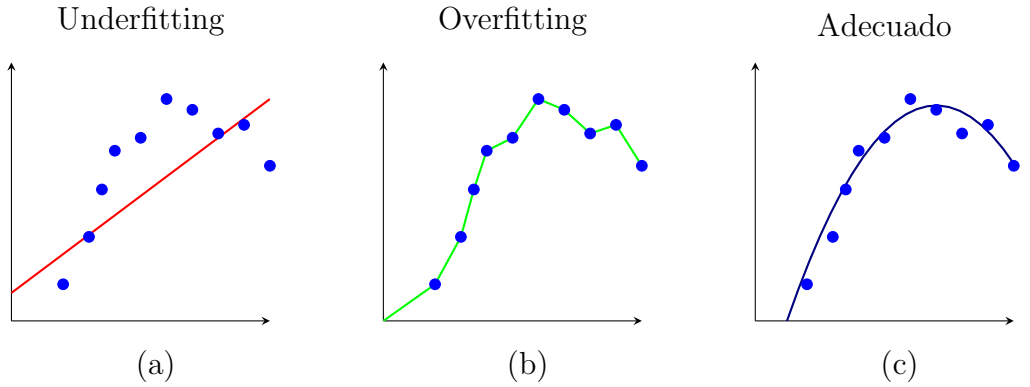


Figura 2.6: Comparación entre regresores denotando (a) un ejemplo de subajuste, (b) uno de sobreajuste, y (c) uno con un ajuste razonable.

rizándolos de la siguiente manera:

$$\mathbf{X} = \begin{pmatrix} 1 & x_{1,1} & x_{1,2} & x_{1,1}^2 & x_{1,2}^2 & x_{1,1}x_{1,2} \\ 1 & x_{2,1} & x_{2,2} & x_{2,1}^2 & x_{2,2}^2 & x_{2,1}x_{2,2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n_{\text{tr}},1} & x_{n_{\text{tr}},2} & x_{n_{\text{tr}},1}^2 & x_{n_{\text{tr}},2}^2 & x_{n_{\text{tr}},1}x_{n_{\text{tr}},2} \end{pmatrix} \tag{2.20}$$

donde $x_{i,k}$ es la muestra i-ésima de la variable k-ésima. El ejemplo presentado en (2.20) corresponde a un *mapa polinómico* de orden 2. En general, el mapa polinómico de orden ν contendrá todos los productos cruzados de las variables hasta orden ν inclusive.

Propiedades 2.1 *La vectorización correspondiente a un mapa polinómico de orden ν sobre d predictores posee una cantidad de columnas $\binom{d+\nu}{\nu}$.*

Demostración 2.1 (Prop. 2.1) *Los coeficientes de la vectorización de un mapa polinómico pueden escribirse como*

$$1^{a_0} \cdot \prod_{k=1}^{d} x_{i,k}^{a_k} \quad \text{para algunos} \quad a_k \in \mathbb{N}_0, \quad \sum_{k=0}^{d} a_k = \nu \tag{2.21}$$

donde los a_k efectúan una **codificación**. *Para determinar la cantidad de columnas del mapa, basta con resolver el problema combinatorio de cuantas maneras posibles se pueden elegir los a_k. En (2.20), las codificaciones son de la forma $(2,0,0)$, $(1,1,0)$, $(1,0,1)$, $(0,2,0)$, $(0,0,2)$ y $(0,1,1)$. Para resolver el problema en general, es conveniente estudiarlo como un problema de conteo de objetos indistinguibles. Se cuenta con ν puntos para repartir en $k+1$ recipientes (por lo tanto es necesario k separadores). Por ejemplo, el $(1,1,0)$ de (2.20) será representado como "$\times|\times|$" y el $(0,2,0)$*

como "| × ×|". Es decir que de las $\nu + d$ posiciones se eligen ν para colocar la × obteniendo así $\binom{d+\nu}{\nu}$ columnas.

Es importante destacar que el mapa polinómico entrega predictores en magnitudes no comparables. Por ejemplo, si una variable estaba medida en m, se incorporarán nuevas variables medidas en m^2, m^3, etc. Por ese motivo es indispensable luego de utilizar un mapa polinómico normalizar[5]. El siguiente ejemplo ilustra esta idea.

Ejemplo 2.3 *Preprocesar el siguiente conjunto de datos, para iniciar el entrenamiento de una regresión polinómica de orden 2. ¿Que cantidad de parámetros tendrá el modelo?*

x_1	*1.2*	*0.8*	*1.0*
x_2	*2.3*	*1.3*	*1.6*

Un mapa polinómico de orden 2 con 2 variables generará un mapa de $\binom{4}{2} = 6$ columnas (y por lo tanto tendrá 6 parámetros la regresión lineal posterior). Las columnas serán: la comunas de unos, las columnas x_1, x_2, x_1^2, x_2^2 y x_1x_2. Calculando las columnas restantes se obtiene

x_1	x_2	x_1^2	x_2^2	x_1x_2
1.2	2.3	1.44	5.29	2.76
0.8	1.3	0.64	1.69	1.04
1.0	1.6	1.0	2.56	1.6

El siguiente paso es normalizar, ya que el mapa polinómico genera variables incomparables en términos de unidad. Calculando la media y el desvío de cada columna, utilizando (2.12), se obtiene:

	x_1	x_2	x_1^2	x_2^2	x_1x_2
μ	1.0	1.73	1.03	3.18	1.8
σ	0.16	0.42	0.33	1.53	0.72

Aplicando la normalización $\frac{x-\mu}{\sigma}$ e incorporando la columna de unos se obtiene el mapa pedido.

[5]La columna de unos no se normaliza Esta solamente se incorpora como requisito para efectuar posteriormente la regresión lineal (para que exista el término constante).

bias	x_1	x_2	x_1^2	x_2^2	$x_1 x_2$
1	1.22	1.35	1.26	1.38	1.34
1	-1.22	-1.03	-1.18	-0.97	-1.06
1	0.0	-0.32	-0.08	-0.4	-0.28

El problema de cambiar la regresión lineal por la polinómica es que se pierde la característica de baja complejidad, pudiendo así tener problemas de *overfitting* como se mencionó en la Sección 2.1. En la Fig. 2.6 se muestran ejemplos de regresores, mostrando posibles problemas tanto de subajuste (complejidad insuficiente) como sobreajuste (complejidad excesiva). Recordando (2.1), mientras que el *underfitting* lo podemos detectar por un alto riesgo empírico el *overfitting* requiere conocer el *gap de generalización*, magnitud que no puede conocerse de forma exacta por depender de valores esperados. Es entonces que surge la necesidad de tener diferentes conjuntos de datos.

2.4.1. Conjuntos de datos

En general se necesitan datos para efectuar distintas funciones: para entrenar, para ajustar la relación de compromiso sesgo/varianza y simplemente para estimar el riesgo esperado.

- **Conjunto de entrenamiento** (*train set*): Datos utilizados para minimizar el riesgo empírico. Sobre estos se produce el "aprendizaje". Las variables definidas a partir de este conjunto se llaman parámetros.
- **Conjunto de validación** (*validation or development set*): Datos utilizados para comparar modelos. Las variables definidas a partir de este conjunto (o definidas previas al entrenamiento) se llaman hiper-parámetros.
- **Conjunto de testeo** (*test set*): Datos utilizados para evaluar el desempeño final del algoritmo. Su única función es presentar estimadores insesgados de las métricas de error y no es imprescindible.

En el caso de una regresión polinómica básica, se pueden entrenar varios modelos para diferentes valores de ν (cada uno de ellos entrenado con el conjunto de entrenamiento) y se elegirá el valor de ν que minimice el error medido con el conjunto de validación. Una vez elegido el ν y teniendo el modelo entrenado para ese valor, se procede a medir el error con el conjunto de testeo (estimando así el riesgo esperado) para evaluar si la decisión fue o no satisfactoria. El mismo procedimiento se puede efectuar para elegir el valor del *learning rate* α o cualquier otro hiper-parámetro involucrado.

En la bibliografía y en la documentación muchas veces se habla de conjunto de testeo en general para referirse a todos los datos no usados durante el entrenamiento. Pero es importante tener en consideración las diferencias entre validación y testeo: la validación se utiliza para tomar decisiones sobre el diseño del modelo, eligiendo hiper-parámetros. Una vez que se utilizó dicho conjunto de datos para tomar una decisión, las estimaciones de error dejan de ser insesgadas: como se utilizaron los datos de validación para elegir el modelo, analizar el desempeño del algoritmo con ellos mismo puede dejar de ser representativo. Sin embargo, si la validación no es minuciosa, puede considerarse aceptable utilizarla como métrica de error (aunque sesgada, la estimación puede ser suficiente). Contar con el conjunto de testeo puede ser prescindible, su única función es medir la eficacia final del sistema. Es recomendable usarlo principalmente luego de validaciones exhaustivas o cuando no se efectuaron validaciones en absoluto (y es razonable en desconfiar en el desempeño del sistema). Incluso en el caso de contar con un solo predictor, se puede analizar el desempeño de un algoritmo gráficamente sin necesidad de contar con el conjunto de testeo, como es el caso de la Fig. 2.6.

El problema con generar los diferentes conjuntos de datos es que disminuye la cantidad de datos usado durante el entrenamiento. No hay una regla para asignar proporciones a estos conjuntos, dependerá de cuantos datos se tenga, de la complejidad del modelo y de la dificultad de la tarea. Por ejemplo en el caso de regresión lineal, basta con tener solamente un conjunto de entrenamiento, ya que se asume que el gap de generalización es pequeño por la complejidad del modelo. En lugar de validar el resultado, simplemente analizo si se alcanzó o no un razonable riesgo empírico.

2.4.2. Regularización

El *problema de sesgo* se detecta cuando el riesgo empírico de entrenamiento es grande comparado con el supuesto error bayesiano. El mismo se soluciona aumentando la complejidad del modelo. En cambio, *problema de varianza* se detecta cuando el riesgo empírico de validación es grande comparado con el de entrenamiento, y la mejor manera de combatirlo es aumentando la cantidad de muestras. Por desgracia esto no siempre es posible, ya sea por la dificultad de obtener los datos o de procesarlos. Las técnicas destinadas a combatir el *overfitting* sin incorporar nuevos datos se denominan **regularización**.

Existen diferentes técnicas de regularización: generar datos sintéticos para incorporarlos, incorporar ruido a las muestras para dificultar el sobreajuste, limitar la complejidad del modelo, entre otras. Pero quizás la más importante es el agregado de un término de penalización al riesgo a minimizar durante el entrenamiento. Básicamente se busca perturbar la optimización del modelo, minimizando en lugar del riesgo empírico, el **riesgo**

regularizado:

$$J(\theta) = \frac{1}{n_{\text{tr}}} \sum_{i=1}^{n_{\text{tr}}} \ell(x_i, y_i) + \lambda \cdot R(\theta) \tag{2.22}$$

donde $\lambda \geq 0$ es un hiper-parámetro que controlará el *overfitting*, y $R(\theta)$ recibe el nombre de **regularizador**. El resigo regularizado define una función a optimizar durante el entrenamiento, rara vez es utilizado en la etapa de testeo. La incorporación del regularizador tiene por objeto acercar la función a optimizar $J(\theta)$ al riesgo esperado (2.1). En ese sentido, un buen regularizador será aquel donde $\lambda R(\theta)$ sea representativo del *gap de generalización*. Cuando $\lambda = 0$ la regularización es ignorada, mientras que para λ muy grandes lo que es ignorado son los datos.

El regularizador más utilizado en regresión polinómica es el denominado **L2**, *weight decay* o *regularización de Tikhonov*: $R(w, b) = \frac{1}{n_{\text{tr}}} \|w\|^2$. Una primera interpretación de esta selección es que el incorporar la norma cuadrática de los pesos w en la función a minimizar tenderá a "apagar" coeficientes $w_j \approx 0$, y por lo tanto la complejidad efectiva del modelo bajará, tendiendo a disminuir el *overfitting*. Otra posible interpretación es que limitando el valor de w lo que estamos haciendo es tendiendo a limitar el máximo valor posible de la función costo $\ell(x, y) \leq L$, y por lo tanto limitando el *gap de generalización* y por lo tanto el *overfitting*.

$$\mathbb{E}[\ell(X, Y)] - \frac{1}{n_{\text{tr}}} \sum_{i=1}^{n_{\text{tr}}} \ell(x_i, y_i) \leq L \tag{2.23}$$

En la práctica, en regresión polinómica se suele dejar un ν fijo dándole un margen suficiente de complejidad al modelo y controlar el overfitting con el hiper-parámetro de regularización. Es decir que la etapa de validación se efectuará sobre λ en lugar de ν. Esto es beneficioso porque permite un incremento controlado de la regularización ya que mientras $\lambda \in \mathbb{R}$, $\nu \in \mathbb{N}$ necesita dar saltos discretos.

Ejemplo 2.4 *Hallar una solución matricial al problema de regresión lineal sin sesgo y con regularización L2. Analizar el comportamiento para algoritmos no regularizados y muy regularizados.*

El riesgo regularizado para este problema es de la forma

$$J(w) = \frac{1}{n_{\text{tr}}} \|\mathbf{X}w - \mathbf{y}\|^2 + \frac{\lambda}{n_{\text{tr}}} \|w\|^2 \tag{2.24}$$

Basta con analizar la primera derivada (gradiente) y la segunda derivada (matriz Hessiana) respecto a w (para más información sobre derivadas respecto vectores/matrices ver [9]).

$$\nabla J(w) = \frac{2}{n_{\text{tr}}} \mathbf{X}^T (\mathbf{X}w - \mathbf{y}) + \frac{2\lambda}{n_{\text{tr}}} w, \qquad \mathcal{H}_J(w) = \frac{2}{n_{\text{tr}}} (\mathbf{X}^T \mathbf{X} + \lambda \cdot \mathbf{I}) \tag{2.25}$$

donde la matriz Hessiana es claramente definida positiva (y por lo tanto inversible). Esto

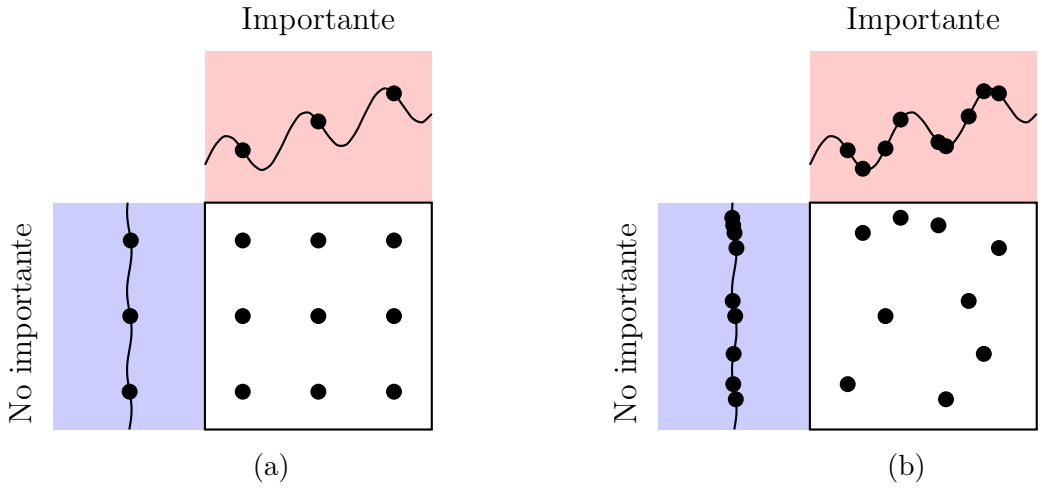


Figura 2.7: Ejemplos de grilla para validar dos hiper-parámetros, de los cuales solo uno importa fuertemente sobre el error. (a) Una grilla regular, y (b) una grilla aleatoria.

implica que el problema es convexo y por lo tanto igualando a cero el gradiente equivale a minimizar el riesgo regularizado. Se puede despejar entonces:

$$w = (\mathbf{X}^T\mathbf{X} + \lambda \cdot \mathbf{I})^{-1}\mathbf{X}^T\mathbf{y} \tag{2.26}$$

Por un lado, si $\lambda = 0$ se obtiene como solución $w = \mathbf{X}^\dagger\mathbf{y}$ donde $\mathbf{X}^\dagger = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ es la *pseudoinversa*. Por el otro si λ es muy grande (lo suficiente para que $\lambda \cdot \mathbf{I}$ enmascare a $\mathbf{X}^T\mathbf{X}$ pero no tanto para que $w \approx 0$) se obtiene $w \approx \frac{1}{\lambda}\mathbf{X}^T\mathbf{y}$. Esto quiere decir que un algoritmo regularizado interpreta la transpuesta como la operación inversa. Esta operación tiene un muy bajo costo computacional por lo que es utilizada para inversión de problemas físicos bajo el nombre de *Linear Back Projection* [15].

