# Linear Models in Regression

Prof. Rafal Kustra

Dalla Lana SPH

September 28, 2016

# Outline

# Degrees of Freedom in LS

- Typically count columns of design matrix $X$ (including intercept and any dummy variable codings and interactions)
- Gives expected drop in RSS (in units of $\sigma^2$) under null hypothesis
- The DoF can also be calculated using the trace of the Hat matrix (for $n > p$ and rank of $X = p$):

$$\text{Tr}\left[X(X'X)^{-1}X'\right] = \text{Tr}\left[X'X(X'X)^{-1}\right] = p$$

- This will be useful for penalized regression models

# Curse of Dimensionality

Bellman, 1961

- All supervised learning methods (including regression) rely on *local data similarities*: when the inputs are similar we expect similar outputs
- The locality of data breaks down in higher dimensions
  - At least for numerical covariate spaces, *local* means distance (usually Euclidean) to nearest neighbours
  - Even for categorical (nominal) spaces, if there are many of them, most of them will be separated by large "distances"
- Consider $k$-Nearest Neighbour method in $p$-dimensional space using Euclidean distance. Assume $p$ covariates are uniformly distributed within a $p$-dim (hyper)cube. How much along each dimension do we have to move to capture 10% of data? On average: $10^{1/p}$. For 10-D space to capture just 10% NN, we need to cover 80% of each side of the cube! Hardly local

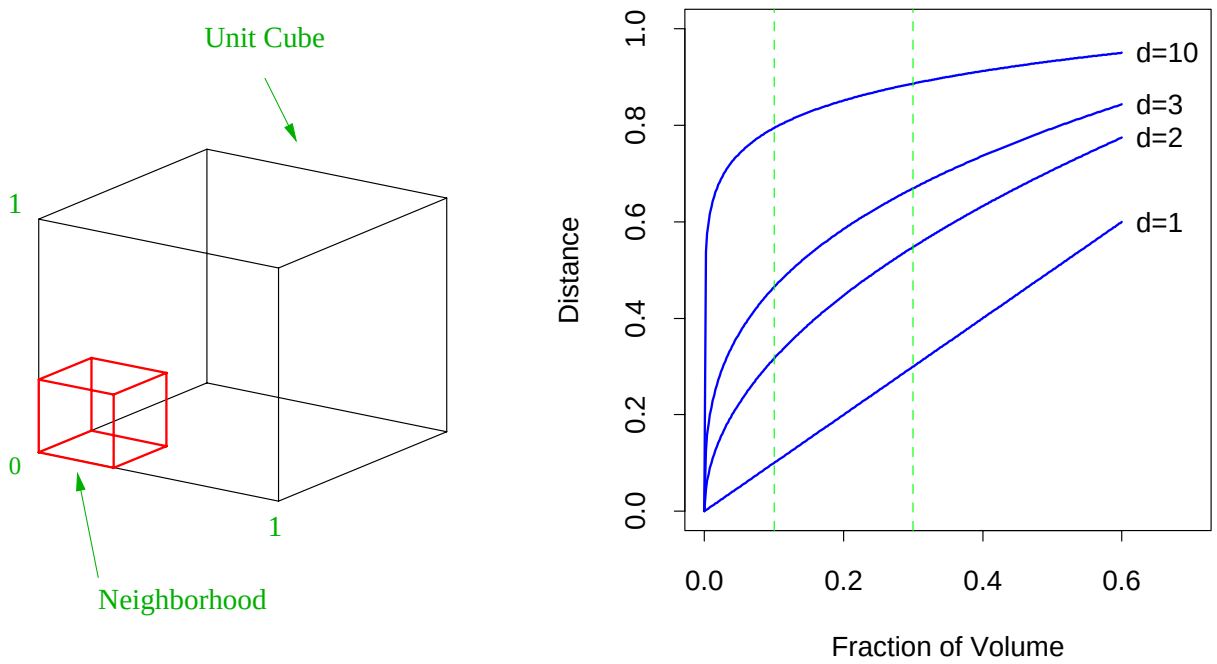



Figure 2.6: *The curse of dimensionality is well illustrated by a subcubical neighborhood for uniform data in*

- Another manifestation of *the curse* is that most of the data lies close to the edge of the space
- Consider $N$ points uniformly distributed within $p$-dim unit ball. From the center, what is the median distance to nearest point?

$$d(p, N) = \left(1 - \frac{1}{2}^{1/N}\right)^{1/p}$$

- For $N = 5000, p = 10$ this is 0.52 so half the time the neareast point lies closer to edge of the ball than to the center. Prediction close to edges is problematic

