Bayesian Linear Regression

Vasilis Gkolemis

ATHENA RC — HUA

June 2025

Logistic Regression as a Probabilistic Model

- In binary classification, we model the probability of label $y \in \{0, 1\}$ given input $\mathbf{x} \in \mathbb{R}^d$:

$$p(y = 1 \mid \mathbf{x}, \mathbf{w}) = \sigma(\mathbf{w}^\top \mathbf{x})$$

- $\sigma(z) = \frac{1}{1+e^{-z}}$ is the logistic sigmoid function.
- In a Bayesian setting, we place a prior over $\mathbf{w}$ and infer the posterior:

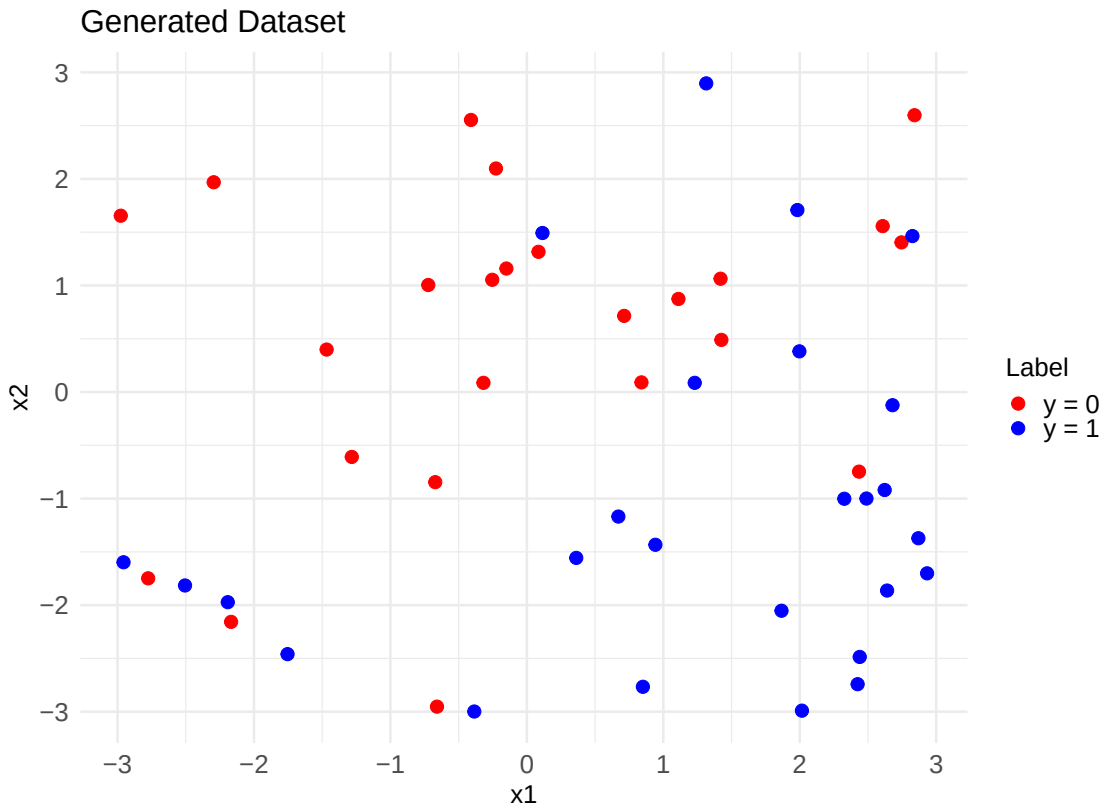
$$p(\mathbf{w} \mid \mathcal{D}) \propto p(\mathbf{w}) \prod_{n=1}^{N} p(y_n \mid \mathbf{x}_n, \mathbf{w})$$

Why Approximate Inference?

