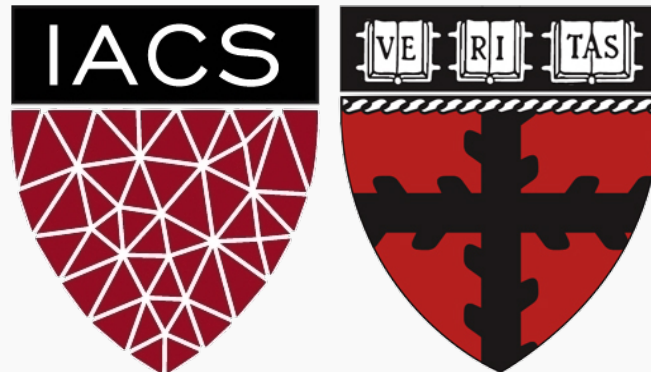


Advanced Section #3: Methods of Regularization and their justifications

Robbert Struyven and Pavlos Protopapas (viz. Camilo Fosco)

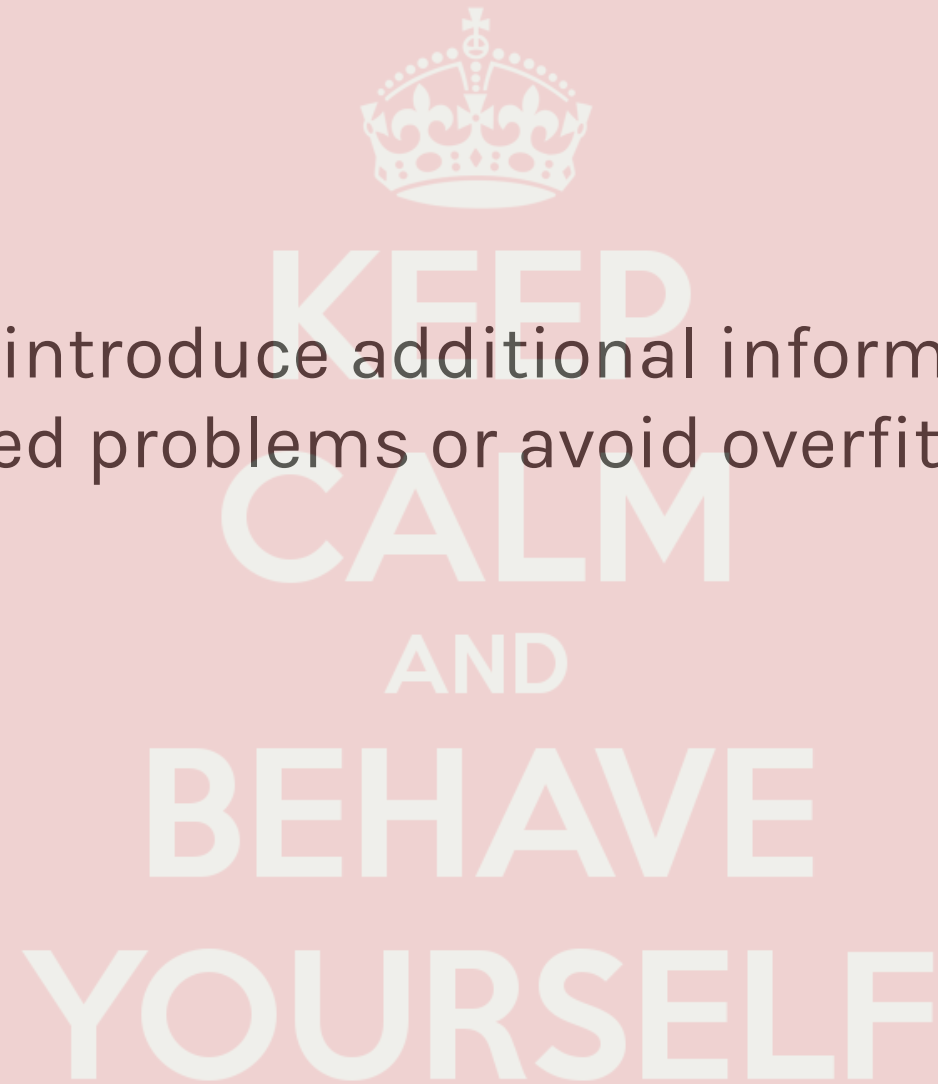
CS109A Introduction to Data Science
Pavlos Protopapas , Kevin Rader and Chris Tanner



Outline

- Motivation for regularization
 - Generalization
 - Instability
- Ridge estimator
- Lasso estimator
- Elastic Net estimator
- Visualizations
- Bayesian approach

Regularization: introduce additional information to solve ill-posed problems or avoid overfitting.

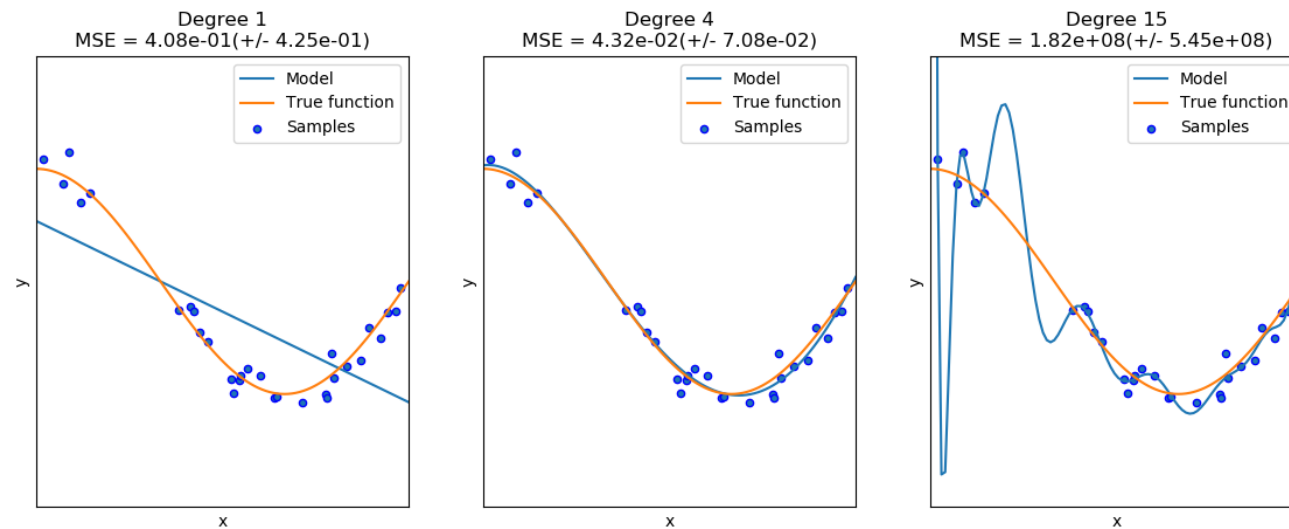


MOTIVATION

Why do we regularize?

Generalization

- Avoid overfitting. Reduce features that have weak predictive power.
- Discourage the use of a model that is too complex.
- Do not fit the noise!



Instability issues

- Linear regression becomes unstable when p (degrees of freedom) is close to n (observations).
 - Think about each obs. as a piece of info about the model. What happens when n is close to the degrees of freedom?
- Collinearity generates instability issues.
 - If we want to understand the effect of X_1 and X_2 on Y , is it easier when they vary together or when they vary separately?
- Regularization helps combat instability by constraining the space of possible parameters.
- Mathematically, instability can be seen through the estimator's variance:
$$\text{var}(\hat{\beta}) = \sigma^2 (X^T X)^{-1}$$

Instability issues

$$\text{var}(\hat{\beta}) = \sigma^2 (X^T X)^{-1}$$

var(Y)

The variance of the estimator is affected by the irreducible noise of the model. We have no control over this.

Inverse of

Gram matrix

But the variance also depends on the predictors themselves! This is the important part.

- if the eigenvalues of $X^T X$ are close to zero, our matrix is almost singular. One or more eigenvalues of $(X^T X)^{-1}$ can be extremely large.
- In that case, on top of having large variance, we have numerical instability.
- In general, we want the condition number of $X^T X$ to be small (well-conditioning).

Remember that for $X^T X$: $\kappa(X^T X) = \frac{\lambda_{max}}{\lambda_{min}}$

Instability and the condition number

More formally, instability can be analyzed through perturbation theory.

Consider the following least-squares problem:

$$\min_{\beta} \|(X + \delta X)\beta - (Y + \delta Y)\|$$

Perturbations

If $\tilde{\beta}$ is the solution of the original least squares problem, we can prove that:

$$\frac{\|\beta - \tilde{\beta}\|}{\|\beta\|} \leq \sqrt{\kappa(X^T X)} \frac{\|\delta X\|}{\|X\|}$$

Condition number of $X^T X$

Small $\kappa(X^T X)$ tightens the bound on how much my coefficients can vary.