2.4.3. Etapa de validación

La etapa de validación está basada en el enfoque de *prueba error*. Por ejemplo, si se desea validar el hiperparámetro λ se deberán realizar V entrenamientos diferentes, donde V es la cantidad de puntos de λ que deseo probar. Finalmente se termina decidiendo por el valor de λ que menor error de validación genere. El problema principal de este enfoque aparese cuando se desea validar varios hiper-parámetros. Por ejemplo, si queremos probar V valores de λ y V valores de α, necesitaríamos realizar V^2 simulaciones. La etapa de validación es computacionalmente costosa, y es necesario tener en cuenta algunas consideraciones para no sufrir tanto esta particularidad. La cantidad de entrenamientos a efectuar crece exponencialmente con la cantidad de hiper-parámetros a validar, fenómeno que recibe el nombre de **maldición de la dimensionalidad**. Este fenómeno está presente también con las muestras de entrenamiento: La necesidad de muestras crece

exponencialmente con la cantidad de predictores.

Supongamos que se desean validar dos hiper-parámetros, probando V valores para cada uno y efectuando V^2 entrenamientos en total. En el caso de que solamente uno de los hiper-parámetros sea influyente sobre el error (a priori esta información era desconocida), se terminan realizando V^2 entrenamientos para probar solo V valores del hiper-parámetro relevante. Este fenómeno puede verse en la Fig. 2.7a, donde en los márgenes se muestra una curva de como varía el error de validación con los respectivos hiper-parámetros. En este ejemplo, vemos que se debieron realizar 9 entrenamientos para probar efectivamente solo 3 valores del hiper-parámetro relevante.

Es por este motivo que al validar dos o más hiper-parámetros se opta por efectuar una grilla aleatoria, como se muestra en la Fig. 2.7b. Al elegir valores al azar, es muy probable que no haya valores repetidos y por lo tanto se terminen probando V^2 valores del hiper-parámetro relevante. En este ejemplo, observando el margen superior, se puede ver que se probaron hiper-parámetros con errores más bajos.

Otra recomendación a la hora de validar parámetros por grilla aleatoria es efectuarla por etapas. Por ejemplo, si cuento con la posibilidad de realizar 300 entrenamientos, quizás sea conveniente primero validar con 100 de ellos. Con esos resultados uno puede limitar la zona donde el error de validación deba ser más chico, y hacer una segunda búsqueda (con otros 100 entrenamientos) dentro de ese espacio limitado. Repitiendo por tercera vez con los 100 entrenamientos restantes logré concentrar muchos valores al rededor de la zona de mayor interés.

Por último, hay muchos casos donde la cantidad de datos observados es limitada y no es posible reservarse un conjunto de validación. Es entonces cuando surgen técnicas un poco más sofisticadas.

2.4.3.1. Validación Cruzada

La etapa de validación es crítica en modelos de alta complejidad, ya que los hiper-parámetros asociados con la regularización suelen ser muy sensibles. Este es el caso de λ en regresión polinómica, el cuál suele necesitar ser validado. El problema es que definir un conjunto de datos de validación, como se mencionó en la Sección 2.4.1, muchas veces es prohibitivo debido a que no se cuenta con suficientes datos. Sin embargo, existen algunas técnicas para validar hiper-parámetros sin necesidad de definir un conjunto de validación, repitiendo el entrenamiento en múltiples ocasiones (y por lo tanto pagando un costo computaional). Estas técnicas se conocen como **validación cruzada**.

Leave-one-out cross-validation (LOOCV) es el método básico de validación cruzada. Propone reservar una sola muestra para validación y entrenar con el resto. Este proceso se repite para cada muestra, realizando n_{tr} entrenamientos. El error entonces se

puede estimar promediando el error de validación de todos los entrenamientos. Para utilizar este método para validar un hiper-parámetro se requiere realizar una gran cantidad de entrenamientos. Por ejemplo, supongamos que se desean probar V valores diferentes de λ en un algoritmo de regresión polinómica. Por cada uno de estos valores, se deberán hacer n_{tr} entrenamientos, realizando en total $V \cdot n_{\text{tr}}$ repeticiones (en contraste con definir un conjunto de validación que solamente necesita V entrenamientos). Finalmente se elige el λ que genere menor error de validación.

Una solución intermedia entre LOOCV y definir un conjunto de validación es el método conocido como **K-Folds**. En él, se propone separar el conjunto de datos de entrenamiento en K paquetes para entrenar con $K-1$ de ellos y validar con el restante. Se repite el procedimiento K veces usando como conjunto de validación siempre un paquete diferente, y se define el error de validación total como el promedio del error de validación de cada experimento. De esta manera, para probar V valores de un hiper-parámetro se deben realizar $V \cdot K$ entrenamientos, donde $1 < K \leq n_{\text{tr}}$ (LOOCV coincide con $K = n_{\text{tr}}$). Un valor grande de K requiere efectuar muchos entrenamientos, pero un valor chico de K reduce demasiado la cantidad de muestras efectivas de entrenamiento. Si cada paquete contiene $\frac{n_{\text{tr}}}{K}$ muestras, el entrenamiento efectivo se hará con $\frac{(K-1)\cdot n_{\text{tr}}}{K}$ muestras. La decisión de K dependerá mucho de la cantidad de muestras con las que se cuente.

3

Clasificación en Inteligencia Artificial

Utilizar un algoritmo para resolver una tarea es importante. Pero imaginemos el potencial de abrir el algoritmo y entender los por menores de su razonamiento. Quizás mirando el algoritmo nos veamos a nosotros mismos.

TODO

3.1. Regresión Logística

3.1.1. Regresión Logística Binaria

3.1.2. Regresión Logística Multiclase

3.2. Análisis del Discriminante

3.3. Vecinos más Cercanos

3.4. Máquina de Vectores Soporte

3.5. Árboles de Decisión

3.5.1. Bosques Aleatorios

4

Aprendizaje no Supervisado

Los datos son la materia prima de la inteligencia artificial. Sacarle el mayor provecho posible a los mismos es el objetivo. No se puede dar el lujo de desperdiciar nada. Ahí radica el verdadero potencial de estos algoritmos.

TODO

4.1. Análisis de Componentes Principales

4.2. K-Means

4.3. Algoritmo Expectación-Maximización

4.3.1. Análisis de Factores

5

Procesamiento de Datos orientado a Aplicaciones Específicas

El algoritmo no manda, castiga ni interroga, sino que sugiere, optimiza y predice. Esa suavidad operativa no lo vuelve menos violento, sino que más eficaz, porque captura al sujeto desde sus deseos y lo vuelve cómplice de su propia subordinación.

TODO

5.1. Procesamiento de Audio

5.1.1. Espectrograma

5.1.2. Coeficientes Mel-Cepstrum

5.2. Procesamiento de Texto

5.3. Sistemas de Recomendación

5.4. Ingeniería de Características

5.4.1. Test de Independencia Chi-Cuadrado

5.4.2. Tests ANOVA

Comparación medias de normales varianza conocida Comparación medias de normales varianza desconocida Coparación de varianza de normales Asintótico

6

Modelos Bayesianos

Es tan cierto que la inteligencia artificial solamente reproduce datos, como que los datos lo reproducen todo.
Ley de los Grandes Números.

Tras una implementación particularmente difícil de un algoritmo, los programadores suelen probar el código con un ejemplo trivial. Si lo supera comienzan a probar dicho código con ejemplos más y más complejos. Cuando el código ya superó cuatro o cinco pruebas de éstas, el programador empieza a creer que posiblemente no haya errores en ese código. En esto consiste el pensamiento bayesiano: actualizar las creencias tras considerar nueva evidencia [6].

6.1. Inferencia Bayesiana

La estadística bayesiana, discutida en la Sección 1.2.4, posee características particulares. Su filosofía radica en buscar verdades en contextos de incertidumbre, modelando no solo el problema a resolver sino también nuestra ignorancia sobre el mismo [7]. Algunas de las características técnicas son:

- Sea T una variable aleatoria representativa de los parámetros y las variables no observables del modelo, con distribución a priori $p_T(\theta)$.
- La estadística bayesiana supone una relación causal $T \to \mathbf{X}$, donde $\mathbf{X}$ es cualquier conjunto aleatorio de muestras a observar.
- La relación anterior implica la independencia entre las muestras *cuando se conoce el parámetro*. Es decir que la verosimilitud de una muestra puede escribirse como $p_{\mathbf{X}|T=\theta}(\mathbf{x}) = \prod_{i=1}^{n} p_{X|T=\theta}(x_i)$.
- La *distribución a posteriori* es proporcional al producto de la *prior* y la verosimilitud

$$p_{T|\mathbf{X}=\mathbf{x}}(\theta) \propto p_T(\theta) \cdot \prod_{i=1}^{n} p_{X|T=\theta}(x_i) \tag{6.1}$$

- Por último, se define la distribución predictiva bayesiana como:

$$p_{X_{\text{test}}|\mathbf{X}=\mathbf{x}}(x_{\text{test}}) = \int_{\Theta} p_{X|T=\theta}(x_{\text{test}}) p_{T|\mathbf{X}=\mathbf{x}}(\theta) d\theta = \mathbb{E}\left[p(x_{\text{test}}|T)|\mathbf{X}=\mathbf{x}\right] \tag{6.2}$$

  donde X_{test} es una variable aleatoria no vista en el conjunto de entrenamiento $\mathbf{X}$.

A continuación se presenta un ejemplo de como calcular analíticamente este tipo de distribuciones.

Ejemplo 6.1 *Lucas dispara a un blanco y el disparo impacta en un punto aleatorio* $(X, 0)$ *con* X *(en decímetros) una variable aleatoria con distribución normal de media nula y varianza* $1/\tau$*, donde* τ *representa la precisión de Lucas. A priori la precisión* τ *tiene una distribución chi-cuadrado de* 8 *grados de libertad. Lucas tiro* 10 *veces al blanco y observó* $\sum_{i=1}^{10} x_i^2 = 17$*. En virtud a la información muestral,*

1. *Hallar la distribución* a posteriori.
2. *Hallar la distribución predictiva.*

Como primer paso en un problema bayesiano, hay que comenzar planteando la distribución *a posteriori*. En este caso evitaremos las constantes de proporcionalidad:

$$p_{T|\mathbf{X}=\mathbf{x}}(\tau) \propto p_T(\tau) \cdot \prod_{i=1}^{n} p_{X|T=\tau}(x_i) \propto \tau^3 e^{-\tau/2} \mathbf{1}\{\tau > 0\} \cdot \prod_{i=1}^{10} \sqrt{\tau} e^{-\frac{\tau}{2} x_i^2} \tag{6.3}$$

donde se utilizó la información de que la versoimilitud es normal y la distribución *a priori* es chi-cuadrado (véase Cuadro 1.2). Utilizando el dato muestreal del enunciado, podemos observar que $p_{T|\mathbf{X}=\mathbf{x}}(\tau) \propto \tau^8 e^{-9\tau} \mathbf{1}\{\tau > 0\}$, es decir que la variable se distribuye *a posteriori* como $T|_{\mathbf{X}=\mathbf{x}} \sim \Gamma(9, 9)$ (véase Cuadro 1.2). La distribución predictiva es de la forma

$$p_{X_{\text{test}}|\mathbf{X}=\mathbf{x}}(x_{\text{test}}) = \int_{\Theta} p_{X|T=\tau}(x_{\text{test}}) p_{T|\mathbf{X}=\mathbf{x}}(\tau) d\tau \propto \int_0^{\infty} \sqrt{\tau} e^{-\frac{\tau}{2} x_{\text{test}}^2} \cdot \tau^8 e^{-9\tau} d\tau \tag{6.4}$$

Reconociendo el núcleo de la integral, se puede observar que el mismo es proporcional a la densidad de una $\Gamma(\nu, \beta)$. Sabiendo que por ser densidad debe integrar 1:

$$p_{X_{\text{test}}|\mathbf{X}=\mathbf{x}}(x_{\text{test}}) \propto \int_0^{\infty} \tau^{\frac{17}{2}} e^{-\tau\left(9+\frac{x_{\text{test}}^2}{2}\right)} d\tau \propto \left(9 + \frac{x_{\text{test}}^2}{2}\right)^{-\frac{19}{2}} \tag{6.5}$$

donde se utilizó $\nu = \frac{19}{2}$ y $\lambda = 9 + \frac{x_{\text{test}}^2}{2}$. Es decir que la densidad predictiva es proporcional a $p_{X_{\text{test}}|\mathbf{X}=\mathbf{x}}(x_{\text{test}}) \propto \left(1 + \frac{x_{\text{test}}^2}{18}\right)^{-\frac{18+1}{2}}$, y por lo tanto es una t-student de 18 grados de libertad $X_{\text{test}}|_{\mathbf{X}=\mathbf{x}} \sim t_{18}$ (véase Cuadro 1.2).

Tanto *a priori* como *a posteriori*, la variable T es una Gamma (la chi-cuadrado es un caso particular de Gamma). Este fenómeno de mantenerse dentro de una familia ocurre por cierta compatibilidad entre la distribución *a priori* y la verosimilitud (en este caso una normal). Cuando se da este fenómeno se dice que la distribución *a priori* es una **conjugada a priori**. Las soluciones analíticas suelen proponer conjugadas, como distribución

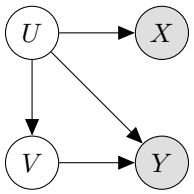


Figura 6.1: Ejemplo de red bayesiana, donde se puede apreciar tanto relaciones de causalidad como independencia.

a priori, ya que así garantizan que la distribución *a posteriori* pertenezca a una familia conocida (la misma que la distribución *a priori*). Es simplemente una recomendación para hacer sencillos (o al menos factibles) los cálculos. Los problemas de la estadística bayesiana, pueden representarse fácilmente con grafos denominados **redes bayesianas**.

6.1.1. Redes Bayesianas

Se denomina *modelo gráfico* a todo modelo probabilístico capaz de representarse con un grafo. En particular el modelado bayesiano es un modelo gráfico. Existen diferentes grafos que se pueden utilizar para describir los modelos, siendo las *redes bayesianas* el estándar en este tipo de estadística.

Se denomina red bayesiana a un grafo acíclico dirigido que representa la relación de causalidad e independencia de sus variables [7, Sección 3.5]. Por un lado, la causalidad está determinada por la dirección de sus vínculos y presenta una configuración a implementar para generar muestras del modelo. Por otro lado, dos variables aleatorias cualesquiera son condicionalmente independientes dados los valores de sus padres causales (y por lo tanto las raíces son independientes).

