

Fit_1-Clase

August 12, 2016

1 Qué significa hacer un ajuste ?

1.1 1. Repaso de Probabilidad

En la clase anterior, definimos la probabilidad como la *plausibilidad* de que cierta afirmación sea cierta:

$$P(A)$$

Las propiedades de la probabilidad son:

1. Es un número entre 0 y 1
2. Regla de la suma:

$$P(A|B) + P(\bar{A}|B) = 1$$

3. Regla del producto:

$$\begin{aligned} P(AB|C) &= P(A|BC) P(B|C) & (1) \\ &= P(B|AC) P(A|C) & (2) \end{aligned}$$

De la regla del producto, es inmediato despejar una fórmula que se conoce *el teorema de Bayes* que dice que:

$$P(A|BC) = \frac{P(B|AC)}{P(B|C)} P(A|C)$$

Es muy útil porque nos permite dar vuelta los condicionales.

En esta expresión se usa tanto que cada factor tiene un nombre y una interpretación: * El último factor es la plausibilidad de A sin tener en cuenta B .

Por ese motivo se suele denominar el factor *prior* * El denominador sólo depende de B y C . Si pensamos a la expresión como algo que evalúa la probabilidad de A , es un número que expresa la normalización en A (esto estará más claro en las aplicaciones). * El numerador de la fracción depende de A , pero del otro lado del condicional. Por eso, a menudo se lo llama *likelihood* (verosimilitud) en lugar de probabilidad, ya que no es una distribución, sino una función rara de A .

1.2 2. Ajuste de parámetros

En la realización de un experimento, tenemos en mente un modelo que tiene ciertos parámetros que encapsulamos en la variable θ . Al medir, resulta que mido x .

El problema del *ajuste* es ver qué información obtuve sobre θ al tener en cuenta la medición x .

En general puedo usar mi *modelo* de los datos para obtener la probabilidad:

$$P(x|\theta)$$

Como quiero estimar θ usando la medición, invierto el condicional utilizando el Teorema de Bayes:

$$P(\theta|x I) \propto P(x|\theta I) P(\theta I)$$

- La proposición I es la que incluye toda la información contextual
- Como me interesan las distribuciones, puedo abandonar el factor de normalización
- El primer factor lo puedo calcular y es la *likelihood*
- El último factor es la *prior* (lo que sé de los parámetros por información contextual)

En general tengo muchas mediciones independientes x_i , en ese caso:

$$P(\{x\}|\theta) \equiv \prod_i P(x_i|\theta)$$

Es evidente de lo que dijimos, pero vale la pena repetir que el resultado del ajuste es una distribución de parámetros, que representa la probabilidad de los parámetros $P(\theta|x I)$ dada la medición realizada y la información contextual. A menudo sólo nos interesará el valor más probable y el ancho de la distribución.

1.3 3. Ejemplo moneda

Supongamos que θ es la probabilidad de que una moneda salga cara ($x = 1$) al tirarla.

En ese contexto:

$$P(x|\theta) = \begin{cases} \theta & x = 1 \quad (\text{cara}) \\ 1 - \theta & x = 0 \quad (\text{ceca}) \end{cases}$$

y vamos a suponer una prior uniforme:

$$P(\theta) = \begin{cases} 1 & 0 < \theta < 1 \\ 0 & \text{caso contrario} \end{cases}$$

La prior uniforme se corresponde con no tener la más mínima idea acerca de la moneda en cuestión. Nuestra experiencia cotidiana con monedas tiene mucha más información que eso, pero resulta más simple comenzar siendo un poco sonzo.

