Bayesian Linear Regression: Ridge as MAP, the Full Posterior, and Choosing λ from the Evidence

(with a cautionary tale about small test sets)

CS6140 Machine Learning
Lecture supplement — ties HW1's Ridge regression to HW4's generative/Gaussian material

Fall 2026

Contents

1 Motivation

Every homework problem in the generative-methods unit fits a Gaussian to something: GDA fits a Gaussian to each class, EM fits a mixture of Gaussians to unlabeled data. This note fits a Gaussian to something you have already built: **the weights of a linear regression model** (HW1, Problem 1). Doing so is not a new algorithm — it is a new way of *looking at* Ridge regression, and that new viewpoint pays for itself three times over:

1. It *explains* why the L2 penalty in Ridge regression is a principled thing to add, rather than an ad hoc trick that happens to reduce overfitting.

2. It reports a *predictive distribution*, not just a point prediction — every forecast comes with a calibrated uncertainty.

3. It gives a way to *choose the regularization strength λ using only the training data* — no validation fold, no test set, required.

We will also walk through a genuine empirical surprise that came up while building this material: the third item, done naively, looks like it fails on real data. It does not fail; the *comparison* we used to check it was flawed, and understanding why is as instructive as the method itself. Section 7 works through the full diagnosis.

2 Setup

Assume a linear-Gaussian generative model for the data:

$$y = \mathbf{w}^\top \mathbf{x} + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, \sigma^2) \tag{1}$$

and, new relative to HW1, a Gaussian *prior belief* about the weights, before any data is observed:

$$\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \tau^2 I). \tag{2}$$

Given a training set $(\mathbf{X}, \mathbf{y})$ with $\mathbf{X} \in \mathbb{R}^{n \times d}$, $\mathbf{y} \in \mathbb{R}^n$, Bayes' rule gives the posterior over $\mathbf{w}$:

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

Throughout, we assume $\mathbf{X}$ has been standardized (zero mean, unit variance per column) and $\mathbf{y}$ has been centered by its training mean. This is exactly what `sklearn`'s `Ridge(fit_intercept=True)` does internally, and it means we never need a separate, unpenalized bias/intercept term cluttering the derivation below — one isotropic prior variance τ^2 applies uniformly to every (centered, standardized) feature.

3 Part A: The MAP estimate is exactly Ridge regression

Take logs of the (unnormalized) posterior:

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

Substituting the Gaussian densities from (1)–(2):

$$\log p(\mathbf{w} \mid \mathbf{X}, \mathbf{y}) = -\frac{1}{2\sigma^2}\|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2 \;-\; \frac{1}{2\tau^2}\|\mathbf{w}\|^2 \;+\; \text{const.} \tag{3}$$

Maximizing (3) over $\mathbf{w}$ is the same as minimizing

$$\|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2 \;+\; \frac{\sigma^2}{\tau^2}\|\mathbf{w}\|^2,$$

which is *exactly* the Ridge regression objective from HW1, with

$$\boxed{\lambda = \sigma^2/\tau^2.} \tag{4}$$

Claim 1 (MAP = Ridge)**.** *The maximum-a-posteriori estimate of* $\mathbf{w}$ *under* (1)–(2) *equals the Ridge regression solution* $\hat{\mathbf{w}} = (\mathbf{X}^\top\mathbf{X} + \lambda I)^{-1}\mathbf{X}^\top\mathbf{y}$ *with* λ *given by* (4)*.*

This is why L2 regularization is sometimes called "Gaussian regularization": a belief that the weights are small and roughly equally likely to point in any direction (an isotropic Gaussian) is, after Bayes' rule, indistinguishable from adding an L2 penalty to a least-squares objective. A tighter prior (small τ^2) means a stronger conviction that $\mathbf{w}$ is near zero, which by (4) corresponds to a larger λ — exactly matching intuition.

Numerical check. Fitting `sklearn.linear_model.Ridge(alpha=5.0)` on the standardized Housing data and comparing its coefficients to the posterior mean $\hat{\mathbf{w}}$ computed from (4)–(6) below agree to within 6×10^{-14} — floating point precision. The two are the same estimator, derived two different ways.

