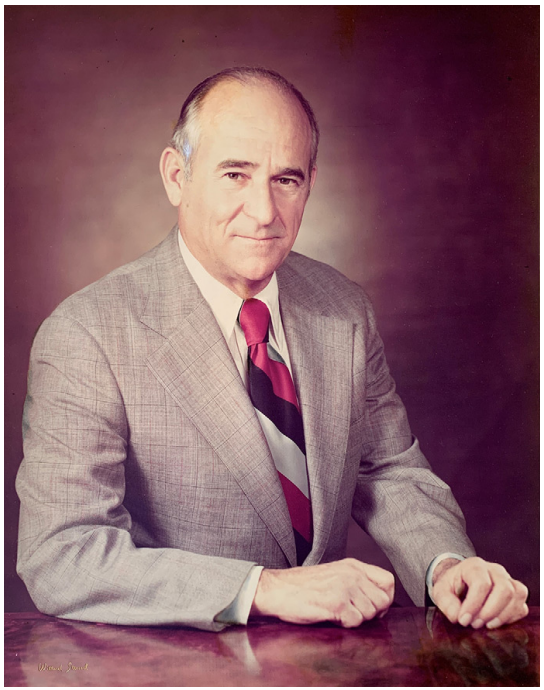
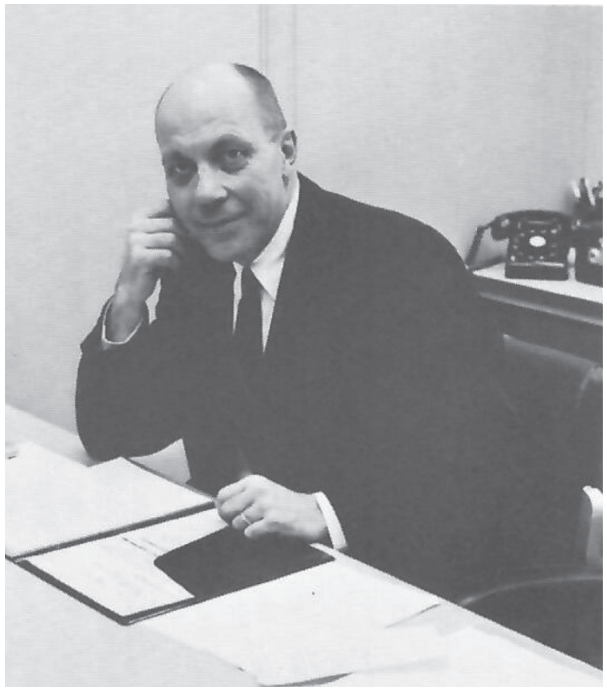
Ridge Regularization: An Essential Concept in Data Science

Trevor Hastie
Department of Statistics
Department of Biomedical Data Science
Stanford University



50 Years Ago





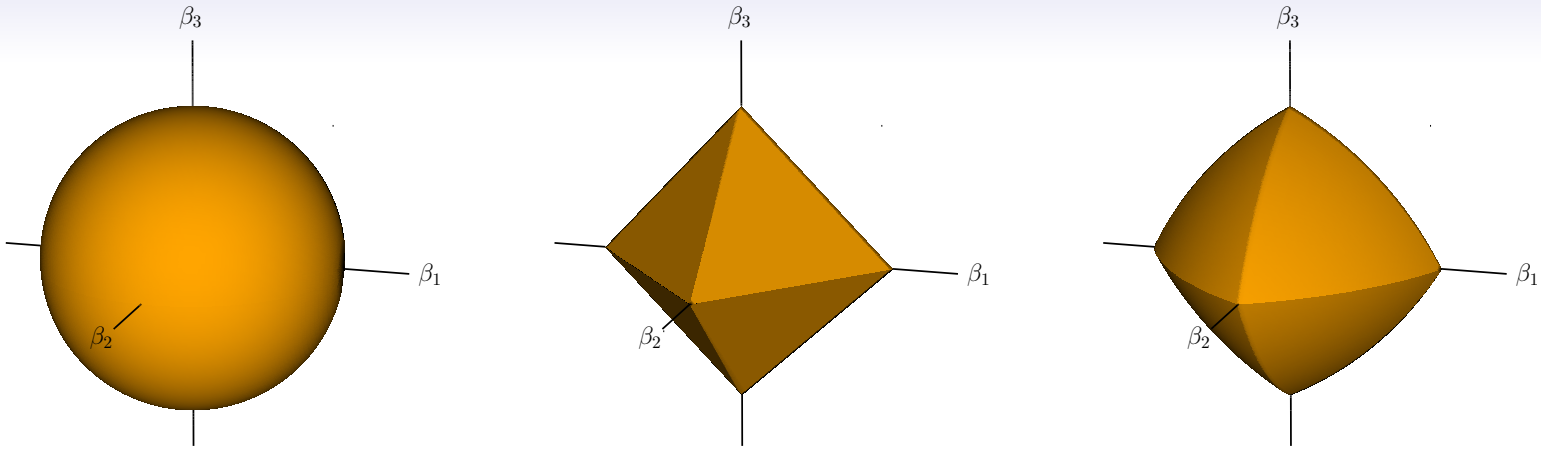
Hoerl, A.E. and Kennard, R.W. (1970a) "Ridge Regression: Biased Estimation for Nonorthogonal Problems " *Technometrics*, 12, 55–67.

Hoerl, A.E. and Kennard, R.W. (1970b) "Ridge Regression: Applications to Nonorthogonal Problems " *Technometrics*, 12, 69–82.

Outline

- Ridge, lasso and elastic net
- SVD and ridge computations
- GLMs for wide data and the kernel trick
- Limiting ridge and support-vector machines
- Dropout and deep learning
- Data augmentation
- Matrix completion
- LOO cross-validation
- Double descent and the bias-variance trade-off

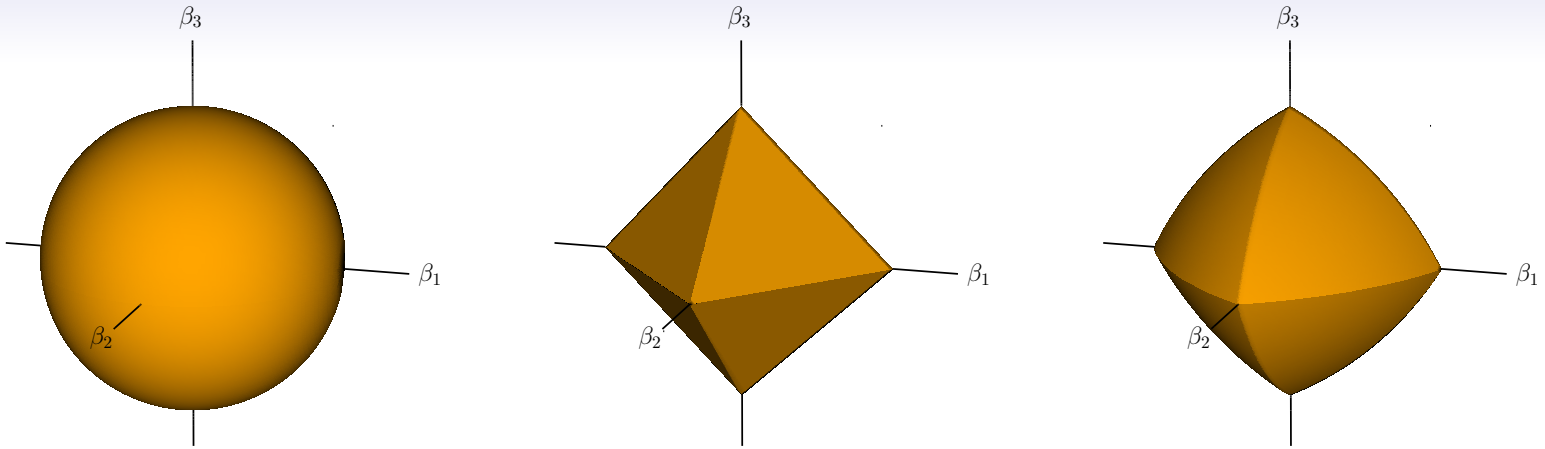
TH (2020) "Ridge Regularization: An Essential Concept in Data Science", *Technometrics*, 62:4, 426-433.



$$\hat{\beta}_\lambda = (X^\top X + \lambda I)^{-1} X^\top y$$

Ridge resolves difficulties with collinear and wide data.

Ridge $\quad \underset{\beta}{\text{minimize}} \, \|y - X\beta\|_2^2 + \lambda \|\beta\|_2^2$

