Lecture 4: Regression

DD2421

Atsuto Maki

Autumn, 2017

Part I: we will visit

- Function approximation
- Linear Regression / Least Squares
  - Robust regression (RANSAC to handle outliers)
- k-NN Regression

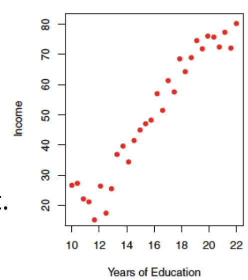
Regression => Real-valued output

Function approximation

- How do we fit this dataset D?

$$D = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$$

of N pairs of inputs x_i and targets $y_i \in R$.
D can be measurements in an experiment.



- Task of regression:
  to predict target associated to any arbitrary new input

Note: Here we have a single *input feature*, but inputs to regression tasks are often vectors $\mathbf{x}$ of *multiple input features*.

Linear Regression (parametric)

Linear regression tries to estimate the function f and predict the output by

$$\hat{f}(x) = \sum_{i=0}^{d} w_i x_i = w^T x$$

How to measure the error:

- To see how well $\hat{f}(x)$ approximates $f(x)$, square error is used: $(\hat{f}(x) - f(x))^2$
- Mean Square Error: $E_{in}(\hat{f}) = \frac{1}{N}\sum_{n=1}^{N}(\hat{f}(x_n) - y_n)^2$
  (in-sample)

Minimizing in-sample MSE

E_{in} can be expressed as:

$$E_{in}(w) = \frac{1}{N}\sum_{n=1}^{N}(w^T x_n - y_n)^2 = \frac{1}{N}\left\|Xw - Y\right\|^2$$

where

$$X = \begin{bmatrix} x_1^T \\ \vdots \\ x_N^T \end{bmatrix}, \quad Y = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}$$

We want to compute the parameters w that minimize E_{in}.

Residual sum of squares (RSS)

The sum of squared errors is a convex function of w

$$E_{in}(w) = \left\|Xw - Y\right\|^2$$

The gradient with respect to the weights is:

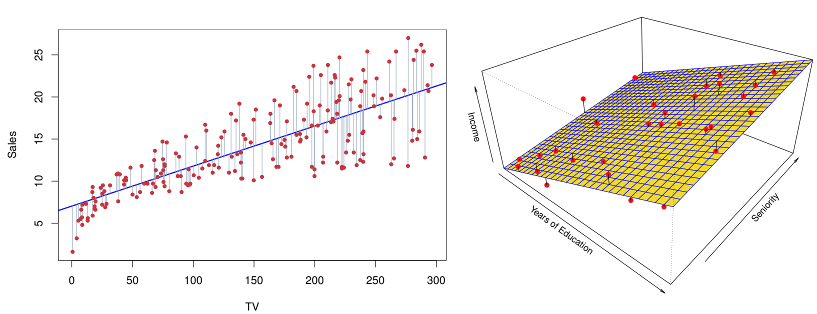
$$\frac{\partial}{\partial w} E_{in}(w) = 2X^T\left(Xw - Y\right)$$

The weight vector that sets the gradient to zero minimizes the errors

$$X^T X w = X^T Y$$

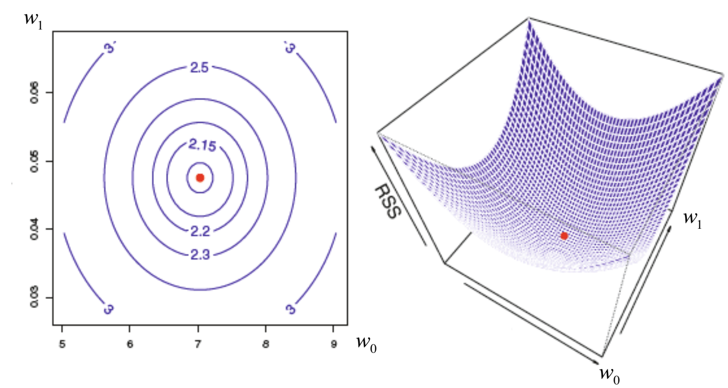
$$w = \left(X^T X\right)^{-1} X^T Y$$

Examples of least squares fit



Figures from An Introduction to Statistical Learning (G. James et al.)