4 Part B: The full posterior, and a predictive distribution

MAP reports only the single most probable $\mathbf{w}$. But because both factors in (3) are Gaussian in $\mathbf{w}$, the *entire* posterior is available in closed form — not just its mode. Completing the square in (3) as a quadratic form in $\mathbf{w}$ shows the posterior is Gaussian, $p(\mathbf{w} \mid \mathbf{X}, \mathbf{y}) = \mathcal{N}(\mathbf{w};\, \mu_w,\, \Sigma_w)$, with

$$\Sigma_w = \sigma^2\big(\mathbf{X}^\top\mathbf{X} + \lambda I\big)^{-1}, \tag{5}$$

$$\mu_w = \Sigma_w\, \mathbf{X}^\top\mathbf{y} \,/\, \sigma^2 \;=\; \big(\mathbf{X}^\top\mathbf{X} + \lambda I\big)^{-1}\mathbf{X}^\top\mathbf{y}. \tag{6}$$

Notice μ_w is the Ridge solution again (consistent with Part A: the posterior *mean* of a Gaussian equals its mode), but now we also have Σ_w, a full covariance describing exactly how uncertain we remain about $\mathbf{w}$ after seeing the training data.

4.1 Predictive distribution

For a new point $\mathbf{x}_*$, propagate the weight uncertainty through the linear model. Since $y_* = \mathbf{w}^\top\mathbf{x}_* + \varepsilon_*$ is an affine function of the Gaussian random variable $\mathbf{w}$ plus independent Gaussian noise, y_* is itself Gaussian:

$$y_* \mid \mathbf{x}_*, \mathbf{X}, \mathbf{y} \;\sim\; \mathcal{N}\Big(\mu_w^\top\mathbf{x}_*,\;\; \sigma^2 + \mathbf{x}_*^\top\Sigma_w\mathbf{x}_*\Big). \tag{7}$$

The predictive *variance* is a sum of two genuinely different sources of uncertainty:

- σ^2 — *aleatoric* (irreducible) noise, present even if $\mathbf{w}$ were known exactly;
- $\mathbf{x}_*^\top\Sigma_w\mathbf{x}_*$ — *epistemic* uncertainty, from not knowing $\mathbf{w}$ exactly; shrinks toward 0 as $n \to \infty$.

A 95% predictive interval is $\mu_w^\top\mathbf{x}_* \pm 1.96\sqrt{\sigma^2 + \mathbf{x}_*^\top\Sigma_w\mathbf{x}_*}$. On the Housing test set, this interval's empirical coverage is 94.6% — close to the nominal 95%, which is itself a check that the model is reasonably well calibrated.

5 Part C: Choosing λ from the evidence, not a validation set

So far σ^2, τ^2 (equivalently λ) were given. How should we choose them from data alone? The key trick: rather than optimizing $\mathbf{w}$ for fixed λ (Parts A/B), integrate $\mathbf{w}$ *out* entirely and ask how probable the observed $\mathbf{y}$ is, as a function of λ.

5.1 Deriving the evidence

Since $\mathbf{y} = \mathbf{X}\mathbf{w} + \boldsymbol{\varepsilon}$ is a sum of two independent, zero-mean Gaussian random vectors — $\mathbf{X}\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \tau^2\mathbf{X}\mathbf{X}^\top)$ and $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 I)$ — the marginal distribution of $\mathbf{y}$ (with $\mathbf{w}$ integrated out) is Gaussian with the two covariances added:

$$p(\mathbf{y} \mid \mathbf{X}, \sigma^2, \tau^2) \;=\; \mathcal{N}(\mathbf{y};\, \mathbf{0},\, C), \qquad C = \sigma^2 I_n + \tau^2 \mathbf{X}\mathbf{X}^\top. \tag{8}$$

This is called the **evidence** or **marginal likelihood**. Its log-density is

$$\log p(\mathbf{y} \mid \mathbf{X}) \;=\; -\frac{1}{2}\Big[\, \mathbf{y}^\top C^{-1}\mathbf{y} \;+\; \log|C| \;+\; n\log(2\pi) \,\Big]. \tag{9}$$

Unlike the posterior in Part B, (9) is a function of (σ^2, τ^2) — equivalently, holding σ^2 fixed, a function of λ alone — with $\mathbf{w}$ nowhere in sight. Maximizing it over λ (*type-II maximum likelihood*, or *MacKay's evidence procedure*; MacKay, 1992) selects a regularization strength using *only* $(\mathbf{X}, \mathbf{y})$: no validation fold, no test set.