En la Fig. 6.1 puede verse un ejemplo de red bayesiana, donde el color gris hace referencia a las variables observables. La causalidad nos indica como podríamos simular el modelo:

- Muestrear U, ya que es raíz del grafo: $u \leftarrow U \sim p_U$.
- Muestrear $X|U = u$, ya que a su nodo le llega una conexión desde U: $x \leftarrow X \sim p_{X|U=u}$.
- Muestrear $V|U = u$: $v \leftarrow V \sim p_{V|U=u}$.
- Muestrear $Y|U = u, W = w$: $y \leftarrow Y \sim p_{Y|U=u,W=w}$.

Este análisis es conceptual, en la práctica no se habitual generar muestras de variables observables. Además, el procedimiento antes descripto nos define la factorización de la

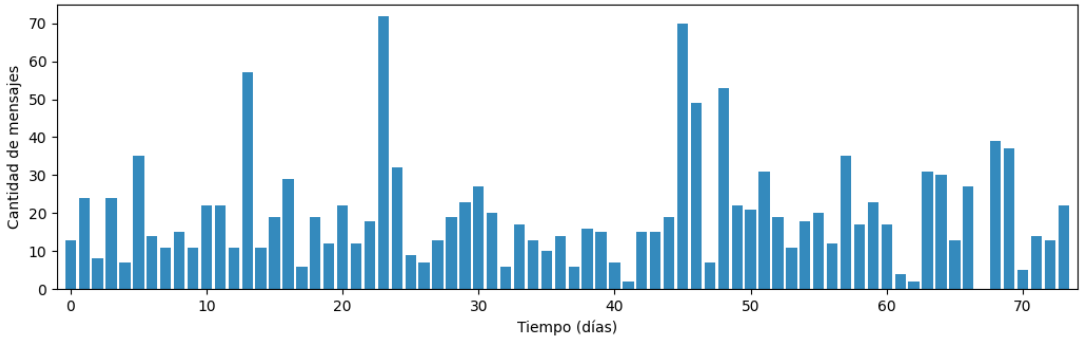


Figura 6.2: Ejemplo tomado de [6].

distribución conjunta:

$$p_{XYUV}(x, y, u, v) = p_U(u) \cdot p_{X|U=u}(x) \cdot p_{V|U=u}(v) \cdot p_{Y|U=u,W=w}(y) \tag{6.6}$$

Esta factorización nos impone condiciones de independencia. Por ejemplo, dado $U = u$, X y V son independientes ya que:

$$p_{XV|U=u}(x, v) = \frac{\int_y p_{XYUV}(x, y, u, v)dy}{p_U(u)} = p_{X|U=u}(x) \cdot p_{V|U=u}(v) \tag{6.7}$$

6.1.2. Ejemplo de Modelo Bayesiano

La esencia de la estadística bayesiana es interpreta la probabilidad como una medida de credibilidad en un evento, es dice buscar verdades en contexto de incertidumbre. La falta de certeza en las ciencias empíricas, lejos de volverlas absurdas, permite evitar afirmar más de lo que se sabe sin ocultar lo que efectivamente se conoce. Los métodos bayesianos no solo pueden adaptarse a intentar resolver los mismos problemas que la estadística frecuentista (por ejemplo predicciones), sino que también pueden intentar resolver problemas donde la estadística clásica es insuficiente o iluminar el sistema subyacente con un modelado más flexible. Veamos el siguiente ejemplo [6].

Un usuario proporciona una serie de recuentos diarios de mensajes de whatsapp enviados. Tiene curiosidad por saber si los hábitos de envío de mensajes han cambiado con el tiempo. En la Fig. 6.2 puede verse la cantidad de mensajes recibidos en los diferentes días. La hipótesis es que el arribo de mensajes tenía cierta tendencia y en algún momento cambio a otra diferente. Se desea plantear un modelo capaz de representar esta información.

La cantidad de mensajes en un día deberá ser modelada como una variable discreta cuyos átomos es $\mathbb{N}_0$. Por ejemplo, se elegirá una $X_i \sim \text{Poi}(\lambda_i)$. Esta será la única variable observable del modelo. Analizando los datos, parecería que el valor de λ_i aumenta en algún momento durante las observaciones. ¿Cómo podemos representar matemáticamente esta observación? Supongamos que algún día τ durante el período de observación, el parámetro

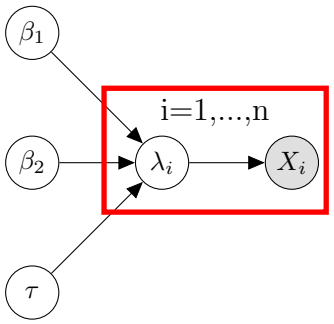


Figura 6.3: Red bayesiana del modelo propuesto para caracterizar el arribo de mensajes.

λ_i se incrementa repentinamente. Entonces realmente tenemos dos tasas: una para el período anterior a τ y otro para el resto del período:

$$\lambda_i = \begin{cases} \beta_1 & i < \tau \\ \beta_2 & i \geq \tau \end{cases} \tag{6.8}$$

Tanto β_1 como β_2 toman valores reales no negativos. Por ejemplo, se puede elegir $\beta_1, \beta_2 \sim \mathcal{E}(\alpha)$. Es importante notar que se definen como variables aleatorias independientes e idénticamente distribuidas (para evitar sesgar a alguna tasa). En este punto, uno podría definir α como variable aleatoria o asignarle un valor fijo. Asignar un valor fijo para α sería menos influyente en el modelo que haberlo asignado en los λs, básicamente por estar más lejos de la variable observable a nivel grafo. Por el contrario, sería más influyente que suponerlo una variable aleatoria y asignarle un valor a los parámetros de esta nueva variable. Para este análisis, donde solamente se quiere estudiar el cambio de tasa, es suficiente con fijar un valor razonable. Notar que

$$\mathbb{E}[X_i] = \mathbb{E}\left[\mathbb{E}[X_i|\lambda_i]\right] = \begin{cases} \mathbb{E}\left[\beta_1\right] & i < \tau \\ \mathbb{E}\left[\beta_2\right] & i \geq \tau \end{cases} = \frac{1}{\alpha} \tag{6.9}$$

por lo que $\alpha = \frac{1}{\frac{1}{n}\sum_{i=1}^{n} X_i}$ es un buen candidato a valor. La última variable a definir es τ. Debido a la varianza de los datos, es difícil caracterizarla en detalle. Podemos asignar entonces la creencia menos informativa posibles *a priori* $\tau \sim \mathcal{U}\{1:n\}$, donde se asumirá τ independiente de β_1 y β_2.

En la Fig. 6.3 puede verse la red bayesiana del modelo descripto. La potencia del modelado bayesiano radica en poder modelar fácilmente situaciones prácticas donde la estadística clásica parece quedarse corta. El problema está en que calcular la distribución predictiva de un modelo de estas características es computacionalmente inviable, al menos de forma exacta. En el resto del capítulo iremos analizando las fortalezas y debilidades de

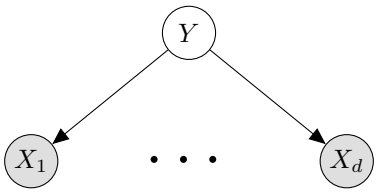


Figura 6.4: Red bayesiana del modelo gráfico bayes naive. Los predictores son independientes conocida la clase.

algunos modelos gráficos, llendo de las más simples a las más complejas incrementando el potencial poco a poco.

6.2. Bayes Naive

Uno de los modelos gráficos más fáciles de analizar son los conocido como **bayes naive**. La hipótesis naive en un problema de clasificación modela una relación de causalidad $Y \to \mathbf{X}$, donde las diferentes componentes $X_j|_{Y=k}$ son independientes, tal como puede verse en la red bayesiana de la Fig. 6.4. En este tipo de modelos, la probabilidad asignada a cada clase, cuando se observa una muestra, es de la forma:

$$p(y|\mathbf{x}) \propto p(y) \prod_{j=1}^{d} p(x_j|y) \tag{6.10}$$

Estos modelos se llaman **bayes naive** por combinar la hipótesis naive con la regla de bayes de (6.10). No es necesariamente un modelo bayesiano en el sentido de interpretar los parámetros como variables aleatorias ni mucho menos calcular una predictiva. De hecho, el cálculo de una predictiva en bayes naive se suele volverse prohibitivo ya que implica resolver la integral sobre una productoria [7, Sección 9.3].

Está claro que difícilmente en la práctica las componentes serán independientes dada la clase a la que pertenecen, de ahí radica el nombre de *naive* (ingenuo). Sin embargo, proponer modelos simples puede ser beneficioso incluso si no se cumplen el la práctica, ya que este tipo de modelos suele poseer menos parámetros y por lo tanto necesita menos datos para ser entrenado.

La simpleza de bayes naive, radica en la separación del modelo $X_j|_{Y=k}$ para cada clase. Nos permite modelar cada clase por separado para finalmente combinarlas por una $p(y)$ usualmente estimada con la proporción de muestras de entrenamiento de esa clase: $Y \sim \text{Cat}(\{c_1, \cdots, c_K\})$ con $c_k = \frac{\#\{y_i=k\}}{n}$. A continuación vamos a estudiar dos modelos bayes naive que utilizan estimadores puntuales: el primero de la estadística clásica y el segundo del modelado bayesiana propiamente dicha.

6.2.1. Bayes Naive Gaussiano

El modelo **bayes naive gaussiano** (GNB) propone modelar el problema como una mezcla de gaussianas naives: $Y \sim \text{Cat}(\{c_1, \cdots, c_K\})$ y $X_j|_{Y=k} \sim \mathcal{N}(\mu_j^{(k)}, \sigma_j^{2(k)})$, donde los parámetros serán estimados utilizando los estimadores estándar.

Este modelo está muy relacionado con los modelos de LDA y QDA estudiados anteriormente. Los tres métodos asumen un modelo de mezclas de gaussianas, su diferencia radica en las hipótesis que hacen sobre las matrices de covarianza, como puede verse en la Fig. 6.5. Sea Σ_k la matriz de covarianza asociadas a la clase k-ésima.

- QDA acepta como covarianza cualquier conjunto de matrices definidas positiva. Suelen estimarse como $\Sigma_k = \frac{1}{|\mathcal{D}_k|-1} \sum_{x \in \mathcal{D}_k} (x - \mu^{(k)}) \left(x - \mu^{(k)}\right)^T$.
- LDA acepta como coviarianza cualquier matriz (definida positiva) pero todas deben iguales. A partir de las matrices de covarianza de QDA, la covarianza de LDA puede computarse como $\Sigma = \frac{1}{n-K} \sum_{k=1}^{K} (|\mathcal{D}_k| - 1)\Sigma_k$.
- GNB permite tener matrices diferentes pero todas deben ser diagonales diagonales. A partir de las matrices de covarianza de QDA, las covarianzas de GNB pueden computarse como $\Sigma_k = \text{DIAG}\left(\sigma_1^{2(k)}, \cdots, \sigma_d^{2(k)}\right)$.

Por razones computacionales, en el caso de GNB, no se calcula toda la covarianza para luego quedarse con la raíz. En cambio, suele implementarse un cálculo de la forma $\sigma_j^{2(k)} = \frac{1}{|\mathcal{D}_k|-1} \sum_{x \in \mathcal{D}_k} (x_j - \mu_j^{(k)})^2$, pero el resultado final sería el mismo.

QDA es el modelo más completo de los tres, pero al necesitar estimar mayor cantidad de parámetros requiere mayor cantidad de muestras para ser entrenado correctamente. Entre GNB y LDA, la cantidad de parámetros será superior en un caso o en el otro según la relación entre la cantidad de clases K y la cantidad de predictores d. De cualquier manera, no solo es importante tener en cuenta la cantidad de parámetros, sino que tan bien representadas están las hiótesis en el conjunto de datos con el que se cuenta.

6.2.2. Bayes Naive Multinomial

El modelo **bayes naive multinomial** (MNB) propone modelar para cada clase $Y = k$ un proceso de Bernoulli generalizado $X_j|Y = k \sim \text{Cat}(\{\theta_1^{(k)}, \cdots, \theta_V^{(k)}\})$, donde se respeta la hiótesis naive. Hablamos de proceso en este caso, porque queremos definir un modelo capaz de entregar muestras de dimensión variable. Este tipo de modelado suele utilizarse en procesamiento de texto, donde cada texto posee una cantidad diferente de palabras. A su vez, supondremos que cada clase posee probabilidad c_k, es decir, $Y \sim \text{Cat}(\{c_1, \cdots, c_K\})$.

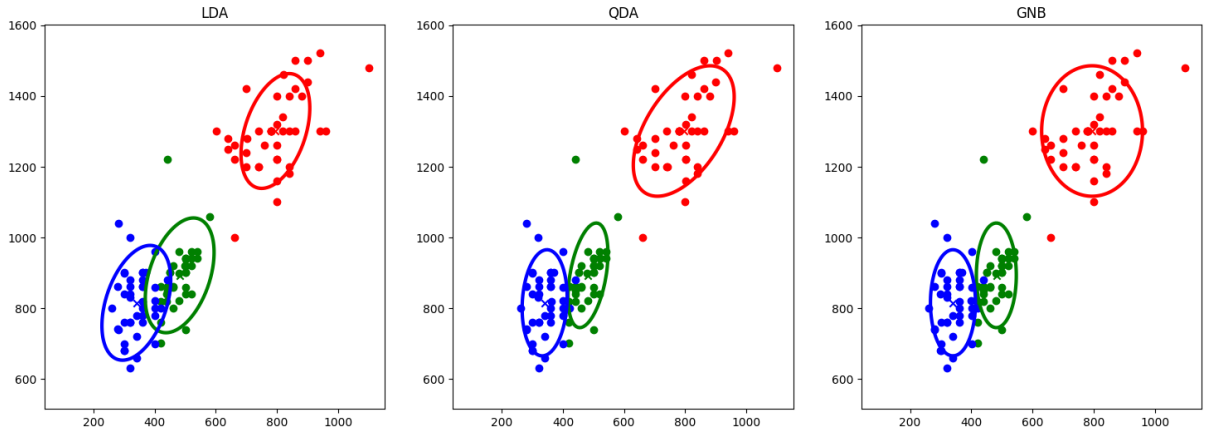


Figura 6.5: Comparación entre LDA, QDA y GNB para un ejemplo de dos dimensiones. La diferencia de sus modelos radica en las covarianzas de cada clase.

Para una muestra d-dimensional $\mathbf{x} = (x_1, \cdots, x_d)$, (6.10) puede escribirse como:

$$p(y|\mathbf{x}) \propto c_y \cdot \prod_{j=1}^{d} \theta_{x_j}^{(y)} = c_y \cdot \prod_{m=1}^{V} \left(\theta_m^{(y)}\right)^{N_m} \tag{6.11}$$

donde N_m representa la cantidad de variables X_j que tomaron el valor m y son estadísticos suficientes en este modelo. El cambio de variable en la productoria nos permite iterar en los valores m que pueden tomar los predictores, en lugar necesitar iterar en la cantidad j de los mismos. Sea $\mathbf{N} = (N_1, \cdots, N_V)$, no solamente se puede notar que $\sum_{m=1}^{V} N_m = d$, sino que también nos define la distribución multinomial de estos estadísticos $\mathbf{N}|_{Y=k} \sim \mathcal{M}_n(d, [\theta_1^{(k)}, \cdots, \theta_V^{(k)}])$. El uso de este estadístico nos permite resolver la inferencia como una solución lineal de la log-probabilidad

$$\log p(y|\mathbf{x}) = \text{cte} + \log(c_y) + \sum_{m=1}^{V} N_m \log\left(\theta_m^{(y)}\right) \tag{6.12}$$

simplemente hay que haber estimado previamente los parámetros c_k y $(\theta_1^{(k)}, \cdots, \theta_V^{(k)})$ para todo $k = 1 \cdots, K$. Estos parámetros serán estimados en la etapa de entrenamiento.

