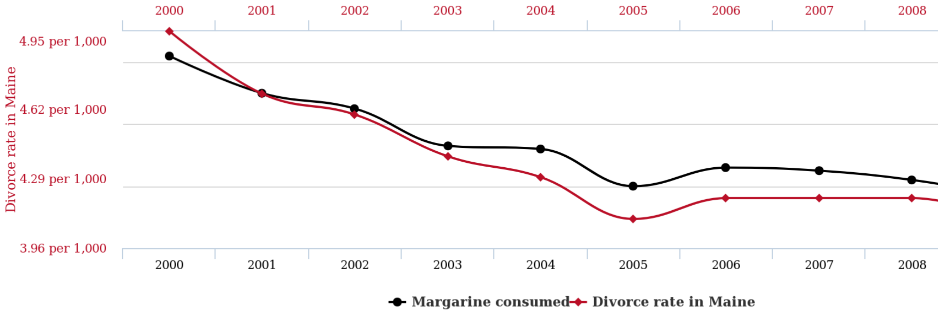
STAT406- Methods of Statistical Learning Lecture 7

Matias Salibian-Barrera

UBC - Sep / Dec 2018

Divorce rate in Maine
correlates with
Per capita consumption of margarine



Correlation: 99.26%

http://www.tylervigen.com/spurious-correlations

Model / feature selection - LASSO

- Another regularized method is given by LASSO

$$\min_{\alpha,\boldsymbol{\beta}} \sum_{i=1}^{n} \left(y_i - \alpha - \boldsymbol{\beta}' \mathbf{x}_i \right)^2 + \lambda \sum_{j=1}^{p} \left| \beta_j \right|$$

$$\min_{\alpha,\boldsymbol{\beta}} \sum_{i=1}^{n} \left(y_i - \alpha - \boldsymbol{\beta}' \mathbf{x}_i \right)^2 + \lambda \, \| \boldsymbol{\beta} \|_1$$

for some $\lambda > 0$

Model / feature selection - LASSO

- The above is equivalent to

$$\min_{\alpha,\beta} \sum_{i=1}^{n} \left(y_i - \alpha - \beta' \mathbf{x}_i \right)^2$$

subject to

$$\sum_{j=1}^{p} \left| \beta_j \right| \leq K$$

for some $K > 0$

LASSO

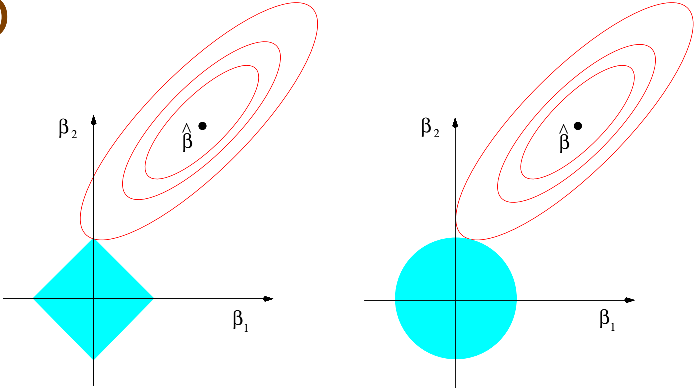
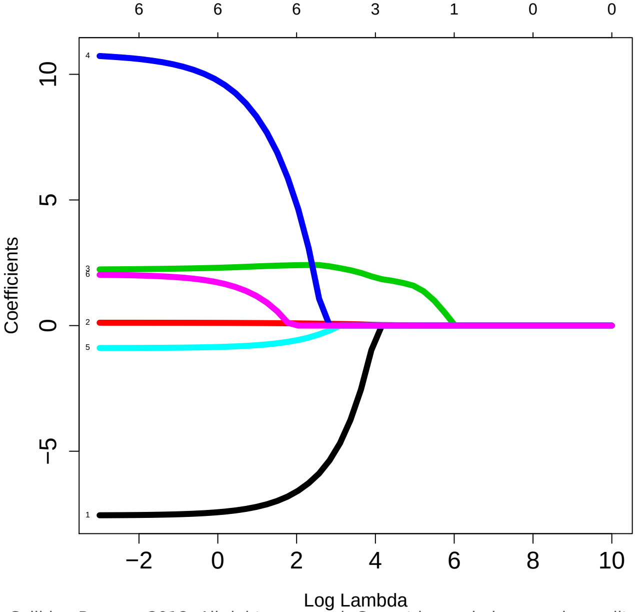


FIGURE 3.11. *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions* $|\beta_1| + |\beta_2| \leq t$ *and* $\beta_1^2 + \beta_2^2 \leq t^2$*, respectively, while the red ellipses are the contours of the least squares error function.*

Credit data - glmnet output



Credit data - glmnet output

```
a <- glmnet(x=xm, y=yc, lambda=lambdas,
    family='gaussian', alpha=1, intercept=FALSE)

> coef(a, s=1)
7 x 1 sparse Matrix of class "dgCMatrix"
                    1
(Intercept)   .
Income       -7.4285710
Limit         0.1078894
Rating        2.3006418
Cards         9.7499618
Age          -0.8515917
Education     1.7182477
```

Credit data - glmnet output

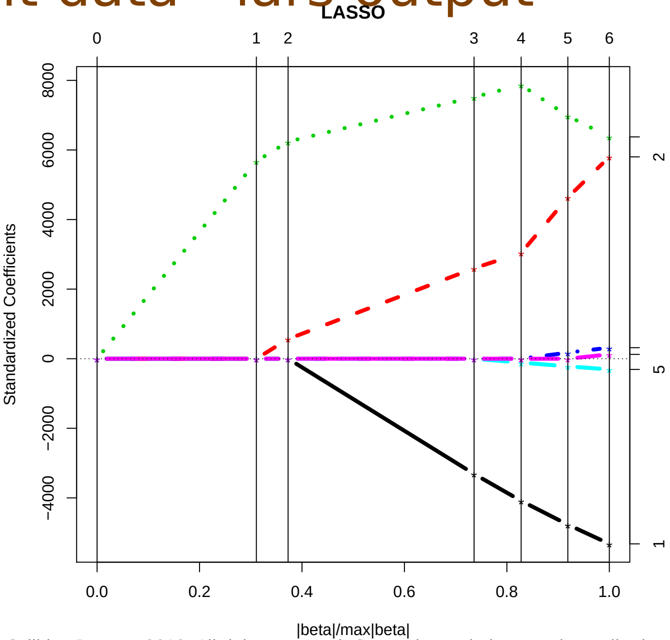
```
> coef(a, s=exp(4))
7 x 1 sparse Matrix of class "dgCMatrix"
                       1
(Intercept)  .
Income      -0.63094341
Limit        0.02749778
Rating       1.91772580
Cards        .
Age          .
Education    .
```