Instability visualized

- Instability can be visualized by regressing on nearly colinear data, and observing the changes on the same data, slightly perturbed

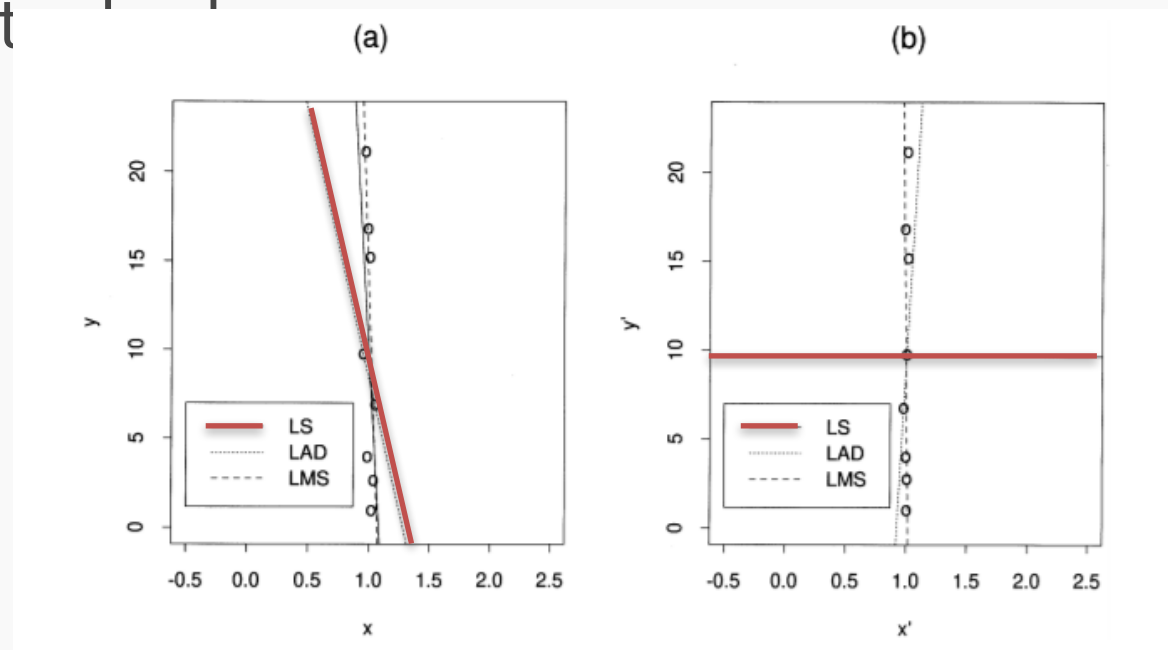


Image from “*Instability of Least Squares, Least Absolute Deviation and Least Median of Squares Linear Regression*”, Ellis et al. (1998)

Motivation in short

- We want **less complex models** to avoid overfitting and increase interpretability.
- We want to be able to solve problems where $p = n$ or $p > n$, and still generalize reasonably well.
- We want to **reduce instability** (increase min eigenvalue/reduce condition number) in our estimators. We need to be better at estimating betas with colinear predictors.
- In a nutshell, we want to avoid **ill-posed problems** (no solutions / solutions not unique / unstable solutions)

RIDGE REGRESSION

Instability destroyer

What is the Ridge estimator?

- Regularized estimator proposed by **Hoerl** and **Kennard** (1970).
- Imposes L2 penalty on the magnitude of the coefficients.

$$\hat{\beta}_{Ridge} = \operatorname{argmin}_{\beta} \|X\beta - Y\|_2^2 + \lambda \|\beta\|_2^2$$

$$\hat{\beta}_{Ridge} = (X^T X + \lambda I)^{-1} X^T Y$$

Regularization factor

- In practice, the ridge estimator reduces the complexity of the model by shrinking the coefficients, but it doesn't nullify them.
- The lambda factor controls the amount of regularization.

Deriving the Ridge estimator

$(X^T X)^{-1}$ is considered unstable (or super-collinear) if eigenvalues are close to zero.

$$(X^T X)^{-1} = \underline{Q \Lambda^{-1} Q^{-1}}$$

Eigendecomposition

$$\Lambda^{-1} = \begin{bmatrix} k_1^{-1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & k_p^{-1} \end{bmatrix}.$$

If the eigenvalues k_i are close to zero, Λ^{-1} will have extremely large diagonal values. $(X^T X)^{-1}$ will be very hard to find numerically.

What can we do?

Deriving the Ridge estimator

Just add a constant to the eigenvalues.

$$Q(\Lambda^{-1}Q^{-1} + \lambda I)Q^{-1} = Q\Lambda^{-1}Q^{-1} + \lambda QQ^{-1} = X^T X + \lambda I$$

↑
Added
constant λ

We can find a new estimator:

$$\hat{\beta}_{Ridge} = (X^T X + \lambda I)^{-1} X^T Y$$

Properties: shrinks the coefficients

The Ridge estimator can be seen as a modification of the OLS estimator:

$$\hat{\beta}_{Ridge} = (I + \lambda(X^T X)^{-1})^{-1} \hat{\beta}_{OLS}$$

Let's look at an example to see its effect on the OLS betas: univariate case ($X = (x_1, \dots, x_n)$) with normalized predictor ($\|X\|_2^2 = X^T X = 1$).

In this case, the ridge estimator is:

$$\hat{\beta}_{Ridge} = \frac{\hat{\beta}_{OLS}}{1 + \lambda}$$

As we can see, Ridge regression shrinks the OLS predictors, but does not nullify them.



Properties: closer to the real beta

- Interesting theorem: there always exists $\lambda > 0$ such that:

$$E \left[\|\hat{\beta}_R - \beta\|_2^2 \right] < E \left[\|\hat{\beta}_{OLS} - \beta\|_2^2 \right]$$

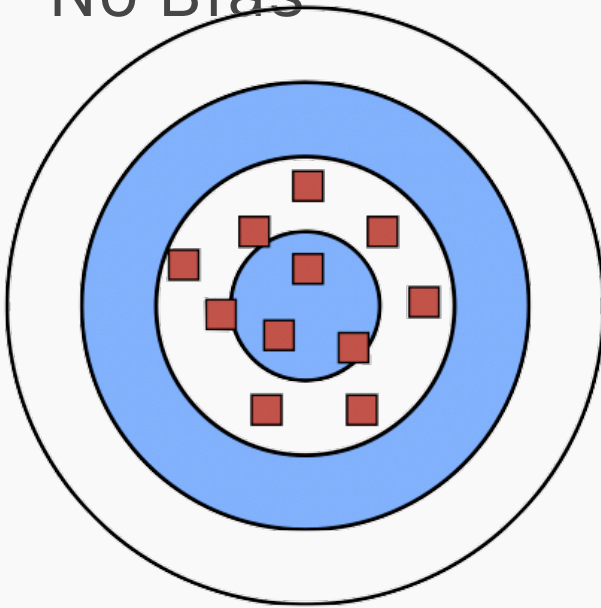
- Regardless of X and Y, there is a value of lambda for which **Ridge performs better than OLS** in terms of MSE.
- Careful: we're talking about MSE in estimating the true coefficient (**inference**), not performance in terms of **prediction**.
- OLS is unbiased, Ridge is not, however estimation is better: Ridge's lower variance more than makes up for increase in bias.

Good **bias-variance tradeoff**.

Good bias-variance tradeoff.