6.2.2.1. Entrenamiento de MNB

La probabilidad de cada clase será simplemente estimada por la proporción de muestras que hay en el conjunto de entrenamiento $c_k = \frac{\#\{y_i=k\}}{n}$, como suele hacerse en cualquier modelo naive. Los θs en cambio serán modelados de forma bayesiana.

Supongamos que contamos con un datos $\{(\mathbf{N}_i, y_i)\}_{i=1}^{n}$, donde cada muestra $\mathbf{N}_i$ tendrá asociada una cantidad de predictores d diferente. Debido a la hipótesis naive, es válido separar los problemas por clase. Es decir que para cada clase k se utilizarán solamente los datos con $\{y_i = k\}$ distribuidos según las probabilidades $(\theta_1^{(k)}, \cdots, \theta_V^{(k)})$. A su vez, dado que las variables N_m cuentan ocurrencias, puedo compactar todas las muestras de

entrenamiento de cada clase en una sola, utilizando suficiencia estadística de las variables categóricas

$$\tilde{N}_m^{(k)} = \sum_{i=1}^{n} N_{i,m} \cdot \mathbf{1}\{y_i = k\} \tag{6.13}$$

Esto quiere decir por ejemplo, que en un problema de procesamiento de texto, podemos pensar que tenemos un solo texto por clase, concatenando todos los textos de dicha clase en uno solo. Esto implica que $(\tilde{N}_1^{(k)}, \cdots, \tilde{N}_V^{(k)})|_{\mathbf{T}_k=(\theta_1^{(k)},\cdots,\theta_V^{(k)})} \sim \mathcal{M}_n(\tilde{d}^{(k)}, [\theta_1^{(k)}, \cdots, \theta_V^{(k)}])$, donde $\tilde{d}^{(k)}$ representa la cantidad de "palabras" que posee el "texto concatenado" de la clase k. Notar que hemos definido una variable aleatoria $\mathbf{T}_k$ representativa de los parámetros del modelo. Dicha variable será modelada *a priori* como $\mathbf{T}_k \sim \text{Dir}([\alpha_1, \cdots, \alpha_V])$, donde el vector aleatorio tendrá distribución Dirichlett.

Definición 6.1 *El vector aleatorio $(\theta_1, \cdots, \theta_V)$ tiene distribución Dirichlett de parámetros $(\alpha_1, \cdots, \alpha_V)$ si su densidad conjunta es de la forma:*

$$p(\theta_1, \cdots, \theta_V) = \frac{\prod_{m=1}^{V} \Gamma(\alpha_m)}{\Gamma\left(\sum_{m=1}^{V} \alpha_m\right)} \left(\prod_{m=1}^{V} \theta_m^{\alpha_m - 1}\right) \cdot \mathbf{1}\left\{\sum_{m=1}^{V} \theta_m = 1, \theta_m \geq 0\right\} \tag{6.14}$$

La principal propiedad de este tipo de vectores es que sus marginales posee **distribución beta** $T_m \sim \beta(\alpha_m, \sum_{\eta \neq m} \alpha_\eta)$, una variable aleatoria utilizada para modelar probabilidades cuya esperanza es $\mathbb{E}[T_m] = \frac{\alpha_m}{\sum_{\eta=1}^{V} \alpha_\eta}$. Con este modelo, la *distribución a posteriori* puede calcularse como:

$$\begin{aligned}
&p\left(\theta_1^{(k)}, \cdots, \theta_V^{(k)} | \tilde{N}_1^{(k)}, \cdots, \tilde{N}_V^{(k)}\right) \\
&\quad \propto P\left(\tilde{N}_1^{(k)}, \cdots, \tilde{N}_V^{(k)} | \theta_1^{(k)}, \cdots, \theta_V^{(k)}\right) \cdot p\left(\theta_1^{(k)}, \cdots, \theta_V^{(k)}\right) &(6.15)\\
&\quad \propto \left(\prod_{m=1}^{V} (\theta_m^{(k)})^{\tilde{N}_m^{(k)}}\right) \left(\prod_{m=1}^{V} (\theta_m^{(k)})^{\alpha_m - 1} \cdot \mathbf{1}\left\{\theta_m^{(k)} \geq 0\right\}\right) \cdot \mathbf{1}\left\{\sum_{m=1}^{V} \theta_m^{(k)} = 1\right\} &(6.16)
\end{aligned}$$

con lo cuál $\mathbf{T}_k|_{\tilde{N}_1^{(k)},\cdots,\tilde{N}_V^{(k)}} \sim \text{Dir}([\tilde{N}_1^{(k)} + \alpha_1, \cdots, \tilde{N}_V^{(k)} + \alpha_V])$. De esta manera, utilizando como estimador puntual bayesiano la **media a posteriori** se obtiene que

$$\theta_m^{(k)} = \mathbb{E}[T_m | \tilde{N}_1^{(k)}, \cdots, \tilde{N}_V^{(k)}] = \frac{\tilde{N}_m^{(k)} + \alpha_m}{\sum_{\eta=1}^{V} \tilde{N}_\eta^{(k)} + \alpha_\eta}. \tag{6.17}$$

Mientras que GNB es un modelo gráfico con estimadores puntuales clásicos, MNB utiliza estimadores puntuales bayesianos. Sin embargo, todavía nos estamos conformando con hacer una estimación puntual en lugar de un cálculo predictivo. A continuación veremos cuanto es necesario complejizar al modelo para efectivamente poder hacer dicho cálculo.

6.3. Bayes Variacional Gaussiano

La estadística bayesiana basa su supuesto en considerar a los parámetros parte del espacio latente, los que se sumarán a las variables no observables propias del modelo (como la variable mezcladora en un problema de *clustering*). Sea $\mathbf{Z}$ el vector de variables no observable, en general será prohibitivo calcular la distribución *a posteriori* $p(\mathbf{z}|\mathbf{x})$ en modelos complejos. Una alternativa es aproximar dicha distribución minimizando la **divergencia de Kullback Leibler**:

$$\arg\min_{q\in\mathcal{P}} \ \mathrm{KL}\left(q(\cdot|\mathbf{x})\|p(\cdot|\mathbf{x})\right) \tag{6.18}$$

donde $q(\mathbf{z}|\mathbf{x})$[1] cumple ciertas restricciones $\mathcal{P}$ que permitan limitar el modelo posible de forma que sea factible emplear un estudio analítico. Dicha divergencia puede descomponerse como $\mathrm{KL}\left(q(\cdot|\mathbf{x})\|p(\cdot|\mathbf{x})\right) = \log p(\mathbf{x}) - \mathrm{ELBO}(q(\cdot|\mathbf{x}))$, donde

$$\mathrm{ELBO}(q(\cdot|\mathbf{x})) = H(q(\cdot|\mathbf{x})) + \mathbb{E}_q\left[\log p(\mathbf{x},\mathbf{Z})|\mathbf{X}=\mathbf{x}\right] \tag{6.19}$$

Similar al algoritmo EM, la **cota inferior esperada** (ELBO) acota la verosimilitud de forma maximizar una cota fomente la maximización de la misma $\mathrm{ELBO}(q(\cdot|\mathbf{x})) \leq \log p(\mathbf{x})$ (ya que la divergencia de Kullback Leibler es no negativa). Esta log-verosimilitud debe ser entendida por muestra, por lo que la para la log-verosimilitud de un conjunto simplemente se sumarían los ELBOs. A su vez, este mismo fenómeno se da acotando a:

$$\mathrm{ELBO}(q) = \mathbb{E}_q\left[\log p(\mathbf{x}|\mathbf{z})|\mathbf{x}\right] - \mathrm{KL}\left(q(\cdot|\mathbf{x})\|p(\cdot)\right) \leq \mathbb{E}_q\left[\log p(\mathbf{x}|\mathbf{z})|\mathbf{x}\right] \tag{6.20}$$

donde $\mathbb{E}_q\left[\log p(\mathbf{x}|\mathbf{z})|\mathbf{x}\right]$ es la *cross-entropy* asociada a un **autoencoder** de *encoder* $q(\cdot|\mathbf{x})$ y *decoder* $p(\cdot|\mathbf{z})$. Es decir que la maximización de la ELBO también tenderá a aumentar dicha magnitud.

Minimizar la divergencia de Kullback Leibler (6.18) equivale a maximizar la ELBO. Para poder tratar el problema, se suele asumir como restricción sobre $\mathcal{P}$ la llamada **Mean field approximation**: Suponer que q se puede factorizar como productos de densidades tratables, separando entre las variables ocultas $\mathbf{u}$ y los parámetros ϕ. Es decir, sea $\mathbf{z} = (\mathbf{u},\phi)$ se relaja el problema suponiendo $q(\mathbf{z}|\mathbf{x}) = q_1(\mathbf{u}|\mathbf{x})q_2(\phi|\mathbf{x})$ para todo $q \in \mathcal{P}$. Con esta hipótesis, la entropía de la distribución producto se descompone como suma $H(q(\cdot|\mathbf{x})) = H(q_1(\cdot|\mathbf{x})) + H(q_2(\cdot|\mathbf{x}))$ y la esperanza del logaritmo de la conjunta se puede descomponer en dos sentidos:

$$\mathbb{E}_q\left[\log p(\mathbf{x},\mathbf{Z})|\mathbf{X}=\mathbf{x}\right] = \int_{\mathcal{U}} q_1(\mathbf{u}|\mathbf{x})\left(\int_{\Phi} q_2(\phi|\mathbf{x})\log p(\mathbf{x},\mathbf{u},\phi)d\phi\right)d\mathbf{u} \tag{6.21}$$

$$= \int_{\Phi} q_2(\phi|\mathbf{x})\left(\int_{\mathcal{U}} q_1(\mathbf{u}|\mathbf{x})\log p(\mathbf{x},\mathbf{u},\phi)d\mathbf{u}\right)d\phi \tag{6.22}$$

El algoritmo **bayes variacional** propone resolver el problema (6.18) de forma iterativa:

[1] Notar que estamos haciendo un análisis para cada muestra $\mathbf{x}$.

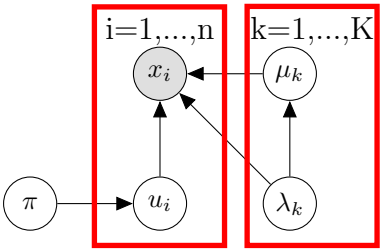


Figura 6.6: Red bayesiana del modelo bayes variacional gaussiano.

suponer q_1 fijo para optimizar en q_2 para luego dejar fijo q_2 para optimizar en q_1.

$$q_1(\cdot|\mathbf{x}) = \arg\max_{q_1} \int_{\mathcal{U}} q_1(\mathbf{u}|\mathbf{x}) \log \frac{e^{E_1(\mathbf{x},\mathbf{u})}}{q_1(\mathbf{u}|\mathbf{x})} d\mathbf{u} \tag{6.23}$$

$$q_2(\cdot|\mathbf{x}) = \arg\max_{q_2} \int_{\Phi} q_2(\mathbf{u}|\mathbf{x}) \log \frac{e^{E_2(\mathbf{x},\phi)}}{q_2(\phi|\mathbf{x})} d\phi \tag{6.24}$$

donde

$$E_1(\mathbf{x},\mathbf{u}) = \int_{\Phi} q_2(\phi|\mathbf{x}) \log p(\mathbf{x},\mathbf{u},\phi) d\phi \equiv f(q_2) \tag{6.25}$$

$$E_2(\mathbf{x},\phi) = \int_{\mathcal{U}} q_1(\mathbf{u}|\mathbf{x}) \log p(\mathbf{x},\mathbf{u},\phi) d\mathbf{u} \equiv f(q_1) \tag{6.26}$$

El problema de optimización (6.23) equivale a minimizar la divergencia de Kullback Leibler entre $q_1(\mathbf{u}|\mathbf{x})$ y $k_1 e^{E_1(\mathbf{x},\mathbf{u})}$, donde k_1 es una constante de proporcionalidad para que la expresión sea una densidad/probabilidad en $\mathbf{u}$. De manera similar ocurre con $q_2(\phi|\mathbf{x})$ y $k_2 e^{E_2(\mathbf{x},\phi)}$ para el problema (6.24). Entonces, el algoritmo bayes varaicional consiste en iterar entre $q_1(\mathbf{u}|\mathbf{x}) \propto e^{E_1(\mathbf{x},\mathbf{u})}$ y $q_2(\phi|\mathbf{x}) \propto e^{E_2(\mathbf{x},\phi)}$.

6.3.1. Mezcla de Gaussianas escalares en Bayes Variacional

Para entender las complicaciones para efectuar los cálculos, incluso en un problema simple, supongamos el modelo de mezcla de gaussianas escalares graficado en la Fig. 6.6. Este problema se lo conoce como **Bayes Variacional Gaussiano** (GVB) [16, Sección 10.2]. Interpretando su red bayesiana, la distribución del modelo puede escribirse como

$$p(\mathbf{x},\mathbf{u},\pi,\lambda,\mu) = p(\pi)\left(\prod_{k=1}^{K} p(\lambda_k)p(\mu_k|\lambda_k)\right)\left(\prod_{i=1}^{n} P(u_i|\pi)p(x_i|u_i,\mu,\lambda)\right) \tag{6.27}$$

donde $u|\pi \sim \text{Cat}(\pi)$ y $x|u,\mu,\lambda \sim \mathcal{N}(\mu_u, \lambda_u^{-1})$. *A priori* supondremos distribuciones conjugadas a priori con la verosimilitud: $\pi \sim \text{Dir}(\alpha)$, $\lambda_k \sim \Gamma(\nu,\beta)$ y $\mu_k|\lambda_k \sim \mathcal{N}(m,(\delta\lambda_k)^{-1})$. Los parámetros de este modelo serán entonces α,ν,β,m y δ. En este caso $\mathbf{u} = (u_1,\cdots,u_n)$ y $\phi = (\pi,\mu,\lambda)$ con $\pi = (\pi_1,\cdots,\pi_K)$, $\mu = (\mu_1,\cdots,\mu_K)$ y $\lambda = (\lambda_1,\cdots,\lambda_K)$. Un caso particular cuando $K = 1$ y $\delta \to \infty$ fue estudiando en el Ej. 6.1, donde la distribución a posteriori de la precisión (inversa de la varianza) es una Gamma y la distribución predic-

tiva es una t-student.

Utilizar como distribución *a priori* una media gaussiana y una precisión Gamma, es una familia muy estudiada en la bibliografía y recibe el nombre de **distribución normal-gamma**.

Definición 6.2 *Un vector aleatorio normal-gamma* (ν, β, m, δ) *tiene una densidad conjunta de la forma:*

$$p(\lambda, \mu) = \frac{\beta^\nu}{\Gamma(\nu)} \sqrt{\frac{\delta}{2\pi}} \lambda^{\nu - \frac{1}{2}} e^{-\lambda\left(\beta + \frac{\delta\mu^2}{2} - \delta m\mu + \frac{\delta m^2}{2}\right)} \mathbf{1}\{\lambda > 0\} \tag{6.28}$$

donde $\lambda \sim \Gamma(\nu, \beta)$ *y* $\mu|\lambda \sim \mathcal{N}(m, (\delta\lambda)^{-1})$.