Credit data - another implementation

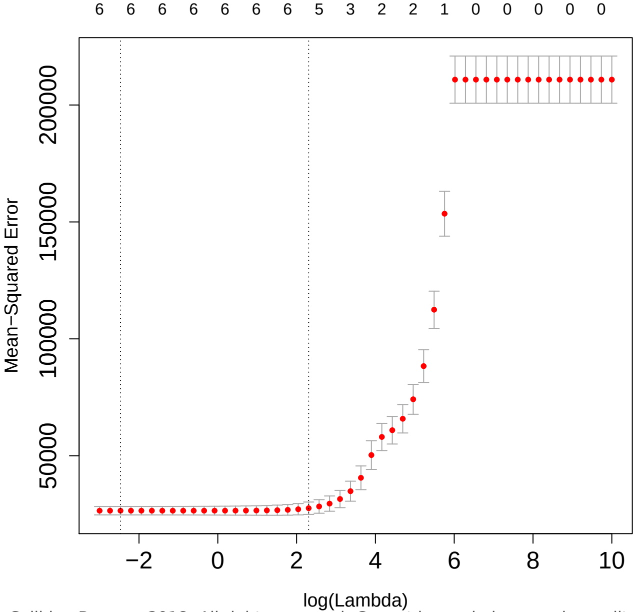
```
> library(lars)
> b <- lars(x=xm, y=yc, type='lasso', intercept=FALSE)
> coef(b)
        Income      Limit  Rating     Cards        Age Education
[1,]  0.000000 0.00000000 0.000000  0.000000  0.0000000  0.000000
[2,]  0.000000 0.00000000 1.835963  0.000000  0.0000000  0.000000
[3,]  0.000000 0.01226464 2.018929  0.000000  0.0000000  0.000000
[4,] -4.703898 0.05638653 2.433088  0.000000  0.0000000  0.000000
[5,] -5.802948 0.06600083 2.545810  0.000000 -0.3234748  0.000000
[6,] -6.772905 0.10049065 2.257218  6.369873 -0.6349138  0.000000
[7,] -7.558037 0.12585115 2.063101 11.591558 -0.8923978  1.998283
> b

Call:
lars(x = xm, y = yc, type = "lasso", intercept = FALSE)
R-squared: 0.878
Sequence of LASSO moves:
     Rating Limit Income Age Cards Education
Var       3     2      1   5     4         6
Step      1     2      3   4     5         6
```

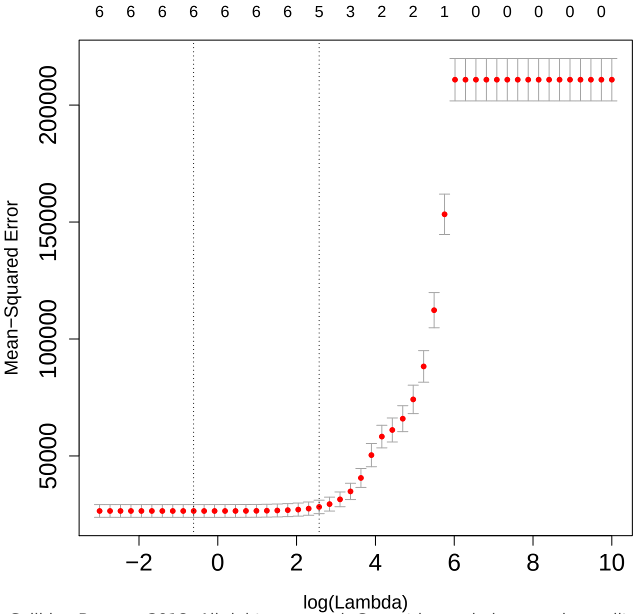
Credit data - lars output



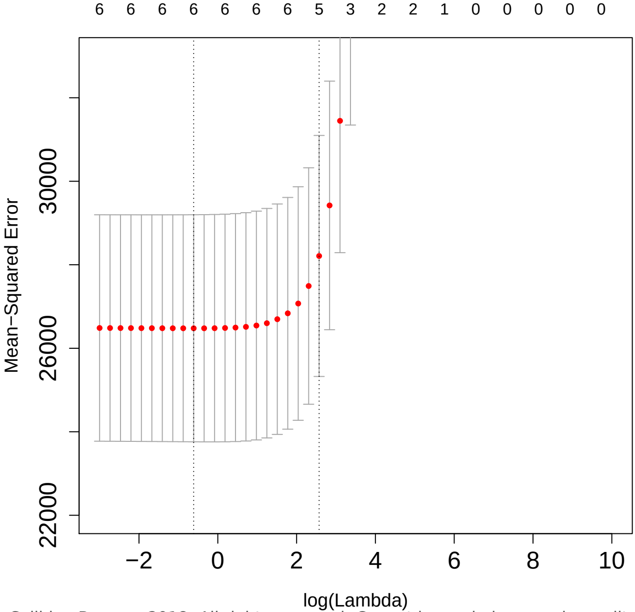
Credit data - CV - glmnet



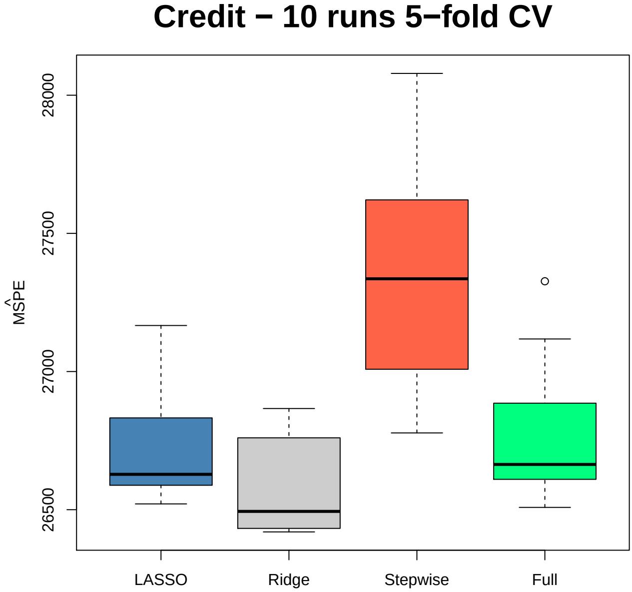
Credit data - CV - another run



Credit data - CV - zoom



Model / feature selection - LASSO



Model / feature selection - LASSO

- Worse estimated MSPE than Ridge Regression in this case
- It provides a sequence of explanatory variables, an ordered set of models
- Much like stepwise, but with better MSPE in this case

