Module 9 — Computational Statistics

Statistics

hbar.university

Numerical Optimization

Position in Knowledge Graph

System: Statistics

Structural Domain: Computational Statistics

Node: 9.1

Maximum likelihood estimation, M-estimation, and posterior mode computation all require maximizing an objective function $Q(\theta)$ over $\theta \in \Theta \subseteq \mathbb{R}^k$. Closed-form solutions exist only in special cases (exponential families with canonical sufficient statistics). This node develops the principal iterative algorithms for numerical optimization in statistics, their convergence properties, and the structural role of curvature information.

Formal Definition

Let $Q : \Theta \to \mathbb{R}$ be a twice continuously differentiable objective function. A point $\theta^* \in \Theta$ is a **stationary point** if $\nabla Q(\theta^*) = 0$. It is a **local maximizer** if, additionally, the Hessian $\nabla^2 Q(\theta^*)$ is negative definite.

An **iterative optimization algorithm** generates a sequence $\theta^{(0)}, \theta^{(1)}, \theta^{(2)}, \ldots$ such that $Q(\theta^{(t+1)}) \geq Q(\theta^{(t)})$ (ascent property) and $\theta^{(t)} \to \theta^*$ under appropriate conditions.

The **gradient ascent** update is

$$\theta^{(t+1)} = \theta^{(t)} + \alpha_t \nabla Q(\theta^{(t)}),$$

where $\alpha_t > 0$ is the step size. The **Newton–Raphson** update is

$$\theta^{(t+1)} = \theta^{(t)} - \left[\nabla^2 Q(\theta^{(t)})\right]^{-1} \nabla Q(\theta^{(t)}).$$

Fisher scoring replaces the Hessian with its expectation under the model, yielding

$$\theta^{(t+1)} = \theta^{(t)} + \mathcal{I}(\theta^{(t)})^{-1} \nabla \ell(\theta^{(t)}),$$

where $\mathcal{I}(\theta) = \mathbb{E}_\theta[-\nabla^2 \ell(\theta)]$ is the expected Fisher information.

Intuition

Gradient ascent moves uphill along the steepest direction but treats all directions equally, ignoring the curvature of the objective surface. Newton–Raphson fits a local quadratic approximation and jumps directly to its maximum, which is why it converges much faster near the optimum. The price is computing and inverting the Hessian matrix at each step, which is $O(k^3)$ in parameter dimension.

Fisher scoring is the statistically natural variant: it replaces the observed curvature (which fluctuates with the data) by the expected curvature (which depends only on the model), producing updates that are equivariant under reparameterization. For generalized linear models, Fisher scoring reduces to iteratively reweighted least squares (IRLS).

Structural Role

Numerical optimization is the computational substrate of all likelihood-based and M-estimation inference. Newton–Raphson is the canonical algorithm for computing the MLE; Fisher scoring is its information-geometric counterpart. The convergence rates of these algorithms are controlled by the curvature of the log-likelihood, which is itself the Fisher information (Node 2.5). This node is prerequisite to the EM algorithm (Node 9.5), which replaces a single intractable optimization with a sequence of tractable ones.

Mathematical Development

Convergence of Gradient Ascent

Suppose Q is L-smooth (i.e., $\|\nabla^2 Q(\theta)\| \le L$ for all θ) and m-strongly concave (i.e., $\nabla^2 Q(\theta) \preceq -mI$ for all θ, with $m > 0$). With fixed step size $\alpha = 1/L$,

$$\|\theta^{(t+1)} - \theta^*\| \le \left(1 - \frac{m}{L}\right) \|\theta^{(t)} - \theta^*\|.$$

This is **linear (geometric) convergence** with rate $\kappa = 1 - m/L$, where L/m is the **condition number** of the problem. Poor conditioning ($L \gg m$) produces slow convergence.

Line Search

The **Armijo backtracking** line search selects α_t as the largest value in $\{\beta^0, \beta^1, \beta^2, \dots\}$ (for shrinkage factor $0 < \beta < 1$) satisfying

$$Q(\theta^{(t)} + \alpha_t d^{(t)}) \ge Q(\theta^{(t)}) + c\,\alpha_t \nabla Q(\theta^{(t)})^\top d^{(t)},$$

where $d^{(t)}$ is the search direction and $0 < c < 1$. This guarantees sufficient increase at each step.

Newton–Raphson Convergence

Theorem (Local Quadratic Convergence). Let Q be three times continuously differentiable in a neighborhood of θ^* with $\nabla^2 Q(\theta^*)$ nonsingular. Then there exists $\delta > 0$ such that for $\|\theta^{(0)} - \theta^*\| < \delta$,

$$\|\theta^{(t+1)} - \theta^*\| \le C \|\theta^{(t)} - \theta^*\|^2,$$

where $C = \frac{1}{2}\|\nabla^2 Q(\theta^*)^{-1}\| \sup_\theta \|\nabla^3 Q(\theta)\|$.

Proof sketch. Taylor-expand $\nabla Q(\theta^{(t)})$ around θ^*:

$$\nabla Q(\theta^{(t)}) = \nabla^2 Q(\theta^*)(\theta^{(t)} - \theta^*) + O(\|\theta^{(t)} - \theta^*\|^2).$$

The Newton step gives $\theta^{(t+1)} - \theta^* = \theta^{(t)} - \theta^* - [\nabla^2 Q(\theta^{(t)})]^{-1}\nabla Q(\theta^{(t)})$. Substituting the Taylor expansion and using continuity of $\nabla^2 Q$, the linear term cancels, leaving only the quadratic remainder. □

Trust Regions

When Newton–Raphson is initialized far from θ^*, the Hessian may not be negative definite and the Newton step may overshoot. A **trust region** method restricts the step to a ball:

$$\theta^{(t+1)} = \arg\max_{\|\delta\| \le \Delta_t} \left[\nabla Q(\theta^{(t)})^\top \delta + \frac{1}{2}\delta^\top \nabla^2 Q(\theta^{(t)})\delta \right],$$

where the trust radius Δ_t is adapted based on how well the quadratic model predicts the actual change in Q.

Profile Likelihood

For a partitioned parameter $\theta = (\psi, \lambda)$ where λ is a nuisance parameter, the **profile log-likelihood** is

$$\ell_p(\psi) = \max_{\lambda} \ell(\psi, \lambda) = \ell(\psi, \hat{\lambda}_\psi),$$

where $\hat{\lambda}_\psi$ is the MLE of λ at fixed ψ. Optimization of ℓ_p over ψ yields the marginal MLE, reducing dimension at the cost of iterative inner optimization. The curvature of ℓ_p provides adjusted standard errors for ψ.

Quasi-Newton Methods

When the Hessian is expensive to compute, **quasi-Newton methods** build an approximation $B_t \approx -\nabla^2 Q(\theta^{(t)})$ from successive gradient evaluations. The **BFGS update** maintains a positive definite approximation:

$$B_{t+1} = B_t + \frac{y_t y_t^\top}{y_t^\top s_t} - \frac{B_t s_t s_t^\top B_t}{s_t^\top B_t s_t},$$

where $s_t = \theta^{(t+1)} - \theta^{(t)}$ and $y_t = \nabla Q(\theta^{(t+1)}) - \nabla Q(\theta^{(t)})$. BFGS achieves **superlinear convergence** – faster than linear gradient descent but slower than quadratic Newton – while requiring only gradient evaluations. The limited-memory variant (L-BFGS) stores only the most recent m pairs (s_t, y_t), making it feasible for high-dimensional problems.

Convergence Criteria

Standard termination criteria include:

1. **Gradient norm:** $\|\nabla Q(\theta^{(t)})\| < \varepsilon_g$.
2. **Relative change:** $|Q(\theta^{(t+1)}) - Q(\theta^{(t)})| / (|Q(\theta^{(t)})| + 1) < \varepsilon_f$.
3. **Parameter change:** $\|\theta^{(t+1)} - \theta^{(t)}\| < \varepsilon_\theta$.

In practice, all three are monitored simultaneously. For MLE computation, typical tolerances are $\varepsilon \sim 10^{-8}$.

Worked Examples

Example 1: Newton–Raphson for Gamma MLE

Let $X_1, \ldots, X_n \overset{\text{i.i.d.}}{\sim} \text{Gamma}(\alpha, \beta)$ with density $f(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}$. The score equations are

$$\frac{\partial \ell}{\partial \alpha} = n \log \beta - n\psi(\alpha) + \sum_{i=1}^{n} \log x_i = 0, \quad \frac{\partial \ell}{\partial \beta} = \frac{n\alpha}{\beta} - \sum_{i=1}^{n} x_i = 0,$$

where $\psi(\alpha) = \Gamma'(\alpha)/\Gamma(\alpha)$ is the digamma function. From the second equation, $\hat{\beta} = n\alpha / \sum x_i$. Substituting into the first gives a single equation in α that requires Newton–Raphson iteration:

$$\alpha^{(t+1)} = \alpha^{(t)} - \frac{n \log(\alpha^{(t)}/\bar{x}) - n\psi(\alpha^{(t)}) + \sum \log x_i}{n/\alpha^{(t)} - n\psi'(\alpha^{(t)})}.$$

For data with $n = 50$, $\bar{x} = 3.2$, $\sum \log x_i = 48.7$, initializing at $\alpha^{(0)} = (\bar{x})^2/s^2$ (method of moments), convergence to six decimal places is typical in 4–5 iterations.

Example 2: Fisher Scoring in Logistic Regression

For logistic regression with $P(Y_i = 1 \mid x_i) = \mu_i(\beta) = (1 + e^{-x_i^\top \beta})^{-1}$, the score is

$$\nabla \ell(\beta) = X^\top (y - \mu(\beta)),$$

and the expected Fisher information is

$$\mathcal{I}(\beta) = X^\top W(\beta) X, \quad W = \text{diag}(\mu_i(1 - \mu_i)).$$

Fisher scoring (equivalently, IRLS) iterates

$$\beta^{(t+1)} = (X^\top W^{(t)} X)^{-1} X^\top W^{(t)} z^{(t)},$$

where the working response is $z^{(t)} = X\beta^{(t)} + (W^{(t)})^{-1}(y - \mu^{(t)})$. For a dataset with $n = 100$, $p = 3$, convergence is typically achieved in 5–8 iterations.

Cross-Links

- **Linear Algebra:** Newton–Raphson requires solving a $k \times k$ linear system at each step; the condition number of the Hessian controls convergence rate.
- **AI:** Adam and other adaptive optimizers approximate second-order information using diagonal or low-rank Hessian estimates; natural gradient descent uses the Fisher information matrix as the Riemannian metric.
- **Economics:** Structural estimation in econometrics relies on Newton–Raphson for GMM and maximum likelihood of DSGE models where closed-form solutions are unavailable.
- **Quantum:** Variational quantum eigensolvers use gradient descent on a parameterized quantum circuit; the quantum Fisher information matrix governs the geometry of the parameter space.