- The posterior is not analytically tractable due to the non-conjugate likelihood.
- The logistic sigmoid does not lead to a conjugate posterior with a Gaussian prior.
- Approximate inference methods are needed:
  - Laplace Approximation
  - Importance Sampling
  - Markov Chain Monte Carlo (MCMC)

2D Toy Example

- We create a small dataset with $N = 50$ samples:
  - $\mathbf{x}_n \in [-3, 3]^2$, $y_n \sim \text{Bernoulli}(\sigma(\mathbf{w}^\top \mathbf{x}_n))$, where $\mathbf{w}^\star = (0.5, -0.6)$
- Classes are slightly overlapping to reflect realistic uncertainty.



Model Specification

- Prior over weights:

$$\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \alpha^{-1}\mathbf{I})$$

- Likelihood:

$$p(\mathcal{D} \mid \mathbf{w}) = \prod_{n=1}^{N} \sigma(\mathbf{w}^\top \mathbf{x}_n)^{y_n} (1 - \sigma(\mathbf{w}^\top \mathbf{x}_n))^{1-y_n}$$

- Posterior:

$$p(\mathbf{w} \mid \mathcal{D}) \propto p(\mathbf{w}) p(\mathcal{D} \mid \mathbf{w}) \quad \text{(approximated)}$$

Laplace Approximation: Idea

- Is it reasonable to approximate the posterior with a Gaussian?
- Yes:
  - It is an incremental improvement over MAP
  - Many ML methods rely on a single configuration; Laplace is a natural extension
  - When the posterior is dominated by a single mode
- No:
  - When the posterior is multimodal or highly skewed
  - Unfortunately, this is often the case in practice

$$p(\boldsymbol{\theta} \mid \mathcal{D}) \approx \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\mu}, \boldsymbol{\Sigma}), \quad \text{where:}$$

$\boldsymbol{\mu} =$ MAP estimate (mode of posterior), often denoted as $\hat{\boldsymbol{\theta}}$

$\boldsymbol{\Sigma} =$ inverse Hessian of the log posterior at the mode, denoted as $\mathbf{H}^{-1}$

Laplace Approximation: How it Works

$$p(\boldsymbol{\theta} \mid \mathcal{D}) \approx \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\mu} = \hat{\boldsymbol{\theta}}, \boldsymbol{\Sigma} = \mathbf{H}^{-1})$$

- Why setting $\boldsymbol{\mu}$ to $\hat{\boldsymbol{\theta}}$, i.e., the MAP estimate?
- $p(\boldsymbol{\theta} \mid \mathcal{D})$ is proportional to the product of the prior and likelihood: Search for the point that maximizes $L(\boldsymbol{\theta})p(\boldsymbol{\theta}) \Rightarrow$ MAP estimate

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta} \mid \mathcal{D}) = \arg\max_{\boldsymbol{\theta}} \left(\log p(\boldsymbol{\theta}) + \sum_{i=1}^{N} \log p(y_i \mid \mathbf{x}_i, \boldsymbol{\theta}) \right)$$

- It is a simple optimization problem:
  - Use gradient-based methods (e.g., Newton-Raphson, L-BFGS)
  - Can easily solve high-dimensional problems, if autograd is available

Laplace Approximation: How it Works

$$p(\boldsymbol{\theta} \mid \mathcal{D}) \approx \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\mu} = \hat{\boldsymbol{\theta}}, \boldsymbol{\Sigma} = \mathbf{H}^{-1})$$

- Why setting $\boldsymbol{\Sigma}$ to $\mathbf{H}^{-1}$, i.e., the inverse Hessian?
- The MAP is a turning point. What happens there?
  - Imagine it as a mountain peak in the log-posterior landscape
  - The gradient vanishes: $\nabla \log p(\boldsymbol{\theta} \mid \mathcal{D})\big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}} = 0$
  - The Hessian captures local curvature: $\mathbf{H} = -\nabla^2 \log p(\boldsymbol{\theta} \mid \mathcal{D})\big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}$
- $\mathbf{H} \in \mathbb{R}^{d \times d}$ (where d is the number of parameters), where:

$$\mathbf{H}_{ij} = -\frac{\partial^2 \log p(\boldsymbol{\theta} \mid \mathcal{D})}{\partial \theta_i \partial \theta_j}\bigg|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}$$