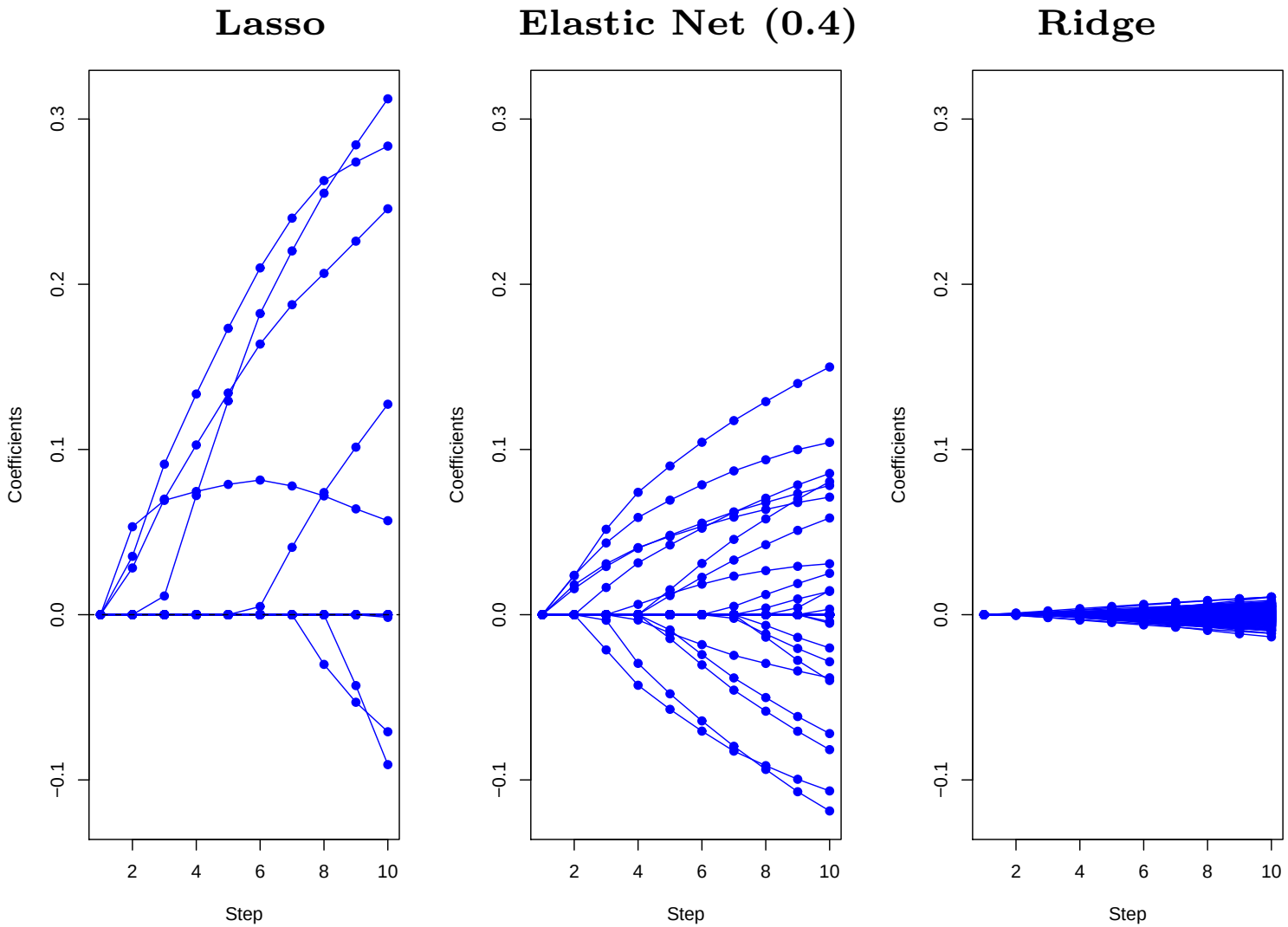


$$\hat{\beta}_\lambda = (X^\top X + \lambda I)^{-1} X^\top y$$

Ridge resolves difficulties with collinear and wide data.

Ridge	$\underset{\beta}{\text{minimize}} \, \|y - X\beta\|_2^2 + \lambda \|\beta\|_2^2$
Lasso	$\underset{\beta}{\text{minimize}} \, \|y - X\beta\|_2^2 + \lambda \|\beta\|_1$
Elastic Net	$\underset{\beta}{\text{minimize}} \, \|y - X\beta\|_2^2 + \lambda \left[(1-\alpha) \|\beta\|_2^2 + \alpha \|\beta\|_1 \right]$

GLMNET on Leukemia Data



Logistic regression, N=72, p=3571, first 10 steps shown.

Ridge and the SVD

Let $X = UDV^\top$ be the SVD of $X_{n\times p}$, with $U_{n\times n}$ orthogonal, $V_{p\times p}$ orthogonal, and $D_{n\times p}$ diagonal with $m = \min(n, p)$ elements $d_1 \geq d_2 \geq \cdots \geq d_m \geq 0$.

$$\begin{aligned}
\hat{\beta}_\lambda &= (X^\top X + \lambda I)^{-1} X^\top y \\
&= V(D^\top D + \lambda I)^{-1} D^\top U^\top y \\
&= \textstyle\sum_{d_j>0} v_j \, \frac{d_j}{d_j^2+\lambda} \, \langle u_j, y\rangle
\end{aligned}$$

- Entire ridge path with one SVD.

Ridge and the SVD

Let $X = UDV^\top$ be the SVD of $X_{n\times p}$, with $U_{n\times n}$ orthogonal, $V_{p\times p}$ orthogonal, and $D_{n\times p}$ diagonal with $m = \min(n, p)$ elements $d_1 \geq d_2 \geq \cdots \geq d_m \geq 0$.

$$\begin{aligned}
\hat{\beta}_\lambda &= (X^\top X + \lambda I)^{-1} X^\top y \\
&= V(D^\top D + \lambda I)^{-1} D^\top U^\top y \\
&= \textstyle\sum_{d_j>0} v_j \, \frac{d_j}{d_j^2+\lambda} \, \langle u_j, y\rangle
\end{aligned}$$

- Entire ridge path with one SVD.
- $\hat{\beta}_\lambda$ lies in *row space* of X.

Ridge and the SVD

Let $X = UDV^\top$ be the SVD of $X_{n\times p}$, with $U_{n\times n}$ orthogonal, $V_{p\times p}$ orthogonal, and $D_{n\times p}$ diagonal with $m = \min(n, p)$ elements $d_1 \geq d_2 \geq \cdots \geq d_m \geq 0$.

$$\begin{aligned}
\hat{\beta}_\lambda &= (X^\top X + \lambda I)^{-1} X^\top y \\
&= V(D^\top D + \lambda I)^{-1} D^\top U^\top y \\
&= \textstyle\sum_{d_j>0} v_j \, \frac{d_j}{d_j^2+\lambda} \, \langle u_j, y\rangle
\end{aligned}$$

- Entire ridge path with one SVD.
- $\hat{\beta}_\lambda$ lies in *row space* of X.
- Computations are $O(\min(np^2,\ n^2p))$.

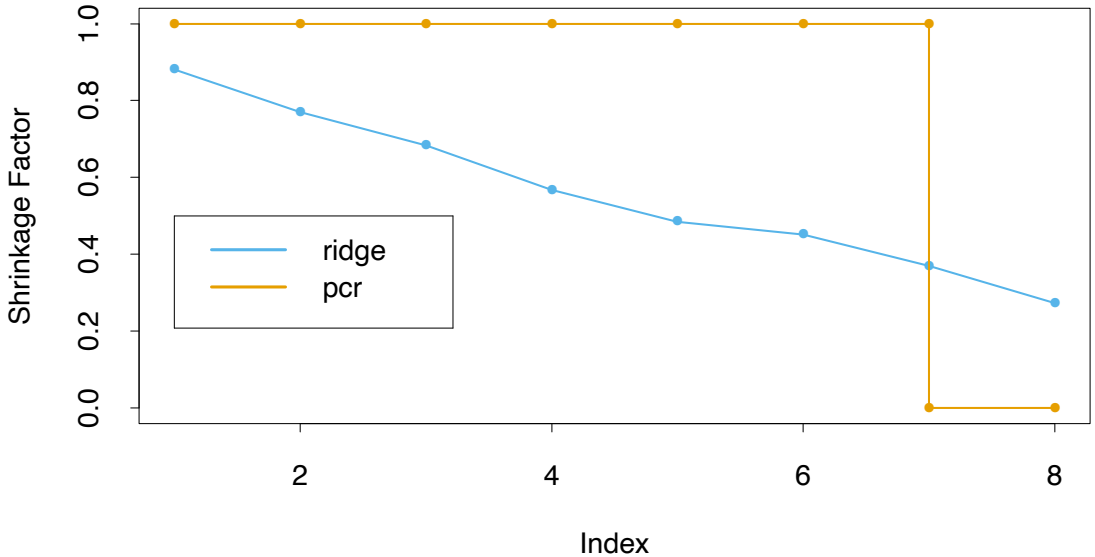
Ridge and the SVD

Let $X = UDV^\top$ be the SVD of $X_{n\times p}$, with $U_{n\times n}$ orthogonal, $V_{p\times p}$ orthogonal, and $D_{n\times p}$ diagonal with $m = \min(n, p)$ elements $d_1 \geq d_2 \geq \cdots \geq d_m \geq 0$.

$$\begin{aligned}
\hat{\beta}_\lambda &= (X^\top X + \lambda I)^{-1} X^\top y \\
&= V(D^\top D + \lambda I)^{-1} D^\top U^\top y \\
&= \textstyle\sum_{d_j>0} v_j \, \frac{d_j}{d_j^2+\lambda} \, \langle u_j, y\rangle
\end{aligned}$$

- Entire ridge path with one SVD.
- $\hat{\beta}_\lambda$ lies in *row space* of X.
- Computations are $O(\min(np^2,\ n^2p))$.
- For *wide* data, $\lambda = 0$ gives *minimum-norm* zero-residual solution: $\hat{\beta}_{mn} = \arg\min \|\beta\|_2$ s.t. $y = X\beta$.

Ridge versus Principal Component Regression

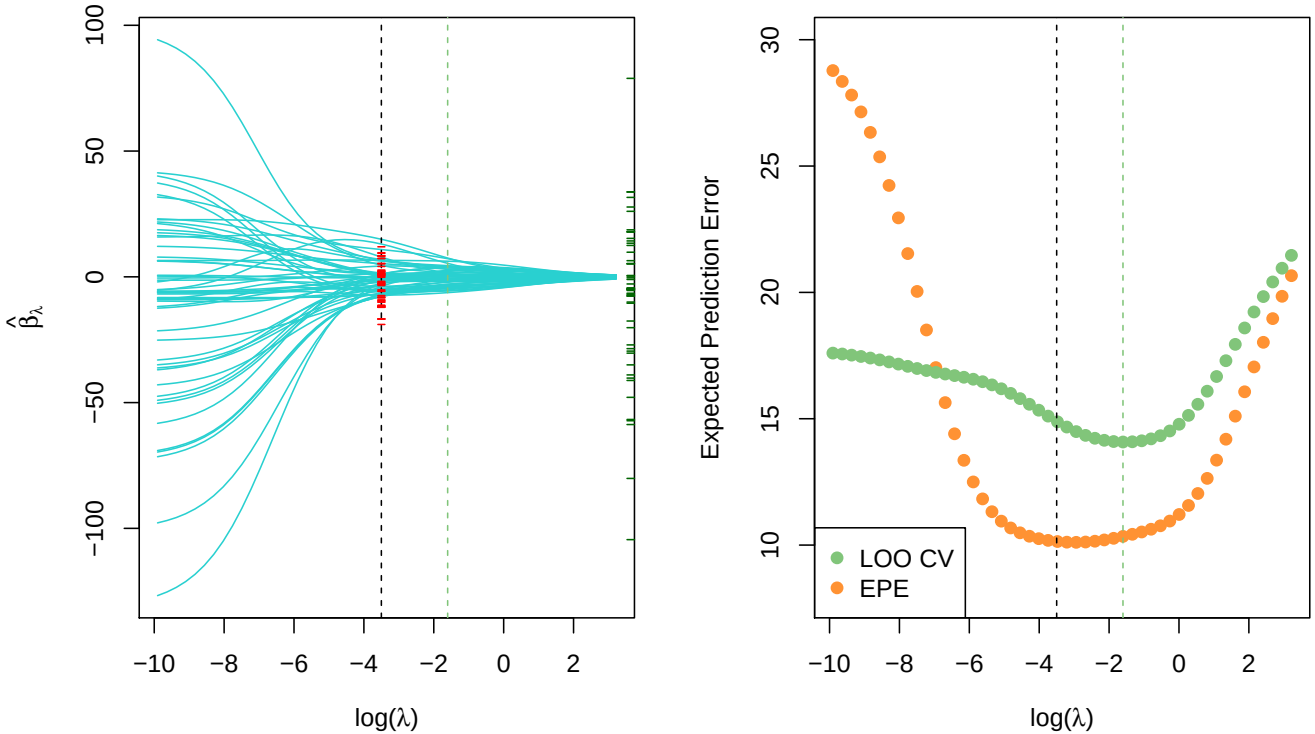


$$\text{Ridge: } X\hat{\beta}_\lambda = \sum_{d_j>0} u_j \frac{d_j^2}{d_j^2+\lambda}\langle u_j, y\rangle$$