# Outline

# Categorical Inputs

- Almost always some predictors will be categorical (nominal, qualitative)
- These are typically converted using `contrasts` – where $K$ levels are converted into $K - 1$ largely binary columns
- Few problems:
  - Potentially catastrophic expansion of input dimensionality
  - Limits predictive ability
  - Un(der)-used levels – high variance
  - Problems with basis expansion, shrinkage

# Two problems with OLS

## Need more flexibility

Linear surface is just not enough to model our data

## Need to reduce flexibility

On the other hand, with very large number of predictors we cannot build a (reliable) linear estimator of $f(x)$

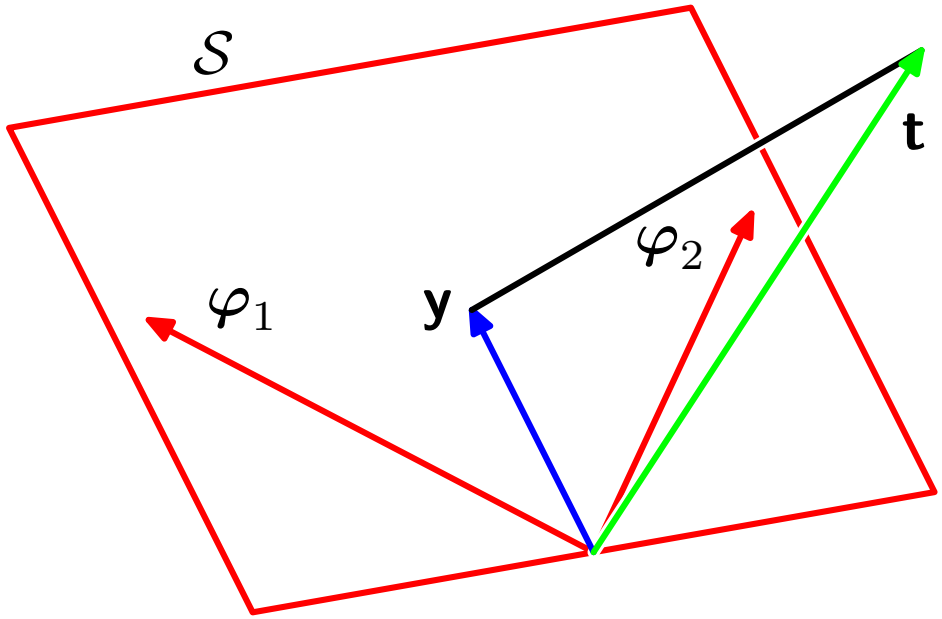
More realistically both problems can occur at once (too many predictors but need non-linear effects nonetheless)

# Basis expansions

- Linear Regression fits a linear model in $x$'s
- To provide non-linearity, can include non-linear functions of some $x$'s into the $x$ matrix (e.g., polynomial regression)

$$x_j \longrightarrow [\varphi_1(x_j), \ldots, \varphi_k(x_j)]$$

- This changes the *space of regression fit*
- The data is represented in enhanced space




- Using my notation: $y \equiv X\beta$ and $\boldsymbol{t} \equiv \boldsymbol{y}$ in figure

# Basis Expansion procedure

1. Choose which $x$ will be expanded in basis
2. Choose a basis set for each $x$ to be expanded
   - May expand some $x$'s in polynomial basis and others in Fourier (cosine) basis
3. Form an *evaluated basis* matrix, $\Phi(X)$
   1. Replace a column for $x_j$ to be expanded with $k_j$ columns, one for each basis function
   2. Evaluate each basis function, $\varphi_k$ at each of $n$ values of $x_j$
4. Fit a LR model $Y \sim \Phi(X)\beta_\Phi$
5. To visualize the effect of an expanded predictor, $x_j$, plot:

$$\widehat{f}_j(x) = \sum_\kappa \beta_{\phi;\kappa}\varphi_\kappa(x)$$

where $\kappa$ indexes these basis functions that were used to expand $x_j$

# Common basis sets

- Polynomials
- Polynomial splines
  - Divide the range of predictor $x$ into $M + 1$ regions using $M$ knots
  - Fit a seperate polynomial for each region
  - But make sure that at knot, where 2 polynomials meet, they meet "nicely": have the same value and same $d - 1$ derivatives for $d$-degree polynomial
- Radial Basis Sets (kernel basis). For example a Gaussian kernel:

$$\varphi_\kappa(x) = \exp\left\{-\frac{||\boldsymbol{x} - \boldsymbol{\mu}_\kappa||}{2s^2}\right\}$$