1.3.1 3.1 Medición todas cara

Supongamos ahora que los datos medidos son todas caras. Si mido N tiradas, resulta que

$$P(\theta|\{x\}) = A \theta^N P(\theta)$$

y no es difícil obtener el factor de proporcionalidad que sale por normalización:

$$A = N + 1$$

Graficamos.

```

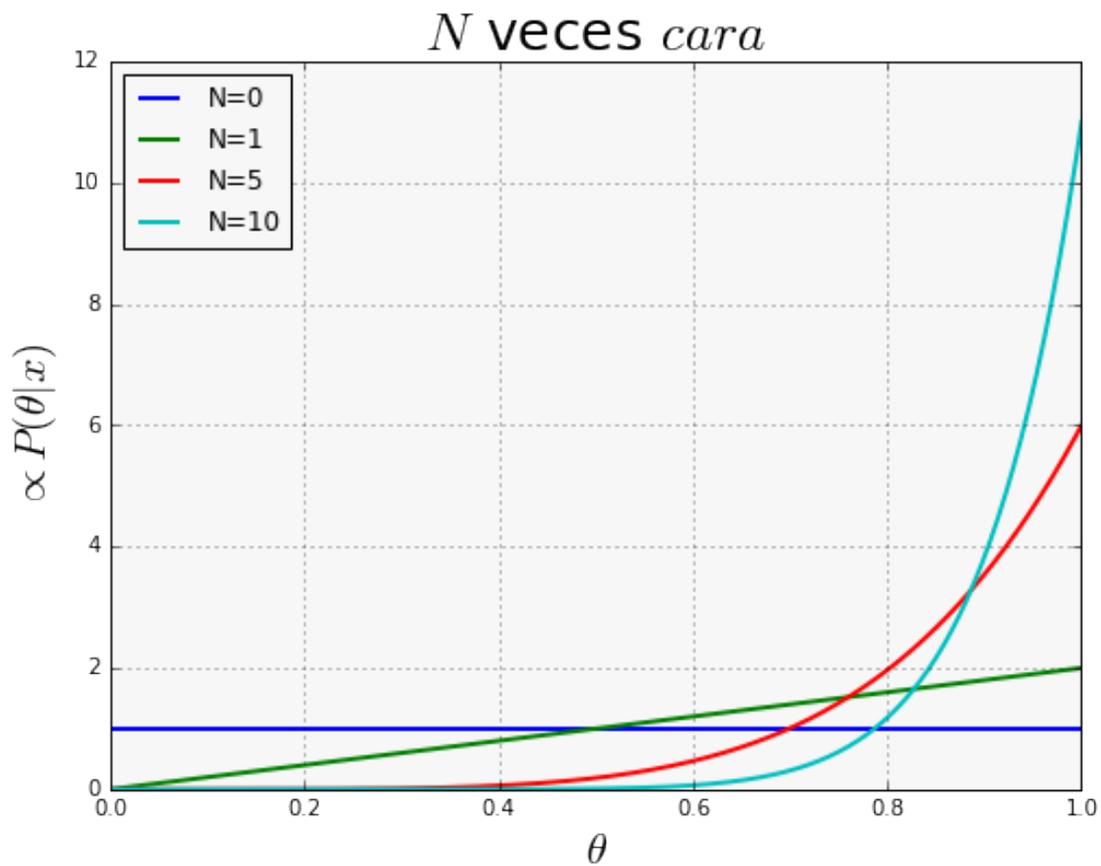
In [9]: %pylab inline
        tita = linspace(0,1,500)

        figure(figsize=(8,6))
        for N in [0,1,5,10]:
            plot(tita, (N+1)*tita**N, label='N={:d}'.format(N))

        legend(loc=0)
        grid()
        xlabel(r'\theta$', fontsize=20)
        ylabel(r'\propto P(\theta|x)$', fontsize=20)
        title(r'$N$ veces $cara$', fontsize=25);

