EST-46115: Modelación Bayesiana

Profesor: Alfredo Garbuno Iñigo — Primavera, 2022 — Bandits.
Objetivo: Con este tema nos encontramos en la parte final del curso. Veremos aplicaciones del estado del arte y temas de inferencia aproximada. En particular en esta sección estudiaremos pruebas A/B en el contexto Bayesiano.
Lectura recomendada: El artículo [8]. El artículo [2]. También encontrarán este mini-curso relevante en la discusión de este tema. También el libro [3] provee de un tratamiento general de este tema.

1. INTRODUCCIÓN

En esta sección consideraremos el problema de `ofrecer versiones distintas` de un producto y buscar cuál es la versión que `mejora ciertas métricas de interés`. Para esto se recolectan datos al respecto y después se hace inferencia estadística. En particular, nos concentraremos en experimentación secuencial que permite al diseñador decidir qué alternativa se presenta en cada iteración.

El objetivo con este mecanismo es diseñar la política que minimice la cantidad de datos necesarios para poder hacer la inferencia.

Exploraremos la noción de los traga-monedas con brazos múltiples (*multi-armed bandits*). Cada alternativa otorga una recompensa estocástica y el objetivo es minimizar el tiempo necesario para explorar las alternativas para encontrar la mejor.

El problema de los traga-monedas es un problema clásico en la literatura de aprendizaje por refuerzo con un problema de decisión secuencial y tiene aplicaciones en en toma de decisiones para distintos ámbitos como diseño de páginas web, sistemas de recomendación personalizados, etc.

En particular nos concentraremos en explorar la política de muestreo Thompson, donde cada brazo del traga-monedas se escoge con probabilidad proporcional a la probabilidad posterior de obtener la máxima recompensa.

La agenda será:

1. Experimentación Bernoulli.
2. Experimentación secuencial.
3. Experimentación contextual.

2. PRUEBAS A/B ESTÁTICAS

Supongamos que estamos hambrientos y nos encontramos en un pueblito con 3 restaurantes. El primero ha recibido 2/2 reseñas positivas; el segundo, 9/10 y el tercero, 32/40. ¿Cuál escogerías?

Supongamos que administramos un sitio *web* y tenemos que escoger que *Ads* tenemos que poner en la página. Cada *click* nos da dinero de los patrocinadores. Supongamos que tenemos tres opciones donde los *Ads* han recibido 2/2 clicks, 9/10 y 32/40 clicks. ¿Cuál nos conviene?

En ambos problemas el conjunto de datos es el mismo. La pregunta es la misma, ¿qué opción tiene mayor probabilidad de éxito?

En general, los resultados no tienen que ser binarios. Estos pueden ser conteos, continuos o multivariados. Pero la pregunta es la misma, ¿cuál es la mejor opción?

2.1. Pruebas A/B Bernoulli

Mantengamos por el momento el escenario binario con éxito o fracaso. Asumiremos que tenemos K objetos para comparar. Para cada $k \in K$, hemos realizado N_k pruebas de las cuales y_k han sido éxitos.

El modelo tiene parámetros $\theta_k \in (0,1)$ para cada una de las posibilidades, que representa las probabilidades de éxito. Nuestro objetivo será definir cuál es la mejor opción. Esto es, qué opción tiene la mejor posibilidad de éxito.

De momento asumiremos previas

$$\theta_k \sim \mathsf{Uniforme}(0,1)\,. \tag{1}$$

El modelo de verosimilitud es

$$y_k \sim \mathsf{Binomial}(N_k, \theta_k)\,. \tag{2}$$

Por último, queremos saber cuál es la mejor opción. Esto lo podemos escribir como

$$\begin{aligned}\mathbb{P}[\text{ la mejor opción es } k|y] &= \mathbb{E}[I[\theta_k \geq \max\theta]|y] \\ &= \int I[\theta_k \geq \max\theta]\,\pi(\theta|y)\,\mathrm{d}\theta \\ &= \frac{1}{M}\sum_{m=1}^{M} I[\theta_k^{(m)} \geq \max\theta^{(m)}]\,,\end{aligned}$$

donde $\theta^{(m)} \sim \pi(\theta|y)$ para $m = 1,\ldots,M$.

