Bayesian Logistic Regression Problems

ELG 5218 - Uncertainty Evaluation in Engineering Measurements and Machine Learning

Instructor: Miodrag Bolic, University of Ottawa

Date: January 28, 2026

PART A: CONCEPTUAL QUESTIONS (Simple to Intermediate)

A1. What is the fundamental difference between classical logistic regression and Bayesian logistic regression?

Answer:

Classical logistic regression optimizes a point estimate $\hat{\mathbf{w}}$ by maximizing the likelihood via cross-entropy loss. This gives a single "best fit" weight vector without uncertainty quantification.

Bayesian logistic regression introduces a prior $p(\mathbf{w})$ on weights and computes the full posterior distribution $p(\mathbf{w} \mid \mathcal{D})$. This provides:

- A distribution over plausible weight vectors, not just one estimate.
- Credible intervals for uncertainty in parameters.
- Predictive distributions with confidence bands.
- Automatic regularization through the prior.

The posterior captures both aleatoric uncertainty (inherent randomness in class labels) and epistemic uncertainty (uncertainty due to limited data).

A2. Why is the logistic regression posterior non-conjugate?

Answer:

Conjugate pairs exist when a prior and likelihood produce a posterior of the same family (e.g., Beta prior + Binomial likelihood $\rightarrow$ Beta posterior).

For Bayesian logistic regression:

- Likelihood: product of sigmoids,
$$p(\mathbf{y} \mid X, \mathbf{w}) = \prod_{n=1}^{N} \sigma(\mathbf{w}^\top \mathbf{x}_n)^{y_n} \big(1 - \sigma(\mathbf{w}^\top \mathbf{x}_n)\big)^{1-y_n}.$$
- Prior: Gaussian $p(\mathbf{w}) = \mathcal{N}(\mathbf{w} \mid 0, \alpha^{-1} I)$.

The sigmoid is not conjugate to the Gaussian. Multiplying a product of sigmoids by a Gaussian does not yield a Gaussian, so the product is not a standard distribution, and the normalization constant
$$Z = \int p(\mathbf{y} \mid X, \mathbf{w})\, p(\mathbf{w})\, d\mathbf{w}$$
is analytically intractable.

A3. What are the three main steps of the Laplace Approximation?

Answer:

Laplace approximation approximates the intractable posterior $p(\mathbf{w} \mid \mathcal{D})$ with a Gaussian around a single mode.

1. **Find the MAP (Maximum A Posteriori):**

$$\mathbf{w}_{\text{MAP}} = \arg\max_{\mathbf{w}} \log p(\mathbf{w} \mid \mathcal{D}),$$

which is equivalent to minimizing the negative log-posterior. For logistic regression, this objective is convex, so we can use gradient descent or Newton–Raphson.

2. **Compute the Hessian at the MAP:**
The Hessian encodes the curvature (second-order information) of the log-posterior. For logistic regression,

$$H = \nabla^2_{\mathbf{w}} \log p(\mathbf{w} \mid \mathcal{D})\big|_{\mathbf{w}=\mathbf{w}_{\text{MAP}}} = -\alpha I - X^\top S X,$$

where S is a diagonal matrix with $S_{nn} = \mu_n(1-\mu_n)$ and $\mu_n = \sigma(\mathbf{w}_{\text{MAP}}^\top \mathbf{x}_n)$.

3. **Construct the Gaussian approximation:**

$$p(\mathbf{w} \mid \mathcal{D}) \approx q(\mathbf{w}) = \mathcal{N}\big(\mathbf{w} \mid \mathbf{w}_{\text{MAP}},\, H^{-1}\big).$$

The mode becomes the mean, and the negative Hessian inverse becomes the covariance.

A4. Explain the physical interpretation of the Hessian in one dimension.

Answer:

For a 1D log-density $\ell(w) = \log p(w \mid \mathcal{D})$:

- $\ell'(w)$ is the slope (gradient).
- $\ell''(w)$ is the curvature (how rapidly the slope changes).

At a peak (where $\ell'(w^*) = 0$):

- Strong negative curvature (large $-\ell''(w^*)$) $\Rightarrow$ narrow, sharp peak $\Rightarrow$ low uncertainty.
- Weak negative curvature (small $-\ell''(w^*)$) $\Rightarrow$ broad, flat peak $\Rightarrow$ high uncertainty.

Quantitatively, near the mode we approximate

$$\ell(w) \approx \ell(w^*) - \frac{1}{2} H (w - w^*)^2,$$

with $H = -\ell''(w^*) > 0$. Exponentiating yields a Gaussian with variance $\sigma^2 = 1/H$. Larger H means smaller variance, reflecting a sharp peak.

A5. What happens to the Laplace approximation when the posterior is multimodal?

Answer:

Laplace approximation is local around a single mode. If the posterior has multiple modes:

- The approximation captures only the mode where $\mathbf{w}_{\text{MAP}}$ was found.
- Other modes are completely ignored.
- The approximate posterior $q(\mathbf{w})$ drastically underestimates total uncertainty.
- Credible intervals and predictive distributions become overconfident.

