Introduction to Bayesian Statistics - 6

PhD Physics course (XXVIII ciclo)

Università di Trieste

Edoardo Milotti

Bayesian estimates often require the evaluation of complex integrals. Usually these integrals can only be evaluated with numerical methods.

⇨ enter the Monte Carlo methods!

1. acceptance-rejection sampling
2. importance sampling
3. statistical bootstrap
4. Bayesian methods in a sampling-resampling perspective
5. introduction to Markov chains and to the Metropolis algorithm
6. Markov Chain Monte Carlo (MCMC)

1. The acceptance rejection method

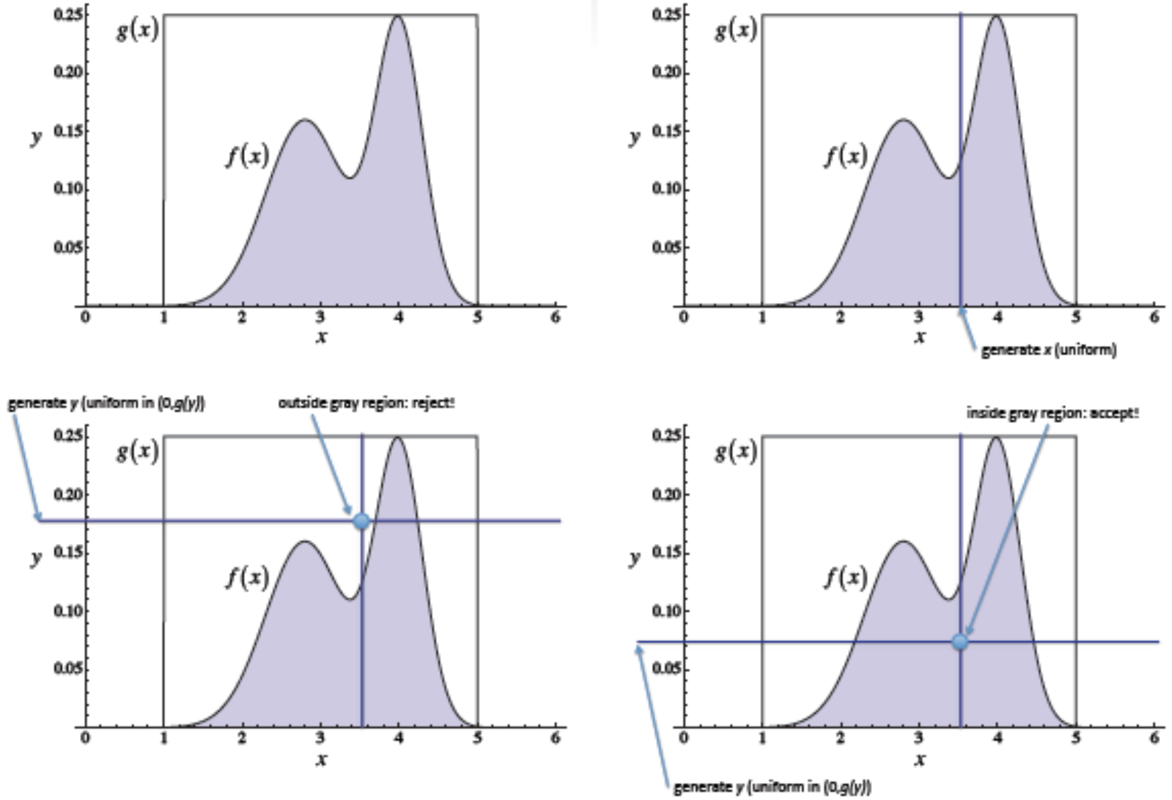
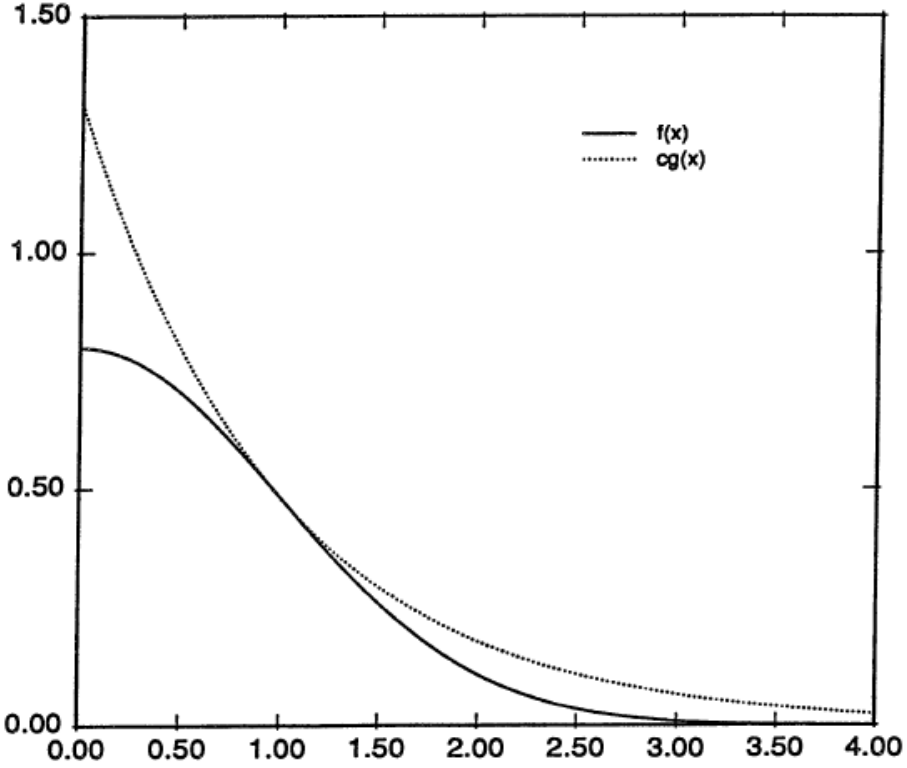


Figure: Schematic illustration of the acceptance-rejection method. Start from a pdf $f(x)$ defined in the interval (x_{min}, x_{max}), and take an enclosing pdf $g(x)$: notice that in general these are unnormalized pdf's. Now generate x according to the pdf $g(x)$ – in the case shown in the figure this means uniform in (x_{min}, x_{max}) – (upper right panel), and y, uniform in $(0, g(x))$. If $y > f(x)$, reject x (lower left panel), otherwise accept it (lower right panel). The accepted values x have pdf $\propto f(x)$.

Example: random numbers with semi-Gaussian distribution from exponentially distributed random numbers.

$$f(x) = \sqrt{\frac{2}{\pi}} \exp\left(-\frac{x^2}{2}\right) \qquad x \geq 0$$

$$g(x) = \exp(-x)$$



Definition of contact point (to maximize efficiency)

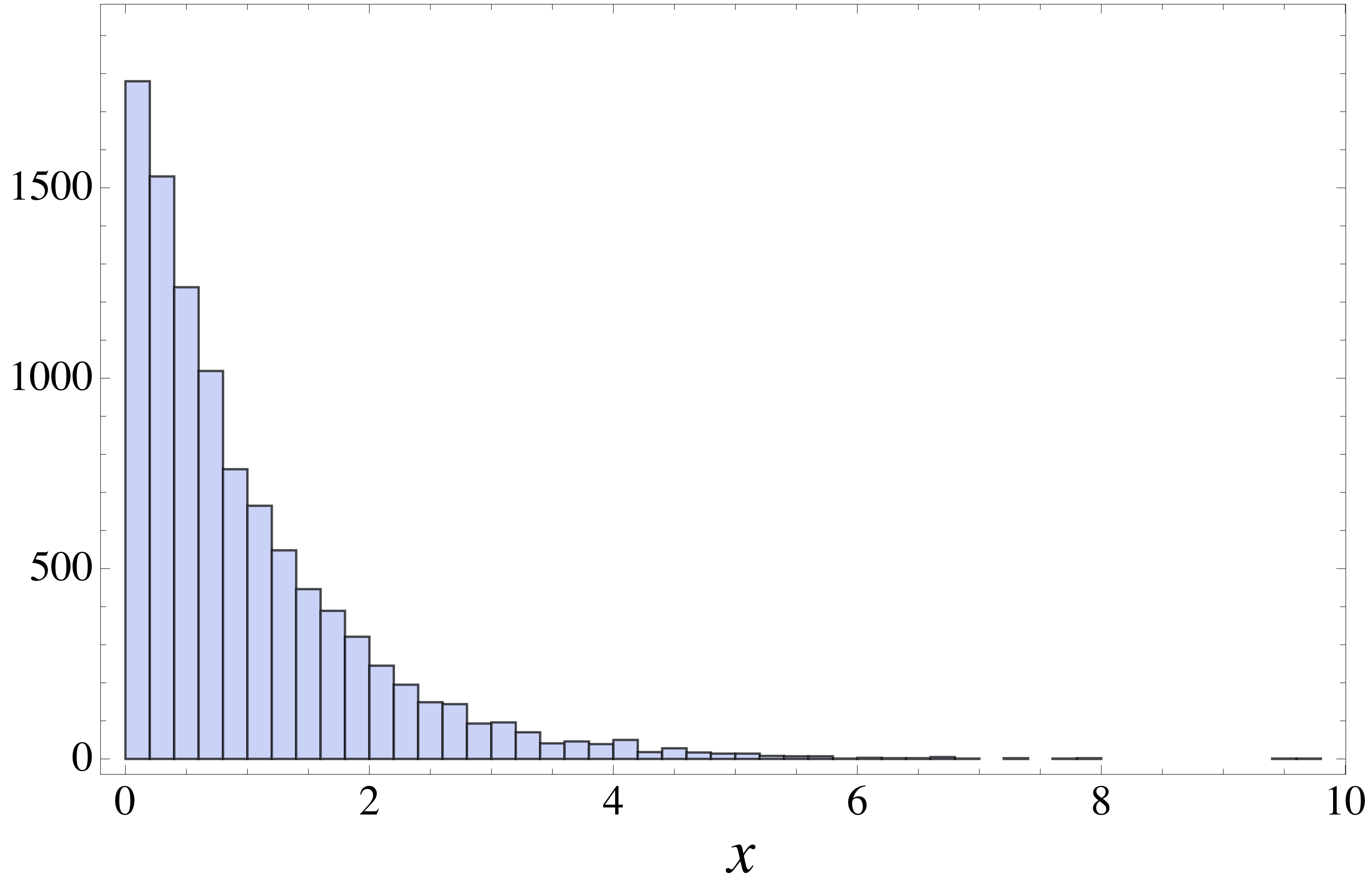
$$f(x) = \sqrt{\frac{2}{\pi}} \exp\left(-\frac{x^2}{2}\right) \qquad x \geq 0$$

$$g(x) = \exp(-x)$$

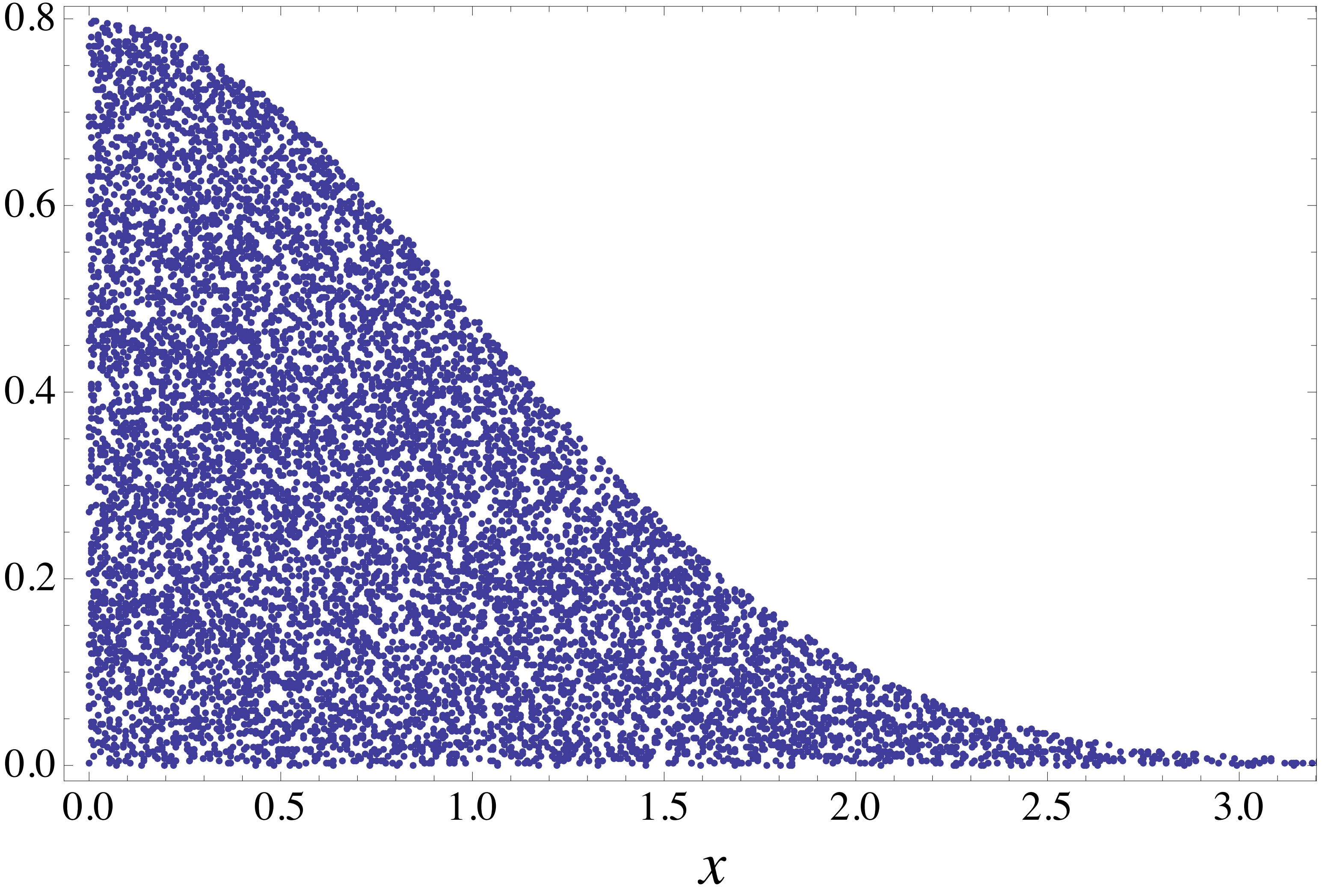
$$\Rightarrow \quad \begin{cases} f(x) = cg(x) \\ f'(x) = cg'(x) \end{cases} \quad \Rightarrow \quad \begin{cases} \sqrt{\frac{2}{\pi}} \exp\left(-\frac{x^2}{2}\right) = c\exp(-x) \\ x\sqrt{\frac{2}{\pi}} \exp\left(-\frac{x^2}{2}\right) = c\exp(-x) \end{cases}$$

$$\Rightarrow \quad x = 1; \quad c = \sqrt{\frac{2}{\pi}} \exp\left(-\frac{x^2}{2} + x\right) \approx 1.31549$$

