

Métodos Computacionalmente Intensivos

Stuart Coles, Gareth Roberts e Soren Jarner
Orientador: Prof. PhD. Paulo Justiniano Ribeiro Jr.

5 de outubro de 2009

Capítulo 1

- 1.1 Introdução
- Âmbito do curso
- Uma classificação inicial de métodos intensivos aplicada a estatística é:
 1. Computadores para exploração gráfica de dados
 2. Computadores para modelagem de dados
 3. Computadores para inferência.Há obviamente alguma sobreposição destes três, mas neste curso pretendo focar principalmente na terceira das opções acima. Isto é, que terá por objetivo a compreensão de técnicas de computador que requerem algoritmos inovadores para aplicar inferências padrão.

Capítulo 1

- Assim, a compreensão dos princípios subjacentes aos diferentes algoritmos pode frequentemente levar a uma melhor compreensão de inferência, probabilidade e estatística em geral. Em suma, as técnicas não são apenas um meio para um fim. Elas têm o seu próprio valor intrínseco como estatística.
- Todos os exemplos serão manipulados usando funções R simples - a maneira mais eficiente de implementar as diversas técnicas.
- 1.2 Computadores como máquinas de inferência
- 1.3 Referências Stochastic simulation, B. Ripley.
Bootstrap methods and their application, AC Davinson and DV Hinkley
Tools for statistical inference, M. Tanner.
- 1.4 Confirmação

Capítulo 2

- **Simulação 2.1** Introdução

- Neste capítulo, vamos olhar diferentes técnicas de simulação de distribuições e processo estocástico.
- 2.2 Problemas na simulação O papel da simulação é gerar dados que temos (para todos os efeitos) das propriedades estatísticas de algum modelo específico. Este gera duas questões: 1. Como fazê-lo, e 2. Como fazê-lo de forma eficiente.
- Vamos ilustrar a idéia, simplesmente com um bem conhecido exemplo.

Agulha de Buffon

- Nós vamos começar com uma experiência de simulação que é intrinsecamente nada a ver com computador. Talvez a experiência de simulação mais famosa é a agulha de Buffon, concebido para calcular (não muito eficaz) uma estimativa de π .

Há também um número de maneiras que a experiência pode ser melhorada para dar melhores estimativas, que irá destacar o princípio geral do projeto, experimentos simulados para alcançar melhor precisão no sentido de minimizar a variabilidade estatística.

Agulha de Buffon

- A experiência de Buffon consiste em jogar ao acaso agulhas em um grid de retas paralelas. Sabendo a distância d entre elas e o comprimento l da agulha, calcula-se a probabilidade delas se cruzarem e com o valor obtido, estima-se a constante π . Repetimos este experimento n vezes, e contagem R , o número de vezes que a agulha cruza uma linha.
- $l \leq d$
- $\rho = l/d$
- $\phi = 1/\pi$, uma estimativa de ϕ é
- $\hat{\phi}_0 = \hat{p}/2\rho$ onde $\hat{p} = R/n$

Agulha de Buffon

- $\hat{\pi}_0 = 1/\hat{\phi}_0 = 2\rho/\hat{p}$ estimativas de π
- Ao jogar a agulha aleatoriamente pressupõem que $x \sim U(0, d)$ e $\theta \sim U(0, \pi)$ onde:
- $p = Pr$ (agulha intercepta o grid)
- $= 1/\pi \int_0^\pi Pr(\text{agulha intercepta} \mid \theta = \phi) d\phi$
- $= 1/\pi \int_0^\pi (2/d * l/2 \sin \theta) d\theta$
- $2l/\pi d = 2\rho\phi$

