محاسبات آماری پیشرفته

جلسه دوازدهم: روش‌های نمونه‌گیری $MCMC$

حسین باغیشنی، دانشگاه صنعتی شاهرود

۱۹ آذر ۱۳۹۱

مقدمه

روش‌های نمونه‌گیری مونت کارلوی زنجیر مارکوفی *Markov chain Monte Carlo* ($MCMC$)، یک چارچوب کلی از مجموعه روش‌هایی است که برای انتگرال‌گیری به روش مونت کارلو، توسط متروپولیس و همکاران (۱۹۵۳) و هستینگز (۱۹۷۰) معرفی شدند.

مساله برآورد انتگرال $\int g(x)dx$ به روش مونت کارلو را به یاد بیاورید.

برای این منظور ابتدا مساله انتگرال‌گیری را به یک مساله محاسبه امید ریاضی بر حسب یک تابع چگالی، مانند $f(\cdot)$، تبدیل می‌کردیم. یعنی:

$$\mathcal{I} = \int g(x)dx = \int h(x)f(x)dx = \mathbb{E}_f(h(X)).$$

آنگاه برآورد مونت کارلوی $\mathcal{I}$، برابر میانگین یک نمونه به دست آمده از توزیع f، وقتی که حجم نمونه بزرگ باشد (قانون اعداد بزرگ)، خواهد بود.

بنابراین مساله انتگرال‌گیری به مساله یافتن راهی برای تولید نمونه از چگالی هدف $f(\cdot)$، تبدیل می‌شود.

رهیافت $MCMC$ برای نمونه‌گیری از $f(\cdot)$:

تشکیل یک زنجیر مارکوف با توزیع مانای f: اجرای الگوریتم به تعداد کافی بزرگ، به طوری که زنجیر (به طور تقریبی) به توزیع مانای خود همگرا شود.

در واقع برای محاسبه انتگرال $\mathcal{I}$ لازم نیست از یک نمونه (مستقل) مستقیما تولید شده از توزیع f، استفاده شود. می‌توان، بدون تولید مستقیم از f، یک نمونه (وابسته) $X_1, \ldots, X_n$ را با استفاده از یک زنجیر مارکوف ارگودیک با توزیع مانای f، به دست آورد.

اکنون سوالی که مطرح می‌شود، چگونگی ساخت چنین زنجیری است که روش‌های $MCMC$ برای پاسخ به این سوال ارایه شده‌اند.

تعریف ۷.۱ صفحه ۲۶۸ کتاب رابرت و کسلا (۲۰۰۴)، روش $MCMC$ برای تولید از توزیع f را روش تولید یک زنجیر مارکوف ارگودیک با توزیع مانای f، معرفی می‌کند.

مروری کوتاه بر زنجیرهای مارکوف

یک زنجیر مارکوف $\{X^{(t)}\}$ دنباله‌ای از متغیرهای تصادفی وابسته

$$X^{(۰)}, X^{(۱)}, \ldots, X^{(t)}, \ldots$$

است، به طوری که توزیع احتمال $X^{(t)}$ به شرط مفروض بودن متغیرهای گذشته، فقط به $X^{(t-۱)}$ وابسته است.

به این توزیع احتمال شرطی، هسته انتقال ($Transition\ Kernel$) یا هسته مارکوف K می‌گویند:

$$X^{(t+۱)}|X^{(۰)}, X^{(۱)}, \ldots, X^{(t)} \sim K(X^{(t)}, X^{(t+۱)}).$$

به عنوان مثال، یک زنجیر مارکوف قدم زدن تصادفی ساده به صورت $X^{(t+۱)} = X^{(t)} + \epsilon_t$ قابل نمایش است، که در آن $\epsilon_t \sim N(۰, ۱)$ و مستقل از $X^{(t)}$ است. بنابراین هسته مارکوف متناظر است با یک چگالی $N(X^{(t)}, ۱)$.

ویژگی‌های زنجیرهای مارکوف: تحویل‌ناپذیری

زنجیرهای مارکوف $MCMC$ معمولا از یک ویژگی پایداری قوی برخوردارند. این ویژگی، وجود یک توزیع احتمال ماناست. یعنی توزیع احتمال f وجود دارد، به طوری که اگر $X^{(t)} \sim f$، آنگاه $X^{(t+1)} \sim f$.

در نتیجه، به طور رسمی، هسته و توزیع مانا در معادله زیر صدق می‌کنند:

$$\int_{\chi} K(x, y) f(x) dx = f(y).$$

وجود یک توزیع مانا، شرط اولیه‌ای را که در نظریه زنجیرهای مارکوف معروف به تحویل‌ناپذیری ($Irreducibility$) است، بر K تحمیل می‌کند. که اگر زنجیر دارای این شرط نباشد، زنجیر مفیدی نخواهد بود.

تحویل‌ناپذیری به این معنی است که هسته K اجازه حرکت بر روی تمام وضعیت‌های فضای حالت زنجیر را دارد.

یا به عبارت ساده‌تر، صرفنظر از مقدار اولیه $X^{(0)}$، دنباله $\{X^{(t)}\}$ با احتمال مثبت به هر ناحیه از فضای حالت می‌تواند سر بزند.

ویژگی‌های زنجیر مارکوف: بازگشتی

وجود یک توزیع مانا، نتایج اساسی بر روی رفتار زنجیر $\{X^{(t)}\}$ دارد. یکی از آن نتیجه‌ها این است که اغلب زنجیرهایی که در الگوریتم‌های $MCMC$ دخیل هستند، بازگشتی می‌باشند.

که اگر زنجیر این ویژگی را نداشته باشد، مفید نخواهد بود.

بازگشتی بودن به این معنی است که زنجیر به هر مجموعه دلخواه غیر قابل اغماض، به دفعات نامتناهی باز خواهد گشت.

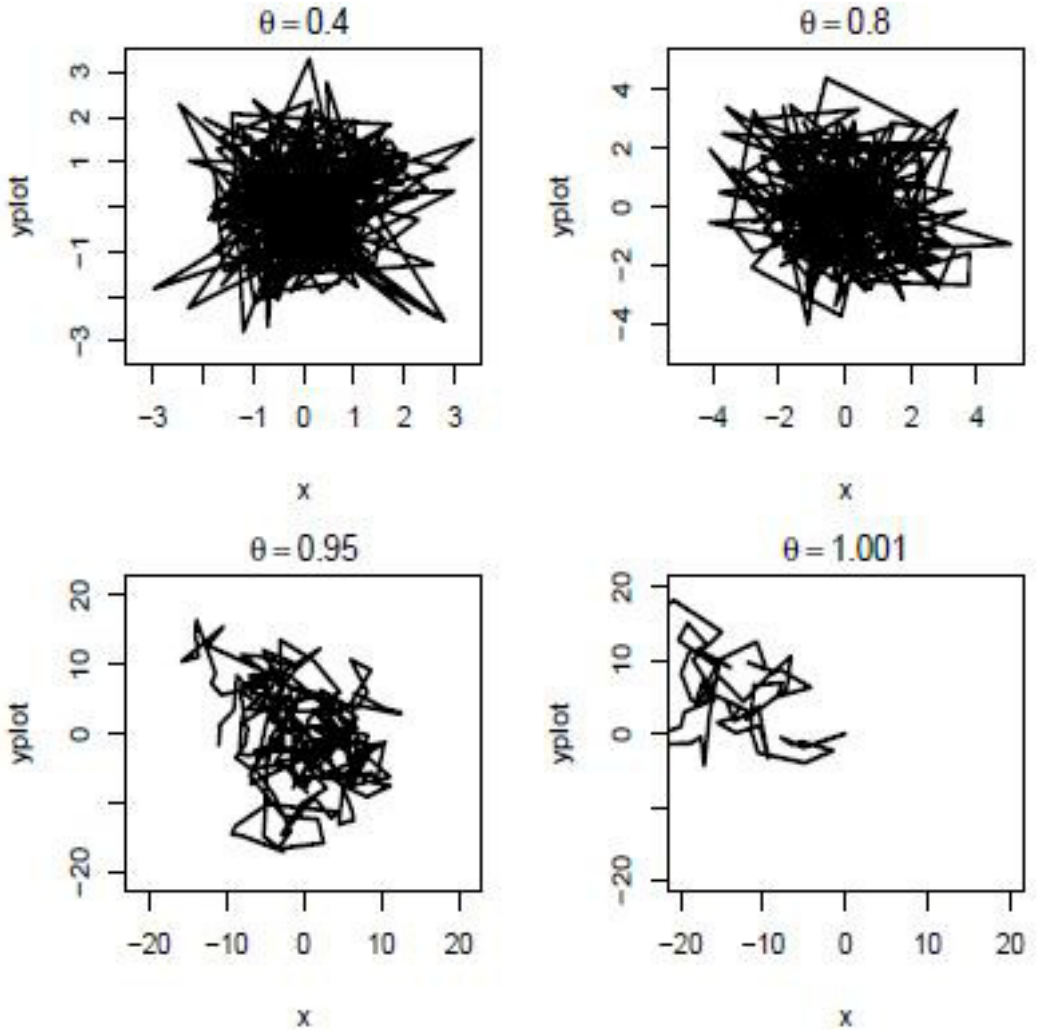
برای تشریح ویژگی بازگشتی، فرآیند $AR(1)$ را در نظر بگیرید:

$$X_t = \theta X_{t-1} + \epsilon_t; \quad \theta \in \Re$$

و $\epsilon_t \sim N(0, \sigma^2)$. اگر ϵ_tها مستقل باشند، X_t، به شرط مفروض بودن X_{t-1}، از $X_{t-2}, X_{t-3}, \ldots$ مستقل است.

توزیع مانای این زنجیر مارکوف عبارتست از $N(0, \frac{\sigma^2}{1-\theta^2})$ که باید شرط $|\theta| < 1$ برقرار باشد.

مثال: فرآیند $AR(1)$



تمرین

رنجیر مارکوف $AR(۱)$ را در نظر بگیرید که در آن $\epsilon_t \sim N(۰, ۱)$.

با شبیه‌سازی $X^{(۰)} \sim N(۰, ۱)$، هیستوگرام یک نمونه از $X^{(t)}$ را برای $t \leq ۱۰^۴$ و $\theta = ۰/۹$ رسم کنید.

بررسی کنید آیا توزیع مانای $N(۰, \frac{۱}{۱-\theta^۲})$ به این هیستوگرام برازش خوبی دارد؟

ویژگی‌های زنجیر مارکوف: ارگودیک بودن

در زنجیرهای بازگشتی، توزیع مانا یک توزیع حدی نیز هست.

به این معنی که توزیع حدی $X^{(t)}$ به ازای هر مقدار اولیه $X^{(0)}$، f است.

این ویژگی، ارگودیک بودن نیز نامیده می‌شود.

به عبارت دیگر ویژگی ارگودیک بودن زنجیر، معادل است با استقلال نسبت به شرایط اولیه.

برای زنجیرهای ارگودیک و برای توابع انتگرال‌پذیر h:

$$\frac{1}{T}\sum_{t=1}^{T} h(X^{(t)}) \longrightarrow \mathbb{E}_f[h(X)]$$

و به این معنی است که قانون قوی اعداد بزرگ برای رهیافت $MCMC$ نیز قابل کاربرد است.

که به آن قضیه ارگودیک نیز می‌گویند.