El código del modelo queda como sigue:

```
data {
  int<lower=1> K;
  array[K] int<lower=0> N;
  array[K] int<lower=0> y;
}
parameters {
  vector<lower=0, upper=1>[K] theta;
}
model {
  y ~ binomial(N, theta);
}
generated quantities {
  array[K] int<lower=0, upper=1> mejor_ix;
  {
    real max_prob = max(theta);
    for (k in 1:K) {
      mejor_ix[k] = theta[k] ≥ max_prob;
    }
  }
}
```

```
data.list ← list(K = 3, y = c(2, 9, 32), N = c(2, 10, 40))
posterior ← modelo$sample(data = data.list, refresh = 1000)
```

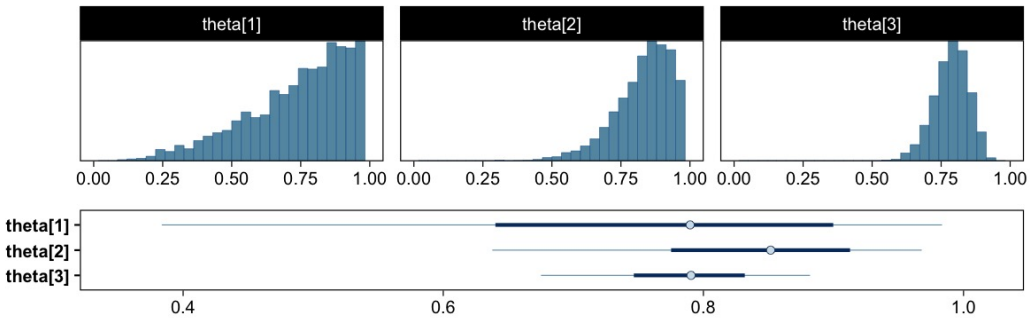

FIGURA 1. *Resúmenes gráficos de la distribución posterior con los datos de los restaurantes.*

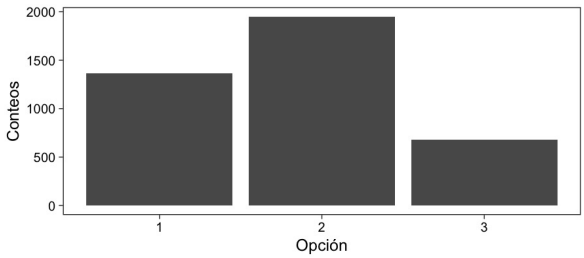

FIGURA 2. *Registro de opciones ganadoras bajo la distribución posterior.*

2.2. Diseño de experimentos

Necesitamos datos para poder determinar cuál es la mejor opción. Para esto se debe de diseñar un experimento para poder determinar el tamaño de muestra necesario para determinar diferencias significativas en las alternativas posibles.

El diseño asume que cada opción es `intercambiable` y por lo tanto se trata sobre establecer cuántas veces se tienen que probar cada opción.

> Intercambiabilidad en las opciones no quiere decir que éstas sean idénticas. Lo que quiere decir es que *a priori* las opciones no son identificables.

3. TRAGAMONEDAS CON BRAZOS MÚLTIPLES

En este escenario tomamos una colección de opciones. Suponemos que cada opción tiene recompensas iid. Esto implica que cada opción siempre tendrá la misma probabilidad para las recompensas, independiente del número de veces que se juegue cada opción. Cada vez que se utiliza una de las opciones tenemos un evento independiente.

3.1. Exploración y explotación

En este contexto hablamos que tenemos que *explorar* la distribución de recompensas de cada una de las opciones y que tendremos que *explotar* nuestro conocimiento sobre la opción que genera mejores retornos.

Llamamos `política` a la forma en que exploramos las posibilidades. Nota que las elecciones no tienen que ser deterministas.

3.2. Diseños secuenciales

Podemos considerar una política que vaya cambiando la forma en que se van escogiendo las opciones. Esto es, ajustar la forma en que escogemos las opciones considerando los resultados previos que hemos observado.

3.3. Pérdidas

Las políticas usualmente se comparan considerando la pérdida esperada. Es decir, el valor esperado de la diferencia de las recompensas entre escoger siempre la mejor opción contra la que escogimos nosotros.