Aguilha Buffon

- Uma pergunta natural é saber como otimizar o tamanho relativo de l e d .
- Para resolver isso, precisamos considerar a variabilidade do estimador $\hat{\phi}_0$.
- Agora, $R \text{ sin } Bin(n, p)$, assim $Var(\hat{p}) = p * (1 - p)/n$.
- Deste modo

$$Var(\hat{\phi}_0) = 2\rho\phi(1 - 2\rho\phi)/4\rho^2n = \phi^2(1/2\rho\phi - 1)/n$$
- O que é minimizado (para $\rho \leq 1$) quando $\rho = 1$.
- Ou seja, devemos definir $l = d$ para otimizar a eficiência.
- Então $(\hat{\phi}_0 = p^2)/2$, com $Var(\hat{\phi}_0) = (\phi/2 - \phi^2)/n$.
- R

Agulha Buffon

- Resulta $\text{Var}(\hat{\phi}) \approx \pi^4 \text{Var}(\hat{\phi}_0) \approx 5.63/n$
- A figura 2 mostra 5000 simulações cada.

Agulha de Buffon

- 1. Usando uma grid retângular ou quadrado (o que é melhor?) e número baseando a estimativa sobre o cruzamentos com uma ou ambas as linhas horizontais ou verticais.
- 2. Usando uma cruz em vez de uma agulha.
- 3. Usando uma agulha de tamanho maior do que a separação da rede.
- Assim, apenas para re-iteração, o ponto é que a simulação pode ser usada para responder a problemas interessantes, mas cuidado com a forma pode ser necessário para alcançar até mesmo eficiência.

2.4 Ingredientes Bruto

A matéria-prima para qualquer exercício de simulação é dígitos aleatórios: transformação ou outros tipos de manipulação pode então ser aplicado para criar simulações de distribuições mais complexas ou sistemas. Assim, como pode dígitos aleatórios ser gerados? Deve ser reconhecido que qualquer tentativa de algoritmos para simular a aleatoriedade é apenas : um imitador.

- Dividindo essa seqüência M dá uma seqüência U_i , que são considerados como realizações distribuição Uniforme $U[0, 1]$.

2.4 Ingredientes Bruto

Por definição, se a seqüência gerada é determinista, então não há aleatoriedade. Assim, o truque é a utilização de algoritmos que geram seqüências de números que possam todos os testes de aleatoriedade (a partir da distribuição exigida ou processo), apesar de sua derivação determinista. A técnica mais comum é usar um gerador congruente. Isso gera uma seqüência de inteiros, através do algoritmo. $X_i = aX_{i-1} \pmod{M}$

- Dividindo essa seqüência M dá uma seqüência U_i , que são considerados como realizações distribuição Uniforme $U[0, 1]$.

2.5 Simulando distribuições especiais

Nesta seção, olhar para formas de simular os dados de uma distribuição F univariada especial, com base numa amostra simulada U_1, U_2, \dots, U_n da distribuição $U[0, 1]$.

2.5.1 Inversão Este é o mais simples de todos os procedimentos, e nada mais é do que uma simples aplicação de probabilidade de transformada integral: se $X \sim F$, então $F(X) \sim U[0, 1]$, por isso, a inversão se $U \sim U[0, 1]$, então $F^{-1}(U) \sim F$. Assim, definindo $x_i = F^{-1}(u_i)$, gera uma seqüência de realizações independentes de F . A figura ilustra como isso funciona. (R)

Inversão

Onde a distribuição da função inversa F^{-1} . Por exemplo, para simular a distribuição exponencial, temos $F(x) = 1 - \exp(-\lambda x)$, então $F^{-1}(u) = -\lambda^{-1} \log(1 - u)$. Assim, e definindo e pode simular a partir da distribuição exponencial (exponencial como unidade padrão). Figura 2.3

- Este procedimento funciona bem para distribuições discretas, desde que interpretar o inverso função de distribuição como $F^{-1}(u) = \min x \mid F(x) \geq u$ (2.2)
- O procedimento então simplesmente leva-se a busca através de uma tabela da função de distribuição.
- R

Tabela

Por exemplo, a função de distribuição de Poisson (2)

x	$F(x)$
0	0,1353353
1	0,4060058
2	0,6766764
3	0.8571235
4	0.9473470
5	0.9834364
6	0.9954662
7	0.9989033
8	0.9997626
9	0.9999535
10	0.9999917

Poisson

Então, nós geramos uma seqüência de uniformes padrão u_1, u_2, \dots, u_n e para cada u_i obter uma Poisson variável x onde $F(x - 1) < u_i \leq F(x)$. Assim por exemplo, se $u_1 = 0,7352$ então $x_1 = 3$. Na verdade, existem muitas situações em que a inversão do método seja (ou ambos) complicado para programar ou excessivamente insuficiente para ser executado. A inversão método só é realmente útil se a função de distribuição inversa é fácil de programar e calcular.