Teniendo en cuenta todas estas definiciones, a continaución se procederá a calcular la distribución *a posteriori* y la predictiva del modelo.

6.3.1.1. Distribución a posteriori en GVB

Bajo la hipótesis *mean field approximation*, $q(\mathbf{u}, \pi, \lambda, \mu|\mathbf{x}) = Q_1(\mathbf{u}|\mathbf{x}) q_2(\pi, \lambda, \mu|\mathbf{x})$, por lo que

$$E_1(\mathbf{x}, \mathbf{u}) = \text{cte} + \sum_{i=1}^{n} \int q_2(\pi|\mathbf{x}) \log P(u_i|\pi) d\pi + \sum_{i=1}^{n} \int\int q_2(\mu, \lambda|\mathbf{x}) \log p(x_i|u_i, \mu, \lambda) d\mu d\lambda \tag{6.29}$$

donde "cte" hace referencia términos que no dependen de q_2. De forma análoga

$$\begin{aligned} E_2(\mathbf{x}, \pi, \lambda, \mu) = &\log p(\pi) + \sum_{k=1}^{K} \log p(\lambda_k) + \sum_{k=1}^{K} \log p(\mu_k|\lambda_k) \\ &+ \sum_{i=1}^{n} \sum_{k=1}^{K} Q_1(u_i = k|\mathbf{x}) \log P(u_i = k|\pi) \\ &+ \sum_{i=1}^{n} \sum_{k=1}^{K} Q_1(u_i = k|\mathbf{x}) \log p(x_i|u_i = k, \mu, \lambda) \end{aligned} \tag{6.30}$$

En primer lugar, se considerará Q_1 fijo y conocido para computar q_2. Esto se conoce como **actualización de los parámetros a posteriori**. Es decir, asumiendo que las familias elegidas son conjuntadas a priori, los parámetros *a priori* son α, ν, β, m y δ tendrán sus contra-partes *a posteriori* $\alpha^*, \nu^*, \beta^*, m^*$ y δ^* (los cuales puede ser diferentes para cada clase). Lo primero a notar es la factorización. Sea $\gamma_{i,k} = Q_1(u_i = k|\mathbf{x})$, luego $q_2(\pi, \lambda, \mu|\mathbf{x}) \propto e^{E_2(\mathbf{x}, \pi, \lambda, \mu)}$:

$$q_2(\pi, \lambda, \mu|\mathbf{x}) \propto p(\pi) \left(\prod_{k=1}^{K} p(\lambda_k) p(\mu_k|\lambda_k) \right) \prod_{k=1}^{K} e^{\sum_{i=1}^{n} \gamma_{i,k} [\log \pi_k + \log \mathcal{N}(x_i|\mu_k, \lambda_k^{-1})]} \tag{6.31}$$

y por lo tanto $q_2(\pi, \lambda, \mu|\mathbf{x}) = q_2(\pi|\mathbf{x}) \prod_{k=1}^{K} q_2(\mu_k, \lambda_k|\mathbf{x})$ (es decir, se acaba de demostrar la relación de independiencia, resta calcular $q_2(\pi|\mathbf{x})$ y $q_2(\mu_k, \lambda_k|\mathbf{x})$ para cada k). Dado que

el logaritmo de la densidad normal es una magnitud cuadrática en las muestras, podemos definir los siguientes **estadísticos suficientes**:

$$N_k = \sum_{i=1}^{n} \gamma_{i,k}, \qquad f_k = \sum_{i=1}^{n} \gamma_{i,k} x_i, \qquad s_k = \sum_{i=1}^{n} \gamma_{i,k} x_i^2 \tag{6.32}$$

y por lo tanto, el último exponente de (6.31) puede simplificarse como:

$$N_k \log \pi_k - \frac{N_k}{2} \log(2\pi) + \frac{N_k}{2} \log(\lambda_k) - \frac{\lambda_k}{2}(s_k - 2 f_k \mu_k + N_k \mu_k^2) \tag{6.33}$$

Para computar $q_2(\pi|\mathbf{x})$, basta por juntar los términos de (6.31) donde aparece π

$$q_2(\pi|\mathbf{x}) \propto \left(\prod_{k=1}^{K} \pi_k^{\alpha_k - 1} e^{N_k \log \pi_k} \right) \mathbf{1}\left\{ \sum_{k=1}^{K} \pi_k = 1, \pi_k \geq 0 \right\} \tag{6.34}$$

con lo cual $\pi|\mathbf{x} \sim \text{Dir}([\alpha_1 + N_1, \cdots, \alpha_K + N_K])$, es decir $\alpha_k^* = \alpha_k + N_k$. Para el cálculo de $q_2(\mu_k, \lambda_k|\mathbf{x})$, resta analizar:

$$q_2(\mu_k, \lambda_k|\mathbf{x}) \propto \underbrace{\lambda_k^{\nu-1} e^{-\beta\lambda_k} \mathbf{1}\{\lambda_k > 0\}}_{\propto p(\lambda_k)} \underbrace{\lambda_k^{1/2} e^{\frac{-\delta\lambda_k(\mu_k - m)^2}{2}}}_{\propto p(\mu_k|\lambda_k)} \lambda_k^{\frac{N_k}{2}} e^{\frac{-\lambda_k(s_k - 2 f_k \mu_k + N_k \mu_k^2)}{2}} \tag{6.35}$$

$$\propto \lambda_k^{\nu + \frac{N_k}{2} - \frac{1}{2}} e^{-\lambda_k \left(\beta + \frac{\delta\mu_k^2}{2} - \delta m \mu_k + \frac{\delta m^2}{2} + \frac{s_k}{2} - f_k \mu_k + \frac{N_k \mu_k^2}{2} \right)} \mathbf{1}\{\lambda_k > 0\} \tag{6.36}$$

Se puede apreciar que esta distribución es una normal-gamma. Para calcular los parámetros *a posteriori* basta con comparar la expresión con (6.28) (con los parámetros a posteriori $\nu_k^*, \beta_k^*, m_k^*, \delta_k^*$) y despejar:

- $\nu_k^* - \frac{1}{2} = \nu + \frac{N_k}{2} - \frac{1}{2}$.
- $\frac{\delta_k^*}{2} = \frac{\delta + N_k}{2}$.
- $\delta_k^* m_k^* = \delta m + f_k$.
- $\beta_k^* + \frac{\delta_k^* m_k^{*2}}{2} = \beta + \frac{\delta m^2}{2} + \frac{s_k}{2}$.

Es decir, $\nu_k^* = \nu + \frac{N_k}{2}$, $\delta_k^* = \delta + N_k$ y $m_k^* = \frac{\delta m + f_k}{\delta + N_k}$. Para el caso de β_k^*, se puede calcular como:

$$\beta_k^* = \beta + \frac{\delta m^2}{2} + \frac{s_k}{2} - \frac{(\delta m + f_k)^2}{2(\delta + N_k)} \tag{6.37}$$

con lo cual $\mu_k|\lambda_k, \mathbf{x} \sim \mathcal{N}\left(\frac{\delta m + f_k}{\delta + N_k}, \frac{1}{\lambda_k(\delta + N_k)} \right)$ y $\lambda_k|\mathbf{x} \sim \Gamma\left(\nu + \frac{1}{2} N_k, \beta + \frac{\delta m^2}{2} + \frac{1}{2} s_k - \frac{(\delta m + f_k)^2}{2(\delta + N_k)} \right)$.

En segundo lugar, se considerará q_2 fijo y conocido para calcular Q_1. Esto se conoce como **actualización de la distribución de las variables ocultas**. La distribución Q_1 será la predicción del problema de *clustering* asociado. Para calcular dicha distribución, se utilizarán las siguientes propiedades de las distribuciones Gamma y Beta.

Propiedades 6.1 *Sean $\lambda \sim \Gamma(\nu, \beta)$ y $\pi \sim \beta(a, b)$, la esperanza del logaritmo de estas variables se calcula como $\mathbb{E}[\log \lambda] = \psi(\nu) - \log(\beta)$ y $\mathbb{E}[\log \pi] = \psi(a) - \psi(a + b)$,*

donde $\psi(\cdot)$ es la función digamma $\psi(z) = \frac{\Gamma'(z)}{\Gamma(z)}$.

Lo primero a notar para el cómputo de $Q_1(\mathbf{u}|\mathbf{x}) \propto e^{E_1(\mathbf{x},\mathbf{u})}$ es la independencia de sus variables:

$$Q_1(\mathbf{u}|\mathbf{x}) \propto \prod_{i=1}^{n} e^{\int q_2(\pi|\mathbf{x}) \log P(u_i|\pi) d\pi + \int\int q_2(\mu,\lambda|\mathbf{x}) \log p(x_i|u_i,\mu,\lambda) d\mu d\lambda} = \prod_{i=1}^{n} Q_1(u_i|\mathbf{x}) \tag{6.38}$$

entonces basta con calcular cada $Q_1(u_i = k|\mathbf{x})$. Sean los parámetros de q_2 definidos como $\pi|\mathbf{x} \sim \text{Dir}(\alpha^*)$, $\mu_k|\lambda_k, \mathbf{x} \sim \mathcal{N}\left(m_k^*, (\delta_k^*\lambda_k)^{-1}\right)$ y $\lambda_k|\mathbf{x} \sim \Gamma\left(\nu_k^*, \beta_k^*\right)$. Luego

$$Q_1(u_i = k|\mathbf{x}) \propto e^{\psi(\alpha_k^*) - \psi\left(\sum_{c=1}^{K} \alpha_c^*\right) + \frac{\psi(\nu_k^*) - \log(\beta_k^*)}{2} - \frac{1}{2}\mathbb{E}_{q_2}[\lambda_k (x_i - \mu_k)^2 | \mathbf{x}]} \tag{6.39}$$

donde se usó que $\pi_k|\mathbf{x} \sim \beta(\alpha_k^*, \sum_{\eta \neq k} \alpha_\eta^*)$. La esperanza de la última expresión puede calcularse con

$$\mathbb{E}_{q_2}[\lambda_k (x_i - \mu_k)^2 | \mathbf{x}] = \mathbb{E}_{q_2}[\lambda_k \mathbb{E}_{q_2}[(x_i - \mu_k)^2 | \lambda_k, \mathbf{x}] | \mathbf{x}] \tag{6.40}$$

$$= \mathbb{E}_{q_2}[\lambda_k \mathbb{E}_{q_2}[(x_i - m_k^* + m_k^* - \mu_k)^2 | \lambda_k, \mathbf{x}] | \mathbf{x}] \tag{6.41}$$

$$= \mathbb{E}_{q_2}[\lambda_k \left((x_i - m_k^*)^2 + \mathbb{E}_{q_2}[(\mu_k - m_k^*)^2 | \lambda_k, \mathbf{x}] + 2(x_i - m_k^*)\mathbb{E}_{q_2}[m_k^* - \mu_k | \mathbf{x}]\right) | \mathbf{x}] \tag{6.42}$$

$$= \frac{\nu_k^*}{\beta_k^*}(x_i - m_k^*)^2 + \frac{1}{\delta_k^*} + 0 \tag{6.43}$$

Finalmente

$$Q_1(u_i = k|\mathbf{x}) \propto e^{\psi(\alpha_k^*) - \psi\left(\sum_{c=1}^{K} \alpha_c^*\right) + \frac{\psi(\nu_k^*) - \log(\beta_k^*)}{2} - \frac{1}{2\delta_k^*} - \frac{\nu_k^*}{2\beta_k^*}(m_k^* - x_i)^2} \tag{6.44}$$

En resumen, la distribución *a posteriori* en un problema de GVB se calcula inicializando $\gamma_{i,k}$ (por ejemplo con el algoritmo EM) e iterando entre:

- Calcular $(\alpha_k^*, m_k^*, \delta_k^*, \nu_k^*, \beta_k^*)$ a partir de $\gamma_{i,k}$ como

$$\alpha_k^* = \alpha_k + \sum_{i=1}^{n} \gamma_{i,k}, \quad m_k^* = \frac{\delta m + \sum_{i=1}^{n} \gamma_{i,k} x_i}{\delta + \sum_{i=1}^{n} \gamma_{i,k}} \tag{6.45}$$

$$\delta_k^* = \delta + \sum_{i=1}^{n} \gamma_{i,k}, \quad \nu_k^* = \nu + \frac{1}{2}\sum_{i=1}^{n} \gamma_{i,k} \tag{6.46}$$

$$\beta_k^* = \beta + \frac{\delta m^2}{2} + \frac{1}{2}\sum_{i=1}^{n} \gamma_{i,k} x_i^2 - \frac{(\delta m + \sum_{i=1}^{n} \gamma_{i,k} x_i)^2}{2(\delta + \sum_{i=1}^{n} \gamma_{i,k})} \tag{6.47}$$

- Calcular $\gamma_{i,k} = \frac{\rho_{i,k}}{\sum_{c=1}^{K} \rho_{i,c}}$ a partir de $(\alpha^*, m_k^*, \delta_k^*, \nu_k^*, \beta_k^*)$ como

$$\rho_{i,k} = e^{\psi(\alpha_k^*) - \psi\left(\sum_{c=1}^{K} \alpha_c^*\right) + \frac{\psi(\nu_k^*) - \log(\beta_k^*)}{2} - \frac{1}{2\delta_k^*} - \frac{\nu_k^*}{2\beta_k^*}(m_k^* - x_i)^2} \tag{6.48}$$

6.3.1.2. Distribución Predictiva en GVB

Por un lado, la distribución *a posteriori* q_2 estimada durante el entrenamiento cumple la relación de independencia $q_2(\pi, \lambda, \mu|\mathbf{x}) = q_2(\pi|\mathbf{x}) \prod_{k=1}^{K} q_2(\mu_k, \lambda_k|\mathbf{x})$. Por el otro, una nueva muestra x_{test} (que no pertenezca al conjunto de entrenaimento $\mathbf{x}$) pertenecerá a un modelo de mezcla $p(x_{\text{test}}|\pi, \mu, \lambda) = \sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(x_{\text{test}}|\mu_k, \lambda_k^{-1})$. Juntando ambas consideraciones

podemos ver que la distribución predictiva tambbién es una mezcla:

$$p(x_{\text{test}}|\mathbf{x}) = \mathbb{E}[p(x_{\text{test}}|\phi)|\mathbf{x}] \tag{6.49}$$

$$= \sum_{k=1}^{K} \mathbb{E}[\pi_k|\mathbf{x}] \cdot \mathbb{E}\left[\sqrt{\frac{\lambda_k}{2\pi}} e^{\frac{-\lambda_k}{2}(x_{\text{test}}-\mu_k)^2} \middle| \mathbf{x} \right] \tag{6.50}$$

$$= \sum_{k=1}^{K} \frac{\alpha_k^*}{\sum_{c=1}^{K} \alpha_c^*} \cdot \tilde{p}_k(x_{\text{test}}|\mathbf{x}) \tag{6.51}$$

donde se utilizó que $\pi_k|\mathbf{x} \sim \beta(\alpha_k^*, \sum_{\eta \neq k} \alpha_\eta^*)$. La distribución predictiva es una mezcla cuyos pesos son la proporción de los α^*; resta calcular las nuevas densidades $\tilde{p}_k(x_{\text{test}}|\mathbf{x})$. Para ello, notar que $(\mu_k, \lambda_k)|\mathbf{x}$ tienen una distribución normal-gamma (6.28) de parámetros $(\nu_k^*, \beta_k^*, m_k^*, \delta_k^*)$:

$$\tilde{p}_k(x_{\text{test}}|\mathbf{x}) \propto \int_0^\infty \int_{-\infty}^\infty \sqrt{\lambda_k} e^{\frac{-\lambda_k}{2}(x_{\text{test}}-\mu_k)^2} \lambda_k^{\nu_k^*-\frac{1}{2}} e^{-\lambda_k\left(\beta_k^* + \frac{\delta_k^*\mu_k^2}{2} - \delta_k^* m_k^* \mu_k + \frac{\delta_k^* {m_k^*}^2}{2}\right)} d\mu_k d\lambda_k \tag{6.52}$$

$$\propto \int_0^\infty \int_{-\infty}^\infty \lambda^{\nu_k^*} e^{-\lambda_k\left(\beta_k^* + \frac{\delta_k^*\mu_k^2}{2} - \delta_k^* m_k^* \mu_k + \frac{\delta_k^* {m_k^*}^2}{2} + \frac{\mu_k^2}{2} - x_{\text{test}}\mu_k + \frac{x_{\text{test}}^2}{2}\right)} d\mu_k d\lambda_k \tag{6.53}$$

