

Guía 4: Acumulación de evidencia (1/2)

Toma de decisiones 2020

Guillermo Solovey

A diferencia de los modelos basados en la teoría de detección de señales, los modelos que acumulación de evidencia tienen en cuenta el tiempo necesario para decidir. Son modelos que buscan responder a la pregunta de cómo acumulamos evidencia a favor o en contra de una opción y cómo decidimos en función de la evidencia acumulada. Los modelos reflejan la actividad neuronal en áreas del cerebro como la corteza lateral intraparietal (LIP), entre otras.

Hay muchas manera de implementar estos modelos. La forma más sencilla, y la que vamos a usar en esta guía, es haciendo una simulación en la que discretizamos el tiempo y en cada paso temporal acumulamos evidencia. Es un modelo de *random walk* (caminata aleatoria) con *drift*, que tiene aplicaciones en área muy diversas de la física, biología, economía, etc.

Ejercicio 1

Simular un *trial* usando el modelo de caminata al azar con los siguientes parámetros:

- número máximo de pasos de tiempo: 200
- drift = 0
- umbral = 10
- evidencia inicial (no sesgada)

Luego hacer un gráfico de la trayectoria de la evidencia.

```
n_pasos <- 100
drift   <- 0.0
umbral  <- 10
x_ini   <- umbral/2

x       <- ___
x[1]    <- x_ini
i       <- 1
while( i < n_pasos & ___ ){
  x[i+1] <- x[i] + ___
  i      <- i + 1
}

x[i] <- ifelse( x[i] > umbral, ___, 0)

plot(x, type = 'l', xlim = c(1,n_pasos), ylim = c(-1,umbral+1), lwd=3, col = "black")
abline(umbral, 0, lwd=3)
abline(0, 0, lwd=3)
```

Ejercicio 2

Crear una función que simule $n = 1000$ *trials* y devuelva el tiempo de respuesta y la opción elegida en cada *trial*.

```
rw_decisiones <- function(n_pasos=1000, x_ini=0, drift=0, sd_rw=1, umbral=10){
  x    <- x_ini + cumsum( ___ )
  rt   <- which( ___ )[1]
  resp <- sign( ___ )
  out  <- c(rt, resp)
  return(out)
}
```

Ejercicio 3

El tiempo de respuesta, en la práctica, no es igual al tiempo de acumulación de evidencia por varios motivos (por ejemplo, una vez que se acumuló suficiente evidencia, la acción motora de reportar la decisión lleva un tiempo). Para tener en cuenta eso, modificar la función del ejercicio anterior llamando “tiempo de decisión” (*td*) al tiempo durante el cual se acumula evidencia y “tiempo de respuesta” (*rt*) a $rt = td + t_0$, donde t_0 es un parámetro que no varía entre *trials* y que podemos fijar en 50 pasos de tiempo.

Ejercicio 4

Usar la función del ejercicio 3 para construir la probabilidad de elegir la opción correcta en función de la dificultad de la tarea. Considerar que el *drift* es proporcional a la dificultad. Usar 5 niveles de dificultad diferentes dentro de los 1000 *trials*.

```
rw_decisiones <- function(n_pasos=1000, x_ini=0, drift=0, sd_rw=1, umbral=10){
  x    <- x_ini + cumsum( ___ )
  rt   <- which( ___ )[1]
  resp <- sign( ___ )
  out  <- c(rt, resp)
  return(out)
}

N      <- 1000
drift  <- sample(seq(____, ____, by = ____), N, replace=TRUE)
rw     <- data.frame(drift=____, rt=____, resp=____)
for (i in 1:N){
  out <- rw_decisiones(n_pasos=____, drift = ____, sd_rw = ____, umbral = ____ )
  rw$rt[i] <- out[1]
  rw$resp[i] <- out[2]
}

library(dplyr)
p <- rw %>%
  group_by(drift) %>%
  summarize(pc = mean(resp==1))

plot(p$drift, p$pc, xlab = "drift", ylab = "% resp 1", ylim=c(0,1))
```

Ejercicio 5

Usar la funcion del ejercicio 3 para construir histogramas de los tiempos de respuesta para distintos valores de dificultad.

```
rw_decisiones <- function(n_pasos=1000, x_ini=0, drift=0, sd_rw=1, umbral=10){
  x    <- x_ini + cumsum( ___ )
  rt   <- which( ___ )[1]
  resp <- sign( ___ )
  out  <- c(rt, resp)
  return(out)
}

N      <- 1000
drift  <- sample(seq(____, ___, by = ___), N, replace=TRUE)
rw     <- data.frame(drift=___, rt=___, resp=___)
for (i in 1:N){
  out <- rw_decisiones(n_pasos=___, drift, sd_rw = ___, umbral = ___)
  rw$rt[i] <- out[1]
  rw$resp[i] <- out[2]
}

# para cada nivel de dificultad, hago un histograma de los tiempos de respuesta.
d1 <- filter(rw, ___)
plot(density(d1$rt), xlim=c(0,1200), ylim=c(0,0.01), col=rgb(1,0,0), lwd=3,
     main='distribuciones de RT', xlab="RT (s)")

d2 <- filter(rw, ___)
lines(density(d2$rt), xlim=c(0,1200), ylim=c(0,0.01), col=rgb(0.8,0,0), lwd=3)

d3 <- filter(rw, ___)
lines(density(d3$rt), xlim=c(0,1200), ylim=c(0,0.01), col=rgb(0.6,0,0), lwd=3)

d4 <- filter(rw, ___)
lines(density(d4$rt), xlim=c(0,1200), ylim=c(0,0.01), col=rgb(0.4,0,0), lwd=3)

d5 <- filter(rw, ___)
lines(density(d5$rt), xlim=c(0,1200), ylim=c(0,0.01), col=rgb(0.2,0,0), lwd=3)

d6 <- filter(rw, ___)
lines(density(d6$rt), xlim=c(0,1200), ylim=c(0,0.01), col=rgb(0.1,0,0), lwd=3)

legend(3, 1.5, legend = c('1','2','3','4','5'), lwd=3)
```