- Fourier basis ($\cos(2\pi kx), \sin(2\pi kx)$ for ($x \in (0, 1)$ and $k = 0, 1, \ldots$)
- Wavelets
- Trees

# Challenges with Basis Expansions

- Must choose basis set well
- Must choose predictors to expand
- Does not work well on non-continuous data
- Significantly expands your dimensionality

# Additive Models

- In general, we think of modeling each(some) predictors as smooth functions, replacing the linear term:

$$f(\boldsymbol{x}) = \sum_j \beta_j x_j$$

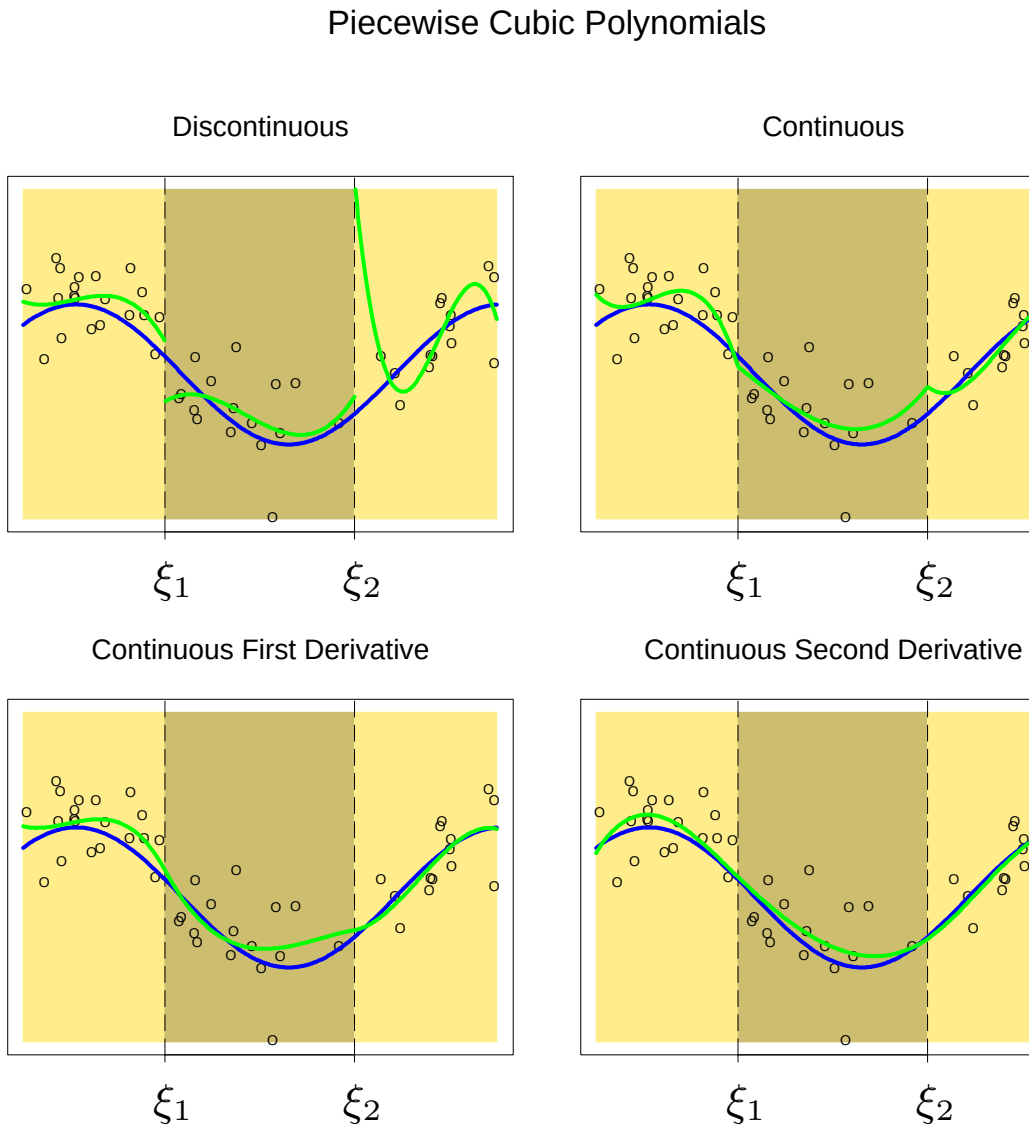
with *additive predictor*:

$$f_{\text{GAM}}(\boldsymbol{x}) = \sum_j f_j(x_j)$$

- A common way to do it is to (implicitely) expand each $x_j$ into a non-linear basis set (usually using polynomial *splines*)
- Important to realize that the model is still *additive*, that is the overall function of $p$-variables, $\eta_{\text{GAM}}(\boldsymbol{x})$, is composed of $p$ individual univariate functions
  - As a consequence a surface cross-section in direction of $x_j$ has the same shape for all combinations of other variables

