Regression

Agenda

- Problem definition
- Loss function
- Regularization
- Likelihood–based approaches
- Representation

Regression

- Predict **real-valued** response variable y by predictors $\mathbf{x} = \left[x_1, \ldots, x_p\right]^T$
- x_j can be
  - Quantitative variables
  - Transformations of variables, e.g., $x_2 = \log(x_1)$
  - Qualitative variables (factors)
  - Interaction terms, e.g., $x_3 = x_1 x_2$
  - etc.

Linear regression

- Assume that the decision function is a linear function; i.e.,

$$f(\mathbf{x}) = \beta_0 + \sum_{j=1}^{p} \beta_j x_j$$

- The simplest regression model
- Easy to understand

Pros

- Simple and stable
- Interpretable: β_j is the effect of x_j then the other predictors are fixed
- If the variables are independent with each other, the estimated coefficients will be consistent even if the model is mis–specified

Cons

- Too simple to be true
- Statistical inferences may be incorrect if the model is misspecified (model selection always works)

Least squares estimation

- The signs of errors are usually not important.
- Quadratic ($\ell 2$) loss is the most commonly used; i.e., $L\big(y, f(\mathbf{x})\big) = \big(y - f(\mathbf{x})\big)^2 = \xi^2$
- Equation (2) becomes

$$\min_{\beta_j' s} \sum_{i=1}^{n} \big(y_i - \beta_0 - \beta_1 x_{i1} - \ldots \beta_p x_{ip}\big)^2 \qquad (4)$$

Loss functions

- $\ell 1$ loss:

$$L(\xi) = |\xi|$$

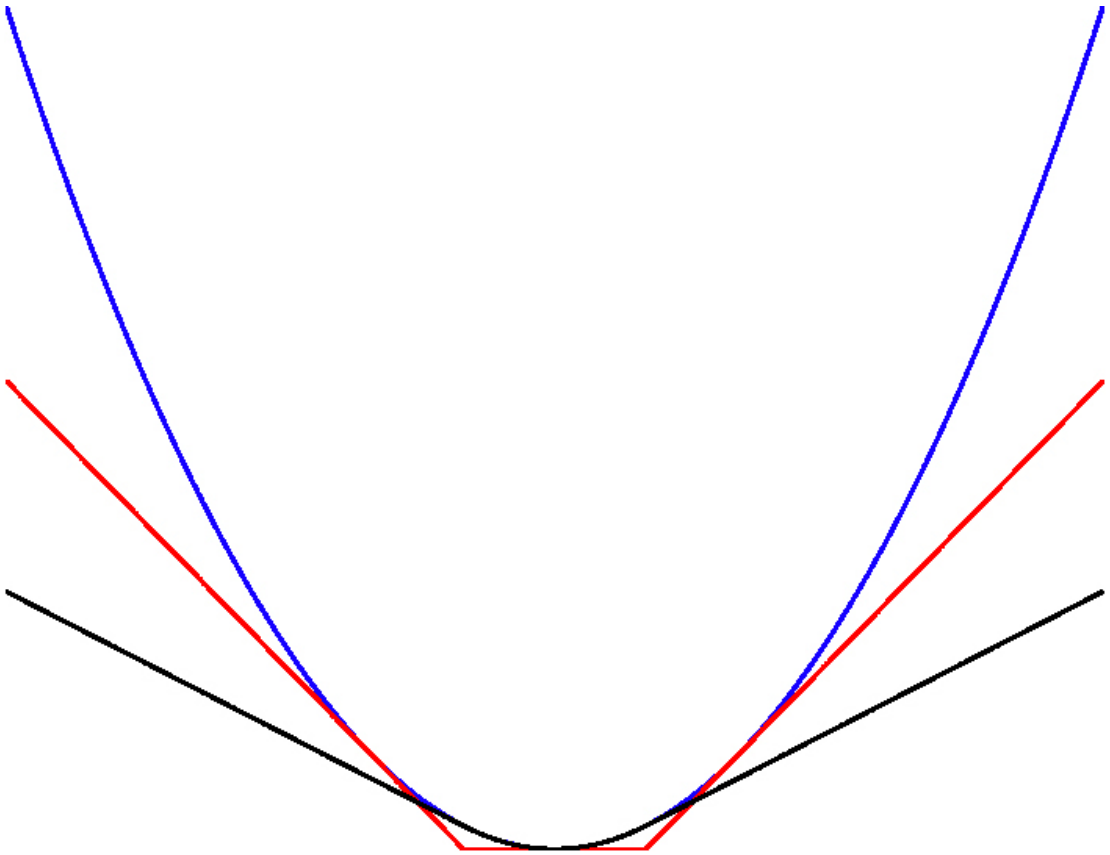
- Huber loss:

$$L_\epsilon(\xi) = \begin{cases} \frac{1}{2}\xi^2 & \text{if } |\xi| \leq \epsilon \\ \epsilon\left(|\xi| - \frac{1}{2}\epsilon\right) & \text{otherwise} \end{cases}$$

- ϵ–sensitive loss:

$$L_\epsilon(\xi) = \begin{cases} 0 & \text{if } |\xi| \leq \epsilon \\ |\xi| - \epsilon & \text{if } |\xi| > \epsilon \end{cases}$$

Loss functions



blue: quadratic
black: huber
red: ϵ-sensitive

Regularizations

- Popular regularizations
  - $\|f\| = \sum_{j=1}^{p} \beta_j^2$
    (ridge, $\ell 2$)
  - $\|f\| = \sum_{j=1}^{p} |\beta_j|$
    (Lasso, $\ell 1$)
  - $\|f\| = (1-\alpha) \sum_{j=1}^{p} \beta_j^2 + \alpha \sum_{j=1}^{p} |\beta_j|$
    (elastic–net, $\ell 1$_$\ell 2$)

Ridge

- If $\ell 2$ loss is utilized, equation (3) becomes

$$\min_{\beta_j' s} \sum_{i=1}^{n} \left(y_i - \beta_0 - \beta_1 x_{i1} - \ldots \beta_p x_{ip}\right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \qquad (5)$$

- Pros
  - works well with correlated predictors (multicollinearity)
  - biased but smaller variance and smaller expected loss

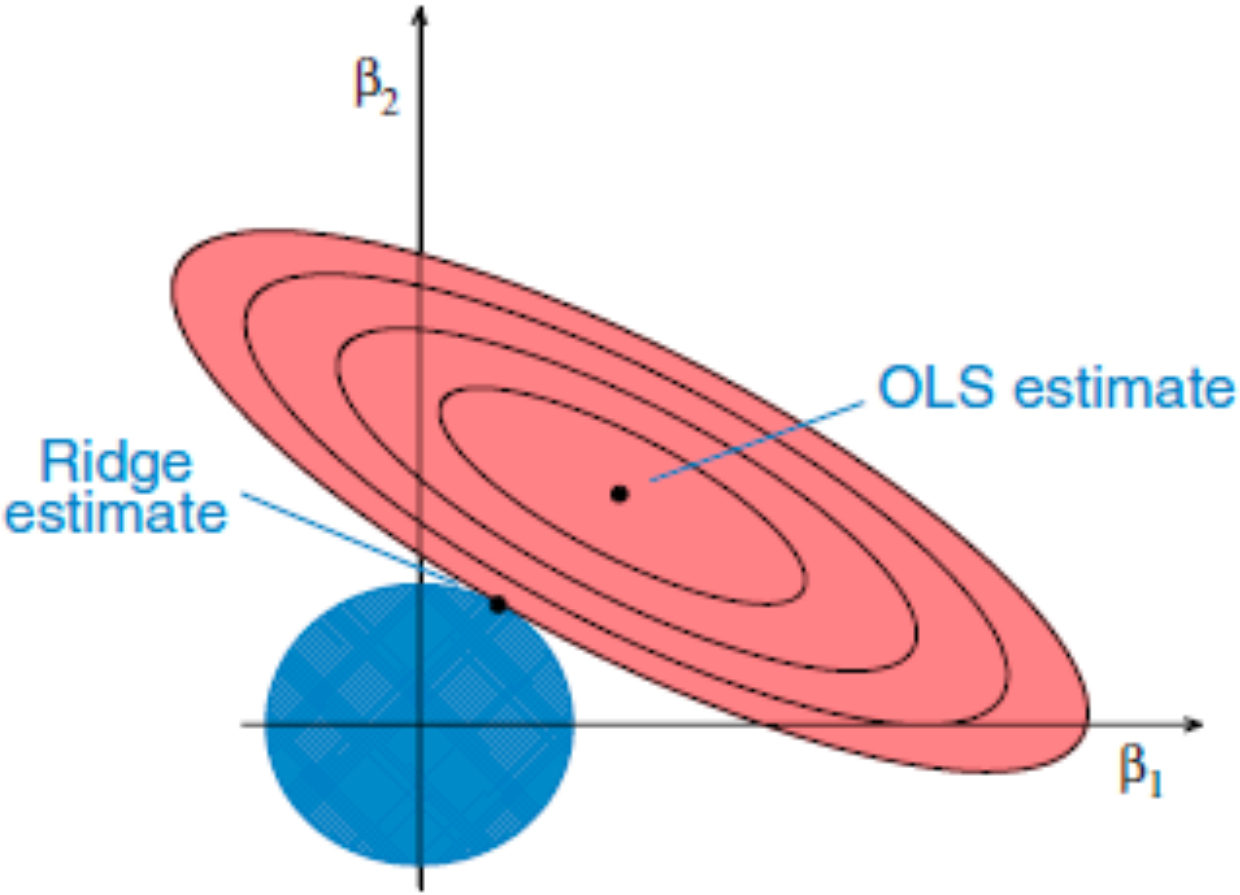
Where are regularizations from?