Example: For

$$p(w) = 0.5\,\mathcal{N}(w \mid -3, 1) + 0.5\,\mathcal{N}(w \mid 3, 1),$$

Laplace around $w = 3$ yields $q(w) \approx \mathcal{N}(w \mid 3, 1)$, completely missing the mode at -3 and the 50% probability mass there.

Solution: Use methods likemVariational Inference (VI), or MCMC (e.g., HMC/NUTS) for multimodal posteriors.

PART B: MATHEMATICAL DERIVATIONS (Intermediate to Advanced)

B1. Derive the gradient of the log-posterior for Bayesian logistic regression.

Problem. Given

$$p(\mathbf{w} \mid X, \mathbf{y}) \propto p(\mathbf{y} \mid X, \mathbf{w})\, p(\mathbf{w}),$$

where

$$p(\mathbf{y} \mid X, \mathbf{w}) = \prod_{n=1}^{N} \sigma(\mathbf{w}^\top \mathbf{x}_n)^{y_n} \big(1 - \sigma(\mathbf{w}^\top \mathbf{x}_n)\big)^{1-y_n}, \quad p(\mathbf{w}) = \mathcal{N}(\mathbf{w} \mid 0, \alpha^{-1} I),$$

derive $\nabla_{\mathbf{w}} \log p(\mathbf{w} \mid X, \mathbf{y})$.

Answer:

The log-posterior is, up to a constant,

$$\log p(\mathbf{w} \mid X, \mathbf{y}) = \log p(\mathbf{y} \mid X, \mathbf{w}) + \log p(\mathbf{w}).$$

Prior term:

$$\log p(\mathbf{w}) = -\frac{\alpha}{2} \mathbf{w}^\top \mathbf{w} + \text{const} \quad \Rightarrow \quad \nabla_{\mathbf{w}} \log p(\mathbf{w}) = -\alpha \mathbf{w}.$$

Likelihood term:

$$\log p(\mathbf{y} \mid X, \mathbf{w}) = \sum_{n=1}^{N} \left[y_n \log \mu_n + (1 - y_n) \log(1 - \mu_n) \right], \quad \mu_n = \sigma(\mathbf{w}^\top \mathbf{x}_n).$$

We know

$$\frac{\partial \mu_n}{\partial \mathbf{w}} = \mu_n (1 - \mu_n) \mathbf{x}_n.$$

Then

$$\nabla_{\mathbf{w}} \log p(\mathbf{y} \mid X, \mathbf{w}) = \sum_{n=1}^{N} (y_n - \mu_n)\mathbf{x}_n.$$

Combined gradient:

$$\nabla_{\mathbf{w}} \log p(\mathbf{w} \mid X, \mathbf{y}) = -\alpha\mathbf{w} + \sum_{n=1}^{N} (y_n - \mu_n)\mathbf{x}_n = -\alpha\mathbf{w} + X^\top(\mathbf{y} - \boldsymbol{\mu}),$$

where $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_N)^\top$.

This is used in gradient descent:

$$\mathbf{w}_{\text{new}} = \mathbf{w}_{\text{old}} + \eta \, \nabla_{\mathbf{w}} \log p(\mathbf{w}_{\text{old}} \mid X, \mathbf{y}),$$

where η is the learning rate.

B2. Derive the Hessian matrix for Bayesian logistic regression.

Problem. Compute $\nabla^2_{\mathbf{w}} \log p(\mathbf{w} \mid X, \mathbf{y})$.

Answer:

From B1:

$$\nabla_{\mathbf{w}} \log p(\mathbf{w} \mid X, \mathbf{y}) = -\alpha\mathbf{w} + \sum_{n=1}^{N} (y_n - \mu_n)\mathbf{x}_n.$$

The prior contributes

$$\nabla^2_{\mathbf{w}} \log p(\mathbf{w}) = -\alpha I.$$

For the likelihood term, recall

$$\mu_n = \sigma(\mathbf{w}^\top \mathbf{x}_n), \quad \frac{\partial \mu_n}{\partial \mathbf{w}} = \mu_n(1 - \mu_n)\mathbf{x}_n.$$

Differentiating again:

$$\nabla^2_{\mathbf{w}} \log p(\mathbf{y} \mid X, \mathbf{w}) = -\sum_{n=1}^{N} \mu_n(1 - \mu_n)\mathbf{x}_n\mathbf{x}_n^\top.$$

Stacking into matrix form, define S diagonal with $S_{nn} = \mu_n(1 - \mu_n)$. Then

$$\sum_{n=1}^{N} \mu_n(1 - \mu_n)\mathbf{x}_n\mathbf{x}_n^\top = X^\top S X.$$

Therefore,

$$H = \nabla^2_{\mathbf{w}} \log p(\mathbf{w} \mid X, \mathbf{y}) = -\alpha I - X^\top S X.$$

Properties:

- S_{nn} is largest when $\mu_n \approx 0.5$ (most informative) and smallest near 0 or 1 (least informative).
- Laplace posterior covariance:

$$\Sigma_N = (-H)^{-1} = (\alpha I + X^\top S X)^{-1}.$$

PART C: PARAMETRIC ANALYSIS (What if we change parameters?)