# Introduction to regression splines

- Global polynomial fits too ... global
- Divide the data range in $K + 1$ regions using $K$ *internal knots*, $\xi_j$
- Each region gets a low-order (almost always cubic) polynomial
- BUT require that at the knots two cubics are continuous, and have first and second derivatives equal



# Fitting regression splines

- Without boundary restrictions have $K + 4$ DFs
- *Natural* cubic splines restrict the fit to be linear beyond range of data (DF=?)
- In practice one often supplies *boundary* knots at extreme data points (DF=$K + 2$)
- You can check that the basis set:

$$\phi_1(x) = 1, \phi_2(x) = x, \phi_3(x) = x^2, \phi_j(x) = (x - \xi_j)_+^3$$

(where $+$ subscript means take positive part only, and $j$ runs through interior knots) satisfies properties of (no-boundary-restricted) cubic splines. Hence one could compose a matrix $\Phi(X)$ with $K + 4$ columns and fit via LS:

$$\widehat{\boldsymbol{y}} = \Phi(\Phi'\Phi)^{-1}\Phi'\boldsymbol{y}$$

- In practice either B-spline or Natural cubic spline basis sets are used (`bs()` and `ns()` functions in **R**)

# Tensor-product basis functions

- Sometimes additive restriction is not appropriate:

$$\eta(\boldsymbol{x}) = \sum_j f_j(x_j)$$

- One easy way to compose muldidimensional basis sets is to use *tensor-product basis*

## TPB in 2-D for $(x_1, x_2)$

1. Start with your favourite basis set in 1-D

$$[\varphi_1(x), \ldots, \varphi_K(x)]$$

2. Form all possible $\frac{K(K-1)}{2}$ products accross two dimensions

$$[\varphi_1(x_1)\varphi_1(x_2), \varphi_1(x_1)\varphi_2(x_2), \ldots, \varphi_K(x_1)\varphi_{K-1}(x_2), \varphi_K(x_1)\varphi_K(x_2)]$$