$$\text{PCR: } X\hat{\beta}^{[k]} = \sum_{j\le k} u_j \langle u_j, y\rangle$$

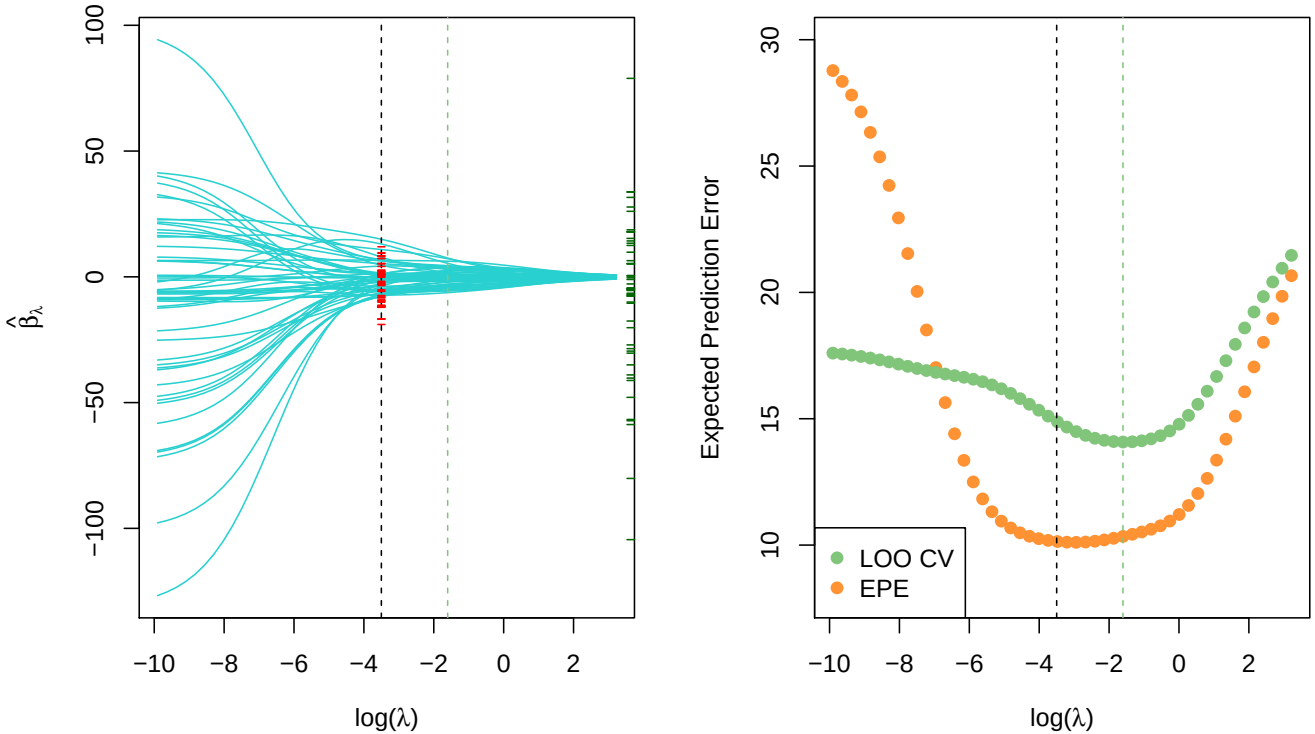
- Ridge uses all the principal components, and shrinks their coefficients.
- PCR projects onto a subset of leading components.

Ridge and the Bias-Variance Trade-off



- Simulation n=100, p=54, SNR = 3.
- Red bars (at minimum EPE line) are true coefficients.

Ridge and the Bias-Variance Trade-off



- Simulation n=100, p=54, SNR = 3.
- Red bars (at minimum EPE line) are true coefficients.
- Green ticks (right axis) are James-Stein

$$\hat{\beta}_{JS} = \left[1 - \frac{(p-2)\sigma^2}{\hat{\beta}^\top X^\top X \hat{\beta}}\right] \hat{\beta}$$

Ridge and Wide GLMs

$$\underset{\beta}{\text{maximize}} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, x_i^\top \beta) - \lambda \|\beta\|_2^2$$

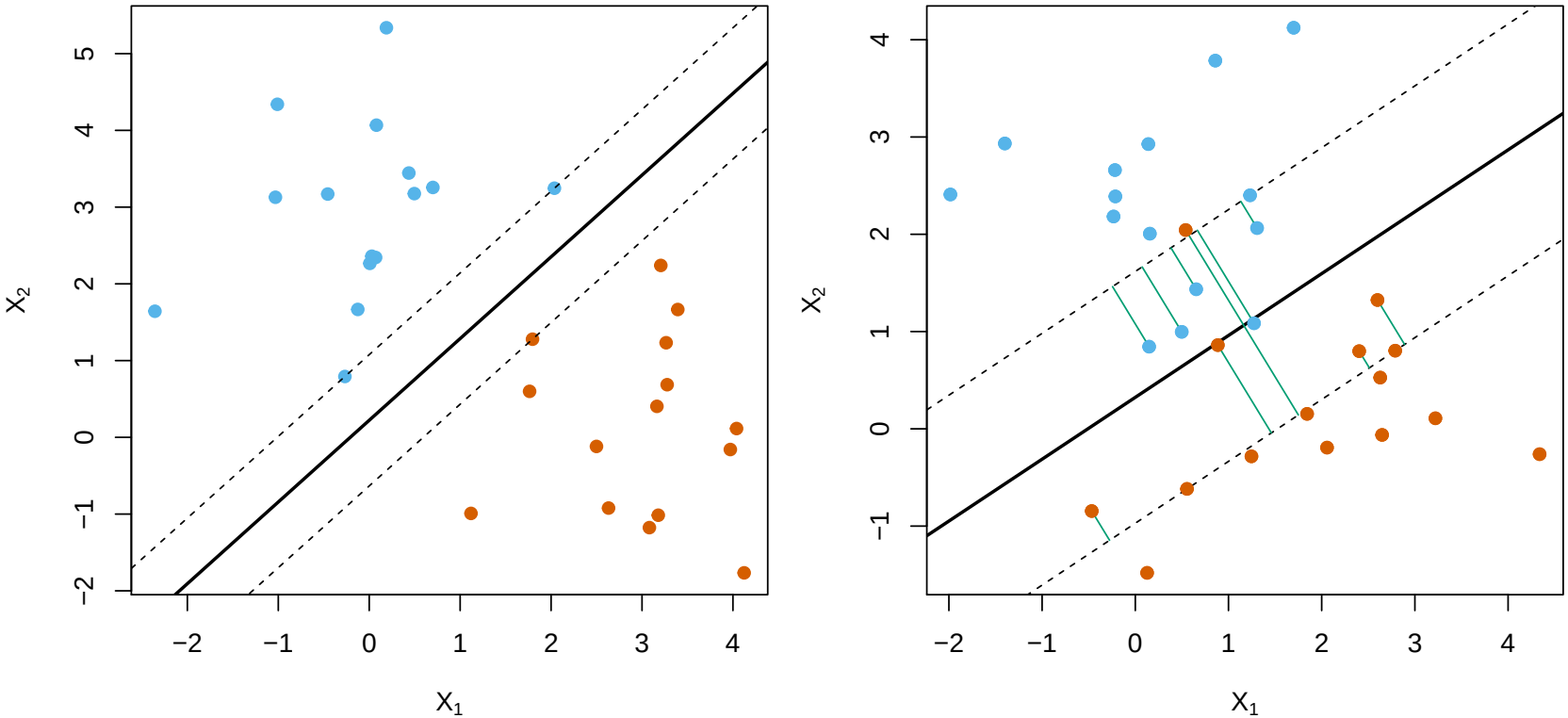
Suppose $X_{n\times p}$ with $p \gg n$ (wide), and let $X = RV^\top$ with $R_{n\times n}$ nonsingular and $V_{p\times n}$ with orthonormal columns. Solve instead

$$\underset{\theta}{\text{maximize}} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, r_i^\top \theta) - \lambda \|\theta\|_2^2$$

Then $\hat{\beta}_\lambda = V\hat{\theta}_\lambda$ — again in row-space of X.