Model / feature selection - LASSO

- Why does it work? It is the convex proxy for the “nuclear norm”
- Also generates infinitely many estimates, but there’s a clever algorithm
- Inference?

Model / feature selection - LASSO

- When covariates are correlated, LASSO will typically pick any one of them, and ignore the rest
- Ridge Regression, on the other hand, combines the coefficients of correlated covariates, but doesn't provide sparse models

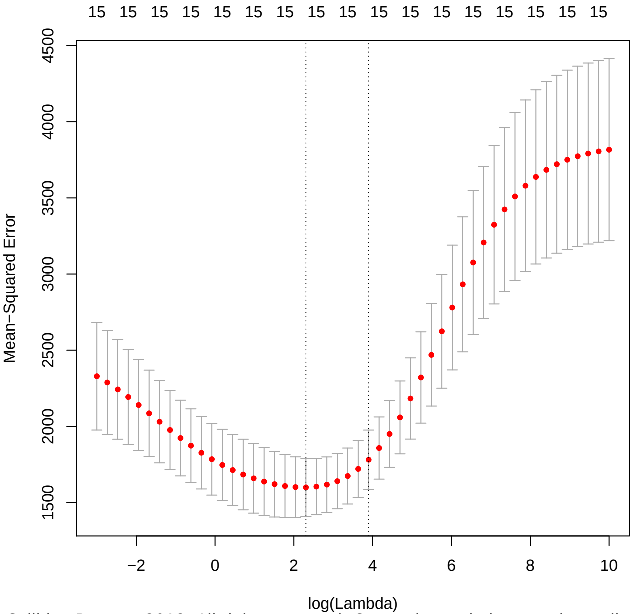
Ridge vs. LASSO

- Compare Ridge and LASSO on the air pollution data

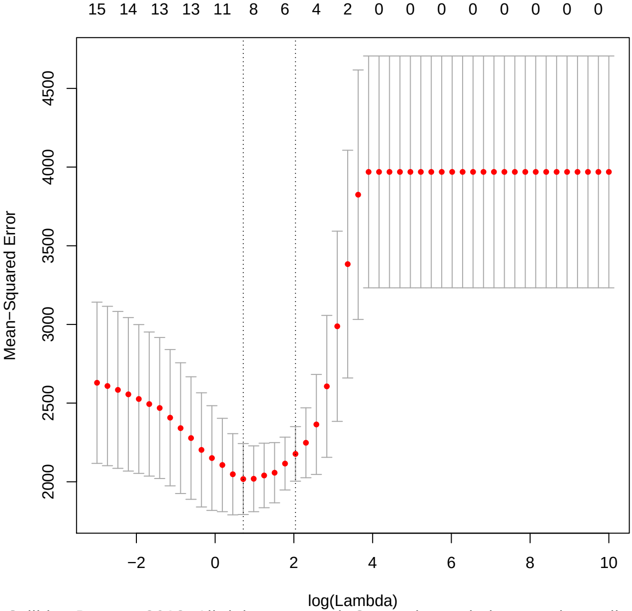
Air pollution example

```
airp <- read.table('..-30861_CSV-1.csv',
    header=TRUE, sep=',')
y <- as.vector(airp$MORT)
xm <- as.matrix(airp[, names(airp) != 'MORT'])
# Ridge
set.seed(123)
air.l2 <- cv.glmnet(x=xm, y=y, lambda=lambdas,
    nfolds=5, alpha=0, family='gaussian',
    intercept=TRUE)
# LASSO
set.seed(23)
air.l1 <- cv.glmnet(x=xm, y=y, lambda=lambdas,
    nfolds=5, alpha=1, family='gaussian',
    intercept=TRUE)
```

Air pollution - Ridge



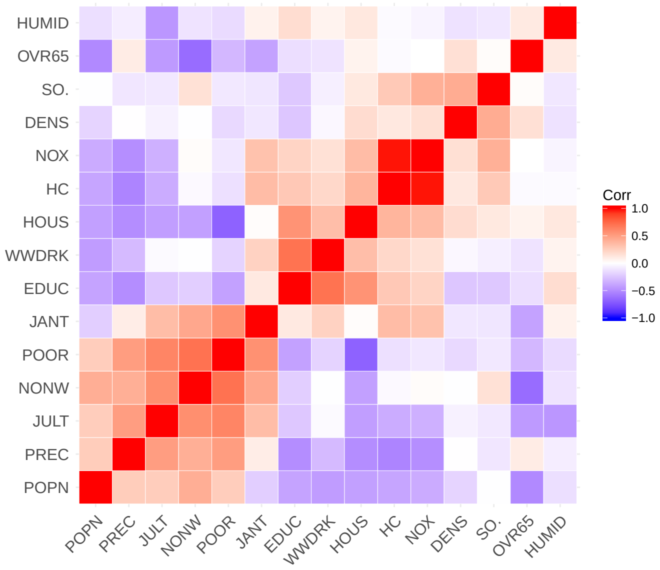
Air pollution - LASSO



Air pollution example

	Ridge	LASSO
(Intercept)	1179.335	1100.355
PREC	1.570	1.503
JANT	-1.109	-1.189
JULT	-1.276	-1.247
OVR65	-2.571	.
POPN	-10.135	.
EDUC	-8.479	-10.510
HOUS	-1.164	-0.503
DENS	0.005	0.004
NONW	3.126	3.979
WWDRK	-0.476	-0.002
POOR	0.576	.
HC	-0.035	.
NOX	0.064	.
SO.	0.240	0.228
HUMID	0.372	.