# TSB with B-splines

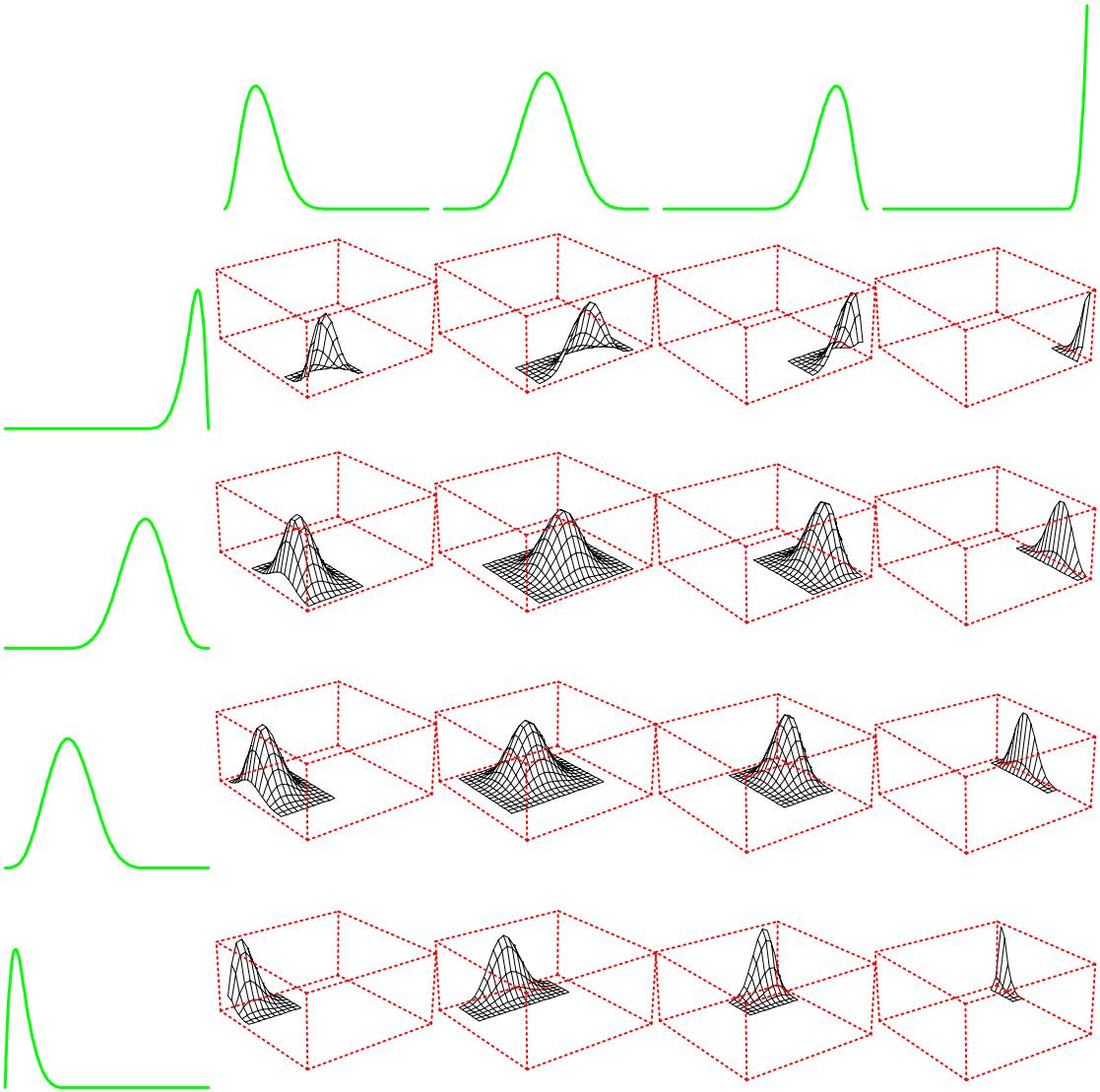


Figure 5.10: *A tensor product basis of B-splines, showing some selected pairs. Each two-dimensional function*

# Penalization-Shrinkage

- Now for reducing complexity - we will talk about removing dimensions later
- So far each predictor (either original one, or each of evaluated basis one) gets “full attention” - 1 degree of freedom
- Hence we project our response, $y$, into full linear space
- This may be too much

# Degrees of Freedom

- Often misunderstood
- Officially an expected drop in sum of squares (as multiple of $\sigma$) resulting from expanding a model under null hypothesis
- In simple linear models - one term gets 1DF
- It means that Least Square minimizer has full flexibility to minimize LS in the direction of the term
- This is equivalent to no restriction on the (abs) size of resulting $\widehat{\beta}$'s:

$$\widehat{\beta}_{\text{OLS}} = \underset{\beta}{\text{argmin}} \sum_{i=1}^{n} \left( y_i - \sum_{j=1}^{p} \beta_j \phi_j(\boldsymbol{x}_i) \right)^2$$

# Ridge Regression

- Any restriction on $\widehat{\beta}$ *potentially* reduce degrees of freedom (complexity). Minimizer has limited ability to reduce RSS.
- Consider the simplest case: restricting total norm of vector $\beta$, $\|\beta\|^2 = \sum_j \beta_j^2$:

$$\widehat{\beta}_{\text{ridge}}(c) = \underset{\beta}{\text{argmin}}\, \text{RSS} \qquad \text{s.t.} \quad \|\widehat{\beta}_{\text{ridge}}\|^2 \leq c$$

  Using Lagrange multipliers this can be shown to be equivalent to:

$$\underset{\beta}{\text{argmin}}\, \text{RSS} + \lambda\|\beta\|^2$$

  Where there is a one-to-one mapping between $c$ and $\lambda$
- The resulting $\beta$ is based on lower-complexity model (fewer DFs) *if* $c < \|\beta_{\text{OLS}}\|^2$. Then the resulting RSS will be larger too.