Amostra rejeitadas

A idéia de amostragem é simular a rejeição de uma distribuição que seja fácil de simular, mas, em seguida, para apenas aceitar que o valor simulado com alguma probabilidade P . Ao escolher P corretamente, podemos assegurar que a seqüência de valores simulados são aceitas a partir do desejada distribuição. A técnica é ilustrada na Figura 2.4.

- R
- Suponha que desejamos simular uma amostra da Beta (2,2) distribuição. Inversão neste caso exigiria solução (não linear da função de distribuição inversa. Em vez disso, vinculados a função de densidade $f(x)$ por um retângulo e simular pontos (x_i, y_i) uniformemente sobre o retângulo. Em seguida, rejeitar os pontos que não se encontram por $f(x)$. A x (coordenadas dos restantes pontos será uma amostra de $f(x)$. Estes são mostrados na Figura 2.5.

2. Rejeição Amostral

A eficiência do método depende de quantos pontos forem rejeitadas, o que depende, por sua vez sobre a forma como $f(x)$ se assemelha ao retângulo. Para melhorar a eficiência e do processo e permitir a situações em que $f(x)$ pode ser ilimitada, a técnica pode ser modificada a permitir a função delimitadora para tomar qualquer forma $kg(x)$, onde $g(x)$ é a densidade de uma distribuição de que é fácil de simular. Em seguida, o algoritmo assume a forma:

- 1. Simular x^* para $g(x)$
- 2. Simular y^* para $U(0, Kg(x^*))$
- 3. Aceitar x^* se $y \leq f(x^*)$
- 4. Continuar.

2. Algoritmo

A razão pela qual isso funciona é a seguinte: vamos denotar X uma variável aleatória de uma distribuição $g(x)$, tal que $f(x) \leq kg(x) \forall x$ para algum valor K . Deixe $h(x)$ a probabilidade de que x é aceita: $h(x) = f(x)/Kg(x)$.

- $Pr(X \leq x | X \text{ aceito}) = \int_{-\infty}^x h(y)g(y)dy$ e

- $Pr(X \text{ aceito}) = \int_{-\infty}^{\infty} h(y)g(y)dy$ assim

- $Pr(X \leq x | X \text{ aceito}) = \frac{\int_{-\infty}^x h(y)g(y)dy}{\int_{-\infty}^{\infty} h(y)g(y)dy}$

- $= \frac{\int_{-\infty}^x f(y)dy}{\int_{-\infty}^{\infty} f(y)dy}$

- de modo que os valores aceitos realmente tem pdf f . Além disso, $Pr(X \text{ aceito}) \int gh = 1/k$

2.1 Algoritmo

A eficiência do processo depende da qualidade do acordo entre F e o delimitador kg , desde que se um grande valor e de K é necessário, então a probabilidade de aceitação é baixo, de modo que um grande número de simulações são necessários para atingir um tamanho de amostra necessário. Como exemplo, considere a distribuição com densidade: $f(x) \propto x^2 e^{-x}; 0 \leq x \leq 1$. uma distribuição gama truncado. Então, uma vez que $f(x) e^{-x}$ toda a parte, podemos definir $g(x) = \exp(-x)$ e assim simular a partir de uma distribuição exponencial, a rejeição de acordo com o algoritmo acima. Figura a seguir mostra tanto $f(x)$ e $g(x)$. É evidente que neste caso, o exemplo é muito pobre para a rotina é altamente eficiente (embora estatisticamente correto). Aplicando esta ao gerar uma amostra de 100 dados usando o seguinte código: R