C1. What happens to the posterior covariance if we increase the regularization parameter α?

Answer:
Recall

$$\Sigma_N = (\alpha I + X^\top S X)^{-1}.$$

As α increases:

- The diagonal term αI grows, so $\alpha I + X^\top S X$ becomes larger.
- Its inverse Σ_N becomes smaller (posterior becomes more concentrated).
- The MAP estimate $\mathbf{w}_{\text{MAP}}$ is pulled closer to zero.

Effects on predictions:

- Credible intervals narrow.
- Predictive variance $v = \mathbf{x}_*^\top \Sigma_N \mathbf{x}_*$ decreases.
- Predictions become more confident (closer to 0 or 1).
- The model is more regularized: better generalization but potential underfitting.

Extreme cases:

- $\alpha \to 0$ (weak prior): Posterior driven by data; high epistemic uncertainty if data are scarce.
- $\alpha \to \infty$ (strong prior): $\mathbf{w}_{\text{MAP}} \to 0$; posterior very tight around zero.

PART D: IMPLEMENTATION AND PRACTICAL QUESTIONS

D1. In the NumPyro notebook, what is the role of `numpyro.plate`?

Answer:
`numpyro.plate("data", N)` declares an i.i.d. (independent and identically distributed) plate of size N, vectorizing likelihood computation.

Without plate:

```
for n in range(N):
    numpyro.sample(f"obs_{n}", dist.Bernoulli(logits=logits[n]), obs=y[n])
```

This loops through data one-by-one (inefficient).

With plate:

```
with numpyro.plate("data", len(X)):
    numpyro.sample("obs", dist.Bernoulli(logits=logits), obs=y)
```

This efficiently computes

$$\log p(\mathbf{y} \mid X, \theta) = \sum_{n=1}^{N} \log \text{Bernoulli}\big(y_n; \sigma(\mathbf{w}^\top \mathbf{x}_n + b)\big)$$

in a vectorized way. The plate tells NumPyro that the likelihood factors over independent samples, enabling efficient broadcasting and MCMC sampling.

D2. How would you modify the component failure prediction model if you had an imbalanced dataset (e.g., 95% functional, 5% failed)?

Answer:

Possible strategies:

Option 1: Weight the observations.

```
def logistic_model_weighted(X, y, weights=None):
    w = numpyro.sample("w", dist.Normal(0, 1).expand([X.shape[1]]))
    b = numpyro.sample("b", dist.Normal(0, 1))
    logits = jnp.dot(X, w) + b
    with numpyro.plate("data", len(X)):
        if weights is not None:
            obs_dist = dist.Bernoulli(logits=logits)
            numpyro.factor("obs", weights * obs_dist.log_prob(y))
        else:
            numpyro.sample("obs", dist.Bernoulli(logits=logits), obs=y)
```

This scales log-probabilities to upweight minority (failed) cases.

Option 2: Class-balanced prior. Use a stronger prior on $\mathbf{w}$ to prevent extreme decision boundaries. For example use $\mathbf{w} \sim \mathcal{N}(0, \alpha^{-1} I)$ with larger α.

Option 3: Resampling. Oversample the minority class or undersample the majority class; use stratified cross-validation.

Option 4: Threshold adjustment. Predict failure if $p(y = 1 \mid \mathbf{x}, \mathcal{D}) > \tau$ for some $\tau < 0.5$, adjusted based on the false positive/false negative trade-off.

D3. What differences would you expect between Laplace Approximation and MCMC (HMC) sampling for this problem?

Answer:

Aspect	Laplace Approximation	HMC Sampling
Speed	Fast (one optimization + Hessian)	Slow (many iterations)
Accuracy	Good for unimodal, near-Gaussian	Excellent, samples from true posterior
Multimodality	Misses other modes	Captures all modes (given mixing)
Non-Gaussian	Underestimates tails	Captures skew/heavy tails
Scalability	Needs Hessian inversion ($O(D^3)$)	Per-iter $O(D)$, but many iters
Interpretability	Single Gaussian	Posterior samples (intuitive)
Credible intervals	From covariance ellipsoid	Empirical quantiles

PART E: OTHER PROBLEMS

E1: Posterior Mode for Logistic Regression

Problem. Consider the logistic regression model with one data point $x = 2$, $y = 1$, and prior $w \sim \mathcal{N}(0, 1)$. Find the posterior mode of w analytically and show it as a formula that can can be solved then numerically.

Solution. Posterior log-density:

$$\log p(w|x, y) \propto yxw - \log(1 + e^{xw}) - \frac{w^2}{2}.$$

Plug in $x = 2$, $y = 1$:

$$\log p(w) \propto 2w - \log(1 + e^{2w}) - \frac{w^2}{2}.$$

Take derivative and set to zero for MAP:

$$2 - \frac{2e^{2w}}{1 + e^{2w}} - w = 0.$$

This can be solved numerically to get the posterior mode.