# Ridge - Solution

- Ridge minimization problem has a closed-form solution:

$$\widehat{\beta}_{\text{RIDGE}} = X(X^T X + \lambda I)^{-1} X^T \boldsymbol{y}$$

  For a given $\lambda$ it is as fast as OLS: advantage.
- Another advantage: computes with $p > n$ (but one never knows how well...)
- It is obviously not unbiased (since OLS is), but the variance is reduced:

$$\text{Var}(\widehat{\beta}_{\text{RIDGE}} = \sigma^2 (X^T X + \lambda I)^{-1}$$

# DF of Ridge

- The ridge hat matrix is:

$$H_r(\lambda) = X(X'X + \lambda I)X'$$

- Using SVD, let:

$$X = UDV'$$

where $D$ contains non-negative square roots of eigenvalues of $X'X$, say $\gamma_j$, $j = 1, \ldots, P$. Then

$$\begin{aligned} \text{Tr}\left(H_r(\lambda)\right) &= \text{Tr}\left(VD_\gamma V'(VD_\gamma V^T + \lambda I)^{-1}\right) \\ &= \text{Tr}\left(VD_\gamma V'(VD_{\gamma^*} V')^{-1}\right) \\ &= \sum_j \frac{\gamma_j}{\gamma_j^*} \end{aligned}$$

where $\gamma_j^* = \gamma + \lambda$

- This shows that ridge penalty (with $\lambda > 0$) reduces degrees of freedom of the fit in the amount that is relative to eigenvalues of $X'X$.

# More on ridge



- In similar fashion one can show that ridge reduces variance of $\widehat{y}$ compared to straight LS
- Ridge really only makes sense when:
  1. only numerical predictors are penalized
  2. predictors (columns of $X$) are centered
  3. They are either on similar scale or normalized

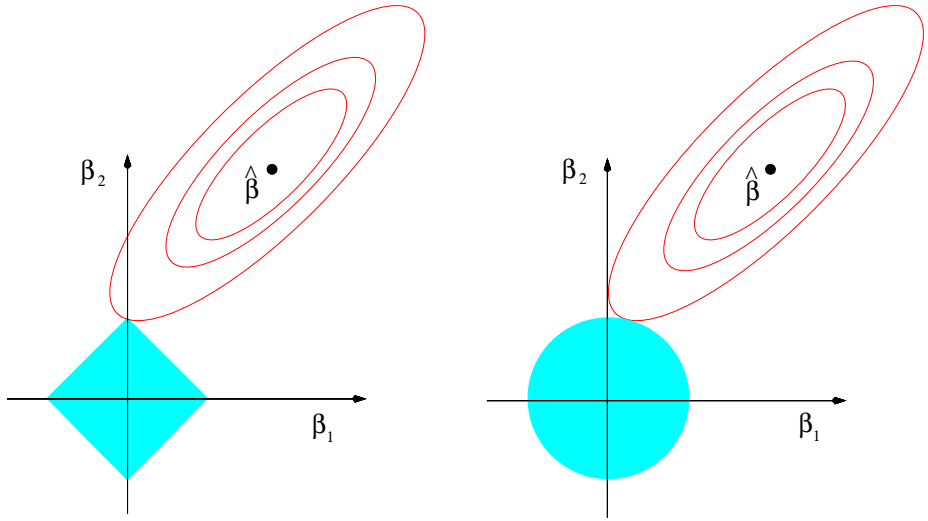



Figure 3.12: *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions* $|\beta_1| + |\beta_2| \leq t$ *and* $\beta_1^2 + \beta_2^2 \leq t^2$*, respectively, while the red ellipses are the contours of the least squares error function.*

# LASSO regression

- Similar in principle to ridge:

$$\underset{\beta}{\operatorname{argmin}}\, \mathrm{RSS} + \lambda \|\beta\|^2$$

- Two caveats:
  - No closed-form solution: quadratic programming (complex computations)
  - Often results in some $\widehat{\beta}$'s equal *exactly* zero
- Extended in LARS models

# Cubic Smoothing Splines

- A very neat solution to the problem of selecting knots which also happens to be a penalized regression problem
- With single continuous predictor, $x$, suppose we require to fit a non-linear regression by finding a minimizing function $f(x)$ subject to smoothness restriction:

$$\widehat{f} = \underset{f}{\operatorname{argmin}} \sum_i [y_i - f(x_i)]^2 + \lambda \int_x \left(f''(x)\right)^2 \mathrm{d}x$$

- Remarkably, it turns out that the solution to that variational problem is a *natural* cubic splines with knots at all unique values of $x_i$

