13 Shrinkage: Ridge Regression, Subset Selection, and Lasso

RIDGE REGRESSION aka Tikhonov Regularization

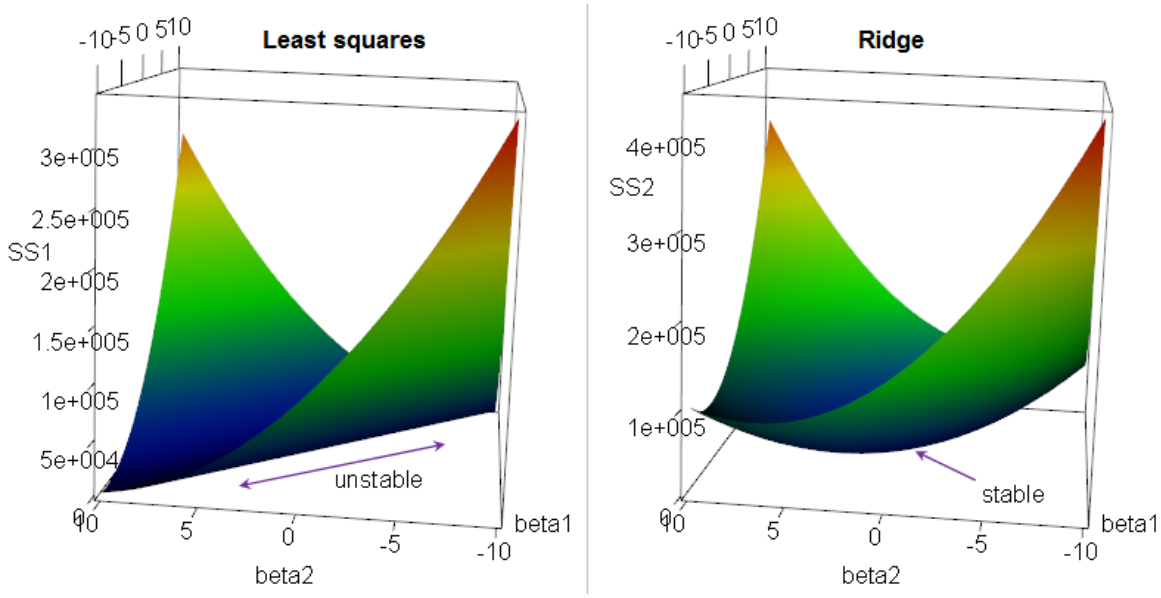
(1) + (A) + ℓ_2 penalized mean loss (d).

$$\boxed{\text{Find } w \text{ that minimizes } \|Xw - y\|^2 + \lambda \|w'\|^2} = J(w)$$

where w' is w with component α replaced by 0.
X has fictitious dimension but we DON'T penalize α.

Adds a regularization term, aka a penalty term, for shrinkage: to encourage small $\|w'\|$. Why?

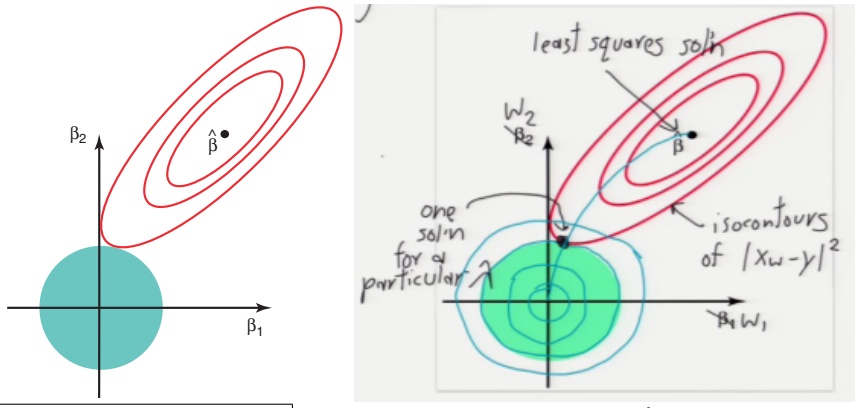
– Guarantees positive definite normal eq'ns; always unique solution.
[Standard least-squares linear regression yields singular normal equations when the sample points lie on a common hyperplane in feature space.] E.g., when $d > n$.



ridgequad.png [The cost function $J(w)$ with and without regularization.]

