Advanced Monte Carlo Methods

Quiz Study Guide — Chapter 9

Manuel D. Rossetti, PhD P.E.

How to Use This Deck

This deck is a **checklist**, not a summary. Every slide is phrased as something you should be able to *state*, *derive*, or *compute* without looking it up.

- **Question formats:** multiple choice, true/false, matching, fill-in-the-blank, multi-slot fill-in-the-blank.
- The **Common Traps** slides are built from the actual wrong answers — read them last.

Four themes

Bootstrapping; variance reduction techniques; generating **dependent** random variables; and Markov chain Monte Carlo.

What This Chapter Covers

Resampling and variance reduction

- Bootstrap replicates, bias, standard error
- Five bootstrap confidence intervals
- CRN, antithetics, indirect estimation
- Control variates, stratification, conditional expectation, importance sampling

Dependence and MCMC

- Multivariate normal via Cholesky
- Copulas and Sklar's theorem
- NORTA and correlated streams
- Metropolis-Hastings

Objectives — Bootstrapping and VRT

You should be able to:

- Define bootstrapping and write the bias and standard-error estimators
- Distinguish the five bootstrap confidence intervals, especially BCa
- State the rule of thumb for B
- Name the KSL bootstrap classes and what each resamples
- Write the CRN and control-variate variance results and say what sign of correlation each technique needs

Objectives — Dependence and MCMC

You should be able to:

- Say how `MVNormalRV` generates a sample and what decomposition it uses
- Define a copula and state Sklar's theorem
- Describe the NORTA transformation
- Write the Metropolis-Hastings acceptance ratio and simplify it for the independence and random-walk samplers

Bootstrapping

Bootstrapping is a **(generally non-parametric) Monte Carlo procedure that resamples WITH REPLACEMENT** from an original sample, treating that sample as essentially the population.

- $\hat{\theta}^*_b$, $b = 1, \ldots, B$, are the **bootstrap replicates —** the estimator computed on the bth bootstrap sample.
- **Standard error:** the sample **standard deviation of the bootstrap replicates** (not of the original sample).
- **Bias:** $\widehat{bias}(\hat{\theta}) = \bar{\hat{\theta}}^* - \hat{\theta}$ — the across-bootstrap average minus the original-sample estimate.
- **Rule of thumb:** $B \approx 40n$, with B at least in the hundreds.

The Five Bootstrap Intervals

Table 1: BCa is the one with **both** $\hat{z}_0$ and $\hat{a}$

Interval	Construction
SNBCI (standard normal)	$\hat{\theta} \pm z_{1-\alpha/2}\,\widehat{se}(\hat{\theta})$
BBCI (basic)	$(2\hat{\theta} - \hat{\theta}^*_{1-\alpha/2},\ 2\hat{\theta} - \hat{\theta}^*_{\alpha/2})$
BPCI (percentile)	$(\hat{\theta}^*_{\alpha/2},\ \hat{\theta}^*_{1-\alpha/2})$ — the quantiles directly
BCa	Percentile interval adjusted by a bias-correction $\hat{z}_0$ **and** an acceleration $\hat{a}$ from **leave-one-out jackknife** replicates
BTCI (bootstrap-t)	Reference distribution of $t^*_b = (\hat{\theta}^*_b - \hat{\theta})/\widehat{se}(\hat{\theta}^*_b)$; often needs an **inner** bootstrap

The KSL Bootstrap Classes

Table 2: `generateSamples(..., saveBootstrapSamples = true, ...)` saves the actual samples, one `DoubleArraySaver` each

Class or interface	Role
`Bootstrap`	Univariate: replicates, bias, std error, and all five intervals
`BSEstimatorIfc`	`estimate(data: DoubleArray): Double`; built-ins `Average`, `Variance`, `Median`, `Minimum`, `Maximum`
`BootstrapSampler`	**Multiple** quantities per sample via `MVBSEstimatorIfc`
`MVBSEstimatorIfc`	Returns several named estimates from one sample
`CaseBootstrapSampler`	Resamples **rows** of a data matrix — case-based
`MatrixBootEstimator` + `OLSBootEstimator`	Re-estimate OLS coefficients per sample

Variance Reduction — The Trivial One

The **trivial** technique, preceding all the others, is simply to **use a larger sample size**, since

$$\mathrm{Var}(\bar{Y}) = \frac{\sigma_Y^2}{n}$$

Everything else in the chapter is about getting a smaller variance **without** paying for more samples.

CRN and Antithetic Variates

Common Random Numbers — comparing two systems via $\bar{D} = \bar{X} - \bar{Y}$:

$$V_{CRN} = V_{IND} - 2\rho_{XY}\,\sigma_X\,\sigma_Y$$

so a **positive** ρ_{XY} gives $V_{CRN} \leq V_{IND}$.