```

Populating the interactive namespace from numpy and matplotlib



1.3.2 3.2 Balance cara-ceca

Si sale una vez cara y una vez ceca alternadamente, cada dos veces aparece el factor $\theta(1 - \theta)$ y resulta (N par):

$$P(\theta|\{x\}) = A \theta^{N/2} (1 - \theta)^{N/2} P(\theta)$$

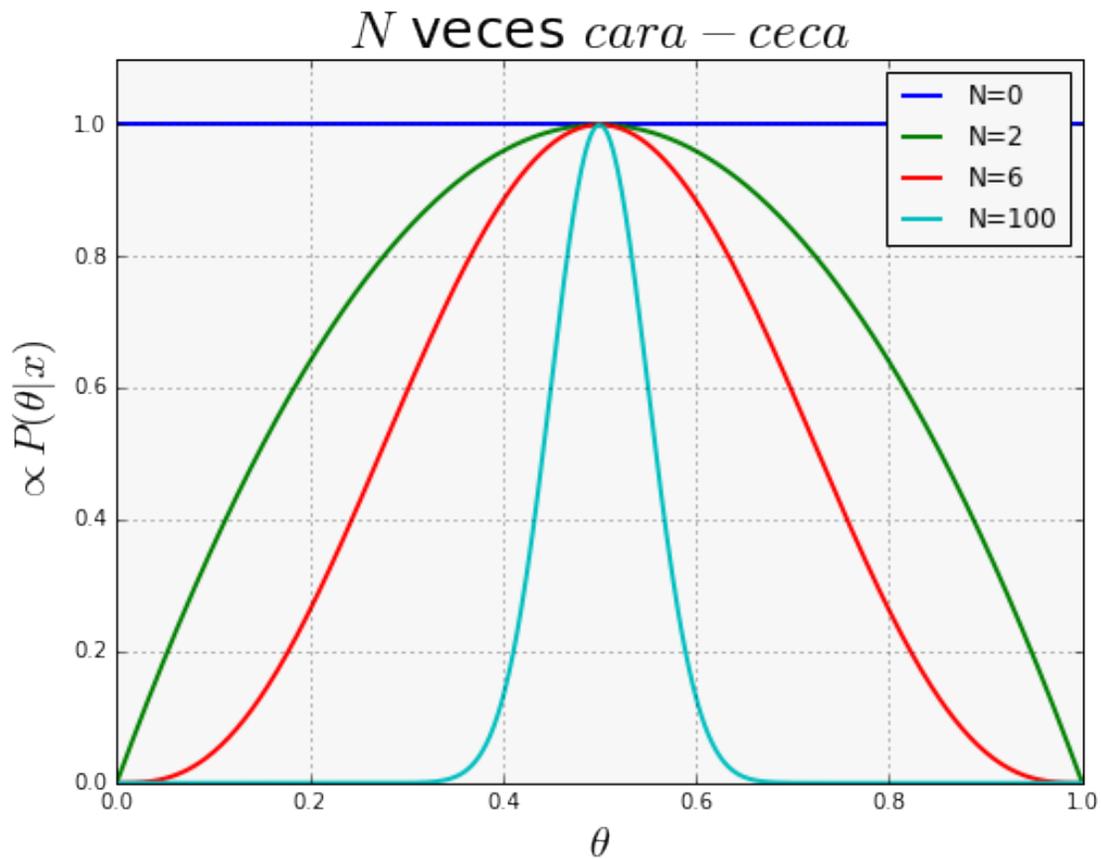
El factor de normalización es mu aburrido de calcular, pero para graficar la forma, podemos hacer que el máximo de la función tenga siempre el mismo valor. Para eso basta hacer

$$A = 2^N$$

Veamos lo que da.

```
In [10]: figure(figsize=(8,6))
         for N in [0,2,6,100]:
             plot(tita,2**N *tita**(N/2)*(1-tita)**(N/2),label='N={:d}'.format(N))

         legend()
         grid()
         xlabel(r'\theta$', fontsize=20)
         ylabel(r'\propto P(\theta|x)$', fontsize=20)
         title(r'$N$ veces $cara-ceca$', fontsize=25)
         ylim(0,1.1);
```



