Aula 8

Roteiro

- *Rejection Sampling* (amostragem por rejeição)
- Exemplos
- *Importance Sampling*
- Exemplos
- Generalização

Rejection Sampling

- Técnica fundamental para geração de amostras aleatórias
  - usada em Markov Chain Monte Carlo (veremos depois)
- **Ideia:** usar distribuição simples para gerar amostras de outra distribuição mais complicada
- Sejam X e Y duas v.a. com distribuições p_x e q_x , definidas no mesmo domínio
  - p_x = P[X = x], q_x = P[Y = x]
- Assuma que $q_x <= c\ p_x$ para alguma constante c e todo x
  - ou seja, $c\ p_x$ é uma *função envelope* para q_x
- Supor que sabemos gerar amostras para v.a. *X*
- Como gerar amostras para v.a. *Y*?

Rejection Sampling

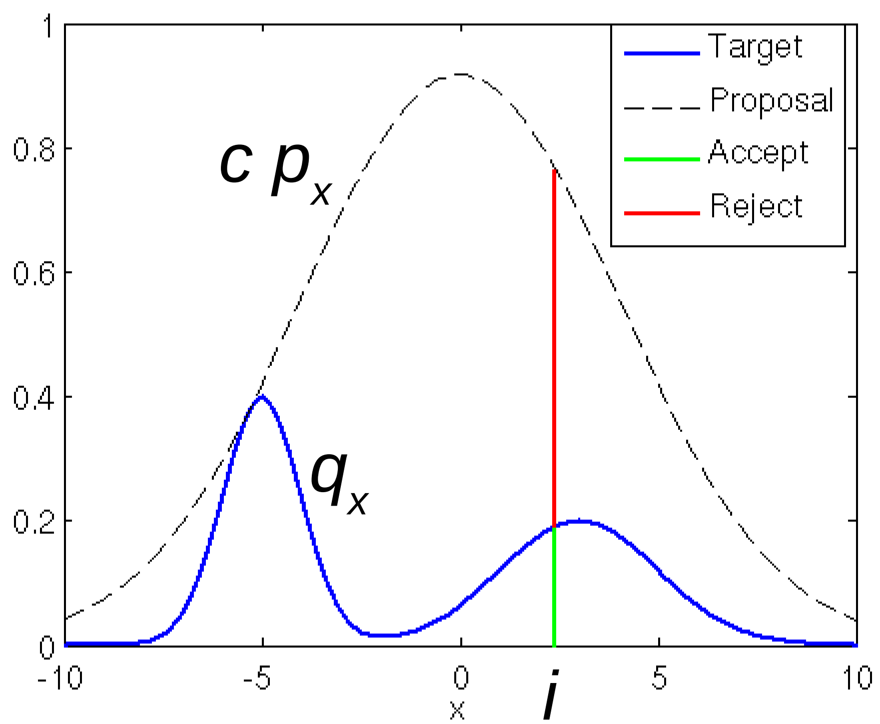
- Algoritmo (proposto por von Neumann)

  1) Gerar valor para i a partir de p_x

  2) Gerar u Uniforme(0, $c\ p_i$) contínua, usando o i gerado

  3) Se $u < q_i$, retorna i, caso contrário vai para 1)

- Graficamente



- Algoritmo pode *rejeitar* amostra de X várias vezes
  - *rejection sampling*
- Aceita com probabilidade dada pela razão entre de q_x e $c\ p_x$

Figueiredo 2021

Rejection Sampling Funciona

- Mostrar que algoritmo gera amostra *i* com probabilidade q_i
  - $q_i = P[Y=i] = P[X=i | aceitar]$
- Considere um valor i e o evento *aceitar*
  - $P[X=i, aceitar] = P[X=i] P[aceitar | X=i] = p_i q_i / (c\, p_i) = q_i / c$
- Probabilidade de aceitar (por prob. Total)
  - $P[aceitar] = \sum_i P[X=i, aceitar] = \sum_i \frac{q_i}{c} = \frac{1}{c}$
- Ao final de cada rodada, algoritmo aceita com prob. 1/c
- Temos então
  - $P[X=i | aceitar] = \frac{P[X=i, aceitar]}{P[aceitar]} = q_i = P[Y=i]$

