

HW2 Review & Exam 1 Problem Discussion

Ridge, bias-variance, kernels, implicit regularization — then practice
across Lectures 01–06

Outline

1. HW2 Review: ridge regression, bias-variance, kernel validity, implicit regularization (Lectures 03–04)
2. Exam 1 Problem Discussion: three problems sampling Lectures 01–06

four original practice problems in HW2's spirit
(HW2 not yet released) — plus three exam-
style problems, not a full review of every lecture

Coming up: HW2 (Lectures 03–04) was due Monday. No new deliverable today — Fall Recess (10/9) and Exam 1 (10/16) are next.

1. HW2 Review

Ridge, bias-variance, kernels, double descent

Problem 1: ridge regression, by hand

Data (bias folded in): $\mathbf{x}_1 = (1, 1)$, $\mathbf{x}_2 = (2, 1)$,
 $\mathbf{x}_3 = (3, 1)$; $\mathbf{y} = (2, 3, 5)$. Ridge penalty $\lambda = 0.5$.

1. Compute $X^T X$ and $X^T \mathbf{y}$.
2. Solve $\mathbf{w}_{\text{ridge}}^* = (X^T X + n\lambda I)^{-1} X^T \mathbf{y}$ by hand.
3. Which direction did λ push the coefficients, and why?

Step 1: form $X^\top X$ and $X^\top \mathbf{y}$

$$X = \begin{pmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} 2 \\ 3 \\ 5 \end{pmatrix}$$

$$X^\top X = \begin{pmatrix} 14 & 6 \\ 6 & 3 \end{pmatrix}, \quad X^\top \mathbf{y} = \begin{pmatrix} 23 \\ 10 \end{pmatrix}$$

e.g. $(X^\top X)_{11} = 1^2 + 2^2 + 3^2 = 14$; $(X^\top \mathbf{y})_1 = 1(2) + 2(3) + 3(5) = 23$.

Step 2: add the ridge penalty, invert

$$n = 3, \lambda = 0.5 \Rightarrow n\lambda I = 1.5I.$$

$$X^T X + n\lambda I = \begin{pmatrix} 15.5 & 6 \\ 6 & 4.5 \end{pmatrix}$$

$$\det = 15.5(4.5) - 6(6) = 33.75$$

$$(X^T X + n\lambda I)^{-1} = \frac{1}{33.75} \begin{pmatrix} 4.5 & -6 \\ -6 & 15.5 \end{pmatrix}$$

Step 3: solve for $\mathbf{w}_{\text{ridge}}^*$

$$\mathbf{w}_{\text{ridge}}^* = \frac{1}{33.75} \begin{pmatrix} 4.5 & -6 \\ -6 & 15.5 \end{pmatrix} \begin{pmatrix} 23 \\ 10 \end{pmatrix} = \frac{1}{33.75} \begin{pmatrix} 4.5(23) - 6(10) \\ -6(23) + 15.5(10) \end{pmatrix}$$

$$\mathbf{w}_{\text{ridge}}^* = \frac{1}{33.75} \begin{pmatrix} 43.5 \\ 17 \end{pmatrix} = \begin{pmatrix} 1.2889 \\ 0.5037 \end{pmatrix}$$

Step 4: compare to OLS

$$\text{OLS } (\lambda = 0): \mathbf{w}_{\text{ols}}^* = (X^\top X)^{-1} X^\top \mathbf{y} = (1.5, 0.3333).$$

Slope: 1.5 $\rightarrow$ 1.289 (**shrunk**). Inter-
cept: 0.333 $\rightarrow$ 0.504 (**grew**) — ridge
does not shrink every coefficient uniformly

It shrinks in the direction the penalty $\|\mathbf{w}\|_2^2$ actually penalizes, which redistributes weight between $X^\top X$'s directions, not just scaling every entry down equally.

Problem 2: bias-variance, same noise, different mix

Noise variance $\text{Var}[\varepsilon] = 2$. True value at a query point: $f^*(x_0) = 10$.