Structural Compression

Invariant: The MLE is a fixed point of the score equation $\nabla \ell(\theta) = 0$; Newton–Raphson converges quadratically to this fixed point when initialized in a sufficiently small neighborhood.

Mechanism: Second-order Taylor expansion of Q around the current iterate; the Newton step solves the resulting quadratic approximation exactly. The Hessian encodes local curvature, collapsing the iterative process to a single step for exactly quadratic objectives.

Cross-System Echo: Newton–Raphson is the natural gradient method in the information geometry of the statistical model, where the Fisher information matrix defines the Riemannian metric on the parameter manifold. The tension between first-order methods (cheap but slow) and second-order methods (expensive but fast) recurs in every optimization discipline: gradient descent versus Newton in statistics, SGD versus K-FAC in deep learning, steepest descent versus conjugate gradients in numerical linear algebra.

Exercises

1. Derive the Fisher scoring update for a Poisson regression model with canonical log link and verify that it reduces to an IRLS iteration.
2. For the normal location model $X_i \sim \mathcal{N}(\mu, 1)$, show that Newton–Raphson converges in exactly one step from any initialization. Explain why.
3. Prove that gradient ascent with step size $\alpha = 1/L$ on an L-smooth, m-strongly concave function satisfies $Q(\theta^*) - Q(\theta^{(t)}) \leq (L/2)(1 - m/L)^t \|\theta^{(0)} - \theta^*\|^2$.
4. Consider the Cauchy location model $X_i \sim \text{Cauchy}(\theta, 1)$. Write the Newton–Raphson update for the MLE. Explain why convergence can fail without a good initialization.
5. Show that for an exponential family in canonical form, the observed and expected Fisher information coincide, so Newton–Raphson and Fisher scoring produce identical iterates.
6. Implement backtracking line search for the log-likelihood of a gamma model. Describe how the step size adapts across iterations as the algorithm approaches the MLE.
7. Argue, structurally, why the condition number L/m of the Hessian plays the same role in optimization convergence that the variance ratio plays in statistical efficiency: both quantify how much information the available data (or curvature) provides per unit of computational (or sampling) effort.

Monte Carlo Integration

Position in Knowledge Graph

System: Statistics

Structural Domain: Computational Statistics

Node: 9.2

Monte Carlo integration converts the analytical problem of evaluating an integral into the statistical problem of estimating a mean. It inherits all the tools of statistical inference – the law of large numbers for consistency, the central limit theorem for error quantification, and variance reduction techniques for efficiency. This node develops the method, its theoretical foundations, and the principal variance reduction strategies.

Formal Definition

Let $h : \mathcal{X} \to \mathbb{R}$ be a measurable function and P a probability distribution on $\mathcal{X}$. The target is

$$I = \mathbb{E}_P[h(X)] = \int h(x)\, dP(x).$$

Given $X_1, \ldots, X_n \overset{\text{i.i.d.}}{\sim} P$, the **Monte Carlo estimator** is

$$\hat{I}_n = \frac{1}{n} \sum_{i=1}^{n} h(X_i).$$

By the Strong Law of Large Numbers, $\hat{I}_n \overset{a.s.}{\to} I$ provided $\mathbb{E}_P[|h(X)|] < \infty$. By the Central Limit Theorem,

$$\sqrt{n}(\hat{I}_n - I) \overset{d}{\to} \mathcal{N}(0, \sigma_h^2), \quad \sigma_h^2 = \mathrm{Var}_P(h(X)),$$

so the Monte Carlo standard error is $\mathrm{SE}(\hat{I}_n) = \sigma_h / \sqrt{n}$.

Intuition

The convergence rate $n^{-1/2}$ is dimension-free. Deterministic quadrature in d dimensions converges at rate $n^{-r/d}$ for r-smooth integrands, suffering the curse of dimensionality. Monte Carlo does not: the rate is $n^{-1/2}$ whether $d = 1$ or $d = 10{,}000$. The constant σ_h depends on the integrand, but the rate does not depend on the dimension of the sample space. This makes Monte Carlo the default method for high-dimensional integration in Bayesian statistics, statistical physics, and finance.

Structural Role

Monte Carlo integration is the foundation for MCMC (Node 9.3), which extends the idea to non-i.i.d. samples from intractable distributions. It underlies the bootstrap (Node 9.4), which simulates the sampling distribution by resampling. Bayesian predictive inference (Node 6.5) uses Monte Carlo averages to compute posterior predictive quantities. The variance reduction techniques developed here reappear in importance-weighted MCMC and particle filtering.

Mathematical Development

Importance Sampling

When direct sampling from P is impossible or inefficient, let Q be a **proposal distribution** with $P \ll Q$. Define the importance weight $w(x) = p(x)/q(x)$. Then

$$I = \int h(x)\frac{p(x)}{q(x)}q(x)\,dx = \mathbb{E}_Q[h(X)w(X)].$$

Given $X_1, \ldots, X_n \overset{\text{i.i.d.}}{\sim} Q$, the **importance sampling estimator** is

$$\hat{I}_n^{\text{IS}} = \frac{1}{n}\sum_{i=1}^{n} h(X_i)w(X_i).$$

This is unbiased for I with variance

$$\text{Var}_Q(\hat{I}_n^{\text{IS}}) = \frac{1}{n}\text{Var}_Q(h(X)w(X)) = \frac{1}{n}\left[\int \frac{(h(x)p(x))^2}{q(x)}\,dx - I^2\right].$$

Optimal proposal. The variance-minimizing proposal is $q^*(x) \propto |h(x)|p(x)$, which gives zero variance when $h \geq 0$. In practice, q^* is unknown (it requires the normalizing constant of $|h|p$), but it guides the design of good proposals: q should be large where $|h(x)|p(x)$ is large.

Self-Normalized Importance Sampling

When p is known only up to a normalizing constant, $p(x) = \tilde{p}(x)/Z$, define unnormalized weights $\tilde{w}(x) = \tilde{p}(x)/q(x)$. The **self-normalized estimator** is

$$\hat{I}_n^{\text{SN}} = \frac{\sum_{i=1}^n h(X_i)\tilde{w}(X_i)}{\sum_{i=1}^n \tilde{w}(X_i)}.$$

This is biased but consistent, with bias $O(n^{-1})$. Its asymptotic variance depends on the **effective sample size**:

$$\text{ESS} = \frac{\left(\sum_{i=1}^n \tilde{w}_i\right)^2}{\sum_{i=1}^n \tilde{w}_i^2} \approx \frac{n}{1 + \text{Var}_Q(w)}.$$

When $Q = P$, all weights are equal and $\text{ESS} = n$. When Q is a poor match, a few weights dominate and $\text{ESS} \ll n$.

Control Variates

Let $g : \mathcal{X} \to \mathbb{R}$ with known expectation $\mu_g = \mathbb{E}_P[g(X)]$. The **control variate estimator** is

$$\hat{I}_n^{\text{CV}} = \frac{1}{n}\sum_{i=1}^{n}\left[h(X_i) - c(g(X_i) - \mu_g)\right].$$

This is unbiased for any c. The variance is

$$\text{Var}(\hat{I}_n^{\text{CV}}) = \frac{1}{n}\left[\sigma_h^2 - 2c\,\text{Cov}(h, g) + c^2\sigma_g^2\right].$$

The optimal coefficient is $c^* = \mathrm{Cov}(h, g)/\mathrm{Var}(g)$, yielding variance reduction factor $1 - \rho_{hg}^2$, where ρ_{hg} is the correlation between h and g. The closer $|\rho_{hg}|$ is to 1, the greater the reduction.

Antithetic Variates

For symmetric distributions, generate pairs (X_i, X_i') that are negatively correlated (e.g., $X_i' = -X_i$ for a distribution symmetric about 0). The estimator

$$\hat{I}_n^{\mathrm{AV}} = \frac{1}{n} \sum_{i=1}^{n/2} \frac{h(X_i) + h(X_i')}{2}$$

has variance reduced by $\mathrm{Cov}(h(X), h(X'))/\mathrm{Var}(h(X))$, which is negative when h is monotone.

Stratified Sampling

Partition $\mathcal{X} = \bigcup_{j=1}^{J} A_j$ with $P(A_j) = p_j$. Draw n_j samples from $P(\cdot \mid A_j)$ and estimate

$$\hat{I}_n^{\mathrm{strat}} = \sum_{j=1}^{J} p_j \hat{I}_{n_j}^{(j)}, \quad \hat{I}_{n_j}^{(j)} = \frac{1}{n_j} \sum_{i=1}^{n_j} h(X_i^{(j)}).$$

With proportional allocation $n_j = np_j$, the variance is

$$\mathrm{Var}(\hat{I}_n^{\mathrm{strat}}) = \frac{1}{n} \sum_{j=1}^{J} p_j \sigma_j^2 \leq \frac{\sigma_h^2}{n},$$

where $\sigma_j^2 = \mathrm{Var}(h(X) \mid X \in A_j)$. The inequality is strict whenever $\mathbb{E}[h(X) \mid A_j]$ varies across strata, because stratification eliminates the between-stratum variance component.

Proof of variance reduction. By the law of total variance,

$$\mathrm{Var}(h(X)) = \mathbb{E}[\mathrm{Var}(h \mid \mathrm{stratum})] + \mathrm{Var}(\mathbb{E}[h \mid \mathrm{stratum}]) = \sum_j p_j \sigma_j^2 + \sum_j p_j (\mu_j - I)^2,$$

where $\mu_j = \mathbb{E}[h(X) \mid A_j]$. Simple Monte Carlo has variance $\mathrm{Var}(h)/n$; stratified sampling has variance $\sum p_j \sigma_j^2 / n$. The difference is $\sum p_j (\mu_j - I)^2 / n \geq 0$. □

Optimal allocation. Neyman allocation sets $n_j \propto p_j \sigma_j$, minimizing the stratified variance to $(\sum_j p_j \sigma_j)^2 / n$, which is further reduced when strata have unequal variability.

CLT for Monte Carlo Estimators

The central limit theorem provides the basis for confidence intervals. For the standard Monte Carlo estimator:

$$P\left(I \in \left[\hat{I}_n - z_{\alpha/2} \frac{\hat{\sigma}_h}{\sqrt{n}},\ \hat{I}_n + z_{\alpha/2} \frac{\hat{\sigma}_h}{\sqrt{n}} \right] \right) \to 1 - \alpha,$$

where $\hat{\sigma}_h^2 = (n-1)^{-1} \sum (h(X_i) - \hat{I}_n)^2$. The confidence interval width is $O(n^{-1/2})$ regardless of dimension, confirming the dimension-free convergence rate. For importance sampling, the CLT applies to $h(X)w(X)$, but the effective variance may be much larger or smaller than σ_h^2 depending on the proposal quality.