# Smoothing Splines and Penalized Regression

- Nominally we have a overparameterization problem since there are ?? DFs here, but this is a *penalized* regression problem, so that effective DFs are controled by $\lambda$. Let $B$ be an $N \times N+2$ *evaluated* B-spline basis set matrix, (with $N+2$ B-spline basis set, $B_j(x)$) and let:

$$\Omega_{jk} = \int B_j''(x) B_k''(x) \mathrm{d}x$$

be an $N+2 \times N+2$ penalty matrix. Then the smoothing spline regression fit (using B-spline basis set) is:

$$\widehat{\boldsymbol{y}} = B(B'B + \lambda\Omega)^{-1} B'\boldsymbol{y}$$

# Outline

# Bias-Variance decomposition: OLS

- With training sample we produce a *prediction rule*:

$$\widehat{y}(x_0) \equiv \widehat{f}_{\mathcal{T}}(x_0)$$

- At a *fixed* point, $x_0$, how good, on average, is it?

$$\begin{aligned}
\mathrm{E}_{\mathcal{T}}\left[f(x_0) - \widehat{y}_0\right]^2 &= \mathrm{E}_{\mathcal{T}}\left[(f(x_0) - \mathrm{E}_{\mathcal{T}}\widehat{y}_0) + (\mathrm{E}_{\mathcal{T}}\widehat{y}_0 - \widehat{y}_0)\right]^2 \\
&= \mathrm{E}_{\mathcal{T}}\left[f(x_0) - \mathrm{E}_{\mathcal{T}}\widehat{y}_0\right]^2 + \mathrm{E}_{\mathcal{T}}\left[\mathrm{E}_{\mathcal{T}}\widehat{y}_0 - \widehat{y}_0\right]^2 \\
&= \mathrm{Bias}^2(\widehat{y}_0) + \mathrm{Var}_{\mathcal{T}}(\widehat{y}_0)
\end{aligned}$$

# Squared Prediction Error of OLS

- MSE shows us how far we are from the true (unknown) function
- SPE shows us how far we are from the observed response, $y_0$, at point $x_0$:

$$\begin{aligned}
\text{SPE}(x_0) =& \text{E}_{y_0|x_0}\text{E}_{\mathcal{T}}\left[y_o - \widehat{y}_0\right]^2 \\
=& \text{E}_{y_0|x_0}\text{E}_{\mathcal{T}}\left[\epsilon_0 + (f(x_0) - \widehat{y}_0)\right]^2 \\
=& \text{E}_{y_0|x_0}\text{E}_{\mathcal{T}}\left[\epsilon_0\right]^2 + \text{E}_{y_0|x_0}\text{E}_{\mathcal{T}}\left[f(x_0) - \widehat{y}_0\right]^2 - 2 \cdot 0 \\
=& \text{Var}(y|x_0) + \text{MSE} \\
=& \sigma^2 + \text{Bias}^2(\widehat{y}_0) + \text{Var}_{\mathcal{T}}(\widehat{y}_0)
\end{aligned}$$

