Whatsappeando con Poisson

Anabel Forte, Gonzalo García-Donato y Joaquín Martínez-Minaya

Índice

1. Introducción: Whatsappeando con Poisson

WhatsApp es una de las aplicaciones de mensajería instantánea más utilizadas y fue creada por Jan Koum y Brian Acton. Esta aplicación permite comunicarnos con cualquier persona desde cualquier lugar del mundo y de forma inmediata. A día de hoy, ¿quién no conoce WhatsApp? El uso de WhatsApp, junto con otras redes sociales, se está popularizando cada vez más entre los jóvenes de entre diez y catorce años.

En este caso práctico, intentaremos estimar el número de mensajes de WhatsApp que recibe un adolescente durante una hora en un día no lectivo. Para ello, se preguntó a veinticinco adolescentes sobre el número de mensajes que recibían de esta aplicación en una hora. A continuación, se muestran los resultados de esos conteos.

```r
# set.seed(100)
# data <- data.frame(Wp = rpois(25, lambda = 140))
data <- data.frame(Wp = c(134, 121, 150, 141, 143, 133,
                          145, 135, 141, 141, 137,
                          154, 133, 135, 154, 143, 126,
                          149, 143, 149, 130, 129, 142, 126, 130))
```

2. Modelo de Poisson con previa gamma

Dado que estamos interesados en el número mensajes de WhatsApp que recibe un adolescente en una hora, asumimos que nuestra variable respuesta (Y), sigue una distribución de **Poisson** de parámetro λ, es decir:

$$Y \mid \lambda \sim Po(\lambda),$$

y cuya distribución de probabilidad, como se ha visto en teoría es:

$$p(Y = y \mid \lambda) = \frac{e^{-\lambda}\lambda^{y}}{y!},$$

siendo sus dos primeros momentos exactamente iguales.

- $\mathrm{E}(Y) = \lambda$
- $\mathrm{Var}(Y) = \lambda$

2.1. Verosimilitud de Poisson

En el apartado anterior hemos definido nuestra variable de interés y, con ella, la distribución de Poisson. Sin embargo, cuando hacemos un análisis estadístico disponemos de mediciones o, dicho de otra forma, de realizaciones de esa variable aleatoria. Para extraer información sobre esas mediciones y poder hacer inferencia sobre el parámetro de interés λ necesitamos calcular la función de verosimilitud.

Así, dado un conjunto de observaciones y_i para $i = 1, \ldots, 25$, la función de verosimilitud Poisson es proporcional a

$$L(\lambda \mid D) \propto e^{-n\lambda}\lambda^r,$$

donde $r = \sum_{i=1}^{n} y_i$, es decir, r es el número total de mensajes de WhatsApp que reciben los veinticinco adolescentes en una hora. Es importante darse cuenta de que estamos utilizando el signo $\propto$ para indicar que las dos cantidades son iguales, a falta de una constante. En concreto, la igualdad, como se ha visto en teoría, se logra diviendo por el productorio del factorial de cada y_i.

La expresión anterior se puede calcular en R con el siguiente código:

- Definimos la expresión de la verosimilitud.

```
y <- data$Wp
r <- sum(y)
Lpois <- function(y, lambda){
  prod(dpois(y, lambda = lambda))
}
```

- Creamos una variable auxiliar para el eje de las x o, dicho de otra forma, el parámetro del que depende la función de verosmilitud, que en este caso es λ. A continuación, calculamos el valor de la verosimilitud en esos puntos.

```
aux <- seq(100, 165,by = 0.01)
aux2 <- sapply(aux,FUN = Lpois,y=y)
```