همگرایی

بیشتر از این در مورد جزییات زنجیرهای مارکوف بحث نخواهم کرد. کسانی که علاقه‌مند به جزییات بیشتر هستند، می‌توانند به فصل ۶ کتاب رابرت و کسلا (۲۰۰۴) مراجعه کنند.

با این حال مایلم حالتی را که همگرایی زنجیر به یک توزیع مانا هرگز اتفاق نمی‌افتد و مهم هم هست، مطرح کنم.

و آن حالت در چارچوب بیزی زمانی است که توزیع پسین، سره ($Proper$) نیست. زیرا در این حالت، زنجیر نمی‌تواند بازگشتی باشد.

استفاده از توزیع‌های پیشین ناسره در مدل‌های پیچیده آماری خیلی معمول است:

- گاهی موارد توزیع پسین سره می‌شود و الگوریتم $MCMC$ بازگشتی خواهد بود
- گاهی موارد نیز توزیع پسین سره نمی‌شود و الگوریتم $MCMC$ کار نخواهد کرد.

مسایل انتگرال‌گیری در استنباط بیزی

اگرچه الگوریتم‌های $MCMC$ روش‌های نمونه‌گیری عمومی هستند، اما اغلب کاربردهای آن‌ها در مسایل مربوط به استنباط بیزی دیده می‌شود.

در یک مدل بیزی، هر دوی مشاهدات و پارامترها متغیر تصادفی محسوب می‌شوند. برای یک نمونه n تایی، توزیع توام (x, θ) عبارتست از:

$$f_{x,\theta} = f_{x|\theta}(x_1, \ldots, x_n|\theta) f_\theta(\theta).$$

بنابراین می‌توان توزیع پیشین θ را با شرطی کردن بر روی اطلاعات موجود در داده‌ها، با استفاده از قاعده بیز، به‌روز کرد (توزیع پسین):

$$f_{\theta|x}(\theta|x) = \frac{f_{x|\theta}(x) f_\theta(\theta)}{\int f_{x|\theta}(x) f_\theta(\theta)\, d\theta}.$$

مسایل انتگرال‌گیری در استنباط بیزی: ادامه

بنابراین، امید ریاضی شرطی تابعی مانند $g(\theta)$ نسبت به توزیع پسین عبارتست از:

$$\mathbb{E}[g(\theta|x)] = \int g(\theta) f_{\theta|x}(\theta)\, d\theta = \frac{\int g(\theta) f_{x|\theta}(x) f_\theta(\theta)\, d\theta}{\int f_{x|\theta}(x) f_\theta(\theta)\, d\theta}.$$

این امید ریاضی حتی زمانی که چگالی پسین $f_{\theta|x}$ بدون ثابت نرمال‌سازش معلوم باشد، قابل تقریب است. این گزاره مهمی است، زیرا در اغلب موارد محاسبه ثابت نرمال‌ساز برای چگالی‌های پسین، کار دشوار و طاقت‌فرسایی است.

اما مساله اصلی، رام نبودن چنین انتگرالی و مشکل محاسبه عددی آن برای ابعاد بالاست.

$MCMC$ روشی را برای این گونه مسایل انتگرال‌گیری فراهم می‌آورد.

تاکید دوباره

ایده:
تولید یک زنجیر ارگودیک $X^{(t)}$، با مقدار اولیه $x^{(0)}$، با استفاده از یک هسته انتقال که دارای توزیع مانای f است.

- همگرایی (در توزیع) $X^{(t)}$ به متغیری با توزیع f را تضمین می‌کند.
- برای یک مقدار به اندازه کافی بزرگ T_0، $X^{(T_0)}$ را می‌توان به عنوان تحققی از توزیع f در نظر گرفت.
- یک نمونه وابسته $X^{(T_0)}, X^{(T_0+1)}, \ldots$ تولید می‌کند که از توزیع f تولید شده است به طوری که این نمونه برای اهداف برآورد کافی خواهد بود.

مساله: چطور می‌توان زنجیر مارکوفی با یک توزیع مانای مفروض ساخت؟

تولید زنجیر مارکوف با توزیع مانای مشخص

روش کار با الگوریتم‌های $MCMC$ واضح و سر راست است:

- با فرض داشتن چگالی هدف f:
  - یک هسته مارکوف K با توزیع مانای f می‌سازیم
  - سپس زنجیر مارکوف $X^{(t)} \to X \sim f$ را تولید می‌کنیم
  - در نهایت انتگرال مورد نظر با استفاده از قضیه ارگودیک قابل محاسبه است
- الگوریتم متروپولیس ـ هستینگز، مثالی از این نوع روش‌هاست
  - با شرط داشتن f، از توزیع نامزد $q(y|x)$ تولید نمونه می‌کنیم
  - تنها چیزی که برای تصمیم‌گیری در مورد پذیرش یا عدم پذیرش نمونه نیاز داریم، آن است که نسبت $\frac{f(y)}{q(y|x)}$ به جز یک ثابت، معلوم باشد

به چگالی (شرطی) $q(y|x)$، چگالی ابزاری یا پیشنهادی ($Proposal$) می‌گویند. این چگالی هر چیزی می‌تواند باشد. تنها نیاز نظری برای انتخاب $q(\cdot|x)$، به جز شرط بالا، آن است که تکیه‌گاه آن به قدری وسیع باشد که کل تکیه‌گاه f را بتواند پوشش دهد.

الگوریتم متروپولیس ـ هستینگز

الگوریتم متروپولیس ـ هستینگز ($Metropolis - Hastings$) را می‌توان به صورت زیر بیان کرد:

با شرط داشتن $x^{(t)}$،

۱. $Y_t \sim q(y|x^{(t)})$ را تولید کن

۲. قرار بده

$$X^{(t+1)} = \begin{cases} Y_t & with\ Porbability\ \ \rho(x^{(t)}, Y_t) \\ x^{(t)} & with\ Probability\ \ 1 - \rho(x^{(t)}, Y_t) \end{cases}$$

که در آن

$$\rho(x, y) = \min\left\{1, \frac{f(y)}{f(x)}\frac{q(x|y)}{q(y|x)}\right\}$$

به $\rho(x, y)$ احتمال پذیرش متروپولیس ـ هستینگز می‌گویند.

اجرای الگوریتم در R

اجرای الگوریتم در R به سادگی قابل انجام است.

با فرض داشتن تابعی برای شبیه‌سازی از $q(y|x)$، مانند $geneq(x)$، اگر $x[t]$ مقدار $X^{(t)}$ را مشخص کند، یک کد کلی برای اجرای الگوریتم بالا به صورت زیر خواهد بود:

```
y = geneq(x[t])
if (runif(1)<f(y)*q(y,x[t])/(f(x[t])*q(x[t],y))){
    x[t+1]=y
   } else {
    x[t+1]=x[t]
}
```

همگرایی الگوریتم متروپولیس ـ هستینگز

نرخ پذیرش الگوریتم، میانگین احتمال پذیرش بر روی تکرارهاست. یعنی:

$$\bar{\rho} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T} \rho(X^{(t)}, Y_t) = \int \rho(x,y) f(x) q(y|x)\, dy dx.$$

از این کمیت برای ارزیابی عملکرد الگوریتم، می‌توان استفاده کرد.

الگوریتم متروپولیس ـ هستینگز در شرطی موسوم به شرط تعادل دقیق
(*Detailed Balance Condition*)

$$f(x)K(y|x) = f(y)K(x|y)$$

صدق می‌کند. با انتگرال‌گیری از طرفین تساوی بالا نسبت به x، می‌توان نتیجه گرفت، توزیع مانای زنجیر $\{X^{(t)}\}$ برابر f است.

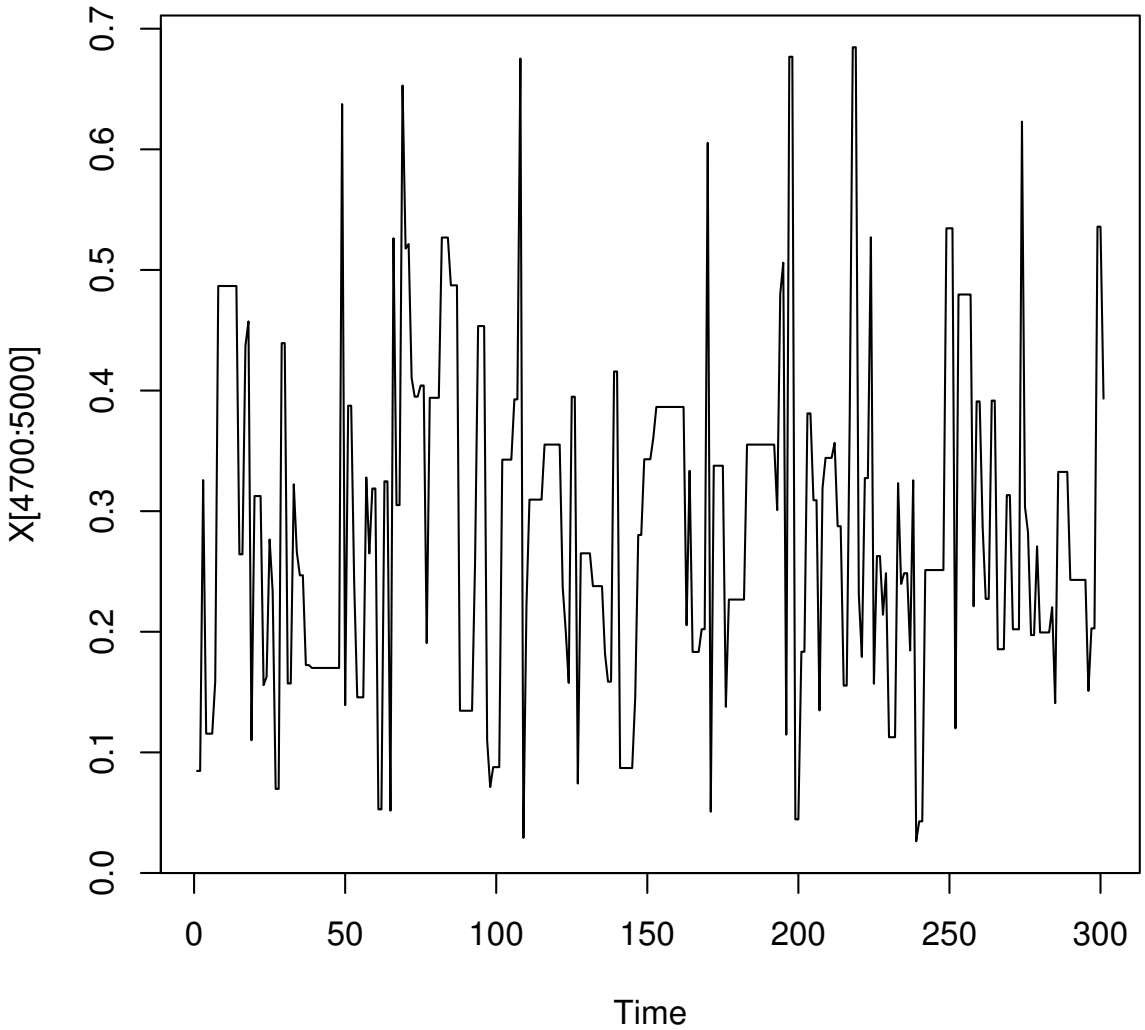
یک مثال ساده

در عمل، عملکرد الگوریتم MH قویا به انتخاب توزیع پیشنهادی q وابسته است. اما در این جا برای مقایسه با نمونه‌گیری مستقیم از یک توزیع، به مثال ساده‌ای توجه کنید.