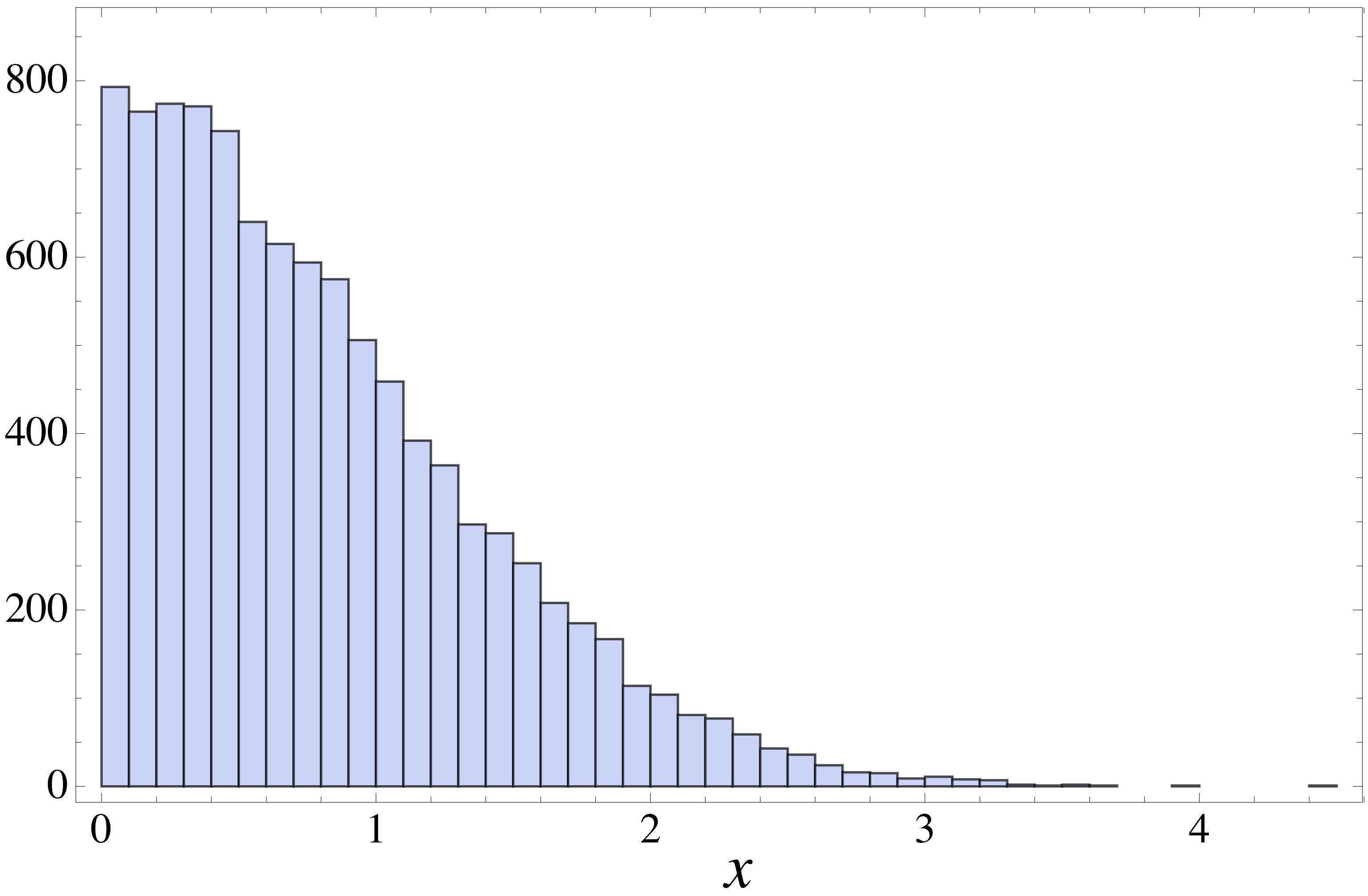
Exponentially distributed values



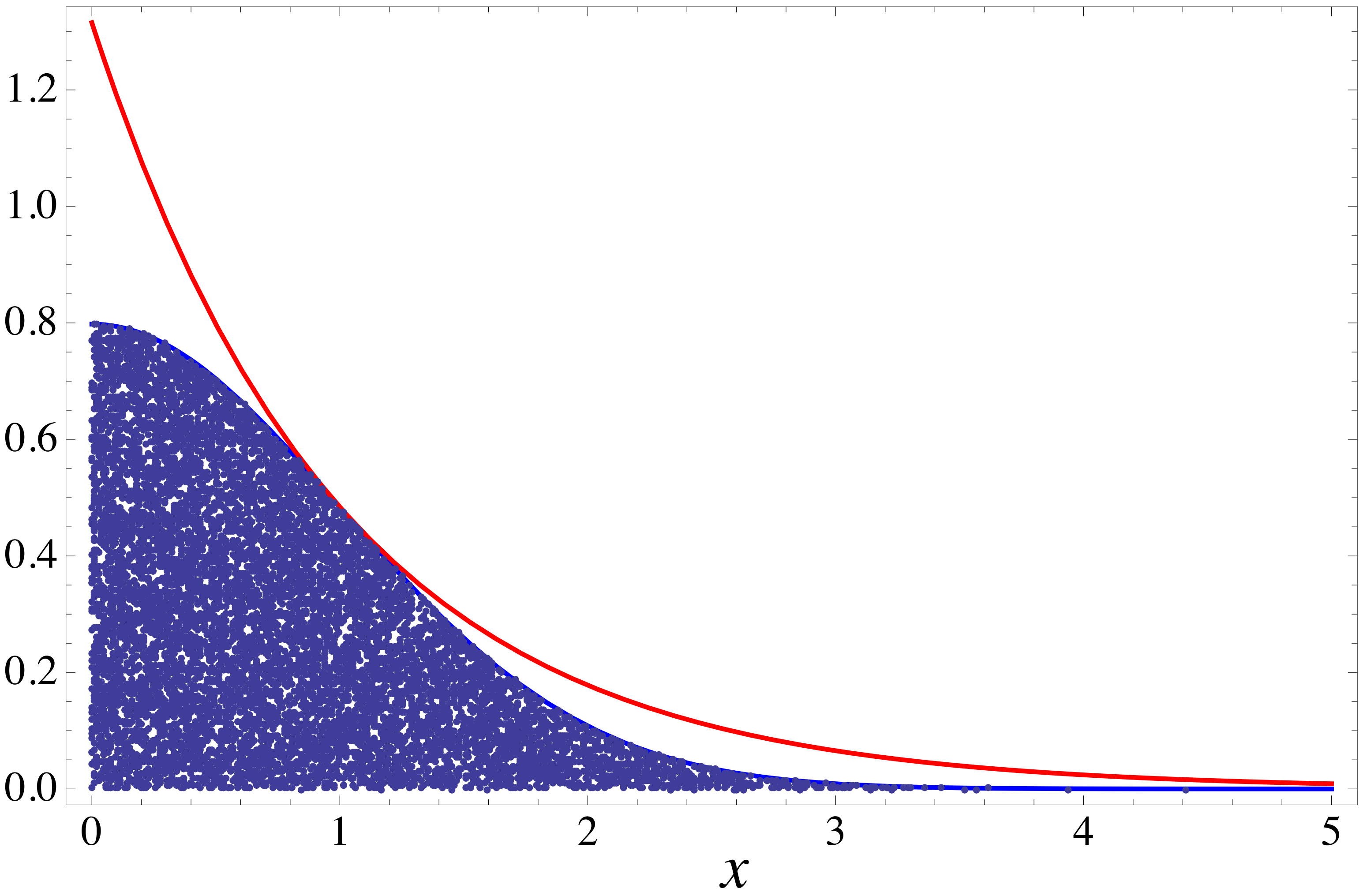
A/R accepted values (10000 accepted sample pairs)

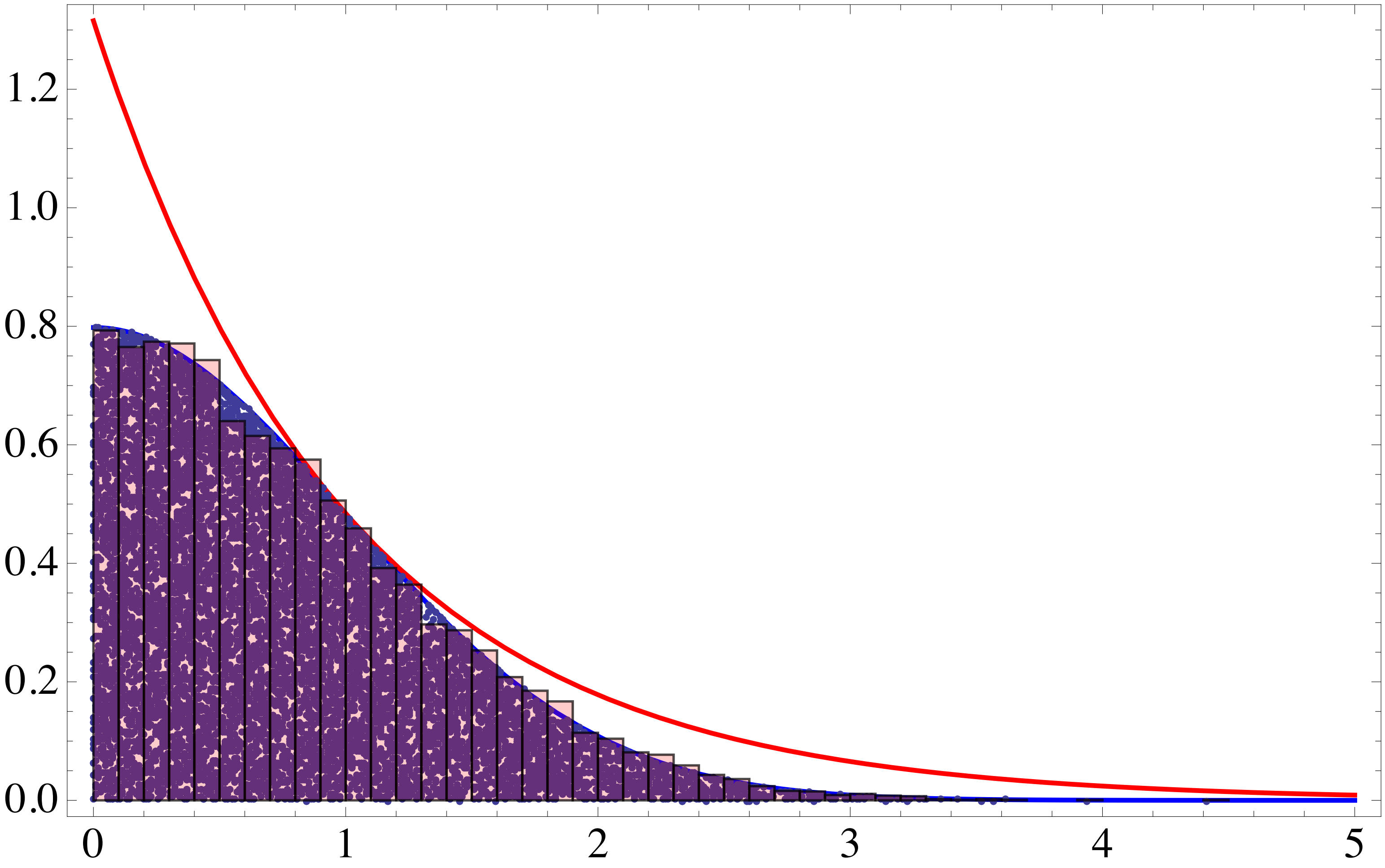


Histogram of accepted *x* values



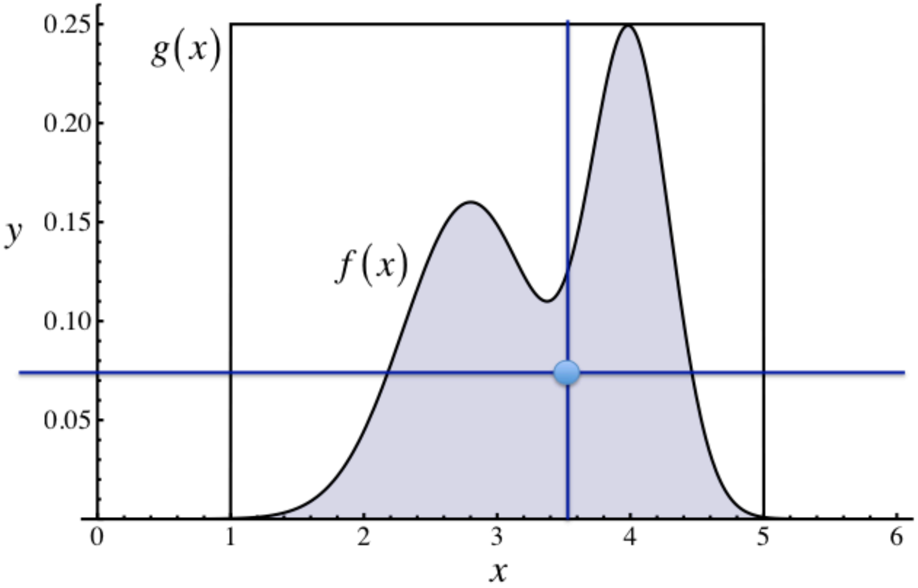
Comparison with the original distributions




1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
1
2
3
4
5

Now notice that in this method we generate pairs of real numbers (u,θ) that are uniformly distributed between $f(\theta)$ and the x-axis, therefore we can use these pairs to estimate the total area under the curve

(here the reference area is the area of the enclosing rectangle which corresponds to a uniform distribution)



$$\text{area} = \frac{\#\text{ of accepted pairs}}{\#\text{ of pairs}}\text{reference area}$$

In general, if $h(x) = f(x)p(x)$, where *p* is a pdf

$$\int_a^b h(x)dx = \int_a^b f(x)p(x)dx = E_p\left[f\left(\underset{\uparrow}{x}\right)\right] \approx \frac{1}{N}\sum_{n=1}^{N} f\left(\underset{\uparrow}{x_n}\right)$$

here the *x* are iid with pdf *p(x)*

and we find that the variance of this estimate of the integral is

$$\frac{1}{N}\left\{\frac{1}{N-1}\sum_{n=1}^{N}\left[f(x_n) - E_p\left[f(x)\right]\right]^2\right\}$$

We encounter a problem with this method when we must sample functions that have many narrow peaks.