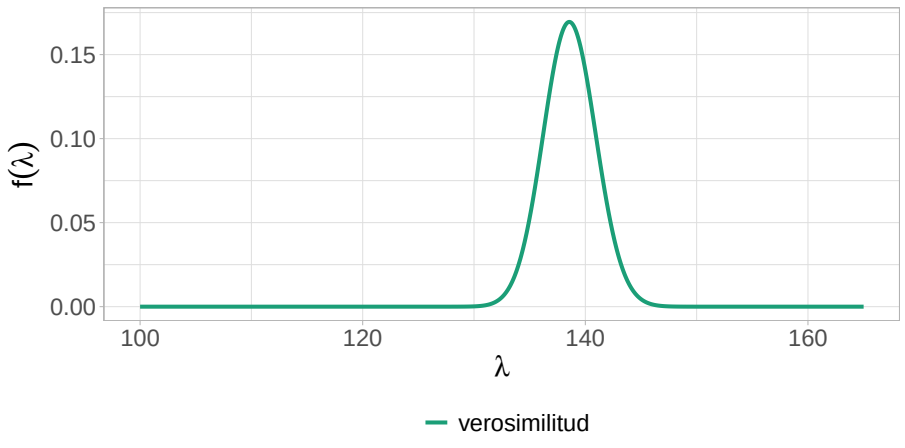
- Ahora bien, sabemos que la verosimilitud en si no es una distribución de probabilidad y, por tanto, no tiene por qué integrar 1. En este caso, vamos a hacer que integre 1 para poder compararla con la distribución a priori y la distribución a posteriori.

```r
C <- 1 / (0.01 * sum(aux2))
```

- Por último, la dibujamos, utilizando el paquete de R `ggplot2`.

```r
library(ggplot2)
data_vero <- data.frame(x = aux, y = C*aux2, class = "verosimilitud")


ggplot(data = data_vero) +
  geom_line(aes(x = x, y = y, color = class, linetype = class),
            size = 1.2) +
  scale_linetype_manual(values=c("solid"))+
  scale_color_brewer(palette="Dark2") +
  ylab(expression(f(lambda))) +
  xlab(expression(lambda)) +
  theme_light() +
  theme(axis.text  = element_text(size=15),
        axis.title = element_text(size=18),
        legend.text = element_text(size=15)) +
  theme(legend.position = "bottom", legend.title = element_blank())
```

```
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```



Ya tenemos cubierto uno de los elementos necesarios para hacer inferencia en el contexto bayesiano, la **verosimilitud**. Es el momento de definir la distribución a priori.

2.2. Distribución a priori: gamma

Sabemos que λ es un parámetro que toma valores en el intervalo $(0, \infty)$, y dado que la distribución **gamma** es su distribución a priori conjugada, la utilizaremos para asignar información previa sobre el parámetro λ.

Así, consideramos la distribución a priori $\lambda \sim \text{gamma}(a, b)$, con función de densidad:

$$\pi(\lambda \mid a, b) = \lambda e^{-b\lambda} \frac{(b\lambda)^a}{\Gamma(a)} \lambda^{a-1}.$$

$\Gamma(\cdot)$ representa la función gamma, que para cualquier entero positivo se define como $\Gamma(a) = (a-1)!$. Los primeros dos momentos de esta distribución son:

- $E(\lambda) = \frac{a}{b}$
- $Var(\lambda) = \frac{a}{b^2}$

Dado que estamos utilizando la función gamma como distribución a priori, los parámetros a y b serán los encargados de resumir la **información previa** que tenemos sobre λ. Veamos cómo podemos establecer esos dos parámetros:

1. Podemos pensar en un adolescente cercano a nosotros, o incluso en nosotros mismos. ¿Cuántos mensajes de WhatsApp recibimos en una hora en un día no lectivo?
2. ¿Cuánto confiamos en esa cifra? Esto va a determinar cómo de grande es el error que cometemos, es decir, va a determinar la variabilidad de la distribución a priori.
3. Una vez hemos fijado el **número** y el **error**, calculamos a y b de la media y la varianza de la distribución gamma. Por ejemplo, si pensamos que un adolescente recibe 128 mensajes con un error de 60 (una desviación estándar de 4 más o menos), obtendremos los valores de a y b resolviendo el siguiente sistema de ecuaciones:

$$a/b = 128$$

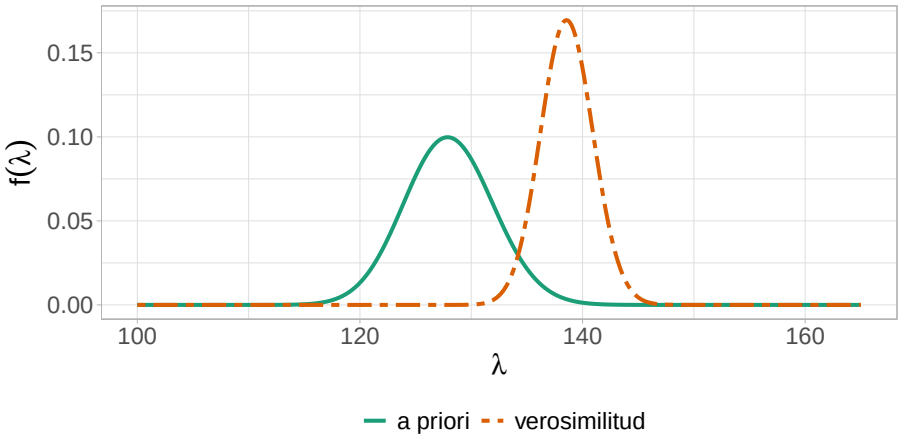
y

$$a/b^2 = 16\,.$$

Despejando, $a = 128b$, $128 = 16b$. Por tanto, obtenemos que $b = 8$ y $a = 128 * 8 = 1024$.

Representamos a continuación el gráfico anterior junto con la distribución a priori.