4. TRAGAMONEDAS BERNOULLI

Consideremos que hay K brazos en el tragamonedas y consideremos que tenemos N iteraciones del proceso. En este caso, consideramos $n \in \{1, \ldots, N\}$ donde hemos escogido el brazo $z_n \in \{1, \ldots, K\}$ y también hemos recibido una recompensa $y_n \in \mathbb{R}$.

El supuesto mas fuerte que hacemos es que cada opción tiene la misma distribución de recompensas. Independiente del número de veces que se ha utilizado o de la historia que hemos observado.

Asumimos que las recompensas tienen distribución

$$y_n \sim \mathsf{Bernoulli}(\theta_{[z_n]})\,. \tag{3}$$

5. POLÍTICAS

Un tomador de decisiones está definido en términos de la estrategia que seguirá para escoger las opciones basado en lo que ha observado en sus decisiones pasadas. Para ser efectivo, se tendrá que balancear entre explorar y explotar las opciones. Matemáticamente consideramos políticas estocásticas por medio de distribuciones

$$\pi(z_{n+1}|y_{1:n}, z_{1:n})\,. \tag{4}$$

5.1. Tipos de políticas

1. Políticas Markovianas, $\pi(z_{n+1}|y_n, z_n)$.
2. Políticas sin memoria, $\pi(z_{n+1})$.
3. Política determinista, $z_{n+1} = f(y_{1:n}, z_{1:n})$.

5.1.1. Política Round Robin*:* Tomar la política como decisiones en secuencia

$$z = 1, 2, \ldots, K, 1, 2, \ldots, K, 1, 2, \ldots, K, \ldots, \tag{5}$$

preserva la idea de que cada opción se tomará de manera uniforme con la misma proporción.

5.1.2. Política uniforme: Se tomará cada opción con una probabilidad equiprobable

$$\pi(z_{n+1}|y_{1:n}, z_{1:n}) = \mathsf{Categorical}\left(\frac{1}{K}, \ldots, \frac{1}{K}\right)\,. \tag{6}$$

5.1.3. Política toma y daca: Se escoge una opción hasta que deja de dar recompensas, después, se cambia a la siguiente opción. Se empieza con la opción $z_n = 1$ y después se escogen las opciones de acuerdo a

$$z_{n+1} = \begin{cases} z_n & \text{si} \\ z_n + 1 & \text{si } y_n = 0 \text{ y } z_n < K \\ 1 & \text{si } y_n = 0 \text{ y } z_n = K \end{cases}. \tag{7}$$

5.2. Política Bayesiana

Thompson [9] introdujo una política que incorpora la historia de las recompensas. Cada opción se escoge de acuerdo a la probabilidad de ser la mejor hasta el momento. Dados los parámetros $\theta = (\theta_1, \ldots, \theta_K)$, se considera que la opción k es la mejor si $\theta_k = \max \theta$.

Las opciones se escogen de acuerdo

$$z_n \sim \mathsf{Categorical}(\phi_n)\,, \tag{8}$$

donde $\sum \phi_{n,k} = 1$.

De acuerdo a los supuesto de recompensas Bernoulli y el supuesto de intercambiabilidad escogemos una previa

$$\theta_k \sim \mathsf{Beta}(\alpha, \alpha)\,. \tag{9}$$

Dado el modelo Bayesiano podemos escribir

$$\begin{aligned}
\phi_{k,n} &= \mathbb{P}[\theta_k = \max \theta | y_{1:n}, z_{1:n}] \\
&= \mathbb{E}[I[\theta_k \geq \max \theta] | y_{1:n}, z_{1:n}] \\
&= \int I[\theta_k \geq \max \theta]\, \pi(\theta | y_{1:n}, z_{1:n})\, \mathrm{d}\theta \\
&= \frac{1}{M} \sum_{m=1}^{M} I[\theta_k^{(m)} \geq \max \theta^{(m)}]\,,
\end{aligned}$$

donde $\theta^{(m)} \sim \pi(\theta | y_{1:n}, z_{1:n})$ para $m = 1, \ldots, M$.