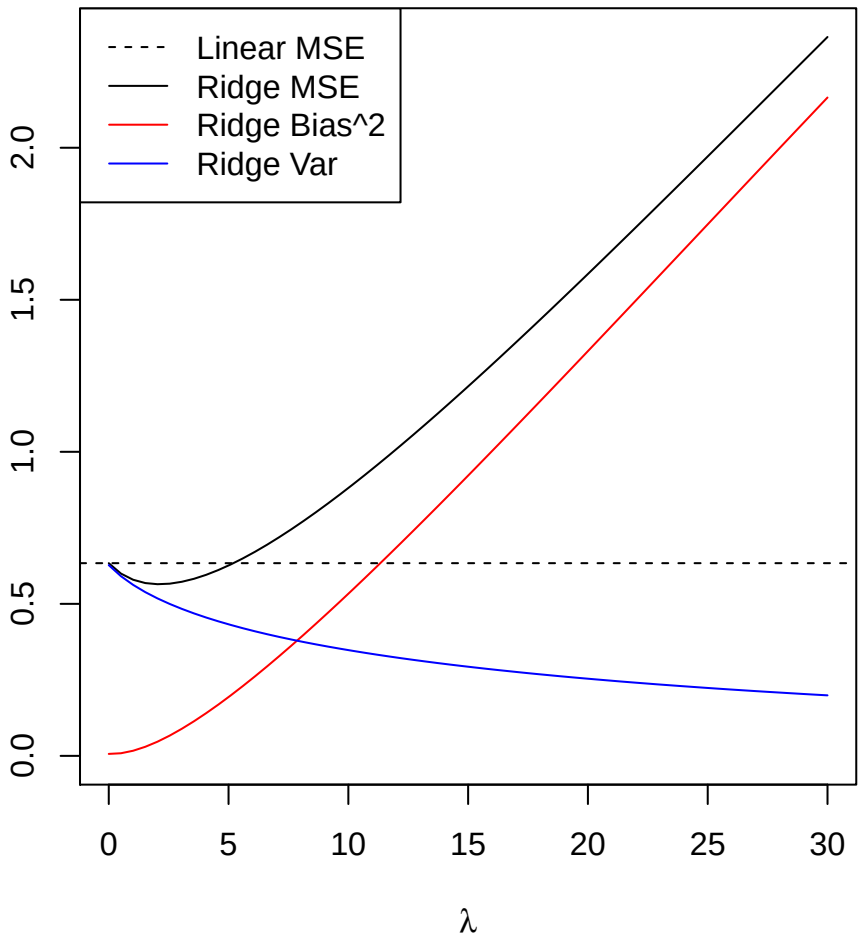
# Bias-Variance: Ridge

- In OLS, variance governed by term $(X'X)^{-1}$ which becomes $(X'X + \lambda I)^{-1}$ in ridge
- Variance will decrease with increasing $\lambda$ (what happens with $\lambda \to \infty$?) and opposite will happen for bias (if true function linear)
- Hence MSE (and PSE) will be minimized for some optimal $\lambda$
- This is even when the *true regression function is linear*

# True Linear model: n=50, p=30

From Ryan Tibshirani's slides

Ridge regression can still outperform linear regression in terms of mean squared error:

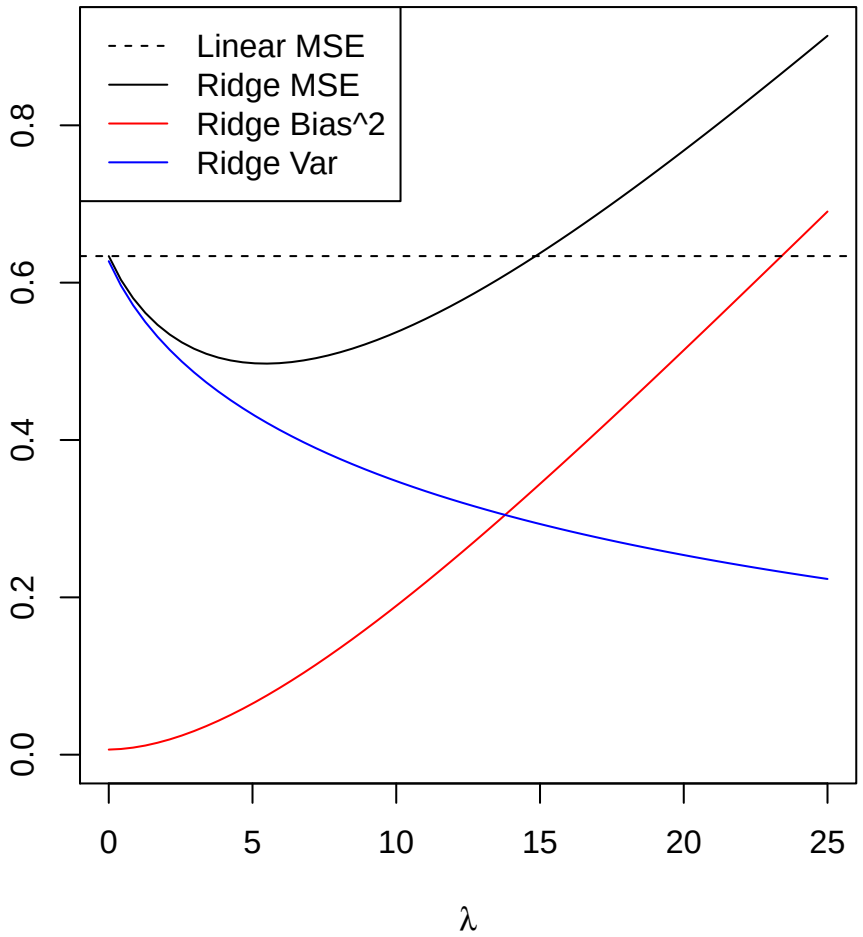



Only works for $\lambda$ less than $\approx 5$, otherwise it is very biased. (Why?)

# True Linear model: n=50, p=30 but 20 $\beta$'s =0

From Ryan Tibshirani's slides

Ridge regression performs well in terms of mean-squared error:




Why is the bias not as large here for large $\lambda$?