Air pollution - Correlations



Model / feature selection - LASSO

- Oracle - consistency
- Problem: when $n < p$, LASSO will only choose up to n variables
- When covariates are correlated, LASSO will typically pick any one of them, and ignore the rest
- Ridge Regression, on the other hand, combines the coefficients of correlated covariates, but doesn't provide sparse models

Elastic Net

- Elastic Net is a compromise between the two:

$$\min_{\beta_0, \boldsymbol{\beta}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \boldsymbol{\beta}' \mathbf{x}_i \right)^2 + \lambda \left[\alpha \| \boldsymbol{\beta} \|_1 + \frac{(1-\alpha)}{2} \| \boldsymbol{\beta} \|_2^2 \right]$$

for some $\lambda > 0$ and $0 \leq \alpha \leq 1$.

Elastic Net

- $\alpha = 0$ reduces to Ridge Regression
- $\alpha = 1$ reduces to LASSO
- α needs to be chosen... how would you find a good choice for α?

Air pollution example

- There are correlated covariates
- LASSO solution picks one of each group early on and relegates the rest to the end of the sequence
- Ridge Regression includes all variables always
- EN with $\alpha = 0.10$ gives a nice path of solutions...
- CV? bivariate search, unless α can be chosen beforehand

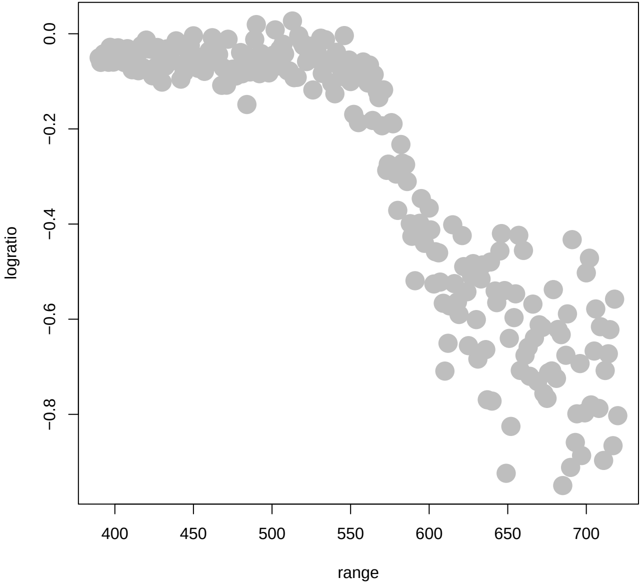
More flexible regression

- What if the regression function

$$E[Y|\mathbf{X}] = f(\mathbf{X})$$

  is not linear?
- Example LIDAR

LIDAR



Non-linear regression

- Model: $E[Y | X_1, X_2, \ldots, X_p] = f(X_1, X_2, \ldots, X_p; \theta_1, \theta_2, \ldots, \theta_k)$
- This is typically a non-linear model
- But it is fully parametric
- The parameters are $\theta_1, \theta_2, \ldots, \theta_k$
- Using MLE (or LS) we can obtain estimates $\hat{\theta}_1, \ldots, \hat{\theta}_k$
- ... and associated standard errors!

Non-linear regression

- Sometimes it's difficult to find an appropriate family of functions

- Polynomials are a natural choice

$$m(x) = m(x_0) + \frac{1}{2} m'(x_0)(x - x_0) + \cdots$$

$$+ \frac{1}{k!} m^{(k-1)}(x_0)(x - x_0)^{k-1} + R_k$$

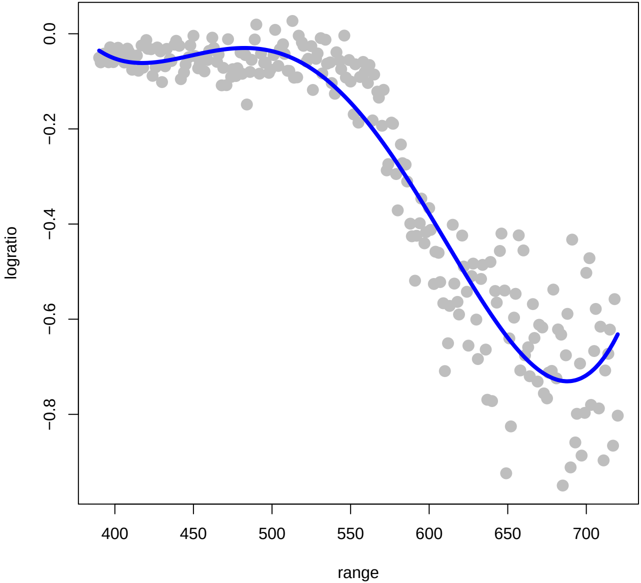
Non-linear regression

- Hence, we can try

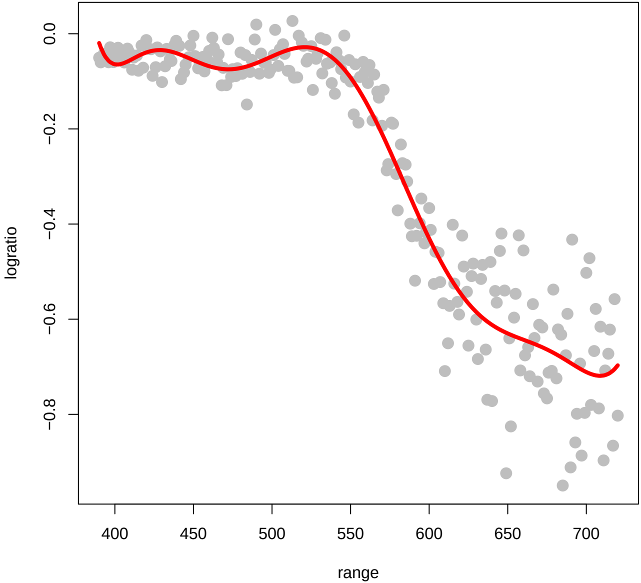
  $$E[Y|X] = \beta_0 + \beta_1 X + \beta_2 X^2 + \ldots + \beta_k X^k$$

- This is a linear model! (**WHY?**)