1.3.3 3.3 Medición al azar

Vamos a sintetizar datos al azar para un $\theta_{true} = 0.4$

Evito explícitamente hacer cuentas analíticas, aunque sé que se pueden, así me parece más instructivo.

```
In [11]: ## Sintetizo datos
N       = 1000 # Número de datos a sintetizar
ttrue   = 0.4  # Valor que uso para armar los datos

tita    = linspace(0,1,500)
values  = [0,1]
probs   = [1-ttrue, ttrue]
x       = random.choice(values, size=N, p=probs)
x       = array(x)

## Defino la Likelihood de una medición
def logLikeH(tita,x):
    ## Prior (más extremos)
    if tita <=0: return -inf
    if tita >=1: return -inf

    ## LH
    if x==1:
        return log(tita)
    return log(1-tita)

## Sumo los logaritmos de todo lo que medí
def lnProb(tita,x):
    ## El truco del try es para que acepte tita como un array
    try:
        out=[]
        for t in tita:
            out.append(lnProb(t,x))
        return array(out)
    except:
        return sum([logLikeH(tita,xi) for xi in x])

## Normalizo
Max = lnProb(0.4,x)

## Grafico el resultado del ajuste
figure(figsize=(8,6))
plot(tita,exp(lnProb(tita,x)-Max))

#legend()
grid()
xlabel(r'\theta$', fontsize=20)
```

```

ylabel(r'\propto P(\theta |x)', fontsize=20)
title(r'1000 valores sintetizados con $\theta_{true}=0.4$', fontsize=15)

## Si sólo me interesa el valor del máximo
from scipy.optimize import minimize
nll = lambda *args: -lnProb(*args)
result = minimize(nll, x0=[ttrue], args=(x))
titaMax = result['x'][0]

print(u'tita Máximo =', titaMax)
axvline(titaMax, color='r');

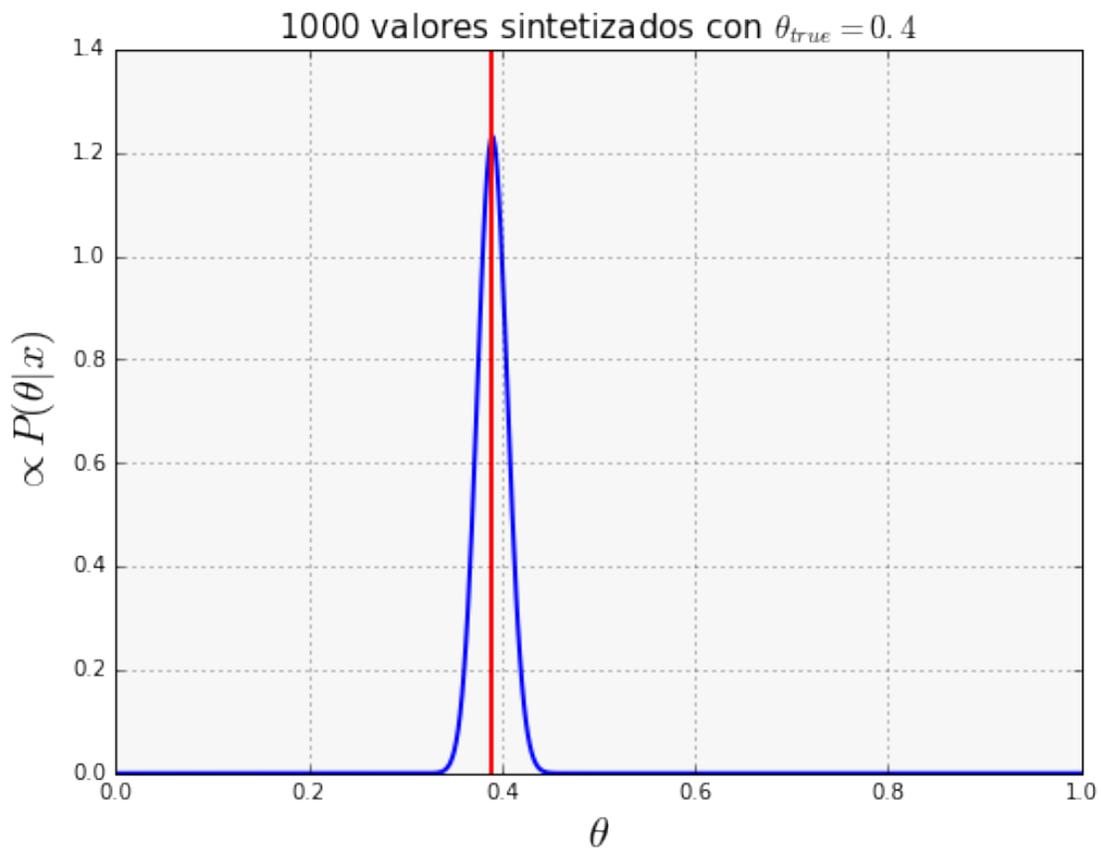
```

```

/home/willy/Programas/anaconda/lib/python3.5/site-packages/scipy/optimize/optimize.
grad[k] = (f*((xk + d,) + args)) - f0) / d[k]