Effective when responses are monotone in the stream, and when you: use **inverse-transform** generation, dedicate a **different stream to each source** of randomness, and **reset each replication to the same place** in each stream.

Antithetic Variates

One system, inducing **negative** correlation within consecutive pairs (Y_{2j-1}, Y_{2j}) by using u_k on the odd run and $1 - u_k$ on the even run. For a pair with correlation ρ:

$$\mathrm{Var}(\hat{Y}_j) = \frac{\sigma_Y^2(1+\rho)}{2}$$

- The KSL Model property is **`antitheticOption`**.
- On an **odd**-numbered replication beyond the first, `handleAntitheticReplications()` turns the antithetic option **off** and **advances each stream to the next sub-stream**.
- On an **even**-numbered replication it resets the start sub-stream and turns antithetic **on**.
- `MC1DIntegration` treats one replication as a **pair**: one run with the supplied sampler and one with `sampler.antitheticInstance()`, returning $(y_1 + y_2)/2$.

Indirect Estimation and Control Variates

Indirect estimation uses Little's formula and $W_s = W_q + E[ST]$ to estimate L_q, L_s, and W_s from $\hat{W}_q$ together with the **known** λ and $E[ST]$ — replacing variability with certainty wherever a quantity is known.

Control Variates

With $Z_j = Y_j + C(X_j - \mu)$ and $\mu = E[X]$:

$$C^* = -\frac{\mathrm{cov}(Y_j, X_j)}{\mathrm{Var}(X_j)}, \qquad \mathrm{Var}(Z_j) = \mathrm{Var}(Y_j)(1 - \rho^2_{XY})$$

Note the **minus sign** in C^*.

With several controls, fit $Y_i = \beta_0 + \beta_1(X_1 - \mu_1) + \cdots + \beta_q(X_q - \mu_q) + \epsilon_i$; the point estimator for $\theta = E[Y]$ is the **intercept** $\hat{\beta}_0$.

Under bivariate normality, a reduction is expected when $\rho^2 \geq 1/(n-2)$ — even a small correlation helps. The KSL class is `ControlVariateDataCollector`.

Stratification, Conditioning, Importance Sampling

Stratified sampling. With known strata probabilities $p_j = P(X \in L_j)$:

$$\hat{\theta} = \sum_{j=1}^{k} \hat{\theta}_j \, p_j, \qquad n_j = \frac{n \, p_j \, \sigma_j}{\sum_i p_i \, \sigma_i} \quad \text{(Neyman optimal allocation)}$$

Conditional expectation. $\hat{\theta} = \frac{1}{n} \sum E[Y \mid X_i]$ works because $\mathrm{Var}[E[Y|X]] \leq \mathrm{Var}[\bar{Y}]$ — replacing Y by its conditional expectation removes variability while preserving the mean. In the SAN example, conditioning is on the **common arcs** X_1 and X_5 that appear in every path, so $P(Y \leq t \mid X_1, X_5)$ **factors into a product** of the remaining marginals.

Importance sampling. $\theta = \int_a^b \frac{g(x)}{w(x)} w(x) \, dx$ with $X \sim w$. The theoretically optimal w is the f in the factorization $g = hf$ — but finding it needs θ, so heuristically choose w **shaped like** g. For $g(x) = x^2$ on $[0, 1]$, using $w_2(x) = 2x$ gives a variance ratio of about $4/3240$. Support comes from `MC1DIntegration` and `MCMultiVariateIntegration`.

Generating Dependent Random Variables

Table 3: `MVRVariable` is the abstract parent; `MVSampleIfc` specifies `sample()`, `sample(array)`, `sample(sampleSize)`, `sampleByColumn(sampleSize)`, and `dimension`

Class	Method
`BivariateNormalRV`	$X_1 = \mu_1 + \sigma_1 Z_0$, $X_2 = \mu_2 + \sigma_2(\rho Z_0 + \sqrt{1-\rho^2}\, Z_1)$
`BivariateLogNormalRV`	Sample a BVN, then exponentiate componentwise
`MVNormalRV`	**Cholesky** decomposition $\Sigma = CC^T$, then $X_i = \mu_i + \sum_{j \le i} c_{ij} Z_j$
`MVGaussianCopulaRV`	Sample $\vec{U}$ from the Gaussian copula, then $Y_i = F_i^{-1}(U_i)$ from `InverseCDFIfc` marginals
`AR1CorrelatedRNStream`	$Z_i = \phi Z_{i-1} + \varepsilon_i$, $\varepsilon_i \sim N(0, 1-\phi^2)$; `randU01()` returns $\Phi(Z_i)$

Copulas, Sklar, and NORTA

A d-dimensional **copula** $C(u_1, \dots, u_d)$ is a **joint CDF whose marginals are** $U(0,1)$.