LIDAR - 4th deg. polynomial



LIDAR - 10th deg. polynomial



More flexible bases

- Consider the (family) of function(s)

$$f_j(x) = \left(x - \kappa_j\right)_+ = \begin{cases} x - \kappa_j & \text{if } x - \kappa_j > 0 \\ 0 & \text{otherwise} \end{cases}$$

  where κ_j are *knots*
- Model

$$E\left[Y | X\right] = \beta_0 + \beta_1 X + \sum_{j=1}^{K} \beta_{j+1} f_j(X)$$

- This is a linear model

More flexible bases

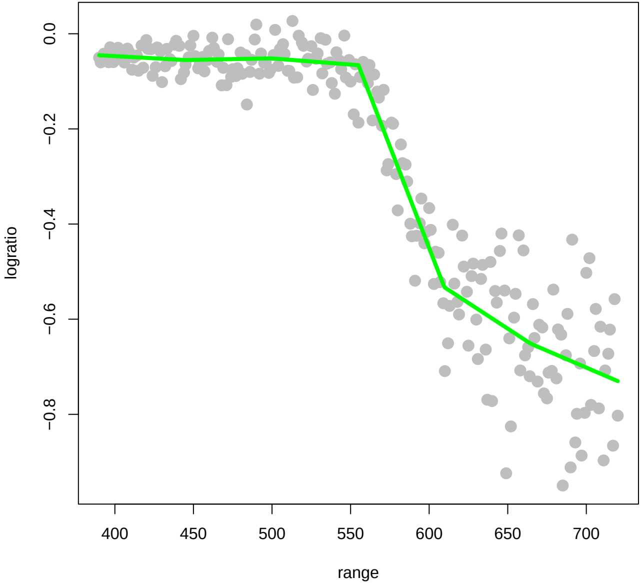
- The knots can be chosen arbitrarily
- It is customary to select them based on the sample

$$\kappa_j = \frac{j}{K+1} \, 100\% \text{ quantile of } x$$

- For example, with $K = 4$:

$$\kappa_1 = 20\% \,, \qquad \kappa_2 = 40\% \,, \qquad \text{etc.}$$

Regression splines, 5 knots



More flexible bases

- Consider a smoother basis

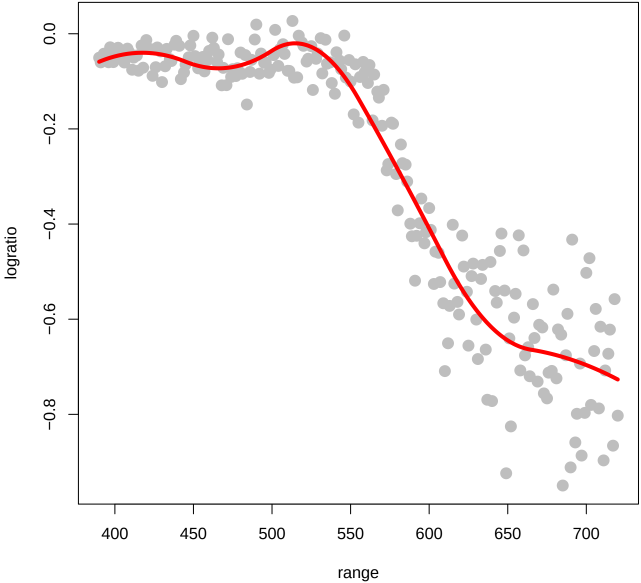
$$f_j(x) = \left(x - \kappa_j\right)_+^2 = \begin{cases} \left(x - \kappa_j\right)^2 & \text{if } x - \kappa_j > 0 \\ 0 & \text{otherwise} \end{cases}$$

  where κ_j, $1 \leq j \leq K$ are *knots*

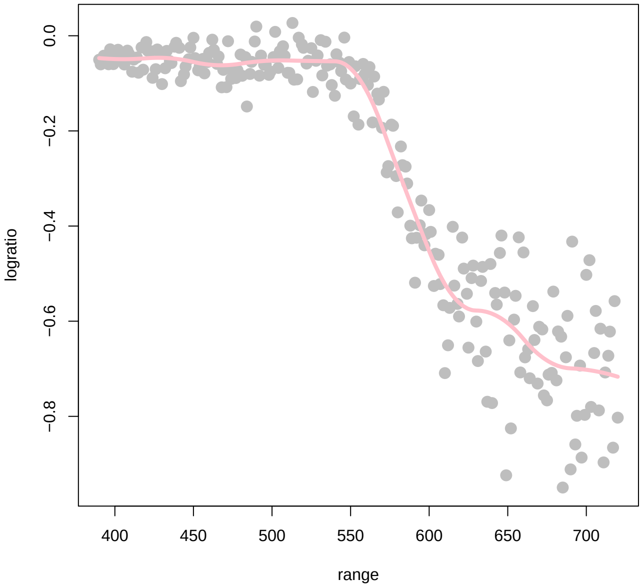
- Model

$$E\left[Y | X\right] = \beta_0 + \beta_1 X + \beta_2 X^2 + \sum_{j=1}^{K} \beta_{j+2} f_j(X)$$