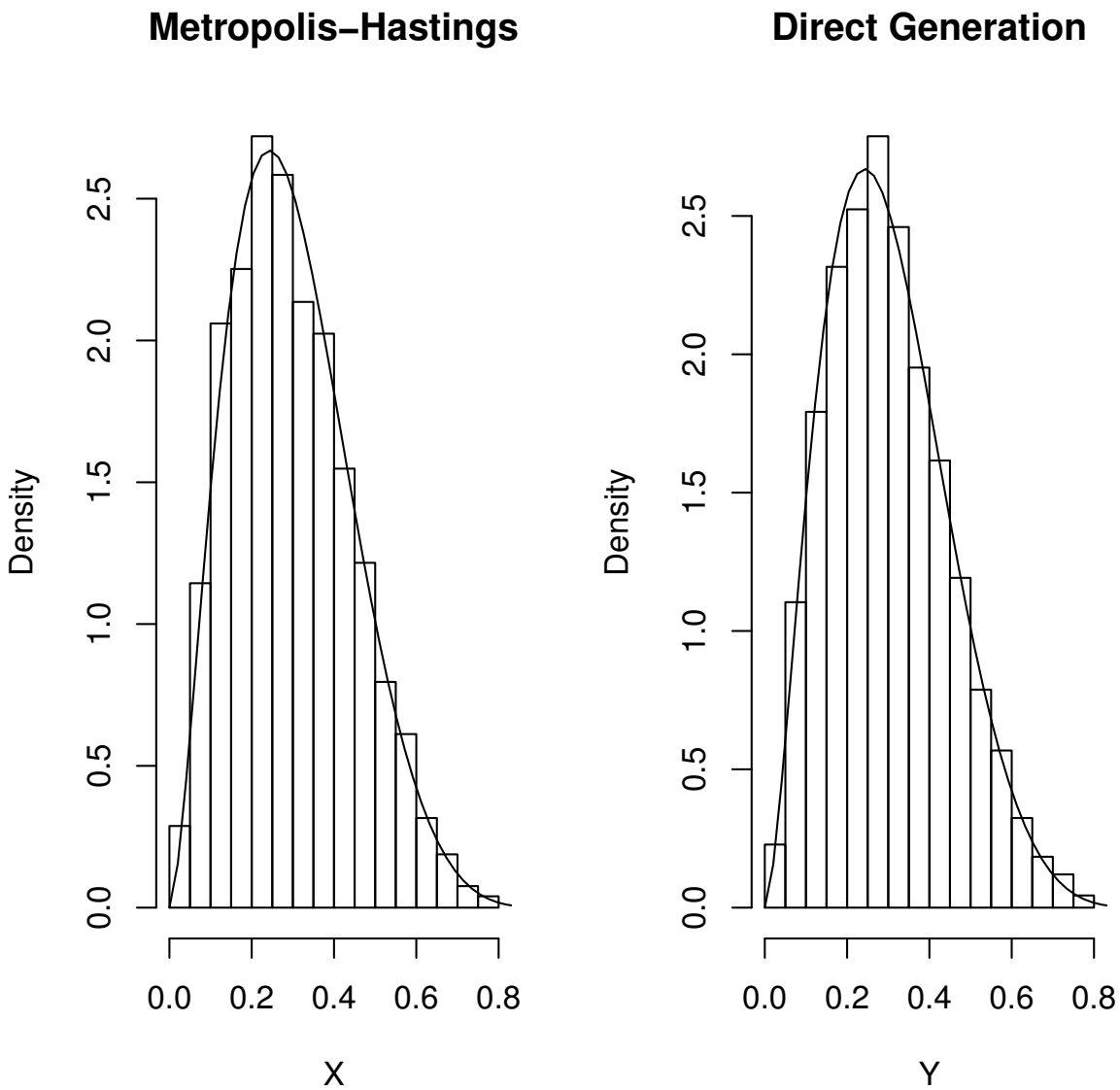
فرض کنید توزیع هدف $Beta(۲/۷, ۶/۳)$ و توزیع پیشنهادی $U(۰, ۱)$ باشد. یک نمونه MH را می‌توان با کد زیر تولید کرد:

```
a=2.7; b=6.3; c=2.669 # initial values
n=5000
X=rep(runif(1),n) # initialize the chain
for (i in 2:n){
    Y=runif(1)
    rho=dbeta(Y,a,b)/dbeta(X[i-1],a,b)
    X[i]=X[i-1] + (Y-X[i-1])*(runif(1)<rho)
}
```

یک مثال ساده: ادامه



یک مثال ساده: ادامه



یک مثال ساده: ادامه

دقت کنید که اگرچه دو روش نمونه‌گیری به نظر یکسان هستند، اما نمونه $MCMC$ وابستگی دارند، در حالی که نمونه مستقیم iid هستند.

این به آن معنی است که در نمونه‌گیری $MCMC$، از کیفیت نمونه کاسته می‌شود یا به عبارت دیگر برای دستیابی به دقت یکسان با نمونه‌گیری مستقیم نیازمند حجم نمونه بزرگتری هستیم.

این مساله از طریق مفهومی به نام حجم نمونه موثر (*Effective Sample Size*) برای زنجیرهای مارکوف، مطرح می‌شود.

چند نکته

- برای توزیع‌های پیشنهادی متقارن، یعنی وقتی که $q(x|y) = q(y|x)$،

$$\rho(x_t, y_t) = \min\{1, \frac{f(y_t)}{f(x_t)}\}.$$

و بنابراین احتمال پذیرش مستقل از q خواهد بود.

- الگوریتم همیشه مقادیر y_t را که برای آن‌ها

$$\frac{f(y_t)}{q(y_t|x^{(t)})} > \frac{f(x^{(t)})}{q(x^{(t)}|y_t)},$$

می‌پذیرد. البته اگر جهت نامساوی عوض شود، ممکن است باز هم پذیرفته شود. اما اگر اختلاف زیاد باشد، مقادیر پیشنهادی اغلب موارد رد خواهند شد. این ویژگی نشان می‌دهد که چگونه انتخاب q می‌تواند بر روی عملکرد الگوریتم MH تاثیر بگذارد.

چند نکته: ادامه

- اگر تکیه‌گاه q در مقایسه با تکیه‌گاه f خیلی کوچک باشد، زنجیر مارکوف تولید شده در جستجوی دامنه f دچار مشکل خواهد شد و در نتیجه خیلی کند همگرا خواهد شد.
- الگوریتم MH فقط به نسبت‌های $\frac{f(y_t)}{f(x^{(t)})}$ و $\frac{q(x^{(t)}|y_t)}{q(y_t|x^{(t)})}$ وابسته است. در نتیجه الگوریتم مستقل از ثابت‌های نرمال‌ساز می‌باشد.
- زنجیر $\{x^{(t)}\}$ ممکن است چندین بار پشت سر هم مقادیر یکسانی داشته باشد، حتی اگر f یک چگالی (نسبت به اندازه لبگ) باشد.
- دنباله $\{y_t\}$ معمولا تشکیل یک زنجیر مارکوف نمی‌دهد. به این معنی که اگر مقادیر پذیرش‌شده را از زنجیر استخراج کنیم، تمام نتایج همگرایی به توزیع مانا خدشه‌دار خواهد شد و دیگر برقرار نخواهند بود.

الگوریتم MH مستقل

در حالتی که توزیع پیشنهادی q مستقل از $X^{(t)}$ باشد، الگوریتم حاصل با این انتخاب را الگوریتم MH مستقل گویند. در این حالت $q(y|x) = g(y)$.

با شرط داشتن $x^{(t)}$،

1. $Y_t \sim g(y)$ را تولید کن
2. قرار بده

$$X^{(t+1)} = \begin{cases} Y_t & with\ Porbability\ \ \min\left\{1, \frac{f(Y_t)g(x^{(t)})}{f(x^{(t)})g(Y_t)}\right\} \\ x^{(t)} & otherwise \end{cases}$$

ویژگی‌های الگوریتم MH مستقل

- این الگوریتم تعمیمی از روش رد و پذیرش است. چون در هر دو روش، توزیع ابزاری یکی است، مقادیر پیشنهادی Y_t مشابه هم هستند اما مقادیر پذیرش شده یکی نیستند.
- با اینکه این دو روش شباهت‌هایی دارند و می‌توان در یک مساله خاص از هر دو استفاده کرد و نتایج را مقایسه کرد، اما اگر در یک مساله بتوان از الگوریتم رد و پذیرش استفاده کرد، به ندرت از روش MH استفاده خواهد شد.
- نمونه‌های روش رد و پذیرش مستقل هستند، اما در مورد روش MH این طور نیست.
- در روش رد و پذیرش، نیازمند محاسبه کران بالای $M \geq \sup_x \frac{f(x)}{g(x)}$ هستیم. اما الگوریتم MH نیاز به محاسبه چنین کمیتی ندارد. این یک جنبه خوب روش MH است زمانی که محاسبه M زمان‌بر است یا M موجود ناکاراست و باعث تلف شدن بخش عمده‌ای از نمونه‌های پیشنهادی تولید شده می‌شود.

در واقع روش MH روش رد و پذیرش برای افراد تنبل است!

مثال

مجموعه داده $cars$ در هسته R، شامل سرعت و فاصله ترمز یک نمونه از اتومبیل‌هاست.

به این داده‌ها یک مدل درجه دو برازش می‌دهیم. یعنی مدل به صورت

$$y_{ij} = \beta_۰ + \beta_۱ x_i + \beta_۲ x_i^۲ + \epsilon_{ij},\ i = ۱, \ldots, k,\ j = ۱, \ldots, n_i$$

است، که در آن فرض می‌کنیم $\epsilon_{ij} \sim N(۰, \sigma^۲)$ و مستقل هستند.

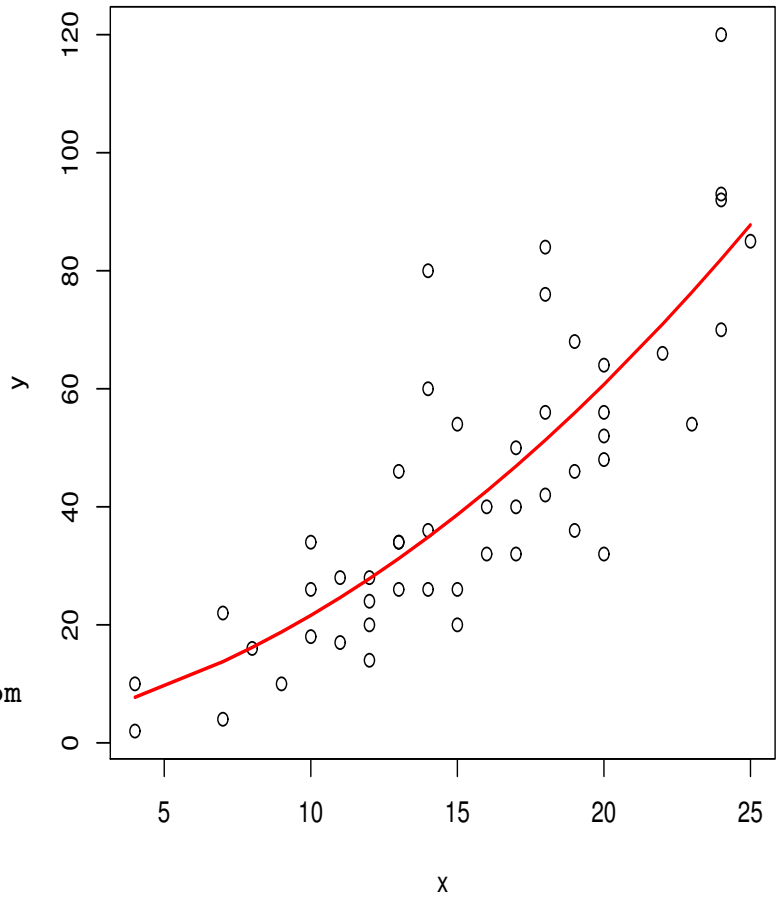
تابع درستنمایی متناسب است با

$$(\frac{۱}{\sigma^۲})^{N/۲} \exp\left\{-\frac{۱}{۲\sigma^۲}\sum_{ij}(y_{ij} - \beta_۰ - \beta_۱ x_i - \beta_۲ x_i^۲)^۲\right\}$$

که در آن $N = \sum_i n_i$ مجموع کل مشاهدات است.