Relação de uniformes

Uma adaptação do algoritmo de rejeição, que funciona bem para muitas distribuições é dar raios entre método uniforme. Suponha que a h é uma função não negativa $\int h < \infty$ e deixamos $Ch = (u, v) : 0 \leq u \leq \sqrt{h(v/u)}$. Então, se (U, V) são uniformemente distribuídas ao longo Ch então $X = V/U$ tem pdf $h / \int h$. Assim, para simular a partir de uma densidade proporcional h , simulamos uniformemente sobre a região Ch e obtemos coordenadas. Na prática, Ch pode ser complicado na forma, assim que a prática só que a solução é vinculado com um retângulo (se possível), simular dentro do retângulo (por um par de uniformes), e aplicar rejeição: daí, a razão de uniformes.

Relação de uniformes

- Seja $bigtriangleup_h$ a área de C_h . Em seguida, sobre as variáveis em mudança $(u, v) \rightarrow (u, x)$ onde $x = v/u$
- $\Delta_h = \int \int C_h du dv = \int \int_0^{\sqrt{h(x)}} u du dx = \int \frac{1}{2} h(x) dx$.
- Devido à uniformidade de (U, V) sobre C_h (U, V) tem pdf $1/\Delta_h$ para que a transformação, (U, X) tem pdf u/Δ_h , e a integração de U dá a pdf marginal de X com:
- $\Delta_h^{-1} \int_0^{\sqrt{h(z)}} u du = h(x)/2\Delta_h = h(x) / \int h(x) dx$
- Assim, $V = U$ tem pdf proporcional a h . Novamente, uma propriedade fundamental deste método é que h é só necessária para ser especificado até a proporcionalidade.

2.2. Algoritmo

Como discutido acima, isto só é útil se podemos gerar uniformemente sobre C_h , que é mais provável a ser alcançado por meio da simulação de maneira uniforme dentro de um retângulo $[0; a] * [b_-, b_+]$ que contém C_h (desde que esse retângulo existe). Caso isso aconteça, temos o seguinte algoritmo.

- 1. Simular independência $U \sim U[0, a], V \sim U[b_-, b_+]$
- 2. Se $(U, V) \in C_h$ aceite $X = V/U$ caso contrário repetir.
- 3. Continuar.
- Como exemplo, considere a distribuição de Cauchy com densidade $h(x) \propto \frac{1}{1+x^2}$
- Então $C_h = (u, v) : 0 \leq u \leq \sqrt{h(v/u)} = (u, v) : 0 \leq u, u^2 + v^2 \leq 1$, um semicírculo.
- Assim podemos tomar $[0, a] * [b_-, b_+] = [0, 1] * [-1, 1]$ e obter o algoritmo

2.3. Algoritmo

- 1. Simular independência $U \sim U[0, 1]$, $V \sim U[-1, 1]$
 - 2. Se $u^2 + v^2$ aceite $x = v/u$ caso contrário repetir.
 - 3. Continuar.
 - Assim pode ser implementado com código
 - Histograma 1000 valores simulados (note a escala incomum porque o peso da cauda de Cauchy faz um ou dois valores simulados de ser extremo em relação ao resto dos dados).
 - Um número modificado têm sido proposto para melhorar a eficiência do presente procedimento, que ascendem o escalonado e localizar distribuições antes de aplicar o método.
- R

Algoritmo

Outro método para melhorar a eficiência é por um processo conhecido como pre-teste

Isto se aplica tanto a rejeição e a relação de métodos uniformes. O ponto é que, o método da relação de uniformes, por exemplo, a parte mais lenta do algoritmo pode ser a verificação $(u, v) \in C_h$ ou não. No entanto, pode haver regiões mais simples C_1 e C_2 que tal $C_1 \subset C_h \subset C_2$ de modo que se (u, v) é encontrado para estar dentro C_1 ou fora C_2 então imediatamente saber se ele está dentro C_h ou não.