Examples of plots of RSS



Figures adapted from An Introduction to Statistical Learning (G. James et al.)

RANSAC: RANdom SAmpling Consensus

Robust regression

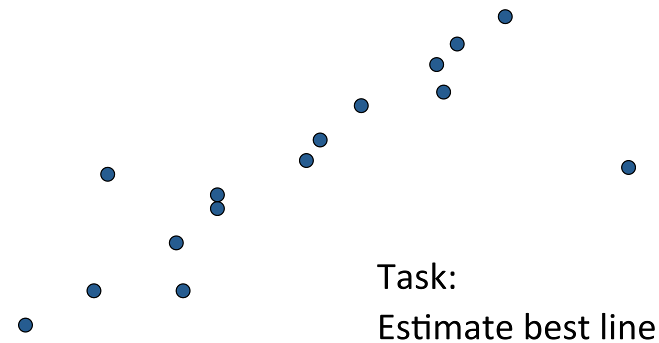
Repeat M times:

- Sample two points to estimate the line.
- Calculate the number of inliers or posterior likelihood for relation.
- Choose relation to maximize the number of inliers.

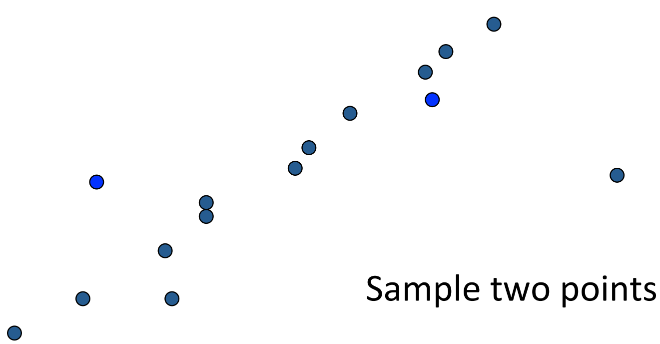
(Last two in *Least Median of Square* (LMedS))

- Calculate error of all data.
- Choose relation to minimize median of errors.