- $\boldsymbol{\Sigma} = \mathbf{H}^{-1}$:
  - High curvature (large values in $\mathbf{H}$) means steep slope, low uncertainty
  - Low curvature (small values in $\mathbf{H}$) means flat slope, high uncertainty

Taylor Expansion and Laplace Approximation

- Taylor expansion approximates a function $f(\boldsymbol{\theta})$ around a point $\hat{\boldsymbol{\theta}}$ with a polynomial, i.e., a smooth curve:

$$f(\boldsymbol{\theta}) \approx f(\hat{\boldsymbol{\theta}}) + \nabla f(\hat{\boldsymbol{\theta}})^\top (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}) + \frac{1}{2}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^\top \nabla^2 f(\hat{\boldsymbol{\theta}})(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})$$

- a second-order Taylor expansion around the MAP estimate is:

$$\log p(\boldsymbol{\theta} \mid \mathcal{D}) \approx \log p(\hat{\boldsymbol{\theta}} \mid \mathcal{D}) - \frac{1}{2}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^\top \mathbf{H} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})$$

  where $\mathbf{H} = -\nabla^2 \log p(\boldsymbol{\theta} \mid \mathcal{D})\big|_{\hat{\boldsymbol{\theta}}}$ is the (negative) Hessian at the mode.
- Exponentiating both sides gives the Laplace approximation of the posterior:

$$p(\boldsymbol{\theta} \mid \mathcal{D}) \approx \mathcal{N}(\boldsymbol{\theta}; \hat{\boldsymbol{\theta}}, \mathbf{H}^{-1})$$

