Informe_Met-Has

La estadística bayesiana se usa para poder dar respuestas a incertidumbres sobre parámetros que afectan a distintas situaciones. Se usan dos fuentes de información para ello, la primera es lo que uno cree conocer sobre los parámetros previamente al realizar un experimento, es decir, la información a *priori*, y la segunda, los resultados que obtiene al realizarlo, que se lo denomina como la *verosimilitud*. Juntando estas informaciones se obtiene lo que se conoce como la distribución a *posteriori* del o los parámetros. Pero para poder encontrar respuesta al desconocimiento sobre el tema se debe poder sacar muestras de la distribución a *posteriori*. Para realizar esto, una de las técnicas que se pueden utilizar es Metropolis-Hasting.

Metropolis-Hasting

Este método consiste en una cadena de Markov que, a grandes rasgos, recorre los distintos valores de una distribución de probabilidad generando un secuencia de valores posibles x de dicha distribución $x(1), x(2), ..., x(S)$, donde para obtener $x(i+1)$ usamos $x(i)$. De esta manera se quiere que el conjunto de los valores simulados se aproximen lo máximo posible a una muestra real de la distribución.

Esta técnica funciona de la siguiente manera:

1. En la iteración i estamos en el valor $x(i)$
2. En función del valor actual $x(i) = x$, proponemos un nuevo valor x' en función de $q(x' \mid x)$ siendo q la distribución de salto que uno debe proponer, que sera la que presente los puntos de salto posibles.
3. Decidimos si vamos a la nueva ubicación $x(i+1) = x'$ o si nos quedamos en $x(i+1) = x$:

    i. Calcular la probabilidad de salto:
      - $\alpha_{\theta \to \theta'} = min(1, f(x')f(x))$

    ii. Saltar a x' con probabilidad $\alpha_{x \to x'}$:

$$x^{(i+1)} = \begin{cases} x' \text{ con probabilidad } \alpha_{\theta \to \theta'} \\ x \text{ con probabilidad } \alpha_{\theta \to \theta'} \end{cases}$$

La función para utilizar el metodo es la siguiente:

```
# n: Tamaño de la cadena que se quiere generar

# p_inicial: Primer punto con el que se inicial la cadena. Es igual a 0 por defecto.

# v_random: Utilizar si se quiere empezar la cadena con un punto aleatorio. Introducir un vec

# d_objetivo: Función que calcule la densidad de la función que se quiere muestrear.

# r_propuesta: Función que genera un punto de la distribución de salto con la media sin espec

# d_propuesta: Función que devuelve la densidad en un punto de la distribución de salto con l

sample_mh_1d = function(n, p_inicial = 0, d_objetivo, r_propuesta = rnorm, d_propuesta = dnor
  stopifnot(n > 0)
  muestras = numeric(n)
  #Seleccion del primer valor de la cadena
  if (length(v_random) == 2){
    muestras[1] = runif(1,min(v_random),max(v_random))
  } else {
    muestras[1] = p_inicial
  }
  #Generación de todos los n-1 puntos siguientes de la cadena
  for (i in 2:n) {
    #Proposición del nuevo valor
    puntos = valor_salto_1d(muestras, r_propuesta, i)
    p_actual = puntos[1]
    p_nuevo = puntos[2]

    #Calculo de la probabilidad de salto
    prob = salto_1d(p_actual, p_nuevo, d_objetivo, d_propuesta)

    #Saltamos?
    salto = test_salto_1d(prob)

    #Salto
    muestras[i] = p_gen_1d(salto, p_actual, p_nuevo)
  }
    return(muestras)
}

#Donde las funciones utilizadas son:
```

```r
#Proposición del nuevo valor

valor_salto_1d = function(muestras, r_propuesta, i){
  p_actual = muestras[i-1]
  p_nuevo = r_propuesta(p_actual)
  puntos = c(p_actual, p_nuevo)
  return(puntos)
}

#Calculo de la probabilidad de salto

salto_1d = function(p_actual, p_nuevo, d_objetivo, d_propuesta) {
  f_nuevo = d_objetivo(p_nuevo)
  f_actual = d_objetivo(p_actual)
  q_actual = d_propuesta(p_actual, p_nuevo)
  q_nuevo = d_propuesta(p_nuevo, p_actual)
  prob = (f_nuevo*q_actual) / (q_nuevo*f_actual)
  alfa = min(1, prob)
  return(alfa)
}

#Saltamos?

test_salto_1d = function(alfa){
  result = rbinom(1,1,alfa)
  return(result)
}

#Salto

p_gen_1d = function(result, p_actual, p_nuevo) {
  if (result){
    return(p_nuevo)
  } else {
    return(p_actual)
  }
}
```

Aplicación de Metropolis-Hasting en una dimensión

Distribución de Kumaraswamy

La distribución de Kumaraswamy es una distribución de probabilidad continua que se utiliza para modelar variables aleatorias con soporte en el intervalo $(0,1)$. Su función de densidad se parece a la beta pero su expresión matematica resulta ser de computo mas sencillo.

$$p(x \mid a, b) = abx^{a-1}(1-x^a)^{b-1} \quad \text{con } a, b > 0$$

======= »»»> e837f3c241e82f6a9b655351ea0048387b22d44d