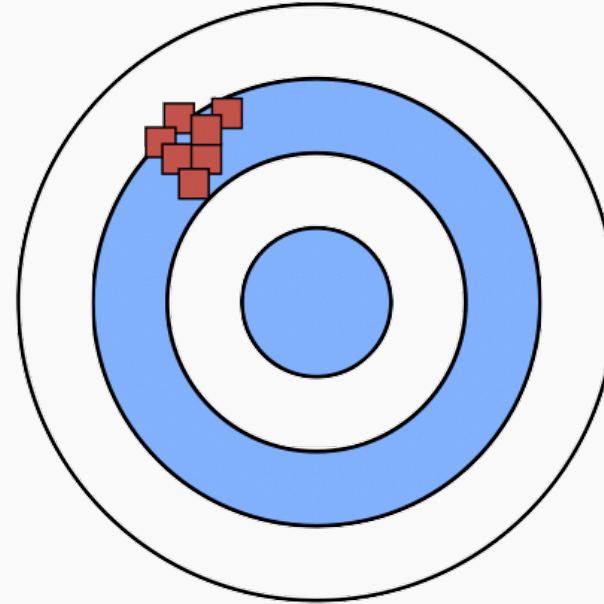
OLS

- Higher Variance (instable Betas)
- No Bias



Ridge

- Lower Variance
- Adding some Bias



Different perspectives on Ridge

- So far, we understand Ridge as a penalty on the optimization objective:

$$\hat{\beta}_{Ridge} = \operatorname{argmin}_{\beta} \|X\beta - Y\|_2^2 + \lambda \|\beta\|_2^2$$

However, there are multiple ways to look at it:

- Transformation (shrinkage) of OLS estimator.
- Estimator obtained from increased eigenvalues of $X^T X$ (better conditioning)
- Normal prior on coefficients (Bayesian interpretation)
- Constraint for curvature on the loss function
- Regression with dummy data
- Special case of Tikhonov Regularization
- Constrained minimization

Optimization perspective

The ridge regression problem is equivalent to the following constrained optimization problem:

$$\min_{\|\beta\|_2^2 \leq \kappa} \|Y - X\beta\|_2^2$$

- From this perspective, we are doing regular least squares with a **constraint on the magnitude of β** .
- We can get from one expression to the other through **Lagrange multipliers**.
- **Inverse relationship** between κ and λ . Namely, $\kappa = \|\hat{\beta}_{Ridge}^*(\lambda)\|_2^2$

Ridge, formal perspective

Ridge is a special case of Tikhonov Regularization:

Tikhonov Matrix

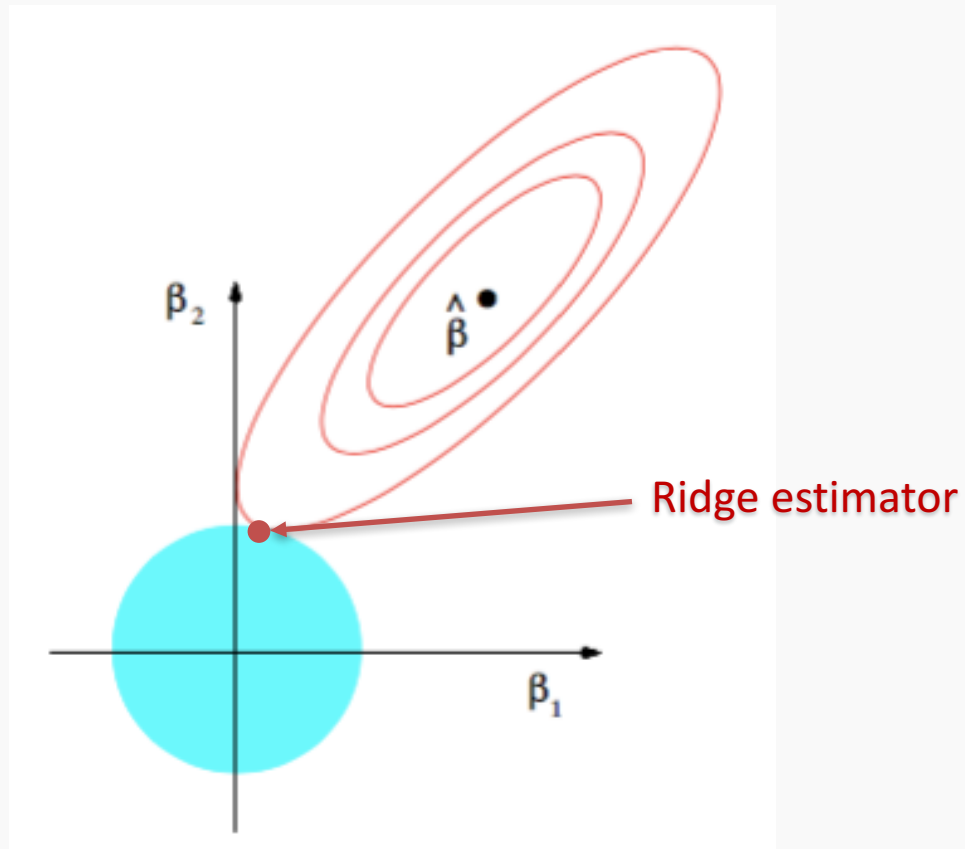
$$\|A\mathbf{x} - \mathbf{b}\|_2^2 + \|\Gamma\mathbf{x}\|_2^2$$
$$\mathbf{x} = (A^T A + \Gamma^T \Gamma)^{-1} A^T \mathbf{b}$$



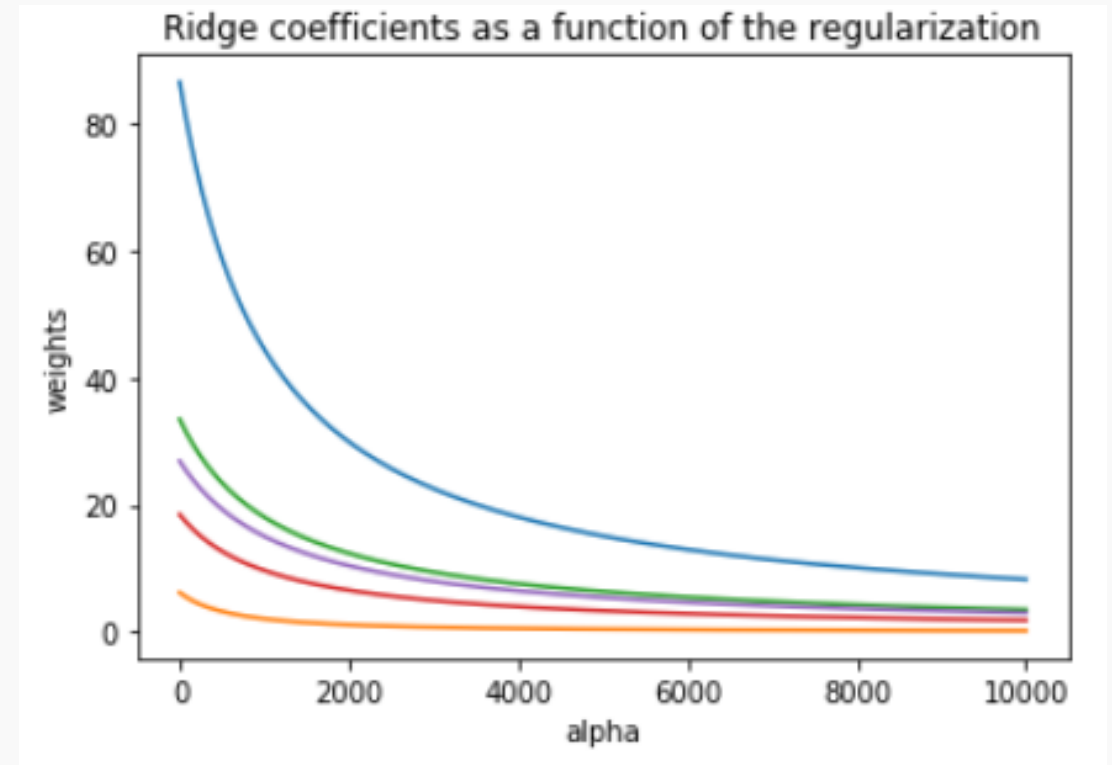
If $\Gamma = \sqrt{\lambda}I$, we have classic Ridge regression.

Tikhonov regularization is interesting, as we can use Γ to generate other constraints, such as smoothness in the estimator values.