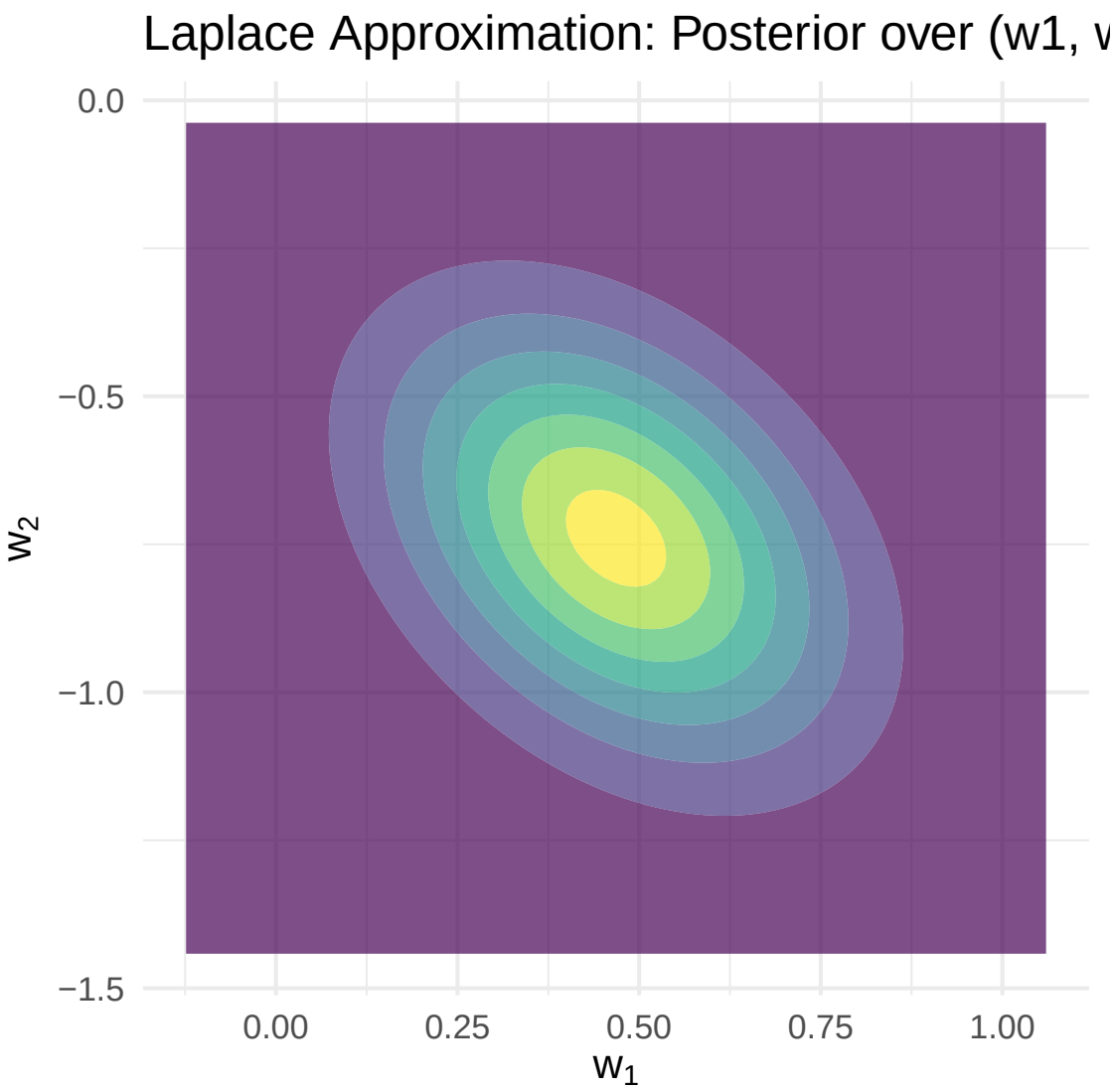
Laplace Approximation: Summary

So the Laplace Approximation gives:

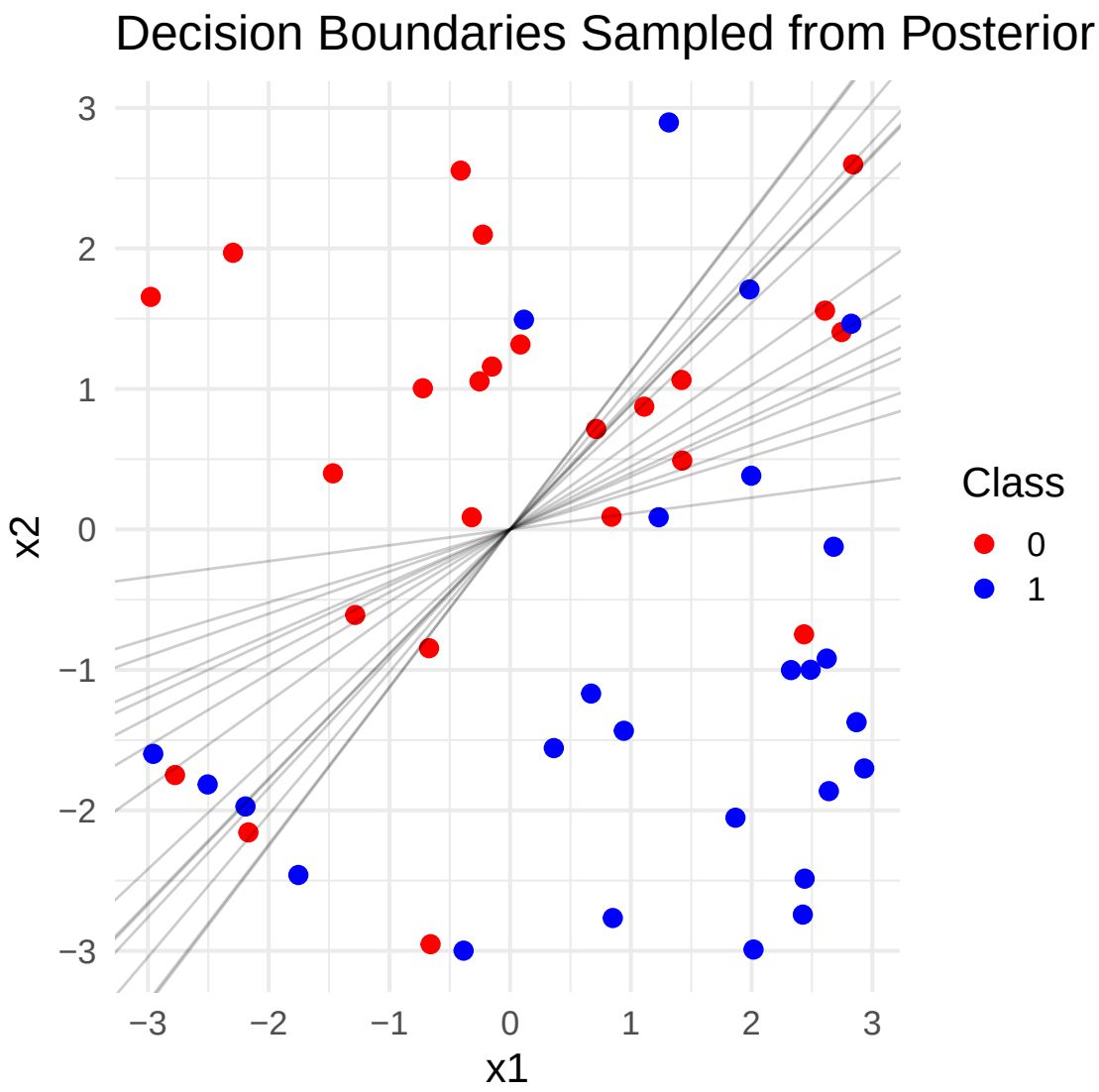
$$p(\boldsymbol{\theta} \mid \mathcal{D}) \approx \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\mu} = \hat{\boldsymbol{\theta}}, \boldsymbol{\Sigma} = \mathbf{H}^{-1})$$