Sklar's theorem. For any random vector with joint CDF F and continuous marginals F_j, there exists a **unique** copula C with

$$F(x_1, \dots, x_d) = C\big(F_1(x_1), \dots, F_d(x_d)\big)$$

NORTA (Normal-to-Anything). Generate correlated $Z_i \sim N(0,1)$, apply $\Phi(Z_i)$ to get correlated uniforms, then set

$$X_i = F^{-1}\big(\Phi(Z_i)\big)$$

Metropolis-Hastings

$$\rho(\vec{x},\vec{y}) = \frac{q(\vec{x} \mid \vec{y})\, f(\vec{y})}{q(\vec{y} \mid \vec{x})\, f(\vec{x})}, \qquad \alpha(\vec{x},\vec{y}) = \min(\rho,\, 1)$$

- **Independence sampler:** $q(\vec{y} \mid \vec{x}) = q(\vec{y})$, so $\rho = \dfrac{q(\vec{x}) f(\vec{y})}{q(\vec{y}) f(\vec{x})}$.
- **Random-walk sampler:** $\vec{Y} = \vec{x} + \vec{W}$ with $\vec{W}$ symmetric, so the **proposal ratio is 1** and $\rho = f(\vec{y})/f(\vec{x})$.
- Because ρ depends on the target only through a **ratio**, π need only be known **up to a multiplicative constant** — useful when the normalizer is unknown.

The KSL MCMC Classes

Class or interface	Role
`MetropolisHastingsMV`	Multivariate MH runner with warm-up, observers, **batch statistics**
`FunctionMVIfc`	Target density: `f(x: DoubleArray)` plus a `dimension`
`ProposalFunctionMVIfc`	`generateProposedGivenCurrent(...)`, `proposalRatio(...)`
`DMarkovChain`	Discrete-state chain from a transition matrix

Trap

`MetropolisHastingsMV` reports across-state summaries using **batch statistics**, because the generated states are **not** independent.

Drills

This chapter's question bank has no numeric items — these are practice problems built from the formulas and rules it tests in other formats. Expect the quiz to test the same relationships, not these exact numbers.

1. An original sample has $n = 25$. What B does the chapter's rule of thumb suggest?
2. Two systems are compared with paired runs. $\sigma_X = 4$, $\sigma_Y = 3$, $\rho_{XY} = 0.5$, and $V_{IND} = 25$. What is V_{CRN}?
3. A single control variate has $\rho_{XY} = -0.6$. By what factor is $\mathrm{Var}(Y)$ reduced? With $n = 20$ replications, is the reduction expected to be worthwhile?
4. Antithetic pair with $\sigma_Y^2 = 8$ and induced correlation $\rho = -0.5$. What is $\mathrm{Var}(\hat{Y}_j)$? Compare to the independent-pair value.
5. In a random-walk Metropolis-Hastings sampler, $f(\vec{y}) = 0.4$ and $f(\vec{x}) = 0.5$. What is the acceptance probability?

Common Traps

- Bootstrapping resamples **with** replacement — duplicates are expected.
- The bootstrap standard error is the SD of the **replicates**, not of the original sample.
- CRN needs **positive** correlation; antithetics need **negative** correlation.
- $C^* = -\mathrm{cov}(Y, X)/\mathrm{Var}(X)$ — the **minus sign** is part of the answer.
- The control-variate estimator is the regression **intercept**, not $\bar{Y}$, and it is **not** guaranteed unbiased if the linearity assumption fails.
- Neyman allocation is proportional to $p_j\sigma_j$ — not to p_j alone, and not inversely to σ_j.
- Conditional expectation **reduces** variance: $\mathrm{Var}[E[Y|X]] \leq \mathrm{Var}[\bar{Y}]$.
- A copula's marginals are **uniform**, not normal.
- `MetropolisHastingsMV` uses **batch** statistics — the states are dependent.

Appendix — Drill Answers

1. $B \approx 40n = 40 \times 25 = \mathbf{1000}$.
2. $V_{CRN} = 25 - 2(0.5)(4)(3) = 25 - 12 = \mathbf{13}$.
3. $\mathrm{Var}(Z) = \mathrm{Var}(Y)(1 - 0.36) = \mathbf{0.64}\,\mathrm{Var}(\mathbf{Y})$ — a 36% reduction. The condition is $\rho^2 \geq 1/(n-2) = 1/18 = 0.056$, and $0.36 \gg 0.056$, so **yes**.
4. $\mathrm{Var}(\hat{Y}_j) = 8(1 - 0.5)/2 = \mathbf{2}$, versus $8/2 = 4$ for an independent pair — the negative correlation halves it.
5. Symmetric proposal, so the proposal ratio is 1 and $\rho = 0.4/0.5 = 0.8$; $\alpha = \min(0.8, 1) = \mathbf{0.8}$.