- Equation (5) comes from the lagrangian of:

$$\min_{\beta_j's} \sum_{i=1}^{n} \left(y_i - \beta_0 - \beta_1 x_{i1} - \ldots \beta_p x_{ip}\right)^2$$

$$\text{subject to} \sum_{j=1}^{p} \beta_j^2 \leq C$$

Ridge regression



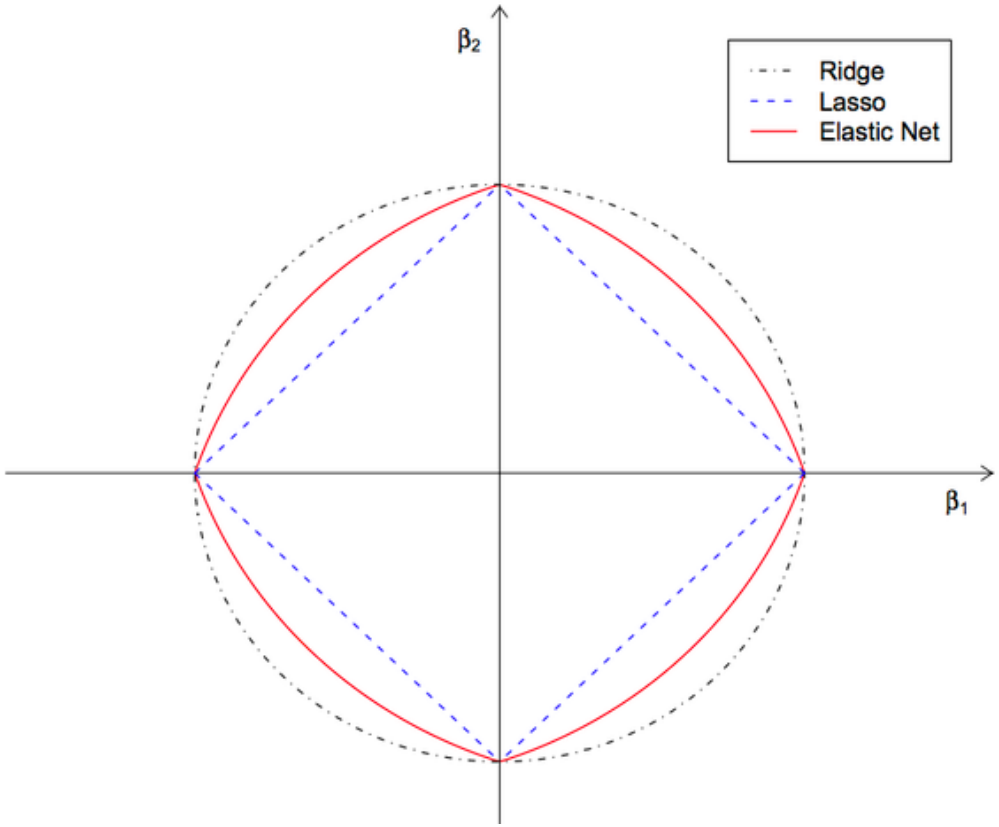
Lasso

- Pros
  - automatic variable selection with model consistency
  - biased but sign consistent
- Cons
  - works poorly with correlated/grouped predictors