Worked Examples

Example 1: Estimating a Posterior Mean

Let $\theta \sim \text{Gamma}(3,1)$ (prior) and $X \mid \theta \sim \text{Poisson}(\theta)$ with $x = 7$ observed. The posterior is $\theta \mid x \sim \text{Gamma}(10,2)$. The posterior mean is $\mathbb{E}[\theta \mid x] = 10/2 = 5$. To verify by Monte Carlo: draw $\theta_1, \ldots, \theta_n \sim \text{Gamma}(10,2)$ and compute $\hat{I}_n = n^{-1} \sum \theta_i$. With $n = 10{,}000$, the standard error is $\sqrt{10/4}/100 = 0.0158$, giving a 95% confidence interval of approximately $(4.969, 5.031)$.

Example 2: Importance Sampling for Rare-Event Probability

Estimate $P(X > 5)$ where $X \sim \mathcal{N}(0,1)$. Direct Monte Carlo requires enormous n since $P(X > 5) \approx 2.87 \times 10^{-7}$. Use importance sampling with proposal $Q = \mathcal{N}(5,1)$:

$$w(x) = \frac{\phi(x)}{\phi(x-5)} = \exp\left(-5x + \frac{25}{2}\right).$$

The estimator $\hat{p} = n^{-1} \sum_{i=1}^{n} \mathbf{1}(X_i > 5) w(X_i)$ with $X_i \sim \mathcal{N}(5,1)$ concentrates samples in the region of interest. With $n = 1{,}000$, approximately half the draws exceed 5, and the weighted average gives an estimate with coefficient of variation below 0.1, compared to a coefficient of variation exceeding 50 for naive Monte Carlo at the same n.

Cross-Links

- **Linear Algebra:** Stratified sampling exploits the decomposition of total variance into within-stratum and between-stratum components, mirroring the ANOVA decomposition.
- **AI:** Stochastic gradient descent is Monte Carlo estimation of the population risk gradient; mini-batch size controls the variance-computation tradeoff. Policy gradient methods in reinforcement learning use control variates (baselines) to reduce gradient variance.
- **Economics:** Monte Carlo simulation is the standard tool for pricing path-dependent derivatives in finance; importance sampling is used to estimate the probability of extreme portfolio losses (Value-at-Risk).
- **Quantum:** Quantum Monte Carlo methods estimate ground-state properties by sampling from the path integral representation; the sign problem is a variance explosion analogous to importance weight degeneracy.

Structural Compression

Invariant: The Monte Carlo estimator $\hat{I}_n$ is unbiased with variance $\text{Var}(h)/n$, converging at rate $n^{-1/2}$ regardless of the dimension of the sample space.

Mechanism: The SLLN guarantees almost-sure convergence of sample averages to population means; the CLT characterizes the $n^{-1/2}$ Gaussian fluctuations. Variance reduction techniques modify the integrand or sampling scheme to decrease the constant σ_h^2 without changing the rate.

Cross-System Echo: The dimension-free convergence rate of Monte Carlo is structurally identical to the statistical principle that sample means converge at rate $n^{-1/2}$ regardless of the population distribution. In every system where high-dimensional integration appears – posterior computation in statistics, partition function estimation in physics, expected utility computation in economics – Monte Carlo is the universal computational primitive precisely because it circumvents the curse of dimensionality.

Exercises

1. Prove that the importance sampling estimator $\hat{I}_n^{\mathrm{IS}}$ is unbiased and derive its variance in terms of $\mathbb{E}_Q[(h(X)w(X))^2]$.
2. For $h(x) = x^2$ and $P = \mathcal{N}(0, 1)$, compute the optimal importance sampling proposal $q^*(x) \propto x^2\phi(x)$. Identify the distribution and show that the resulting estimator has zero variance.
3. Derive the optimal control variate coefficient $c^* = \mathrm{Cov}(h, g)/\mathrm{Var}(g)$ and show that the variance reduction factor is $1 - \rho_{hg}^2$.
4. Show that stratified sampling with proportional allocation always reduces variance compared to simple random sampling, with equality iff the stratum means are all equal.
5. For self-normalized importance sampling, derive the $O(n^{-1})$ bias using a Taylor expansion of the ratio estimator, and show that the bias vanishes as $n \to \infty$.
6. Estimate $\int_0^1 e^x\,dx$ using Monte Carlo with $n = 1{,}000$ and the control variate $g(x) = x$ (with $\mu_g = 1/2$). Compare the standard error with and without the control variate.
7. Argue, structurally, why the dimension-free convergence rate of Monte Carlo integration represents the same principle as the dimension-free convergence of sample means in the CLT: both are consequences of the fact that averaging independent random variables reduces variance at rate $1/n$, and this rate depends on the number of observations, not the complexity of the underlying space.

Markov Chain Monte Carlo

Position in Knowledge Graph

System: Statistics

Structural Domain: Computational Statistics

Node: 9.3

MCMC extends Monte Carlo integration (Node 9.2) to settings where direct i.i.d. sampling from the target distribution is infeasible. By constructing a Markov chain whose stationary distribution is the target, MCMC converts the problem of sampling into the problem of simulating a chain long enough for ergodic averages to converge. This node develops the Metropolis–Hastings algorithm, proves that detailed balance implies stationarity, derives Gibbs sampling as a special case, and establishes the convergence diagnostic framework.

Formal Definition

Let $\pi(\theta)$ be a target density on $\Theta \subseteq \mathbb{R}^k$, known up to a normalizing constant. A **Markov chain Monte Carlo** method constructs a Markov chain $(\theta^{(0)}, \theta^{(1)}, \theta^{(2)}, \dots)$ with transition kernel $K(\theta, \theta')$ such that:

1. π is a **stationary distribution**: $\int K(\theta, \theta')\pi(\theta)\, d\theta = \pi(\theta')$.
2. The chain is **ergodic** (irreducible, aperiodic, positive recurrent).

Under these conditions, the **ergodic theorem** guarantees: for any h with $\mathbb{E}_\pi[|h|] < \infty$,

$$\frac{1}{T}\sum_{t=1}^{T} h(\theta^{(t)}) \overset{a.s.}{\to} \mathbb{E}_\pi[h(\theta)].$$

A sufficient condition for stationarity is **detailed balance**:

$$\pi(\theta)K(\theta, \theta') = \pi(\theta')K(\theta', \theta) \quad \text{for all } \theta, \theta'.$$

Intuition

MCMC trades the requirement of independent samples for the ability to sample from distributions known only up to a normalizing constant. The chain wanders through parameter space, spending time in regions proportional to their probability under π. The art of MCMC is designing chains that explore the target distribution efficiently – avoiding both slow random walks and proposals that are rejected too often. The burn-in period discards early samples that are influenced by the initialization, and convergence diagnostics assess whether the chain has reached stationarity.

Structural Role

MCMC is the computational engine of modern Bayesian statistics. It extends posterior computation beyond conjugate families (Node 6.2) to arbitrary posteriors, including hierarchical models, nonparametric models, and latent variable models. It depends on Monte Carlo integration (Node 9.2) for the ergodic averaging framework and enables variational inference (Node 9.6) as a faster alternative when MCMC mixing is prohibitively slow. The bootstrap (Node 9.4) is a parallel computational strategy that uses resampling rather than Markov chains.

Mathematical Development

Metropolis–Hastings Algorithm

Let $q(\theta' \mid \theta)$ be a proposal kernel. The **Metropolis–Hastings** algorithm generates the chain as follows:

1. Given $\theta^{(t)}$, draw $\theta' \sim q(\cdot \mid \theta^{(t)})$.
2. Compute the acceptance probability:

$$\alpha(\theta^{(t)}, \theta') = \min\left(1,\ \frac{\pi(\theta')\, q(\theta^{(t)} \mid \theta')}{\pi(\theta^{(t)})\, q(\theta' \mid \theta^{(t)})}\right).$$

3. With probability α, set $\theta^{(t+1)} = \theta'$; otherwise, set $\theta^{(t+1)} = \theta^{(t)}$.

Theorem (Detailed Balance). The Metropolis–Hastings chain satisfies detailed balance with respect to π.

Proof. The transition kernel is

$$K(\theta, \theta') = q(\theta' \mid \theta)\alpha(\theta, \theta') + r(\theta)\delta_\theta(\theta'),$$

where $r(\theta) = 1 - \int q(\theta' \mid \theta)\alpha(\theta, \theta')\, d\theta'$ is the rejection probability. For the continuous part, we must show $\pi(\theta)q(\theta' \mid \theta)\alpha(\theta, \theta') = \pi(\theta')q(\theta \mid \theta')\alpha(\theta', \theta)$.

Without loss of generality, assume $\pi(\theta')q(\theta \mid \theta') \le \pi(\theta)q(\theta' \mid \theta)$. Then $\alpha(\theta, \theta') = \pi(\theta')q(\theta \mid \theta')/[\pi(\theta)q(\theta' \mid \theta)]$ and $\alpha(\theta', \theta) = 1$. Substituting:

$$\pi(\theta)q(\theta' \mid \theta) \cdot \frac{\pi(\theta')q(\theta \mid \theta')}{\pi(\theta)q(\theta' \mid \theta)} = \pi(\theta')q(\theta \mid \theta') \cdot 1. \quad \square$$

Since π is the stationary distribution and the chain is typically irreducible and aperiodic (given mild conditions on q), the ergodic theorem applies.

Special Case: Random Walk Metropolis

When $q(\theta' \mid \theta) = g(\theta' - \theta)$ for a symmetric density g, the acceptance ratio simplifies to

$$\alpha(\theta^{(t)}, \theta') = \min\left(1, \frac{\pi(\theta')}{\pi(\theta^{(t)})}\right).$$

The proposal variance controls the tradeoff: too small yields high acceptance but slow exploration; too large yields frequent rejection. The optimal acceptance rate for a k-dimensional Gaussian target is approximately 0.234 (Roberts, Gelman, and Gilks, 1997).

Gibbs Sampling

When $\theta = (\theta_1, \ldots, \theta_k)$ and the full conditional distributions $\pi(\theta_j \mid \theta_{-j})$ are available, **Gibbs sampling** cycles through:

For $j = 1, \ldots, k$: draw $\theta_j^{(t+1)} \sim \pi(\theta_j \mid \theta_1^{(t+1)}, \ldots, \theta_{j-1}^{(t+1)}, \theta_{j+1}^{(t)}, \ldots, \theta_k^{(t)})$.