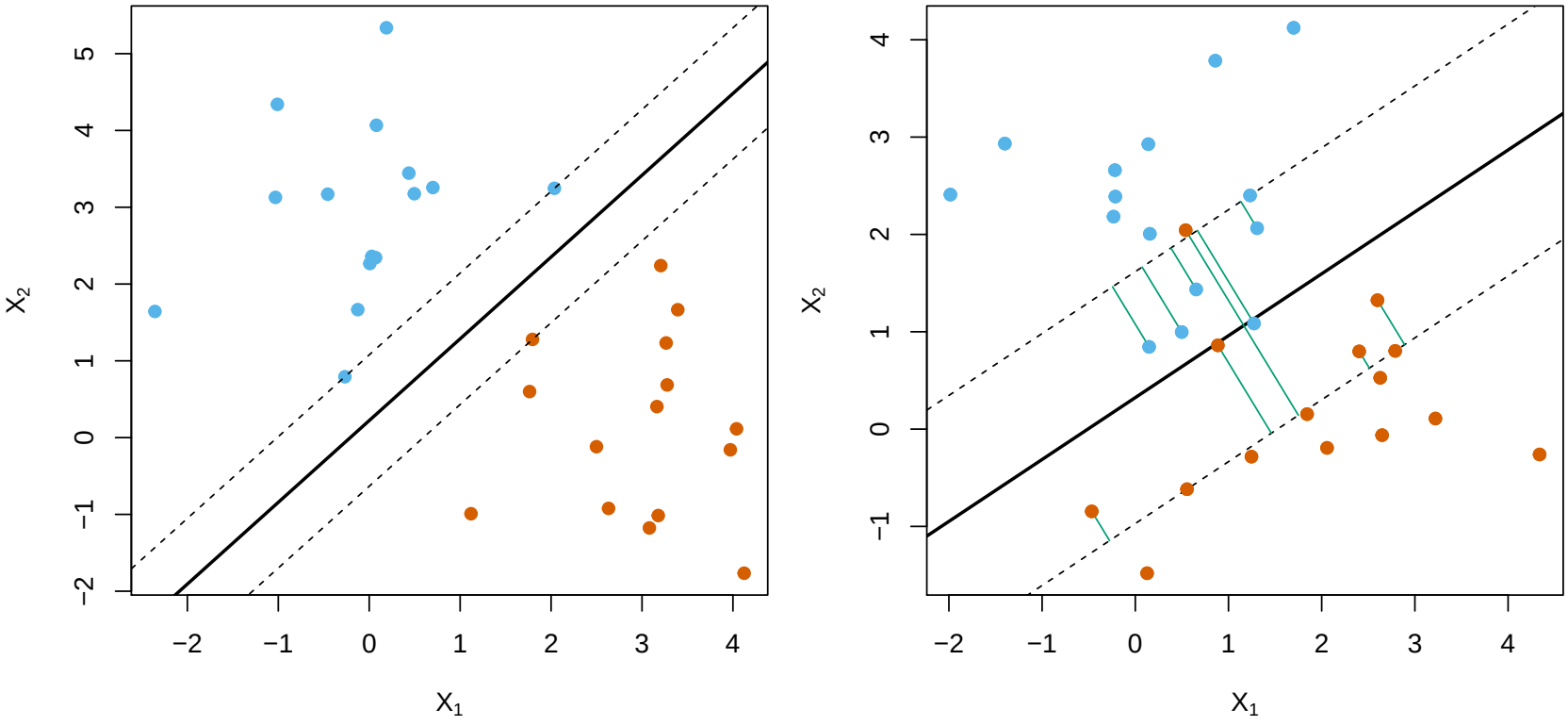
- True for any linear model with ridge regularization.
- Can use CV etc on R to select λ.
- n-dimensional estimation rather than p-dimensional.

Ridge and Support Vector Machines



Fact: $\frac{1}{\|\beta\|_2}(\beta_0 + x^\top \beta)$ is signed distance from x to decision boundary $\{x : \beta_0 + x^\top \beta = 0\}$.

Ridge and Support Vector Machines

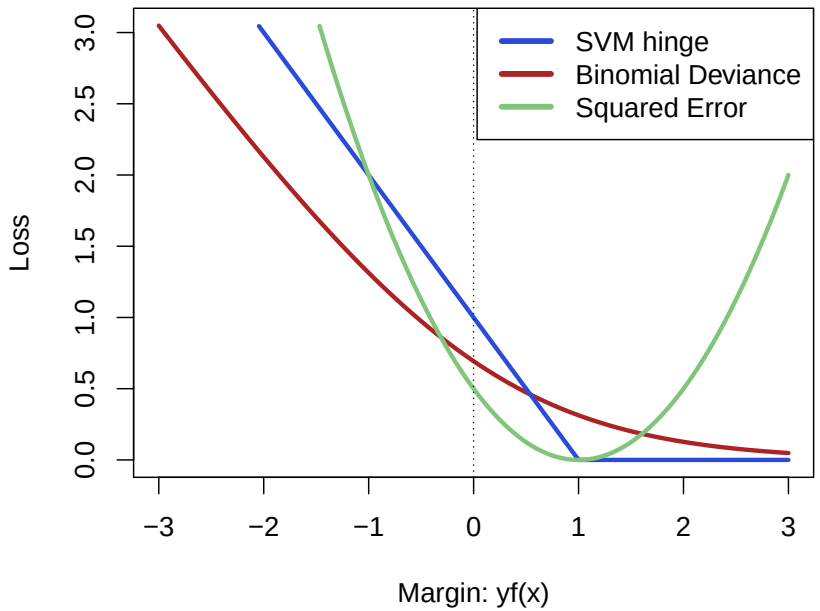


Fact: $\frac{1}{\|\beta\|_2}(\beta_0 + x^\top \beta)$ is signed distance from x to decision boundary $\{x : \beta_0 + x^\top \beta = 0\}$.

$$\underset{\beta_0,\, \beta}{\text{minimize}} \|\beta\|_2 \quad \text{subject to}$$

$$y_i(\beta_0 + x_i^\top \beta) \geq 1 - \epsilon_i, \quad \epsilon_i \geq 0 \; \forall \; i, \quad \sum_{i=1}^{n} \epsilon_i \leq B.$$

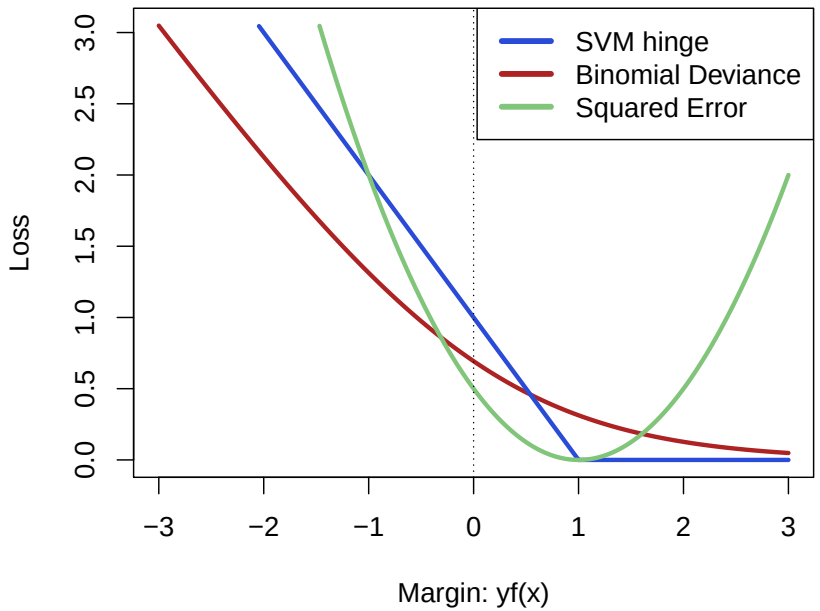
SVM as Loss + Penalty



- $\min_{\beta_0,\,\beta} \sum_{i=1}^{n} [1-y_i(\beta_0+x_i^\top \beta)]_+ + \lambda\|\beta\|_2^2$
- $\min_{\beta_0,\,\beta} \sum_{i=1}^{n} \log[1+e^{-y_i(\beta_0+x_i^\top \beta)}] + \lambda\|\beta\|_2^2$
- $\min_{\beta_0,\,\beta} \sum_{i=1}^{n} \frac{1}{2}[1-y_i(\beta_0+x_i^\top \beta)]^2 + \lambda\|\beta\|_2^2$

$y_i \in \{-1, +1\}$. Margin $y \cdot f(x) > 0$ means correctly classified.

SVM as Loss + Penalty

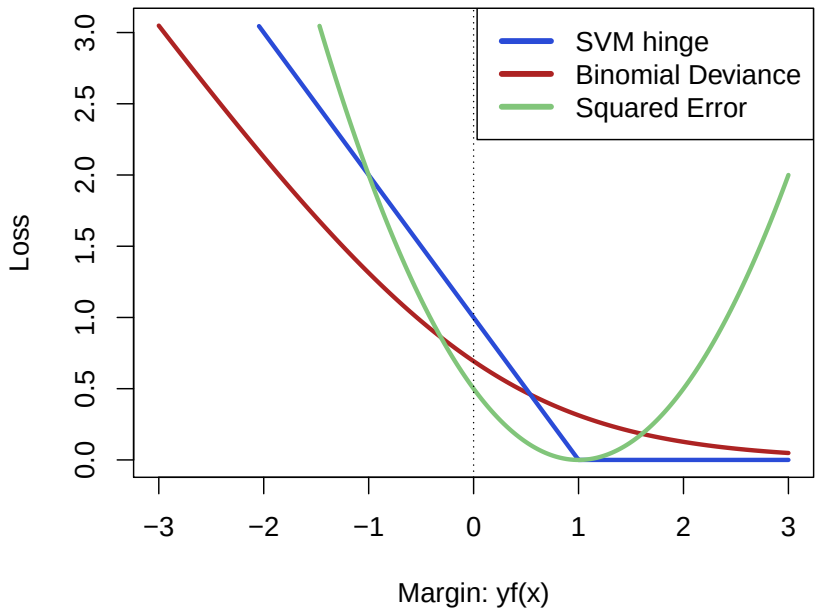


- $\min_{\beta_0,\,\beta} \sum_{i=1}^{n} [1-y_i(\beta_0+x_i^\top \beta)]_+ + \lambda\|\beta\|_2^2$
- $\min_{\beta_0,\,\beta} \sum_{i=1}^{n} \log[1+e^{-y_i(\beta_0+x_i^\top \beta)}] + \lambda\|\beta\|_2^2$
- $\min_{\beta_0,\,\beta} \sum_{i=1}^{n} \frac{1}{2}[1-y_i(\beta_0+x_i^\top \beta)]^2 + \lambda\|\beta\|_2^2$

$y_i \in \{-1, +1\}$. Margin $y \cdot f(x) > 0$ means correctly classified.
If data are *separable*, $\lim \lambda \downarrow 0$ solution leads to:

- maximal margin classifier for SVM;