Nuevamente el interior de la integral es una distribución normal-gamma. Como toda densidad integra 1, basta con reconocer los parámetros $(\tilde{\nu}, \tilde{\beta}, \tilde{m}, \tilde{\delta})$ de la nueva distribución para resolver la integral. Igualando con (6.28), se obtiene:

- $\tilde{\nu} - \frac{1}{2} = \nu_k^*$
- $\frac{\tilde{\delta}}{2} = \frac{\delta_k^*}{2} + \frac{1}{2}$
- $\tilde{\delta}\tilde{m} = \delta_k^* m_k^* + x_{\text{test}}$
- $\tilde{\beta} + \frac{\tilde{\delta}\tilde{m}^2}{2} = \beta_k^* + \frac{\delta_k^* {m_k^*}^2}{2} + \frac{x_{\text{test}}^2}{2}$

De las primeras tres ecuaciones es inmediato notar que $\tilde{\nu} = \nu_k^* + \frac{1}{2}$, $\tilde{\delta} = \delta_k^* + 1$ y $\tilde{m} = \frac{x_{\text{test}} + \delta_k^* m_k^*}{\delta_k^* + 1}$. Para la ecuación de $\tilde{\beta}$ basta con notar que

$$\tilde{\beta} = \beta_k^* + \frac{\delta_k^* {m_k^*}^2}{2} + \frac{x_{\text{test}}^2}{2} - \frac{(x_{\text{test}} + \delta_k^* m_k^*)^2}{2(\delta_k^* + 1)} \tag{6.54}$$

$$= \beta_k^* + \frac{(\delta_k^* {m_k^*}^2 + x_{\text{test}}^2)(\delta_k^* + 1) - (x_{\text{test}} + \delta_k^* m_k^*)^2}{2(\delta_k^* + 1)} \tag{6.55}$$

$$= \beta_k^* + \frac{\delta_k^{*2} {m_k^*}^2 + \delta_k^* {m_k^*}^2 + \delta_k^* x_{\text{test}}^2 + x_{\text{test}}^2 - x_{\text{test}}^2 - 2\delta_k^* m_k^* x_{\text{test}} - \delta_k^{*2} {m_k^*}^2}{2(\delta_k^* + 1)} \tag{6.56}$$

$$= \beta_k^* + \frac{\delta_k^* (x_{\text{test}} - m_k^*)^2}{2(\delta_k^* + 1)} \tag{6.57}$$

Como en la única variable que aparece x_{test} es en $\tilde{\beta}$, podemos decir que la integral es

$\tilde{p}_k(x_{\text{test}}|\mathbf{x}) \propto \tilde{\beta}^{-\tilde{\nu}}$, es decir:

$$\tilde{p}_k(x_{\text{test}}|\mathbf{x}) \propto \left(\beta_k^* + \frac{\delta_k^*(x_{\text{test}} - m_k^*)^2}{2(\delta_k^*+1)}\right)^{-(\nu_k^*+1/2)} \tag{6.58}$$

$$\propto \left(1 + \frac{\delta_k^*\nu_k^*}{(\delta_k^*+1)\beta_k^*}\frac{(x_{\text{test}} - m_k^*)^2}{2\nu_k^*}\right)^{-\frac{2\nu_k^*+1}{2}} \tag{6.59}$$

Este tipo distribución se conoce como t-student generalizada.

Definición 6.3 *Una variable aleatoria tiene distribución t-Student Generalizada:* $X \sim t(\mu, \Lambda, \nu)$ *si su densidad es de la forma*

$$p(x) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)}\sqrt{\frac{\Lambda}{\pi\nu}}\left(1 + \Lambda\frac{(x-\mu)^2}{\nu}\right)^{-\frac{\nu+1}{2}} \tag{6.60}$$

Por lo tanto, la densidad predictiva es una mezcla de t-students de la forma:

$$p(x_{\text{test}}|\mathbf{x}) = \sum_{k=1}^{K} \frac{\alpha_k^*}{\sum_{c=1}^{K}\alpha_c^*} \cdot t\left(x_{\text{test}}\middle| m_k^*, \frac{\delta_k^*\nu_k^*}{(\delta_k^*+1)\beta_k^*}, 2\nu_k^*\right) \tag{6.61}$$

Finalmente hemos analizado la distribución predictiva de un modelo relativamente simple. Incluso en modelos escalares, que propongan **distribuciones conjugadas a priori**, el estudio analítico es complejo. Esto demuestra la necesidad de contar con métodos numéricos que permitan abordar el tema.

6.4. Monte Carlo por Cadenas de Markov (MCMC)

Efectuar el cálculo con distribuciones predictivas, evaluando la integral directamente, puede ser computacionalmente inviable ya que rara vez se cuenta con una forma cerrada o conocida para la distribución a posteriori. Para resolver este problema, se plantea una solución Monte Carlo: combinar métodos de muestreo con la **ley de los grandes números**.

$$\mathbb{E}\left[p(x_{\text{test}}|T)|\mathbf{X} = \mathbf{x}\right] \approx \frac{1}{t_{\max}}\sum_{t=1}^{t_{\max}} p_{X|T=\theta_t}(x_{\text{test}}) \tag{6.62}$$

donde $\theta_1, \cdots, \theta_{t_{\max}}$ son muestras generadas a partir de la distribución *a posteriori*.

En capítulos anteriores se discutió por que la inteligencia artificial no es más que una mezcla de patrones encontrados en datos observados con ruidos aleatorios. Sin embargo, la ley de los grandes números nos muestra el potencial de los datos: cualquier comportamiento esperado puede aproximarse tan bien como uno desee si se cuenta la cantidad de datos suficiente. Este resultado da que pensar acerca de como y cuando entregamos nuestros datos. Quizás estemos alimentando nuestro reemplazo.

El problema con este método es que difícilmente podamos generar muchas muestras independientes. Es entonces cuando surge el Muestreo Monte Carlo por Cadenas de Mar-

kov (MCMC) como una alternativa [16, Capítulo 11]. Una **cadena de Markov** es una secuencia de variables aleatorias $\{\theta_t\}_{t\in\mathbb{N}}$ que cumple la propiedad de *falta de memoria*: la distribución del próximo estado depende únicamente del estado actual, y no del pasado completo. Formalmente,

$$p(\theta_{t+1}|\theta_t, \theta_{t-1}\cdots, \theta_0) = p(\theta_{t+1}|\theta_t) = P(\theta_t \to \theta_{t+1}) \tag{6.63}$$

donde $P(\theta_t \to \theta_{t+1})$ es la densidad de transición de θ_t a θ_{t+1} (en la estadística bayesiana todas estas distribuciones serán computadas implícitamente *a posteriori* de observar los datos de entrenamiento). Cuando las probabilidades de transición no dependen del tiempo t, se dice que la cadena es **homogénea**. Una distribución π se dice *estacionaria* para una cadena de Markov si no varía su estadística al propagarse por la cadena;

$$\pi(\theta') = \int \pi(\theta)P(\theta \to \theta')\, d\theta \tag{6.64}$$

es decir, que la distribución de θ y θ' sea la misma (que la distribución de θ no varíe al efectuar un paso en la cadena). Una condición suficiente para asegurar esto es que la conjunta de ambas sea simétrica:

$$\pi(\theta)P(\theta \to \theta') = \pi(\theta')P(\theta' \to \theta) \tag{6.65}$$

la cual se cumple def orma trivial si $\theta = \theta'$ (repetir el estado actual no afecta la condición (6.65) correspondiente al estado estacionario).

La idea general del MCMC es construir una cadena de Markov homogénea $\{\theta_t\}_{t\in\mathbb{N}}$ cuya distribución estacionaria sea la *distribución a posteriori*. Bajo ciertas condiciones, el **teorema de ergodicidad** garantiza que el promedio de los valores generados por la cadena converge a la esperanza, dando por válida (6.62) aunque no se trate de muestras independientes. Para una cadena de Markov homogenea, dichas condiciones se pueden resumir en:

- Irreducible: La transición $\theta \to \theta'$ debe poder ser alcanzada en una cantidad finita de pasos para todo θ y θ'.
- Áperiódica: La transición $\theta \to \theta$ (mantener el estado actual) debe tener probabilidad positiva.
- Recurrente positiva: El tiempo esperado para volver al estado actual es finito.

Los algoritmos utilizados para los modelos bayesianos fueron construidos para cumplir con estas condiciones. A continuación se presentará un ejemplo de cálculo para calcular el estado estacionario de una cadena de Markov homogénea.

Ejemplo 6.2 *Encontrar el estado estacionario de una cadena de Markov homogénea, donde θ es una variable discreta que toma valores en $\{0, 1, 2\}$ y las probabilidades de transición se definen con la matriz* $P = \begin{pmatrix} 0.7 & 0.2 & 0.1 \\ 0.4 & 0.6 & 0.0 \\ 0.0 & 0.9 & 0.1 \end{pmatrix}$.

No es difícil probar que esta cadena es irreducible, aperiódica y recurrente positiva. Toda cadena de Markov irreducible en un espacio de estados finitos tiene una distribución estacionaria única. Se desea encontrar $\pi(0)$, $\pi(1)$ y $\pi(2)$ (representadas por un vector π), tal que se cumpla (6.64). Para variables discretas y finitas eso se puede representar como $\pi = P \cdot \pi$ con $\mathbf{1}^T \cdot \pi = 1$, donde $\mathbf{1}$ es un vector con todas sus entradas en 1. Escribiendo todo eso como un sistema de ecuaciones se obtiene

$$\begin{pmatrix} 0.7 & 0.2 & 0.1 \\ 0.4 & 0.6 & 0 \\ 0 & 0.9 & 0.1 \\ 1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} \pi(0) \\ \pi(1) \\ \pi(2) \end{pmatrix} = \begin{pmatrix} \pi(0) \\ \pi(1) \\ \pi(2) \\ 1 \end{pmatrix} \tag{6.66}$$

La segunda ecuación $0.4\pi(0) + 0.6\pi(1) = \pi(1)$ implica que $\pi(0) = \pi(1)$. De la tercera $0.9\pi(1) + 0.1\pi(2) = \pi(2)$ se obtiene que $\pi(1) = \pi(2)$. Reemplazando podemos ver que se cumple la primera ecuación $0.7\pi(0) + 0.2\pi(1) + 0.1\pi(2) = \pi(0)$ y, dado que deben sumar 1 (cuarta ecuación), se obtiene $\pi(0) = \pi(1) = \pi(2) = \frac{1}{3}$.

6.4.1. Algoritmos de Muestreo MCMC

El objetivo de los algoritmos de muestreo es generar un proceso cuya secuencia de muestras sea ergódica. Esto no siempre es sencillo, ya que puede no disponerse de toda la información necesaria sobre las distribuciones.

En la Fig. 6.7 puede verse un ejemplo de experimento de muestreo. Se denomina *tune* a la cantidad de muestras a descartar para considerar que se alcanzó el estado estacionario, y *draws* a la cantidad de muestras efectivas que fueron generadas. Se suelen generar varias cadenas y verificar los resultados en cada una (cantidad definida en *chains*). Una disparidad de resultados en las cadenas evidencia que no se alcanzó el mencionado estado estacionario. Dependiendo del tipo de variable aleatoria a muestrear, conviene usar una u otra estrategia. A continuación se presentarán algunos de los muestreos más habituales.

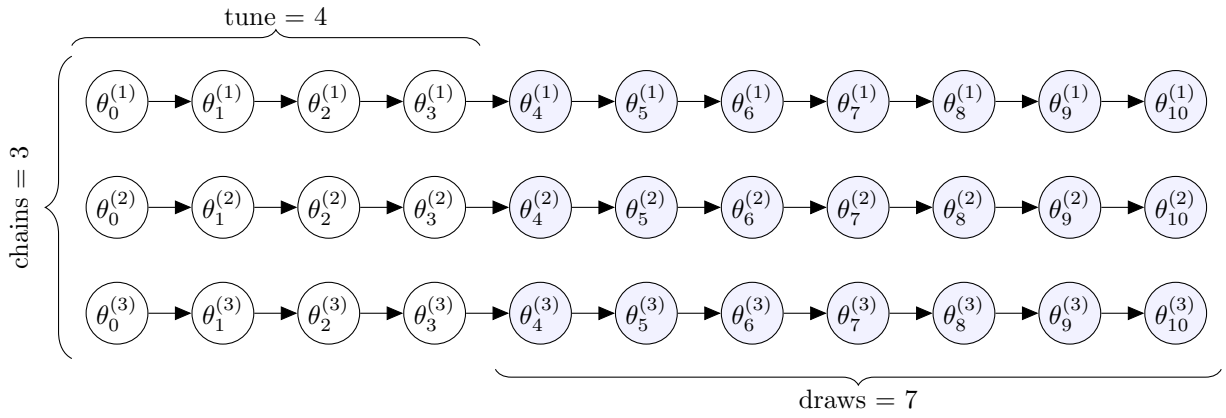


Figura 6.7: Ejemplo de experimento de muestreo. En este caso se simularon tres cadenas independientes (chains = 3), se esperaron cuatro pasos para considerar que se alcanzó el estado estacionario (tune = 4) y se recolectaron siete muestras efectivas en el presunto estado estacionario (draws = 7).

6.4.1.1. Muestreo de Gibbs

Supongamos que, debido a su complejidad, no podemos simular muestras de $\pi(x, y)$, pero que sí es posible generar muestras de las condicionales $\pi(x|y)$ y $\pi(y|x)$ (conocidas y fáciles de muestrear). El muestreo de Gibbs consiste en, a partir de un x_0, iterar alternadamente entre $y_t \sim \pi(y|x_t)$ y $x_{t+1} \sim \pi(x|y_t)$. Luego de suficientes pasos, al alcanzar el estado estacionario, los pares (x, y) estarán distribuidos por $\pi(x, y)$. En la siguiente sección se demostrará por qué este proceso cumple (6.65).

Esta técnica se extiende a más dimensiones muestreando de una componente a la vez. El problema con este muestreo es que es necesario conocer perfectamente todas las distribuciones condicionales, lo cual en la práctica suele ser problemático en muchos casos. Una excepción importante la constituyen las variables Bernoulli, donde se pueden computar todas las probabilidades necesarias.

A continuación se demostrará que el muestreo de Gibbs posee a la distribución a posteriori como estado estacionario y se mostrará un ejemplo de aplicación.