Proposition. Gibbs sampling is a special case of Metropolis–Hastings with proposal $q(\theta' \mid \theta) = \pi(\theta'_j \mid \theta_{-j}) \prod_{i \neq j} \delta_{\theta_i}(\theta'_i)$ and acceptance probability identically 1.

Proof. The MH ratio for coordinate j is

$$\frac{\pi(\theta')\, q(\theta \mid \theta')}{\pi(\theta)\, q(\theta' \mid \theta)} = \frac{\pi(\theta'_j, \theta_{-j})\, \pi(\theta_j \mid \theta_{-j})}{\pi(\theta_j, \theta_{-j})\, \pi(\theta'_j \mid \theta_{-j})} = \frac{\pi(\theta_{-j})\pi(\theta'_j \mid \theta_{-j})\, \pi(\theta_j \mid \theta_{-j})}{\pi(\theta_{-j})\pi(\theta_j \mid \theta_{-j})\, \pi(\theta'_j \mid \theta_{-j})} = 1. \quad \square$$

Conjugate priors (Node 6.2) yield tractable full conditionals, making Gibbs the canonical algorithm for hierarchical Bayesian models.

Effective Sample Size

MCMC samples are correlated. The **effective sample size** is

$$\text{ESS} = \frac{T}{1 + 2\sum_{k=1}^{\infty} \rho_k},$$

where $\rho_k = \text{Corr}(h(\theta^{(t)}), h(\theta^{(t+k)}))$ is the lag-k autocorrelation. The Monte Carlo standard error for $\bar{h}_T = T^{-1}\sum_t h(\theta^{(t)})$ is

$$\text{SE}(\bar{h}_T) = \sqrt{\frac{\text{Var}_\pi(h)}{\text{ESS}}}.$$

In practice, the infinite sum is truncated at the first negative autocorrelation or estimated using spectral methods.

Convergence Diagnostics

Gelman–Rubin $\hat{R}$ statistic. Run $M \geq 2$ chains of length T. Let B be the between-chain variance and W the within-chain variance. Define

$$\hat{R} = \sqrt{\frac{(T-1)/T \cdot W + B/T}{W}}.$$

As $T \to \infty$, $\hat{R} \to 1$ if all chains have converged to the stationary distribution. Values $\hat{R} > 1.01$ indicate incomplete convergence.

Trace plots provide visual assessment: a well-mixing chain shows rapid, uncorrelated fluctuations around a stable level. Persistent trends or multimodal behavior indicate convergence failure.

Autocorrelation function (ACF): slow decay of ρ_k indicates poor mixing and low ESS per iteration.

MCMC Central Limit Theorem

Under regularity conditions (geometric ergodicity), the MCMC ergodic average satisfies a CLT:

$$\sqrt{T}(\bar{h}_T - \mathbb{E}_\pi[h]) \xrightarrow{d} \mathcal{N}(0, \sigma^2_{\text{MCMC}}),$$

where the **asymptotic variance** is

$$\sigma^2_{\text{MCMC}} = \text{Var}_\pi(h) \cdot \left(1 + 2\sum_{k=1}^{\infty} \rho_k\right) = \text{Var}_\pi(h) \cdot \frac{T}{\text{ESS}}.$$

The factor $1 + 2\sum \rho_k$ is the **integrated autocorrelation time** (IACT). A well-mixing chain has small IACT and large ESS; a poorly mixing chain has large IACT and ESS much smaller than the nominal sample size T.

Burn-in and Thinning

Burn-in discards the first T_0 samples to reduce sensitivity to initialization. Formally, if the chain converges geometrically, the bias from initialization decays as $O(\rho^{T_0})$, where $\rho < 1$ is the spectral gap.

Thinning retains every m-th sample, reducing storage but not improving ESS per unit of computation. Thinning is generally unnecessary for estimation but may be practical for storage-limited applications.

Worked Examples

Example 1: Metropolis for a Beta Posterior

Let $X \sim \text{Binomial}(20, \theta)$ with $x = 13$ observed, and $\theta \sim \text{Beta}(1, 1)$ (uniform prior). The posterior is $\theta \mid x \sim \text{Beta}(14, 8)$. Use random walk Metropolis with proposal $\theta' = \theta^{(t)} + \varepsilon$, $\varepsilon \sim \mathcal{N}(0, 0.05^2)$, truncated to $[0, 1]$. The acceptance ratio is $\alpha = \min(1, \theta'^{13}(1-\theta')^7 / [\theta^{(t)}]^{13}(1-\theta^{(t)})^7)$. After 10,000 iterations with 1,000 burn-in, the ergodic mean $\bar{\theta} \approx 0.636$ agrees with the exact posterior mean $14/22 \approx 0.636$, and a histogram of the chain matches the $\text{Beta}(14, 8)$ density.

Example 2: Gibbs Sampler for a Bivariate Normal

Let $(X, Y) \sim \mathcal{N}_2(0, \Sigma)$ with $\Sigma = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$. The full conditionals are $X \mid Y = y \sim \mathcal{N}(\rho y, 1 - \rho^2)$ and $Y \mid X = x \sim \mathcal{N}(\rho x, 1 - \rho^2)$. The Gibbs sampler alternates between these. For $\rho = 0.9$, the lag-1 autocorrelation of the X-chain is $\rho^2 = 0.81$, so $\text{ESS} \approx T(1 - 0.81)/(1 + 0.81) = T \cdot 0.105$. For $T = 10{,}000$, $\text{ESS} \approx 1{,}050$. High correlation in the target distribution produces high autocorrelation in the chain, reducing effective sample size.

Cross-Links

- **Linear Algebra:** The mixing rate of a Metropolis chain on a discrete state space is governed by the spectral gap of the transition matrix; fast mixing requires the second-largest eigenvalue to be bounded away from 1.
- **AI:** Langevin MCMC (gradient of log-density added to random walk proposal) is used for sampling from energy-based models; stochastic gradient Langevin dynamics combines MCMC with mini-batch gradient estimation for scalable Bayesian deep learning.
- **Economics:** MCMC is used to estimate posterior distributions of structural parameters in DSGE models and Bayesian VARs where the likelihood is available only numerically.
- **Quantum:** Quantum Monte Carlo methods (path-integral Monte Carlo, diffusion Monte Carlo) use MCMC to sample from the Boltzmann distribution of quantum systems; the sign problem is a fundamental barrier analogous to multimodality in classical MCMC.

Structural Compression

Invariant: A Markov chain satisfying detailed balance with respect to π has π as its stationary distribution; ergodic averages converge almost surely to π-expectations.

Mechanism: The Metropolis acceptance probability is constructed so that detailed balance holds by construction. The ratio form $\pi(\theta')/\pi(\theta)$ cancels the normalizing constant, enabling sampling from distributions known only up to proportionality.

Cross-System Echo: MCMC converts integration into simulation, just as Monte Carlo integration converts integration into averaging. The structural pattern – replacing an analytically intractable global computation with an iterative local computation that converges to the global answer – appears throughout computational

science: relaxation methods in PDE solvers, message passing in graphical models, and simulated annealing in combinatorial optimization are all instances of the same principle.

Exercises

1. Prove that detailed balance $\pi(\theta)K(\theta, \theta') = \pi(\theta')K(\theta', \theta)$ implies π is a stationary distribution of K, by integrating both sides over θ.

2. For a random walk Metropolis targeting $\pi(\theta) \propto e^{-\theta^4}$ on $\mathbb{R}$ with Gaussian proposal variance σ^2, describe qualitatively how the acceptance rate and mixing behavior change as σ^2 varies from 0.01 to 100.

3. Derive the full conditional distributions for a Bayesian normal model $X_i \mid \mu, \sigma^2 \sim \mathcal{N}(\mu, \sigma^2)$ with priors $\mu \sim \mathcal{N}(0, \tau^2)$ and $\sigma^2 \sim \text{Inv-Gamma}(a, b)$. Write the Gibbs sampling algorithm.

4. Show that the effective sample size satisfies $\text{ESS} \leq T$, with equality iff the chain produces independent samples (all autocorrelations are zero).

5. For the Gibbs sampler on a bivariate normal with correlation ρ, prove that the lag-1 autocorrelation of each coordinate is ρ^2, and derive the ESS as a function of T and ρ.

6. Implement the Gelman–Rubin diagnostic for 4 chains of length 5,000 targeting a $\text{Beta}(2, 5)$ distribution. Report $\hat{R}$ and compare it to the threshold 1.01.

7. Argue, structurally, why the normalizing-constant cancellation in the Metropolis ratio is the same structural principle as the sufficiency reduction in classical statistics: both allow inference to proceed without computing a quantity (the marginal likelihood or the normalizing constant) that depends on the full model but not on the parameter of interest.

Bootstrap Implementation

Position in Knowledge Graph

System: Statistics

Structural Domain: Computational Statistics

Node: 9.4

The bootstrap translates the theoretical coverage guarantees of Node 4.6 into a concrete computational algorithm. It implements the plug-in principle: the empirical distribution $\hat{P}_n$ replaces the unknown population P, and resampling from $\hat{P}_n$ simulates the sampling variability of estimators. This node specifies the nonparametric and parametric bootstrap algorithms, develops bias correction, establishes consistency conditions, and identifies failure modes.

Formal Definition

Let $X_1, \ldots, X_n \overset{\text{i.i.d.}}{\sim} P$ and let $\hat{\theta}_n = T(\hat{P}_n)$ be a functional of the empirical distribution $\hat{P}_n = n^{-1} \sum_{i=1}^{n} \delta_{X_i}$.

The **nonparametric bootstrap distribution** of $\hat{\theta}_n$ is the conditional distribution of $\hat{\theta}_n^* = T(\hat{P}_n^*)$, where $\hat{P}_n^*$ is the empirical distribution of a sample drawn with replacement from $\{X_1, \ldots, X_n\}$, conditional on the observed data.

The **bootstrap principle** asserts that the distribution of $\hat{\theta}_n^* - \hat{\theta}_n$ (given the data) approximates the distribution of $\hat{\theta}_n - \theta(P)$ (over the sampling distribution).

Intuition

The bootstrap is the computational realization of a simple idea: if the empirical distribution is close to the true distribution (which Glivenko–Cantelli guarantees), then resampling from the empirical distribution should mimic sampling from the true distribution. The power of the bootstrap is its generality – it requires only the ability to compute $\hat{\theta}_n$ on resampled data, with no need for analytical variance formulas or distributional assumptions. The limitation is that the bootstrap is not magic: it fails when the empirical distribution is a poor approximation to aspects of P that matter for the distribution of $\hat{\theta}_n$.