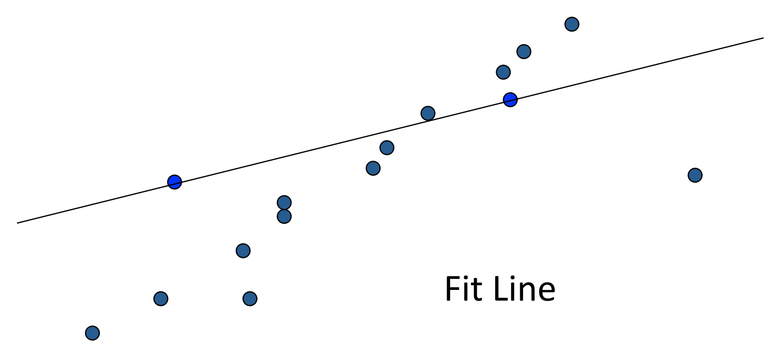
RANSAC line fitting example



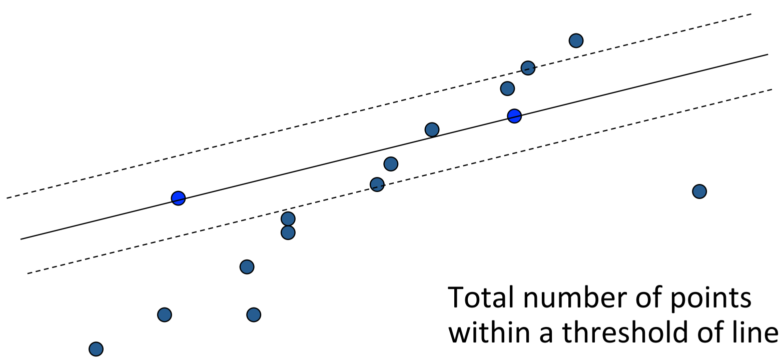
RANSAC line fitting example



RANSAC line fitting example



RANSAC line fitting example

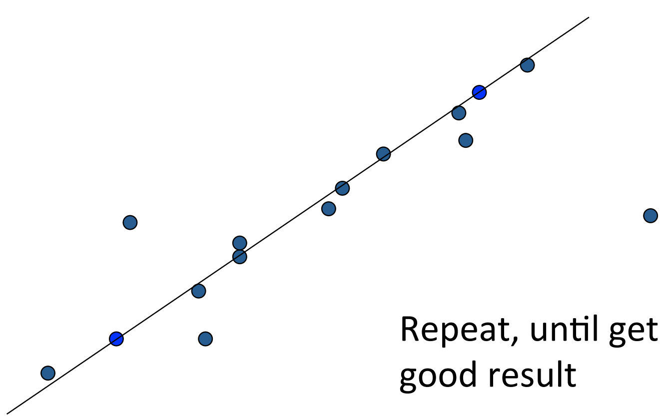


Total number of points within a threshold of line.

RANSAC line fitting example

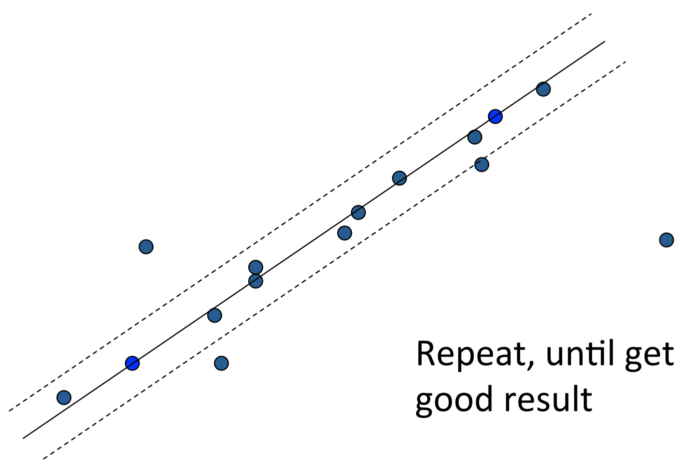
Repeat, until get a good result

RANSAC line fitting example



Repeat, until get a good result

RANSAC line fitting example



Repeat, until get a good result

RANSAC: RANdom SAmpling Consensus

<u>Objective</u>

Robust fit of model to data set S which contains outliers

<u>Algorithm</u>

(i) Randomly select a (minimum number of) sample of s data points from S and instantiate the model from this subset.

(ii) Determine the set of data points S_i which are within a distance threshold t of the model. The set S_i is the consensus set of samples and defines the inliers of S.

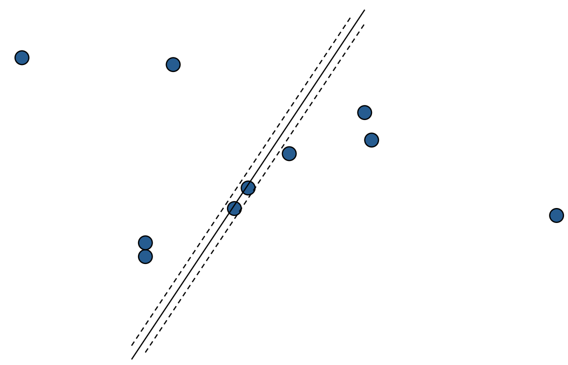
(iii) If the subset of S_i is greater than some threshold T, re-estimate the model using all the points in S_i and terminate

(iv) If the size of S_i is less than T, select a new subset and repeat the above.

