Edit or run this notebook

Nonparametric probability estimation

Statistics III – Dr. Arturo Erdely

The random variables $Z_i := \mathbb{1}_{\{X_i \in B\}}$ for $i \in \{1, \dots, n\}$ are i.i.d Bernoulli with unknown parameter $\theta = \mathbb{P}(X \in B)$. Using a $\text{Uniform}(0, 1)$ non-informative prior distribution for θ the posterior distribution is $\text{Beta}(1 + nT_n(B), 1 + n(1 - T_n(B)))$ where $Tn(B) = \frac{1}{n} \sum_{i=1}^{n} Z_i$ is a nonparametric estimation of $\mathbb{P}(X \in B)$. Under a quadratic loss penalization, the bayesian point estimate for $\theta = \mathbb{P}(X \in B)$ is the posterior *expected value* which is the following:

$$\theta^* = \frac{1 + nT_n(B)}{2 + n}$$

An interval estimation for $\theta = \mathbb{P}(X \in B)$ with probability $0 < \gamma < 1$ is an interval $[\theta_1, \theta_2]$ such that $F_\theta(\theta_2) - F_\theta(\theta_1) = \gamma$, where F_θ is the posterior distribution of θ, and such that the interval length $\theta_2 - \theta_1$ is minimum. Since $\theta_2 = F_\theta^{-1}(\gamma + F_\theta(\theta_1))$, where $0 \leq \theta_1 \leq F_\theta^{-1}(1 - \gamma)$, the minimum length interval must be the solution to minimize the following function:

$$h(z) = F_\theta^{-1}(\gamma + F_\theta(z)) - z\,, \qquad 0 \leq z \leq F_\theta^{-1}(1 - \gamma)$$

```
using Distributions
```

Tn (generic function with 1 method)

Edit or run this notebook

```julia
function Tn(interval::String, xobs::Vector{<:Real})
    m = length(interval)
    brackets = ['[', ']']
    if !issubset([interval[1], interval[m]], brackets)
        error("Error: interval must start and end with brackets.")
        return nothing
    end
    icomma = 0
    for i ∈ 1:m
        if interval[i] == ','
            icomma = i
        end
    end
    if icomma == 0
        error("Error: interval |a,b| extremes must be separated by a comma.")
        return nothing
    end
    a = parse(Float64, interval[2:(icomma - 1)])
    b = parse(Float64, interval[(icomma + 1):(m-1)])
    if a > b
        error("Error: interval |a,b| extremes must satisfy a ≤ b.")
        return nothing
    end
    n = length(xobs)
    if interval[1] == ']' && interval[m] == ']'
        tn = count(a .< xobs .≤ b) / n
    end
    if interval[1] == ']' && interval[m] == '['
        tn = count(a .< xobs .< b) / n
    end
    if interval[1] == '[' && interval[m] == ']'
        tn = count(a .≤ xobs .≤ b) / n
    end
    if interval[1] == '[' && interval[m] == '['
        tn = count(a .≤ xobs .< b) / n
    end
    return tn
end
```

Edit or run this notebook

```
EDA (generic function with 1 method)
```

```julia
function EDA(fobj, valmin, valmax; iEnteros = zeros(Int, 0), tamgen = 1000,
             propselec = 0.3, difmax = 0.00001, maxiter = 1000)
    numiter = 1
    println("Iterando... ")
    numvar = length(valmin)
    nselec = Int(round(tamgen * propselec))
    G = zeros(tamgen, numvar)
    Gselec = zeros(nselec, numvar)
    for j ∈ 1:numvar
        G[:, j] = valmin[j] .+ (valmax[j] - valmin[j]) .* rand(tamgen)
    end
    if length(iEnteros) > 0
        for j ∈ iEnteros
            G[:, j] = round.(G[:, j])
        end
    end
    d(x, y) = sqrt(sum((x .- y) .^ 2))
    rnorm(n, μ, σ) = μ .+ (σ .* randn(n))
    promedio(x) = sum(x) / length(x)
    desvest(x) = sqrt(sum((x .- promedio(x)) .^ 2) / (length(x) - 1))
    fG = zeros(tamgen)
    maxGselec = zeros(tamgen)
    minGselec = zeros(tamgen)
    media = zeros(numvar)
    desv = zeros(numvar)
    while numiter < maxiter
        # evaluando función objetivo en generación actual:
        print(numiter, "\r")
        for i ∈ 1:tamgen
            fG[i] = fobj(G[i, :])
        end
        # seleccionando de generación actual:
        umbral = sort(fG)[nselec]
        iSelec = findall(fG .≤ umbral)
        Gselec = G[iSelec, :]
        for j ∈ 1:numvar
            maxGselec[j] = maximum(Gselec[:, j])
            minGselec[j] = minimum(Gselec[:, j])
            media[j] = promedio(Gselec[:, j])
            desv[j] = desvest(Gselec[:, j])
        end
        # salir del ciclo si se cumple criterio de paro:
        if d(minGselec, maxGselec) < difmax
            break
```

```julia
            end
            # y si no se cumple criterio de paro, nueva generación:
            numiter += 1
            for j ∈ 1:numvar
                G[:, j] = rnorm(tamgen, media[j], desv[j])
            end
            if length(iEnteros) > 0
                for j ∈ iEnteros
                    G[:, j] = round.(G[:, j])
                end
            end
        end
        println("...fin")
        fGselec = zeros(nselec)
        for i ∈ 1:length(fGselec)
            fGselec[i] = fobj(Gselec[i, :])
        end
        xopt = Gselec[findmin(fGselec)[2], :]
        if length(iEnteros) > 0
            for j ∈ iEnteros
                xopt[j] = round(xopt[j])
            end
        end
        fxopt = fobj(xopt)
        r = (x = xopt, fx = fxopt, iter = numiter)
        if numiter == maxiter
            println("Aviso: se alcanzó el máximo número de iteraciones = ", maxiter)
        end
        return r
    end
```

```
Bn (generic function with 2 methods)
```

```julia
function Bn(interval::String, obs, γ = 0.95)
    # using: Distributions
    # Dependencies: Tn, EDA
    n = length(obs)
    tn = Tn(interval, obs)
    α, β = 1 + n*tn, 1 + n*(1 - tn) # posterior parameters
    Θ = Beta(α, β) # posterior distribution
    θmedia, θmediana = mean(Θ), median(Θ)
    h(z) = (quantile(Θ, γ + cdf(Θ, z[1])) - z[1]) * Inf^(z[1] > quantile(Θ, 1 - γ))
    sol = EDA(h, [0], [quantile(Θ, 1 - γ)])
    θ₁ = sol[1][1]
    θ₂ = quantile(Θ, γ + cdf(Θ, θ₁))
    estimación = (insesgado = tn, media = θmedia, mediana = θmediana, intervalo =
(θ₁, θ₂))
    return estimación
end
```

Let W be a $\text{Normal}(\mu = -2, \sigma = 3)$ random variable. Then $\mathbb{P}(0 < W < 3) = F_W(3) - F_W(0)$, that is:

```
0.20470218527410822
```

```julia
begin
    W = Normal(-2, 3)
    P = cdf(W, 3) - cdf(W, 0)
end
```

Let's now simulate $1,000$ observations from W and make like we forget its distribution, and calculate a nonparametric point and interval estimations for $\mathbb{P}(0 < W < 3)$.

```
(insesgado = 0.207, media = 0.207585, mediana = 0.20739, intervalo = (0.175351, 0.241233)
```

```julia
begin
    wobs = rand(W, 1_000)
    estimP = Bn("]0,3[", wobs, 0.99)
end
```