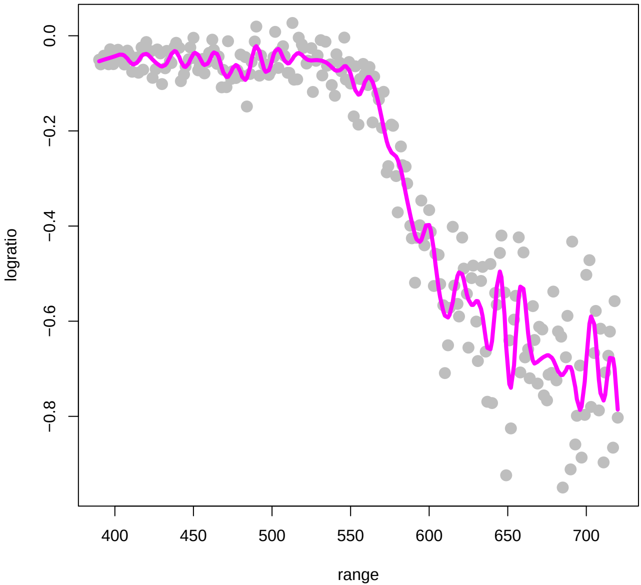
Quadratic splines, 5 knots



Quadratic splines, 10 knots



Quadratic splines, 50 knots



More flexible bases

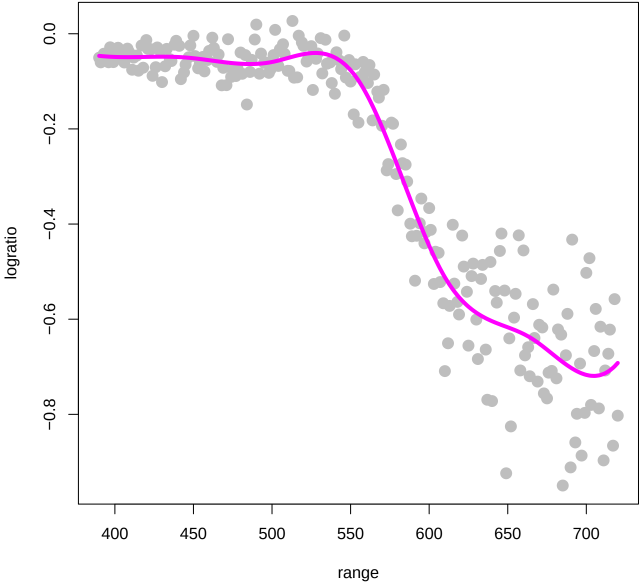
- Cubic splines will be useful

$$f_j(x) = \left(x - \kappa_j\right)_+^3 = \begin{cases} \left(x - \kappa_j\right)^3 & \text{if } x - \kappa_j > 0 \\ 0 & \text{otherwise} \end{cases}$$

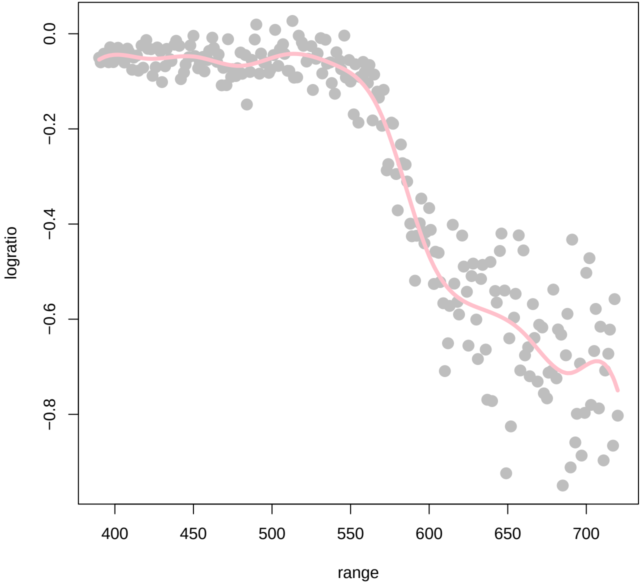
  where κ_j, $1 \le j \le K$ are *knots*
- Model

$$E\left[Y | X\right] = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \sum_{j=1}^{K} \beta_{j+3} f_j(X)$$

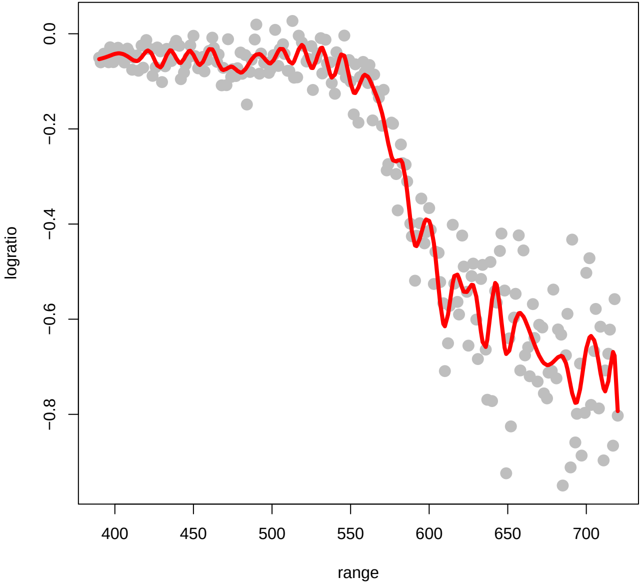
Cubic splines, 5 knots



Cubic splines, 10 knots



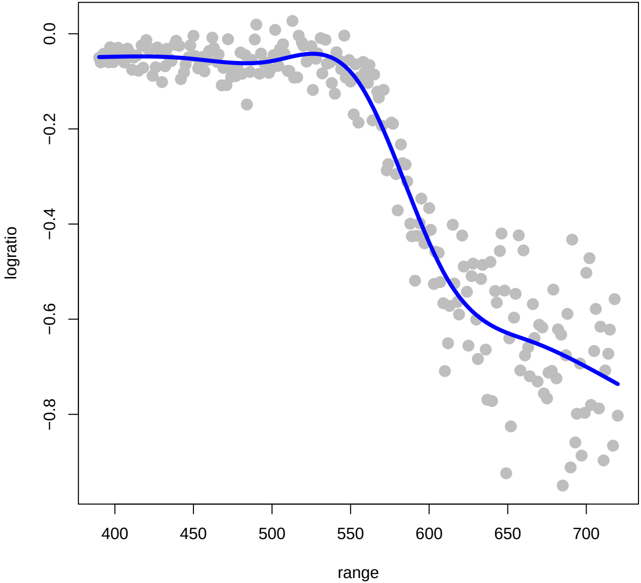
Cubic splines, 50 knots



More flexible bases

- Need to choose number and location of knots
- Need to make them less wiggly at the ends (Natural cubic splines)