برازش کمترین توان‌های دوم

```
data(cars)
x=cars$speed
y=cars$dist
x2=x^2
#
fit=lm(y~x+x2)
#
plot(y~x)
lines(x,predict(fit),col="red",lwd=2)
> summary(fit)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.47014   14.81716   0.167    0.868
x            0.91329    2.03422   0.449    0.656
x2           0.09996    0.06597   1.515    0.136
Residual standard error: 15.18 on 47 degrees of freedom
```



الگوریتم MH

می‌توان به تابع درستنمایی به عنوان توزیع پسین پارامترهای $\beta_0, \beta_1, \beta_2, \sigma^2$ نگاه کرد (به عنوان مثال با در نظر گرفتن توزیع‌های پیشین یکنواخت).

و به عنوان یک مثال برای نمایش نحوه اجرای الگوریتم، با استفاده از الگوریتم MH از این توزیع نمونه تولید کرد (دقت کنید چون توزیع نرمال است، به راحتی می‌توان به طور مستقیم از آن تولید نمونه کرد).

برای شروع، مقادیر اولیه، یعنی $X^{(0)} = (\beta_0^{(0)}, \beta_1^{(0)}, \beta_2^{(0)}, \sigma^{2(0)})$، برآوردهای کمترین توان‌های دوم در نظر می‌گیریم.

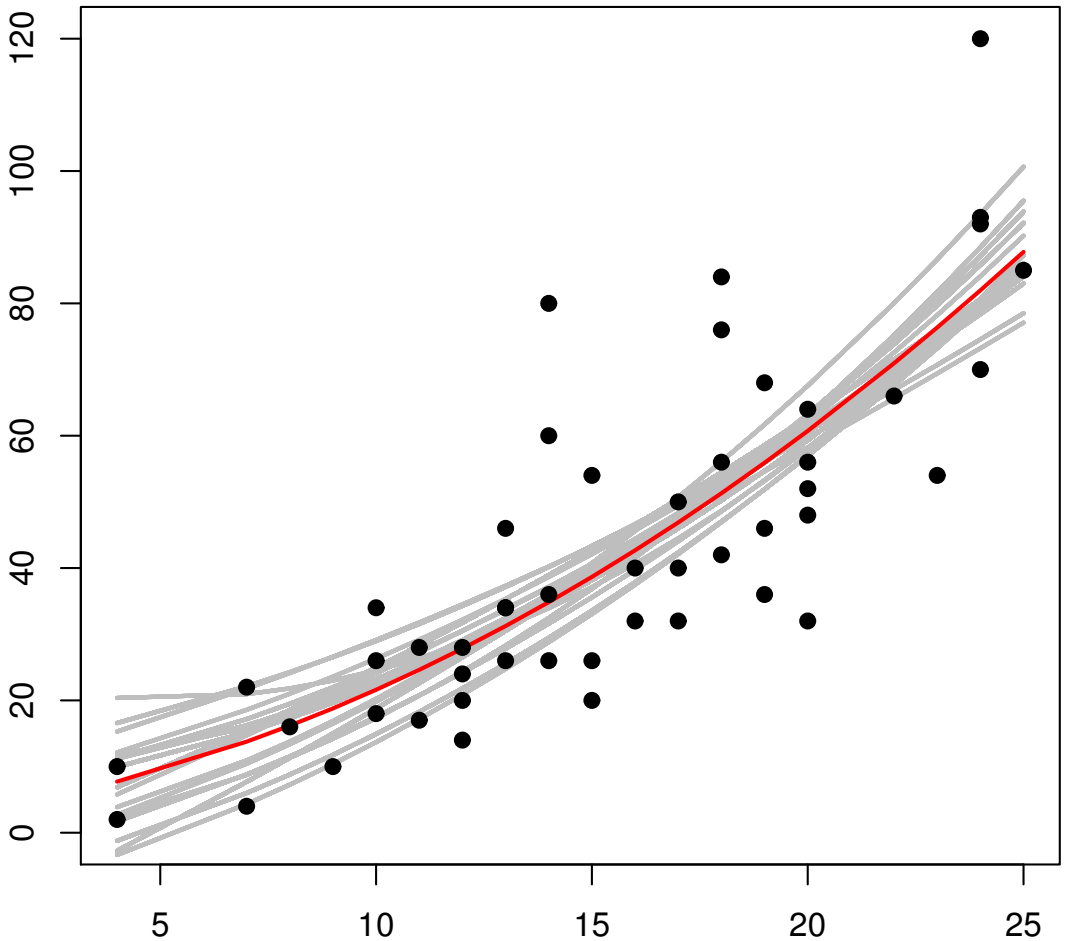
توزیع‌های پیشنهادی را نیز به صورت زیر در نظر می‌گیریم که همگی در MLE مرکزی شده‌اند:

$$\beta_0 \sim N(2/47, 15^2), \quad \beta_1 \sim N(0/91, 2^2), \quad \beta_2 \sim N(0/100, 0/06^2)$$

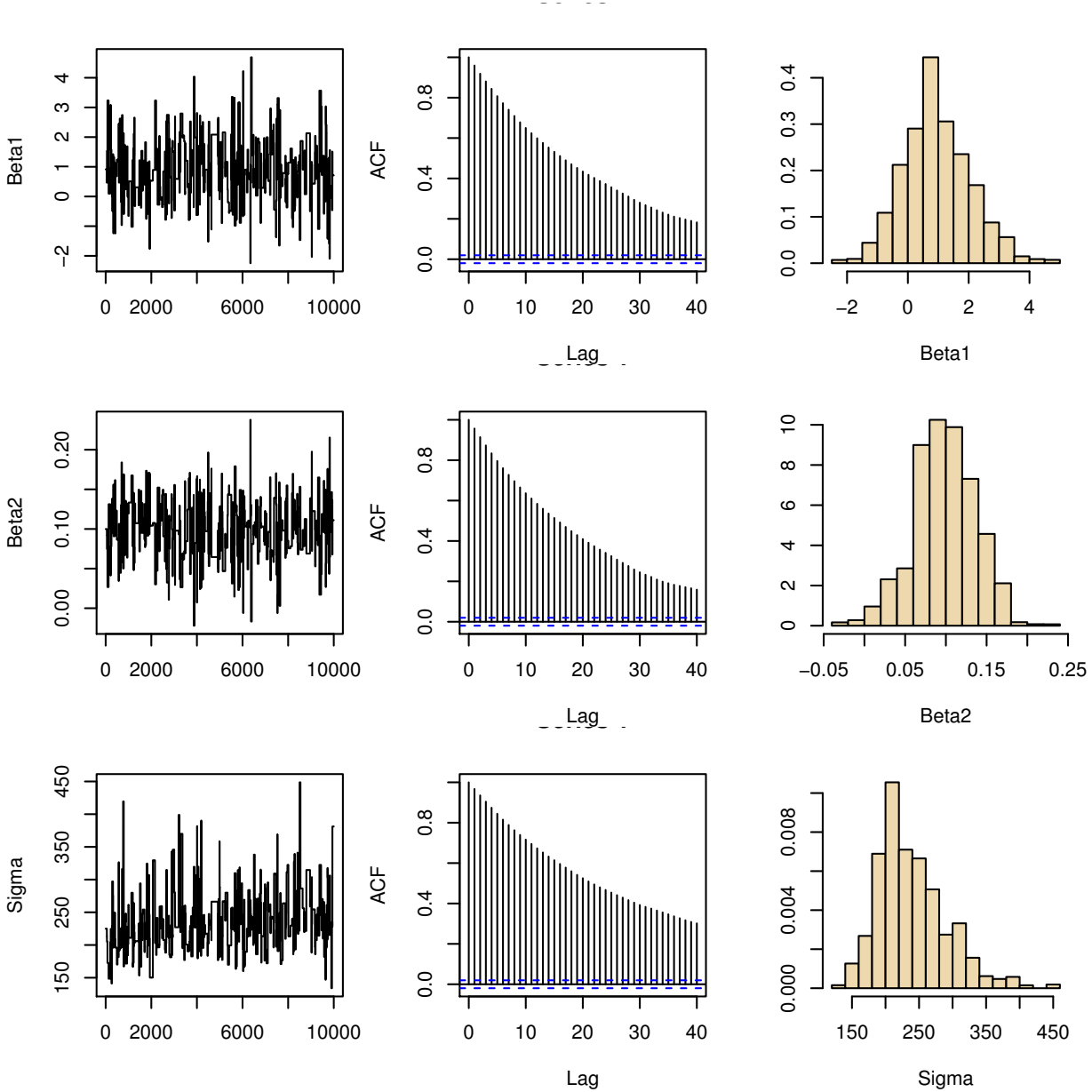
$$\sigma^{-2} \sim Gamm(N/2, (N-3)(15/18)^2)$$

```
Barking = function (nsim = 10^3){
    data = cars; x = data[, 1]; y = data[, 2]; n = length(x); x2 = x^2
    MSR = 225;  SSR = 10809
    bhat = coefficients(lm(y ~ x + x2))
    loglike = function(a, b, c, s2) { -(n/2) * log(s2) - sum((y - a - b * x - c * x2)^2)/(2 * s2) }
    dcand = function(a, b, c, s2) {
        dnorm(a, bhat[1], sd = 15, log = TRUE) + dnorm(b, bhat[2],
            sd = 2, log = TRUE)
        +dnorm(c, bhat[3], sd = 0.06, log = TRUE) - (n/2) * log(s2) -
            SSR/(2 * s2)
    }
    b1hat = array(bhat[1], dim = c(nsim, 1))
    b2hat = array(bhat[2], dim = c(nsim, 1))
    b3hat = array(bhat[3], dim = c(nsim, 1))
    s2hat = array(MSR, dim = c(nsim, 1))
    for (i in 2:nsim) {
        bcand = c(rnorm(1, mean = bhat[1], sd = 15), rnorm(1,
            mean = bhat[2], sd = 2), rnorm(1, mean = bhat[3],
            sd = 0.06), 1/rgamma(1, n/2, rate = SSR/2))
        test = min(exp(loglike(bcand[1], bcand[2], bcand[3],
            bcand[4]) - loglike(b1hat[i - 1], b2hat[i - 1], b3hat[i -
            1], s2hat[i - 1]) + dcand(b1hat[i - 1], b2hat[i -
            1], b3hat[i - 1], s2hat[i - 1]) - dcand(bcand[1],
            bcand[2], bcand[3], bcand[4])), 1)
        rho <- (runif(1) < test)
        b1hat[i] = bcand[1] * rho + b1hat[i - 1] * (1 - rho)
        b2hat[i] = bcand[2] * rho + b2hat[i - 1] * (1 - rho)
        b3hat[i] = bcand[3] * rho + b3hat[i - 1] * (1 - rho)
        s2hat[i] = bcand[4] * rho + s2hat[i - 1] * (1 - rho)
    }
    return(list('Beta0'=b1hat, 'Beta1'=b2hat,'Beta2'=b3hat,'Sigma'=s2hat))
}
```