- Computational considerations:
  - MAP estimate $\hat{\boldsymbol{\theta}}$ is a point estimate, not a distribution
  - If gradients are available, it is efficient to compute
  - Hessian $\mathbf{H}$ is computed at the MAP estimate
  - Up to $O(d^3)$ for inversion, where d is the number of parameters
  - Cannot work for very high-dimensional problems (e.g., $d > 1000$)
- Final conclusion:
  Laplace Approximation is easy to implement and compute but may be inaccurate if the posterior is multimodal or skewed.

Laplace Approximation in Logistic Regression Example



Laplace Approximation of Posterior



Importance Sampling: Motivation

- Goal: Compute expectations under a difficult distribution $p(\theta)$ (e.g., posterior).
- Direct sampling from $p(\theta)$ is hard or impossible.
- Instead, sample from a simpler proposal distribution $q(\theta)$.

$$\mathbb{E}_p[f(\theta)] = \int f(\theta)p(\theta)d\theta \quad \text{but} \quad p(\theta) \text{ is hard to sample from.}$$

Importance Sampling Estimator

$$\mathbb{E}_p[f(\theta)] = \int f(\theta)\frac{p(\theta)}{q(\theta)}q(\theta)d\theta = \mathbb{E}_q\left[f(\theta)w(\theta)\right]$$

where the importance weights are

$$w(\theta) = \frac{p(\theta)}{q(\theta)}.$$

Practical Importance Sampling

Given samples $\{\theta_i\}_{i=1}^N \sim q(\theta)$:

$$\hat{\mu} = \frac{\sum_{i=1}^N w_i f(\theta_i)}{\sum_{i=1}^N w_i}, \quad \text{where} \quad w_i = \frac{p(\theta_i)}{q(\theta_i)}.$$

- Weights are normalized to sum to 1.
- Effective when $q(\theta)$ covers $p(\theta)$ well.