Model 1: always predicts the constant 7 (extreme high bias, zero variance)

Model 2: $\text{Bias}(x_0) = 1$, $\text{Var}_S[f_S(x_0)] = 6$

Compute total MSE for both. Which is lower? Does lower bias always mean lower total error?

Model 1: total error

$$\text{Bias}(x_0) = f^*(x_0) - 7 = 10 - 7 = 3, \quad \text{Bias}(x_0)^2 = 9.$$

$$\text{MSE}_1 = \text{Var}[\varepsilon] + \text{Bias}(x_0)^2 + \text{Var}_S[f_S(x_0)] = 2 + 9 + 0 = \mathbf{11}$$

A constant predictor has zero variance by construction — it never looks at the training sample at all.

Model 2: total error

$$\text{Bias}(x_0) = 1 \text{ (given)}, \quad \text{Var}_S[f_S(x_0)] = 6 \text{ (given)}.$$

$$\text{MSE}_2 = 2 + 1^2 + 6 = \mathbf{9}$$

Lower bias *and* lower total MSE than Model 1 — but only because its variance (6) is still small enough not to erase that advantage.

Lesson: lower bias does not mean lower total error

a model with $\text{Bias}(x_0) = 1$ but $\text{Var}_S = 10$ would have $\text{MSE} = 2 + 1 + 10 = 13$ — **worse** than Model 1, despite far lower bias

The tradeoff is real: bias and variance both cost something, and only their *sum* (plus irreducible noise) determines total error.

Problem 3: is this a valid kernel?

$$\kappa(x, x') = (xx' + 1)^2 \text{ on scalar in-}$$

puts $x_1 = 1, x_2 = 2, x_3 = -1$

1. Compute the 3×3 Gram matrix and check Mercer's condition via its eigenvalues.
2. Find an explicit feature map $\phi : \mathbb{R} \rightarrow \mathbb{R}^3$ with $\phi(x)^\top \phi(x') = \kappa(x, x')$.
3. Verify $\phi(1)^\top \phi(2)$ matches $\kappa(1, 2)$ directly.

Step 1: Gram matrix, Mercer check

$$K_{ij} = \kappa(x_i, x_j) : \quad K = \begin{pmatrix} 4 & 9 & 0 \\ 9 & 25 & 1 \\ 0 & 1 & 4 \end{pmatrix}$$

Compute the eigenvalues by solving the characteristic equation

$$\det(K - \lambda I) = 0 \quad \implies \quad (4 - \lambda)(\lambda^2 - 29\lambda + 18) = 0.$$

Mercer check: $\lambda(K) \approx \{0.63, 4.00, 28.37\}$;
all eigenvalues are nonnegative ✓

Therefore, K is symmetric and positive semidefinite, so Mercer's condition holds and a valid feature map exists.

Step 2: find the explicit feature map

General steps: (1) expand the kernel, (2) write each term as $f_j(x)f_j(x')$, and (3) collect the functions $f_j(x)$ into $\phi(x)$.

Expand $(xx' + 1)^2$ algebraically:

$$\begin{aligned}(xx' + 1)^2 &= x^2x'^2 + 2xx' + 1 \\ &= (x^2)(x'^2) + (\sqrt{2}x)(\sqrt{2}x') + (1)(1).\end{aligned}$$

$$\phi(x) = (x^2, \sqrt{2}x, 1) \in \mathbb{R}^3$$

Step 3: verify directly

$$\phi(1) = (1, \sqrt{2}, 1), \quad \phi(2) = (4, 2\sqrt{2}, 1).$$

$$\phi(1)^\top \phi(2) = 4 + 2(2) + 1 = 9 = \kappa(1, 2) = (1 \cdot 2 + 1)^2 = 9 \quad \checkmark$$

This is Lecture 04's kernel trick in reverse — confirming a valid κ really is some $\phi(x)^\top \phi(x')$, not just asserting PSD and moving on.

Problem 4: which interpolating solution does GD find?

One training constraint, two parameters: $w_1 + w_2 = 4$ (the overparameterized regime, Lecture 04). Two candidates, both fitting exactly: $\mathbf{w}_a = (2, 2)$ and $\mathbf{w}_b = (4, 0)$

1. Compute $\|\mathbf{w}_a\|_2, \|\mathbf{w}_b\|_2$. Which is smaller?
2. Find the minimum-norm solution via the pseudoinverse.
3. Predict what gradient descent from $\mathbf{w}_0 = 0$ converges to, and why.