[At left, we see a quadratic form for a positive semidefinite cost function associated with least-squares regression. This cost function has many minima, and the regression problem is said to be ill-posed. By adding a small penalty term, we obtain a positive definite quadratic form (right), which has one unique minimum. The term "regularization" implies that we are turning an ill-posed problem into a well-posed problem.]
[That was the original motivation, but the next has become more important in machine learning . . .]

– Reduces overfitting by reducing variance. Why?
Imagine: $500x_1 - 500x_2$ is best fit for well-separated points with $y_i \in [0, 1]$.
Small change in x $\Rightarrow$ big change in y!
[Given that all the y values in the data are small and the x values are not, it's a sure sign of overfitting if tiny changes in x cause huge changes in y.]
So we penalize large weights.
[This use of regularization is closely related to the first one. When you have large variance and a lot of overfitting, it implies that your problem is *close to* being ill-posed, even though technically it might be well-posed.]



ridgeterms.pdf (ISL, Figure 6.7) [In this plot of weight space, $\hat{\beta}$ (read as "$\hat{w}$") is the least-squares solution. The red ellipses are the isocontours of $\|Xw - y\|^2$. The isocontours of $\|w'\|$ are circles centered at the origin (blue). The solution lies where a red isocontour just touches a blue isocontour tangentially. As λ increases, the solution will occur at a more outer red isocontour and a more inner blue isocontour. This helps to reduce overfitting.]

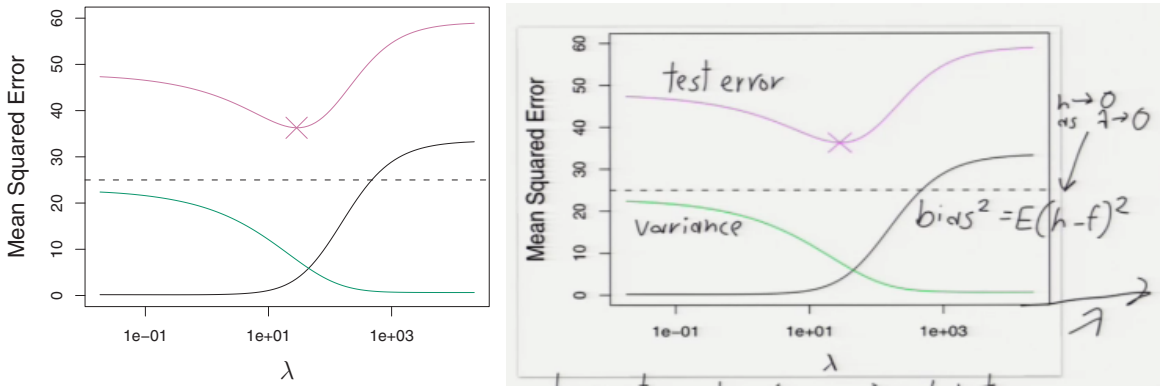
Setting $\nabla J = 0$ gives normal eq'ns

$$(X^\top X + \lambda I')\, w = X^\top y$$

where I' is identity matrix w/bottom right set to zero. [Don't penalize the bias term α.]
[Don't worry; $X^\top X + \lambda I'$ is always positive definite for $\lambda > 0$.]

Algorithm: Solve for w. Return $h(z) = w^\top z$.

Increasing λ $\Rightarrow$ more regularization; smaller $\|w'\|$
Recall [from the previous lecture] our data model $y = Xv + e$, where e is noise.
Variance of ridge regr. is $\mathrm{Var}(z^\top (X^\top X + \lambda I')^{-1} X^\top e)$.
As $\lambda \to \infty$, variance $\to 0$, but bias increases.



ridgebiasvar.pdf (ISL, Figure 6.5) [Plot of bias^2 & variance as λ increases.]

[So, as usual for the bias-variance trade-off, the test error as a function of λ is a U-shaped curve. We find the bottom by validation.]

λ is a hyperparameter; tune by (cross-)validation.

Ideally, features should be "normalized" to have same variance.
Alternative: use asymmetric penalty by replacing I' w/other diagonal matrix. [For example, if you use polynomial features, you could use different penalties for monomials of different degrees.]

Bayesian Justification for Ridge Reg.