```
a0 <- 1024
b0 <- 8

data_prior <- data.frame(
  x = aux,
  y = dgamma(aux,shape = a0, rate = b0),
  class = "a priori")

data_plot <- rbind(data_vero, data_prior)

ggplot(data = data_plot) +
  geom_line(aes(x = x, y = y, color = class, linetype = class),
            size = 1.2) +
  ylab(expression(f(lambda))) +
  xlab(expression(lambda)) +
  theme_light() +
  theme(axis.text  = element_text(size=15),
        axis.title = element_text(size=18),
        legend.text = element_text(size=15)) +
  scale_linetype_manual(values=c("solid", "twodash"))+
  scale_color_brewer(palette="Dark2") +
  theme(legend.position = "bottom", legend.title=element_blank())
```



Ya hemos definido uno de los elementos característicos de la estadística bayesiana, aquel que nos permite incorporar información previa sobre los parámetros en el modelo, la distribución a priori. Veamos cómo se calcula la distribución a posteriori.

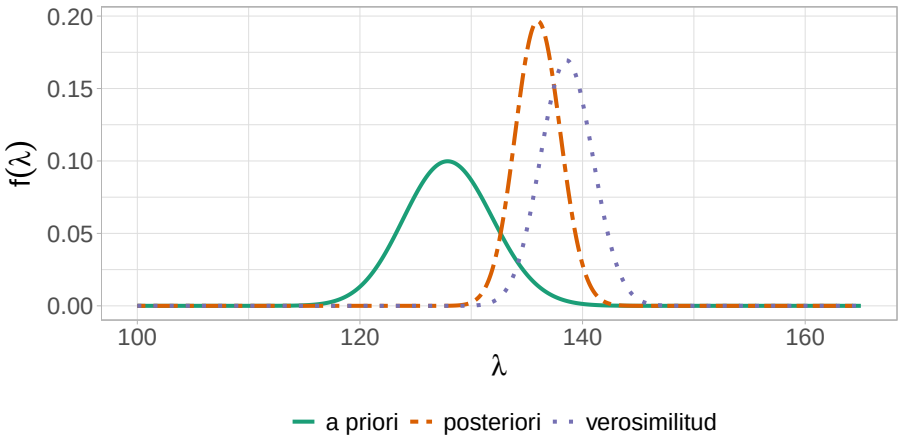
2.3. Distribución a posteriori: gamma

Como estamos haciendo un análisis donde la distribución a priori es conjugada, la distribución a posteriori será de la misma naturaleza que la distribución a priori, es decir, será una distribución **gamma** con parámetros $a + r$ y $b + N$.

Gráficamente, la podemos representar como:

```
aP <- a0 + r
bP <- b0 + length(y)

data_posterior <- data.frame(
  x = aux,
  y = dgamma(aux, shape = aP, rate = bP ),
  class = "posteriori")

data_plot <- rbind(data_vero, data_prior, data_posterior)
ggplot(data = data_plot) +
  geom_line(aes(x = x, y = y, color = class, linetype = class),
            size = 1.2) +
  ylab(expression(f(lambda))) +
```

