Why M-H algorithm works.

Basic 'all-at-once' M-H: use detailed balance argument.

Need : $\pi(x)\, K(x,y) = \pi(y)\, K(y,x)$

i.e., $\pi(x)\; q(x,y)\, \alpha(x,y) = \pi(y)\; q(y,x)\, \alpha(y,x)$ (*)

If $x=y$, (*) trivially satisfied.

Assume w.l.o.g. that $\pi(y)\, q(y,x) > \pi(x)\; q(x,y)$

then LHS of (*) is

$$\pi(x)\; q(x,y) \times \min\left\{1,\ \frac{\pi(y)\; q(y,x)}{\pi(x)\; q(x,y)}\right\}$$

$$= \pi(x)\; q(x,y) \cdot 1$$

RHS of * is $\pi(y)\, q(y,x) \times \min\left\{1,\ \frac{\pi(x)\; q(x,y)}{\pi(y)\; q(y,x)}\right\}$

$$= \cancel{\pi(y)\, q(y,x)} \cdot \frac{\pi(x)\; q(x,y)}{\cancel{\pi(y)\; q(y,x)}} = \text{RHS}.$$

Hence detailed balance is satisfied.

MCMC 21 (or 5(a)?)

Sufficient conditions for M. chain SLLN to hold:

(A) If $q(x,y) > 0 \quad \forall x, y \in \Omega$, M.C. is trivially irreducible: every set A can be reached in 1 step.

(B) If for each x, $P(\pi(x) q(x,y) \leq \pi(y) q(y,x)) < 1$

then M.C. has positive prob. of staying at x.

M.C. is strongly aperiodic.

Why: $\alpha(x,y) = 1$ w/ prob < 1

$\alpha(x,y) < 1$ w/ prob > 0

Hence, positive probability of staying at x.

(C) Detailed balance is satisfied w.r.t. π.

If (A), (B), (C) satisfied, M.C. is Harris ergodic w/ stationary distr. π. Thm. 1 (SLLN) applies. pg. 273 Robert & Casella

Why Variable-at-a-time M-H works

Consider transition kernel of M.C. to be the product of transition kernels for each block.

For e.g. two blocks, $\underset{\sim}{x} = (x_1, x_2)$

Transition kernel of chain, $K\left((x_1, x_2), (y_1, y_2)\right)$

$$= K_{1|2}\left(\underbrace{(x_1, x_2), (y_1, x_2)}_{\text{update}}\right) K_{2|1}\left(\underbrace{(y_1, x_2), (y_1, y_2)}_{\text{update}}\right)$$

Also: see this as composition of kernels, preserving stationarity

~~Why this works~~: Can show trans. kernel K has π as its stationary distr.

$$\int\int_{(x_1,x_2)\in\mathcal{X}} \underbrace{K_{1|2}(x_1,y_1|x_2)\, K_{2|1}(x_2,y_2|y_1)}_{K(x,y)} \underbrace{\pi(x_1,x_2)}_{\pi(x)} \underbrace{dx_1\, dx_2}_{dx}$$

$$= \int K_{2|1}(x_2,y_2|y_1) \underbrace{\left[\int K_{1|2}(x_1,y_1|x_2)\, \pi_{1|2}(x_1|x_2)\, dx_1\right]}_{=\ \pi_{1|2}(y_1|x_2)} \pi_2(x_2)\, dx_2$$

$$= \int K_{2|1}(x_2,y_2|y_1)\, \pi_{1|2}(y_1|x_2)\, \pi_2(x_2)\, dx_2$$

($\because K_{1|2}$ ~~has~~ is trans. kernel ~~for~~ w/ stationary distr. $\pi_{1|2}$)

$$= \int K_{2|1}(x_2,y_2|y_1)\, \pi_{2|1}(x_2|y_1)\, \pi_1(y_1)\, dx_2$$

$$= \pi_1(y_1) \int K_{2|1}(x_2,y_2|y_1)\, \pi_{2|1}(x_2|y_1)\, dx_2$$

$$= \pi_1(y_1)\, \pi_2(y_2|y_1)$$

$$= \pi(y)$$

So, $\displaystyle\int_{x\in\mathcal{X}} K(x,y)\, \pi(x)\, dx = \pi(y)$

$\Rightarrow \pi$ is stationary distr. of K.