Assign a prior probability on w': $w' \sim \mathcal{N}(0, \sigma^2)$. Apply MLE to the posterior prob.
[This prior probability says that we think weights close to zero are more likely to be correct.]

$$\begin{aligned}
\text{Bayes' Theorem: posterior } f(w|X,y) &= \frac{f(y|X,w) \times \text{ prior } f(w')}{f(y|X)} = \frac{\mathcal{L}(w)\, f(w')}{f(y|X)} \\
\text{Maximize log posterior} &= \ln \mathcal{L}(w) + \ln f(w') - \text{const} \\
&= -\text{const}\, \|Xw - y\|^2 - \text{const}\, \|w'\|^2 - \text{const} \\
&\Rightarrow \text{Minimize } \|Xw - y\|^2 + \lambda \|w'\|^2
\end{aligned}$$

This method (using likelihood, but maximizing posterior) is called maximum *a posteriori* (MAP).
[A prior probability on the weights is another way to understand regularizing ill-posed problems.]

FEATURE SUBSET SELECTION

[Some of you may have noticed as early as Homework 1 that you can sometimes get better performance on a spam classifier simply by dropping some useless features.]

All features increase variance, but not all features reduce bias.
Idea: Identify poorly predictive features, ignore them (weight zero).
Less overfitting, smaller test errors.
2nd motivation: Inference. Simpler models convey interpretable wisdom.
Useful in all classification & regression methods.
Sometimes it's hard: Different features can partly encode same information.
Combinatorially hard to choose best feature subset.

Alg: Best subset selection. Try all $2^d - 1$ nonempty subsets of features. [Train one classifier per subset.]
Choose the best classifier by (cross-)validation. Slow.

[Obviously, best subset selection isn't feasible if we have a lot of features. But it gives us an "ideal" algorithm to compare practical algorithms with. If d is large, there is no algorithm that's guaranteed to find the best subset and that runs in acceptable time. But heuristics often work well.]

Heuristic 1: Forward stepwise selection.
Start with null model (0 features); repeatedly add best feature until validation errors start increasing (due to overfitting) instead of decreasing. At each outer iteration, inner loop tries every feature & chooses the best by validation. Requires training $O(d^2)$ models instead of $O(2^d)$.
Not perfect: e.g., won't find the best 2-feature model if neither of those features yields the best 1-feature model. [That's why it's a heuristic.]

Heuristic 2: Backward stepwise selection.
Start with all d features; repeatedly remove feature whose removal gives best reduction in validation error.
Also trains $O(d^2)$ models.
Additional heuristic: Only try to remove features with small weights.
Q: small relative to what?
Recall: variance of least-squ. regr. is proportional to $\sigma^2(X^\top X)^{-1}$
z-score of weight w_i is $z_i = \frac{w_i}{\sigma\sqrt{v_i}}$ where v_i is ith diagonal entry of $(X^\top X)^{-1}$.
Small z-score hints "true" w_i could be zero.

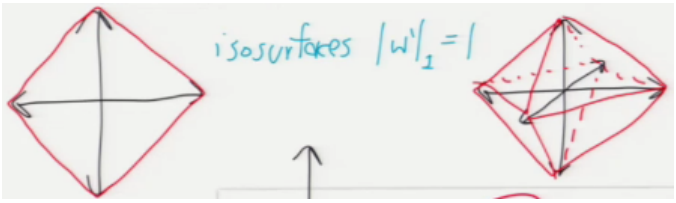
[Forward stepwise is a better choice when you suspect only a few features will be good predictors; e.g., spam. Backward stepwise is better when most features are important. If you're lucky, you'll stop early.]

LASSO (Robert Tibshirani, 1996)

Regression w/regularization: (1) + (A) + ℓ_1 penalized mean loss (e).
"Least absolute shrinkage and selection operator"
[This is a regularized regression method similar to ridge regression, but it has the advantage that it often naturally sets some of the weights to zero.]

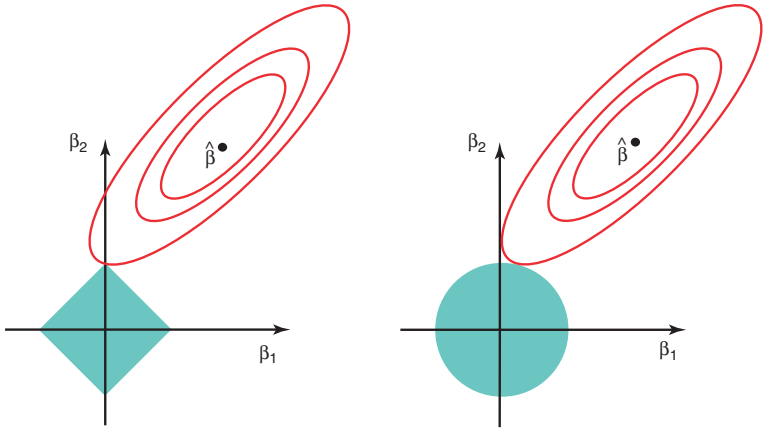
Find w that minimizes $\|Xw - y\|^2 + \lambda \|w'\|_1$ where $\|w'\|_1 = \sum_{i=1}^{d} |w_i|$ (Don't penalize α.)