Ridge visualized

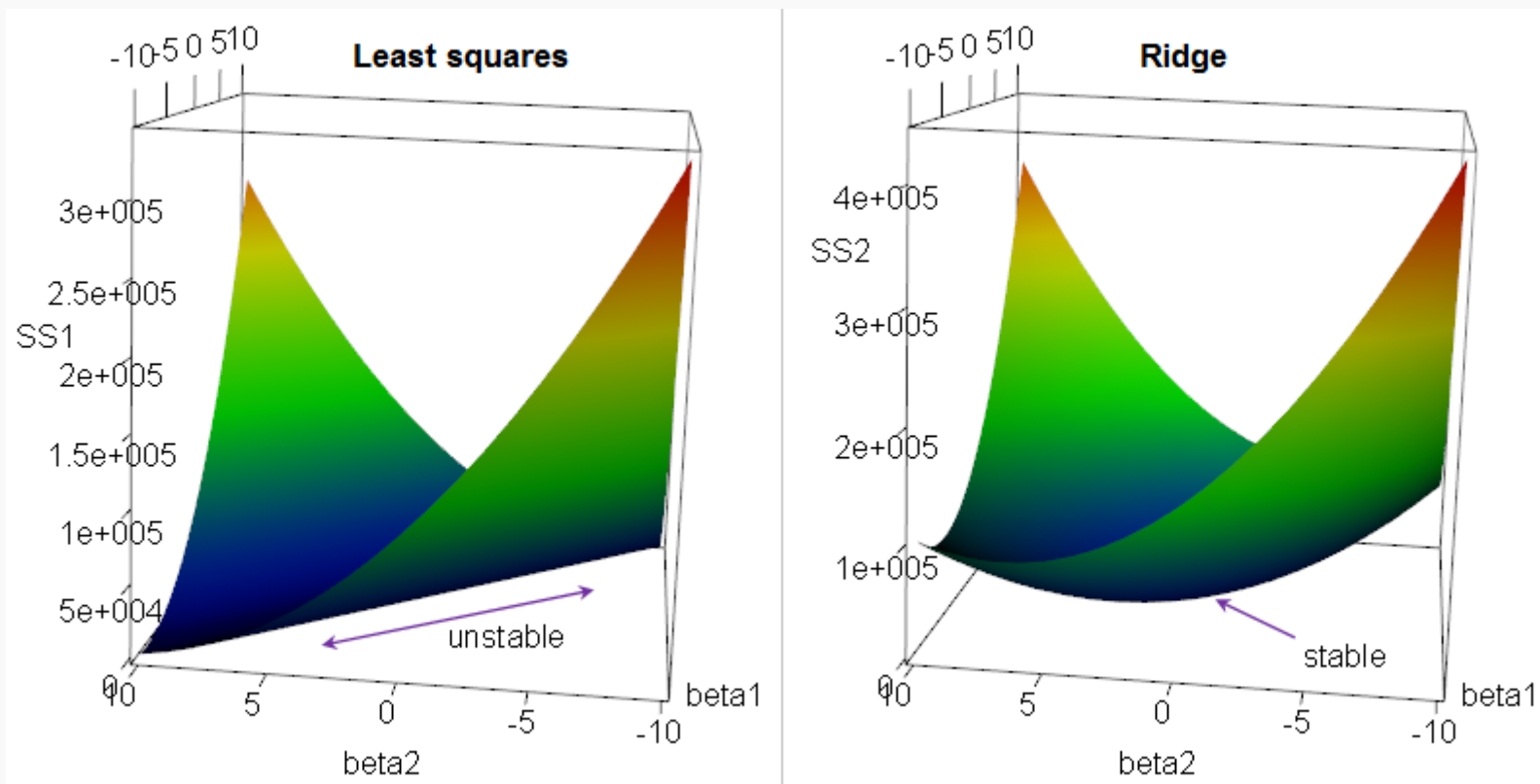


The ridge estimator is where the constraint and the loss intersect.



The values of the coefficients decrease as lambda increases, but they are not nullified.

Ridge visualized



Ridge curves the loss function in colinear problems, avoiding instability.

LASSO REGRESSION

Yes, LASSO is an acronym

What is LASSO?

- Least Absolute Shrinkage and Selection Operator
- Originally introduced in geophysics paper from 1986 but popularized by Robert Tibshirani (1996)
- Idea: L1 penalization on the coefficients.

$$\beta_{LASSO} = \operatorname{argmin}_{\beta} \|X\beta - Y\|_2^2 + \lambda \|\beta\|_1$$

- Remember that $\|\beta\|_1 = \sum_i |\beta_i|$
- This looks deceptively similar to Ridge, but behaves very differently. Tends to zero-out coefficients.

Deriving the LASSO estimator

The original LASSO definition comes from the constrained optimization problem:

$$\min_{\|\beta\|_1 \leq \kappa} \|X\beta - Y\|_2^2$$

This is similar to Ridge.

We should be able to easily find a closed form solution like Ridge, right?

No.

Subgradient to the rescue

- LASSO has no conventional analytical solution, as the L1 norm has no derivative at 0. We can, however, use the concept of **subdifferential** or **subgradient** to find a manageable expression.
- Let h be a convex function. The **subgradient** at point x_0 in the domain of h is equal to the set:

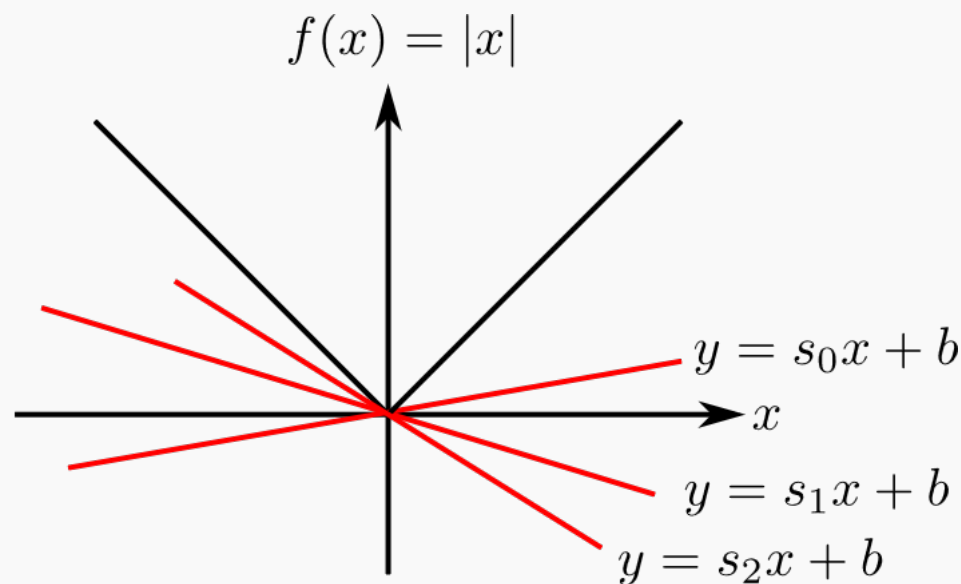
$$\partial(h)(x_0) = \left\{ c \in \mathbb{R} \text{ s.t. } c \leq \frac{h(x) - h(x_0)}{x - x_0} \quad \forall x \in \text{Dom}(h) \right\}$$

Subgradient to the rescue

In a nutshell, it is the set of **all slopes which are tangent to the function at the point x_0** .

For example, the subdifferential of the absolute value function is:

$$\triangleright \partial(|\cdot|)(x) = \begin{cases} -1 & x < 0 \\ [-1, 1] & x = 0 \\ 1 & x > 0 \end{cases}$$



Deriving LASSO

With this tool, we can find a solution for the case where the predictors are uncorrelated and normalized (X is orthonormal).

We have $X^T X = I$, so we minimize:

$$f(\beta) = \|X\beta - Y\|_2^2 + \lambda \|\beta\|_1$$

$$f(\beta) = \beta^T \beta - 2\beta^T X^T Y + Y^T Y + 2\lambda' \|\beta\|_1$$

Where $\lambda' = \frac{\lambda}{2}$ to simplify the equations.

Deriving LASSO

The i -th component of the subdifferential is then given by:

$$\partial(f)(\beta_i) = \begin{cases} 2\beta_i - 2x_i^T y + \lambda, & \beta_i > 0 \\ [-\lambda, \lambda] - 2x_i^T y, & \beta_i = 0 \\ 2\beta_i - 2x_i^T y - \lambda, & \beta_i < 0 \end{cases}$$

If we manage to make these equations zero for all i , we have found the LASSO estimator.

Deriving LASSO

Cases one and three can be solved easily, and yield:

$$\begin{aligned}\beta_i &= x_i^T y - \lambda' & \text{if } x_i^T y > \lambda' \\ \beta_i &= x_i^T y + \lambda' & \text{if } -x_i^T y > \lambda'\end{aligned}$$

Which can be translated into:

$$\beta_i = x_i^T y - \text{sign}(x_i^T y) \cdot \lambda' \quad \text{if } |x_i^T y| > \lambda'$$

Deriving LASSO

For the last case ($\beta_i = 0$), we need:

$$0 \in [-2\lambda', 2\lambda'] - 2x_i^T y$$

Which implies:

$$-2\lambda' - 2x_i^T y < 0 \Leftrightarrow \lambda' > -x_i^T y$$

$$2\lambda' - 2x_i^T y > 0 \Leftrightarrow \lambda' > x_i^T y$$

Deriving LASSO

This gives us a closed form for the LASSO estimation when $X^T X = I$:

$$\hat{\beta}_i^{\lambda'} = \begin{cases} 0 & \lambda' > |x_i^T y| \\ x_i^T y - \text{sign}(x_i^T y) \cdot \lambda', & \lambda' \leq |x_i^T y| \end{cases}$$

- As we can see, LASSO nullifies components of β when the corresponding $|x_i^T y|$ is smaller than $\lambda/2$.
- Both shrinkage and variable selection can be seen.