Structural Role

Bootstrap implementation is the computational companion to the asymptotic theory of Node 4.6. It provides standard errors and confidence intervals for estimators where analytical variance formulas are unavailable or unreliable. The Monte Carlo approximation to the bootstrap distribution uses the framework of Node 9.2. In machine learning, the bootstrap underlies bagging (bootstrap aggregating), which reduces variance in ensemble methods.

Mathematical Development

Nonparametric Bootstrap Algorithm

Given observed data $x_1, \ldots, x_n$:

1. For $b = 1, \ldots, B$:
   - Draw $x_1^{(b)}, \ldots, x_n^{(b)}$ by sampling with replacement from $\{x_1, \ldots, x_n\}$.

- Compute $\hat{\theta}_n^{(b)} = T(\hat{P}_n^{(b)})$.

2. The bootstrap distribution is $\{\hat{\theta}_n^{(1)}, \ldots, \hat{\theta}_n^{(B)}\}$.

Number of resamples. For standard errors: $B = 200$ suffices. For confidence intervals: $B = 1{,}000$ to $2{,}000$. For tail probabilities: $B \geq 5{,}000$.

Parametric Bootstrap

When the model $\{P_\theta : \theta \in \Theta\}$ is specified:

1. Compute $\hat{\theta}_n$ from the data.
2. For $b = 1, \ldots, B$: draw $X_1^{(b)}, \ldots, X_n^{(b)} \overset{\text{i.i.d.}}{\sim} P_{\hat{\theta}_n}$ and compute $\hat{\theta}_n^{(b)}$.

The parametric bootstrap samples from $P_{\hat{\theta}_n}$ rather than $\hat{P}_n$. It achieves higher accuracy when the model is correctly specified, because $P_{\hat{\theta}_n}$ is a smoother (and typically better) estimate of P than $\hat{P}_n$.

Bootstrap Consistency

Theorem (Bootstrap Consistency). Suppose $\sqrt{n}(\hat{\theta}_n - \theta) \xrightarrow{d} \mathcal{N}(0, \sigma^2)$ and σ^2 is a smooth functional of P. Then the bootstrap is consistent:

$$\sup_t \left| P^*\Big(\sqrt{n}(\hat{\theta}_n^* - \hat{\theta}_n) \leq t\Big) - P\Big(\sqrt{n}(\hat{\theta}_n - \theta) \leq t\Big) \right| \xrightarrow{P} 0,$$

where P^* denotes probability under the bootstrap measure.

Proof sketch. By the continuous mapping theorem applied to the functional T and the convergence $\hat{P}_n \to P$ in the Levy–Prokhorov metric, the bootstrap distribution converges to the same Gaussian limit as the sampling distribution. The key condition is that T is Hadamard differentiable at P, which ensures that the linearization $T(\hat{P}_n^*) - T(\hat{P}_n) \approx T'_P(\hat{P}_n^* - \hat{P}_n)$ is valid. □

Bootstrap Standard Error and Confidence Intervals

The bootstrap standard error is

$$\widehat{\mathrm{SE}}_B = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}\left(\hat{\theta}_n^{(b)} - \bar{\theta}^*\right)^2}, \quad \bar{\theta}^* = \frac{1}{B}\sum_{b=1}^{B}\hat{\theta}_n^{(b)}.$$

Percentile interval:

$$C^{\text{perc}} = \left[q^*_{\alpha/2}, q^*_{1-\alpha/2}\right],$$

where q_p^* is the p-quantile of $\{\hat{\theta}_n^{(b)}\}$.

Basic (reverse) interval:

$$C^{\text{basic}} = \left[2\hat{\theta}_n - q^*_{1-\alpha/2}, 2\hat{\theta}_n - q^*_{\alpha/2}\right].$$

Bootstrap-t interval:

$$C^t = \left[\hat{\theta}_n - q^{*,t}_{1-\alpha/2}\,\hat{\sigma}_n,\ \hat{\theta}_n - q^{*,t}_{\alpha/2}\,\hat{\sigma}_n\right],$$

where $q_p^{*,t}$ are quantiles of the studentized bootstrap statistics $(\hat{\theta}_n^{(b)} - \hat{\theta}_n)/\hat{\sigma}_n^{(b)}$. The bootstrap-$t$ achieves second-order accuracy (coverage error $O(n^{-1})$) compared to $O(n^{-1/2})$ for the percentile interval. The bootstrap-t requires a nested computation: for each bootstrap resample, an internal standard error $\hat{\sigma}_n^{(b)}$ must be computed, either analytically or by a second-level bootstrap.

Coverage error comparison. Let C_n denote a nominal $(1-\alpha)$ confidence interval. The coverage error is $|P(\theta \in C_n) - (1-\alpha)|$:

- Normal approximation: $O(n^{-1/2})$.
- Percentile bootstrap: $O(n^{-1/2})$ (same order, different constant).
- Bootstrap-t and BCa: $O(n^{-1})$ (second-order accurate).
- Double bootstrap: $O(n^{-2})$ (third-order accurate).

BCa (Bias-Corrected and Accelerated) Method

The **BCa interval** corrects for both bias and skewness:

$$C^{\mathrm{BCa}} = \left[q^*_{\alpha_1}, q^*_{\alpha_2}\right],$$

where the adjusted quantile levels are

$$\alpha_1 = \Phi\left(z_0 + \frac{z_0 + z_{\alpha/2}}{1 - a(z_0 + z_{\alpha/2})}\right), \quad \alpha_2 = \Phi\left(z_0 + \frac{z_0 + z_{1-\alpha/2}}{1 - a(z_0 + z_{1-\alpha/2})}\right).$$

Here $z_0 = \Phi^{-1}(\#\{\hat{\theta}_n^{(b)} < \hat{\theta}_n\}/B)$ is the bias correction and a is the acceleration constant, estimated from the jackknife:

$$a = \frac{\sum_{i=1}^n (\bar{\theta}_{(\cdot)} - \hat{\theta}_{(-i)})^3}{6\left[\sum_{i=1}^n (\bar{\theta}_{(\cdot)} - \hat{\theta}_{(-i)})^2\right]^{3/2}},$$

where $\hat{\theta}_{(-i)}$ is the estimate with observation i deleted.

When the Bootstrap Fails

The bootstrap fails when $\hat{P}_n$ does not capture the features of P relevant to the distribution of $\hat{\theta}_n$:

1. **Heavy tails without finite variance.** If $\mathrm{Var}_P(X) = \infty$, the bootstrap variance is finite (since $\hat{P}_n$ has finite support) and underestimates the true variability.
2. **Extremes and boundary estimators.** For $\hat{\theta}_n = \max(X_1, \ldots, X_n)$, the bootstrap distribution concentrates on the observed maximum, failing to capture the correct $n(X_{(n)} - \theta)$ scaling.
3. **Non-smooth functionals.** If T is not Hadamard differentiable (e.g., the mode), the bootstrap may be inconsistent.
4. **Dependent data.** The i.i.d. bootstrap is invalid for time series; block bootstrap or sieve bootstrap are required.

Double Bootstrap

For calibrated confidence intervals with coverage error $O(n^{-2})$, the **double bootstrap** uses nested resampling: for each bootstrap sample $b = 1, \ldots, B_1$, generate B_2 second-level bootstrap resamples to estimate the coverage of the first-level interval, then adjust the quantile accordingly.

The double bootstrap is computationally intensive ($B_1 \times B_2$ total evaluations of T) but achieves second-order accuracy without requiring analytical calculation of the acceleration constant a in the BCa method.

Exact Bootstrap Distribution

The exact nonparametric bootstrap distribution averages over all $\binom{2n-1}{n}$ possible resamples (each corresponding to a multinomial draw of size n from n categories). The Monte Carlo bootstrap with B resamples is an approximation to this exact distribution. The Monte Carlo error (from finite B) is distinct from the bootstrap error (from $\hat{P}_n \neq P$); the former is $O(B^{-1/2})$ and can be made arbitrarily small by increasing B, while the latter is an intrinsic statistical limitation.

Residual Bootstrap for Regression

For the linear model $Y = X\beta + \varepsilon$, the **residual bootstrap** resamples the residuals $\hat{e}_i = Y_i - X_i^\top \hat{\beta}$ and constructs bootstrap responses $Y_i^{(b)} = X_i^\top \hat{\beta} + \hat{e}_{\pi(i)}$, where π is a random permutation with replacement. This preserves the fixed design while resampling the error distribution, and is appropriate when the regression structure is assumed correct but the error distribution is unknown.

Worked Examples

Example 1: Bootstrap Standard Error of the Median

Let $x_1, \ldots, x_{20}$ be a sample from an unknown distribution with observed median $\hat{m} = 3.7$. There is no simple formula for $\mathrm{SE}(\hat{m})$ without distributional assumptions. Generate $B = 2{,}000$ bootstrap resamples, compute the median of each, and take the standard deviation: $\widehat{\mathrm{SE}}_B = 0.42$. The 95% percentile interval is $[2.9, 4.5]$. Contrast this with the normal-based interval $\hat{m} \pm 1.96 \cdot \widehat{\mathrm{SE}}_B = [2.88, 4.52]$, which is similar here but would differ for skewed bootstrap distributions.

Example 2: Parametric vs. Nonparametric Bootstrap for Normal Variance

Let $x_1, \ldots, x_{15} \sim \mathcal{N}(5, 4)$ with observed sample variance $s^2 = 3.8$. The exact sampling distribution gives $(n-1)s^2/\sigma^2 \sim \chi^2_{14}$, yielding a 95% CI for σ^2 of $[2.09, 8.44]$.

Nonparametric bootstrap: resample with replacement, compute $s^{2(b)}$ for $B = 2{,}000$ resamples. The percentile interval is approximately $[1.8, 7.1]$ – shorter than the exact interval because the empirical distribution has lighter tails than the normal.

Parametric bootstrap: draw $X^{(b)} \sim \mathcal{N}(\bar{x}, s^2)$, compute $s^{2(b)}$. The percentile interval is approximately $[2.1, 8.3]$, closely matching the exact interval. The parametric bootstrap is superior here because the normality assumption is correct and the parametric model estimates P more accurately than $\hat{P}_n$ for tail behavior.

Example 3: Bootstrap for Correlation Coefficient

Let $(X_i, Y_i)_{i=1}^{30}$ be bivariate data with sample correlation $r = 0.72$. Fisher's z-transformation gives an approximate confidence interval, but the bootstrap provides a nonparametric alternative. Resample pairs $(X_i^{(b)}, Y_i^{(b)})$ with replacement and compute $r^{(b)}$ for $B = 2{,}000$. The bootstrap distribution of r is left-skewed (bounded above by 1). The BCa interval $[0.48, 0.87]$ accounts for this skewness, while the percentile interval $[0.45, 0.86]$ does not. The acceleration constant $a = -0.03$ reflects the negative skewness induced by the upper bound.