Can similarly show such a result for VMH w/ ~~an arbitrary~~ # of blocks > 2. Elegant proof requires

Note: MC ~~from~~ block-at-a-time Met-Hastings is generally not reversible. However, can easily make it reversible by randomizing order of update of the blocks, or using 'palindromic' updates

$K_{1|2,3}\ K_{2|1,3}\ K_{3|1,2}\ K_{2|1,3}\ K_{1|2,3}$
$K_A\ K_B\ K_C\ K_B\ K_A$

Rev. useful for proving theoretical results. (harder when non-rev.)

Total variational distance between two distributions π_1 and π_2 is defined

$$\| \pi_1 - \pi_2 \|_{TV} = \sup_{A} |\pi_1(A) - \pi_2(A)| \quad \text{where}$$

$(A \in \mathcal{B}(\mathcal{X}))$

$$\pi_1(A) = \int_A \pi_1(x)\,dx \quad \text{when } \pi_1, \pi_2 \text{ are densities}$$

$$\pi_2(A) = \int_A \pi_2(x)\,dx$$

<u>Thm 2</u>: If $\{X_n\}$ is a Harris-ergodic M.C. w/ stationary distr. π and n-step transition kernel $k^n(y|x)$

$$\lim_{n \to \infty} \| P_n^n(x, \cdot) - \pi(\cdot) \|_{TV} = 0 \quad \forall x \in \mathcal{X}$$

where $P^n(x, A) = \int_A k^n(y|x)\,dy$

i.e, $P^n(x, A) = P(X_{i+n} \in A \mid X_i = x)$

<u>Implication</u>: regardless of starting value, Harris-ergodic M.C. w/ stationary distr. π will produce values that look like draws from π, after the M.C. has been running for a long time.

Rates of convergence in total-variational distance:

$$\| P^n(x, \cdot) - \pi(\cdot) \|_{TV} \le M(x) \underset{\ge 0}{t^n}$$

Geometric ergodicity: above holds w/ $t < 1$, $M(x)$ is extended real-valued function

Uniform ergodicity: above holds w/ $t < 1$ and $M(x) \le M^* < \infty$.

w/ $\pi M < \infty$

A rough guide to writing an MCMC algorithm

Here is a rough guide for constructing an MCMC algorithm. Each step describes some of the decisions that need to be made at each stage along with associated tradeoffs.

1. **Derive $h(\Theta)$, the unnormalized joint distribution** of interest, which is proportion to the target $\propto \pi(\Theta)$ where $\Theta = (\theta_1, \ldots, \theta_p)$.

2. **Identify blocks or components of Θ that would be relatively easy to simulate 'all-at-once'.** This is a very difficult decision to make in general and will typically require some trial and error though, sometimes, the structure of the problem may suggest a blocking strategy. A basic principle to follow is to look for blocks that appear to be highly correlated, as updating these jointly in a sensible fashion will typically make the sampler more efficient. The tradeoff will often be as follows: updating large blocks of parameters will allow the sampler to move around the distribution more efficiently but designing good proposals for large blocks can be difficult *and* updating large blocks of parameters may be computationally more demanding (due to matrix operations that may be involved in the proposal and accept-reject steps of the algorithm.)

3. **Derive full conditional distributions** for each block based on the decision made in Step 2.

4. **Identify any blocks with full conditionals that have a known form.** Use Gibbs updates for these full conditionals. Please note, however, that this is only the first approach to take — if the sampler works poorly, you may even decide to use a Metropolis-Hastings update instead of a Gibbs update.

5. Construct Metropolis-Hastings updates for the remaining blocks. **There are a large number of possibilities when it comes to determining what M-H update to use.** The first M-H update to try is a simple Metropolis update with a symmetric proposal. It may also be useful to try transformations. For example, if θ_i has positive support, try $\phi_k = \log(\theta_i)$ instead. ϕ_i's full conditional distribution may be easier to deal with and it is easy to transform the sampled value back to θ_i.

MCMC 26

6. **Run the M-H algorithm for short trial runs**, printing out lots of intermediate results:
   (a) This can help you make sure there are no obvious errors with your algebra or your programming.
   (b) You can see if you need to make changes to your algorithm. For example: If some of the parameters seem to be highly correlated, you may decide to sample them in a block. If Metropolis updates seem to result in highly autocorrelated samples, try changing the tuning parameters.
   (c) **Save the last draw of each of your trial runs and use it to start your next run.** This is a great way to obtain reasonable starting values. Any value you would not mind having in your sample is a reasonable value to start the sampler at (Geyer 2000).
7. Once obvious problems have been fixed and you have fine tuned your algorithm, **run the chain for as long as possible. If the Monte Carlo standard errors for the estimates of expectations of interest are acceptable, you can stop the chain.** There are many estimates of Monte Carlo standard error. We recommend the `consistent batch means` estimate ((3)).Of course, be aware that this approach works well only when the sampler is working reasonably well. If the Markov chain sampler is poor (slow mixing) and unable to find multiple modes of a multimodal distribution, all methods for deciding when to stop the chain will be ineffective.
8. **Report your final estimates, along with any estimates of error associated with them.** It is useful to also indicate (for future reference) the length of the chain used and what algorithm you finally ended up using.