Natural cubic spline, 5 knots



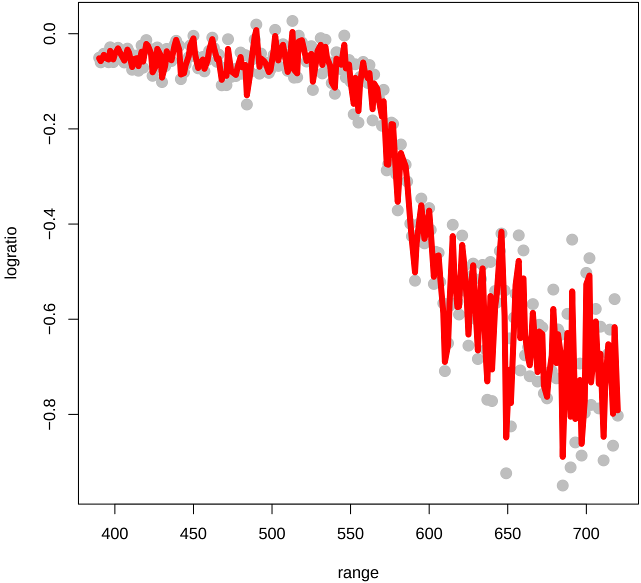
Smoothing splines

- Consider the following problem

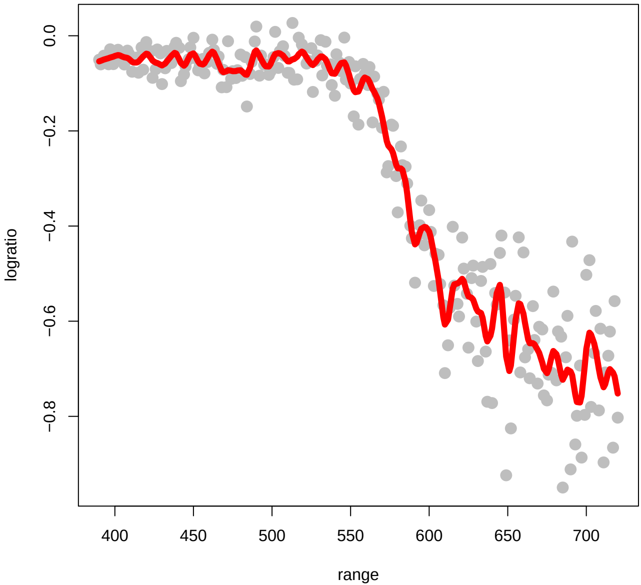
$$\min_f \sum_{i=1}^n (Y_i - f(X_i))^2 + \lambda \int \left(f^{(2)}(t) \right)^2 dt$$

- The solution is a *natural* cubic spline with n knots at $X_1, X_2, \ldots, X_n$.
- *Natural* cubic splines are cubic splines with the restriction that they are linear beyond the boundary knots.

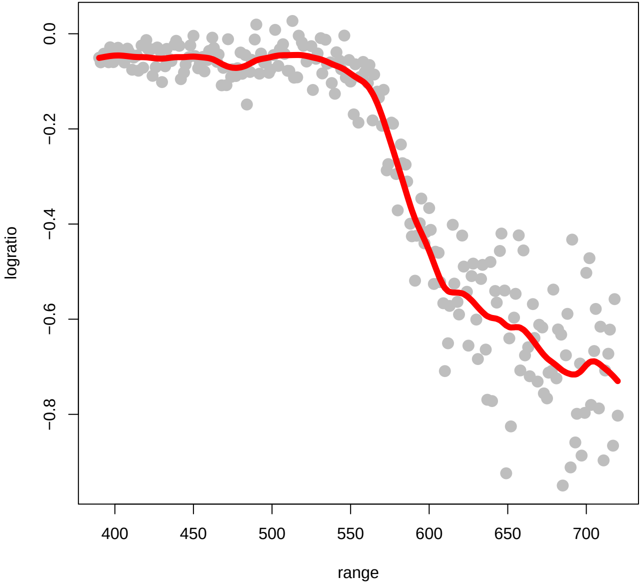
Smoothing spline, $\lambda = 0.20$



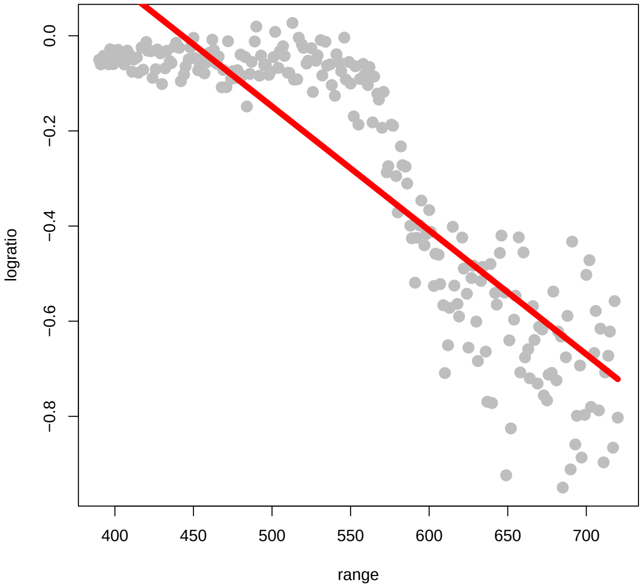
Smoothing spline, $\lambda = 0.50$



Smoothing spline, $\lambda = 0.75$



Smoothing spline, $\lambda = 2.00$



Selecting the penalty parameter

- How do we select λ?
- Minimizing

$$RSS(\lambda) = \sum_{i=1}^{n} \left(Y_i - \mathbf{X}_i' \beta_\lambda \right)^2$$

is not a good idea...

Selecting the penalty parameter

- Cross-validation: consider

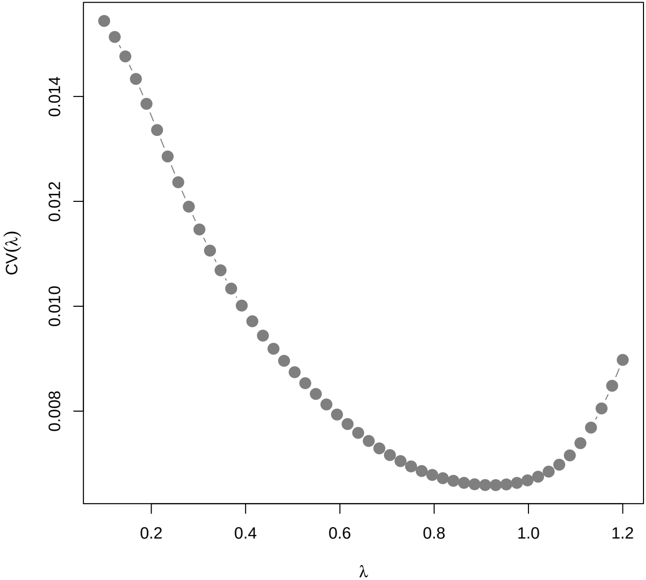
$$CV(\lambda) = \sum_{i=1}^{n} \left(Y_i - \mathbf{X}_i' \beta_\lambda^{(-i)} \right)^2$$

where $\beta_\lambda^{(-i)}$ is the fit without using the point (Y_i, X_i)