Demostración 6.1 (Estado estacionario) *Notar que, en esta cadena de Markov donde $\theta = (x, y)$, el proceso evoluciona como*

$$(x_0, y_0) \to (x_1, y_0) \to (x_1, y_1) \to (x_2, y_1) \to (x_2, y_2) \to (x_3, y_2) \to (x_3, y_3) \to \cdots \tag{6.67}$$

Para corroborar que cumple la condición suficiente de estacionariedad (6.65) *tomemos un paso de Gibbs $(x_t, y_t) \to (x_{t+1}, y_t)$:*

$$\pi(x_t, y_t)\pi(x_{t+1}|y_t) = \pi(x_{t+1}, y_t)\pi(x_t|y_t) \tag{6.68}$$

donde todas las distribuciones son computadas a posteriori *(la notación queda implícita). Es fácil notar que, si la medida de probabilidad es la misma, la identidad* (6.68) *representa la misma distribución conjunta de* (x_t, y_t, x_{t+1}) *en ambos lados de la igualdad.*

Ejemplo 6.3 *Sea la distribución correspondiente al modelo gráfico:*

$$P(x,y) \propto e^{x^T a + y^T b + x^T W y} \cdot \mathbf{1}\{x \in \{0,1\}^{d_x}, y \in \{0,1\}^{d_y}\} \tag{6.69}$$

donde $\{a, b, w\}$ *son valores conocidos. Explicar el procedimiento para muestrear esta distribución utilizando muestreo de Gibbs.*

Este tipo de distribución se conoce como modelo de Boltzmann restringido. Supongamos que buscamos muestrear esta distribución utilizando el muestreo de Gibbs. Se puede ver que la constante de proporcionalidad sería computacionalmente pesada de buscarla para altas dimensiones, pero sus condicionales son más sencillas. Empezando por $P(y|x)$, se puede ver que sus componentes son independientes:

$$P(y|x) \propto e^{(b+W^T x)^T y} \cdot \mathbf{1}\{y \in \{0,1\}^{d_y}\} = \prod_{j=1}^{d_y} e^{[b+W^T x]_j y_j} \cdot \mathbf{1}\{y_j \in \{0,1\}\} \tag{6.70}$$

y por lo tanto $P(y_j = 1|x) \propto e^{[b+W^T x]_j}$ y $P(y_j = 0|x) \propto 1$. Juntando esta información puede verse que $y|x \sim \text{Ber}(\sigma(b + W^T x))$[2] es una vector Bernoulli de componentes independientes. Análogamente, puede deducirse que $x|y \sim \text{Ber}(\sigma(a + Wy))$.

En este caso, podrían generarse muestras inicializando x_0 y $k = 0$ e iterando entre:

- Se genera un y_k a partir de $\text{Ber}(\sigma(b + W^T x_k))$.
- Se genera un x_{k+1} a partir de $\text{Ber}(\sigma(a + W y_k))$.
- $k \leftarrow k + 1$.

6.4.1.2. Muestreo Metropolis

El muestreo Metropolis es usado típicamente en variables aleatorias discretas no binarias (como Poisson, geométrica, hipergeométrica, etc.) que toman valores en los enteros $\mathbb{Z}$, así como también en variables continuas donde no hay diferenciabilidad debido al modelo. Su característica esencial es que solamente le basta con conocer la distribución (conjunta, de todos los parámetros simultáneamente) salvo una constante de proporcionalidad. Es decir que si $\pi(\theta) = \frac{f(\theta)}{Z}$ con $Z > 0$, es suficiente con conocer $f(\theta)$ (que habitualmente se plantea como distribución *a priori* por verosimilitud).

Este tipo de muestreo propone transicionar de θ_t a θ_{t+1} con el siguiente algoritmo simétrico:

[2] La función sigmoide es $\sigma(z) = \frac{1}{1+e^{-z}} = \frac{e^z}{e^z+1}$.

1. Se genera una $\theta' = \theta_t + \delta$, donde δ es una variable aleatoria; en el caso que θ' no esté en el soporte de la variable (por ejemplo una Poisson no puede tomar valores negativos), se repite el proceso. Para el caso discreto, típicamente se propone que $\delta \sim \mathcal{U}\{-1, 0, 1\}$ (uniforme discreta de 3 átomos) y para el caso continuo se propone $\delta \sim \mathcal{N}(0, \sigma^2)$.

2. Se sortea una variable aleatoria Bernoulli de probabilidad $\alpha(\theta_t, \theta')$. Si dicha variable vale 1, $\theta_{t+1} = \theta'$. Caso contrario $\theta_{t+1} = \theta_t$.

donde

$$\alpha(\theta_a, \theta_b) = \text{mín}\left\{1, \frac{f(\theta_b)}{f(\theta_a)}\right\} \tag{6.71}$$

con $\pi(\theta) = \frac{f(\theta)}{Z}$ (no depende de la constante de normalización). A continuación se efectuará un análisis para demostrará que la distribución *a posteriori* cumple la condición de estacionariedad dada por (6.65).

Demostración 6.2 (Estado Estacionario - Caso Discreto) *La probabilidad de transición del caso discreto descripta anteriormente es de la forma*

$$P(\theta_t \to \theta_{t+1}) = \begin{cases} \frac{\alpha(\theta_t, \theta_t - 1)}{3} & \theta_{t+1} = \theta_t - 1 \\ \frac{\alpha(\theta_t, \theta_t + 1)}{3} & \theta_{t+1} = \theta_t + 1 \\ 1 - \frac{\alpha(\theta_t, \theta_t - 1) + \alpha(\theta_t, \theta_t + 1)}{3} & \theta_{t+1} = \theta_t \\ 0 & \textit{Otros} \end{cases} \tag{6.72}$$

El único caso no trivial que vale la pena analizar en (6.65) *es que ocurre si* $\theta_{t+1} = \theta_t \pm 1$*; en los demás casos, la condición se cumple automáticamente. En este caso*

$$\pi(\theta_t)\frac{\alpha(\theta_t, \theta_{t+1})}{3} = \pi(\theta_{t+1})\frac{\alpha(\theta_{t+1}, \theta_t)}{3} \tag{6.73}$$

Se puede ver que si $\pi(\theta) = \frac{f(\theta)}{Z}$*, la ecuación en cuestión puede reducirse a*

$$f(\theta_t)\alpha(\theta_t, \theta_{t+1}) = f(\theta_{t+1})\alpha(\theta_{t+1}, \theta_t) \tag{6.74}$$

Utilizando el $\alpha(\theta_a, \theta_b)$ *definido en* (6.71)*, se puede comprobar la identidad* (6.74)*. El primer término cumple que* $f(\theta_t)\alpha(\theta_t, \theta_{t+1}) = \text{mín}\{f(\theta_t), f(\theta_{t+1}\}$*, y de forma análoga para el otro término* $f(\theta_{t+1})\alpha(\theta_{t+1}, \theta_t) = \text{mín}\{f(\theta_{t+1}), f(\theta_t)\}$*. De esta manera, queda garantizando que la distribución* a posteriori *es un estado estacionario de la cadena.*

Demostración 6.3 (Estado Estacionario - Caso Continuo) *La transición del caso continuo tiene una distribución mixta: una mezcla entre una normal y una masa*

puntual (delta de Dirac) en $\theta_{t+1} = \theta_t$. Dado que el único caso relevante a chequear de (6.65) *es el caso donde $\theta_{t+1} \neq \theta_t$, basta con analizar la parte continua: $P(\theta_t \to \theta_{t+1}) = \alpha(\theta_t, \theta_{t+1}) \cdot \mathcal{N}(\theta_{t+1}|\theta_t, \sigma^2)$ donde $\mathcal{N}(x|\mu, \sigma^2)$ hace referencia a la densidad de una normal de parámetros μ y σ^2 evaluada en x. En este caso la condición* (6.65) *puede escribirse como:*

$$\pi(\theta_t) \cdot \alpha(\theta_t, \theta_{t+1}) \cdot \mathcal{N}(\theta_{t+1}|\theta_t, \sigma^2) = \pi(\theta_{t+1}) \cdot \alpha(\theta_{t+1}, \theta_t) \cdot \mathcal{N}(\theta_t|\theta_{t+1}, \sigma^2) \tag{6.75}$$

La condición anterior puede reducirse a $f(\theta_t)\alpha(\theta_t, \theta_{t+1}) = f(\theta_{t+1})\alpha(\theta_{t+1}, \theta_t)$ usando que la normal es simétrica respecto a la media $\mathcal{N}(\theta_t|\theta_{t+1}, \sigma^2) = \mathcal{N}(\theta_{t+1}|\theta_t, \sigma^2)$. De esta manera llegamos a la misma identidad que en el caso discreto y por lo tanto, podemos concluir que la distribución a posteriori *cumple la condición de estacionariedad* (6.65).

6.4.1.3. NUTS (No-U-Turn Sampler)

El algoritmo NUTS (No-U-Turn Sampler) es un método de muestreo para variables aleatorias continuas con distribución a posteriori es diferenciable. La versión completa del algoritmo puede verse en la Fig. 6.8; a continuación se presentarán las ideas generales [17]. El algoritmo introduce una variable auxiliar $r \in \mathbb{R}^d$ cuyo objetivo es actuar como dirección de exploración de la distribución *a posteriori*. Esta variable se genera en cada iteración a partir de una distribución normal estándar multivariada. A continuación, se define la función de energía, como:

$$H(\theta, r) = -\log \pi(\theta) + \frac{1}{2}\|r\|^2 \tag{6.76}$$

Notar que una constante de proporcionalidad en $\pi(\theta)$ se transformaría en una constante sumando y podría omitirse sin afectar el procedimiento. Esta función de energía combina la log-posterior de θ con una penalización cuadrática sobre r y tiene un papel central en la determinación de la calidad de las propuestas. En cada iteración, partiendo del estado actual (θ_t, r), el algoritmo genera una secuencia de nuevas propuestas (θ, r) mediante un método numérico que utiliza la información del gradiente de la energía. Este procedimiento se conoce como integración tipo **leapfrog**, y consiste en aplicar transformaciones que mantengan aproximadamente constante el valor de $H(\theta, r)$. Así como el gradiente descendente desplaza los parámetros hacia la dirección de decrecimiento de la función objetivo, la integración leapfrog desplaza los parámetros sobre una curva de nivel de $H(\theta, r)$.

Una característica distintiva de NUTS es que, en lugar de requerir un número fijo de pasos de integración (como en otros algoritmos relacionados), a partir de θ_t, construye dinámicamente un árbol de posibles (θ, r), expandiéndose hacia adelante y hacia atrás, hasta que detecta que continuar expandiendo llevaría a una región ya visitada o a una

Algorithm 6 No-U-Turn Sampler with Dual Averaging

Given θ^0, δ, $\mathcal{L}$, M, M^{adapt}:
Set $\epsilon_0 = \text{FindReasonableEpsilon}(\theta)$, $\mu = \log(10\epsilon_0)$, $\bar{\epsilon}_0 = 1$, $\bar{H}_0 = 0$, $\gamma = 0.05$, $t_0 = 10$, $\kappa = 0.75$.
for $m = 1$ to M **do**
  Sample $r^0 \sim \mathcal{N}(0, I)$.
  Resample $u \sim \text{Uniform}([0, \exp\{\mathcal{L}(\theta^{m-1} - \frac{1}{2} r^0 \cdot r^0\}])$
  Initialize $\theta^- = \theta^{m-1}$, $\theta^+ = \theta^{m-1}$, $r^- = r^0, r^+ = r^0, j = 0, \theta^m = \theta^{m-1}, n = 1, s = 1$.
  while $s = 1$ **do**
    Choose a direction $v_j \sim \text{Uniform}(\{-1, 1\})$.
    if $v_j = -1$ **then**
      $\theta^-, r^-, -, -, \theta', n', s', \alpha, n_\alpha \leftarrow \text{BuildTree}(\theta^-, r^-, u, v_j, j, \epsilon_{m-1}\theta^{m-1}, r^0)$.
    else
      $-, -, \theta^+, r^+, \theta', n', s', \alpha, n_\alpha \leftarrow \text{BuildTree}(\theta^+, r^+, u, v_j, j, \epsilon_{m-1}, \theta^{m-1}, r^0)$.
    end if
    if $s' = 1$ **then**
      With probability $\min\{1, \frac{n'}{n}\}$, set $\theta^m \leftarrow \theta'$.
    end if
    $n \leftarrow n + n'$.
    $s \leftarrow s' \mathbb{I}[(\theta^+ - \theta^-) \cdot r^- \geq 0] \mathbb{I}[(\theta^+ - \theta^-) \cdot r^+ \geq 0]$.
    $j \leftarrow j + 1$.
  end while
  if $m \leq M^{\text{adapt}}$ **then**
    Set $\bar{H}_m = \left(1 - \frac{1}{m + t_0}\right) \bar{H}_{m-1} + \frac{1}{m + t_0}(\delta - \frac{\alpha}{n_\alpha})$.
    Set $\log \epsilon_m = \mu - \frac{\sqrt{m}}{\gamma} \bar{H}_m, \log \bar{\epsilon}_m = m^{-\kappa} \log \epsilon_m + (1 - m^{-\kappa}) \log \bar{\epsilon}_{m-1}$.
  else
    Set $\epsilon_m = \bar{\epsilon}_{M^{\text{adapt}}}$.
  end if
end for

function $\text{BuildTree}(\theta, r, u, v, j, \epsilon, \theta^0, r^0)$
if $j = 0$ **then**
  Base case—take one leapfrog step in the direction v.
  $\theta', r' \leftarrow \text{Leapfrog}(\theta, r, v\epsilon)$.
  $n' \leftarrow \mathbb{I}[u \leq \exp\{\mathcal{L}(\theta') - \frac{1}{2} r' \cdot r'\}]$.
  $s' \leftarrow \mathbb{I}[u < \exp\{\Delta_{\max} + \mathcal{L}(\theta') - \frac{1}{2} r' \cdot r'\}]$.
  return $\theta', r', \theta', r', \theta', n', s', \min\{1, \exp\{\mathcal{L}(\theta') - \frac{1}{2} r' \cdot r' - \mathcal{L}(\theta^0) + \frac{1}{2} r^0 \cdot r^0\}\}, 1$.
else
  Recursion—implicitly build the left and right subtrees.
  $\theta^-, r^-, \theta^+, r^+, \theta', n', s', \alpha', n'_\alpha \leftarrow \text{BuildTree}(\theta, r, u, v, j - 1, \epsilon, \theta^0, r^0)$.
  if $s' = 1$ **then**
    if $v = -1$ **then**
      $\theta^-, r^-, -, -, \theta'', n'', s'', \alpha'', n''_\alpha \leftarrow \text{BuildTree}(\theta^-, r^-, u, v, j - 1, \epsilon, \theta^0, r^0)$.
    else
      $-, -, \theta^+, r^+, \theta'', n'', s'', \alpha'', n''_\alpha \leftarrow \text{BuildTree}(\theta^+, r^+, u, v, j - 1, \epsilon, \theta^0, r^0)$.
    end if
    With probability $\frac{n''}{n' + n''}$, set $\theta' \leftarrow \theta''$.
    Set $\alpha' \leftarrow \alpha' + \alpha''$, $n'_\alpha \leftarrow n'_\alpha + n''_\alpha$.
    $s' \leftarrow s'' \mathbb{I}[(\theta^+ - \theta^-) \cdot r^- \geq 0] \mathbb{I}[(\theta^+ - \theta^-) \cdot r^+ \geq 0]$
    $n' \leftarrow n' + n''$
  end if
  return $\theta^-, r^-, \theta^+, r^+, \theta', n', s', \alpha', n'_\alpha$.
end if

Figura 6.8: Algoritmo NUTS presentado por Hoffman en [17].

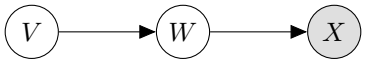


Figura 6.9: Ejemplo de modelado de 3 variables $V \to W \to X$, donde la única variable observable es X.

dirección contraria a la actual, según un criterio geométrico. Esta condición de detención se conoce como la "No-U-Turn". Una vez elegido un candidato (θ', r'), se acepta el cambio $\theta_{t+1} = \theta'$ con una determinada probabilidad, por el contrario, se decide por $\theta_{t+1} = \theta_t$.