2. Importance sampling

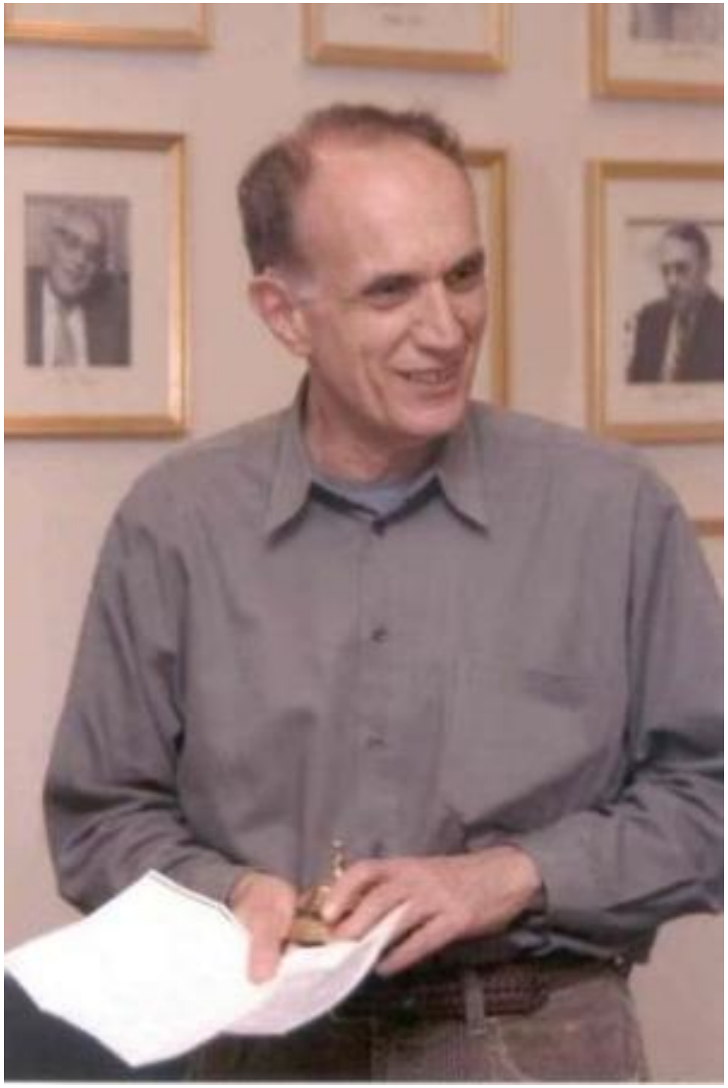
this pdf is troublesome ... therefore we use this ...

$$\int_a^b h(x)dx = \int_a^b f(x)p(x)dx = \int_a^b \left[f(x)\frac{p(x)}{q(x)} \right] q(x)dx$$

$$= E_q\left[f(x)\frac{p(x)}{q(x)} \right] \approx \frac{1}{N}\sum_{n=1}^{N} f(x_n)\frac{p(x_n)}{q(x_n)}$$

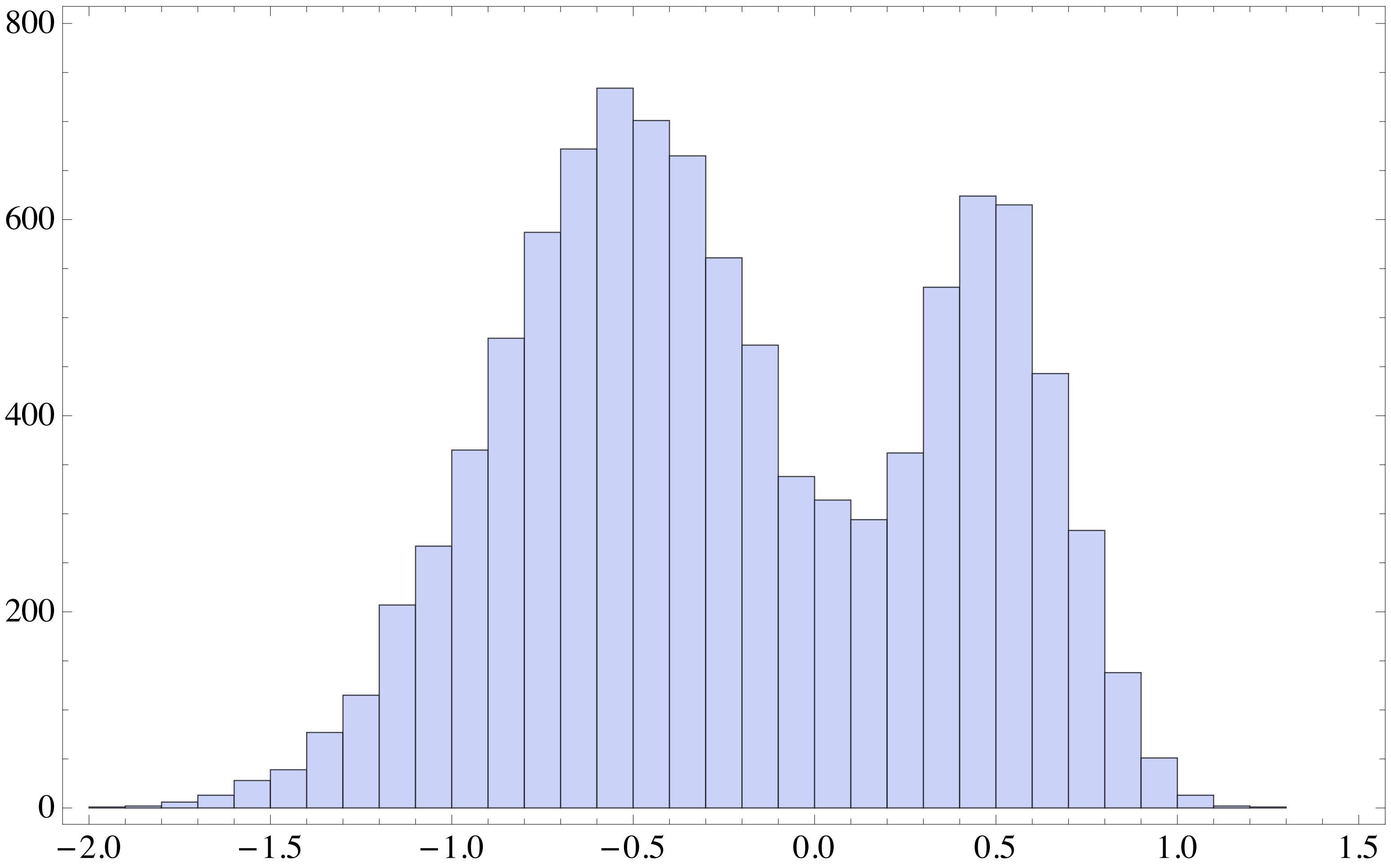
These methods are not very efficient and there is a better alternative, the Markov Chain Monte Carlo method

3. Bootstrap (B. Efron, 1977)

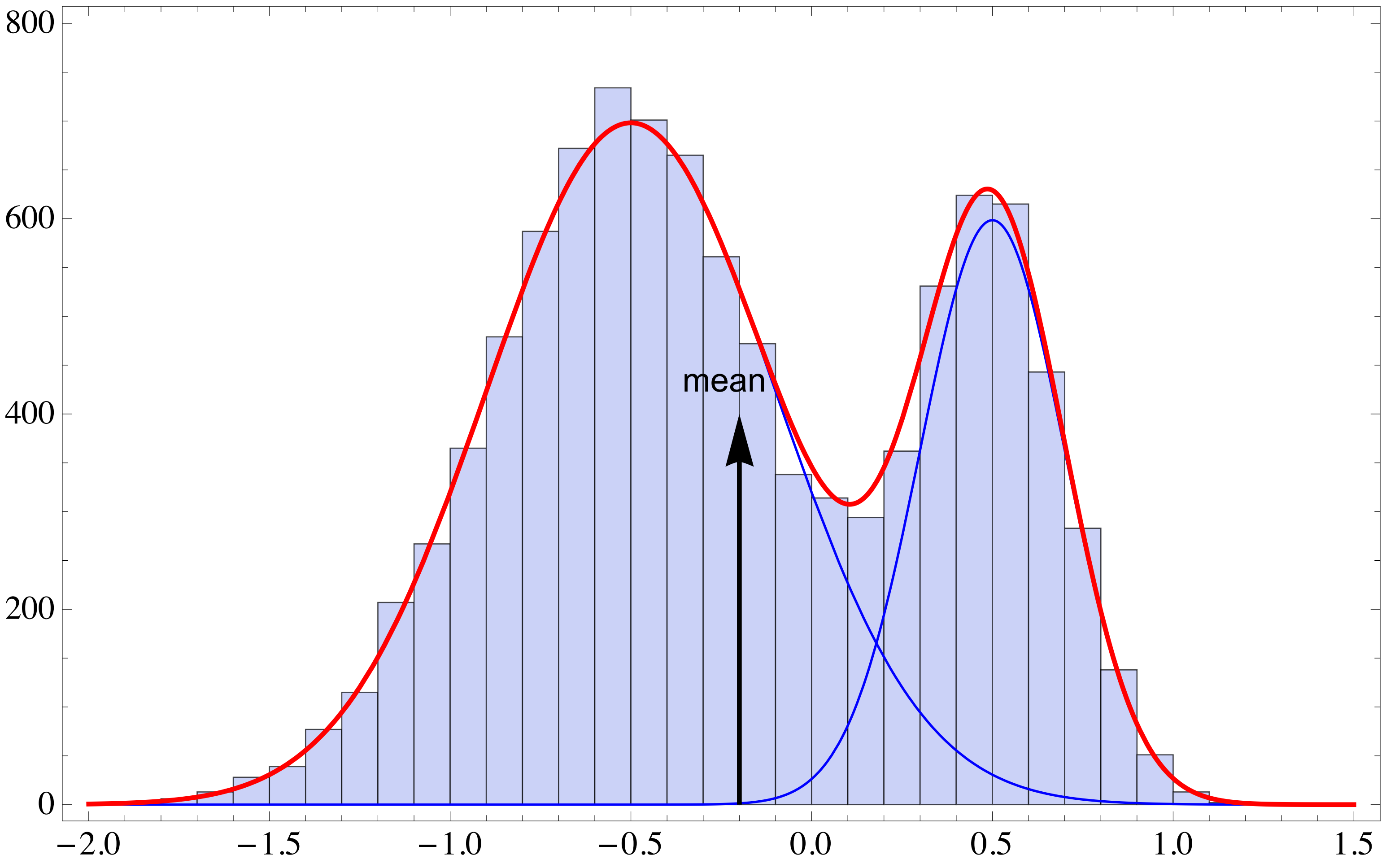


The bootstrap method is a resampling technique that helps calculate many statistical estimators

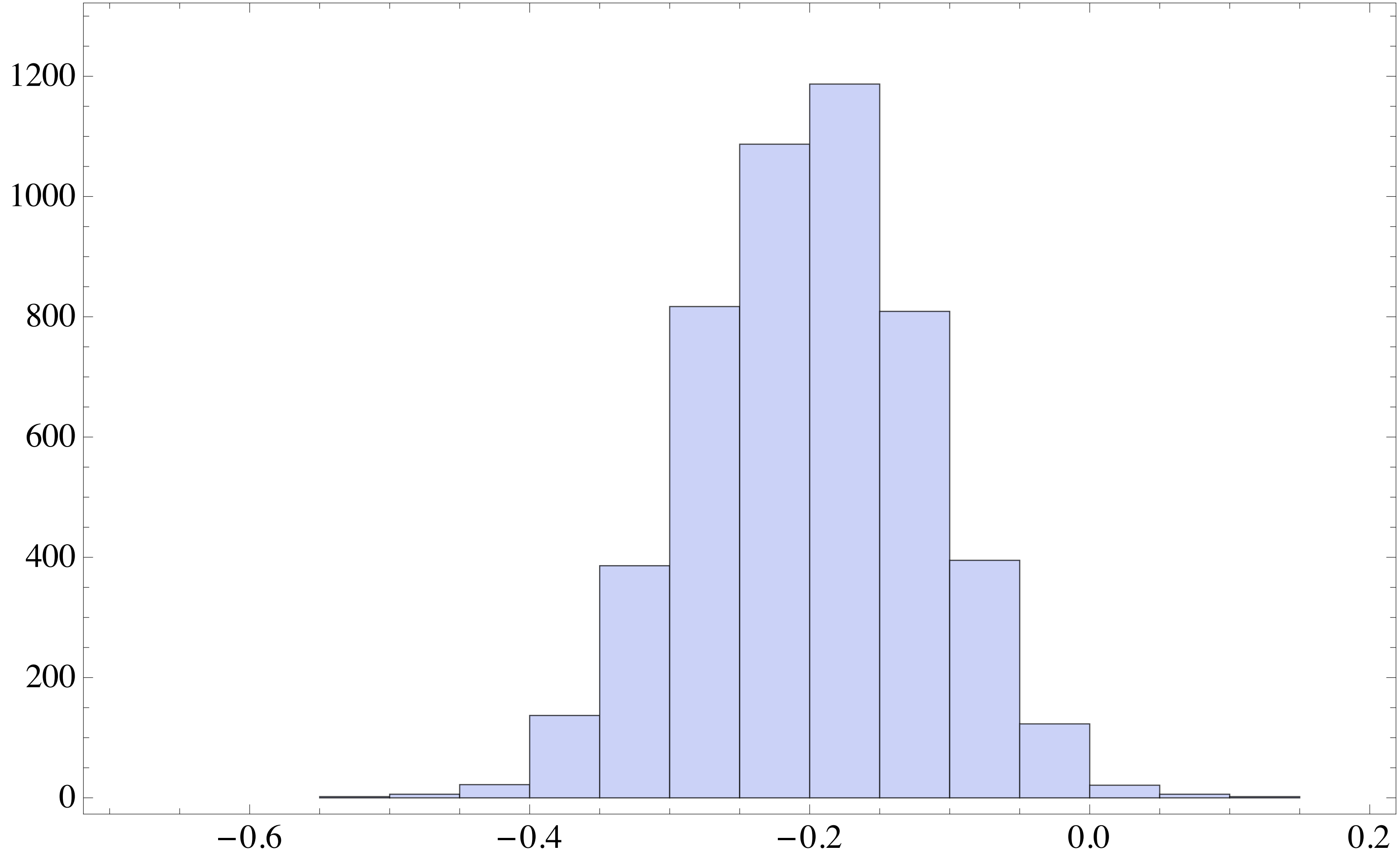
consider the distribution of a set of measurements



the distribution of data is an approximation of the “true” underlying distribution (in this case a mixture model)

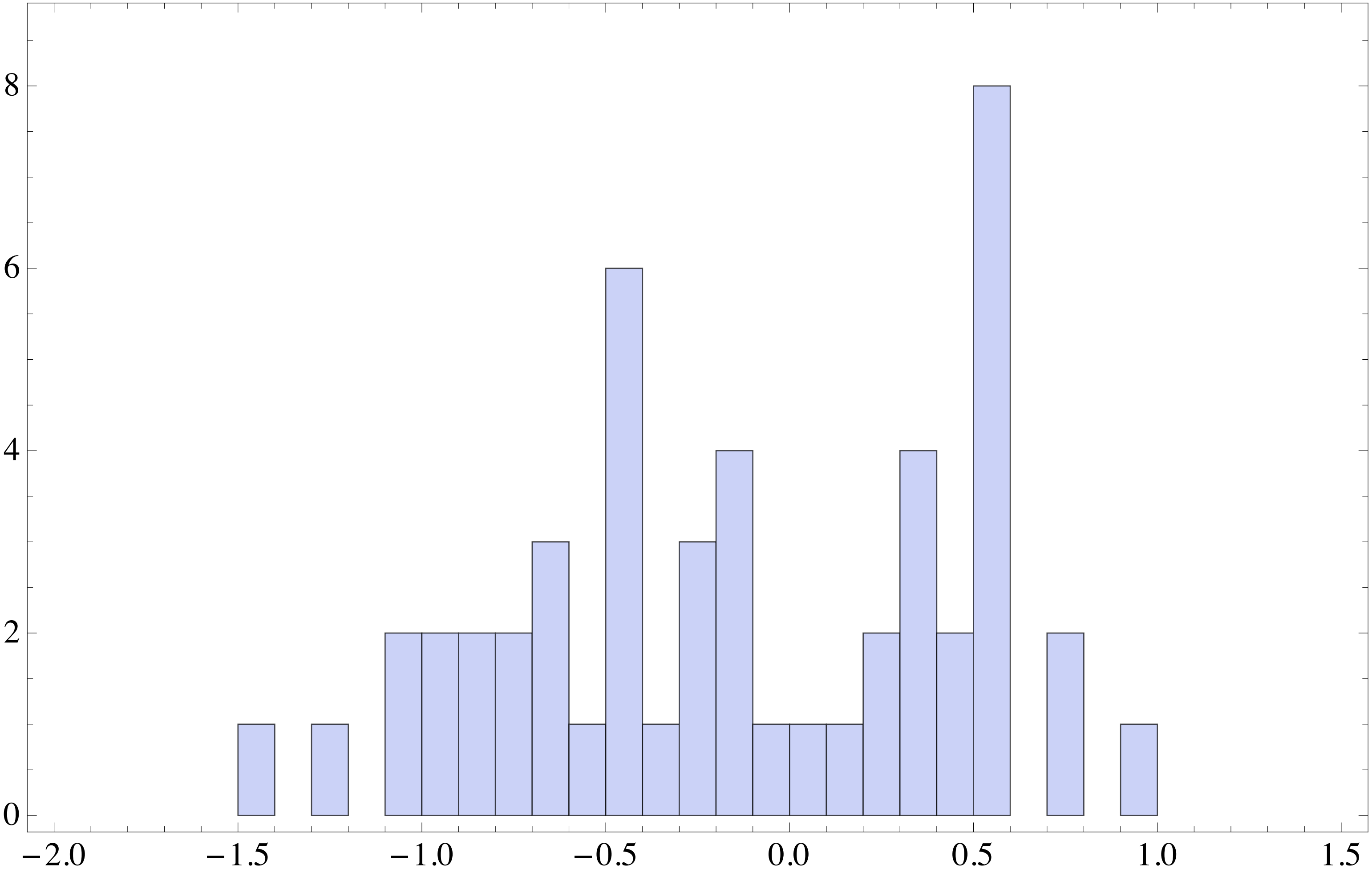


distribution of mean value obtained from 5000 sets of data (sample size = 50)