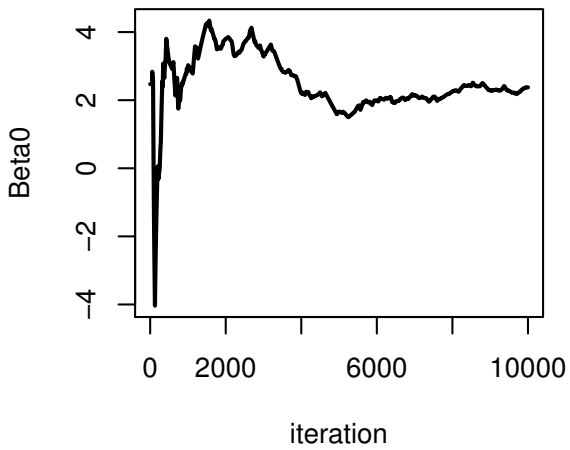
الگوریتم MH : نتیجه برازش

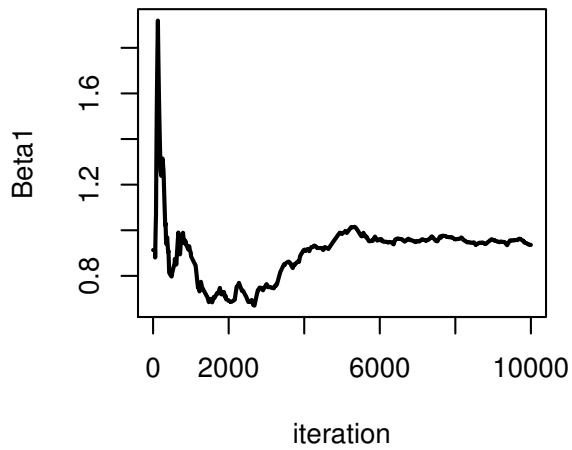


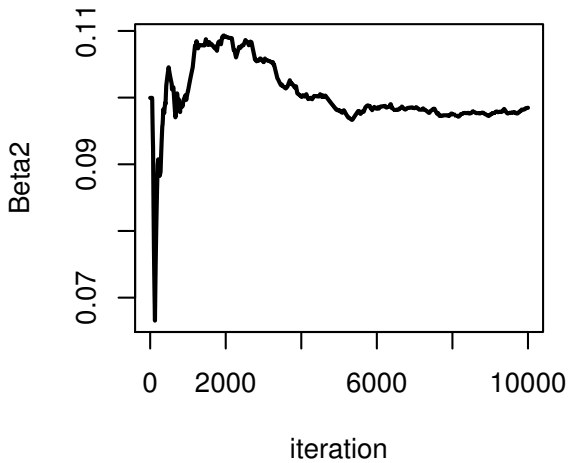
الگوریتم MH : آمیختگی زنجیر

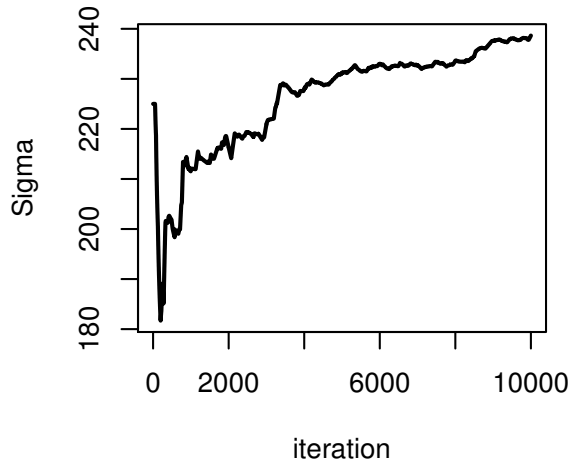


الگوریتم MH: بررسی همگرایی زنجیر









الگوریتم MH قدم زدن تصادفی: مقدمه

اجرای الگوریتم MH مستقل گاهی مشکل‌ساز می‌شود:

- ساخت چگالی پیشنهادی ممکن است پیچیده باشد
- توزیع‌های پیشنهادی در این نوع الگوریتم، اطلاعات موضعی را نادیده می‌گیرند

یک راه جانشین، گردآوری اطلاعات مرحله به مرحله است:

- یعنی کاوش همسایگی موقعیت جاری زنجیر
- این ایده، برای تولید مقدار بعدی زنجیر، نمونه تولید شده قبلی را به حساب می‌آورد. به این معنی که در همسایگی موقعیت جاری زنجیر، کاوش موضعی بیشتری را انجام می‌دهد.

الگوریتم MH قدم زدن تصادفی

اجرای ایده کاوش موضعی توسط کاوش در همسایگی مقدار جاری زنجیر، شبیه‌سازی Y_t از رابطه

$$Y_t = X^{(t)} + \epsilon_t,$$

است، که در آن ϵ_t یک نوفه تصادفی مستقل از $X^{(t)}$ با تابع چگالی g است. به عنوان مثال، انتخاب توزیع یکنواخت برای نوفه به این معنی است که

$$Y_t \sim U(X^{(t)} - \delta, X^{(t)} + \delta)$$

یا توزیع نرمال به معنی

$$Y_t \sim N(X^{(t)}, \tau^2)$$

است. در این حالت توزیع پیشنهادی عبارتست از

$$q(y|x) = g(y - x).$$

اگر چگالی g حول صفر متقارن باشد، که معمولا این طور است، زنجیر مارکوف متناظر با آن، یک فرآیند قدم زدن تصادفی است.

الگوریتم

با شرط داشتن $x^{(t)}$،

۱. $Y_t \sim g(y - x^{(t)})$ را تولید کن

۲. قرار بده

$$X^{(t+1)} = \begin{cases} Y_t & with\ Porbability\ \ \min\left\{1, \frac{f(Y_t)}{f(x^{(t)})}\right\} \\ x^{(t)} & otherwise \end{cases}$$

ویژگی‌های الگوریتم

- به دلیل وجود مرحله پذیرش متروپولیس‌ـ هستینگز، زنجیر $\{X^{(t)}\}$ یک زنجیر قدم زدن تصادفی نیست.
- احتمال پذیرش به g وابسته نیست. به این معنی که، برای یک جفت مفروض $(x^{(t)}, y_t)$، احتمال پذیرش چه y_t از توزیع نرمال تولید شود چه از توزیع کوشی، یکسان خواهد بود.
- اما انتخاب gهای مختلف منجر به دامنه‌های مختلف مقادیر برای Y_t ها و در نتیجه نرخ‌های پذیرش متفاوت خواهد شد. بنابراین نمی‌توان گفت انتخاب g بر رفتار الگوریتم تاثیری ندارد.

کالبیدن پارامتر مقیاس

پارامتر مقیاس فرآیند قدم زدن تصادفی، که وابسته به واریانس نوفه تصادفی است، باید به گونه‌ای انتخاب شود (کالبیده شود) تا کاوش به خوبی انجام شود.

این کالبیدن برای رسیدن به یک تقریب مناسب از توزیع هدف در تعداد تکرار قابل قبول، خیلی مهم است.

- اگر واریانس جمله نوفه خیلی بزرگ باشد، اکثر نمونه‌های پیشنهادی رد می‌شوند و الگوریتم شدیدا ناکاراست
- وقتی که این واریانس خیلی کوچک است، اکثر نمونه‌های پیشنهادی پذیرفته شده و زنجیری تولید می‌شود که خیلی شبیه به یک فرآیند قدم زدن تصادفی است و باز هم ناکاراست.

یک راه برای انتخاب پارامتر مقیاس، بررسی نرخ‌های پذیرش است. اگر نرخ پذیرش الگوریتم در دامنه [۰/۱۵, ۰/۵] قرار گیرد، مناسب است.

مثال: تولید توزیع t

هدف: تولید نمونه از توزیع t با ν درجه آزادی، با استفاده از توزیع پیشنهادی $N(X^{t)}, \sigma^{۲})$ و کالبیدن مقیاس زنجیر با انتخاب مقادیر متفاوت برای σ

در این حالت داریم:

$$\frac{f(y_t)}{f(x^{(t)}} = \frac{(۱ + \frac{y_t^{۲}}{\nu})^{-(\nu+۱)/۲}}{(۱ + \frac{x^{(t)۲}}{\nu})^{-(\nu+۱)/۲}}$$

مثال: کد R

```
rw.Metropolis <- function(nu, sigma, x0, n) {
x <- numeric(n)
x[1] <- x0
u <- runif(n)
k <- 0
for (i in 2:n) {
     y <- rnorm(1, x[i-1], sigma)
     if (u[i] <= (dt(y, nu) / dt(x[i-1], nu)))
     x[i] <- y  else {
              x[i] <- x[i-1]
              k <- k + 1
              }
     }
     k=(1-k/n)
     return(list(x=x, k=k))
}
```

مثال: نرخ پذیریش

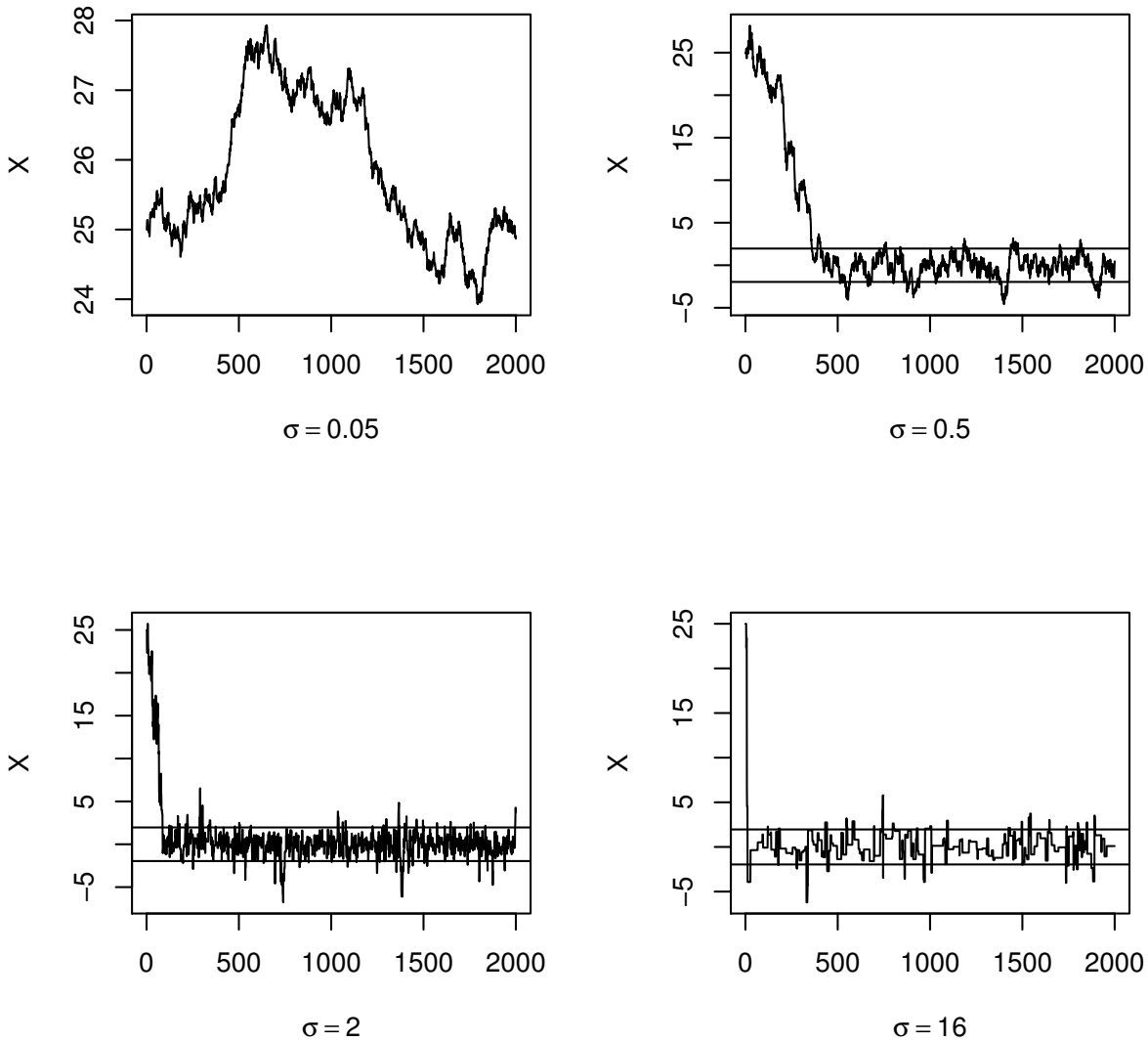
```
nu <- 4 # degrees of freedom
n <- 2000
sigma <- c(.05, .5, 2,  16)
x0 <- 25
##
rw1 <- rw.Metropolis(nu, sigma[1], x0, n)
rw2 <- rw.Metropolis(nu, sigma[2], x0, n)
rw3 <- rw.Metropolis(nu, sigma[3], x0, n)
rw4 <- rw.Metropolis(nu, sigma[4], x0, n)
# number of candidate points rejected
> print(c(rw1$k, rw2$k, rw3$k, rw4$k))
[1] 0.9955 0.8605 0.5460 0.0985
```