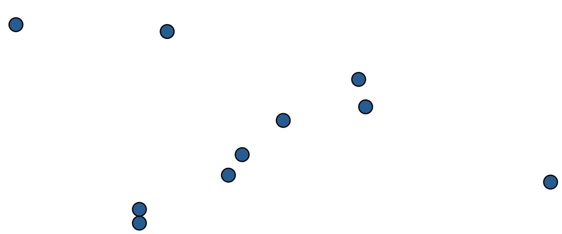
(v) After N trials the largest consensus set S_i is selected, and the model is re-estimated using all the points in the subset S_i

(in Hartley and Zisserman, adapted from Fischler '81)

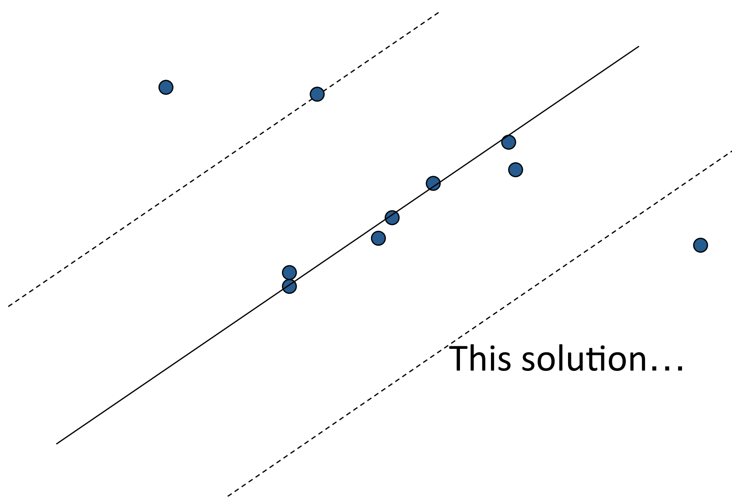
Problem with RANSAC; threshold too low-no support