Cross-Links

- **Linear Algebra:** The bootstrap covariance matrix of a vector estimator $\hat{\theta}_n \in \mathbb{R}^k$ is the sample covariance of the bootstrap replicates, providing a nonparametric estimate of the asymptotic covariance matrix that would otherwise require the delta method and Fisher information.
- **AI:** Bagging (bootstrap aggregating) generates an ensemble of predictors trained on bootstrap resamples; the variance reduction from averaging parallels the bootstrap's variance estimation. Random forests extend bagging with feature subsampling.
- **Economics:** The bootstrap is the standard tool for inference in structural econometric models where the asymptotic distribution of estimators (GMM, two-stage least squares) depends on nuisance parameters in complex ways.
- **Quantum:** Quantum state tomography uses bootstrap resampling to quantify uncertainty in reconstructed density matrices when analytical error bars are unavailable.

Structural Compression

Invariant: The empirical distribution $\hat{P}_n$ is a consistent estimate of P; resampling from $\hat{P}_n$ consistently estimates the sampling distribution of $T(\hat{P}_n)$ whenever T is sufficiently smooth (Hadamard differentiable).

Mechanism: Drawing with replacement from the observed sample generates a Monte Carlo approximation to the bootstrap distribution; quantiles of this approximation yield confidence interval endpoints. The BCa correction adjusts for bias and skewness, achieving second-order accuracy.

Cross-System Echo: The bootstrap implements the same structural principle as the plug-in estimator: replace the unknown P with its empirical estimate and propagate uncertainty through the functional T. This principle – that consistent estimation of the input yields consistent estimation of smooth functionals of the input – is the computational form of the continuous mapping theorem and appears in every system where simulation replaces analysis.

Exercises

1. Prove that the bootstrap estimate of the mean, $\bar{X}^* = n^{-1}\sum_{i=1}^{n} X_i^*$, has conditional variance $\hat{\sigma}^2/n$ where $\hat{\sigma}^2 = n^{-1}\sum(X_i - \bar{X})^2$, matching the plug-in variance estimator.

2. For the sample variance S^2, derive the bootstrap distribution analytically when $n = 3$ by enumerating all $3^3 = 27$ bootstrap resamples.

3. Show that the percentile bootstrap interval is transformation-respecting: if C^{perc} is an interval for θ, then $g(C^{\text{perc}})$ is a valid interval for $g(\theta)$ for any monotone g.

4. Explain why the bootstrap fails for $\hat{\theta}_n = X_{(n)} = \max(X_1, \ldots, X_n)$ when P is uniform on $[0, \theta]$. What is the correct rate of convergence, and why does the bootstrap miss it?

5. Derive the bias-correction constant z_0 in the BCa method and show that when $z_0 = 0$ and $a = 0$, the BCa interval reduces to the percentile interval.

6. Compare the parametric bootstrap (assuming normality) and the nonparametric bootstrap for the sample median from a skewed distribution. Generate data from Exp(1), $n = 30$, and compare coverage of 95% intervals over 500 simulation replications.

7. Argue, structurally, why the bootstrap's reliance on the plug-in principle means it inherits both the strengths and limitations of empirical distribution estimation: it succeeds whenever the empirical distribution captures the relevant features of P (bulk behavior, smooth functionals) and fails whenever the relevant features are in the tails or boundaries that the empirical distribution cannot resolve with n observations.

Expectation-Maximization Algorithm

Position in Knowledge Graph

System: Statistics

Structural Domain: Computational Statistics

Node: 9.5

The EM algorithm is the canonical method for maximum likelihood estimation in latent variable models, where the observed-data log-likelihood involves an intractable integral over latent variables. It converts this intractable optimization into a sequence of tractable complete-data optimizations by alternating between computing expected sufficient statistics (E-step) and maximizing the complete-data likelihood (M-step). This node derives the ELBO decomposition that justifies EM, proves monotonicity of the likelihood sequence, analyzes convergence rate, and establishes the connection to variational inference.

Formal Definition

Let Y denote observed data and Z denote latent (unobserved) variables. The complete-data log-likelihood is $\ell_c(\theta; Y, Z) = \log p(Y, Z \mid \theta)$. The observed-data log-likelihood is

$$\ell(\theta; Y) = \log p(Y \mid \theta) = \log \int p(Y, Z \mid \theta)\, dZ.$$

The **Q-function** at iteration t is

$$Q(\theta \mid \theta^{(t)}) = \mathbb{E}_{Z \mid Y, \theta^{(t)}}[\log p(Y, Z \mid \theta)] = \int \log p(Y, Z \mid \theta)\, p(Z \mid Y, \theta^{(t)})\, dZ.$$

The **EM algorithm** iterates:

E-step: Compute $Q(\theta \mid \theta^{(t)})$ (or equivalently, the sufficient statistics of $p(Z \mid Y, \theta^{(t)})$).

M-step: Set $\theta^{(t+1)} = \arg\max_\theta Q(\theta \mid \theta^{(t)})$.

Intuition

The observed-data likelihood marginalizes over the latent variables, creating a complex surface with potential local maxima. The EM algorithm avoids directly climbing this surface. Instead, it constructs a simpler surrogate function (the Q-function) that touches the log-likelihood at the current parameter value and lies below it everywhere else. Maximizing this surrogate is guaranteed to increase (or maintain) the log-likelihood. The E-step computes what the latent variables "should" look like given the current parameters; the M-step finds the best parameters given those imputed latent variables.

Structural Role

EM is the bridge between latent variable models and computational tractability. It depends on numerical optimization (Node 9.1) for the M-step, on Bayesian posterior computation (Node 6.1) for the E-step, and on the MLE framework (Node 3.2) for the overall objective. It enables variational inference (Node 9.6) by providing the ELBO framework that VI generalizes. EM is the canonical algorithm for Gaussian mixtures, hidden Markov models, factor analysis, and probabilistic PCA.

Mathematical Development

The ELBO Decomposition

For any distribution $q(Z)$ over the latent variables, we can decompose the observed log-likelihood:

$$\ell(\theta; Y) = \log p(Y \mid \theta) = \log \int p(Y, Z \mid \theta)\, dZ.$$

Introduce $q(Z)$ and apply Jensen's inequality:

$$\ell(\theta; Y) = \log \int \frac{p(Y, Z \mid \theta)}{q(Z)} q(Z)\, dZ \geq \int q(Z) \log \frac{p(Y, Z \mid \theta)}{q(Z)}\, dZ.$$

The right-hand side is the **Evidence Lower BOund (ELBO)**:

$$\mathrm{ELBO}(q, \theta) = \mathbb{E}_q[\log p(Y, Z \mid \theta)] - \mathbb{E}_q[\log q(Z)].$$

The gap between $\ell(\theta; Y)$ and the ELBO is exactly the KL divergence:

$$\ell(\theta; Y) = \mathrm{ELBO}(q, \theta) + \mathrm{KL}(q(Z) \| p(Z \mid Y, \theta)).$$

Proof. Write

$$\mathrm{ELBO}(q, \theta) = \mathbb{E}_q[\log p(Y, Z \mid \theta)] - \mathbb{E}_q[\log q(Z)] = \mathbb{E}_q\left[\log \frac{p(Y, Z \mid \theta)}{q(Z)}\right].$$

Since $p(Y, Z \mid \theta) = p(Z \mid Y, \theta) p(Y \mid \theta)$:

$$\mathrm{ELBO}(q, \theta) = \mathbb{E}_q[\log p(Z \mid Y, \theta)] + \log p(Y \mid \theta) - \mathbb{E}_q[\log q(Z)].$$

Rearranging:

$$\log p(Y \mid \theta) = \mathrm{ELBO}(q, \theta) + \mathbb{E}_q\left[\log \frac{q(Z)}{p(Z \mid Y, \theta)}\right] = \mathrm{ELBO}(q, \theta) + \mathrm{KL}(q \| p(Z \mid Y, \theta)). \quad \square$$

EM as Coordinate Ascent on the ELBO

The EM algorithm maximizes the ELBO by alternating:

E-step (optimize over q at fixed $\theta^{(t)}$): The ELBO is maximized when the KL gap is zero, i.e., $q^{(t)}(Z) = p(Z \mid Y, \theta^{(t)})$. At this q, the ELBO equals $\ell(\theta^{(t)}; Y)$.

M-step (optimize over θ at fixed $q^{(t)}$): Maximize $\mathrm{ELBO}(q^{(t)}, \theta) = \mathbb{E}_{q^{(t)}}[\log p(Y, Z \mid \theta)] + H(q^{(t)})$. Since $H(q^{(t)})$ does not depend on θ, this reduces to maximizing $Q(\theta \mid \theta^{(t)}) = \mathbb{E}_{p(Z \mid Y, \theta^{(t)})}[\log p(Y, Z \mid \theta)]$.

Proof of Monotone Likelihood Ascent

Theorem. $\ell(\theta^{(t+1)}; Y) \geq \ell(\theta^{(t)}; Y)$ for every EM iteration.

Proof. After the E-step, $q^{(t)} = p(Z \mid Y, \theta^{(t)})$, so

$$\ell(\theta^{(t)}; Y) = \mathrm{ELBO}(q^{(t)}, \theta^{(t)}).$$

After the M-step, $\theta^{(t+1)}$ maximizes the ELBO over θ, so

$$\mathrm{ELBO}(q^{(t)}, \theta^{(t+1)}) \geq \mathrm{ELBO}(q^{(t)}, \theta^{(t)}) = \ell(\theta^{(t)}; Y).$$

Since the ELBO is always a lower bound on the log-likelihood:

$$\ell(\theta^{(t+1)}; Y) \geq \mathrm{ELBO}(q^{(t)}, \theta^{(t+1)}) \geq \ell(\theta^{(t)}; Y). \quad \square$$

Convergence Rate

EM converges **linearly** near a local maximum. The rate matrix is

$$M = I_c(\theta^*)^{-1} I_m(\theta^*),$$

where $I_c(\theta^*)$ is the complete-data Fisher information and $I_m(\theta^*) = I_c(\theta^*) - I_{\mathrm{obs}}(\theta^*)$ is the missing information. The convergence rate is the largest eigenvalue of M:

$$\rho = \lambda_{\max}(M) = 1 - \frac{\lambda_{\min}(I_{\mathrm{obs}}(\theta^*))}{\lambda_{\max}(I_c(\theta^*))}.$$

High missingness ($\rho \to 1$) implies slow convergence. When the fraction of missing information is large, the E-step provides little new information and the parameter updates are small.