تنها زنجیر سوم مقداری نزدیک به دامنه [۰/۱۵, ۰/۵] دارد.

مثال: کد R برای تولید نمودارهای اثر و چندک‌ها

```
par(mfrow=c(2,2)) # display 4 graphs together
refline <- qt(c(.025, .975), df=n)
rw <- cbind(rw1$x, rw2$x, rw3$x,  rw4$x)
for (j in 1:4) {
     plot(rw[,j], type="l",
     xlab=bquote(sigma == .(round(sigma[j],3))),
     ylab="X", ylim=range(rw[,j]))
     abline(h=refline)
}
## Computing quantiles of target distribution and chains
a <- c(.05, seq(.1, .9, .1), .95)
Q <- qt(a, n)
rw <- cbind(rw1$x, rw2$x, rw3$x, rw4$x)
mc <- rw[501:N, ]
Qrw <- apply(mc, 2, function(x) quantile(x, a))
print(round(cbind(Q, Qrw), 3))
```

مثال: نمودارهای اثر



مثال: مقایسه چندک‌ها با مقادیر واقعی آن‌ها از توزیع t

	Q	V2	V3	V4	V5
5%	-1.65	24.30	-2.82	-2.20	-1.77
10%	-1.28	24.43	-2.05	-1.51	-1.38
20%	-0.84	24.85	-1.28	-0.89	-0.99
30%	-0.52	25.02	-0.83	-0.59	-0.64
40%	-0.25	25.22	-0.45	-0.27	-0.25
50%	0.00	25.83	-0.13	-0.05	0.09
60%	0.25	26.70	0.17	0.21	0.24
70%	0.52	26.89	0.47	0.51	0.92
80%	0.84	27.10	0.89	0.87	1.31
90%	1.28	27.42	1.38	1.30	1.79
95%	1.65	27.58	1.79	1.70	2.14

مثال: آمیخته نرمال

هدف: نمونه‌گیری از یک توزیع نرمال آمیخته دو متغیره دو مولفه‌ای با مشخصات زیر است:

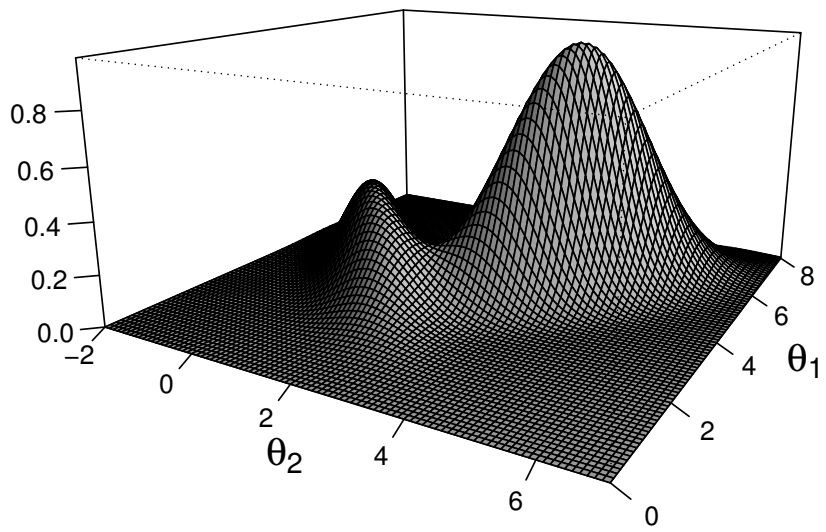
$$\mu_۱ = (۴, ۵)^T,\ \mu_۲ = (۰/۷, ۳/۵)^T,$$

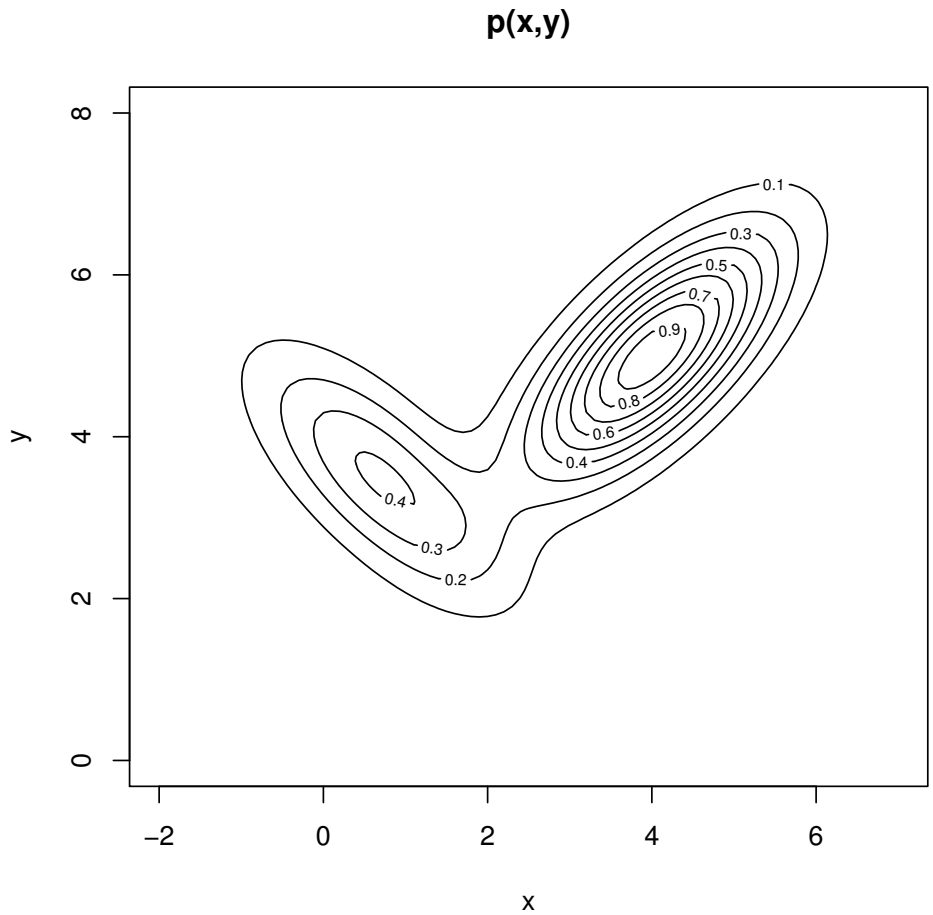
$$\Sigma_۱ = \begin{pmatrix} ۱ & ۰/۷ \\ ۰/۷ & ۱ \end{pmatrix},\ \Sigma_۲ = \begin{pmatrix} ۱ & -۰/۷ \\ -۰/۷ & ۱ \end{pmatrix}$$

توزیع پیشنهادی را نیز نرمال دومتغیره با بردار میانگین صفر و ماتریس قطری $\sigma \times I_۲$ در نظر می‌گیریم، که در آن سه مقدار متفاوت برای پارامتر مقیاس σ تعریف می‌کنیم:
$\sigma = (۰/۰۱, ۱, ۱۰۰)$.

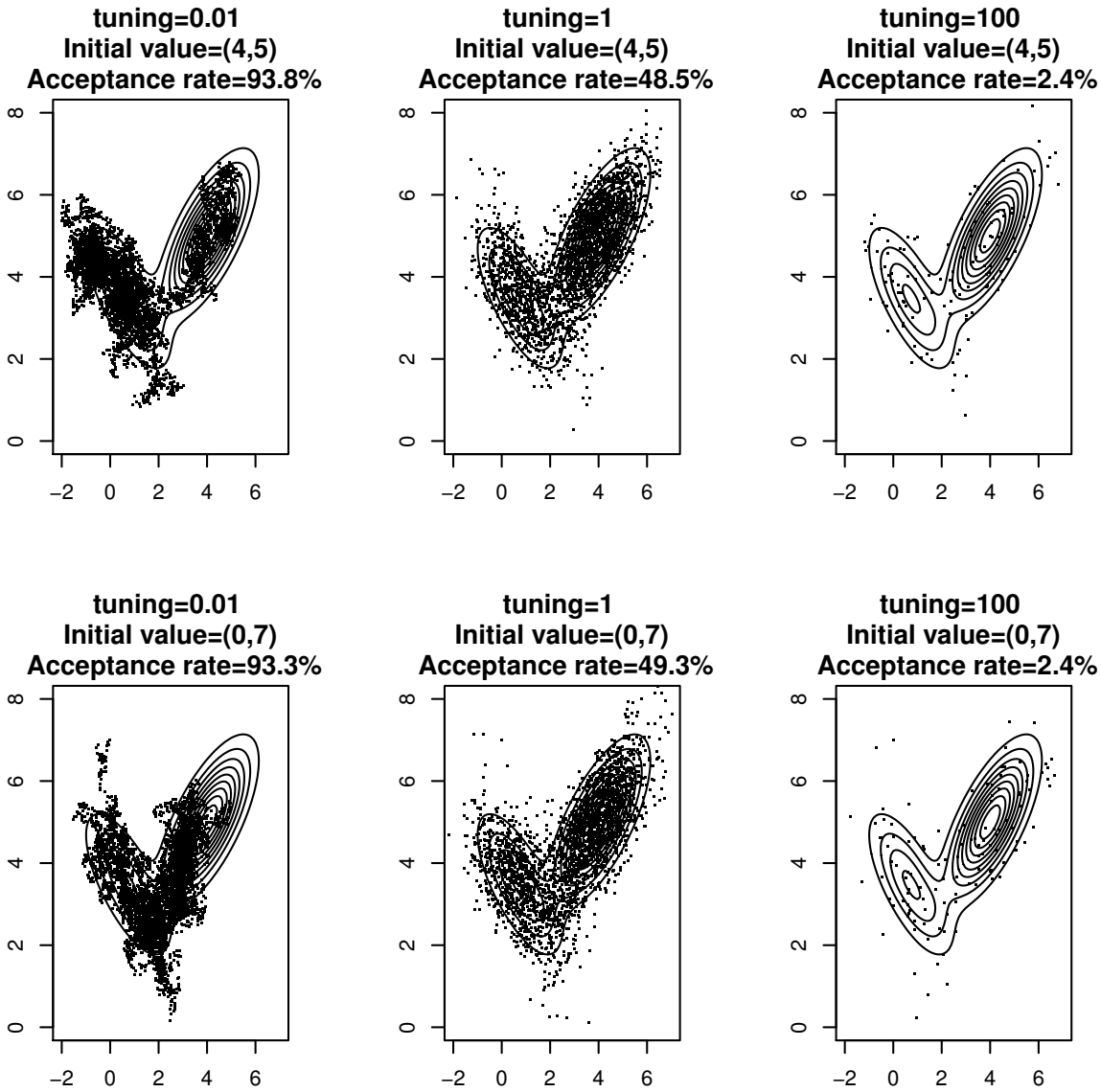
کد برنامه R از طریق صفحه شخصی من، قسمت تابلو اعلانات قابل دسترسی است.