You can do this if you have large datasets ... but what if you have only a handful of measurements?

example: single dataset (same size as before, 50 measurements)



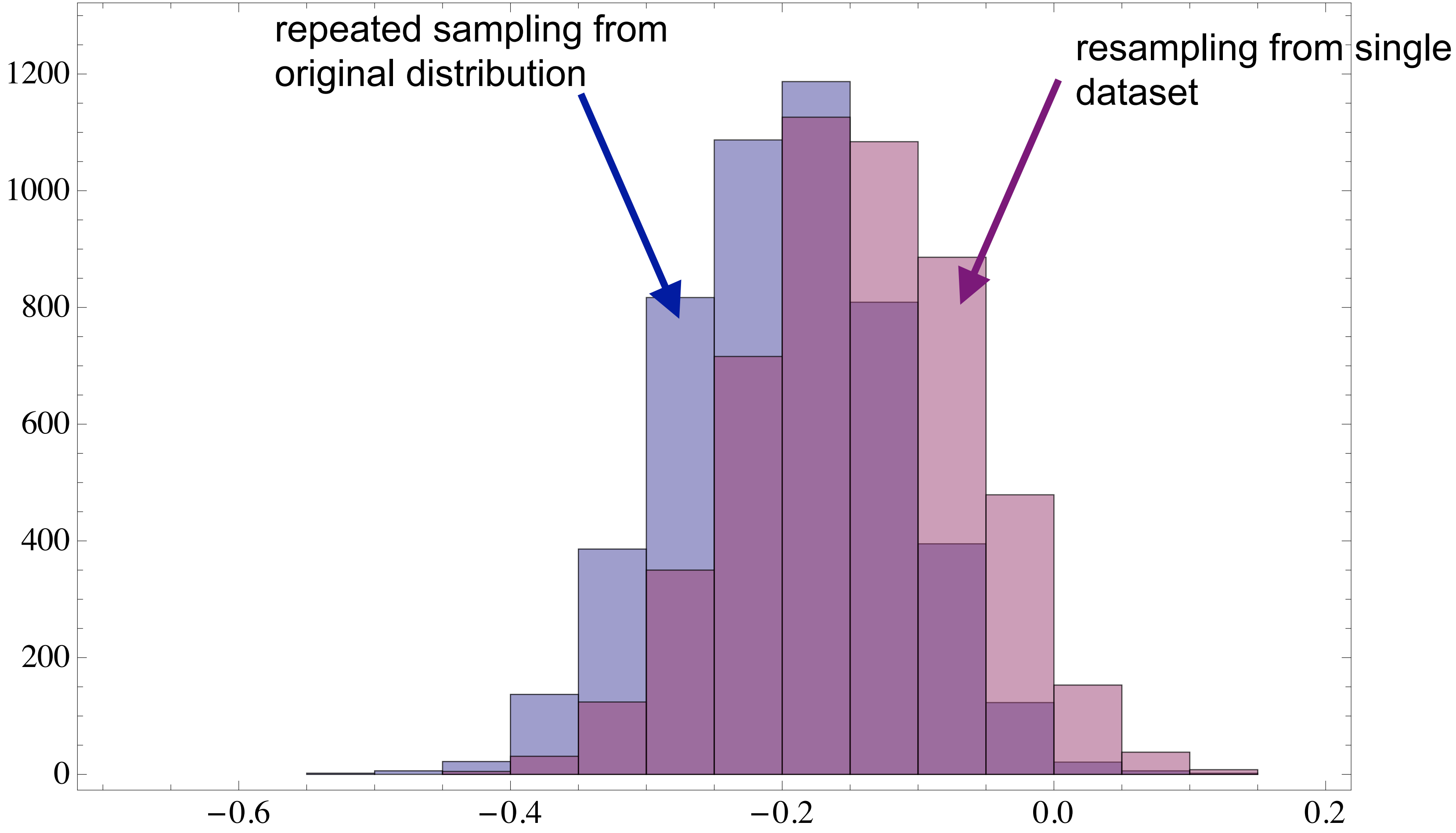
the distribution is a rough representation of the underlying distribution ... and yet it can be used as before ...

Bootstrap recipe:

if you want to find the distribution of the mean (or any other statistical estimator) use the dataset itself to generate new datasets

⇨ resample from dataset (with replacement)

distribution of mean value



true mean: -0.2

mean from repeated sampling (size = 250000): -0.200222 ± 0.0813632

mean from resampling dataset (size = 50): -0.142699 ± 0.0838678 counts of CD4 limphocytes

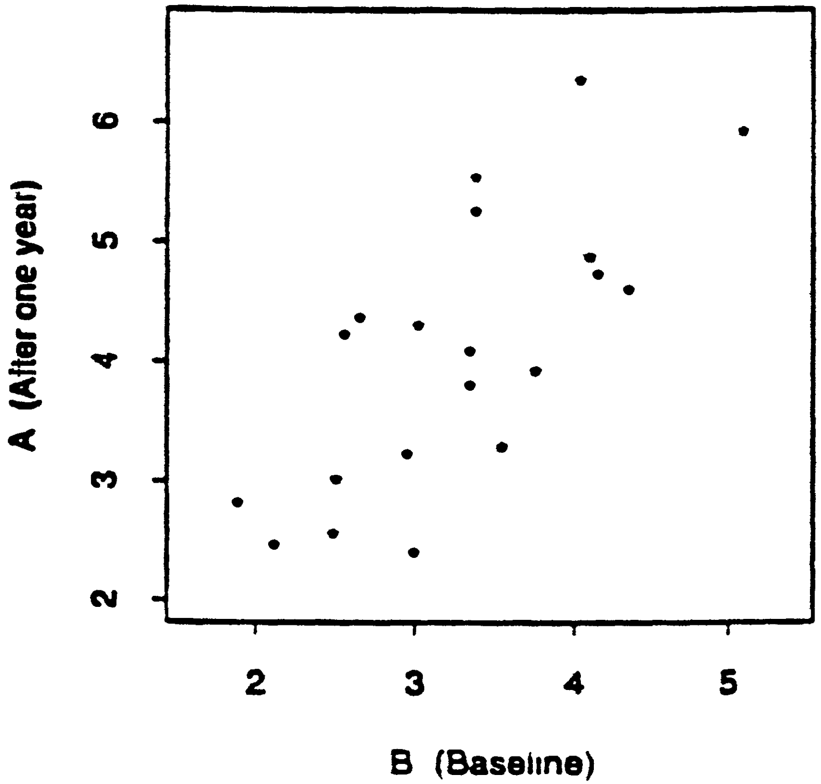


FIG. 1. *The cd4 data; cd4 counts in hundreds for 20 subjects, at baseline and after one year of treatment with an experimental anti-viral drug; numerical values appear in Table 1.*

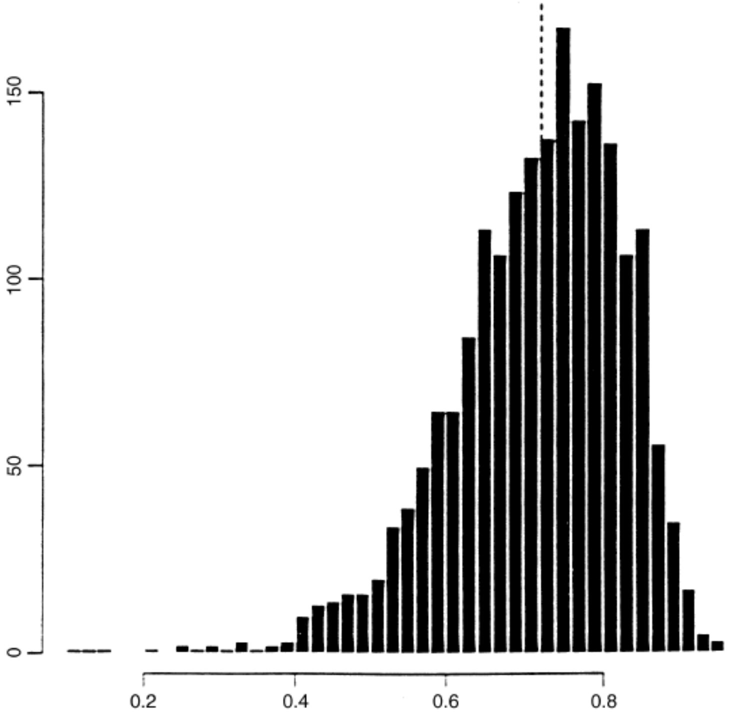
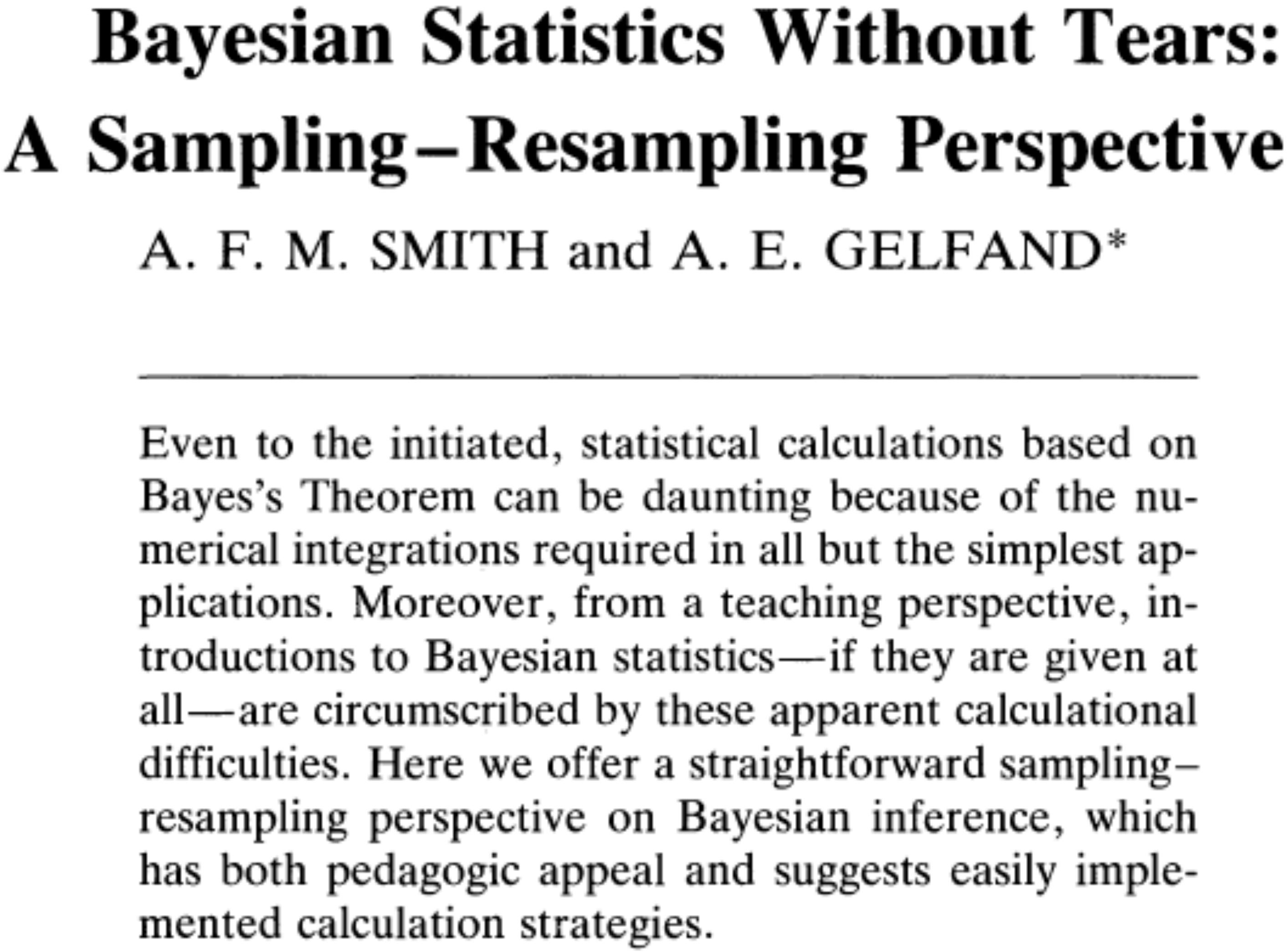


FIG. 3. *Histogram of 2,000 bootstrap correlation coefficients; bivariate normal sampling model.*

bootstrap estimate of correlation coefficient distribution

Example from Di Ciccio & Efron, Statistics of Science **11** (1996) 189 and Efron, Statistics of Science **13** (1998) 95

4. Bayesian methods in a sampling-resampling perspective (Smith & Gelfand, 1992)



Bayesian Statistics Without Tears: A Sampling–Resampling Perspective

A. F. M. SMITH and A. E. GELFAND*

Even to the initiated, statistical calculations based on Bayes's Theorem can be daunting because of the numerical integrations required in all but the simplest applications. Moreover, from a teaching perspective, introductions to Bayesian statistics—if they are given at all—are circumscribed by these apparent calculational difficulties. Here we offer a straightforward sampling–resampling perspective on Bayesian inference, which has both pedagogic appeal and suggests easily implemented calculation strategies.

In Bayesian methods we have to evaluate many integrals, like, e.g.,

$$p(\theta|x) = \frac{l(\theta; x)p(\theta)}{\int l(\theta; x)p(\theta)\, d\theta} \longleftarrow$$

normalization (evidence)

$$p(\phi|x) = \int p(\phi, \psi|x)\, d\psi. \longleftarrow$$

marginalization

$$E[m(\theta)|x] = \int m(\theta)p(\theta|x)\, d\theta \longleftarrow$$

averages (statistical estimators)

except in simple cases, explicit evaluation of such integrals will rarely be possible, and realistic choices of likelihood and prior will necessitate the use of sophisticated numerical integration or analytic approximation techniques (see, for example, Smith et al. 1985, 1987; Tierney and Kadane, 1986). This can pose problems for the applied practitioner seeking routine, easily implemented procedures. For the student, who may already be puzzled and discomforted by the intrusion of too much calculus into what ought surely to be a simple, intuitive, statistical learning process, this can be totally off-putting.