Convergence to Stationary Points

Under regularity conditions (compactness of Θ, continuity of Q, identifiability), the EM sequence $\{\theta^{(t)}\}$ converges to a stationary point of $\ell(\theta; Y)$. The monotone ascent property, combined with boundedness of ℓ, guarantees that $\ell(\theta^{(t)})$ converges; additional regularity ensures that the parameter sequence converges and the limit satisfies the score equation $\nabla \ell(\theta^*; Y) = 0$. Convergence to a local rather than global maximum is possible; multiple initializations are standard practice.

Worked Examples

Example 1: Gaussian Mixture Model

Let $Y_i \sim \sum_{k=1}^{K} \pi_k \mathcal{N}(\mu_k, \sigma_k^2)$ with latent indicators $Z_i \in \{1, \ldots, K\}$.

E-step. Compute responsibilities:

$$r_{ik}^{(t)} = \frac{\pi_k^{(t)} \phi(Y_i; \mu_k^{(t)}, \sigma_k^{2(t)})}{\sum_{j=1}^{K} \pi_j^{(t)} \phi(Y_i; \mu_j^{(t)}, \sigma_j^{2(t)})}.$$

M-step. Update:

$$n_k = \sum_{i=1}^{n} r_{ik}^{(t)}, \quad \mu_k^{(t+1)} = \frac{\sum_i r_{ik}^{(t)} Y_i}{n_k}, \quad \sigma_k^{2(t+1)} = \frac{\sum_i r_{ik}^{(t)} (Y_i - \mu_k^{(t+1)})^2}{n_k}, \quad \pi_k^{(t+1)} = \frac{n_k}{n}.$$

For a mixture of $K = 2$ components with $n = 200$ observations, initializing with k-means, EM typically converges in 20–50 iterations. The log-likelihood increases monotonically, with the largest increases in the first few iterations.

Example 2: Censored Exponential Data

Let $Y_i \sim \text{Exp}(\lambda)$ with right-censoring at c. For censored observations, $Y_i = c$ and the true value $Z_i > c$ is unobserved. The complete-data log-likelihood is $\ell_c(\lambda) = n \log \lambda - \lambda \sum Z_i$.

E-step. For uncensored observations, $\mathbb{E}[Z_i \mid Y_i, \lambda^{(t)}] = Y_i$. For censored observations ($Y_i = c$),

$$\mathbb{E}[Z_i \mid Z_i > c, \lambda^{(t)}] = c + 1/\lambda^{(t)}.$$

M-step. $\lambda^{(t+1)} = n / \sum_i \tilde{Z}_i^{(t)}$, where $\tilde{Z}_i^{(t)}$ uses the imputed expectations.

With $n = 50$, $\lambda = 2$, and censoring at $c = 0.5$ (about 63% censored), EM converges in 10–15 iterations. The high censoring fraction corresponds to large missing information and slower convergence, consistent with the rate formula.

Cross-Links

- **Linear Algebra:** The M-step for Gaussian mixtures involves weighted least squares computations; the convergence rate matrix M is analyzed through its eigenvalue decomposition.
- **AI:** The EM structure underlies training of latent variable models including Gaussian mixture models for clustering, hidden Markov models for sequence labeling, and probabilistic PCA for dimensionality reduction. At the population level, VAE training optimizes the same ELBO that EM maximizes.
- **Economics:** EM is used in econometrics for models with missing data (e.g., sample selection models), regime-switching models (Hamilton filter), and finite mixture models for unobserved heterogeneity.
- **Quantum:** Quantum state tomography with missing measurement settings uses EM to estimate the density matrix; the latent variables correspond to the unobserved measurement outcomes.

Structural Compression

Invariant: Each EM iteration increases or maintains the observed log-likelihood; the algorithm converges to a stationary point of $\ell(\theta; Y)$.

Mechanism: The ELBO decomposition $\ell(\theta; Y) = \text{ELBO}(q, \theta) + \text{KL}(q \| p(Z \mid Y, \theta))$ shows that EM performs coordinate ascent: the E-step sets $q = p(Z \mid Y, \theta^{(t)})$ to close the KL gap, and the M-step maximizes the ELBO over θ. Since the ELBO is a lower bound that touches ℓ at the current iterate, each M-step guarantees ascent.

Cross-System Echo: EM is a special case of the majorize-maximize (MM) principle: construct a tight lower bound on the objective and maximize the bound. The same principle appears in variational inference (Node 9.6), where the bound is looser because the E-step uses an approximate rather than exact posterior, and in convex optimization, where proximal operators play the role of the M-step. The ELBO is the central object connecting EM, variational inference, and variational autoencoders across statistics and machine learning.

Exercises

1. Derive the ELBO decomposition $\ell(\theta; Y) = \text{ELBO}(q, \theta) + \text{KL}(q \| p(Z \mid Y, \theta))$ from first principles, without using Jensen's inequality, by direct algebraic manipulation.

2. For the two-component Gaussian mixture, write the complete E-step and M-step updates including the variance parameters σ_k^2. Verify that the M-step for π_k involves a Lagrange multiplier enforcing $\sum_k \pi_k = 1$.

3. Prove that the EM convergence rate for the normal mean with known variance and a fraction f of missing observations is $\rho = f$. Interpret this in terms of the missing information principle.

4. Show that Gibbs sampling (Node 9.3) applied to the joint posterior $p(\theta, Z \mid Y)$ produces an ergodic chain, while EM applied to the same model produces a deterministic sequence converging to a point estimate. Explain the structural difference.

5. Implement EM for a $K = 3$ Gaussian mixture on simulated data. Plot the log-likelihood as a function of iteration and verify monotonicity. Run from 5 different initializations and compare the local maxima found.

6. For censored exponential data, derive the E-step formula $\mathbb{E}[Z_i \mid Z_i > c, \lambda] = c + 1/\lambda$ from the memoryless property of the exponential distribution.

7. Argue, structurally, why the ELBO decomposition is the fundamental identity connecting EM, variational inference, and the marginal likelihood: in each case, the decomposition $\log p(Y) = \text{ELBO} + \text{KL}$ determines which quantity is being optimized (the ELBO), what the gap measures (approximation quality), and how the algorithm trades exactness for tractability.

Variational Inference

Position in Knowledge Graph

System: Statistics

Structural Domain: Computational Statistics

Node: 9.6

Variational inference converts the problem of computing an intractable posterior distribution into an optimization problem. Where MCMC (Node 9.3) produces asymptotically exact samples at high computational cost, VI produces a fast deterministic approximation whose quality depends on the expressiveness of the chosen variational family. This node develops the ELBO, the mean-field approximation, the CAVI algorithm, the directional asymmetry of KL divergence, and the extension to stochastic variational inference for large datasets.

Formal Definition

Let $\pi(\theta \mid x) = p(\theta \mid x)$ be a posterior distribution on $\Theta \subseteq \mathbb{R}^k$ that is intractable – either the normalizing constant $m(x) = \int p(x, \theta)\, d\theta$ cannot be computed or MCMC mixing is prohibitively slow.

Let $\mathcal{Q}$ be a family of tractable distributions on Θ. **Variational inference** seeks

$$q^* = \arg\min_{q \in \mathcal{Q}} \mathrm{KL}(q(\theta) \| p(\theta \mid x)).$$

Since $\mathrm{KL}(q \| p(\cdot \mid x))$ involves the unknown normalizing constant $m(x)$, VI instead maximizes the **Evidence Lower BOund (ELBO)**:

$$\mathcal{L}(q) = \mathbb{E}_q[\log p(x, \theta)] - \mathbb{E}_q[\log q(\theta)] = \mathbb{E}_q[\log p(x, \theta)] + H(q),$$

where $H(q) = -\mathbb{E}_q[\log q(\theta)]$ is the entropy of q.

Intuition

Bayesian inference requires integrating over the entire parameter space. MCMC does this by simulating a random walk and averaging. Variational inference replaces the integral with a search: find the distribution in a tractable family that is closest to the true posterior. "Closest" is measured by KL divergence, and the ELBO provides a computable objective that avoids the normalizing constant. The approximation quality depends entirely on whether the variational family $\mathcal{Q}$ can capture the shape of the true posterior – if the posterior is multimodal and $\mathcal{Q}$ contains only unimodal distributions, the approximation will miss important structure.

Structural Role

Variational inference trades exactness for scalability. It depends on the ELBO framework established in Node 9.5 (EM) and serves as the scalable alternative to MCMC (Node 9.3) for large datasets and complex models. EM is the special case where the variational family is unrestricted (the E-step uses the exact posterior over latent variables). Variational inference is the foundation of variational autoencoders (VAEs) and amortized inference in modern machine learning.

Mathematical Development

The ELBO Decomposition

For any $q(\theta)$,

$$\log m(x) = \mathcal{L}(q) + \mathrm{KL}(q(\theta) \| p(\theta \mid x)).$$

Proof. Expanding the KL divergence:

$$\mathrm{KL}(q \| p(\cdot \mid x)) = \mathbb{E}_q\left[\log \frac{q(\theta)}{p(\theta \mid x)}\right] = \mathbb{E}_q[\log q(\theta)] - \mathbb{E}_q[\log p(\theta \mid x)].$$

Since $p(\theta \mid x) = p(x, \theta)/m(x)$:

$$\mathrm{KL}(q \| p(\cdot \mid x)) = \mathbb{E}_q[\log q(\theta)] - \mathbb{E}_q[\log p(x, \theta)] + \log m(x).$$

Rearranging:

$$\log m(x) = \underbrace{\mathbb{E}_q[\log p(x, \theta)] - \mathbb{E}_q[\log q(\theta)]}_{\mathcal{L}(q)} + \mathrm{KL}(q \| p(\cdot \mid x)). \quad \square$$

Since $\mathrm{KL} \geq 0$, $\mathcal{L}(q) \leq \log m(x)$, with equality iff $q = p(\theta \mid x)$. Thus maximizing the ELBO over $\mathcal{Q}$ is equivalent to minimizing $\mathrm{KL}(q \| p(\cdot \mid x))$ over $\mathcal{Q}$.

Alternative Form of the ELBO

The ELBO can be decomposed as

$$\mathcal{L}(q) = \mathbb{E}_q[\log p(x \mid \theta)] - \mathrm{KL}(q(\theta) \| p(\theta)),$$

where $p(\theta)$ is the prior. This form reveals the tradeoff: the first term rewards q for placing mass where the likelihood is large; the second term penalizes q for deviating from the prior.

Mean-Field Approximation

The **mean-field family** assumes full factorization:

$$\mathcal{Q}_{\mathrm{MF}} = \left\{ q(\theta) = \prod_{j=1}^{k} q_j(\theta_j) \right\}.$$

Theorem (Optimal Mean-Field Update). Fixing all factors q_i for $i \neq j$, the optimal q_j satisfies

$$\log q_j^*(\theta_j) = \mathbb{E}_{q_{-j}}[\log p(x, \theta)] + \text{const},$$

where $\mathbb{E}_{q_{-j}}$ denotes expectation over all components except j, and the constant ensures q_j^* normalizes to 1.