An important general computing principle is to **always do calculations on the log scale as far as possible.** For instance, for the Metropolis-Hastings accept-reject step, instead of simulating $U \sim$Unif(0,1) and checking if

$$U < \frac{\pi(\theta_i^*|\Theta_{-i})q(\theta_i^*, \theta_i)}{\pi(\theta_i|\Theta_{-i})q(\theta_i, \theta_i^*)},$$

check whether:

$$\log(U) < \log(\pi(\theta_i^*|\Theta_{-i})) + \log(q(\theta_i^*, \theta_i)) - \log(\pi(\theta_i|\Theta_{-i})) - \log(q(\theta_i, \theta_i^*)).$$

MCML 27

Of course, this is the same principle that was applied to classical Monte Carlo approaches, leading to much more stable and accurate computations.

Burn-in, subsampling, and multiple chains

Burn-in: Since the Markov chain is typically not started with a value from the stationary distribution, the estimates based on the samples are biased. Burn-in is an ad-hoc method used by MCMC practitioners to try to reduce the bias of the estimates by simply removing an arbitrary number of initial values from the chain and using the remainder of the chain to do estimation. Unfortunately, there is no way to know how much bias is being removed (if any) by discarding some initial draws. The only thing we know for sure is that the variability of the estimate has been increased! For more on this, read the thoughtful (and entertaining) discussion on MCMC diagnostics by C.J.Geyer (`http://www.stat.umn.edu/~charlie/mcmc/diag.html`). To quote Geyer (2000) "Any value you do not mind having in your sample is a reasonable starting value." A useful recommendation is to simply start the chain off at the last sample from the previous MCMC run. Since there is often a fair amount of tuning involved in developing an effective Metropolis-Hastings algorithm, these tuning runs can be helpful in providing the initial value for the final run of the sampler.

: Since estimates based on $X_{b+1}, X_{b+2}, \ldots$ has (presumably) smaller bias than estimates based on $X_1, X_2, \ldots$ for large b (we are 'closer' to π).

Check: Geyer 92: effect of bias $\downarrow$ n while effect of ~~variance~~ s. error $\downarrow$ $\sqrt{n}$

Subsampling: Many MCMC users are troubled by heavy autocorrelation in their samples, as they should be. One way to deal with this issue is to simply subsample the chain, i.e., pick off every kth sample. For k large enough, this would result in a much less autocorrelated sample. From basic Markov chain theory, the subsampled Markov chain inherits the important properties of the original Markov chain (crucially, it has the same stationary distribution.) *If samples are easy to produce and expensive to process, subsampling can be a simple but useful approach.* For example: If it takes very little time to generate samples from a 10,000 dimensional distribution, and the draws are heavily autocorrelated, it may be worth subsampling the chain just to reduce the burden on storage and processing the samples (computing estimates of different kinds based on the samples.) The flipside is that *if it is expensive to generate samples, and the cost of processing them is not very high, subsampling is wasteful.* Using fewer samples, even if the samples are fairly dependent, is usually going to result in greater variability (reduced accuracy) of the estimates — the Monte Carlo standard errors are going to increase.

Multiple chains: Starting the sampler off at a few different values is a reasonable way to make sure that nothing eggregiously wrong is happening with the sampler. This is primarily useful for diagnosing coding errors and can be helpful (if we are lucky) with figuring out obvious multimodalities in the distribution. Luck comes into the picture because we will need to have chosen starting points that are located appropriately near the different modes to detect them, something that users typically cannot do. *Much more important is the idea of running the chain for as long as feasible, since a long run is likely to eventually pick up unusual features of the target distribution.*

MCMC 29 (a)

Being very careful

(1) Construct two different M-H alg., both resulting in same distr. Similar results?

(2) Plot marginal densities: overlay after $N/2$ and after N

Similar ?

(3) Run chain for very long

(4) MC se < small value, ESS for each component > 5,000.

(5) Try a few fairly different starting values

Similar ?

To compare alg: look at acf plots.