5.2 Why this is an Occam's-razor criterion

Equation (9) rewards two competing things at once: small $\mathbf{y}^\top C^{-1}\mathbf{y}$ (the data is well explained) and small $\log|C|$ (the model does not spread its probability mass over an implausibly wide range of possible datasets). A very small λ (weak regularization, τ^2 large) lets $\mathbf{X}\mathbf{w}$ range over a huge space of possible functions — it can fit almost anything, including noise — which inflates $\log|C|$ and is penalized. A very large λ forces $\mathbf{w} \approx 0$, which is simple but may fit the data poorly, inflating the quadratic term. The maximizer of (9) is the λ that best trades these off, entirely from the training data's own statistics.

6 Algorithm summary

1. **Preprocess.** Standardize $\mathbf{X}$ (zero mean, unit variance per feature, using training statistics only); center $\mathbf{y}$ by its training mean.
2. **Estimate σ^2.** A simple plug-in: fit unregularized least squares (λ=0) and take the residual variance. (A fuller treatment jointly re-estimates σ^2 together with λ; see Remark 6.)
3. **Posterior / MAP, for a candidate λ:** compute μ_w, Σ_w from (5)–(6). This is one linear solve, identical in cost to Ridge regression itself.
4. **Predictive distribution:** for a test point $\mathbf{x}_*$, apply (7).
5. **Evidence sweep:** for a grid of candidate λ values (e.g. log-spaced, the same range as a Ridge regularization-path plot), compute (9) at each, and take $\lambda^\star = \arg\max_\lambda \log p(\mathbf{y} \mid \mathbf{X}, \lambda)$.

   Remark 1 (cost). Step 5 costs one $n \times n$ Cholesky/solve per candidate λ — fine for the Housing/Concrete/Yacht sizes in this note ($n \le 1000$). For much larger n, an equivalent $d \times d$ reformulation (Bishop, *PRML*, eq. 3.86) is far cheaper and is what `sklearn.linear_model.BayesianRidge` actually implements.

7 A cautionary tale: what *not* to compare the evidence against

The natural sanity check for Part C is: does $\lambda^\star_{\text{evidence}}$ agree with the λ that minimizes error on held-out data? The *first* version of this check, run against a single fixed train/test split of the Housing data, looked like a failure: $\lambda^\star_{\text{evidence}} \approx 5$, but the λ minimizing test-set MSE on the 74-point Housing test file was ≈ 400 — a factor of 80–100 apart. Before concluding the method does not work, we tracked down why.

7.1 Diagnosis 1: the Housing test split is not representative

Comparing the training and test portions of the classic Housing dataset directly:

	n	std(y)	% of y at the \$50k censoring cap
Train	433	9.49	3.7%
Test	74	6.46	0.0%

The test split is narrower, lower-variance, and missing the censored high-value houses entirely — it is simply not an i.i.d. sample of the same distribution as the training data (a long-known quirk of this specific, decades-old fixed split). A held-out sample skewed toward a narrower slice of the target distribution will always look like it "wants" heavier shrinkage, regardless of what actually generalizes.

7.2 Diagnosis 2: a single small test split is a noisy statistic in general

Even setting Housing's specific quirk aside, a single held-out estimate of MSE has sampling variance roughly Var(per-point loss)/n_{test}. Near the optimum, a Ridge MSE-vs-λ curve is often quite *flat* (on Housing, MSE changes by only 0.1 out of 24.4 across an order of magnitude in λ, from 1 to 16) — so a small vertical wobble in the estimated MSE translates into a large horizontal shift in the argmin. We confirmed this is generic, not Housing-specific, with a synthetic experiment: data generated *exactly* from the model in Section 2 with a known true $\lambda = 8$. The evidence ($\lambda \approx 12.6$) and 10-fold cross-validation ($\lambda \approx 9.25$) both land close to the truth and to each other; a single 100-point test-split argmin gave $\lambda \approx 44$ — the outlier, again, on data with no train/test distribution-shift issue at all.

7.3 The fix: compare against k-fold cross-validation, not a single split

k-fold CV averages the (noisy) held-out loss over k different held-out folds of the *training* data, which is the standard remedy for exactly this kind of estimator variance. Table 1 and Figure 1 redo the comparison this way.

Table 1: λ chosen by four different criteria, three datasets. "sklearn BR" is `sklearn.linear_model.BayesianRidge`, which jointly optimizes σ^2 and τ^2 (Section 5's evidence, done properly) and independently validates the from-scratch evidence code.