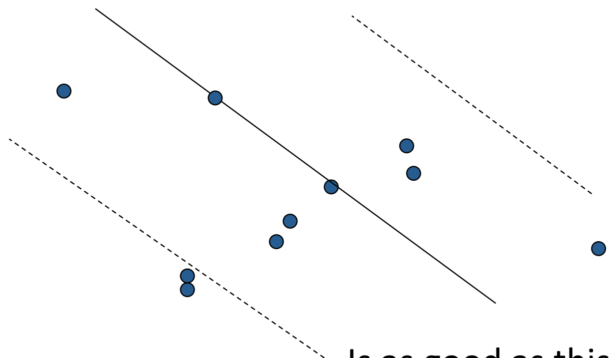
Problem with RANSAC; threshold too high



Problem with RANSAC; threshold too high



Problem with RANSAC; threshold too high

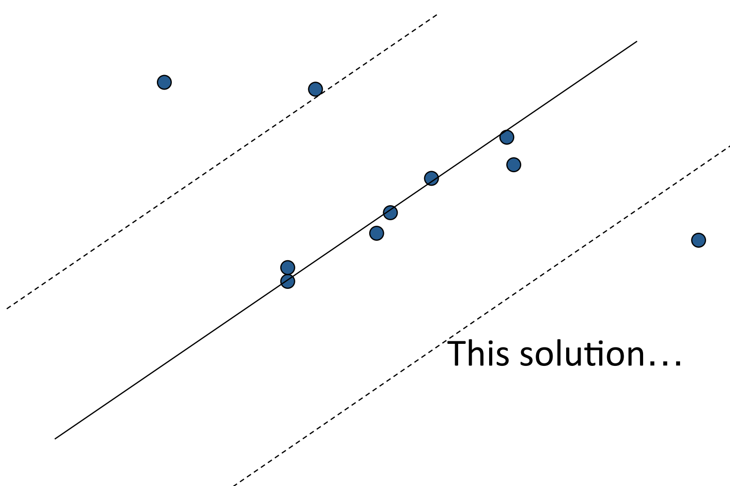


Is as good as this solution

Cost function

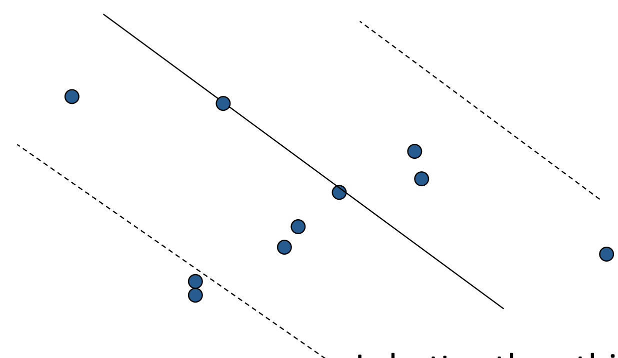
- RANSAC can be vulnerable to the correct choice of the threshold:
  - Too large all hypotheses are ranked equally.
  - Too small leads to an unstable fit.
- The interesting thing is that the same strategy can be followed with any modification of the cost function.

MLESAC



This solution…

MLESAC



Is better than this solution

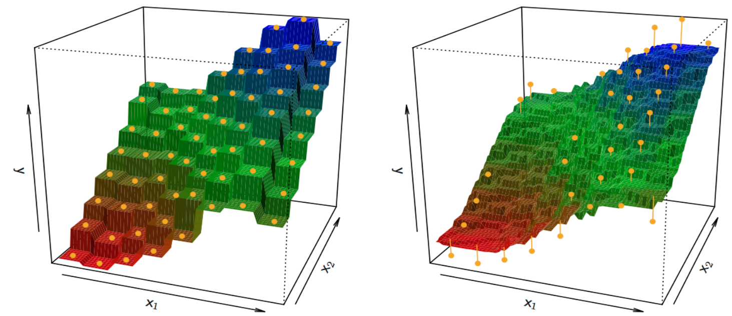
k-NN Regression (non-parametric)

- Similar to the k-NN classifier
- To regress Y for a given value of X, consider k closest points to X in training data and take the average of the responses.

$$f(x) = \frac{1}{k} \sum_{x_i \in N_i} y_i$$

- Larger values of k provide a smoother and less variable fit (lower variance!)

Example plots of $\hat{f}(x)$ with k-NN regression (2d)

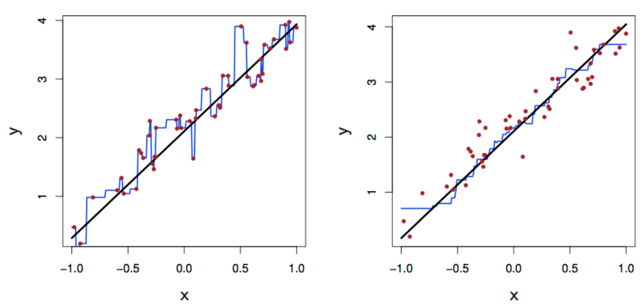


k = 1 $\quad$ k = 9

In higher dimensions k-NN often preforms worse than linear regression.

Figures from An Introduction to Statistical Learning (G. James et al.)

Example plots of $\hat{f}(x)$ with k-NN regression (1d)



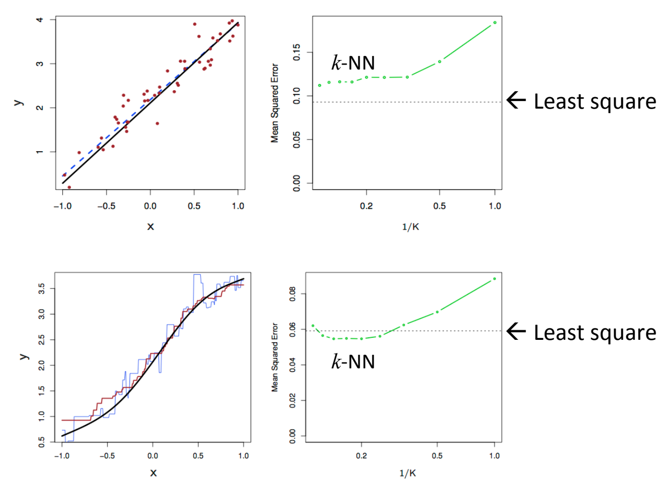
k = 1 $\quad$ k = 9

Black: function
Red: observed
Blue: estimated

Figures from An Introduction to Statistical Learning (G. James et al.)

k-NN vs. Linear Regression (MSE)

k = 1 (blue)
k = 9 (red)



Figures from An Introduction to Statistical Learning (G. James et al.)