Bayesian learning as a resampling procedure

$$p\left(\theta \middle| x\right) = \frac{l\left(\theta; x\right)}{\int l\left(\theta; x\right) p\left(\theta\right) d\theta}\, p\left(\theta\right)$$

1. prior distribution defined by initial samples

2. Bayes factor distorts the distribution of initial samples

3. posterior distribution corresponds to a resampling of initial samples

So, how do we resample?

- acceptance-rejection
- bootstrap
- weighted-bootstrap

modified acceptance-rejection to resample prior samples with probability

$$f(\theta) \propto l(\theta;x)\,p(\theta)$$

and with $\hat{\theta}$ the MAP estimator, so that

$$\hat{\theta} = \arg\max_{\theta} f(\theta) = \arg\max_{\theta} l(\theta;x)\,p(\theta); \quad M = l\left(\hat{\theta};x\right)p\left(\hat{\theta}\right)$$

and then we resample from a population $\left\{\theta_i\right\}$, accepting θ_i with probability

$$\frac{f(\theta)}{Mp(\theta)} = \frac{l(\theta;x)}{l\left(\hat{\theta};x\right)}$$

Standard bootstrap

consider samples θ_i extracted from a distribution with PDF $g(\theta)$ then

$$\Pr(\theta \le a) = \int_{-\infty}^{a} g(\theta)\,d\theta = \int_{-\infty}^{+\infty} 1_{(-\infty,a)}\, g(\theta)\,d\theta = \mathbf{E}_g\left(1_{(-\infty,a)}\right)$$

$$\approx \sum_{i=1}^{n} \frac{1}{n} 1_{(-\infty,a)}\left(\theta_i\right) = \frac{1}{n} \sum_{i=1}^{n} 1_{(-\infty,a)}\left(\theta_i\right)$$

(the samples approximate the “true” underlying distribution)

Weighted bootstrap

if the samples θ_i are distributed according to $g(\theta)$ BUT we have a target distribution

$$h(\theta) = \frac{f(\theta)}{\int_{-\infty}^{+\infty} f(\theta)\,d\theta}$$

this normalization factor is unknown

then for each sample compute the weights $w_i = \frac{f(\theta_i)/g(\theta_i)}{\sum_j f(\theta_j)/g(\theta_j)}$

weights are self-normalized, no need of the unknown integral

resampling with probability *w*$_i$ yields the distribution *h*

Indeed

here we do not need knowledge of the target distribution

samples extracted from g

$$\Pr(\theta \le a) \approx \sum_{i=1}^{n} w_i 1_{(-\infty,a)}(\theta_i)$$

$$= \frac{\frac{1}{n}\sum_{i=1}^{n}\frac{f(\theta_i)}{g(\theta_i)}1_{(-\infty,a)}(\theta_i)}{\frac{1}{n}\sum_{i=1}^{n}\frac{f(\theta_i)}{g(\theta_i)}} \to \frac{E_g\left(\frac{f(\theta)}{g(\theta)}1_{(-\infty,a)}\right)}{E_g\left(\frac{f(\theta)}{g(\theta)}\right)}$$

$$= \frac{\int_{-\infty}^{a}\frac{f(\theta)}{g(\theta)}g(\theta)d\theta}{\int_{-\infty}^{+\infty}\frac{f(\theta)}{g(\theta)}g(\theta)d\theta} = \frac{\int_{-\infty}^{a}f(\theta)d\theta}{\int_{-\infty}^{+\infty}f(\theta)d\theta} = \int_{-\infty}^{a}h(\theta)d\theta$$

eventually we get *h* without the need to compute normalization integrals

Example (McCullagh & Nelder): take two sets of binomially distributed independent random variables X_{i1} and X_{i2} (i=1,2,3)

$$X_{i1} = \text{Binomial}\left(n_{i1}, \theta_1\right)$$
$$X_{i2} = \text{Binomial}\left(n_{i2}, \theta_2\right)$$

The observed random variables are

$$Y_i = X_{i1} + X_{i2}$$

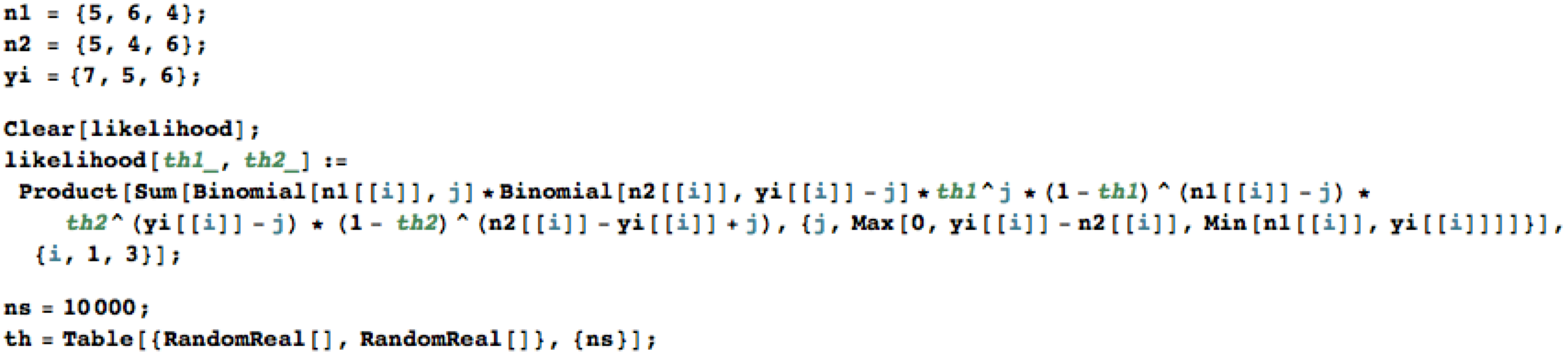
$$\text{likelihood} = \prod_{i=1}^{3} \sum_{j_i} \binom{n_{i1}}{j_i} \binom{n_{i2}}{y_i - j_i} \theta_1^{j_i} \left(1-\theta_1\right)^{n_{i1}-j_i} \theta_2^{y_1-} \left(1-\theta_2\right)^{n_{i1}-y_1+j_i}$$

$$\max\left(0, y_i - n_{i2}\right) \le j_i \le \min\left(n_{i1}, y_i\right)$$

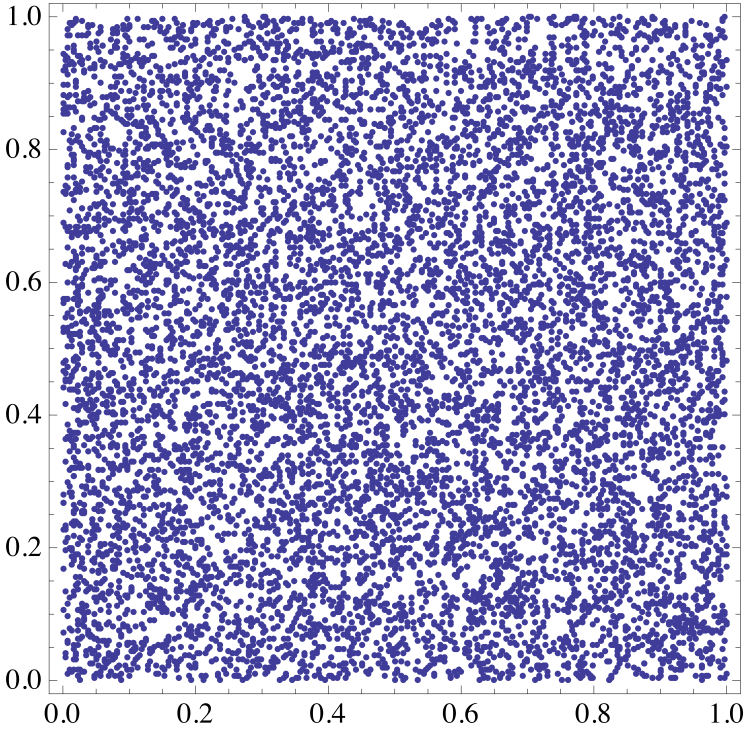
Sample data

	1	**2**	**3**
n_{i1}	5	6	4
n_{i2}	5	4	6
y_i	7	5	6

Example of implementation in *Mathematica*

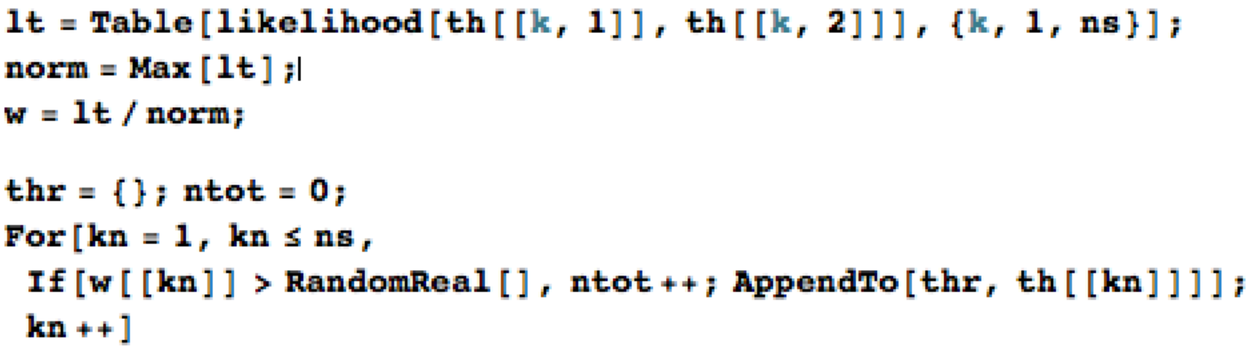


```
n1 = {5, 6, 4};
n2 = {5, 4, 6};
yi = {7, 5, 6};

Clear[likelihood];
likelihood[th1_, th2_] :=
 Product[Sum[Binomial[n1[[i]], j] * Binomial[n2[[i]], yi[[i]] - j] * th1^j * (1 - th1) ^ (n1[[i]] - j) *
    th2^ (yi[[i]] - j) * (1 - th2) ^ (n2[[i]] - yi[[i]] + j), {j, Max[0, yi[[i]] - n2[[i]]], Min[n1[[i]], yi[[i]]]}],
  {i, 1, 3}];

ns = 10000;
th = Table[{RandomReal[], RandomReal[]}, {ns}];
```

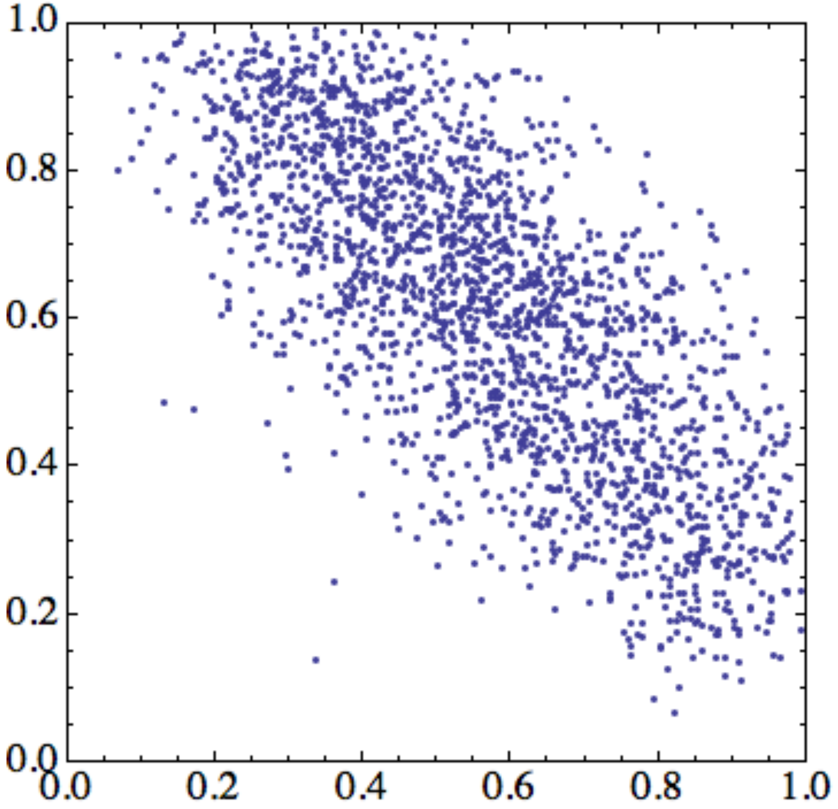


prior distribution (uniform in 2D parameter space)

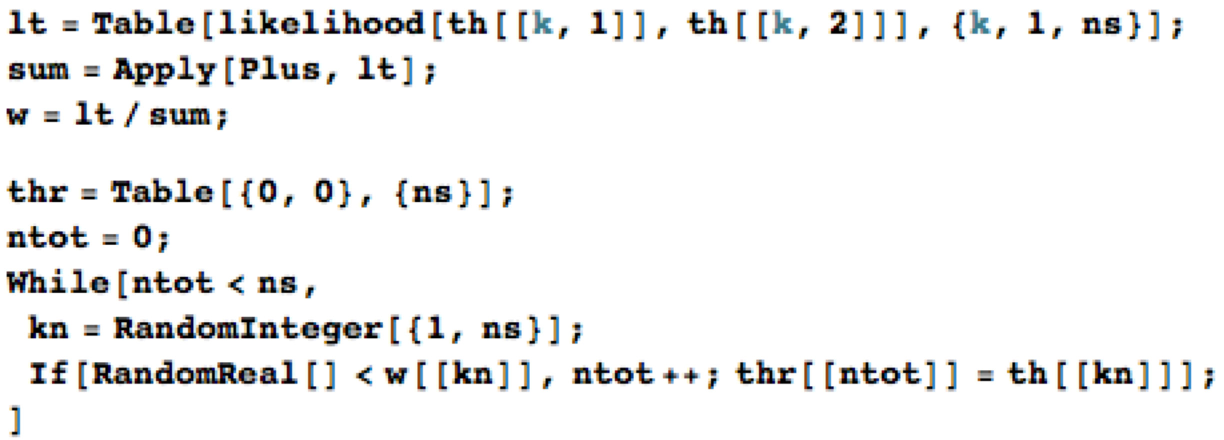
Posterior as a resampled prior using acceptance-rejection



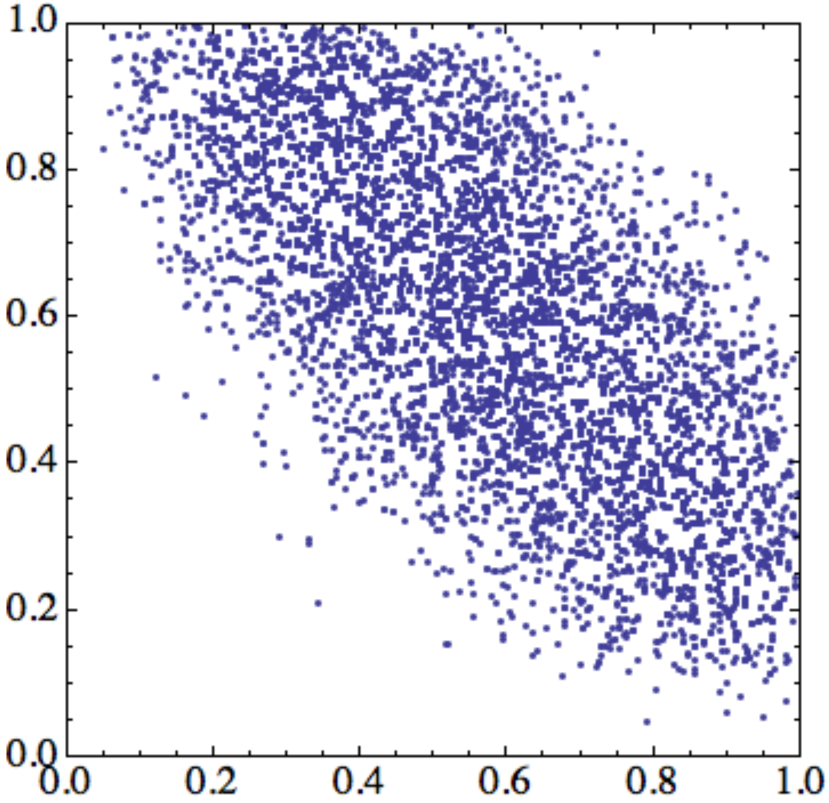
```
lt = Table[likelihood[th[[k, 1]], th[[k, 2]]], {k, 1, ns}];
norm = Max[lt];
w = lt / norm;

thr = {}; ntot = 0;
For[kn = 1, kn ≤ ns,
 If[w[[kn]] > RandomReal[], ntot++; AppendTo[thr, th[[kn]]]];
 kn++]
```