Step 1: compare norms

$$\|\mathbf{w}_a\| = \sqrt{2^2 + 2^2} = 2\sqrt{2} \approx 2.83, \quad \|\mathbf{w}_b\| = \sqrt{4^2 + 0^2} = 4$$

$\mathbf{w}_a$ is smaller

Step 2: the pseudoinverse solution

$$X = (1, 1), \mathbf{y} = (4): \quad XX^\top = 1(1) + 1(1) = 2.$$

$$\mathbf{w}^* = X^\top (XX^\top)^{-1} \mathbf{y} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \cdot \frac{1}{2} \cdot 4 = \begin{pmatrix} 2 \\ 2 \end{pmatrix} = \mathbf{w}_a$$

the minimum-norm solution is exactly $\mathbf{w}_a$, not $\mathbf{w}_b$

Step 3: implicit regularization, confirmed

gradient descent from $\mathbf{w}_0 = 0$ on underdetermined least squares converges to the **minimum-norm** interpolating solution — here, $\mathbf{w}_a$, not $\mathbf{w}_b$, even though both fit the single constraint exactly

Verified numerically: 2,000 full-batch GD steps from $\mathbf{w}_0 = (0, 0)$ land at exactly $(2.0, 2.0)$ — matching $\mathbf{w}_a$, confirming the prediction rather than just asserting it.

This is *why* which interpolating solution you land on matters for generalization, not just whether you interpolate at all.

2. Exam 1 Problem Discussion

Sampling across Lectures 01–06

Exam 1 (10/16) covers Lectures 01–06: generalization/ evaluation, optimization and the three foundational models, bias-variance/regularization, feature engineering/kernels/double descent, SVM/multiclass, and unsupervised learning (PCA/K-means/GMM/EM)

Three problems today, chosen to *sample* across that range — not a complete review of every lecture.

Problem A: underfitting or something else?

A team fits a binary classification model: 61% train accuracy, 60% test accuracy, both on the same 2024 distribution. They conclude “no overfitting, so the model is fine.”

1. Is the conclusion right? What does a small gap rule out, and *not* rule out?
2. Which bias-variance term is most likely dominating, and why?
3. Name two fixes. Why is “get more training data” *not* one of them?

Part 1: what a small gap does (and doesn't) tell you

Correct that this rules out overfitting (a large train/test gap on the same distribution)

Wrong that this means the model is fine — a small gap says nothing about whether both numbers are simply *bad*. Here, 61% train accuracy is itself the problem.

Part 2: which term is dominating

high **bias** (underfitting): the model is too rigid to capture the true relationship, so it does poorly on train *and* test alike

Exactly Lecture 03's definition — the signature “low train, low test, small gap” pattern.

Part 3: more data does not fix bias

underfitting is a model-**capacity** problem, not a data-**quantity** problem — a model too simple to represent f^* stays too simple no matter how many more samples it sees (Lecture 03's bias term has no n in it)

Fixes: increase capacity via feature engineering (Lecture 03's explicit $\phi(\mathbf{x})$) or a richer kernel (Lecture 04), or reduce regularization strength λ if one is applied (Lecture 03).

Problem B: a 1D dataset no linear classifier separates

$$\begin{aligned}x_1 &= -1 \text{ (}\mathbf{y} = +1\text{),} & x_2 &= 0 \text{ (}\mathbf{y} = -1\text{),} \\x_3 &= 1 \text{ (}\mathbf{y} = +1\text{)} \text{ — one scalar feature}\end{aligned}$$

1. Argue no linear threshold on x alone separates these.
2. Propose a feature map $\phi(x)$ that makes the data separable, and verify it.
3. What is $\kappa(x, x') = \phi(x)\phi(x')$, and how does kernel SVM use it?