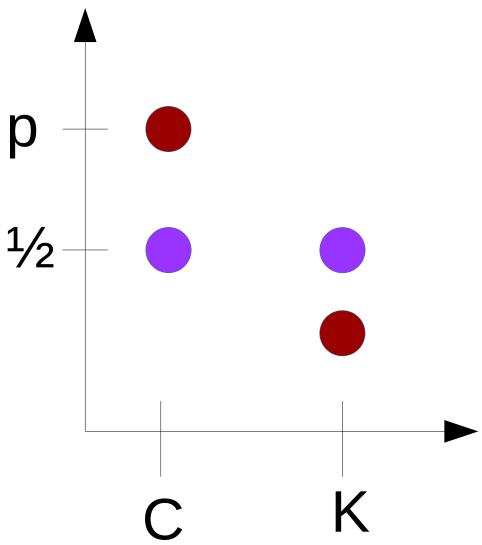
Rodadas de *Rejection Sampling*

- A cada rodada, algoritmo aceita com prob. 1/c
- Número de rodadas até aceitar é aleatório. Distribuição?
  - Geométrica com parâmetro 1/c
  - valor esperado = c
  - complexidade de caso médio para cada amostra
- Escolha do valor para c é muito importante
  - menor valor tal que $q_x \le c\, p_x$ para todo x
- Valor depende da "distância" entre q_x e p_x
  - se muito diferentes, pode demandar c muito grande

Técnica funciona também com v.a. contínua

Exemplo

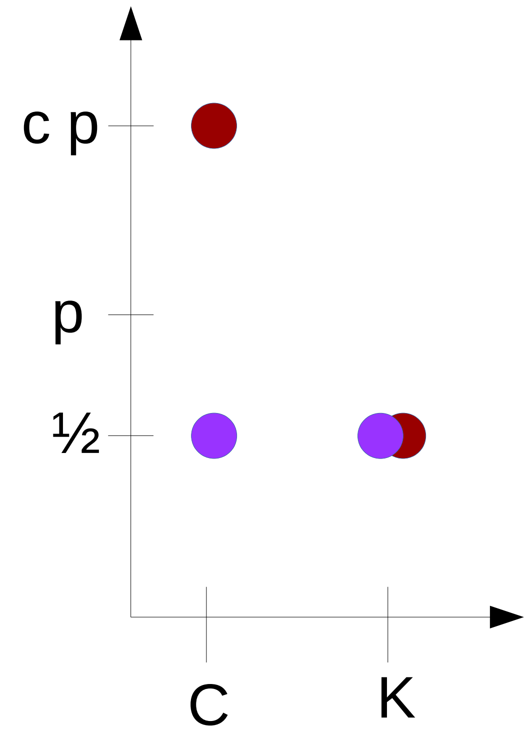
- Dado moeda enviesada, cara com prob $p > ½$
- Como gerar moeda sem viés?



Moeda enviesada

Moeda honesta

- Encontrar constante c tal que $q_x <= c\ p_x$
- $c(1-p) = ½ \rightarrow$ $c = 1/(2(1-p))$



1) Gerar valor para i = {C, K} a partir de p_x *(moeda enviesada)*

2) Gerar u uniforme(0, $c\ p_i$) contínua, usando o i gerado

3) Se $u < q_i$, retorna i, caso contrário vai para 1)

Exemplo 2

- Gerar amostras da v.a. *Y* contínua com densidade $f_Y(x) = 20x(1-x)^3$, 0<=x<=1
- Usando método da rejeição, qual distribuição proponente?
  - uniforme[0,1], $g_X(x) = 1$, 0<=x<=1
- Determinar c tal que f(x) <= c g(x) para 0<=x<=1
- **Ideia:** encontrar máximo de f(x)/g(x)
  - derivar, igualar a zero, resolver para x, encontrar valor
  - máximo em x = ¼, c = 135/64

1) Gerar valor para *x* a partir de *g(x)* (uniforme[0,1])
2) Gerar *u* uniforme(0, *c g(x)*) contínua, usando o *x* gerado
3) Se *u* < *f*(x), retorna *x*, caso contrário vai para 1)

Cenário Problemático

- Algoritmo de Monte Carlo é estimador universal
  - média amostral converge para valor esperado
- **Problema:** variância do estimador pode ser muito alta!
  - muitas, muitas amostras serão necessárias
- Exemplo

$$G_N = \sum_{i=1}^{N} g(i) \qquad M_n = \frac{1}{n} \sum_{i=1}^{n} g(X_i) \text{ , } X_i \sim \text{Unif}(1,N)$$

- *g*(*i*) é desprezível (ou zero) para muitos valores de *i*, e grande para poucos valores
- Teremos que gerar muitas amostras para "acertar" os valores importantes de *g*(*i*)
- Ideias para atacar o problema?