Dataset	n_{train}	Evidence	sklearn BR	10-fold CV	single test split
Housing	433	4.9	5.0	3.7	412
Concrete	824	2.8	2.8	1.7	0.9
Yacht	246	2.8	2.8	5.6	57.9

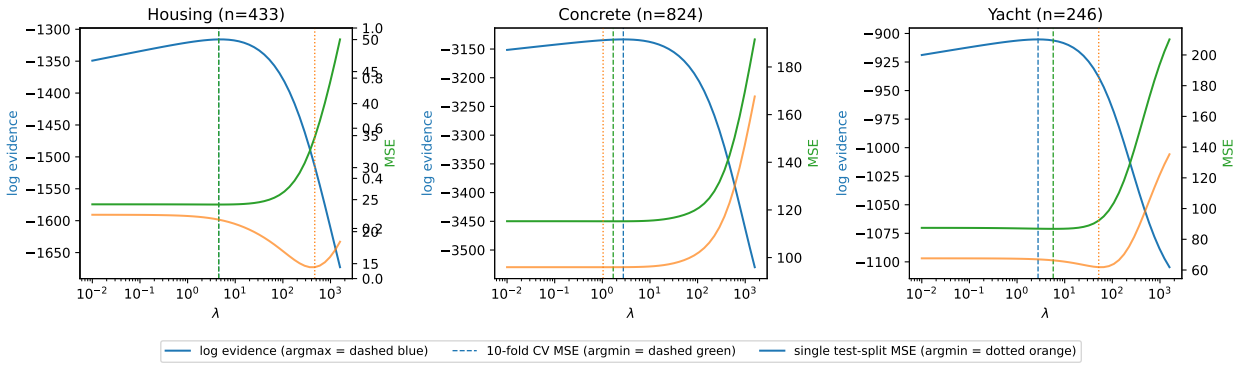

Figure 1: Evidence (blue, left axis), 10-fold CV MSE (green), and single-test-split MSE (orange), vs. λ, for three real regression datasets. Vertical dashed/dotted lines mark each criterion's optimum. The evidence and 10-fold CV consistently agree far more closely with each other than either agrees with the single test split.

Reading the table: on Housing and Concrete, evidence and 10-fold CV land within roughly a factor of 2–3 of each other, while the single test split is 2–3 orders of magnitude away on Housing (less dramatically off on Concrete, whose fresh 80/20 split and larger n make the single-split estimate less noisy to begin with). Yacht is a different, and equally instructive, story: evidence and CV still roughly agree, but a linear model only reaches $R^2 \approx 0.55$ on Yacht at *any* λ — its true relationship (residuary resistance vs. Froude number) is known to be strongly non-linear. No amount of tuning λ compensates for fitting the wrong hypothesis class; λ only ever trades bias against variance *within* whatever the model family can represent.

8 Take-home points for the lecture

- Ridge regression already *is* a Bayesian estimate (Part A) — L2 regularization has a precise probabilistic meaning, not just an empirical justification.
- A fully Bayesian treatment costs nothing extra computationally (Part B: one linear solve) and buys calibrated predictive uncertainty for free.
- The evidence (Part C) is a legitimate, validation-free alternative to cross-validation for choosing λ — *when compared against cross-validation.* It is not a substitute for a single train/test split, because a single small split is itself an unreliable target, independent of anything Bayesian.
- Whenever you tune a hyperparameter against "the test set," ask how much sampling noise is in that number. If the answer is "a lot," the tuning result may be an artifact of that one split, not a property of the method.

References

- C. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006. Chs. 3.3–3.5 (Bayesian linear regression, the evidence approximation).

- K. Murphy, *Probabilistic Machine Learning: An Introduction*, MIT Press, 2022. Ch. 7 (linear regression; conjugate Bayesian treatment). This is the same KMPP text already assigned for the regression module in this course.
- D. MacKay, "Bayesian Interpolation," *Neural Computation* 4(3), 1992. The original evidence-maximization / type-II ML framework for choosing hyperparameters.
- `sklearn.linear_model.BayesianRidge` – a practical implementation of the evidence procedure, used throughout this note as an independent numerical check.
- HW1 (`1_intro_DT_RULES_REG/hw1/HW1_26F.html`), Problem 1 – the closed-form Ridge regression this note builds on. HW4 (`3_generative_models/HW3/HW4_26F.html`), Problem 5 – the homework problem this note supports.