6. TRAGAMONEDAS BERNOULLI EN STAN

El modelo lo implementamos como sigue

```
data {
  int<lower=1> K;
  int<lower=0> N;
  array[N] int<lower=1, upper=K> z;
  array[N] int<lower=0, upper=1> y;
}
parameters {
  vector<lower=0, upper=1>[K] theta;
}
model {
  theta ~ beta(1, 1);
  y ~ bernoulli(theta[z]);
}
generated quantities {
  simplex[K] mejor_ix;
  {
    real mejor_prob = max(theta);
    for (k in 1 : K) {
      mejor_ix[k] = theta[k] ≥ mejor_prob;
    }
    mejor_ix /= sum(mejor_ix);
  }
}
```

6.1. Estadísticas suficientes

El código anterior puede ser lento pues los experimentos son Bernoulli. Se puede hacer el código mas eficiente si agrupamos para tener experimentos Binomiales. El agrupado se puede hacer desde `Stan`

```
transformed data {
  int<lower = 0> experimentos[K] = rep_array(0, K);
  int<lower = 0> exitos[K] = rep_array(0, K);
  for (n in 1:N) {
    experimentos[z[n]] += 1;
    exitos[z[n]] += y[n];
  }
}
```

Y utilizaríamos un modelo

```
model {
  theta ~ beta(1, 1);
  exitos ~ binomial(experimentos, theta);
}
```

Así que el código queda

```
data {
  int<lower=1> K;
  int<lower=0> N;
  array[N] int<lower=1, upper=K> z;
  array[N] int<lower=0, upper=1> y;
}
transformed data {
  array[K] int<lower = 0> experimentos = rep_array(0, K);
  array[K] int<lower = 0> exitos = rep_array(0, K);
  for (n in 1:N) {
    experimentos[z[n]] += 1;
    exitos[z[n]] += y[n];
  }
}
generated quantities {
  array[K] real<lower = 0, upper = 1> theta;
  for (k in 1:K)
    theta[k] = beta_rng(1 + exitos[k], 1 + experimentos[k] - exitos[k]);

  simplex[K] mejor_ix;
  {
    real mejor_prob = max(theta);
    for (k in 1 : K) {
      mejor_ix[k] = theta[k] ≥ mejor_prob;
      }
      mejor_ix /= sum(mejor_ix);
    }
  }
```

Lo que va a cambiar con los ejemplos anteriores que hemos visto en el curso es que haremos una actualización Bayesiana secuencial y necesitaremos hacer unos pequeños cambios en la forma que interactuamos con el código.

Hacer inferencia secuencial no es trivial y son sólo estos casos donde podemos explotar ciertas propiedades de nuestros modelos. El área de *Asimilación de datos* ([4, 7]) y los métodos secuenciales Monte Carlo como los filtros de partículas ([1]) son instancias donde se estudian y proponen nuevos algoritmos con buenas propiedades teóricas.

```
## Declaramos el problema
K ← 2
theta ← c(0.05, 0.04)
N ← 5000

## Inicializamos
p_best ← matrix(0, N, K)
r_hat ← matrix(0, N, K)
y ← array(0.0, 0)
z ← array(0.0, 0)
prefix ← function(y, n) array(y, dim = n - 1)

## Hacemos el aprendizaje secuencial
for (n in 1:N) {
  data ← list(K = K, N = n - 1, y = prefix(y, n), z = prefix(z, n))
  posterior ← modelo$sample(data, fixed_param = TRUE,
                            chains = 1, iter_sampling = 1000, refresh = 0)
  p_best[n, ] ← posterior$summary(variables = "mejor_ix")$mean
  r_hat[n, ] ← posterior$summary(variables = "theta")$rhat
  z[n] ← sample(K, 1, replace = TRUE, p_best[n, ])
  y[n] ← rbinom(1, 1, theta[z[n]])
}
```

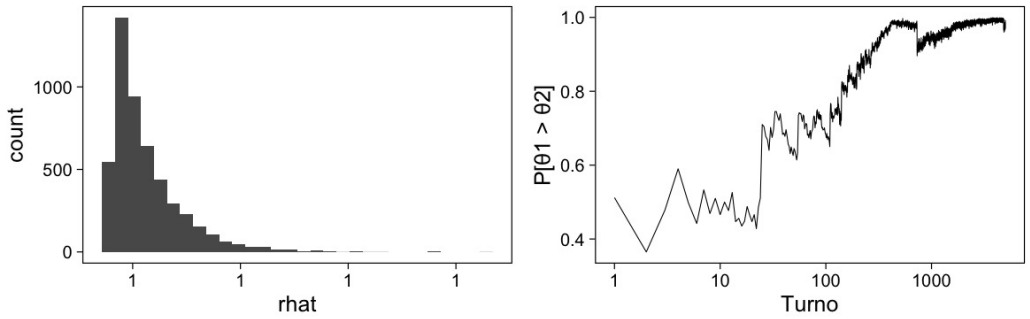

FIGURA 3. *Histogramas del diagnóstico $\hat{R}_n$ y trayectoria de la probabilidad posterior de que $\theta_1 > \theta_2$.*