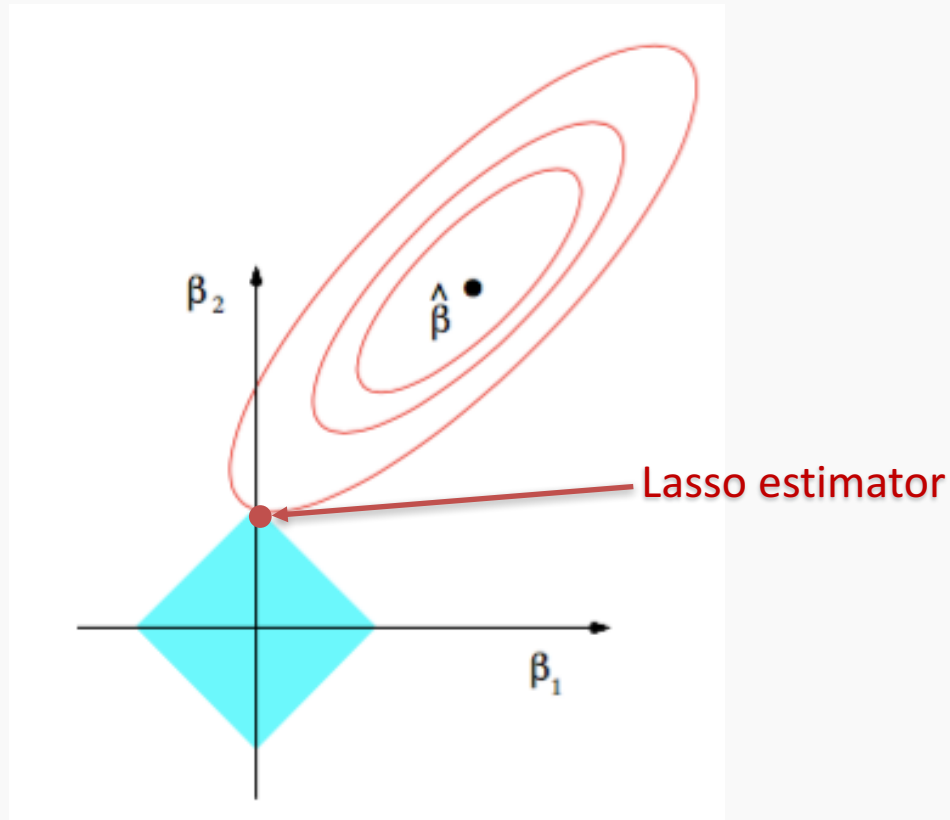
Connections of LASSO with OLS

The previous equation gives us the connection to OLS (when $X^T X = I$):

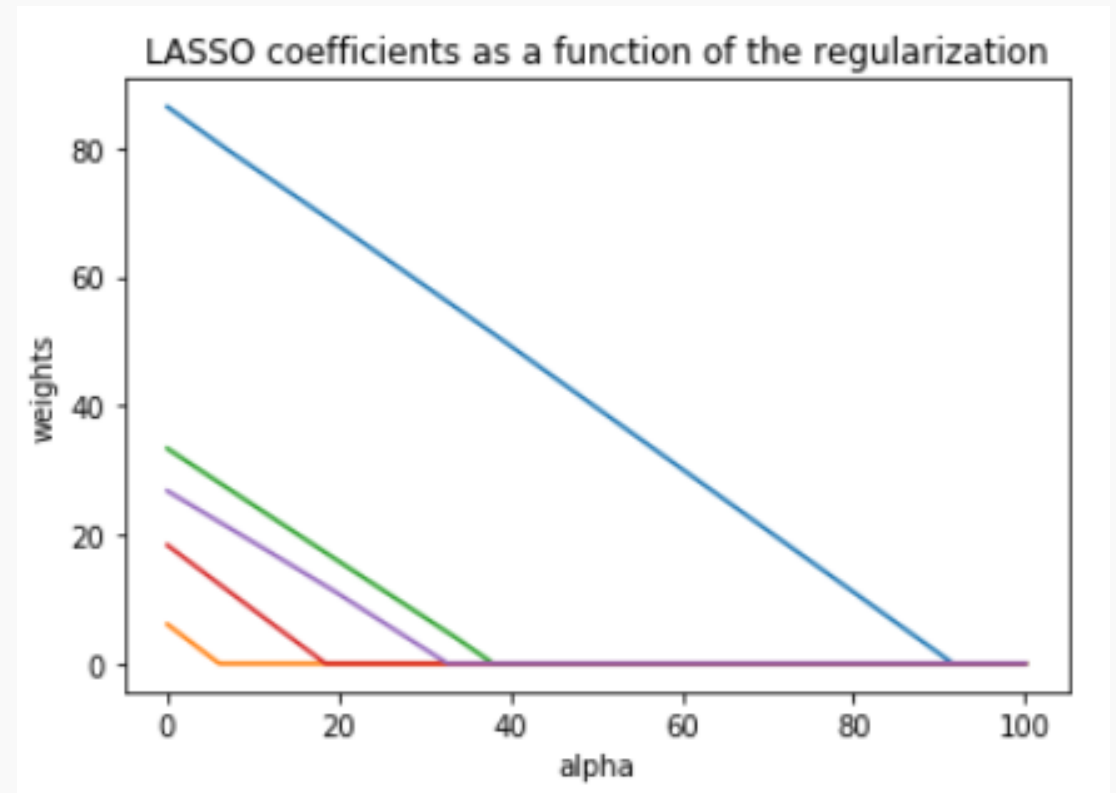
$$\beta_{LASSO_i} = \text{sign}(\hat{\beta}_{OLS_i}) \left[|\hat{\beta}_{OLS_i}| - \frac{\lambda}{2} \right]^+$$

Again, it is easy to see that LASSO reduces the coefficients and zeroes them out if they are too small.

LASSO visualized



The Lasso estimator tends to zero out parameters as the OLS loss can easily intersect with the constraint on one of the axis.



The values of the coefficients decrease as lambda increases, and are nullified fast.

ELASTIC NET ESTIMATOR

Estimators, assemble

Problems with Ridge and LASSO

- Ridge does not perform feature selection.
- Ridge and Lasso are sensible to outliers.
- When $p > n$, LASSO can choose at most n predictors to use. The rest are nullified.
- When there are multiple correlated predictors, LASSO tends to indifferently choose one and discard the rest.
- For example, if you run a problem with large number features multiple times, you might have a very different feature set each time.

Combine Ridge and LASSO!

In light of these points, Zou and Hastie developed the Elastic Net (EN) estimator in 2005.

The basic idea of EN is simple: **add both regularization terms** to the minimization objective.

$$\hat{\beta}_{EN} = \arg \min_{\beta} \|X\beta - Y\|_2^2 + \lambda_1 \|\beta\|_1 + \lambda_2 \|\beta\|_2^2$$

LASSO Ridge

EN tries to capture the best of both worlds: it **increases stability** in the estimation, reduces model complexity by shrinking the parameters and also performs **feature selection**.

Combine Ridge and LASSO!

EN can be rewritten as:

$$\hat{\beta}_{EN} = \arg \min_{\beta} \|X\beta - Y\|_2^2 + \lambda [\alpha \|\beta\|_1 + (1 - \alpha) \|\beta\|_2^2]$$

Where $\lambda = \lambda_1 + \lambda_2$ and $\alpha = \frac{\lambda_1}{\lambda_1 + \lambda_2}$.

Elastic Net can be seen as combining both penalties in one regularization term, which is a **convex combination** of LASSO and Ridge.

Combine Ridge and LASSO!