Parametric or Non-parametric?

- How will those compare in what setting?
  - If the parametric form is close to the true form of f, the *parametric* approach will outperform the *non-parametric*
  - As a general rule, *parametric* methods will tend to outperform *non-parametric* when there is a small number of observations per predictor (i.e. in a high dimension).
  - Interpretability stand point: Linear regression preferred to KNN if the test MSEs are similar or slightly lower.

Part II: we will visit

- Linear regression + regularization
  - Ridge regression
  - The Lasso (a more recent alternative)

Motivation for shrinkage

- Interpretability
  - Among a large number of variables X in the model there are generally many that have little (or no) effect on Y
  - Leaving these variables in the model makes it harder to see the big picture, i.e. the effect of the "important variables"
  - Would be easier to interpret the model by removing unimportant variables (setting the coefficients to zero)

Sample problem: The Credit dataset

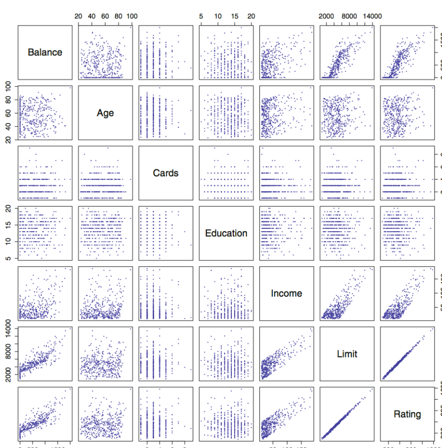


FIGURE 3.6. *The* Credit *data set contains information about* balance, age, cards, education, income, limit, *and* rating *for a number of potential customers.*

Figure from An Introduction to Statistical Learning (G. James et al.)

Ridge regression

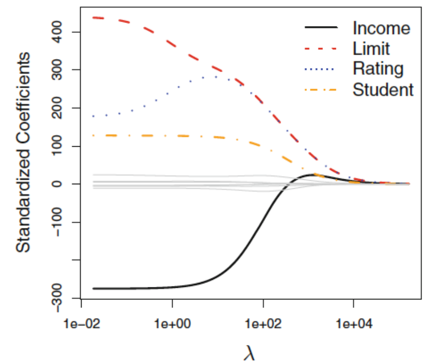
Similar to least squares but minimizes different quantity:

$$RSS + \lambda \sum_{i=1}^{d} w_i^2$$

The second term is called shrinkage penalty

- Shrinkage penalty: small when w_i are close to zero
- The parameter λ : controls the relative impact of the two terms, the selection is critical!

Ridge regression coefficients

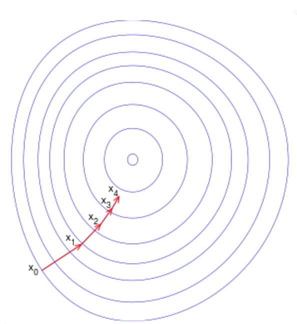


As λ increases, the standardized coefficients shrinks towards zero (but not exactly forced to zero).

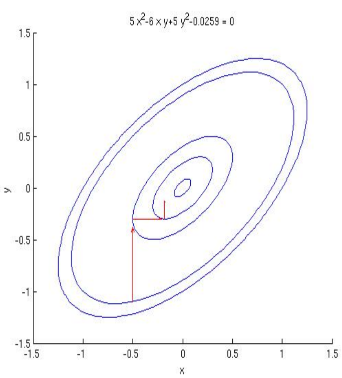
Figure from An Introduction to Statistical Learning (G. James et al.)