La figura anterior nos muestra un resultado bastante poderoso. El aprendizaje secuencial Bayesiano sobre la incertidumbre en las tasas de recompensas nos puede ayudar a identificar con alta probabilidad la mejor opción. Un análisis de potencia frecuentista nos diría que necesitamos hasta 10 veces más experimentos para detectar la proporción correcta.

```
power.prop.test(p1 = .05, p2 = .04, power = .95)
```

```

    Two-sample comparison of proportions power calculation

```

```
              n = 11166
             p1 = 0.05
             p2 = 0.04
      sig.level = 0.05
          power = 0.95
    alternative = two.sided

NOTE: n is number in *each* group
```

7. DECISIONES, DECISIONES, ...

En el marco de teoría de la decisión utilizaremos la opción maximice la utilidad esperada. Esto es, nuestra política óptima será aquella que en cada turno n escogerá

$$k_n^* = \arg \max_{k=1,\dots,K} \mathbb{E}[Y_k | y_{1:n}, z_{1:n}] \,, \tag{10}$$

donde

$$\mathbb{E}[Y_k | y_{1:n}, z_{1:n}] = \int y_k \, \pi(y_k | y_{1:n}, z_{1:n}) \, \mathrm{d}y_k \,. \tag{11}$$

8. TRAGAMONEDAS CONTEXTUALES

Se pueden utilizar modelos predictivos para obtener recompensas contextuales. Esto se utiliza en sistemas de recomendación personalizados. Para esto, utilizamos covariables que nos ayuden a modelar de mejor manera

$$\mathbb{E}[Y_k | X_k] \,, \tag{12}$$

donde se pueden utilizar cualquier modelo de regresión generalizada, o modelos basados en *splines*, o modelos BART (ver [5, 6]).

Alternativas –y una breve revisión de literatura– también se pueden encontrar en el artículo [2]. Por último, la sesión de conferencia en tragamonedas de brazos múltiples por parte del equipo de ciencia de datos de Netflix es muy informativa sobre el tema.

REFERENCIAS

[1] P. Del Moral, A. Doucet, and A. Jasra. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, jun 2006. ISSN 1369-7412, 1467-9868. . 7

[2] Q. F. Gronau, K. N. A. Raj, and E.-J. Wagenmakers. Informed Bayesian Inference for the A/B Test. *arXiv:1905.02068 [stat]*, nov 2020. 1, 8

[3] T. Lattimore and C. Szepesvári. *Bandit Algorithms*. Cambridge University Press, jul 2020. ISBN 978-1-108-48682-8. 1

[4] K. J. H. Law, A. M. Stuart, and K. C. Zygalakis. Data Assimilation: A Mathematical Introduction. *arXiv:1506.07825 [math, stat]*, jun 2015. 7

[5] L. Li, W. Chu, J. Langford, and R. E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web - WWW '10*, page 661, Raleigh, North Carolina, USA, 2010. ACM Press. ISBN 978-1-60558-799-8. . 8

[6] O. A. Martin, R. Kumar, and J. Lao. *Bayesian Modeling and Computation in Python*. Chapman and Hall/CRC, Boca Raton, First edition, 2021. 8

[7] S. Reich and C. Cotter. *Probabilistic Forecasting and Bayesian Data Assimilation*. Cambridge University Press, Cambridge, 2015. ISBN 978-1-107-06939-8 978-1-107-66391-6. 7

[8] S. L. Scott. A modern Bayesian look at the multi-armed bandit. *Applied Stochastic Models in Business and Industry*, 26(6):639–658, nov 2010. ISSN 15241904. . 1

[9] W. R. Thompson. On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples. *Biometrika*, 25(3/4):285–294, 1933. ISSN 0006-3444. . 5