```

tita Máximo = 0.389999992605



1.4 4. Ajuste de una curva de datos x, y

Un problema frecuente que aparece al analizar los datos, es que tenemos un parámetro de control del experimento x que variamos para realizar una medición y .

Este proceso se repite varias veces y tenemos un conjunto de datos $\{x_i, y_i\}$ que son independientes.

Vamos a suponer que los datos x_i los conozco exactamente.

En ese caso, la expresión del ajuste que obtuvimos antes es:

$$P(\theta | \{x, y\} I) \propto \prod_i P(y_i | x_i \theta) P(\theta I)$$

Por una cuestión numérica, en las probabilidades que aparecen en la productoria, pueden aparecer cosas de muy diferentes órdenes de magnitud. Resulta conveniente escribir este resultado como:

$$P(\theta | \{x, y\} I) \propto e^{-\chi^2/2} P(\theta I)$$

donde:

$$\chi^2_{(\theta)} = -2 \sum_i \ln P(y_i | x_i \theta)$$

Los valores de $P(y_i | x_i \theta)$ nos lo proporcionará el modelo.

En principio, el problema está resuelto dibujando la función $P(\theta | \{x, y\} I)$. Tengo ya todo para hacerlo.

1.5 5. Ruido gaussiano y prior uniforme

En el caso particular de ruido gaussiano y de tener una prior uniforme, las cosas se simplifican mucho.

Si el ruido es gaussiano, resulta que los datos responden a una distribución:

$$y_i \sim \mathcal{N}(\mu = f(x_i, \theta), \sigma = \sigma_i)$$

y eso hace que (por la definición de la gaussiana):

$$P(y_i | x_i \theta) \propto \exp -\frac{1}{2\sigma_i^2} (y_i - f(x_i, \theta))^2$$

y entonces:

$$\chi^2_{(\theta)} = \sum_i \frac{(y_i - f(x_i, \theta))^2}{\sigma_i^2}$$

Con estas definiciones, resulta inmediato ver que los valores más probables para los parámetros, dada la medición, son los que hacen que $\chi^2_{(\theta)}$ sea mínimo. Este es un famoso resultado que se conoce como el *método de cuadrados mínimos*.

Cómo se calcula el valor de θ que minimiza el χ^2 ?

Lo mejor es dejar eso a las computadoras. Pero algo conviene saber.

Para empezar, el problema tiene soluciones analíticas cuando la función es lineal en los parámetros. Un caso particular característico es el ajuste por una recta, y a eso nos dedicaremos a continuación.

Continuará...