SVM as Loss + Penalty

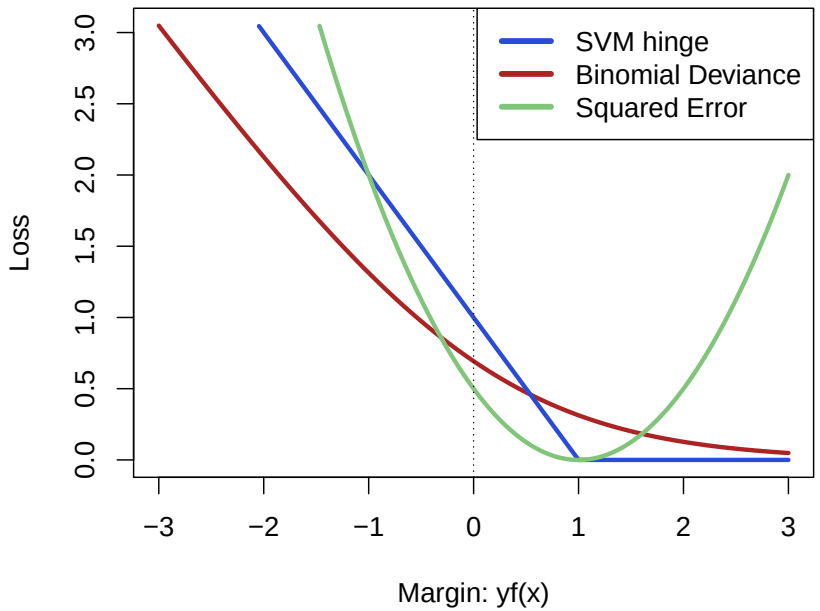


- $\min_{\beta_0,\,\beta} \sum_{i=1}^{n} [1-y_i(\beta_0+x_i^\top \beta)]_+ + \lambda\|\beta\|_2^2$
- $\min_{\beta_0,\,\beta} \sum_{i=1}^{n} \log[1+e^{-y_i(\beta_0+x_i^\top \beta)}] + \lambda\|\beta\|_2^2$
- $\min_{\beta_0,\,\beta} \sum_{i=1}^{n} \frac{1}{2}[1-y_i(\beta_0+x_i^\top \beta)]^2 + \lambda\|\beta\|_2^2$

$y_i \in \{-1, +1\}$. Margin $y \cdot f(x) > 0$ means correctly classified.
If data are *separable*, $\lim \lambda \downarrow 0$ solution leads to:

- maximal margin classifier for SVM;
- same for binomial, if β renormalized;

SVM as Loss + Penalty

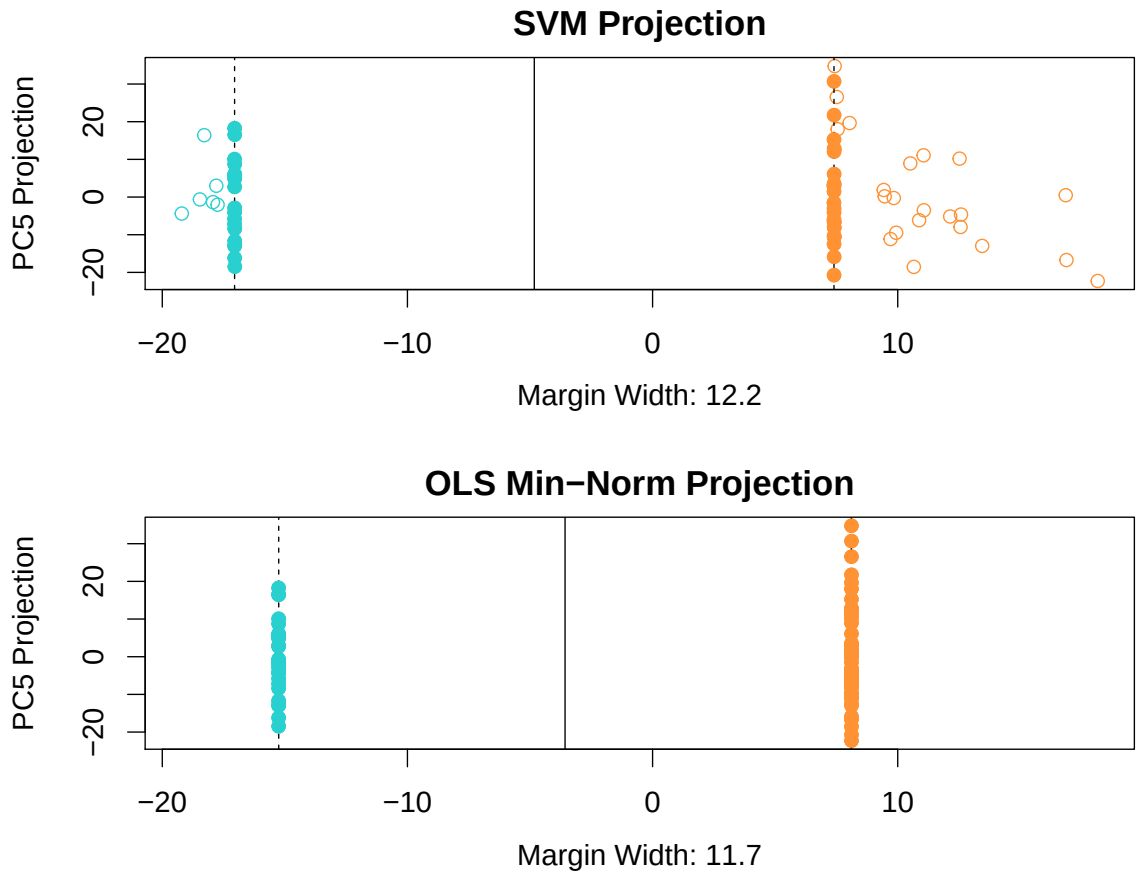


- $\min_{\beta_0,\,\beta} \sum_{i=1}^{n} [1-y_i(\beta_0+x_i^\top \beta)]_+ + \lambda\|\beta\|_2^2$
- $\min_{\beta_0,\,\beta} \sum_{i=1}^{n} \log[1+e^{-y_i(\beta_0+x_i^\top \beta)}] + \lambda\|\beta\|_2^2$
- $\min_{\beta_0,\,\beta} \sum_{i=1}^{n} \frac{1}{2}[1-y_i(\beta_0+x_i^\top \beta)]^2 + \lambda\|\beta\|_2^2$

$y_i \in \{-1, +1\}$. Margin $y \cdot f(x) > 0$ means correctly classified.
If data are *separable*, $\lim \lambda \downarrow 0$ solution leads to:

- maximal margin classifier for SVM;
- same for binomial, if β renormalized;
- *data piling* direction (Ahn and Marron 2010), for least squares.

Maximal Margin Classifier for Leukemia Data



- Both SVM and LS are minimum-norm solutions, which translates to maximum margin classifiers.
- LS has additional implicit constraint that all points fall on the margin (zero residual), so margin not as wide.

More on SVMs and Kernel Trick

- Computations for linear SVMs use $n \times n$ *gram* matrix $XX^\top$ — equivalent to using R as before ($X = RV^\top$).
- More general SVMs expand features into higher dimensional space and solve

$$\min_{\beta_0,\,\beta} \sum_{i=1}^{n} [1 - y_i(\beta_0 + h(x_i)^\top \beta)]_+ + \lambda \|\beta\|_2^2$$

  with $\langle h(x_i), h(x_{i'}\rangle$ computed implicitly by *kernel* $K(x_i, x_{i'})$.
- Nevertheless limiting $\lambda \downarrow 0$ solution is often preferred, leading to even wider maximal margin separators.
- Early move to *overfit* solutions via minimum norm.

Kernel Trick and Linear Regression

With $X_{n\times p}$ and $y_{n\times 1}$

$$\underset{\beta}{\text{minimize}}\, \|y - X\beta\|_2^2 + \lambda\|\beta\|_2^2$$

- The vector of fitted values is

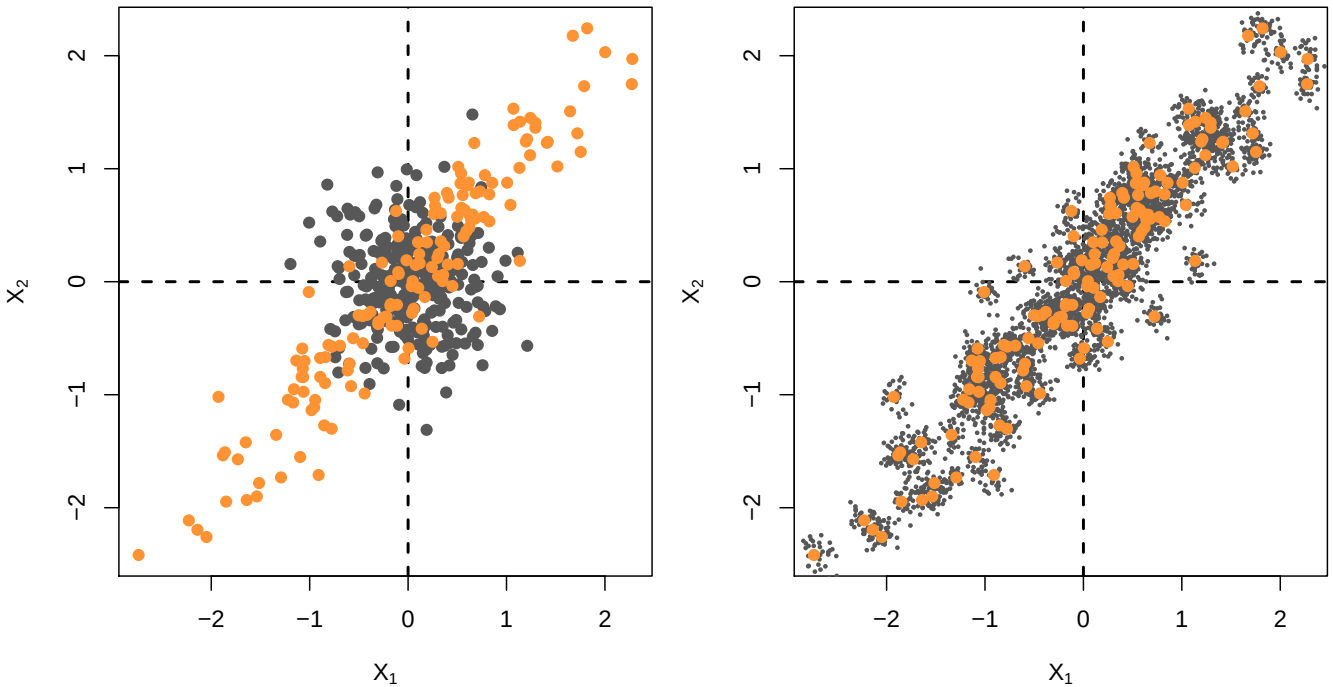
$$\hat{y}_\lambda = X(X^\top X + \lambda I_p)^{-1}X^\top y$$