Approaches to parameter estimations

- Gradient decent

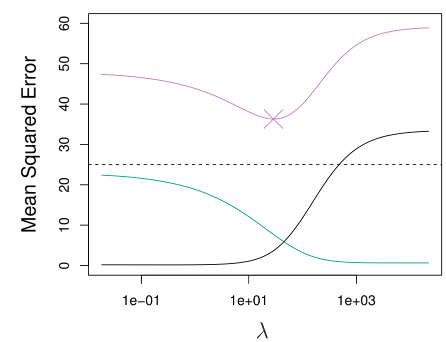


- Coordinate decent



Ridge Regression Bias/Variance

- Green: Variance
- Black: Bias
- Purple: MSE



Increased λ decreases variance while increasing bias

Figure from An Introduction to Statistical Learning (G. James et al.)

The Lasso
(Least Absolute Shrinkage and Selection Operator)

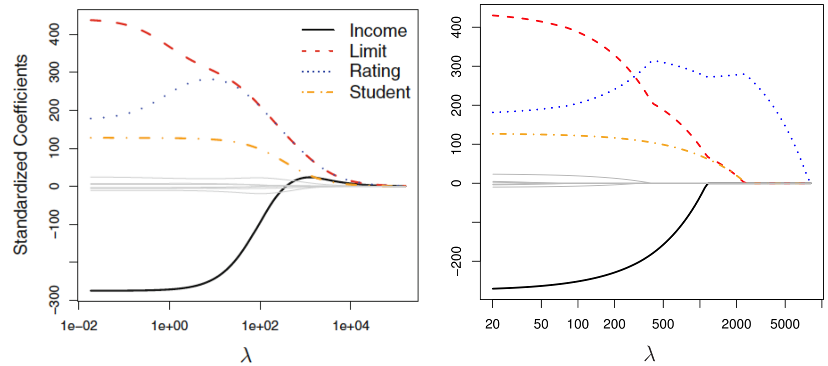
Similar to ridge regression but with slightly different term:

$$RSS + \lambda \sum_{i=1}^{d} |w_i|$$

The shrinkage penalty is now replaced by l_1 norm

- Ridge regression: it includes all features in the final model, making it harder to interpret – its drawback
- The lasso could be proven mathematically that some coefficients end up being set to exactly zero
  - variable selection
  - yielding sparse model

Comparison of estimated coefficients



Ridge regression The Lasso

Figures from An Introduction to Statistical Learning (G. James et al.)

Another formulations

For every value of λ there is some s such that the equations will give the same coefficient estimates:

- Ridge regression: Mimimizing $RSS + \lambda \sum_{i=1}^{d} w_i^2$

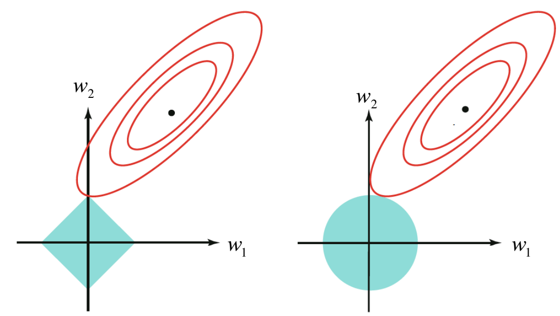
  Mimimizing $RSS, sub.to \sum_{i=1}^{d} w_i^2 \leq s$

- Lasso: Mimimizing $RSS + \lambda \sum_{i=1}^{d} |w_i|$

  $RSS, sub.to \sum_{i=1}^{d} |w_i| \leq s$

The variable selection property

The coefficient estimates: the first point where an ellipse contacts the constraint region as it expands.



The solid blue areas are the constraint regions for

Left: the Lasso Right: Ridge regression

Figures from An Introduction to Statistical Learning (G. James et al.)