```
  xlab(expression(lambda)) +
  theme_light() +
  theme(axis.text  = element_text(size=15),
        axis.title = element_text(size=18),
        legend.text = element_text(size=15)) +
  scale_linetype_manual(values=c("solid", "twodash", "dotted"))+
  scale_color_brewer(palette="Dark2") +
  theme(legend.position = "bottom", legend.title=element_blank())
```



Podemos observar que la distribución a posteriori se encuentra localizada entre la distribución a priori y la verosimilitud. Si tuviéramos más datos, la distribución a posteriori se acercaría más a la verosimilitud, y si utilizásemos una distribución a priori más restrictiva, esta distribución a posteriori se acercaría más a la distribución a priori.

Una vez tenemos dibujada la distribución a posteriori podemos calcular su media:

```
aP/bP
```

```
## [1] 136
```

o un intervalo de credibilidad al 95 %.

```
qgamma(0.025, aP, bP)
```

```
## [1] 132.0499
```

```r
qgamma(0.975, aP, bP)
```

```
## [1] 140.0075
```

Hasta ahora hemos hecho inferencia sobre el parámetro del modelo, λ. Ahora bien, nos podríamos preguntar, si escogemos al azar un adolescente, ¿cuántos mensajes de WhatsApp recibirá en un día no lectivo? Para ello, hemos de obtener la distribución predictiva.

2.4. Distribución predictiva: gamma-Poisson

Otras distribuciones importantes en el contexto bayesiano, como ya sabemos, son las distribuciones predictivas a priori y a posteriori, $p(y)$ y $p(y \mid D)$. La distribución predictiva a priori se calcula integrando sobre las distribuciones a priori de los parámetros:

$$p(Y = y) = \int_{\Theta} p(y \mid \lambda) p(\lambda) d\lambda,$$

mientras que la distribución predictiva a posteriori se calcula integrando sobre las distribuciones a posteriori de los parámetros:

$$p(Y = y \mid D) = \int_{\Theta} p(y \mid \lambda) p(\lambda \mid D) d\lambda.$$

Para el caso particular de gamma-Poisson, la distribución predictiva a posteriori del total de conteos en n nuevas observaciones (r) es una distribución conocida como **gamma-Poisson**.

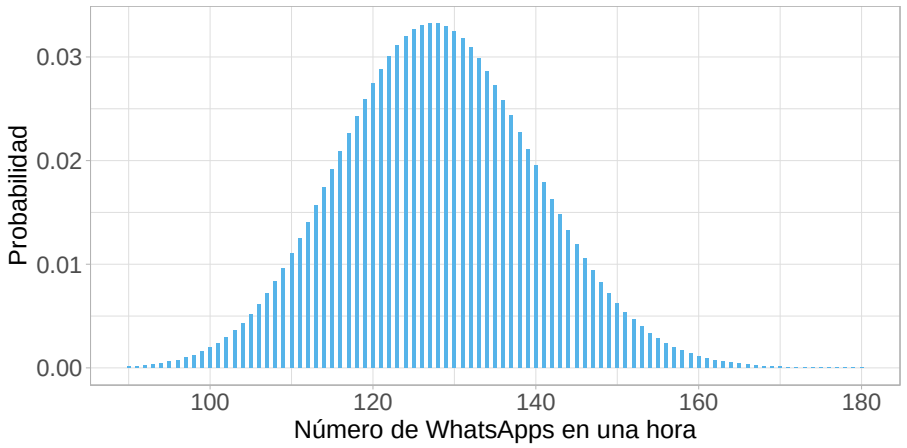
La gamma-Poisson es un distribución de probabilidad para una variable discreta r con soporte y los números naturales, incluyendo el 0. Además, depende de tres parámetros: a, b y N. Su función de probabilidad es:

$$p(r \mid a, b, N) = \frac{b^a}{\Gamma(a)} \frac{\Gamma(a + r)}{r!} \frac{N^r}{(b + N)^{a+r}}.$$

En el caso en que estemos interesados en la obtención de la distribución predictiva a priori, los parámetros de la distribución gamma-Poisson vendrán dados por los parámetros de la distribución gamma a priori. Si, por el contrario, estamos interesados en la obtención de la distribución predictiva a posteriori, los parámetros de la distribución gamma-Poisson vendrán dados por la distribución gamma a posteriori.

Así, la forma de la distribución predictiva a priori del número de mensajes de WhatsApp que un adolescente recibe en una hora queda como sigue:

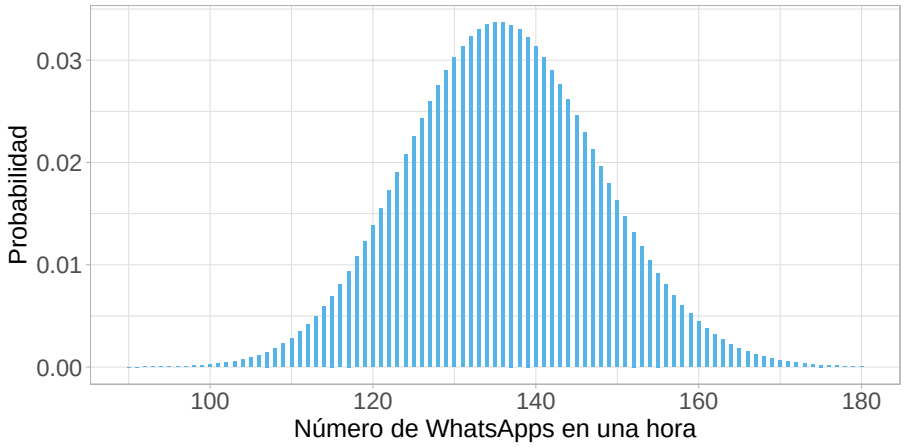
```r
dgammapois <- function(r, a, b, n){
  C <- a * log(b) - lgamma(a)
  C2 <- lgamma(a + r) + log(n) * r - lgamma(r + 1) - log(b + n) * (a + r)
  exp(C+C2)
}