Terminology Notes:

Metropolis-Hastings alg: (V-MH):

Produce Harris ergodic M.C. w/ stationary distr. π by using M-H update for each 'block' of variables.

M-H update: suppose target distr. is $\pi(\theta_1, \dots, \theta_p)$.

MH Update of θ_k: propose $\theta_k^* \sim q(\theta_k, \theta_k^* \mid \underbrace{\theta_{-k}}_{\text{all } \theta_i\text{'s except } \theta_k})$

Accept w/ prob. $\alpha(\theta_k, \theta_k^*) = \min\left\{1, \frac{\pi(\theta_k^* \mid \theta_{-k})}{\pi(\theta_k \mid \theta_{-k})} \frac{q(\theta_k^*, \theta_k \mid \theta_{-k})}{q(\theta_k, \theta_k^* \mid \theta_{-k})}\right\}$

~~Ex~~ Special cases:

<u>Metropolis update</u> of θ_k: $q(\theta_k, \theta_k^* \mid \theta_{-k}) = q(\theta_k^*, \theta_k \mid \theta_{-k})$

so it is 'symmetric', ~~in its argum~~ e.g. random walk w/ Normal proposal

Accept w/ prob. $\alpha(\theta_k, \theta_k^*) = \min\left\{1, \frac{\pi(\theta_k^* \mid \theta_{-k})}{\pi(\theta_k \mid \theta_{-k})}\right\}$

<u>Gibbs update</u> of θ_k: ~~$\pi(\theta_k^* \mid \theta_{-k})$~~ $q(\theta_k, \theta_k^* \mid \theta_{-k}) = \pi(\theta_k^* \mid \theta_{-k})$

that is, simulate proposal <u>directly</u> from full condtl. distr.

Accept w/ prob $\alpha(\theta_k, k^*) = \min\left\{1, \frac{\cancel{\pi(\theta_k^* \mid \theta_{-k})}}{\cancel{\pi(\theta_k \mid \theta_{-k})}} \frac{\cancel{\pi(\theta_k \mid \theta_{-k})}}{\cancel{\pi(\theta_k^* \mid \theta_{-k})}}\right\}$

$= 1$. Always accept.

M-H algorithm allows for combination of all these kinds of updates.

Gibbs sampler $\Rightarrow$ M-H algorithm where <u>all</u> updates are Gibbs updates.

Note: Data augmentation : also falls under M-H algorithms in Situations where adding random variables (augmentation of target distr.) makes things easier. e.g. missing data problems – treating missing data as additional random variables to be sampled. e.g. mixture models, classification problems, survival analysis

<u>No need</u> for terminology like 'Metropolis within Gibbs', 'Hybrid M-H' etc.

Reminder of distinction between notions of variability in frequentist (classical) and Bayesian inference:

Frequentist: variability of estimates – if more random samples are obtained (of the same size as the observations), from same distr. how would this change the estimate? <u>Sampling variability</u> of estimates <u>estimate</u>: Asymptotic theory, bootstrap parametric or non-parametric

Bayesian: Suppose we quantify our knowledge about a parameter via a prob. distr. ("prior distr") and we now observe data that informs us about the param. (via a prob. model / likelihood fn.). What is our knowledge of the parameter now (condtl on data)? <u>Posterior distr. of parameter</u>. <u>estimate via</u>: asymptotic theory, Monte Carlo / MCMC.

driven smaller by ↑n.

Monte Carlo s.error versus

[posterior (if Bayes)] standard deviation ← fixed

MCse is an estimate of the quality of your estimate 'simulation error'.

E.g. estimate of [posterior (if Bayes) mean], $E_\pi(\beta) = 2.5$

w/ MCse of 0.0003

est. of [posterior (if Bayes) variance], $E_\pi\{(\beta - E_\pi(\beta))^2\} = 1.2$

w/ MCse of 0.004

[Posterior (if Bayes)] s.dev. $= \sqrt{E_\pi\{(\beta - E_\pi(\beta))^2\}}$

est. w/ sample posterior s.deviation.

For inference: care abt. posterior mean, s.dev.

To ensure estimates of above are accurate: care about MCs.e.

Central Limit Thm. & s.errors for MCMC

I.i.d. case: If $\mathrm{Var}_\pi(g(X_i)) < \infty$

$$\sqrt{n}(\hat{\mu}_n - \mu) \longrightarrow N(0, \sigma^2)$$

where $\sigma^2 = \mathrm{Var}_\pi\, g(X_i)$ and $\mathrm{Var}(\hat{\mu}_n) = \frac{\sigma^2}{n}$

Easy to estimate σ^2 from $X_1, \ldots, X_n \overset{iid}{\sim} \pi$

$\hat{\sigma}^2 = s^2 =$ sample variance of $X_1, \ldots, X_n$.