Importance Sampling

- Amostrar com probabilidade diferente da original
  - uniforme no exemplo anterior
- Amostrar com maior probabilidade região mais importante para o problema em questão
  - *importance sampling*
- Compensar pelo aumento desta probabilidade
- Usar método de Monte Carlo em problema reformulado
  - com novas funções de distribuição
- **Objetivo:** reduzir a variância do estimador
  - isto nem sempre ocorre, pois se exagerar demais vai ter que compensar com mais amostras!
  - técnica tem que ser usada com cuidado

Importance Sampling

- Seja *X* uma v.a. uniforme(1,N)

$$E[g(X)]=\sum_{i=1}^{N}P[X=i]g(i)=\frac{1}{N}\sum_{i=1}^{N}g(i)=\frac{G_N}{N}$$

- Seja *Y* v.a. com distribuição dada por h(i) > 0 para todo i

$$E_h\left[\frac{g(Y)}{h(Y)}\right]=\sum_{i=1}^{N}\frac{g(i)}{h(i)}h(i)=\sum_{i=1}^{N}g(i)=G_N$$

- Podemos estimar G_N estimando o valor esperado através da média amostral
  - Monte Carlo aplicado a outro valor esperado
- Usar *h* para reduzir variância!

Importance Sampling

- Seja Y_i uma sequência iid de v.a. com distribuiição $h(i)$

$$M_n = \frac{1}{n} \sum_{i=1}^{n} \frac{g(Y_i)}{h(Y_i)}$$

- Média e variância do novo estimador

$$E_h[M_n] = E_h\left[\frac{1}{n} \sum_{i=1}^{n} \frac{g(Y_i)}{h(Y_i)}\right] = \frac{1}{n} \sum_{i=1}^{n} E_h\left[\frac{g(Y_i)}{h(Y_i)}\right] = G_N$$

$$Var_h[M_n] = \frac{\sigma_{g/h}^2}{n}$$

Variância da v.a. g(Y)/h(Y)

- M_n converge para $E_h[\, g(Y) \,/\, h(Y) \,] = G_N$
- Variância do estimador depende da variância de g(Y)/h(Y)

Figueiredo 2021

Variância da Nova v.a.

- Temos g(Y)/h(Y), onde *Y* possui distribuição *h*, com $h_i > 0$ para todo *i*

$$\sigma^2_{g/h} = Var_h\left[\frac{g(Y)}{h(Y)}\right] = E_h\left[\left(\frac{g(Y)}{h(Y)}\right)^2\right] - \underbrace{E_h\left[\frac{g(Y)}{h(Y)}\right]}_{= G_N}{}^2$$

- Segundo momento

$$E_h\left[\left(\frac{g(Y)}{h(Y)}\right)^2\right] = \sum_{i=1}^{N}\left(\frac{g(i)}{h(i)}\right)^2 h(i) = \sum_{i=1}^{N}\frac{g(i)^2}{h(i)}$$

- E_h[g(Y)/h(Y)] não depende de *h*, mas segundo momento depende
  - variância de M_n depende apenas do segundo momento

Exemplo

- Dado *N*, calcular $G_N = \sum_{i=1}^{N} g(i)$ onde $g(i) = i \log i$
- Seja Y_i seq iid de v.a. com distribuição $h(i) > 0$ para todo *i*
- **Opção 1**: $h(i) = 1/N$ para todo i, ou seja, h(i) é uniforme(1,N)
  - o que temos feito até agora
- Segundo momento?

$$\sum_{i=1}^{N} \frac{g(i)^2}{h(i)} = N \sum_{i=1}^{N} g(i)^2 = N \sum_{i=1}^{N} i^2 \log^2 i$$

- Como reduzir segundo momento?
- **Ideia:** escolher h(i) proporcionalmente a g(i)
  - maiores valores tem maior probabilidade (com cuidado)

Exemplo

- **Opção 2**: h(i) = i / K_2 , ou seja linearmente proporcional a *i*
  - onde K_2 = 1+2+...+N = N(N+1)/2
- Segundo momento?

$$\sum_{i=1}^{N} \frac{g(i)^2}{h(i)} = K_2 \sum_{i=1}^{N} \frac{g(i)^2}{i} = K_2 \sum_{i=1}^{N} i \log^2 i$$

- **Opção 3**: h(i) = i^3 / K_3 , ou seja cúbica em *i*
  - onde $K_3 = 1^3+2^3+...+N^3 = N^2(1 + N)^2/4$

$$\sum_{i=1}^{N} \frac{g(i)^2}{h(i)} = K_3 \sum_{i=1}^{N} \frac{g(i)^2}{i^3} = K_3 \sum_{i=1}^{N} \frac{\log^2 i}{i}$$

Exemplo

- Qual é a melhor opção?
- A que tiver menor variância (ou menor segundo momento)
- Com N=1000, vamos calcular!