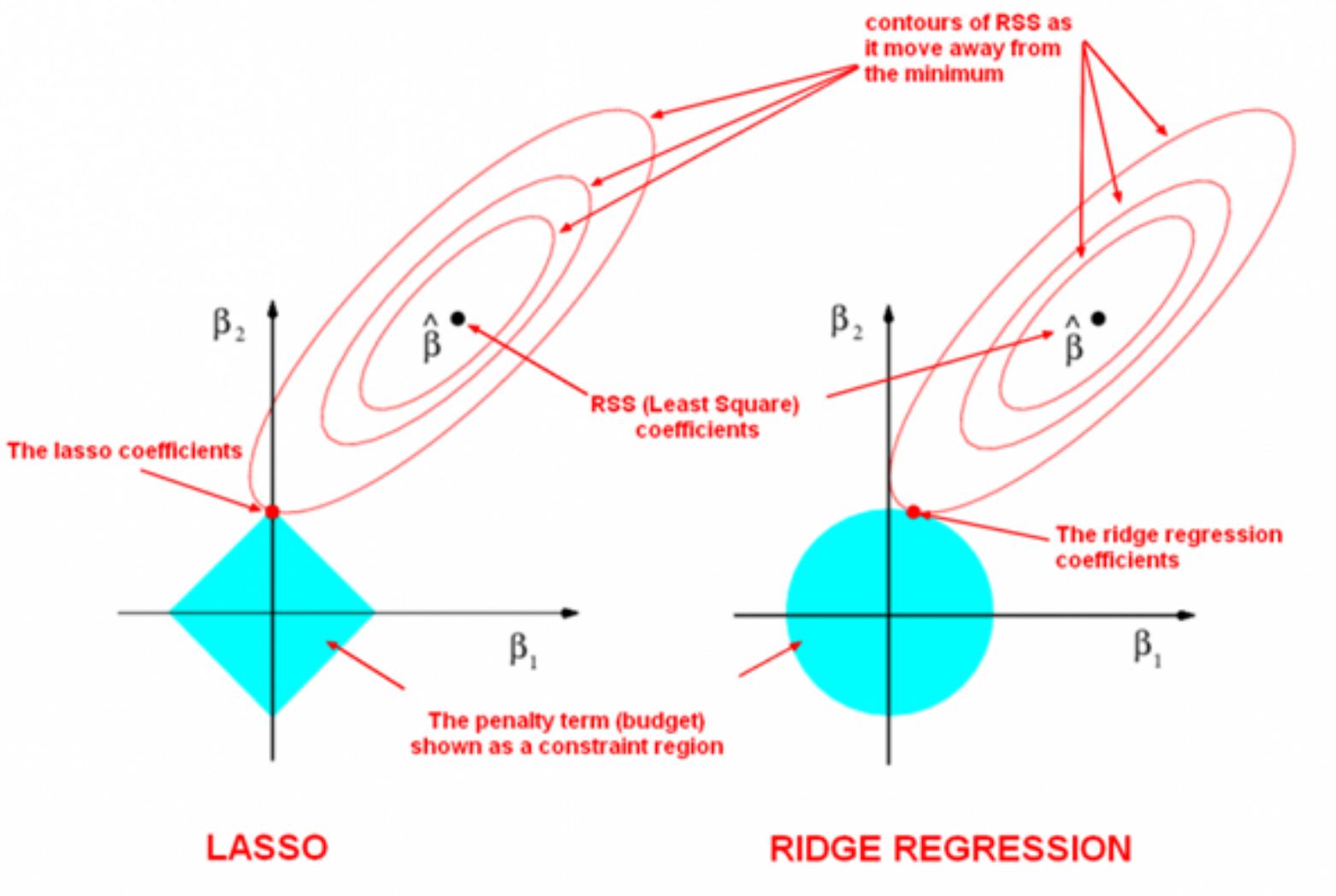
Again, the estimator can be seen as a constrained optimization problem:

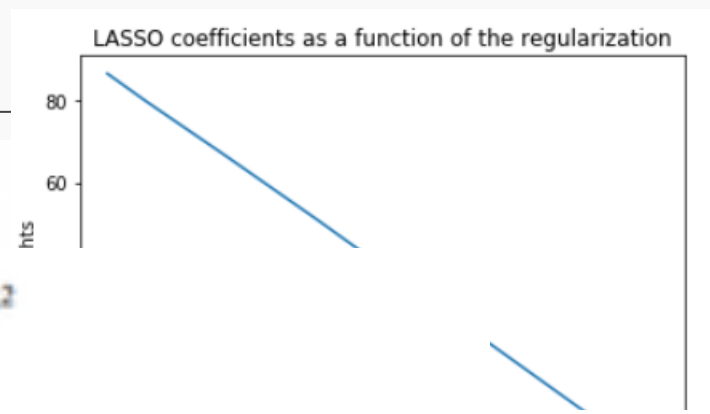
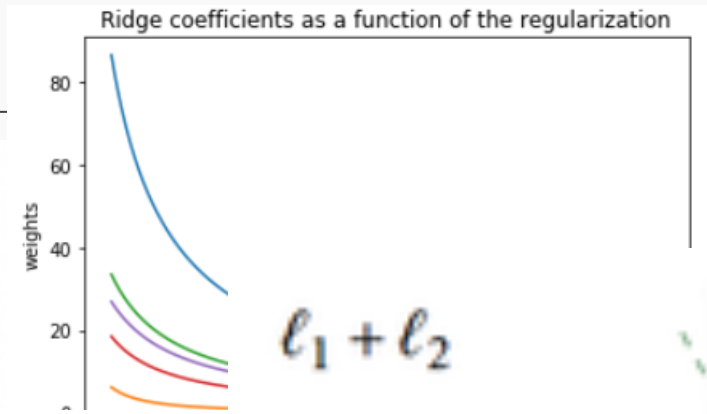
$$\min_{\alpha \|\beta\|_1 + (1-\alpha) \|\beta\|_2^2 \leq t} \|X\beta - Y\|_2^2$$

Where $\alpha \in [0,1]$. We can see that Ridge and LASSO are special cases of EN, where $\alpha = 1$ and $\alpha = 0$ respectively.

GEOMETRY OF ESTIMATORS

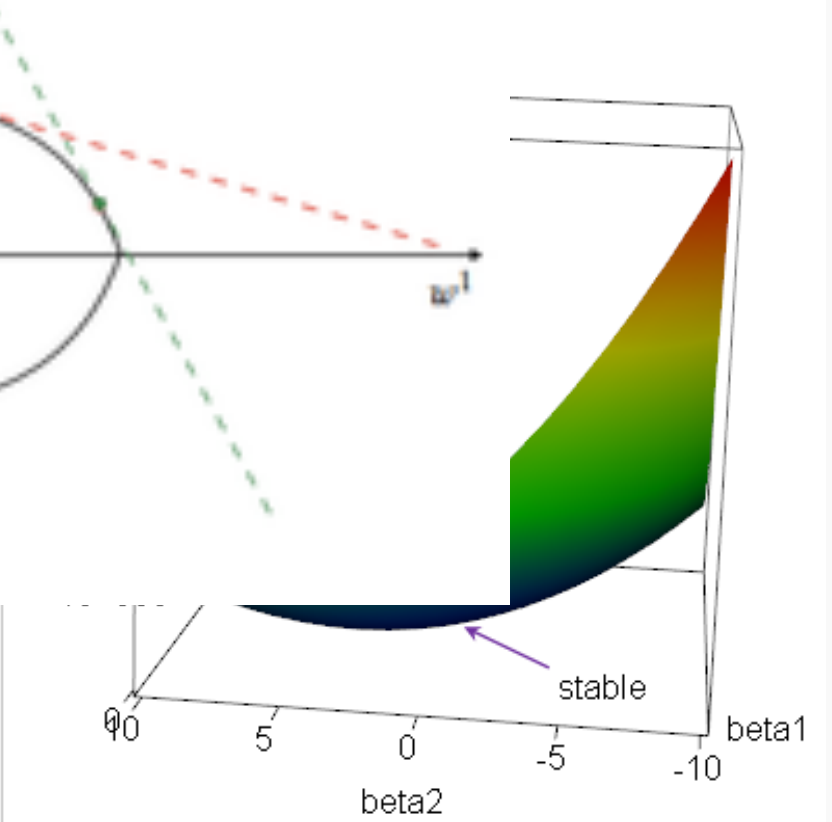
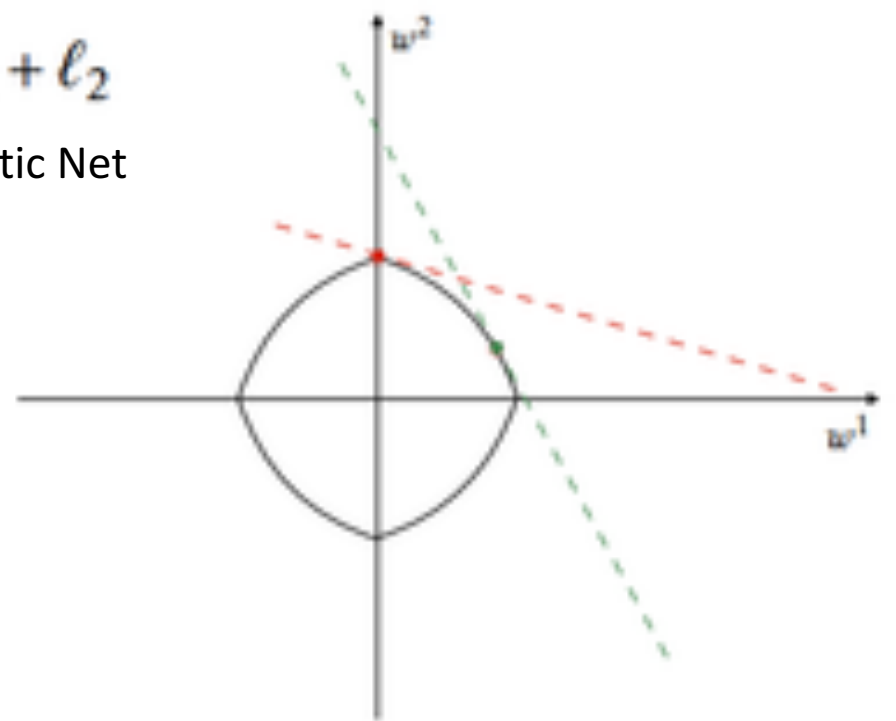
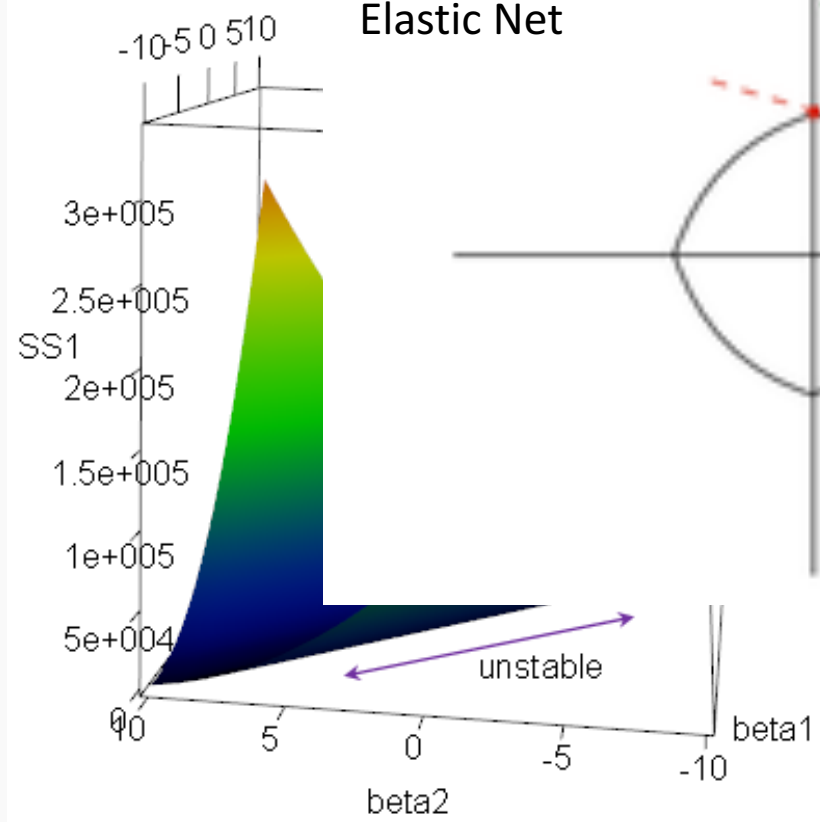
Visualization is key