ggplot(data = data.frame(
  x = 90:180,
  y = dgammapois(90:180, a = a0, b = b0, n = 1)),
        aes(x = x, y = y)) +
  geom_bar(stat="identity", fill = "#56B4E9", width = 0.4) +
  theme_light() +
  xlab("Número de WhatsApps en una hora") +
  ylab("Probabilidad") +
    theme(axis.text  = element_text(size=15),
        axis.title = element_text(size=16),
        legend.text = element_text(size=15))
```



Mientras que la distribución predictiva a posteriori es:

```r
ggplot(data = data.frame(
  x = 90:180,
  y = dgammapois(90:180, a = aP, b = bP, n = 1)),
        aes(x = x, y = y)) +
  geom_bar(stat="identity", fill = "#56B4E9", width = 0.4) +
```

```
  theme_light() +
  xlab("Número de WhatsApps en una hora") +
  ylab("Probabilidad") +
    theme(axis.text  = element_text(size=15),
        axis.title = element_text(size=16),
        legend.text = element_text(size=15))
```



Podemos observar que la distribución predictiva se calcula integrando para todos los posibles valores de λ. Por este motivo es diferente al hecho de utilizar una distribución de Poisson usando $E(\lambda)$ o $E(\lambda \mid \mathbf{y})$. En el siguiente gráfico podemos ver las diferencias.

```
data_pred_prior <- data.frame(
  x = seq(90, 180, 2),
  y = dgammapois(seq(90, 180, 2), a = a0, b = b0, n = 1),
  class = "pred_prior")
data_pred_post <- data.frame(
  x = seq(90, 180, 2),
  y = dgammapois(seq(90, 180, 2), a = aP, b = bP, n = 1),
  class = "pred_post")
data_pred_pois <- data.frame(
  x = seq(90, 180, 2),
  y = dpois(seq(90, 180, 2), aP / bP),
  class = "poisson")
data_pred_plot <- rbind(data_pred_prior, data_pred_post, data_pred_pois)
```

```r
ggplot(data = data_pred_plot, aes(x = x, y = y,  fill = class)) +
   geom_bar(stat="identity", position=position_dodge()) +
  xlab("Número de WhatsApps en una hora") +
  ylab("Probabilidad") +
 theme_light() +
 theme(axis.text  = element_text(size=15),
       axis.title = element_text(size=16),
       legend.text = element_text(size=15)) +
 scale_fill_brewer(palette="Dark2") +
 theme(legend.position = "bottom", legend.title=element_blank())
```