Opção 1

$$E_h\left[\left(\frac{g(Y)}{h(Y)}\right)^2\right]=1.44\,x\,10^{13}$$

Opção 2

$$E_h\left[\left(\frac{g(Y)}{h(Y)}\right)^2\right]=1.03\,x\,10^{13}$$

Opção 3

$$E_h\left[\left(\frac{g(Y)}{h(Y)}\right)^2\right]=2.75\,x\,10^{13}$$

- Melhor estimador é a **opção 2** (menor variância)
  - menos amostras para um mesmo erro
  - erro menor para um mesmo número de amostras
- Qual seria a melhor h(i) possível?

O Melhor *h*(*i*) Possível

- h(i) que induz variância zero é o melhor possível!

$$\sigma^2_{g/h} = E_h\left[\left(\frac{g(Y)}{h(Y)} - \mu_{g/h}\right)^2\right] = \sum_{i=1}^{N}\left(\frac{g(i)}{h(i)} - \mu_{g/h}\right)^2 h(i)$$

E[g(Y)/h(Y)] = G_N

- Se $\dfrac{g(i)}{h(i)} = \mu_{g/h}$ para todo *i*, então variância é zero!
- A princípio, podemos escolher qualquer *h*(*i*).
  Qual o problema então?
- Precisamos saber $\mu_{g/h}$ que é justamente o que queremos estimar!
- **Ideia**: tentar aproximar esta relação com heurísticas

Figueiredo 2021

Generalização

- Supor que queremos calcular um determinado valor esperado, onde X tem distribuição dada por *f*

$$\mu_f = E_f[g(X)] = \sum_i g(i) f(i)$$

- Podemos aplicar *Importance Sampling* neste problema
  - amostrar mais regiões mais importantes para *g*
- Seja *h* outra distribuição para v.a. X, tal que f(i) > 0 → h(i) > 0

$$\mu_f = \sum_i \frac{g(i) f(i)}{h(i)} h(i) = E_h\left[\frac{g(X) f(X)}{h(X)}\right]$$

- Temos um outro valor esperado E_h que pode ser estimado
  - *X* em E_h tem distribuição *h*
- Podemos reduzir variância escolhendo *h*

Figueiredo 2021

Importance Sampling

- Calcular valor esperado da função *g(X)*, onde *X* tem distribuição dada por *f*

$$\mu_f = E_f[g(X)] = \sum_i^N g(i) f(i) \longleftarrow f(i) = P[X=i]$$

- **Problemas:** *N* é muito grande; difícil gerar amostras de *f*; *g*(*i*) não "combina" com *f*(*i*)
- **Solução:** usar outra distribuição *h* para gerar amostras!
- Seja *h* distribuição para v.a. *X*, tal que *f*(*i*) > 0 → *h*(*i*) > 0

$$\mu_f = \sum_i \frac{g(i) f(i)}{h(i)} h(i) = E_h\left[\frac{g(X) f(X)}{h(X)}\right]$$

- Podemos estimar E_h usando MC para estimar μ_f

Algoritmo

- Monte Carlo para estimar $\mu_f = E_h[g(X)f(X)/h(X)]$:

  S = 0

  para *i* = 1,…,n

  Gerar amostra *x* com distribuição h

  S = S + g(x)f(x)/h(x)

  retorne *S*/*n*

- Algoritmo gera amostras de *h* e não de *f*
- Se *h* for bem escolhida, variância do estimador pode ser bem menor que estimador usando f
  - boa razão para usar IS

Generalização 2

- Tudo vale para o caso de v.a. contínua
  - trocar distribuição por densidade, somatório por integral
- Muitas aplicações no caso contínuo
  - *Monte Carlo Ray Tracing* não funciona sem *Importance Sampling*
  - Espaço de integração é muito grande para uniforme dar bons resultados
  - Muitas heurísticas são usadas nesta aplicação
- **Outro uso:** estimar eventos de baixa probabilidade
  - E[I(evento A)] = p_A , onde A é evento de interesse
  - Mesma ideia, reduzir variância do estimador, aumentar amostragem do evento de interesse