Recall ridge regr.: isosurfaces of $\|w'\|^2$ are hyperspheres.
The isosurfaces of $\|w'\|_1$ are cross-polytopes.
The unit cross-polytope is the convex hull of all the positive & negative unit coordinate vectors.



[Draw this figure by hand crosspolys.png]

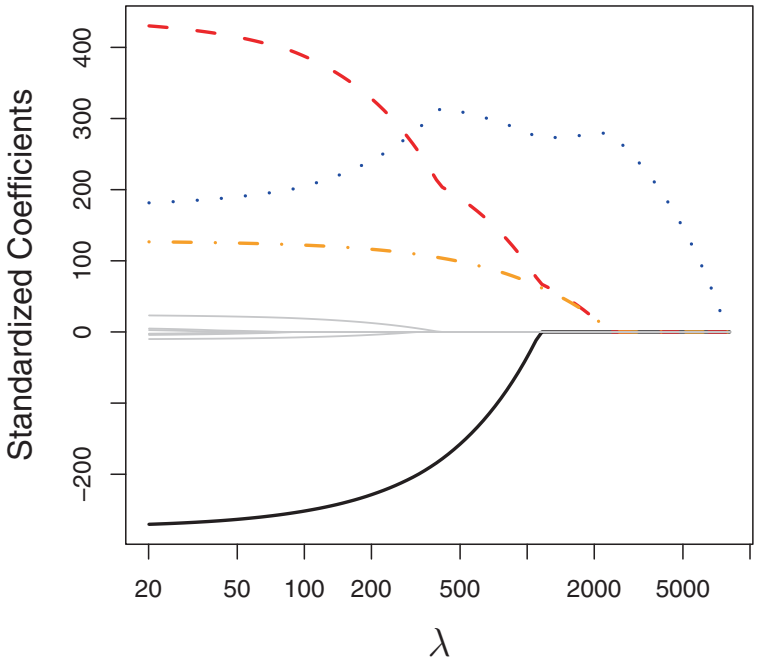
[You get larger and smaller cross-polytope isosurfaces by scaling these.]



lassoridge.pdf [Isocontours of the terms of the objective function for the Lasso appear at left. Compare with the ridge regression isocontours at right.]

[The red ellipses are the isocontours of $\|Xw - y\|^2$, and the least-squares solution lies at their center. The isocontours of $\|w'\|_1$ are diamonds centered at the origin (blue). The solution lies where a red isocontour just touches a blue diamond. What's interesting here is that in this example, the red isocontour touches just the tip of the diamond. So the weight w_1 gets set to zero. That's what we want to happen to weights that don't have enough influence. This doesn't always happen—for instance, the red isosurface could touch a side of the diamond instead of a tip of the diamond.]

[When you go to higher dimensions, you might have several weights set to zero. For example, in 3D, if the red isosurface touches a sharp vertex of the cross-polytope, two of the three weights get set to zero. If it touches a sharp edge of the cross-polytope, one weight gets set to zero. If it touches a flat side of the cross-polytope, no weight is zero.]



lassoweights.pdf (ISL, Figure 6.6) [Weights as a function of λ.]

[This shows the weights for a typical linear regression problem with about 10 variables. You can see that as lambda increases, more and more of the weights become zero. Only four of the weights are really useful for prediction; they're in color. Statisticians used to choose λ by looking at a chart like this and trying to eyeball a spot where there aren't too many predictors and the weights aren't changing too fast. But nowadays they prefer validation.]

Sometimes sets some weights to zero, especially for large λ.
Algs: subgradient descent, least-angle regression (LARS), forward stagewise

[Lasso can be reformulated as a quadratic program, but it's a quadratic program with 2^d constraints, because a d-dimensional cross-polytope has 2^d facets. In practice, special-purpose optimization methods have been developed for Lasso. I'm not going to teach you one, but if you need one, look up the last two of these algorithms. LARS is built into the R Programming Language for statistics.]

[As with ridge regression, you should probably normalize the features first before applying Lasso.]