- The vector of fitted values can *also* be written

$$\hat{y}_\lambda = XX^\top(XX^\top + \lambda I_n)^{-1}y$$

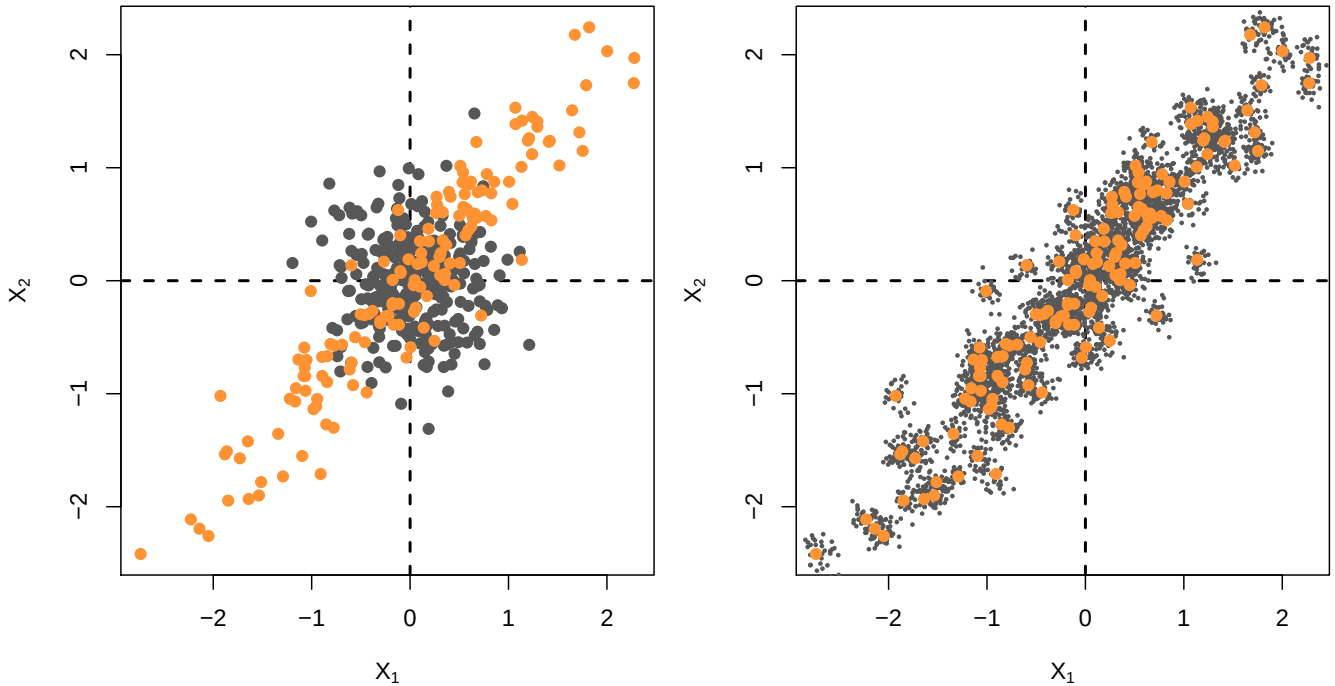
- If $n \gg p$ we use the former, if $n \ll p$ we use the latter.

Ridge and Data Augmentation



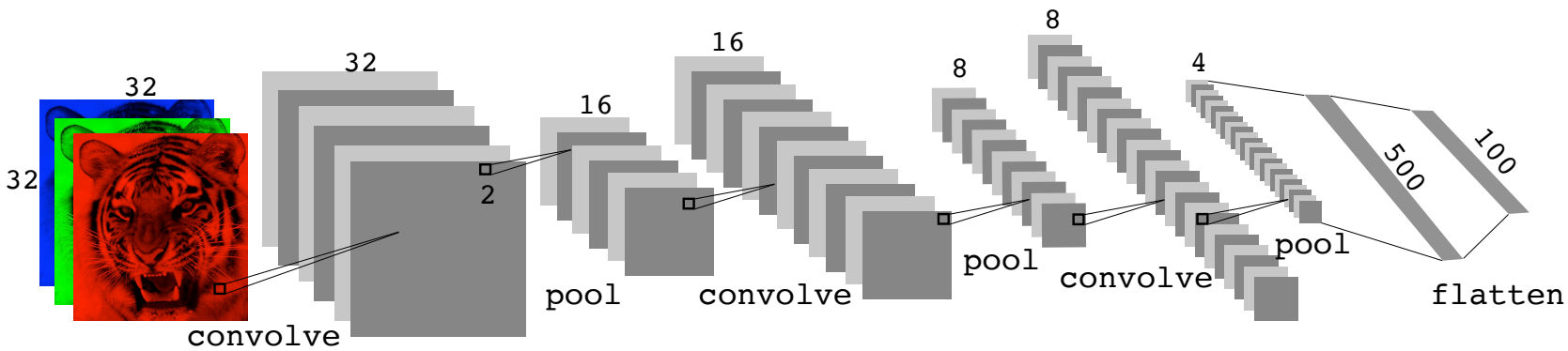
- Augment rows of X with m draws from $N(0, \frac{\lambda}{m} I_p)$, and pad y with zeros. Then $\tilde{X}^\top \tilde{X} \approx X^\top X + \lambda I$ and $\tilde{X}^\top \tilde{y} \approx X^\top y$.

Ridge and Data Augmentation



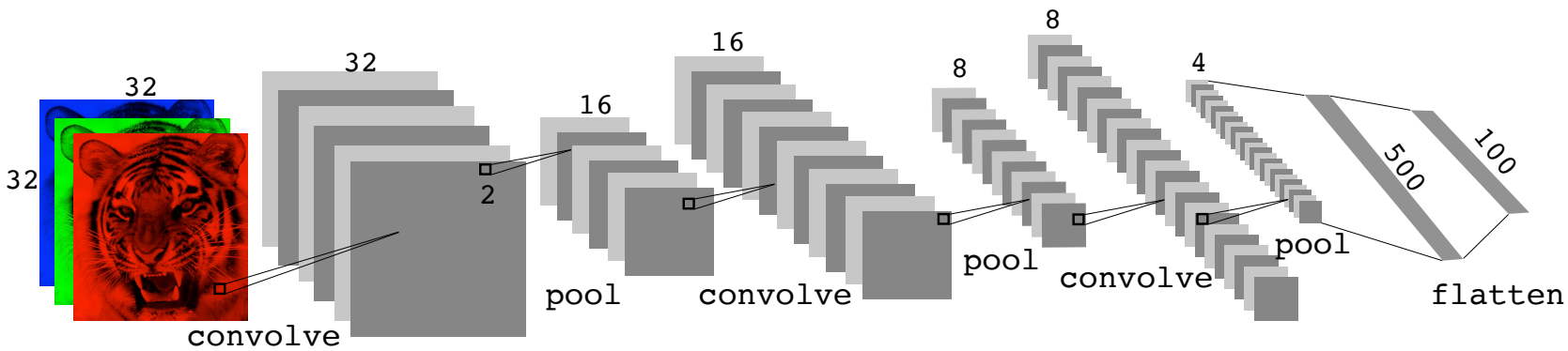
- Augment rows of X with m draws from $N(0, \frac{\lambda}{m} I_p)$, and pad y with zeros. Then $\tilde{X}^\top \tilde{X} \approx X^\top X + \lambda I$ and $\tilde{X}^\top \tilde{y} \approx X^\top y$.
- Make m copies of each (x_i, y_i) and add $\epsilon_{ij} \sim N(0, \frac{\lambda}{n} I_p)$ noise to each copy of $x_i : x_{ij} = x_i + \epsilon_{ij}$. Then $\frac{1}{m}\tilde{X}^\top \tilde{X} \approx X^\top X + \lambda I$ and $\frac{1}{m}\tilde{X}^\top \tilde{y} \approx X^\top y$.

Data Augmentation on the Fly



Deep CNNs use stochastic gradient descent to train classifiers.

Data Augmentation on the Fly



Deep CNNs use stochastic gradient descent to train classifiers.

Natural transformations are made of each image when it is sampled by SGD, thus ultimately making a cloud of images around each original training image.



Taken from new "Deep Learning" chapter in forthcoming 2nd edition of *Introduction to Statistical Learning* (May 2021)

Ridge and Dropout Learning



- With neural networks, at each SGD update, randomly set units to zero with probability ϕ.
- As in ridge, the other units *stand in* for them, and their weights are drawn closer together.
- In simple scenarios like linear regression, formally equivalent to ridge (Wager, Wang and Liang, 2013).
- Similar to earlier ideas of randomly omitting variables when growing tree ensembles and random forests (Amit and Geman, 1997; Breiman 2001)

Ridge + Ridge = Lasso

- With $X_{n\times p}$, solve *convex* problem:

$$\underset{M_{n\times p}}{\text{minimize}}\, \|X - M\|_F^2 + \lambda \|M\|_*$$

Ridge + Ridge = Lasso

- With $X_{n\times p}$, solve *convex* problem:

$$\underset{M_{n\times p}}{\text{minimize}}\, \|X - M\|_F^2 + \lambda \|M\|_*$$

Let SVD $X = UDV^\top$, solution is $\widehat{M} = US_\lambda(D)V^\top$, with $S_\lambda(D)_{ii} = (D_{ii} - \lambda)_+$. *Lasso for matrices! (Fazel, 2002)*

Ridge + Ridge = Lasso

- With $X_{n\times p}$, solve *convex* problem:

$$\underset{M_{n\times p}}{\text{minimize}}\, \|X - M\|_F^2 + \lambda \|M\|_*$$

  Let SVD $X = UDV^\top$, solution is $\widehat{M} = US_\lambda(D)V^\top$, with $S_\lambda(D)_{ii} = (D_{ii} - \lambda)_+$. *Lasso for matrices! (Fazel, 2002)*

- If $\text{rank}(\widehat{M}) \leq q$, solve *biconvex* problem:

$$\underset{A_{n\times q}, B_{p\times q}}{\text{minimize}}\, \|X - AB^\top\|_F^2 + \frac{\lambda}{2}\|A\|_F^2 + \frac{\lambda}{2}\|B\|_F^2$$

  $\widehat{A}\widehat{B}^\top = \widehat{M}$ (Srebro, Renni, Jaakkola 2005).