Proof. Write the ELBO as

$$\mathcal{L}(q) = \int q_j(\theta_j) \left[\int \prod_{i \neq j} q_i(\theta_i) \log p(x, \theta) \, d\theta_{-j} \right] d\theta_j - \int q_j(\theta_j) \log q_j(\theta_j) \, d\theta_j + C,$$

where C collects terms not depending on q_j. Define $f_j(\theta_j) = \mathbb{E}_{q_{-j}}[\log p(x, \theta)]$. Then

$$\mathcal{L}(q) = \int q_j(\theta_j) f_j(\theta_j)\, d\theta_j - \int q_j(\theta_j) \log q_j(\theta_j)\, d\theta_j + C.$$

This is maximized when $q_j(\theta_j) \propto \exp(f_j(\theta_j))$, since the functional is $-\mathrm{KL}(q_j \| \tilde{q}_j) + \text{const}$ where $\tilde{q}_j \propto \exp(f_j)$. □

Coordinate Ascent Variational Inference (CAVI)

CAVI cycles through coordinates $j = 1, \ldots, k$, updating each q_j according to the optimal update while holding q_{-j} fixed. Each update increases (or maintains) the ELBO, guaranteeing convergence to a local optimum. The algorithm terminates when the relative change in ELBO falls below a threshold.

For exponential family models with conjugate priors, each CAVI update yields a distribution in the same exponential family, and the update reduces to updating the natural parameters:

$$\eta_j^{(t+1)} = \eta_j^{\text{prior}} + \mathbb{E}_{q_{-j}^{(t)}}[t_j(x, \theta_{-j})],$$

where η_j^{prior} are the prior natural parameters and t_j are the sufficient statistics contributed by factor j.

KL Divergence Asymmetry

The choice of KL direction has profound consequences:

KL$(q\|p)$ **(variational inference):** This is **mode-seeking** (or zero-avoiding). Where $p(\theta \mid x) \approx 0$, q must also be approximately 0 (otherwise the KL divergence diverges). Consequently, q tends to concentrate on a single mode of p and underestimates posterior variance.

KL$(p\|q)$ **(expectation propagation):** This is **mean-seeking** (or mass-covering). Where $p(\theta \mid x) > 0$, q must also be positive. Consequently, q spreads to cover all modes, typically overestimating variance.

For a bimodal posterior with modes at $\pm\mu$: - KL$(q\|p)$-optimal Gaussian: concentrates on one mode, $q^* \approx \mathcal{N}(\mu, \sigma^2)$ or $\mathcal{N}(-\mu, \sigma^2)$. - KL$(p\|q)$-optimal Gaussian: centers at 0 with large variance, $q^* \approx \mathcal{N}(0, \mu^2 + \sigma^2)$.

Relation to EM

EM (Node 9.5) is variational inference with $\mathcal{Q} = \{p(Z \mid Y, \theta) : \theta \in \Theta\}$, the family of exact posteriors over latent variables parameterized by θ. The E-step sets $q(Z) = p(Z \mid Y, \theta^{(t)})$ (exact posterior, KL gap = 0), and the M-step optimizes θ. Variational EM replaces the exact E-step with an approximate one: $q(Z) \in \mathcal{Q}_{\text{approx}}$, enabling inference when the E-step is itself intractable.

Stochastic Variational Inference

For large datasets $x = (x_1, \ldots, x_n)$, CAVI requires a full pass through the data at each update. **Stochastic variational inference (SVI)** exploits the fact that the ELBO decomposes as a sum over data points (for conditionally i.i.d. models):

$$\mathcal{L}(q) = \sum_{i=1}^{n} \mathbb{E}_q[\log p(x_i \mid \theta)] - \mathrm{KL}(q(\theta) \| p(\theta)).$$

SVI replaces the full-data gradient with a noisy mini-batch estimate:

$$\hat{\nabla}\mathcal{L} = \frac{n}{|S|} \sum_{i \in S} \nabla_\lambda \mathbb{E}_q[\log p(x_i \mid \theta)] - \nabla_\lambda \mathrm{KL}(q_\lambda \| p),$$

where S is a random mini-batch and λ parameterizes q. With a Robbins–Monro step size schedule ($\sum \alpha_t = \infty$, $\sum \alpha_t^2 < \infty$), SVI converges to a local optimum of the ELBO.

Worked Examples

Example 1: Mean-Field VI for Normal-Normal Model

Let $x_1, \ldots, x_n \mid \mu, \tau \sim \mathcal{N}(\mu, \tau^{-1})$ with priors $\mu \sim \mathcal{N}(0, \sigma_0^2)$ and $\tau \sim \text{Gamma}(a_0, b_0)$. The mean-field approximation is $q(\mu, \tau) = q_\mu(\mu) q_\tau(\tau)$.

Update for q_μ:

$$\log q_\mu^*(\mu) = \mathbb{E}_{q_\tau}\left[-\frac{\mathbb{E}[\tau]}{2} \sum_{i=1}^{n} (x_i - \mu)^2 - \frac{\mu^2}{2\sigma_0^2} \right] + \text{const.}$$

This is a quadratic in μ, so $q_\mu^* = \mathcal{N}(\mu_q, \sigma_q^2)$ with

$$\sigma_q^2 = (n\mathbb{E}_{q_\tau}[\tau] + 1/\sigma_0^2)^{-1}, \quad \mu_q = \sigma_q^2 \cdot n\mathbb{E}_{q_\tau}[\tau]\bar{x}.$$

Update for q_τ:

$$q_\tau^* = \text{Gamma}\left(a_0 + n/2,\ b_0 + \frac{1}{2} \sum_{i=1}^{n} \mathbb{E}_{q_\mu}[(x_i - \mu)^2] \right).$$

CAVI alternates these updates until the ELBO converges. With $n = 100$, $\bar{x} = 2.1$, convergence is typically achieved in 10–20 iterations. The mean-field approximation underestimates the posterior covariance between μ and τ (which is zero in the factorized approximation but nonzero in the true posterior).

Example 2: KL Direction Comparison

Consider a posterior $p(\theta \mid x) = 0.5\,\mathcal{N}(-3, 0.5^2) + 0.5\,\mathcal{N}(3, 0.5^2)$ and restrict $\mathcal{Q}$ to Gaussians $q = \mathcal{N}(m, s^2)$.

Minimizing $\text{KL}(q\|p)$: the optimizer selects one mode, yielding $q^* \approx \mathcal{N}(3, 0.5^2)$ (or symmetrically $\mathcal{N}(-3, 0.5^2)$). The KL divergence penalizes placing mass where p is near zero (the region between modes), so q collapses to a single mode.

Minimizing $\text{KL}(p\|q)$: the optimizer must cover both modes, yielding $q^* \approx \mathcal{N}(0, 9.25)$. This captures both modes but vastly overestimates the spread, assigning substantial mass to the region between modes where p is nearly zero.

Cross-Links

- **Linear Algebra:** The CAVI updates for Gaussian mean-field approximations reduce to solving linear systems; the natural gradient of the ELBO with respect to the natural parameters of an exponential family q is the Euclidean gradient in the natural parameterization.
- **AI:** Variational autoencoders (VAEs) use amortized variational inference: an encoder network maps each data point x_i to variational parameters λ_i, optimizing the ELBO jointly over the encoder and decoder. The reparameterization trick enables backpropagation through the sampling step.
- **Economics:** Variational Bayes is increasingly used in macroeconometrics for approximating posterior distributions of high-dimensional VAR models where MCMC is too slow for real-time forecasting.
- **Quantum:** Variational quantum algorithms (VQE, QAOA) optimize a parameterized quantum circuit to minimize an energy functional, mirroring the VI principle of optimizing over a tractable family to approximate an intractable distribution.

Structural Compression

Invariant: $\log m(x) = \mathcal{L}(q) + \mathrm{KL}(q \| p(\theta \mid x))$: maximizing the ELBO over $\mathcal{Q}$ minimizes the KL divergence to the posterior, with the gap measuring the approximation quality.

Mechanism: Jensen's inequality applied to the log marginal likelihood yields the ELBO as a lower bound. The mean-field factorization restricts $\mathcal{Q}$ to product distributions, enabling coordinate-wise closed-form updates (CAVI). Each CAVI step increases the ELBO, converging to a local optimum. The mode-seeking direction $\mathrm{KL}(q\|p)$ produces compact approximations that may miss posterior mass, trading coverage for computational tractability.

Cross-System Echo: The ELBO decomposition is the master identity of approximate inference. In EM, the KL gap is zero by construction (exact E-step). In VI, the gap is nonzero but controlled. In VAEs, the ELBO is the training objective, with the encoder parameterizing q and the decoder parameterizing $p(x \mid \theta)$. In information theory, the ELBO is the negative description length under code q; minimizing KL is equivalent to minimizing the excess bits required to describe the data. The pattern – bounding an intractable quantity by a tractable one and optimizing the bound – is the universal computational strategy for problems where exact solutions are infeasible.

Exercises

1. Derive the ELBO decomposition $\log m(x) = \mathcal{L}(q) + \mathrm{KL}(q\|p(\theta \mid x))$ and verify that the ELBO equals the log marginal likelihood when $q = p(\theta \mid x)$.
2. For the normal-normal model with known variance, derive the mean-field CAVI updates for q_μ and q_τ and show that each update is a closed-form exponential family distribution.
3. Prove that CAVI monotonically increases the ELBO at each coordinate update. State precisely what regularity conditions are needed.
4. For a bimodal target $p = 0.5\mathcal{N}(-\mu, 1) + 0.5\mathcal{N}(\mu, 1)$, compute the $\mathrm{KL}(q\|p)$-optimal and $\mathrm{KL}(p\|q)$-optimal Gaussian approximations analytically as functions of μ. Plot both as μ ranges from 0 to 5.
5. Show that EM is a special case of variational inference where the E-step uses the unrestricted family $\mathcal{Q} = \{q(Z) : q \text{ is a density}\}$, so the KL gap is zero after the E-step.
6. For a model with $n = 10{,}000$ observations, estimate the per-iteration cost of CAVI versus SVI with mini-batch size 100. Explain the tradeoff between noise in the gradient and computational savings.
7. Argue, structurally, why the ELBO is the unifying object across EM, variational inference, and variational autoencoders: in each case, the same decomposition $\log p(x) = \mathrm{ELBO} + \mathrm{KL}$ determines the objective, the approximation gap, and the algorithm, with the only difference being the expressiveness of the variational family and whether the parameters θ are integrated out or optimized.