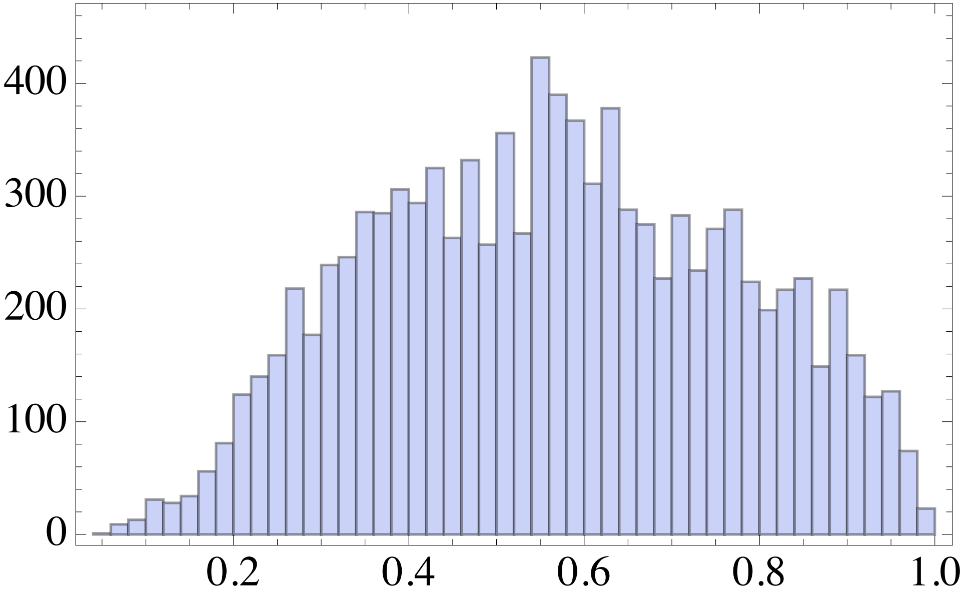
Posterior as a resampled prior using weighted bootstrap



```
lt = Table[likelihood[th[[k, 1]], th[[k, 2]]], {k, 1, ns}];
sum = Apply[Plus, lt];
w = lt / sum;

thr = Table[{0, 0}, {ns}];
ntot = 0;
While[ntot < ns,
 kn = RandomInteger[{1, ns}];
 If[RandomReal[] < w[[kn]], ntot ++; thr[[ntot]] = th[[kn]]];
]
```

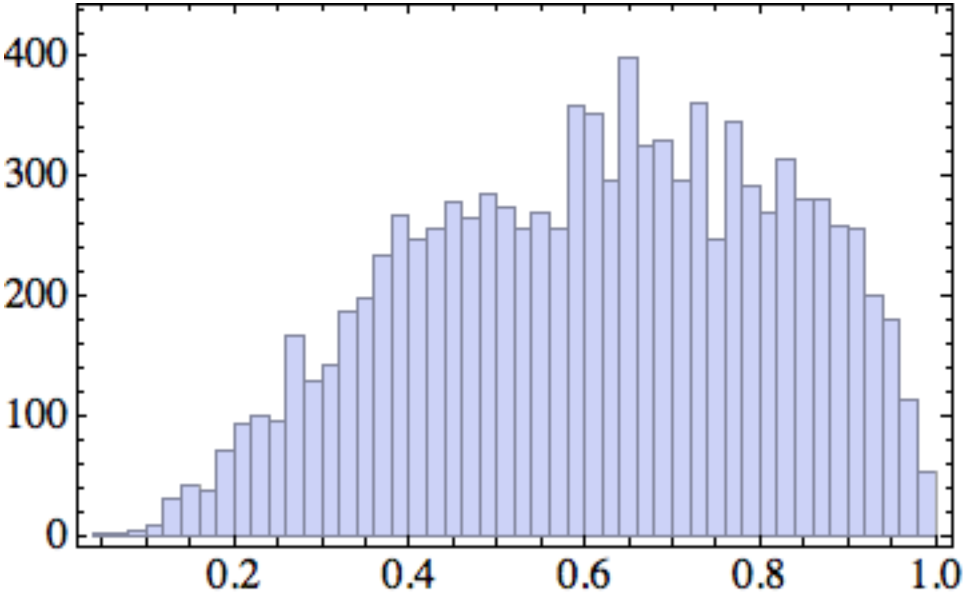


The resampled points are representative of the posterior distribution and can be used to evaluate any sample estimate



Marginalized distribution of θ_1

Sample mean: 0.564±0.002



Marginalized distribution of θ_2

Sample mean: 0.613±0.002

5. Very short introduction to Markov chains

Consider a system such that:

- the system can occupy a finite or countably infinite set of states S_n;
- the system changes state randomly at discrete times t = 1, 2, . . . ;
- if the system is in state S_i, then the probability that the system goes into state S_j is

$$P\left(S_{t+1} = s_j \middle| S_t = s_k, \ldots, S_{t-n} = s_l, \ldots\right)$$

  i.e., this probability depends only on the previous state, and is independent of all previous states (**this is the Markov property**);
- the transition probabilities p_{ij} do not depend on time n.
- **Such a system is a special type of discrete time stochastic process, which is called Markov chain.**

In general we should have

$$P\left(S_{t+1} = s_j \middle| S_t = s_k, \ldots, S_{t-n} = s_l, \ldots\right)$$

however the Markov property tells us that only the previous state is important in determining the next state

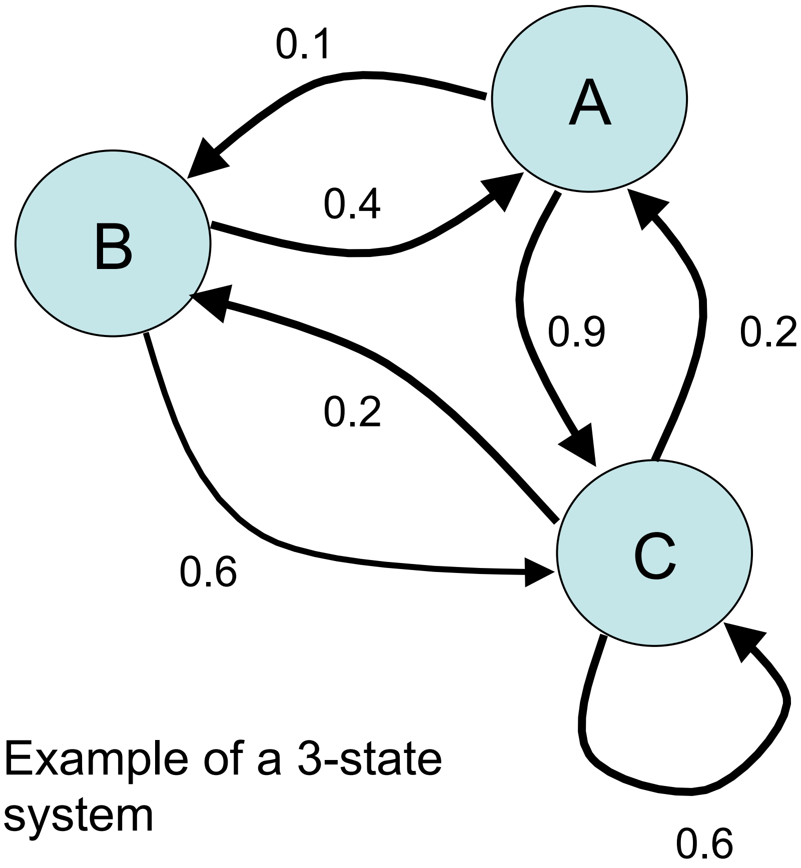
$$P\left(S_{t+1} = s_j \middle| S_t = s_k, \ldots, S_{t-n} = s_l, \ldots\right) = P\left(S_{t+1} = s_j \middle| S_t = s_k\right) = P(k \to j)$$

Now let $\pi_j(t)$ be the probability that the system is in state *j* at time *t*:

$$\pi_j(t+1) = \sum_k P(k \to j)\pi_k(t)$$

$$= \sum_k T_{kj}\pi_k(t)$$

$$\boldsymbol{\pi}(t+1) = \boldsymbol{\pi}(t)\mathbf{T}$$



Example of a 3-state system

$$\mathbf{T} = \begin{pmatrix} 0 & 0.1 & 0.9 \\ 0.4 & 0 & 0.6 \\ 0.2 & 0.2 & 0.6 \end{pmatrix}$$

transition matrix; it belongs to the class of "stochastic matrices" (rows add to 1)

The following equation also holds

$$\boldsymbol{\pi}(t+1)=\boldsymbol{\pi}(t)\mathbf{T}=\boldsymbol{\pi}(t-1)\mathbf{T}^2$$

and more generally

$$\boldsymbol{\pi}(t+k)=\boldsymbol{\pi}(t)\mathbf{T}^k$$

Moreover we find

$$\boldsymbol{\pi}(t+k+m)=\boldsymbol{\pi}(t+k)\mathbf{T}^m=\boldsymbol{\pi}(t)\mathbf{T}^k\mathbf{T}^m=\boldsymbol{\pi}(t)\mathbf{T}^{k+m}$$

and therefore

discrete version of the Chapman-Kolmogorov eq.

$$\mathbf{T}^{k+m}=\mathbf{T}^k\mathbf{T}^m \quad \Rightarrow \quad \boxed{T_{ij}^{(k+m)}=\sum_n T_{in}^{(k)}T_{nj}^{(m)}}$$

It can be shown that Markov chains have a stationary distribution

$$\boldsymbol{\pi}^* = \boldsymbol{\pi}^* \mathbf{T}$$

such that the *detailed balance* also holds

$$\pi_i P(i \to k) = \pi_k P(k \to i) \quad \text{i.e.} \quad \pi_i T_{ik} = \pi_k T_{ki}$$

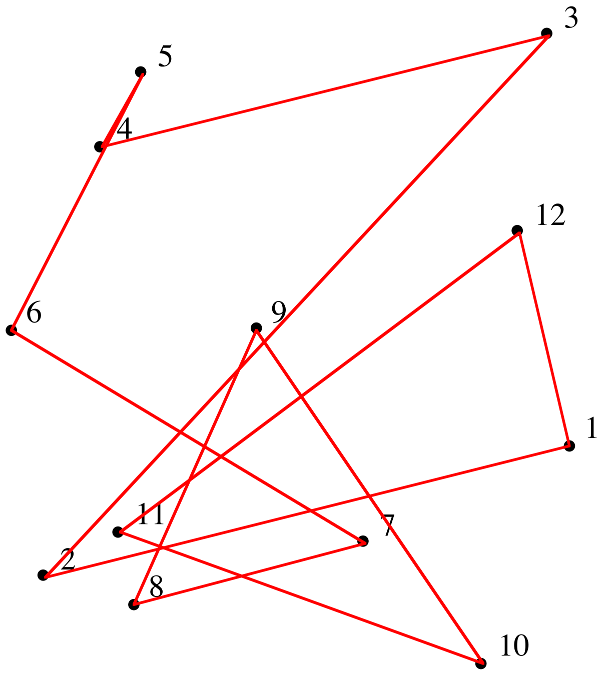
Indeed we see that

$$\pi_i(t+1) = \sum_k T_{ki}\pi_k(t) = \sum_k T_{ik}\pi_i(t) = \pi_i(t)\sum_k T_{ik} = \pi_i(t)$$

and therefore the distribution is stationary.

Detailed balance holds if and only if the distribution is stationary.

Now we consider a complex optimization problem, the *Traveling Salesman Problem* (TSP), where we want to find the shortest closed path that connects N cities.



12 “cities” randomly distributed in the (0,1) square: the path corresponds to a random permutation of the sequence of cities.

(path length L=1.93834)

Paths are enumerated by permutations of “city names”, e.g., {9, 2, 7, 8, 1, 12, 4, 5, 3, 10, 11, 6} (start at 9, step to 2, and so on until you reach 6 and then return to 9).

The problem belongs to the class of NP-complete problems (Non-Polynomial complexity, a class of particulary hard problems)

The total number of configurations is

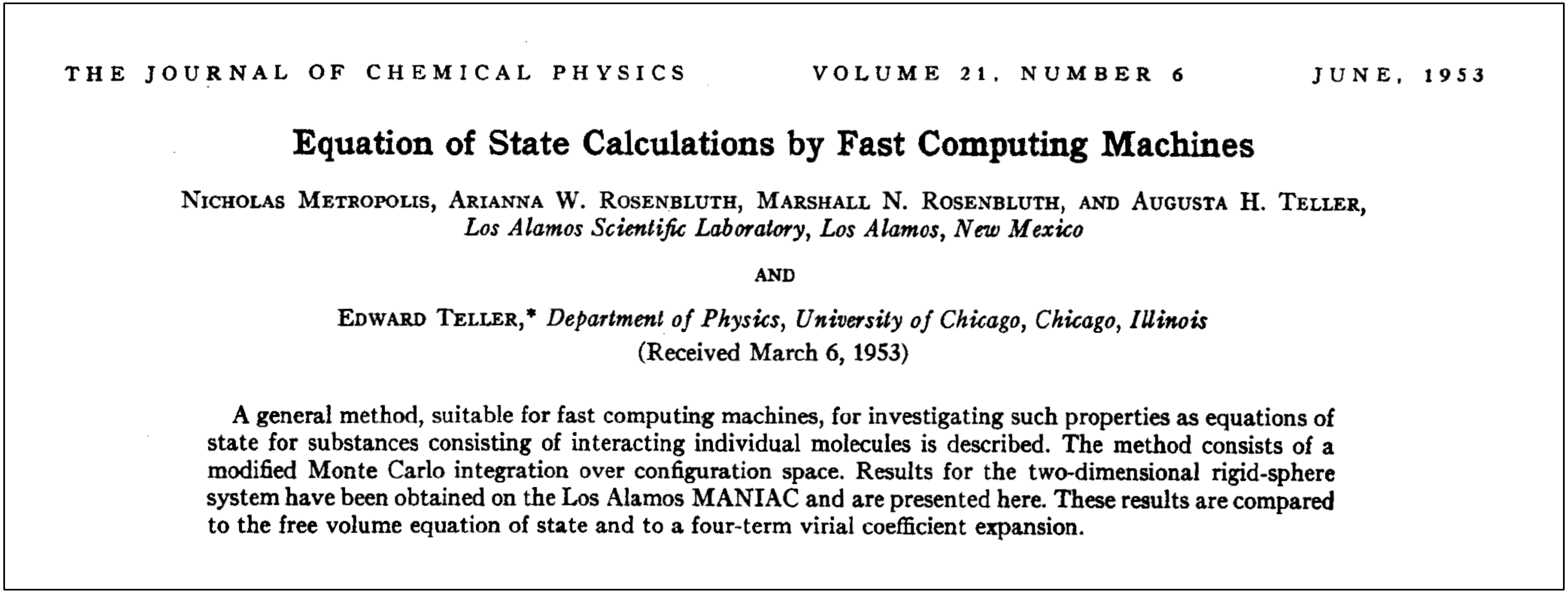
$$\frac{1}{2}(n-1)!$$

In such cases there is only one known solution: the full enumeration of all paths

Approximate solution of the TSP with the Simulated Annealing algorithm

path length → energy of the system

exploration of the configuration space with the *Metropolis algorithm* (Metropolis, Rosenbluth Rosenbluth ,Teller and Teller, 1953)



THE JOURNAL OF CHEMICAL PHYSICS VOLUME 21, NUMBER 6 JUNE, 1953

Equation of State Calculations by Fast Computing Machines

NICHOLAS METROPOLIS, ARIANNA W. ROSENBLUTH, MARSHALL N. ROSENBLUTH, AND AUGUSTA H. TELLER,
Los Alamos Scientific Laboratory, Los Alamos, New Mexico

AND

EDWARD TELLER,* *Department of Physics, University of Chicago, Chicago, Illinois*

(Received March 6, 1953)

A general method, suitable for fast computing machines, for investigating such properties as equations of state for substances consisting of interacting individual molecules is described. The method consists of a modified Monte Carlo integration over configuration space. Results for the two-dimensional rigid-sphere system have been obtained on the Los Alamos MANIAC and are presented here. These results are compared to the free volume equation of state and to a four-term virial coefficient expansion.