So, $\hat{\mu}_n \sim N(\mu, \frac{\hat{\sigma}^2}{n})$ and, $\mathrm{Var}(\hat{\mu}_n) \doteq \frac{\hat{\sigma}^2}{n}$

which can be used to estimate C.I.'s.

MCMC case:

① $\mathrm{Var}_\pi\, g(X_i) < \infty$ does not control whether a CLT exists.

② $\mathrm{Var}(\hat{\mu}_n) = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \mathrm{Cov}(g(X_i), g(X_j))$

since $g(X_i)$s are not iid.

Need to estimate covariances.

e.g. batch means, IMSE, (skip details)

MCMC CLT

There are many different sets of sufficient conditions for a Markov chain CLT.

Here are ~~two~~ some of the more intuitive sufficient conditions:

If $\{X_n\}$ is a Harris-ergodic Markov chain on Ω w/ stationary distr. π. For any real valued function g $(g : \mathcal{X} \to \mathbb{R})$ if either:

1. $\{X_n\}$ is geometrically ergodic and $E_\pi |g(x)|^{2+\varepsilon} < \infty$ for some $\varepsilon > 0$.
2. $\{X_n\}$ is geometrically ergodic, reversible and $E(g(x)^2) < \infty$.
3. $\{X_n\}$ is uniformly ergodic and $E_\pi(g(x)^2) < \infty$

then, for <u>any</u> initial distr.,

$$\lim_{n \to \infty} \sqrt{n}\,(\hat{\mu}_n - \mu) \longrightarrow N(0, \sigma^2)$$

More details: see Jones '04 and Roberts & Rosenthal '04.

Monte Carlo s.error for MCMC (MCMCse)

Need to estimate asymptotic variance, σ^2

$MCMC_{s.e} = \sigma/\sqrt{n}$

[G]enerally, σ is not easy to estimate.

One approach: batch means. Idea:

Run M.chain for $n = ab$ length.

Define $$Y_k = \frac{\sum_{i=(k-1)b+1}^{kb} g(X_i)}{b}, \quad k = 1, \ldots, a$$

= M.C. estimate of $\mu = \int g(x)\pi(x)dx$ based on k^{th} "batch"

$$\hat{\sigma}^2 = \frac{b}{a-1}\sum_{k=1}^{a}(Y_k - \hat{\mu}_n)^2 \qquad MC_{se} = \frac{\hat{\sigma}}{\sqrt{n}}$$

Above: $Var(Y_k) \approx \sum_{k=1}^{a}(Y_k-\hat{\mu})^2/(a-1)$, and $Var(\hat{\mu}_n) \approx b\,Var(Y_k)$

b batch size should large enough so we can treat Y_ks as approximately independent

"Consistent batch means", $b = \sqrt{n}$, so $a = \sqrt{n}$.

Simulation studies as Monte Carlo

Examples: Comparing two estimators

Studying/Validating asymptotic results in finite samples

Studying coverage prob. of a C.I.

E.g.1. My estimator is better than theirs. Investigate a scenario.

say data, $\underset{\sim}{y} \sim f(\theta)$

			Mine	Theirs	Truth
Sim. 1 :	$(y_{11}, \ldots, y_{1n}) = \underset{\sim}{y}_1$	:	$\hat{\phi}_1$	$\hat{\psi}_1$	θ_*
Sim. 2 :	$\underset{\sim}{y}_2$	:	$\hat{\phi}_2$	$\hat{\psi}_2$	
Sim. B :	y_B	:	$\hat{\phi}_B$	$\hat{\psi}_B$	

Assuming data $\sim f(\theta)$:

MSE of $\hat{\phi} = E_\theta(\hat{\phi} - \theta)^2$

Estimate by Monte Carlo: $\dfrac{\sum_{b=1}^{B} (\hat{\phi}_b - \theta)^2}{B}$, $\hat{\phi}_{\hat{MSE}}$

MSE of $\hat{\psi} = E_\theta(\hat{\psi} - \theta)^2$, ~~an expectation~~!

est. by $\dfrac{\sum_{b=1}^{B} (\hat{\psi} - \theta)^2}{B}$, $\hat{\psi}_{\hat{MSE}}$

Compare: $\hat{\phi}_{\hat{MSE}} \pm 2$ $\underbrace{\text{s.error}}_{\text{M.C. error}}$ to $\hat{\psi}_{\hat{MSE}} \pm 2$ s.error

Hopefully my estimator is better! (significantly smaller MSE)

Of course this may change w/ scenario.