مثال: آمیخته نرمال

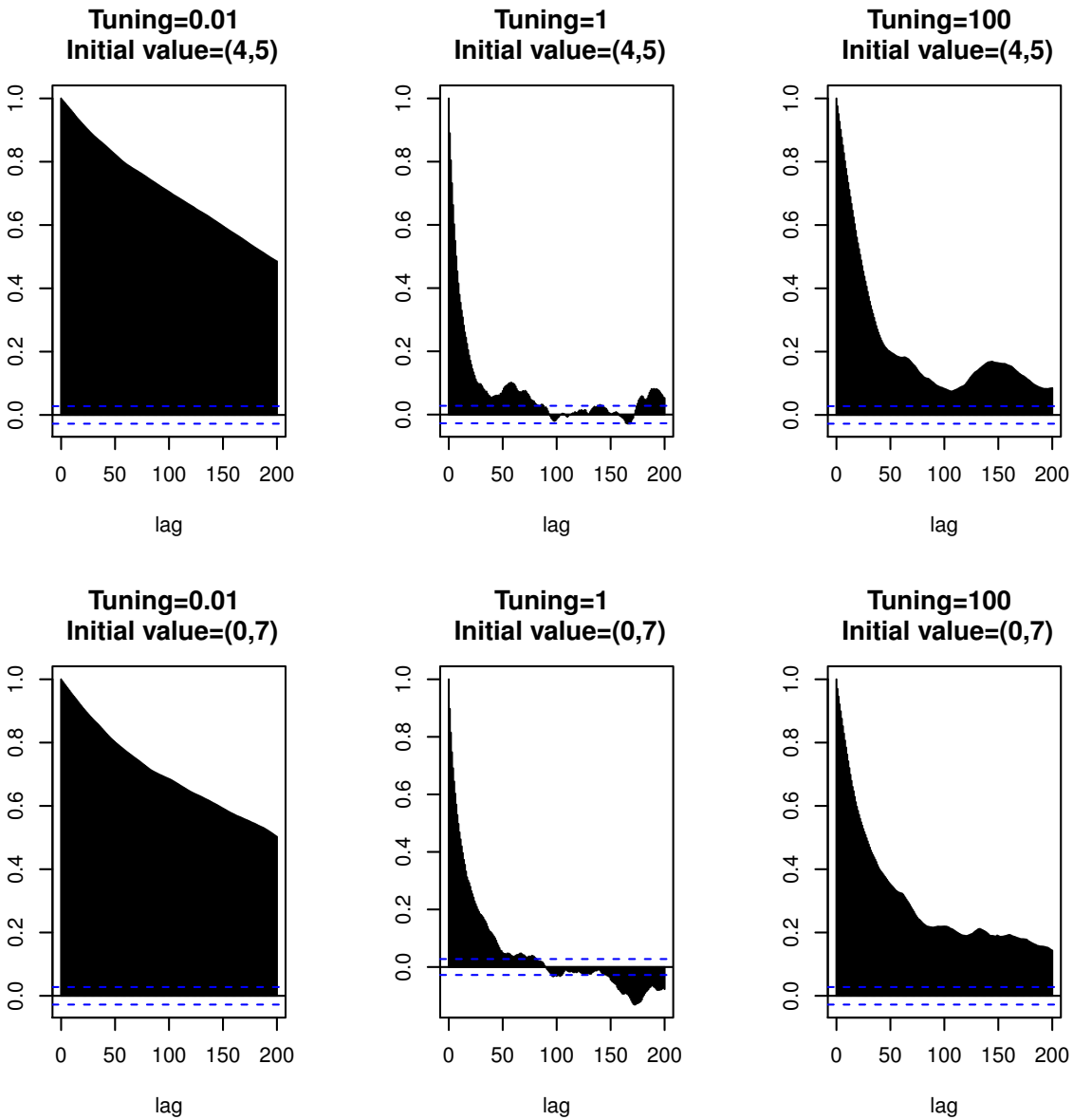




مثال: آمیخته نرمال



مثال: آمیخته نرمال



الگوریتم نمونه‌گیر گیبز: مقدمه

نام نمونه‌گیری گیبز از مقاله معروف گمان و گمان (۱۹۸۴) گرفته شده است.

اما استفاده شایع از این الگوریتم با کار گلفاند و اسمیت (۱۹۹۰) شروع شد، که از این الگوریتم برای حل مسایلی، در استنباط بیزی، که قبلا بدون حل مانده بودند، استفاده کردند.

الگوریتم‌های نمونه‌گیر گیبز، مسایل پیچیده (با بعد بالا)، مانند یک تابع چگالی هدف با بعد بالا، را به دنباله‌ای از مسایل ساده‌تر تقسیم می‌کنند

- ممکن است این مسایل پیچیده به وسیله الگوریتم MH قابل حل نباشند.

ممکن است دنباله مسایل ساده‌تر زمان زیادی برای همگرا شدن نیاز داشته باشند

- با این وجود این الگوریتم یک الگوریتم مفید و جالب توجه است.

الگوریتم نمونه‌گیر گیبز: مقدمه

الگوریتم گیبز، با استفاده از یک توزیع توام یک زنجیر مارکوف تشکیل می‌دهد:

- اگر دو متغیر تصادفی X و Y دارای توزیع توام $f(x, y)$ باشند
- به طوری که چگالی‌های شرطی متناظرشان $f_{X|Y}$ و $f_{Y|X}$ باشند
- الگوریتم یک زنجیر مارکوف (X_t, Y_t) را بر اساس مراحل زیر، می‌سازد:
  با شرط مفروض بودن $X_0 = x_0$، برای $t = 1, 2, \ldots,$ تولید کن:
  1. $Y_t \sim f_{Y|X}(\cdot|x_{t-1})$
  2. $X_t \sim f_{X|Y}(\cdot|y_t)$

چنانچه شبیه‌سازی از هر دو چگالی شرطی راحت باشد، اجرای الگوریتم بسیار روشن و ساده است.

توزیع مانای این زنجیر، $f(x, y)$ است و همگرایی زنجیر، به جز مواردی که تکیه‌گاه توزیع‌های شرطی بهم وصل نباشند، تضمین‌شده است.

مثال: نرمال دومتغیره

در اینجا مثال ساده‌ای را در نظر می‌گیریم: مدل نرمال دومتغیره

$$(X, Y) \sim N_۲ \left(۰, \begin{pmatrix} ۱ & \rho \\ \rho & ۱ \end{pmatrix} \right)$$

که برای آن نمونه‌گیر گیبز عبارتست از:

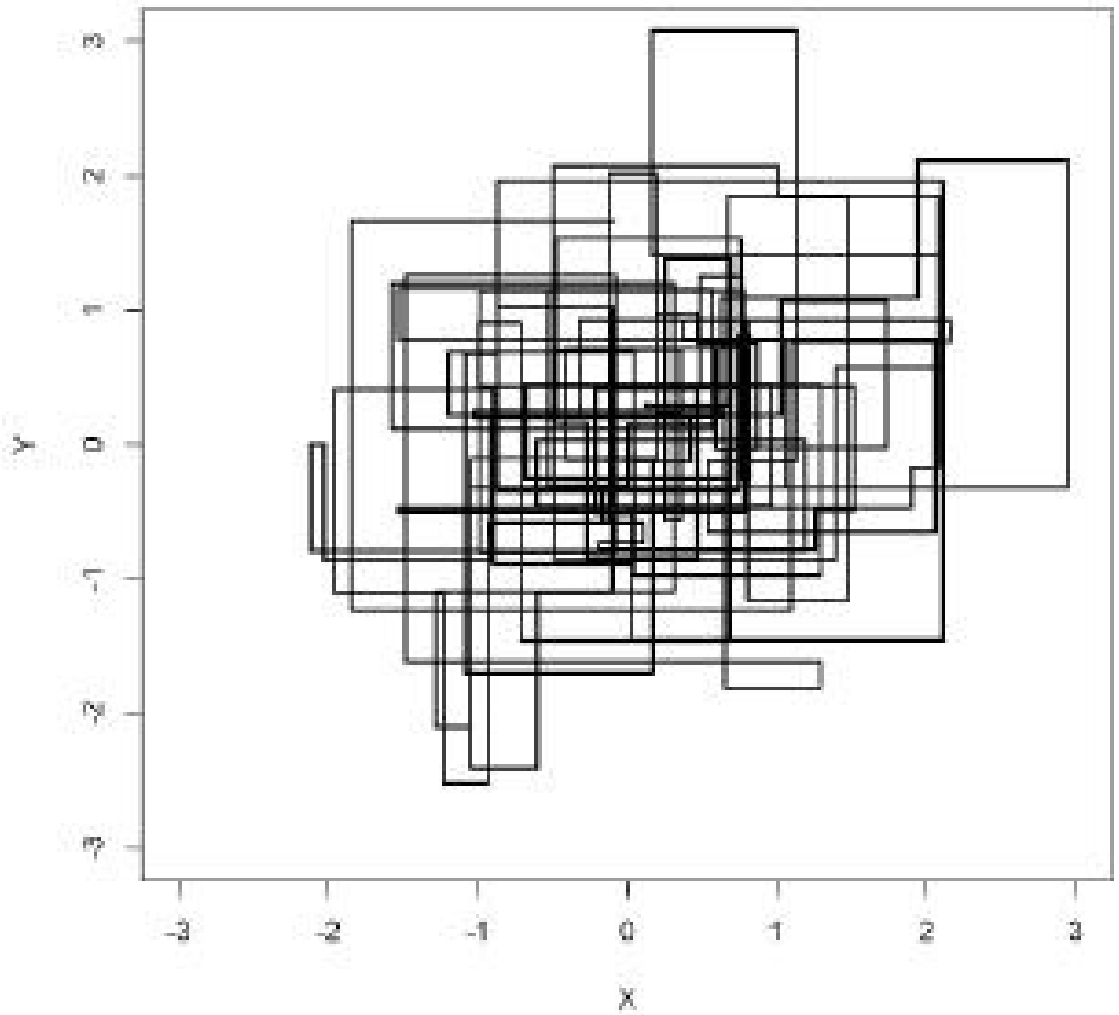
برای مقدار مفروض x_t، مقادیر زیر را تولید کن

$$\begin{aligned} Y_{t+1}|x_t &\sim N(\rho x_t, 1-\rho^2) \\ X_{t+1}|y_{t+1} &\sim N(\rho y_{t+1}, 1-\rho^2) \end{aligned}$$