Ridge + Ridge = Lasso

- With $X_{n\times p}$, solve *convex* problem:

$$\underset{M_{n\times p}}{\text{minimize}}\, \|X - M\|_F^2 + \lambda \|M\|_*$$

  Let SVD $X = UDV^\top$, solution is $\widehat{M} = US_\lambda(D)V^\top$, with $S_\lambda(D)_{ii} = (D_{ii} - \lambda)_+$. *Lasso for matrices! (Fazel, 2002)*

- If $\text{rank}(\widehat{M}) \leq q$, solve *biconvex* problem:

$$\underset{A_{n\times q}, B_{p\times q}}{\text{minimize}}\, \|X - AB^\top\|_F^2 + \frac{\lambda}{2}\|A\|_F^2 + \frac{\lambda}{2}\|B\|_F^2$$

  $\widehat{A}\widehat{B}^\top = \widehat{M}$ (Srebro, Renni, Jaakkola 2005).

- Extremely useful for *matrix completion* problems, where many entries of X are missing.

Double Descent

- With neural networks, more hidden units better than fewer.
- Likewise more hidden layers.
- Often running stochastic gradient descent till zero training error gives good out-of-sample error.
- Often increasing the number of units or layers and again training till zero error gives even better out-of-sample error.

Double Descent

- With neural networks, more hidden units better than fewer.
- Likewise more hidden layers.
- Often running stochastic gradient descent till zero training error gives good out-of-sample error.
- Often increasing the number of units or layers and again training till zero error gives even better out-of-sample error.

What happened to overfitting and the usual bias-variance trade-off?

Belkin, Hsu, Ma and Mandal (arXiv 2018) *Reconciling Modern Machine Learning and the Bias-Variance Trade-off.*

Some Facts

- In a wide linear model fit by least squares, stochastic gradient descent with small step size leads to *minimum norm* solution [1]:

$$\hat{\beta}_{mn} = \arg\min \|\beta\|_2 : X\beta = y.$$

- Equivalently

$$\hat{\beta}(\lambda) = \arg\min \left\{ \|y - X\beta\|_2^2 + \lambda \|\beta\|_2^2 \right\}$$

  and $\hat{\beta}_{mn} = \lim_{\lambda \to 0} \hat{\beta}(\lambda)$.

- Stochastic gradient *flow* very similar to ridge path [2].

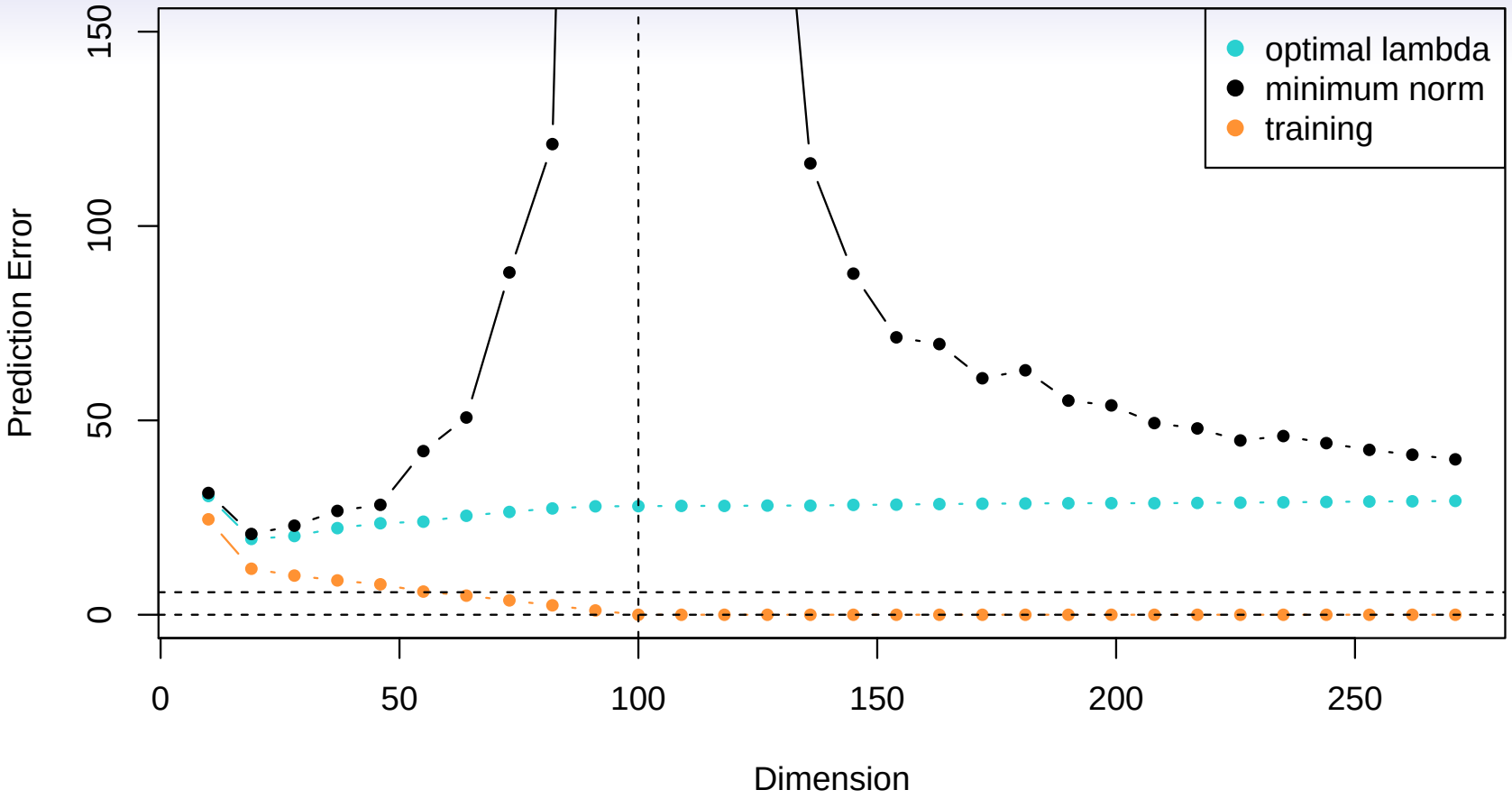
[1] Zhang, S. Bengio, Hardt, Recht, and Vinyals (arXiv 2016) *Understanding deep learning requires rethinking generalization.*
[2] Ali, Kolter, and Ryan Tibshirani (arXiv 2018) *A continuous-time view of early stopping for least squares.*

Simulation

- $y = f(x) + \varepsilon$ with $x \in \mathbb{R}^9$ and f nonlinear and nonadditive, ε Gaussian with $SNR = 3$.
- Training set $n = 100$, test set very large (10K).
- Use additive model

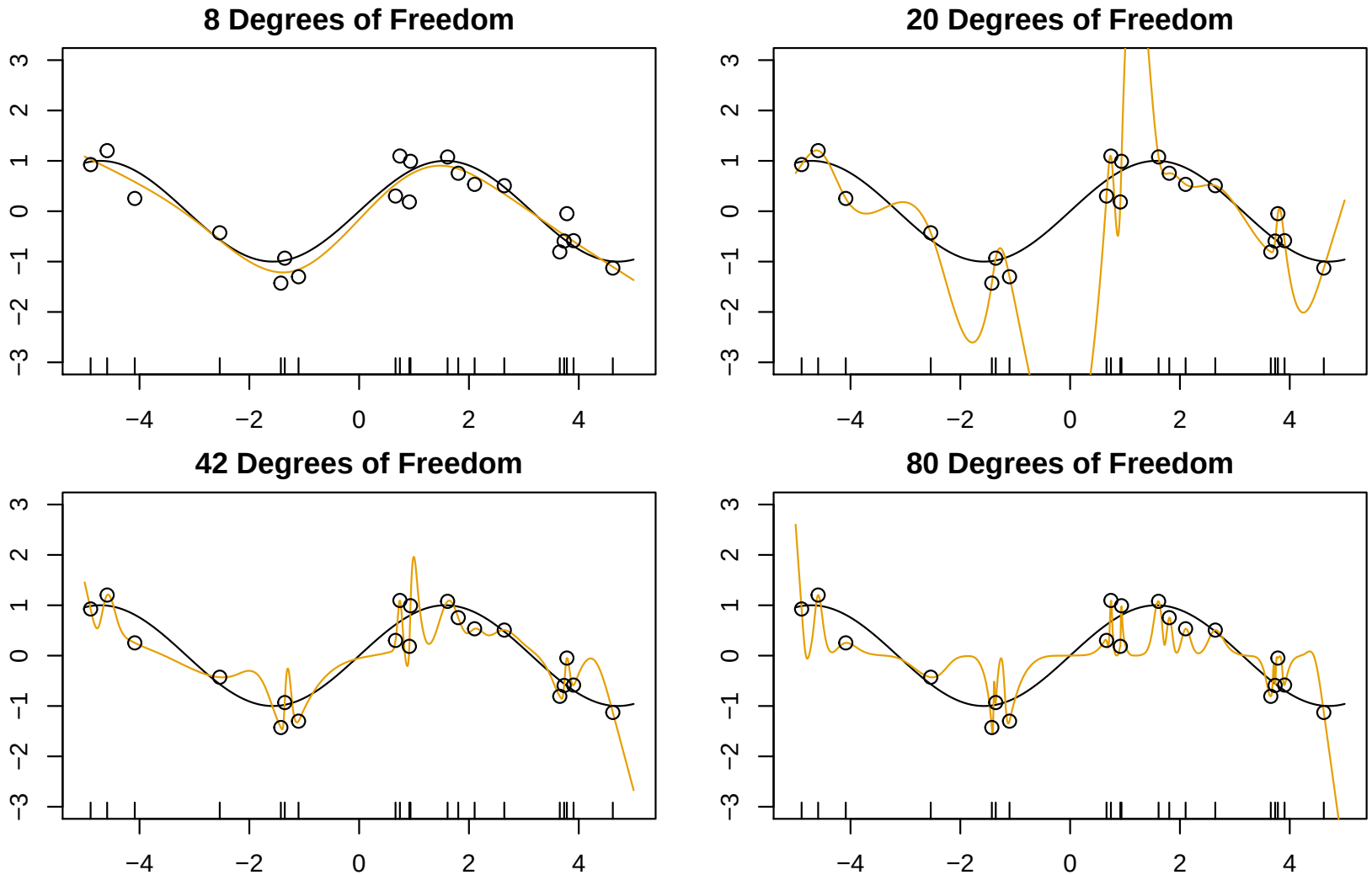
$$g(x; d) = \theta_0 + \sum_{j=1}^{9} \theta_j^\top h_j(x_j; d)$$

  with each $h_j(\cdot; d)$ a d-vector of natural-spline basis functions, with knots chosen at uniform quantiles of the training values for variable x_j.
- The total number of parameters or dimension of the model is $9d + 1$, with d stepping from 1 thru 30. The dimension reaches the training sample size of $n = 100$ when $d = 11$.