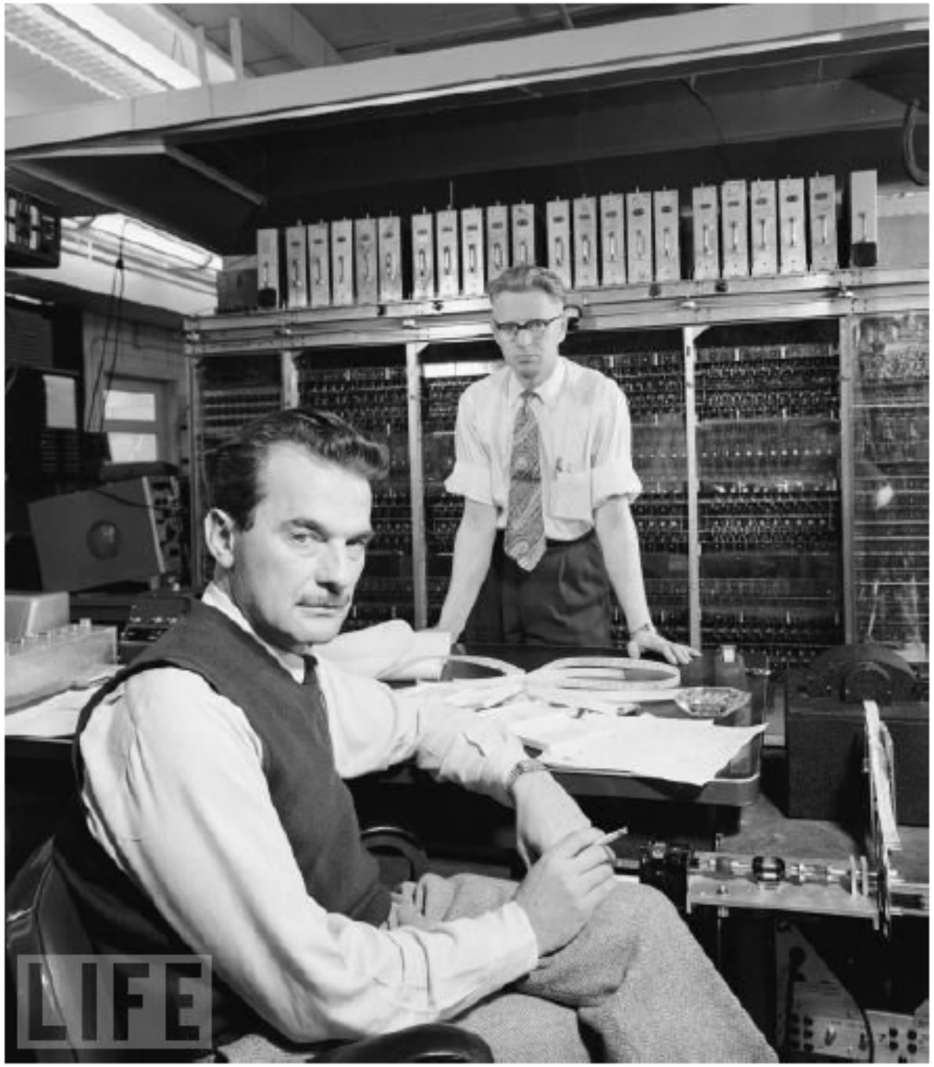


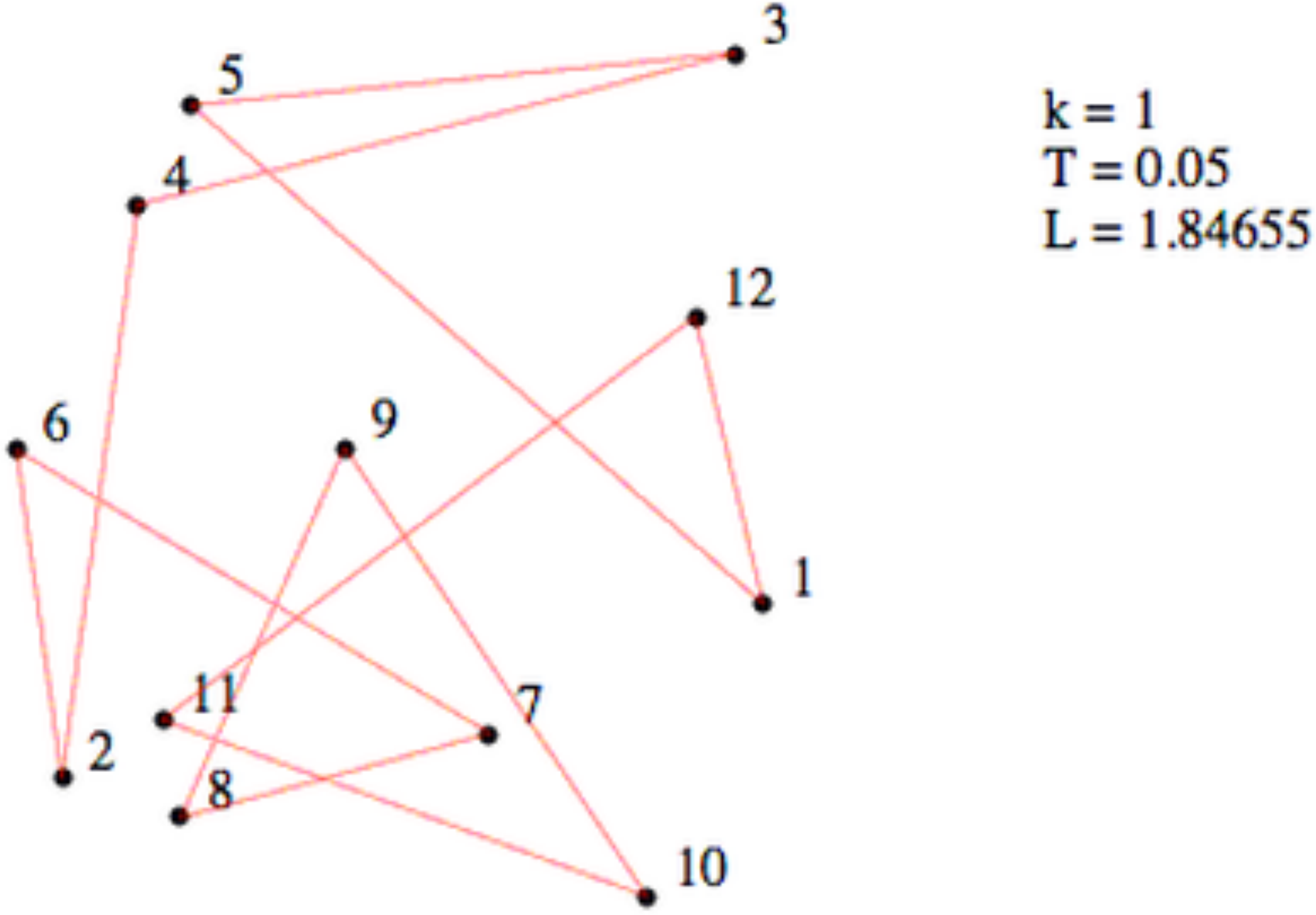
Figure 8.14: Portrait of American computer scientists Nicholas Metropolis (1915 - 1999) (seated) and James Henry Richardson (1918 - 1996) at Los Alamos National Laboratory, Los Alamos, New Mexico, November 1953 (from `http://www.life.com`).

1. We generate a new configuration C′ from the present configuration C
2. We compute the energy of the new configuration, E'
3. We compute the energy difference $\Delta E = E' - E$
4. The new configuration is accepted with probability p

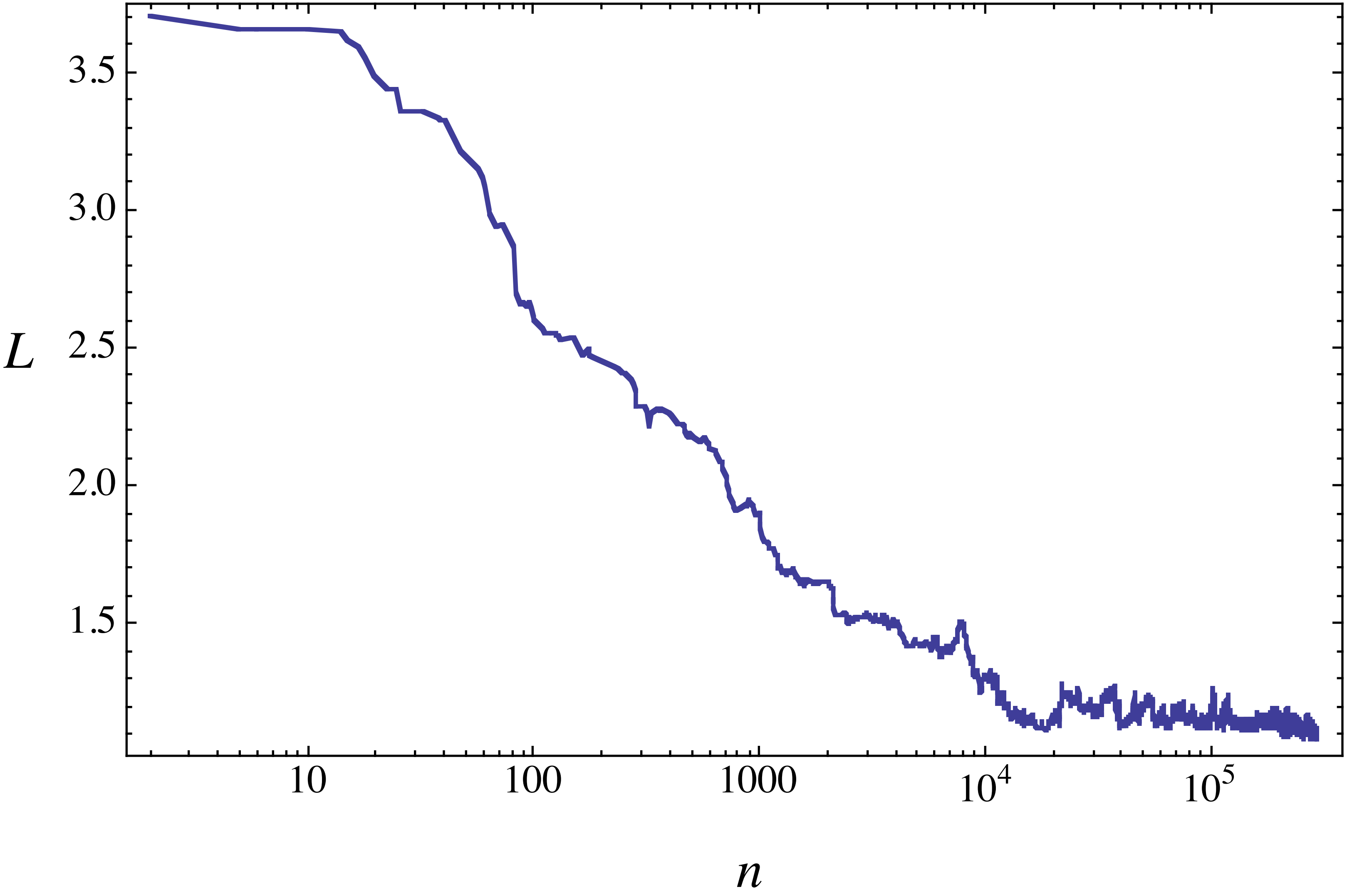
$$\begin{cases} p = 1 & \Delta E < 0 \\ p = \exp\left(-\dfrac{\Delta E}{kT}\right) & \Delta E \geq 0 \end{cases}$$

Additional details

- the algorithm needs a slow cooling (it is common to choose an exponential cooling schedule)
- if cooling is not gradual, the system can get stuck into a local minimum
- simple exchanges of pairs of cities are the individual moves in the SA solution of the TSP
- the individual steps from one configuration to the next can be described by a Markov chain


3
5
4
12
6
9
1
11
7
2
8
10
k = 1
T = 0.05
L = 1.84655

Decrease of total path length in a realization of the SA solution of the 50-cities problem



Here we note that the transition probability can be written as follows

$$T(C \rightarrow C') = \min\left[1, \exp\left(-\frac{(E' - E)}{kT}\right)\right]$$

Moreover, the algorithm preserves detailed balance

$$P(C)T(C \rightarrow C') = P(C')T(C' \rightarrow C)$$

where P(C) is the stationary probability of configuration C. Indeed, if E’ > E

$$P(C)\exp\left(-\frac{(E' - E)}{kT}\right) = P(C')$$

$$\frac{P(C')}{P(C)} = \exp\left(-\frac{(E' - E)}{kT}\right)$$

⇦ Boltzmann’s distribution

Moreover

$$T\left(C \rightarrow C'\right) = \min\left[1, \frac{P\left(C'\right)}{P\left(C\right)}\right]$$

This algorithm is the starting point for an important further step, the Metropolis-Hastings algorithm.

6. MCMC – definition of the Metropolis-Hastings (M-H) algorithm (1970)

- we define the transition probability

$$q(\mathbf{x},\mathbf{y}) = P(\mathbf{x} \to \mathbf{y})$$

and the target density $\pi(\mathbf{x})$

- we take state $\mathbf{x} = \mathbf{x}_n$

- we choose randomly another state $\mathbf{y}$ and we accept it $(\mathbf{y} \to \mathbf{x}_{n+1})$ with probability

$$\alpha(\mathbf{x},\mathbf{y}) = \min\left\{1, \frac{\pi(\mathbf{y})q(\mathbf{y},\mathbf{x})}{\pi(\mathbf{x})q(\mathbf{x},\mathbf{y})}\right\}$$

If the transition probability is symmetrical, then the acceptance probability takes on the simpler form

$$\alpha(\mathbf{x},\mathbf{y}) = \min\left\{1, \frac{\pi(\mathbf{y})q(\mathbf{y},\mathbf{x})}{\pi(\mathbf{x})q(\mathbf{x},\mathbf{y})}\right\} \rightarrow \min\left\{1, \frac{\pi(\mathbf{y})}{\pi(\mathbf{x})}\right\}$$

and it depends on the target density only.