Part 1: why no linear threshold works

a linear threshold in 1D always separates a **low range** from a **high range** (or vice versa) — it cannot isolate a *middle* value ($x_2 = 0$) as one class while both *outer* values ($x_1 = -1, x_3 = 1$) share the other

True for any threshold (or pair of thresholds) defining a single linear (half-line) decision region.

Part 2: a feature map that works

$\phi(x) = x^2$: values become $(1, 0, 1)$ for (x_1, x_2, x_3) .

threshold $\phi > 0.5 \Rightarrow \mathbf{y} = +1$ separates all three
— **verified directly**: $\phi(x_1) = 1 > 0.5 (+1 \checkmark)$,
 $\phi(x_2) = 0 < 0.5 (-1 \checkmark)$, $\phi(x_3) = 1 > 0.5 (+1 \checkmark)$

Part 3: the kernel view

$$\kappa(x, x') = x^2 x'^2 = (xx')^2$$

kernel SVM (Lecture 05) needs only $\kappa(x_i, x_j)$ values to fit and predict — it never needs to materialize $\phi(x) = x^2$ explicitly

Here ϕ is only 1-dimensional, so materializing it costs nothing. The real payoff shows up when ϕ is high- or infinite-dimensional (Lecture 04's RBF kernel), where forming $\phi(x)$ explicitly is not an option at all.

Problem C: K-means, traced by hand

Points $(1, 1)$, $(1.5, 2)$, $(3, 4)$, $(5, 7)$; $K = 2$; initial centroids $\mu_1 = (1, 1)$, $\mu_2 = (5, 7)$ (two of the data points themselves)

1. Assignment step: compute squared distances, assign each point.
2. Update step: recompute both centroids as the mean of their assigned points.
3. Repeat the assignment step once more. Converged? Final objective?

Step 1: assignment

point	to $\mu_1 = (1, 1)$	to $\mu_2 = (5, 7)$	assign
(1, 1)	0	52	μ_1
(1.5, 2)	1.25	37.25	μ_1
(3, 4)	13	13	tie
(5, 7)	52	0	μ_2

point (3, 4) is an exact tie — broken here by convention (lower-indexed cluster wins, the same convention as Lecture 05's OvR/OvO ambiguous-region discussion), assigning it to μ_1

Step 2: update

Cluster 1 = $\{(1, 1), (1.5, 2), (3, 4)\}$: mean = $(1.833, 2.333)$.

Cluster 2 = $\{(5, 7)\}$: mean = $(5, 7)$ (unchanged, one point).

new centroids: $\mu_1 =$
 $(1.833, 2.333)$, $\mu_2 = (5, 7)$

Step 3: re-assign, check convergence

Recompute distances with the updated centroids.

every point's nearest centroid is unchanged
— points 1, 2, 3 $\rightarrow$ cluster 1; point 4 $\rightarrow$
cluster 2 — **same assignment as before**

The algorithm has converged after one full update.

Final objective, and what this trace shows

$$\sum_i \|\mathbf{x}_i - \mu_{z_i}\|^2 = 2.472 + 0.222 + 4.139 + 0 = \mathbf{6.833}$$

both the tie-breaking step and the one-update convergence are exactly the mechanics Lecture 06's convergence proof guarantees — monotonic decrease, finitely many steps, but nothing about *which* local optimum (a different initialization could group the points differently)

Wrap-up: HW2 review (1/2)

ridge shrinks coefficients, but not uniformly — solving the 2×2 system by hand makes that concrete rather than asserted

bias-variance is a genuine tradeoff: lower bias only wins if it doesn't cost too much variance — only the *sum* determines total error

Wrap-up: HW2 review (2/2)

checking Mercer's condition and finding an explicit ϕ are two views of the same fact: valid kernel $\Leftrightarrow$ some feature map's inner product

gradient descent's implicit bias toward the minimum-norm interpolant is not abstract — computed here directly, and confirmed by running GD

Wrap-up: Exam 1 discussion, and what's next

Exam 1 problems sampled deliberately across Lectures 01–06: underfitting diagnosis, the feature-map/kernel view of separability, and a full K-means trace with an explicit tie

Coming up: Fall Recess (10/9) — project topic and team due. Exam 1 (10/16) — Lectures 01–06.