- Before dimension=100, model is OLS, and we see usual bias-variance trade-off
- After dimension=100, revert to minimum-norm. As dimension increases above 100, $\|\hat{\theta}_{mn}\|_2$ *decreases* since easier to achieve zero error, and hence less wiggly solutions.

Less Wiggly Solutions



Simpler example — one dimensional function, $n = 20$.

Leave-One-Out Cross-Validation

- Amazing formula for LOO in linear ridge regression:

$$\mathrm{LOO}_\lambda = \sum_{i=1}^{n} (y_i - x_i^\top \hat{\beta}_\lambda^{(-i)})^2 = \sum_{i=1}^{n} \left(\frac{\{(I - H_\lambda)y\}_i}{\{I - H_\lambda\}_{ii}} \right)^2$$

  where $H_\lambda = X(X^\top X + \lambda I)^{-1} X^\top$.

Leave-One-Out Cross-Validation

- Amazing formula for LOO in linear ridge regression:

$$\mathrm{LOO}_\lambda = \sum_{i=1}^{n} (y_i - x_i^\top \hat{\beta}_\lambda^{(-i)})^2 = \sum_{i=1}^{n} \left(\frac{\{(I - H_\lambda)y\}_i}{\{I - H_\lambda\}_{ii}} \right)^2$$

  where $H_\lambda = X(X^\top X + \lambda I)^{-1} X^\top$.

- What about LOO_{mn} when $p \gg n$?

Leave-One-Out Cross-Validation

- Amazing formula for LOO in linear ridge regression:

$$\mathrm{LOO}_\lambda = \sum_{i=1}^n (y_i - x_i^\top \hat{\beta}_\lambda^{(-i)})^2 = \sum_{i=1}^n \left(\frac{\{(I - H_\lambda)y\}_i}{\{I - H_\lambda\}_{ii}} \right)^2$$

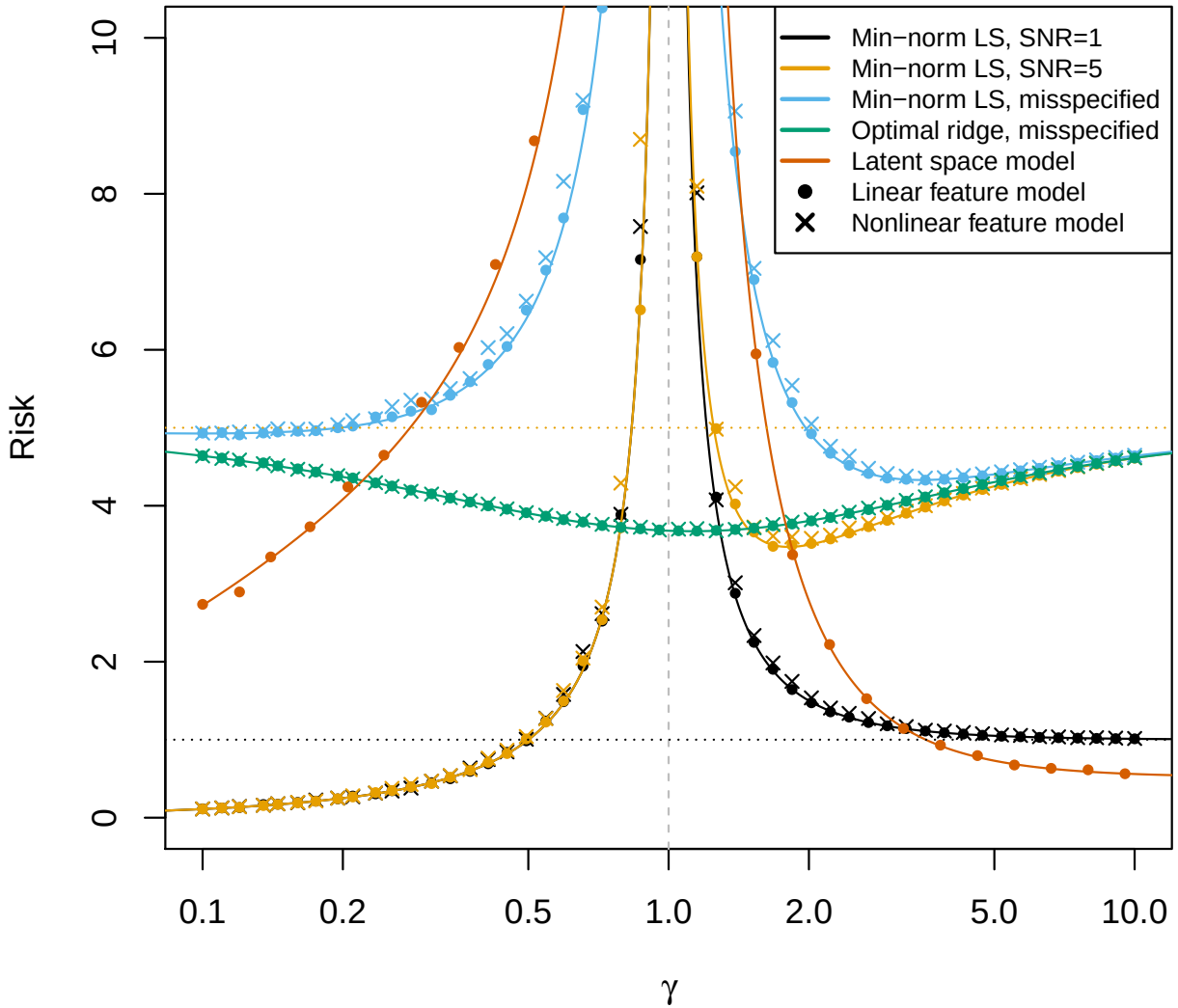
  where $H_\lambda = X(X^\top X + \lambda I)^{-1} X^\top$.

- What about LOO_{mn} when $p \gg n$?

$$\mathrm{LOO}_{mn} = \sum_{i=1}^n \left(\frac{\{K^{-1}y\}_i}{\{K^{-1}\}_{ii}} \right)^2$$

  where $K = XX^\top$. Simple modification when model includes an unpenalized intercept.

Theoretical Study of Double Descent



Surprises in High-Dimensional Ridgeless Least Squares Interpolation TH, Andrea Montanari, Ryan Tibshirani and Saharon Rosset (2020, submitted)

Concluding Comments

- Limiting (minimum-norm) ridge helps explain double descent.

Concluding Comments

- Limiting (minimum-norm) ridge helps explain double descent.
- Lots left out — eg ridge and Tikhonov, splines, and GAMs.

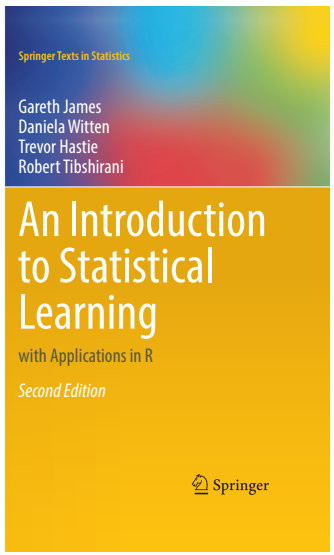
Concluding Comments

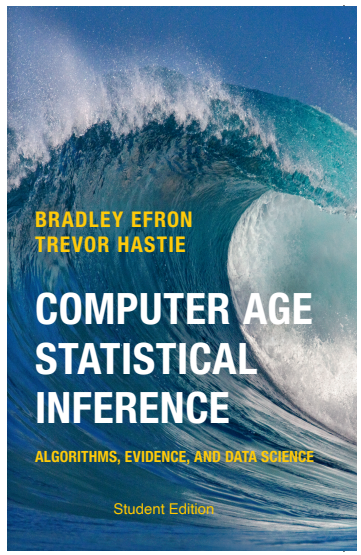
- Limiting (minimum-norm) ridge helps explain double descent.
- Lots left out — eg ridge and Tikhonov, splines, and GAMs.
- I do love ridge!

Concluding Comments

- Limiting (minimum-norm) ridge helps explain double descent.
- Lots left out — eg ridge and Tikhonov, splines, and GAMs.
- I do love ridge!

And now for some free advertising

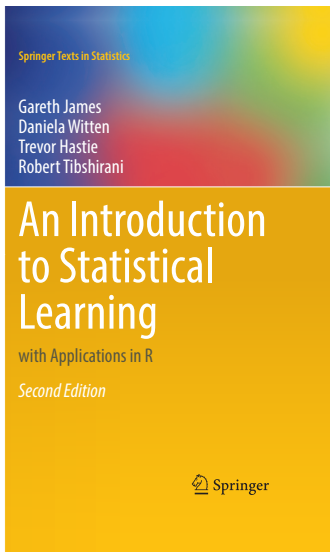


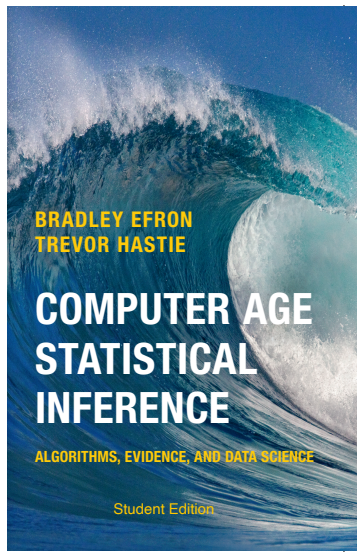


Concluding Comments

- Limiting (minimum-norm) ridge helps explain double descent.
- Lots left out — eg ridge and Tikhonov, splines, and GAMs.
- I do love ridge!

And now for some free advertising





Thank you for your attention!