E.g. Does my method for constructing a CI have the right coverage, i.e., does my 95% C.I. A have $P(\underset{\substack{\uparrow \\ \text{truth}}}{\theta} \in A) \approx 0.95$

		C.I.	Does C.I. include θ?
Sim 1.	y_1	A_1	$I(\theta \in A_1)$
⋮	⋮	⋮	⋮
Sim B	y_B	A_B	$I(\theta \in A_B)$

$$\text{M.C. estimate of covg.} = \frac{\sum_{b=1}^{B} I(\theta \in A_b)}{B} = \widehat{\text{covg}}(A)$$

Again, use M.C.s.error as a guide to determine B.

Note: easy to get s.err estimate for $\widehat{\text{covg}}(A)$: $\sqrt{\dfrac{\widehat{\text{covg}}\,(1-\widehat{\text{covg}})}{B}}$

Monte Carlo (or MCMC) Maximum Likelihood

Suppose $\mathcal{F} = \{f_\theta : \theta \in \Theta\}$ is a family of normalized densities

Now suppose $f_\theta(x) = \dfrac{h_\theta(x)}{c(\theta)}$ and you only now

$h_\theta(x)$ $\quad c(\theta)$ is normalizing function.

This can happen when using flexible models for complicated problems. e.g. $h_\theta(x) = e^{\langle t(x), \theta \rangle}$ (exponential family)

$t(x)$ is vector valued stats.

θ is parameter (unknown).

Goal: Want MLE for θ, $\hat{\theta} = \underset{\theta \in \Theta}{\operatorname{argmax}}\, f_\theta(x)$

Problem: Don't know $c(\theta)$!

MCML solution: Pick some $\psi \in \Theta$ s.t.

$h_\psi(x) = 0 \Rightarrow h_\theta(x) = 0$ for (almost) all x

Now, $\dfrac{c(\theta)}{c(\psi)} = \dfrac{1}{c(\psi)} \int h_\theta(x)\,dx = \int \dfrac{h_\theta(x)}{c(\psi)}\,dx$

$= \displaystyle\int \frac{h_\theta(x)}{h_\psi(x)/f_\psi(x)}\,dx = \int \frac{h_\theta(x)}{h_\psi(x)} f_\psi(x)\,dx$

$= E_{f_\psi}\left\{\dfrac{h_\theta(x)}{h_\psi(x)}\right\}$

Can simulate $Y_1, \ldots, Y_M \sim f_\psi$ by rej.sampling, M-H etc.

and estimate $\dfrac{c(\theta)}{c(\psi)}$ by $\displaystyle\sum_{j=1}^{M} \frac{h_\theta(Y_j)}{h_\psi(Y_j)} \Big/ M$

Goal: $\underset{\theta \in \Theta}{\operatorname{argmax}} \; f_\theta(x) = \underset{\theta \in \Theta}{\operatorname{argmax}} \; \dfrac{f_\theta(x)}{f_\psi(x)}$ ← constant w.r.t. θ

$$\Rightarrow \hat{\theta} = \underset{\theta \in \Theta}{\operatorname{argmax}} \log\left\{\frac{f_\theta(x)}{f_\psi(x)}\right\} = \underset{\theta \in \Theta}{\operatorname{argmax}} \; l(\theta) \text{, say}$$

$$\text{Now, } l(\theta) = \log\left[\frac{h_\theta(x)}{c(\theta)}\right] - \log\left[\frac{h_\psi(x)}{c(\psi)}\right]$$

$$= \log\left[\frac{h_\theta(x)}{h_\psi(x)}\right] - \log\left[\frac{c(\theta)}{c(\psi)}\right]$$

$$\Rightarrow \hat{l}(\theta) = \underbrace{\log\left[\frac{h_\theta(x)}{h_\psi(x)}\right]}_{\text{completely known}} - \underbrace{\log\left\{\frac{1}{M}\sum_{j=1}^{M}\frac{h_\theta(Y_j)}{h_\psi(Y_j)}\right\}}_{\text{Monte Carlo estimate}}$$

$\tilde{\theta}$ that maximizes $\hat{l}(\theta)$ is estimate of MLE.

Overview of Prob., Stats., Stoch. Proc. & Monte Carlo