The M-H algorithm defines a Markov chain and it is easy to show that detailed balance holds. The transition probability is

$$P(\mathbf{x} \to \mathbf{y}) = q(\mathbf{x},\mathbf{y})\alpha(\mathbf{x},\mathbf{y}) = q(\mathbf{x},\mathbf{y})\min\left\{1, \frac{\pi(\mathbf{y})q(\mathbf{y},\mathbf{x})}{\pi(\mathbf{x})q(\mathbf{x},\mathbf{y})}\right\}$$

- case $\dfrac{\pi(\mathbf{y})q(\mathbf{y},\mathbf{x})}{\pi(\mathbf{x})q(\mathbf{x},\mathbf{y})} \geq 1$

$$\Rightarrow \quad \alpha(\mathbf{x},\mathbf{y}) = 1; \quad \alpha(\mathbf{y},\mathbf{x}) = \frac{\pi(\mathbf{x})q(\mathbf{x},\mathbf{y})}{\pi(\mathbf{y})q(\mathbf{y},\mathbf{x})} \quad \Rightarrow \quad \begin{aligned} &P(\mathbf{x} \to \mathbf{y}) = q(\mathbf{x},\mathbf{y}) \\ &P(\mathbf{y} \to \mathbf{x}) = q(\mathbf{y},\mathbf{x})\frac{\pi(\mathbf{x})q(\mathbf{x},\mathbf{y})}{\pi(\mathbf{y})q(\mathbf{y},\mathbf{x})} \end{aligned}$$

$$\Rightarrow \quad \begin{aligned} &\pi(\mathbf{x})P(\mathbf{x} \to \mathbf{y}) = \pi(\mathbf{x})q(\mathbf{x},\mathbf{y}) \\ &\pi(\mathbf{y})P(\mathbf{y} \to \mathbf{x}) = \pi(\mathbf{y})q(\mathbf{y},\mathbf{x})\frac{\pi(\mathbf{x})q(\mathbf{x},\mathbf{y})}{\pi(\mathbf{y})q(\mathbf{y},\mathbf{x})} = \pi(\mathbf{x})q(\mathbf{x},\mathbf{y}) \end{aligned}$$

- case $\frac{\pi(\mathbf{y})q(\mathbf{y},\mathbf{x})}{\pi(\mathbf{x})q(\mathbf{x},\mathbf{y})} < 1$

⇨ $\alpha(\mathbf{x},\mathbf{y}) = \frac{\pi(\mathbf{y})q(\mathbf{y},\mathbf{x})}{\pi(\mathbf{x})q(\mathbf{x},\mathbf{y})}; \quad \alpha(\mathbf{y},\mathbf{x}) = 1$ ⇨

$$P(\mathbf{x}\to\mathbf{y}) = q(\mathbf{x},\mathbf{y})\frac{\pi(\mathbf{y})q(\mathbf{y},\mathbf{x})}{\pi(\mathbf{x})q(\mathbf{x},\mathbf{y})}$$

$$P(\mathbf{y}\to\mathbf{x}) = q(\mathbf{y},\mathbf{x})$$

⇨

$$\pi(\mathbf{x})P(\mathbf{x}\to\mathbf{y}) = \pi(\mathbf{x})q(\mathbf{x},\mathbf{y})\frac{\pi(\mathbf{y})q(\mathbf{y},\mathbf{x})}{\pi(\mathbf{x})q(\mathbf{x},\mathbf{y})} = \pi(\mathbf{y})q(\mathbf{y},\mathbf{x})$$

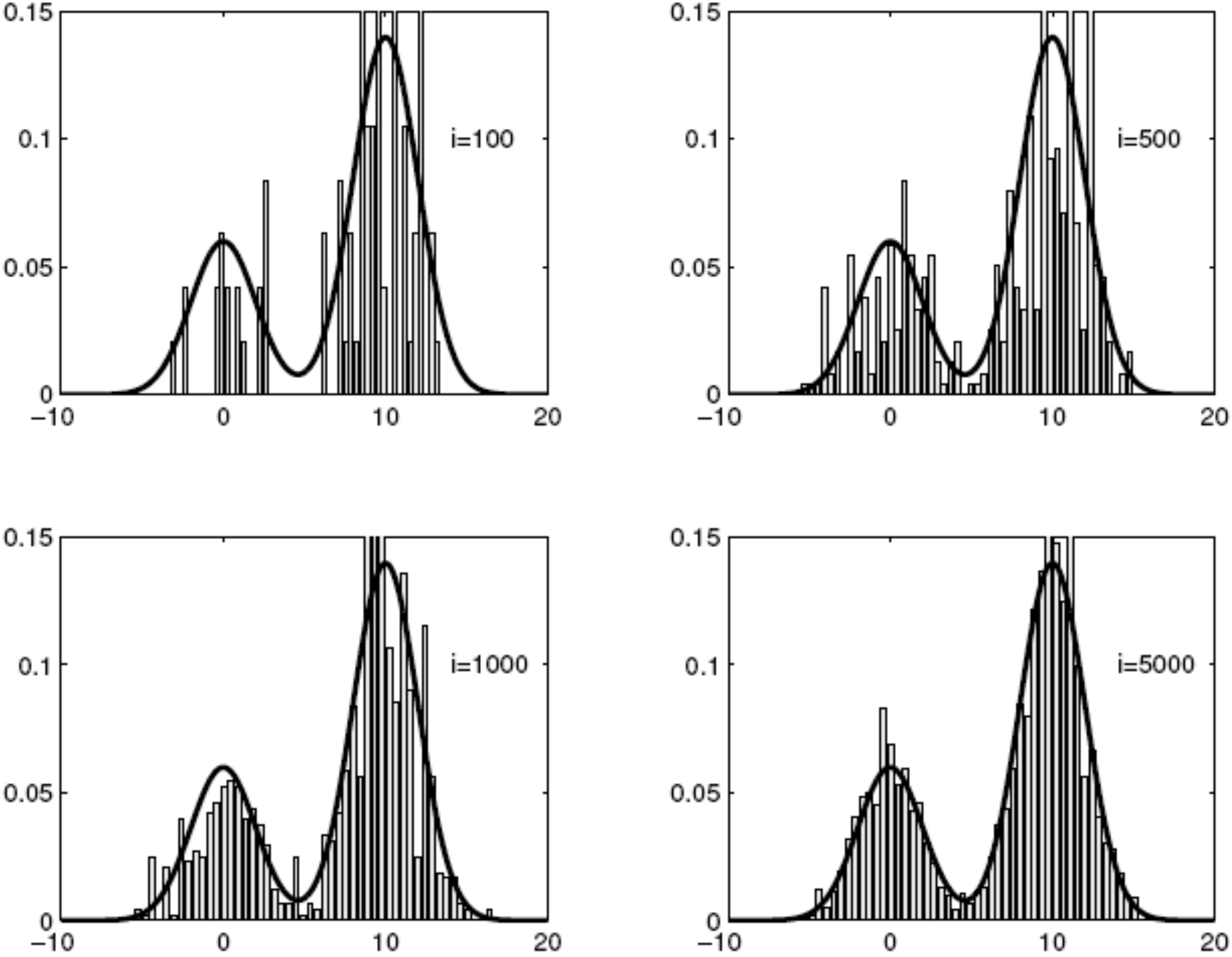
$$\pi(\mathbf{y})P(\mathbf{y}\to\mathbf{x}) = \pi(\mathbf{y})q(\mathbf{y},\mathbf{x})$$

Detailed balance holds in both cases and therefore $\pi(\mathbf{x})$ is stationary

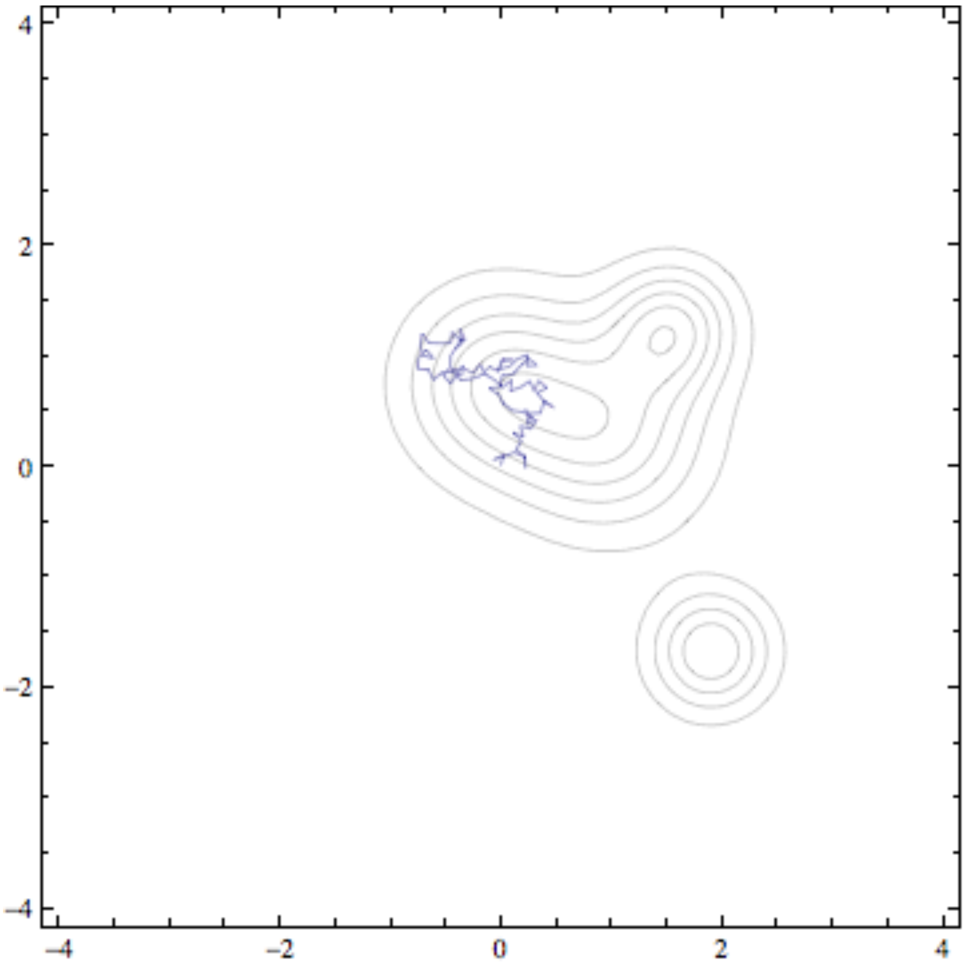
The following figure shows a simulation with the MCMC algorithm and the distribution

$$p(x) = 0.3\exp\left(-0.2x^2\right) + 0.7\exp\left(-0.2(x-10)^2\right)$$

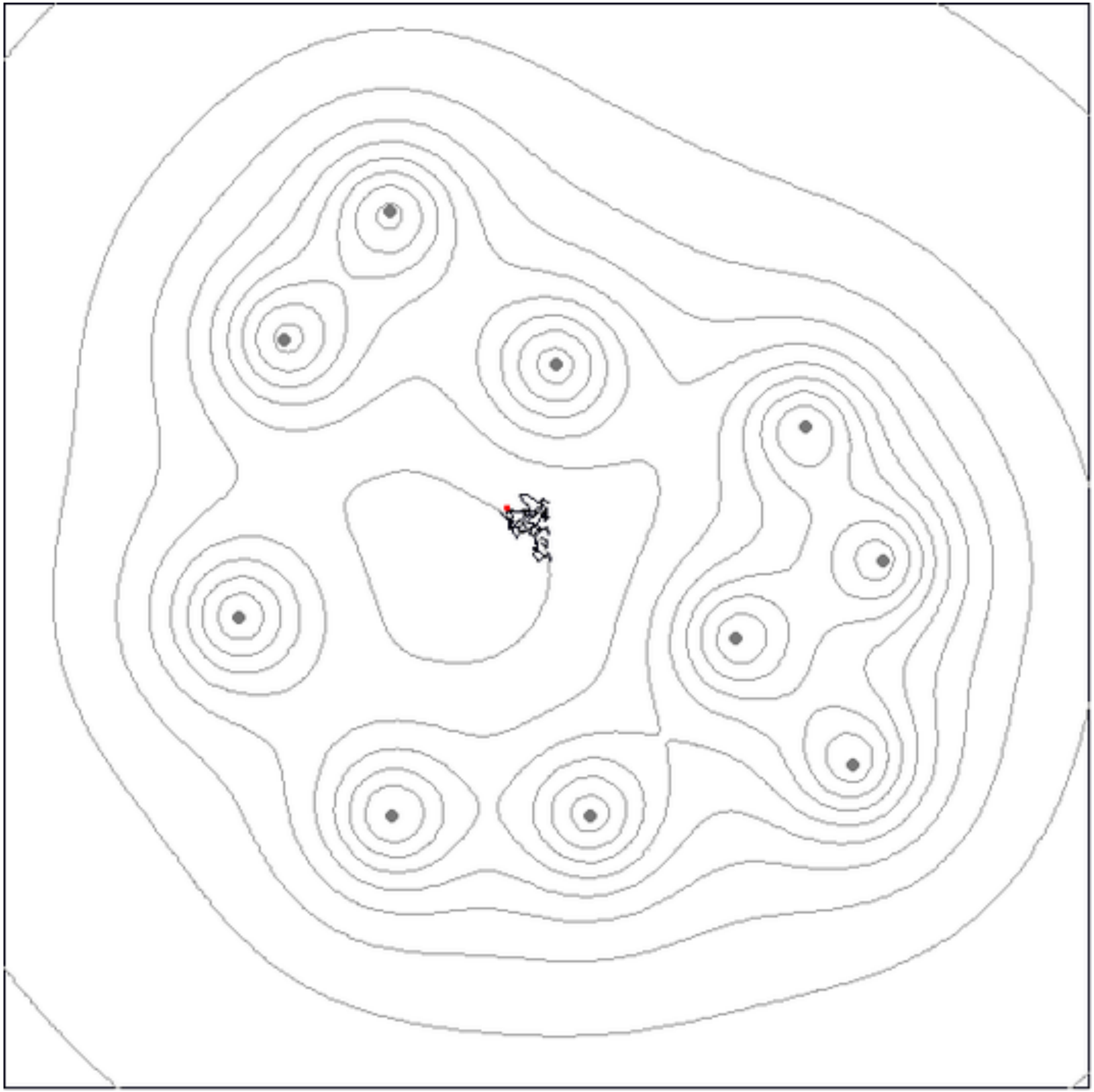
(a two-component mixture model)



6. Target distribution and histogram of the MCMC samples at different iteration points.



4
2
0
−2
−4
−4
−2
0
2
4



•*References:*

- D. Flury, “ Acceptance-Rejection Sampling Made Easy ” SIAM Review **32** (1990) 474
- H. Varian, “Bootstrap Tutorial”, The Mathematica Journal **9** (2005) 768
- A. F. M. Smith and A. E. Gelfand: “Bayesian Statistics Without Tears: A Sampling-Resampling Perspective”, Am. Stat. **46** (1992) 84
- B. Walsh: “Markov Chain Monte Carlo and Gibbs Sampling”, http://web.mit.edu/~wingated/www/introductions/mcmc-gibbs-intro.pdf
- S. P. Brooks: “Markov Chain Monte Carlo and Its Application”, The Statistician **47** (1998) 69