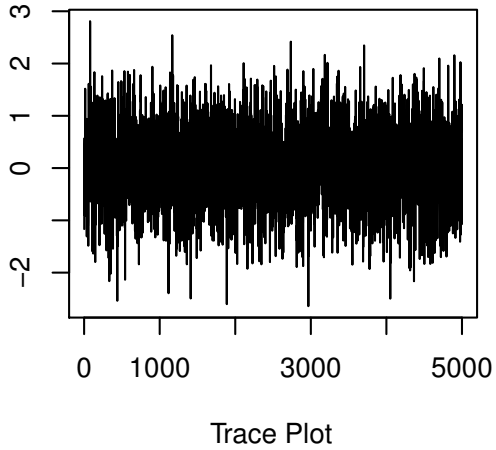
مثال: کد R

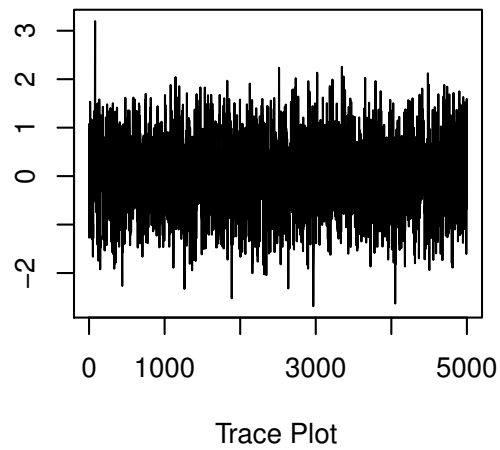
```
Nsim=5000 # initial values
rho=0.7
X=Y=array(0,dim=c(Nsim,1)) # init arrays
X[1]=rnorm(1) # init chains
Y[1]=rnorm(1)
for (i in 2:Nsim){ # sampling loop
     Y[i]=rnorm(1,rho*X[i-1],1-rho^2)
     X[i]=rnorm(1,rho*Y[i],1-rho^2)
}
mcmc=cbind(X,Y)
par(mfrow=c(2,2))
plot(ts(mcmc[,1]), xlab="Trace Plot", ylab="")
plot(ts(mcmc[,2]), xlab="Trace Plot", ylab="")
hist(mcmc[,1],40, main="", xlab="")
hist(mcmc[,2],40, main="", xlab="")
par(mfrow=c(1,1))
```

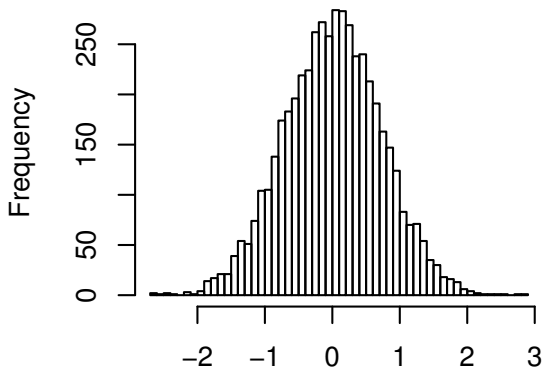
نمایش هندسی

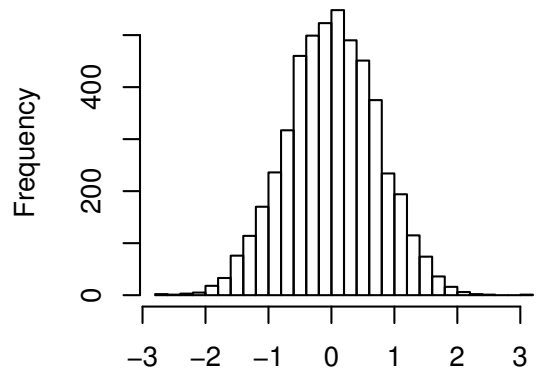


مثال: نمودارهای اثر









مثال: مدل نرمال

هدف: تولید نمونه از توزیع پسین (θ, σ^2) در مدل

$$X_i \sim N(\theta, \sigma^2),\ i = 1, \ldots, n$$

$$\theta \sim (\theta_0, \tau^2),\ \sigma^2 \sim IG(a, b)$$

می‌باشد، که در آن مقادیر θ_0، τ^2، a و b معلوم هستند. و داده‌های x مربوط به مطالعه متابولیسم در دختران ۱۵ ساله است که داده‌های آن میزان انرژی مصرف‌شده طی ۲۴ ساعت است:

```
> x=c(91,504,557,609,693,727,764,803,857,929,970,1043, 089,1195,1384,1713)
```

توزیع پسین به صورت زیر خواهد بود:

$$\pi(\theta, \sigma^2|x) \propto \frac{1}{(\sigma^2)^{n/2}} e^{-\sum_i (x_i-\theta)^2/2\sigma^2} \times \left[\frac{1}{\tau} e^{-(\theta-\theta_0)^2/2\tau^2}\right] \times \left[\frac{1}{(\sigma^2)^{a+1}} e^{1/b\sigma^2}\right]$$

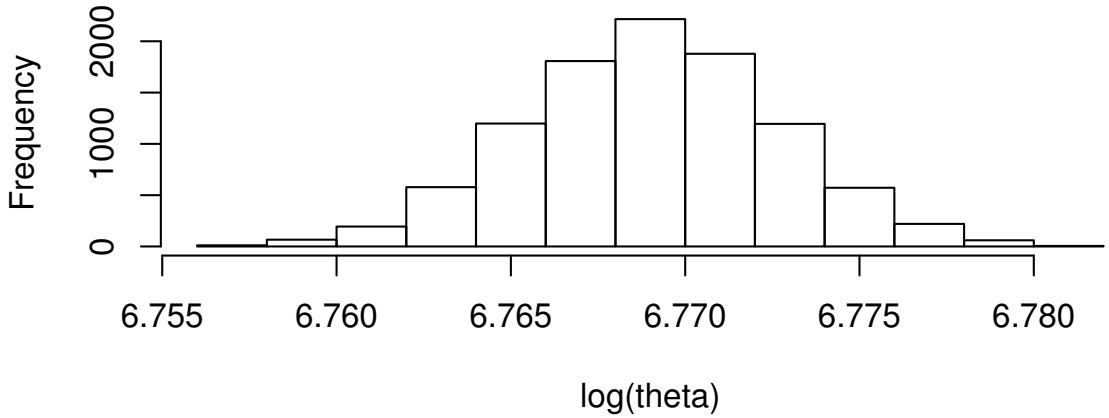
مثال: مدل نرمال

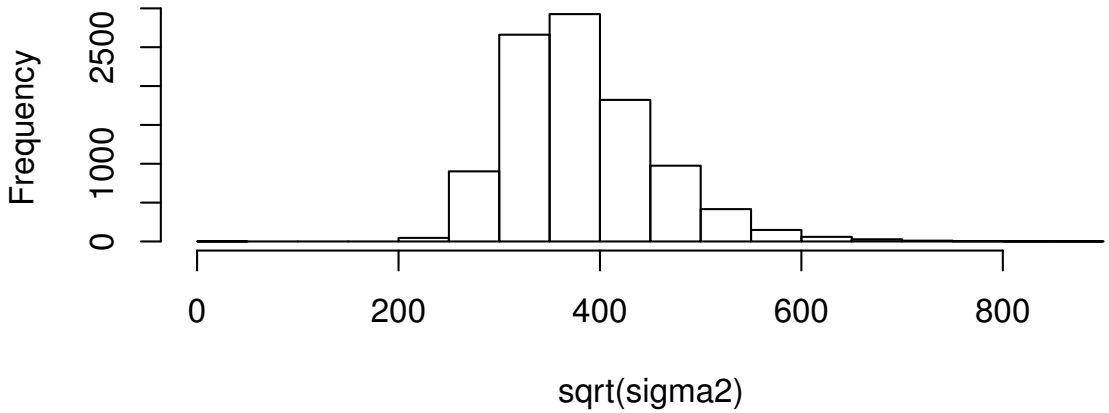
چگالی‌های شرطی کامل به صورت زیر قابل محاسبه‌اند (به عنوان تمرین نشان دهید):

$$\begin{aligned}
\theta|x,\sigma^2 &\sim N\left(\frac{\sigma^2}{\sigma^2+n\tau^2}\theta_0+\frac{n\tau^2}{\sigma^2+n\tau^2}\bar{x},\frac{\sigma^2\tau^2}{\sigma^2+n\tau^2}\right)\\
\sigma^2|x,\theta &\sim IG\left(\frac{n}{2}+a,\frac{1}{2}\sum_i(x_i-\theta)^2+b\right)
\end{aligned}$$

```
n=length(x)
Nsim=10000
a=b=0.1
tau2=10
theta0=xbar
xbar=mean(x)
sh1=(n/2)+a
sigma=theta=rep(0,Nsim) #init arrays
sigma[1]=1/rgamma(1,shape=a,rate=b) #init chains
B=sigma2[1]/(sigma2[1]+n*tau2)
theta[1]=rnorm(1,m=B*theta0+(1-B)*xbar,sd=sqrt(tau2*B))
for (i in 2:Nsim){
     B=sigma2[i-1]/(sigma2[i-1]+n*tau2)
     theta[i]=rnorm(1,m=B*theta0+(1-B)*xbar,sd=sqrt(tau2*B))
     ra1=(1/2)*(sum((x-theta[i])^2))+b
     sigma2[i]=1/rgamma(1,shape=sh1,rate=ra1)
}
##
> mean(theta)
[1] 870.459
> sqrt(mean(sigma2))
[1] 389.7115
##
par(mfrow=c(2,1))
hist(log(theta),main="")
hist(sqrt(sigma2),main="")
```

مثال: هیستوگرام





نمونه‌گیری گیبز چندمرحله‌ای

آنچه که بیان شد، معروف به الگوریتم نمونه‌گیر گیبز دومرحله‌ای است و به سادگی می‌توان آن را به حالت چندمرحله‌ای تعمیم داد.

فرض کنید $X = (X_1, \ldots, X_p)$ و فرض کنید بتوان از چگالی‌های شرطی کامل به سادگی نمونه تولید کرد:

$$X_i | x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_p \sim f(x_i | x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_p)$$

الگوریتم نمونه‌گیر گیبز چندمرحله‌ای به صورت زیر قابل نمایش است:

- فرض کنید در تکرار $t = 1, 2, \ldots$، بردار $x^{(t)} = (x_1^{(t)}, \ldots, x_p^{(t)})$ مفروض باشد. در این صورت مقادیر زیر را تولید کن
- $X_1^{(t+1)} \sim f_1(x_1 | x_2^{(t)}, \ldots, x_p^{(t)})$
- $X_2^{(t+1)} \sim f_2(x_2 | x_1^{(t+1)}, x_3^{(t)}, \ldots, x_p^{(t)})$
- $\vdots$
- $X_p^{(t+1)} \sim f_p(x_p | x_1^{(t+1)}, \ldots, x_{p-1}^{(t+1)})$