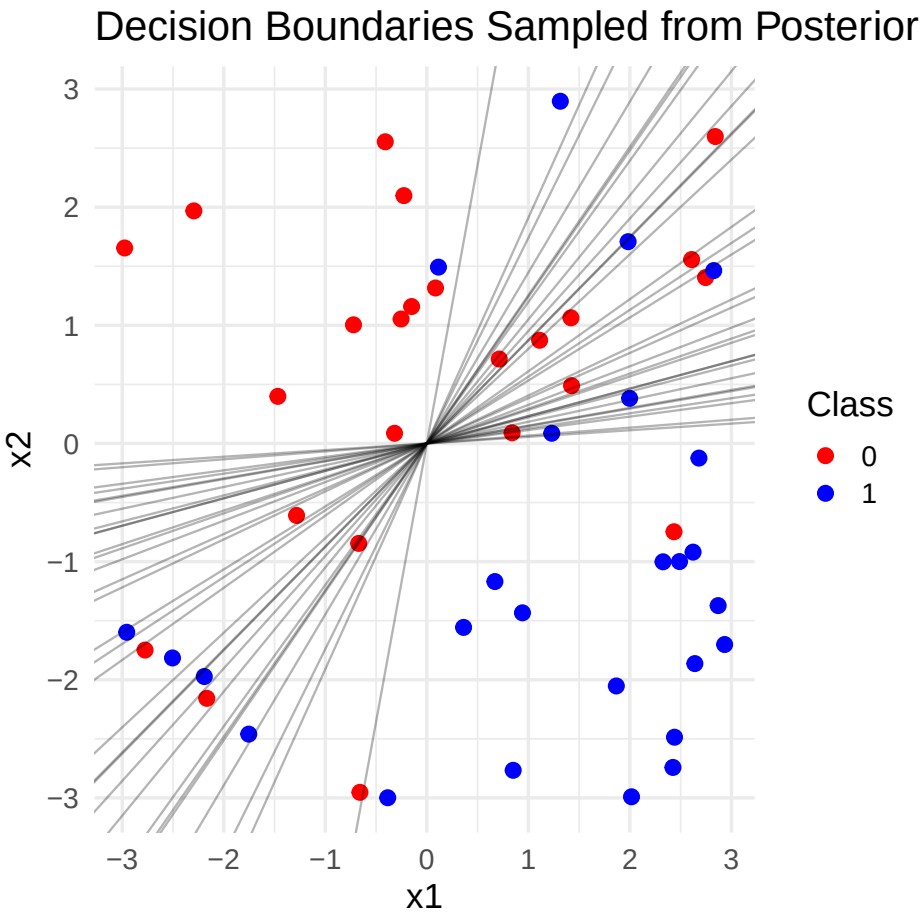
Importance Sampling in Bayesian Inference

- Target posterior: $p(\theta|X, y) \propto p(y|X, \theta)p(\theta)$.
- Proposal $q(\theta)$ can be prior or Laplace approximation.
- Importance weights:

$$w_i = \frac{p(y|X, \theta_i)p(\theta_i)}{q(\theta_i)}.$$

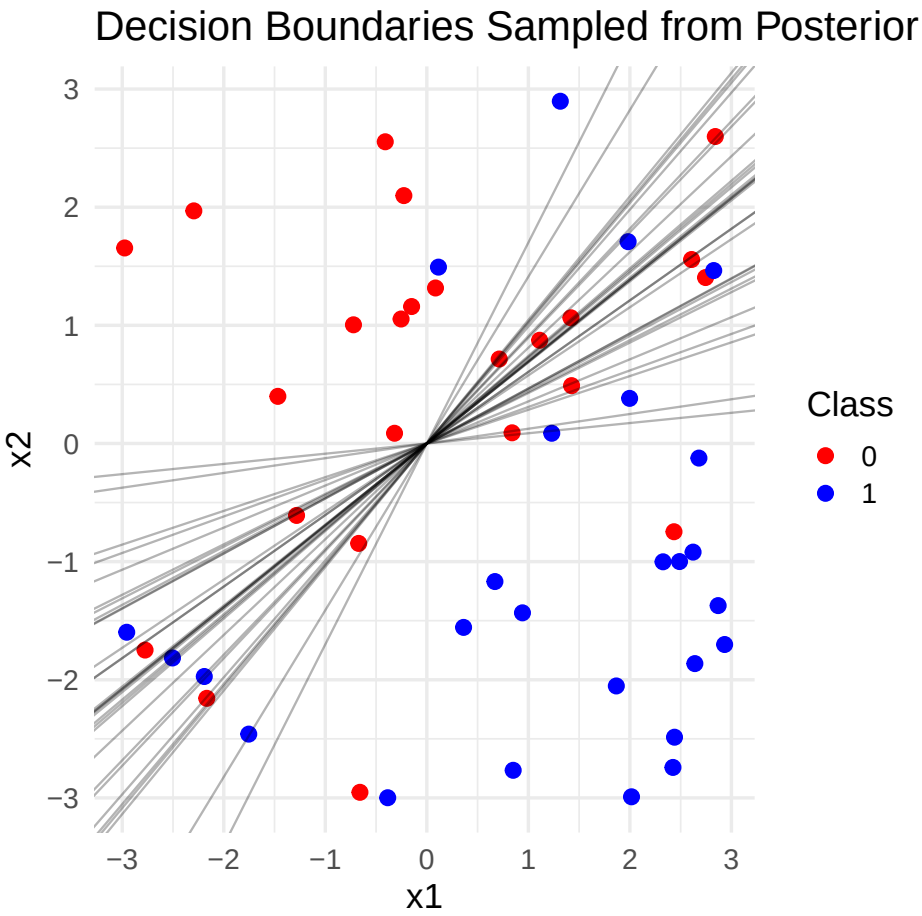
- Samples $\theta_i \sim q(\theta)$ weighted to approximate the posterior.

Importance Sampling with Prior Proposal



- Proposal distribution $q(\theta) = p(\theta)$ (prior).
- Samples drawn directly from prior.
- Importance weights correct for data likelihood.
- May have high variance if prior poorly matches posterior.

Importance Sampling with Laplace Approximation Proposal



- Proposal distribution $q(\theta) \approx \mathcal{N}(\hat{\theta}, H^{-1})$ (Laplace approx).
- Samples concentrated near MAP estimate.
- Importance weights reweight samples to correct approximation.
- Typically lower variance than prior proposal.

Markov Chain Monte Carlo (MCMC) for Bayesian Inference

- Goal: Sample from posterior distribution $p(\theta \mid X, y)$ when direct sampling is difficult.
- Construct a Markov chain whose stationary distribution is the posterior.
- Generates dependent samples that approximate the posterior as the chain runs.
- Widely applicable to complex models where exact inference is intractable.

Metropolis-Hastings Algorithm

- Start from an initial parameter $\theta^{(0)}$.
- At step t, propose θ^* from proposal distribution $q(\theta^* \mid \theta^{(t-1)})$.
- Calculate acceptance probability:

$$\alpha = \min\left(1, \frac{p(\theta^* \mid X, y)q(\theta^{(t-1)} \mid \theta^*)}{p(\theta^{(t-1)} \mid X, y)q(\theta^* \mid \theta^{(t-1)})}\right)$$

- Accept θ^* with probability α, else keep $\theta^{(t-1)}$.
- Ensures the chain converges to posterior distribution.

MCMC for Bayesian Logistic Regression

- Posterior $p(\theta \mid X, y) \propto p(y \mid X, \theta)p(\theta)$ is non-conjugate.
- MCMC provides a way to approximate the posterior without analytic form.
- Samples $\{\theta^{(t)}\}_{t=1}^{T}$ can be used for:
  - Estimating expectations (posterior means, variances).
  - Predictive distributions.
  - Visualizing uncertainty, e.g. decision boundary variation.

Practical Considerations

- **Burn-in**: Discard initial samples until chain stabilizes.
- **Thinning**: Keep every k-th sample to reduce autocorrelation.
- **Tuning**: Proposal distribution parameters (e.g., step size) affect acceptance rate and mixing.
- Diagnostics needed to check convergence (trace plots, effective sample size).

MCMC Samples: Decision Boundaries from Posterior