$l_1 + l_2$

Elastic Net



Let's see it live!

DEMO TIME

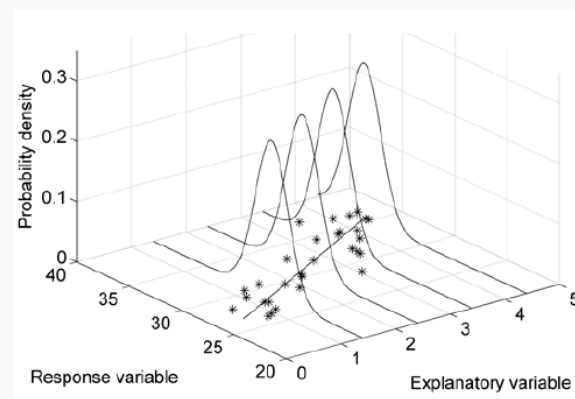
BAYESIAN INTERPRETATIONS

“The right way of looking at it” - Kevin Rader, probably

A different but useful perspective

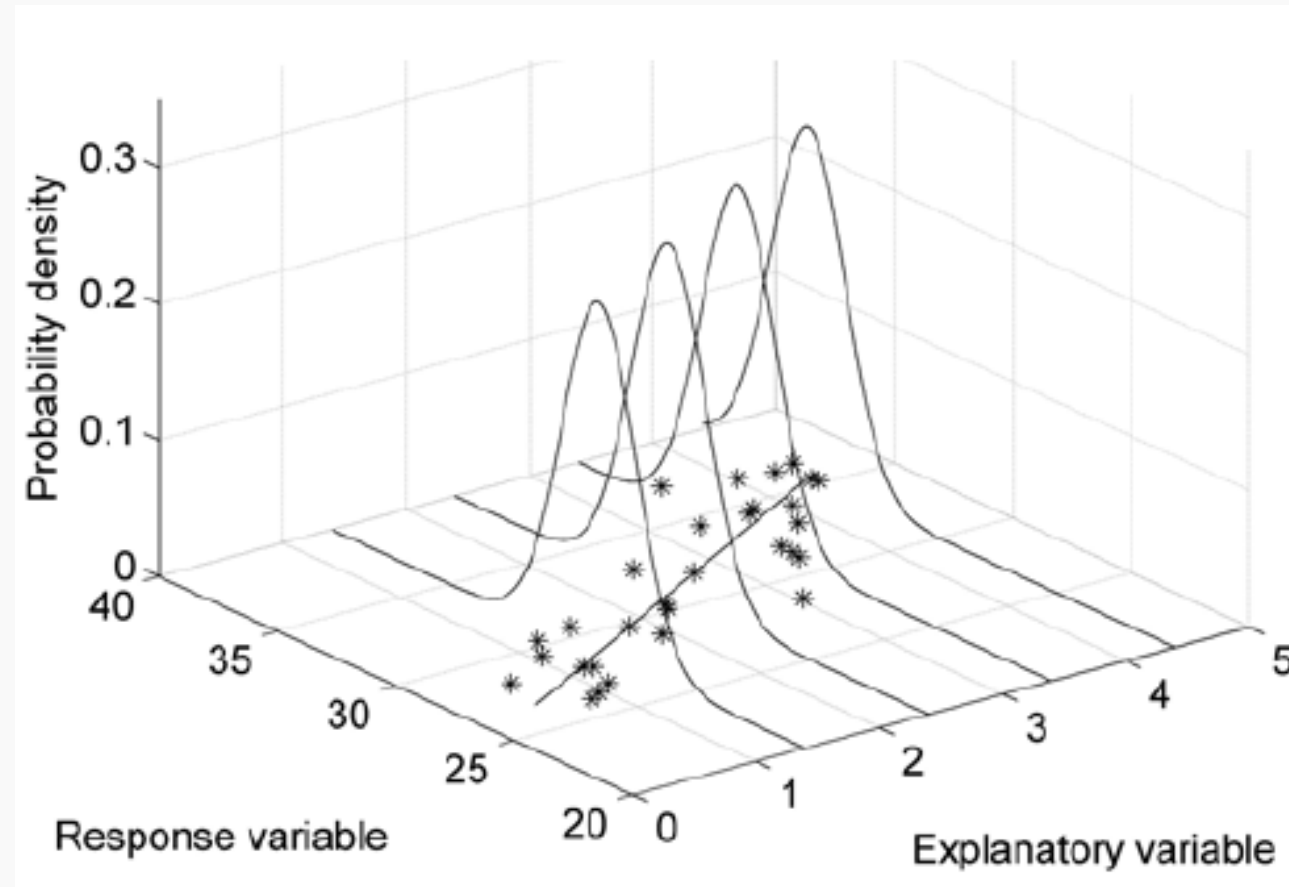
- Both Ridge and LASSO have a very natural interpretation from a Bayesian viewpoint.
- For this, we need to see our response as a multivariate normal distribution with varying means:

$$Y|\beta \sim N(X\beta, \sigma^2 I)$$



A different but useful perspective

$$Y|\beta \sim N(X\beta, \sigma^2 I)$$



Ridge and LASSO as MAP estimates

Consider $Y|\beta \sim N(X\beta, \sigma^2 I)$, and the MAP estimator:

$$\hat{\beta}_{MAP} = \operatorname{argmax}_{\beta} p(\beta|Y)$$

If the prior is $\beta \sim N(0, \sigma^2 / \lambda)$

Then $\beta_{MAP} = \beta_{Ridge}$

If the prior is $\beta \sim L(0, 2\sigma^2 / \lambda)$

Then $\beta_{MAP} = \beta_{LASSO}$

MAP: Maximum a posteriori estimation

Bayes Rule: Posterior

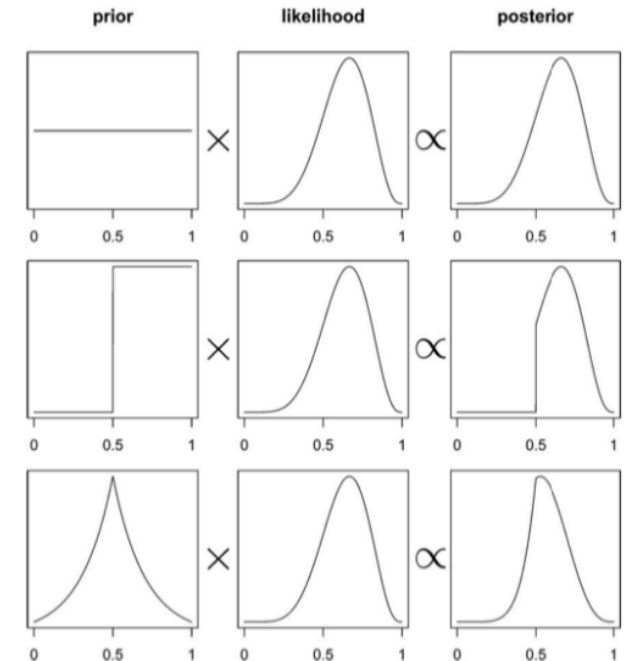
$$p(\beta|Y) = \frac{p(Y|\beta)p(\beta)}{p(Y)}$$
$$\sim p(Y|\beta)p(\beta)$$

Posterior

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}$$

$$\text{posterior} \propto \text{likelihood} \times \text{prior}$$

- evidence is just the normalization
- usually don't care about normalization (until model comparison), just pdf/pmf or samples



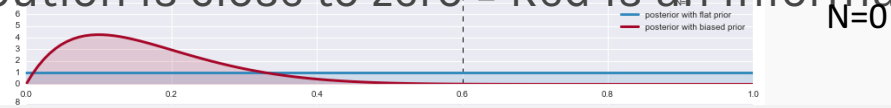
IACS AM 207

Maximum a posteriori estimation wants to maximize the posterior:

$\max(p(\beta|Y)) =$ the most likely β given /conditioned on our observed data

Posterior: priors and posteriors as we see more and more data

- Blue Player: assumes before seeing any data a uniform distribution = Blue is a non-informative Prior
- Red Player: assumes our distribution is close to zero = Red is an informative biased Prior



Proof of Bayesian interpretations

Bayes Rule:

$$p(\beta|Y) = \frac{p(Y|\beta)p(\beta)}{p(Y)} \propto p(Y|\beta)p(\beta)$$

We want to maximize the posterior, which is the same as maximizing the log because of its monotonicity:

$$\arg \max_{\beta} p(\beta|Y) = \arg \max_{\beta} [\log p(Y|\beta) + \log p(\beta)]$$

Remember that from the Bayesian perspective, we have:

$$\begin{aligned} p(Y|\beta) &\sim N(X\beta, \sigma^2 I) && \implies \log p(Y|\beta) \propto -(2\sigma^2)^{-1} \|X\beta - Y\|_2^2 \\ p(\beta) &\sim N(0, \tau^2 I) && \implies \log p(\beta) \propto -(2\tau^2)^{-1} \|\beta\|_2^2 \end{aligned}$$

Proof of Bayesian interpretations

Multiplying the entire optimization problem by -1, we turn a maximization into a minimization, and we have:

$$\arg \max_{\beta} p(\beta|Y) = \arg \min_{\beta} [(2\sigma^2)^{-1} \|X\beta - Y\|_2^2 + (2\tau^2)^{-1} \|\beta\|_2^2]$$

And setting $\tau^2 = \sigma^2/\lambda$, we can multiply the whole problem by $2\sigma^2$ without altering it and we get Ridge expression.

Similarly, if we set $\beta \sim L(0, b)$, we can get to:

$$\arg \max_{\beta} p(\beta|Y) = \arg \min_{\beta} [(2\sigma^2)^{-1} \|X\beta - Y\|_2^2 + b^{-1} \|\beta\|_1]$$

Which gives us LASSO by setting $b = 2\sigma^2/\lambda$.

Considerations on Bayesian Linear Regression

- The Bayesian perspective inspires other regression models. What if we change the prior on β ?
- We could, for example, put an asymmetric distribution if we have information that suggests that some β are likely to be positive.
- Bayesian analysis can go beyond finding point estimates on the betas. We can obtain full distributions.
- Regularizing with prior ends up yielding more information about the betas.
- The Bayesian formulation allows us to find the most likely lambda given our data.

Bayesian priors instead of cross-validation

- So far, we've assumed that we know λ . In the **frequentist** case, we get it through **cross-validation**.
- In the **Bayesian** perspective, there's an alternative **empirical Bayes** approach for picking hyperparameters: **Evidence Procedure/ (Sparse Bayesian learning) SBL**.
- Consists of maximizing the marginal likelihood resulting of integrating out the betas (finding the MLE of a new likelihood, where the parameter of interest is λ)
- This is also called Level-2 Maximum Likelihood.
- Principle practical advantage of Evidence Procedure: we can easily find optimal lambdas for **each parameter separately**.

Evidence Procedure: The math in a nutshell

Assume the following model:

$$p(Y|\beta) \sim N(X\beta, \sigma^2 I)$$

$$p(\beta) \sim N(0, A^{-1})$$

$$A^{-1} = \tau^2 I \quad \tau^2 = \left[\frac{\sigma^2}{\lambda_1}, \frac{\sigma^2}{\lambda_2}, \dots, \frac{\sigma^2}{\lambda_p} \right]$$

The marginal likelihood can be computed as follows:

$$p(Y|\tau^2) = \int N(Y; X\beta, \sigma^2 I) N(\beta; 0, A^{-1}) d\beta$$

$$= N(Y; 0, \sigma^2 I + XA^{-1}X^T)$$

$$= (2\pi)^{-\frac{N}{2}} |C_\tau|^{-\frac{1}{2}} \exp\left(-\frac{1}{2} Y^T C_\tau^{-1} Y\right)$$

$$C_\tau = \frac{1}{\sigma^2} I + XA^{-1}X^T$$

Evidence Procedure: The math in a nutshell

We want the tau that maximizes this likelihood. We minimize the negative log likelihood:

$$\tau_{EB}^2 = \arg \min_{\tau} \log |C_{\tau}| + Y^T C_{\tau}^{-1} Y$$


- And we can obtain our optimal regularization parameter from here.¹
- Note: we worked through the problem with **different lambdas for every beta**! If lambdas all equal: back to classic Ridge regression.

¹ There is an easy formula to automatically obtain the betas as well, available in chapter 13, p. 464 of Murphy's "Machine Learning – A Probabilistic Perspective".

THANK YOU!

Practical side: how to check for multicollinearity?

- Check if at least one eigenvalue of Gram Matrix ($X^T X$) is close to 0.
- Check for large condition numbers (κ) in $X^T X$.
- Condition number > 30 usually indicates multicollinearity.
- Check for high variance inflation factors (VIFs). $VIF > 10$ usually indicates multicollinearity.

$$VIF = \frac{1}{1 - R_i^2}$$


This R_i^2 is the coefficient of determination obtained when regressing X_i with all other X as predictors.

Augmented problem - Elastic Net

We can actually prove that EN is a generalized LASSO with augmented data. Construct the augmented problem:

$$Y^* = \begin{pmatrix} Y \\ 0 \end{pmatrix} \in \mathbb{R}^{n+p}$$
$$X^* = (1 + \lambda_2)^{-\frac{1}{2}} \begin{pmatrix} X \\ \sqrt{\lambda_1} I \end{pmatrix} \in \mathbb{R}^{(n+p) \times p}$$

and define:

$$\gamma = \frac{\lambda_1}{\sqrt{(1 + \lambda_2)}}$$
$$\beta^* = \sqrt{(1 + \lambda_2)}\beta$$

Augmented problem - Elastic Net

Then, the elastic net problem can be written as:

$$\hat{\beta}^* = \arg \min_{\beta^* \in \mathbb{R}^p} \|X^* \beta^* - Y^*\|_2^2 + \gamma \|\beta^*\|_1$$



LASSO
problem!

Where $\hat{\beta}_{EN} = (1 + \lambda_2)^{-\frac{1}{2}} \hat{\beta}^*$

- As we can see, the EN problem can be reformulated as a **LASSO problem on augmented data**.
- Note that since sample size of X is $n + p > p$, the elastic net estimator can actually select **all p predictors**.
- $\hat{\beta}_E$ is a shrunk version of $\hat{\beta}^*$: EN does both **variable shrinking** and **variable selection**.