Una vez finalizado este período, el algoritmo congela sus parámetros de control y comienza a generar las muestras válidas. En el contexto de modelos con parámetros continuos, NUTS es especialmente eficiente en espacios de alta dimensión, ya que genera propuestas informadas por la geometría local de la distribución, evitando caminatas aleatorias ineficientes y generando muestras con menor autocorrelación. Esto lo convierte en el algoritmo por defecto en muchas librerías bayesianas modernas como `PyMC`.

6.4.1.4. Ejemplo de Modelo Complejo

En la Fig. 6.9 se muestra un ejemplo de modelado con tres variables $V \to W \to X$, donde la única variable observable es $X = x$. Supongamos que a priori $V \sim \exp(2)$, que $W|_{V=v} \sim \text{Poi}(v)$ y que $X|_{W=w} \sim \chi^2(w+1)$. En este caso, las cadenas se generan con el siguiente procedimiento.

1. Se inicializa v_0, usualmente utilizando una muestra de la distribución a priori exp(2).

2. Se inicializa w_0, a partir de su distribución latente $\text{Poi}(v_0)$ o eventualmente con Poi(0.5) (usando la media de V en lugar de su valor).

3. Se actualiza V, definiendo v_1. Como V es continua derivable, habitualmente se utilizará NUTS aplicando sobre la distribución *a posteriori* $\pi(v, w_0) \propto p(x|w_0)P(w_0|v)p(v) \propto \text{Poi}(w_0|v) \cdot \exp(v|2)$, donde $\text{Poi}(\cdot|\mu)$ es la función de probabilidad de una Poisson de media μ y $\exp(\cdot|\lambda)$ es la función de densidad de una exponencial de intensidad λ. En este caso se absorbió la verosimilitud como constante de proporcionalidad por no depender v.

4. Se actualiza W, definiendo w_1. Al tratarse de una variable discreta, se recomienda utilizar el muestreo Metrópolis. Para el muestreo se utilizará la distribución *a posteriori* $\pi(v_1, w) \propto p(x|w)P(w|v_1)p(v_1) = \chi^2(x|w+1) \cdot \text{Poi}(w|v_1)$ donde $\chi^2(\cdot|\nu)$ es la función de densidad de una *chi-cuadrado* de ν grados de libertad. En este caso se absorbió la distribución a priori $p(v_1)$ como constante de proporcionalidad por no depender w.

5. Se repite el paso (**3**) definiendo v_2 y se continúa iterando la cantidad de pasos que sea necesario.

6.4.2. Calidad de las muestras

Para evaluar la calidad de las muestras de un experimento de MCMC, suelen considerarse dos propiedades: la ergodicidad y la estacionariedad. Gracias al teorema de ergodicidad, podemos aproximar esperanzas a partir de promedios sin pretender que las muestras sean independientes. El problema radica en que la velocidad de convergencia y la varianza de dicho promedio no son las mismas que en el caso de variables independientes. En MCMC, se denomina tamaño de muestra efectiva (Effective Sample Size, ESS) a la cantidad de datos independientes necesarios para alcanzar la misma varianza que posee el promedio de las muestras. Se define como

$$\text{ESS} = \frac{t_{\max}}{1 + 2\sum_{t=1}^{k_{\max}} \rho_t} \tag{6.77}$$

donde ρ_t es la autocorrelación de la cadena y $k_{\max}$ es el valor a partir del cual las autocorrelaciones se vuelven pequeñas o negativas. La ecuación (6.77) será demostrada más adelante. Esta ESS se conoce como **bulk** y se utiliza para cálculos predictivos. Para intervalos de confianza suele usarse otro ESS denominado **tail**.

Otra característica importante, además de la ergodicidad, es verificar si las muestras fueron generadas una vez alcanzado el estado estacionario. Supongamos que se cuenta con una simulación con varias cadenas independientes. Si todas convergieron a la misma distribución, entonces la varianza entre cadenas debería ser similar a la varianza dentro de cada cadena. Se denomina R-hat (o $\hat{R}$, también conocido como diagnóstico Gelman–Rubin) al cociente entre estas varianzas. Si las cadenas aún no convergieron, habrá más variabilidad entre cadenas, y $\hat{R}$ será mayor que 1. En la práctica suele considerarse $\hat{R} > 1.01$ una señal de alerta y un valor $\hat{R} > 1.1$ suele considerarse un problema a resolver.

Demostración 6.4 (Cálculo de ESS) *En cálculo predictivo numérico, la hipótesis de trabajo es aproximar una esperanza con el promedio de las densidades $p_{X|T=\theta_i}(x)$ en lugar de los parámetros pero, en la práctica, se suele analizar la varianza de los parámetros para estandarizar resultados. No sería particularmente complejo trabajar con las verosimilitudes evaluadas en los parámetros, pero no hay grandes diferencias. En última instancia, bajo ciertas condiciones de regularidad, deberían ser comparables ambas ESS.*

Sea $\theta_1, \cdots, \theta_{t_{max}}$ un conjunto de muestras idénticamente distribuidas y sea $\bar{\theta}$ el promedio de las mismas; es simple ver que la varianza si fueran independientes sería de $\frac{\sigma^2}{ESS}$, donde $\sigma^2 = var(\theta_i)$. En el caso de existir un $\rho_t = \frac{cov(\theta_i, \theta_{i+t})}{\sigma^2}$, la varianza se

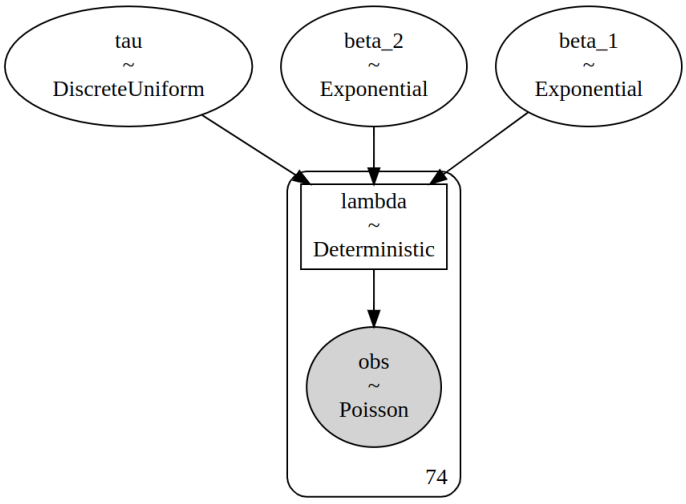


Figura 6.10: Modelo de `PyMC` generado por `model_to_graphviz`.

puede calcular como:

$$var(\bar{\theta}) = var\left(\frac{1}{t_{max}}\sum_{t=1}^{t_{max}}\theta_t\right) = \frac{1}{t_{max}^2}\sum_{i=1}^{t_{max}}\sum_{j=1}^{t_{max}} cov(\theta_i, \theta_j) \tag{6.78}$$

$$= \frac{1}{t_{max}^2}\left(\sigma^2 \cdot t_{max} + 2\sigma^2 \sum_{i=1}^{t_{max}-1}\sum_{j=i+1}^{t_{max}} \rho_{j-i}\right) \tag{6.79}$$

$$= \frac{\sigma^2}{t_{max}}\left(1 + 2\sum_{t=1}^{t_{max}-1}\left(1 - \frac{t}{t_{max}}\right)\cdot \rho_t\right) \tag{6.80}$$

donde se aplicó el cambio de variables $t = j - i$*. Bajo las hipótesis usuales en modelos MCMC, es razonable que tanto* ρ_t *como* $1 - \frac{t}{t_{max}}$ *se vayan achicando a medida que se avanza en la cadena. Los algoritmos suelen truncar la suma, en un* k_{max}*, cuando las autocorrelaciones se vuelven pequeñas o negativas. Con este procedimiento suele ocurrir que* $k_{max} \ll t_{max}$*, por lo que despreciando* $\frac{t}{t_{max}}$ *se puede aproximar:*

$$var(\bar{\theta}) \approx \frac{\sigma^2}{t_{max}}\left(1 + 2\sum_{t=1}^{k_{max}} \rho_t\right) \tag{6.81}$$

Igualando este resultado con la expresión de la varianza para variables independientes $\frac{\sigma^2}{ESS}$*, y despejando, se obtiene* (6.77) *finalizando la demostración.*

6.4.3. Introducción a `PyMC`

`PyMC` es una biblioteca de Python para inferencia estadística bayesiana, que permite construir modelos probabilísticos complejos y realizar inferencia sobre ellos de forma automatizada [6]. Es muy intuitivo de usar, el programador debe simplemente describir el grafo a implementar.

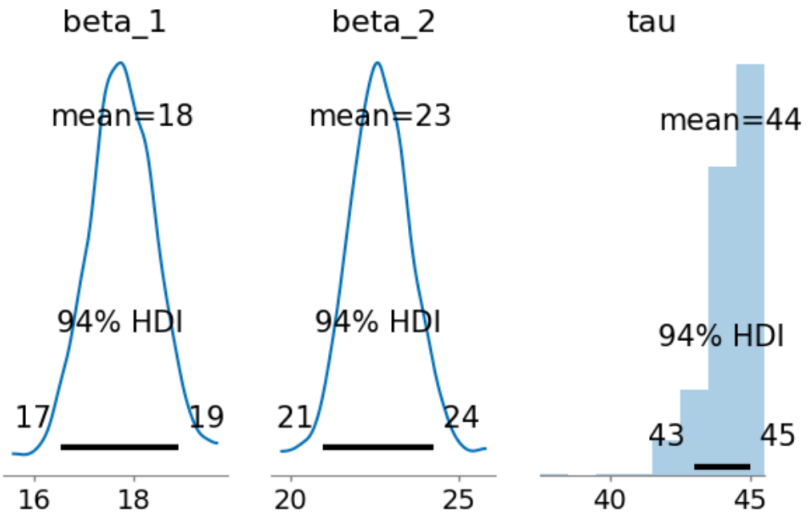


Figura 6.11: Gráfico de las densidades/probabilidades *a posteriori* generado la función `plot_posterior` de `PyMC`.

Supongamos que queremos implementar el modelo descripto en la Sec. 6.1.2. El mismo puede ser definido mediante el siguiente código

```
import pymc as pm
import numpy as np

count_data = np.loadtxt("txtdata.csv")
n_count_data = len(count_data)
idx = np.arange(n_count_data) # Index
alpha = 1.0/count_data.mean()
with pm.Model() as model:
  beta_1 = pm.Exponential("beta_1", alpha)
  beta_2 = pm.Exponential("beta_2", alpha)
  tau = pm.DiscreteUniform("tau", lower=0, upper=n_count_data - 1)
  lambda_func = pm.math.switch(tau > idx, beta_1, beta_2)
  lambda_ = pm.Deterministic("lambda",lambda_func)
  observation = pm.Poisson("obs", lambda_, observed=count_data)
pm.model_to_graphviz(model)
```

donde la última instrucción `pm.model_to_graphviz(model)` graficó la Fig. 6.10 como resultado. El código describe el grafo dentro del *entorno* `Model`, mencionando que datos corresponden a la variable observable (`count_data`). Para muestrear la distribución a posteriori, basta con utilizar el comando `sample` indicando en número de *draws*, *tune* y *chains* deseado.

```
with model:
  trace = pm.sample(draws=1000, tune=1000, chains=2)


import arviz as az
summary = az.summary(trace, var_names=["beta_1","beta_2","tau"])
print(summary)
```

donde la función `summary` nos permite conocer el ESS y el $\hat{R}$ por variable. El objeto `trace` contiene las muestras generadas durante el proceso.

```
beta_1_samples = trace.posterior['beta_1'].values
beta_2_samples = trace.posterior['beta_2'].values
tau_samples = trace.posterior['tau'].values
lambda_samples = trace.posterior['lambda'].values
_ = pm.plot_posterior(trace.posterior[['beta_1','beta_2','tau']])
```

donde la función `plot_posterior` generó el gráfico de las densidades/probabilidades *a posteriori* de la Fig. 6.11.

En el caso de quererse calcular la distribución predictiva o sus derivados (alguna probabilidad, la esperanza, la varianza), se deberán combinar las muestras de `trace` para lograrlo (usualmente se recomienda trabajar cada cadena por separado para corroborar que los resultados sean estacionarios). Pero, puede existir el caso donde lo que efectivamente se desee son muestras de la distribución predictiva[3]. Para ello basta con programar:

```
with model:
  posterior_pred = pm.sample_posterior_predictive(trace,predictions=True)
pred_samples = posterior_pred.predictions['obs'].values
```

La función `sample_posterior_predictive` genera una cantidad de muestras igual a las observadas durante el entrenamiento en cada paso de la cadena. Por lo que si uno desea un solo ejemplo de muestra, podría quedarse con `pred_samples[0,-1]`: De la cadena (0) quedarse con el último (-1) eslabón.

[3]Un muestreo de una distribución aproximada por muestreo. La calidad de estas muestras es menor debido al doble muestreo, por lo que no es recomendable usarlas para estimar la distribución predictiva.

Bibliografía

[1] W. Feller, *An Introduction to Probability Theory and Its Applications, Volume I.* USA: Society for Industrial and Applied Mathematics, 1969.

[2] T. M. Cover and J. A. Thomas, *Elements of Information Theory.* Wiley Series in Telecommunications and Signal Processing, Wiley-Interscience, 2006.

[3] R. Duda, P. Hart, and D. Stork, *Pattern Classification.* John Wiley, 2 ed., 2001.

[4] G. Casella and R. Berger, *Statistical Inference.* Duxbury advanced series in statistics and decision sciences, Thomson Learning, 2nd ed., 2002.

[5] P. W. Zehna, "Invariance of Maximum Likelihood Estimators," *The Annals of Mathematical Statistics*, vol. 37, no. 3, p. 744, 1966.

[6] C. Davidson-Pilon, *Bayesian Methods for Hackers: Probabilistic Programming and Bayesian Inference.* Addison-Wesley Professional, 1st ed., 2015.

[7] K. P. Murphy, *Machine Learning: A Probabilistic Perspective.* The MIT Press, 2012.

[8] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning.* Springer Series in Statistics, Springer New York Inc., 2001.

[9] K. Petersen and M. Pedersen, *The Matrix Cookbook.* Technical University of Denmark, 2012.

[10] A. Cauchy, "Méthode générale pour la résolution des systèmes d'équations simultanées," *Comp. Rend. Sci. Paris*, vol. 25, pp. 536–538, 1847.

[11] D. Wolpert and W. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, pp. 67–82, April 1997.

[12] D. G. Luenberger and Y. Ye, *Linear and Nonlinear Programming.* New York: Springer, 3rd ed., 2008.

[13] R. A. Horn and C. R. Johnson, *Matrix Analysis.* Cambridge University Press, 2nd ed., 2012.

[14] A. H. Sayed, *Adaptive Filters.* Hoboken, NJ: Wiley–IEEE Press, 1st ed., Apr. 2008. Hardcover.

[15] M. G. González, M. Vera, A. Dreszman, and L. J. Rey Vega, "Diffusion assisted image reconstruction in optoacoustic tomography," *Optics and Lasers in Engineering*, vol. 178, 2024.

[16] C. M. Bishop, *Pattern Recognition and Machine Learning*. Information Science and Statistics, Springer-Verlag New York, Inc., 2006.

[17] M. Hoffman and A. Gelman, "The No-U-Turn sampler: adaptively setting path lengths in hamiltonian monte carlo," *Journal of Machine Learning Research*, vol. 15, p. 1593–1623, Jan. 2014.