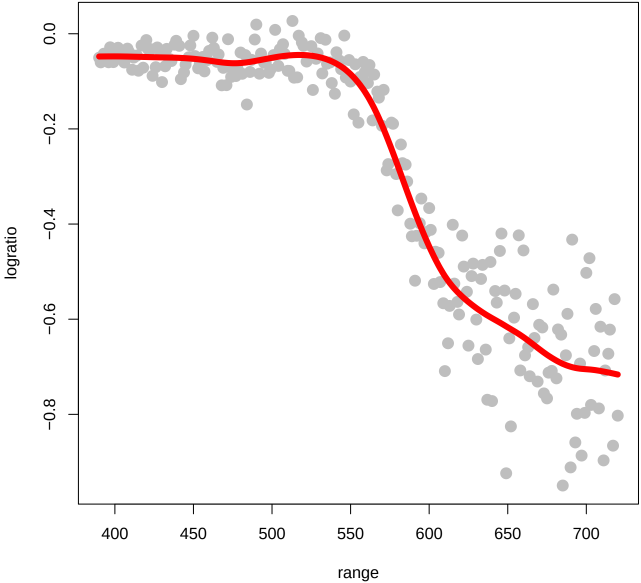
and choose a value λ_0 such that

$$CV(\lambda_0) \ \leq \ CV(\lambda) \quad \forall \, \lambda \geq 0$$

5-fold CV, smoothing spline



Optimal fit via 5-fold CV



Selecting the penalty parameter

- Computing leave-one-out CV

$$CV(\lambda) = \sum_{i=1}^{n} \left(Y_i - \mathbf{X}_i' \beta_\lambda^{(-i)} \right)^2$$

We might need to re-fit the model n times

Selecting the penalty parameter

- For some smoothers and models this is not necessary. For many linear smoothers $\hat{\mathbf{Y}} = \mathbf{S}_\lambda \mathbf{Y}$ we have

$$\hat{\mathbf{Y}}_r = \sum_{i=1}^{n} \mathbf{S}_{\lambda,r,i} Y_i \qquad r = 1, \ldots, n$$

and then

$$\hat{\mathbf{Y}}_r^{(-r)} = \frac{\sum_{i \neq r} \mathbf{S}_{\lambda,r,i} Y_i}{\sum_{i \neq r} \mathbf{S}_{\lambda,r,i}}$$

Selecting the penalty parameter

- Furthermore

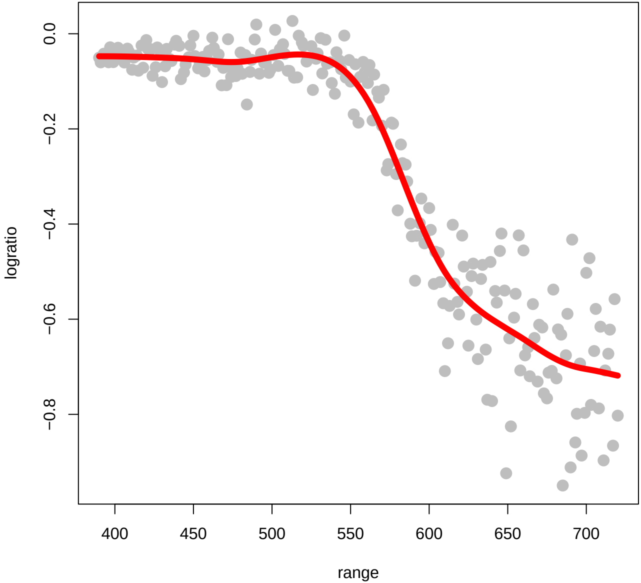
$$\mathbf{S}_\lambda \mathbf{1} = \mathbf{1}$$

thus

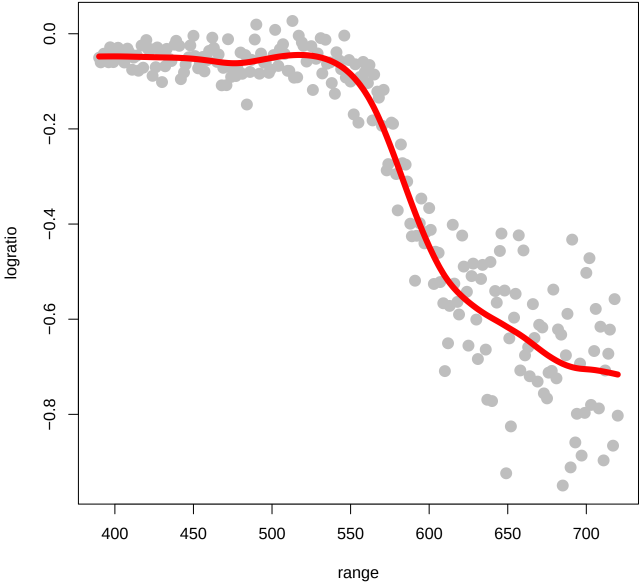
$$\hat{\mathbf{Y}}_r^{(-r)} = \frac{\sum_{i \neq r} \mathbf{S}_{\lambda,r,i} Y_i}{1 - \mathbf{S}_{\lambda,r,r}}$$

$$CV(\lambda) = \sum_{i=1}^{n} \left(\frac{Y_i - \hat{\mathbf{Y}}_i}{1 - \mathbf{S}_{\lambda,i,i}} \right)^2$$

Optimal fit via leave-1-out CV



Compare with 5-fold CV optimal



Selecting the penalty parameter

- Computing $\mathbf{S}_{\lambda,i,i}$, $i = 1, \ldots, n$ can be demanding

$$GCV(\lambda) = \sum_{i=1}^{n} \left(\frac{Y_i - \hat{\mathbf{Y}}_i}{1 - \text{tr}(\mathbf{S}_\lambda)/n} \right)^2 =$$

$$= \frac{\sum_{i=1}^{n} \left(Y_i - \hat{\mathbf{Y}}_i \right)^2}{\left(1 - \text{tr}(\mathbf{S}_\lambda)/n\right)^2}$$