Elastic–net

- Pros
  - inherit the pros from both ridge and LASSO
  - encourage grouping effect in the presence of highly correlated predictors
- Cons
  - introduce an additional hyperparameter
  - more biased results

Regularizations



Likelihood approach

- Let $y_i \overset{iid}{\sim} N\big(f(\mathbf{x}_i), \sigma^2\big)$, the negative log–likelihood becomes

$$\frac{n}{2}\log(2\pi) + \frac{n}{2}\log\sigma^2 + \frac{1}{2\sigma^2}\sum_{i=1}^{n}\big(y_i - f(\mathbf{x}_i)\big)^2 \qquad (6)$$

- Notice that minimizing (6) over $(\beta_0, \boldsymbol{\beta})$ is independent of the nuisance parameter σ^2; thus minimizing (6) is equivalent to minimizing (4).

Bayesian approach

- If we further assume that $\beta \sim N(0, \tau^2)$, the negative log–posterior probability becomes

$$C + \frac{1}{2\sigma^2} \sum_{i=1}^{n} \left(y_i - f(\mathbf{x}_i)\right)^2 + \frac{1}{2\tau^2} \sum_{j=1}^{p} \beta_j^2 \qquad (7)$$

- Minimizing (7) is now equivalent to minimizing (5)

Linear regression ≠ linear line regression

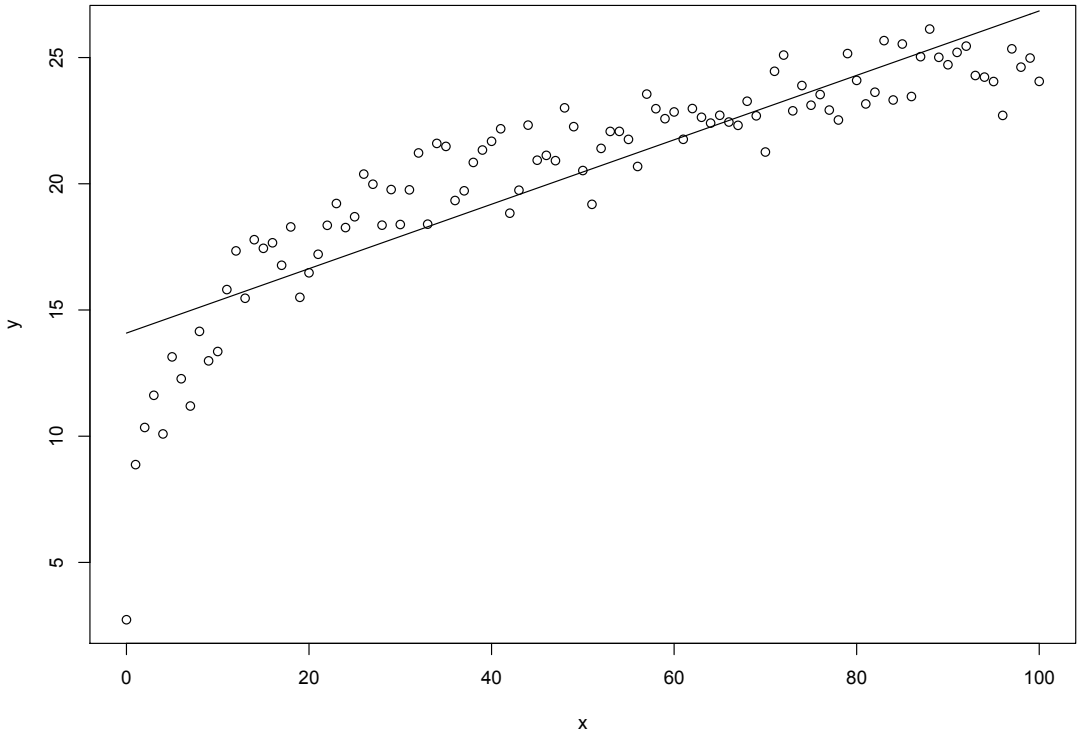
- In linear regressions we mean that f is a linear combination of x_j's.
- Recall that x_j can be transformation of $x_{j'}$'s

Polynomial regression

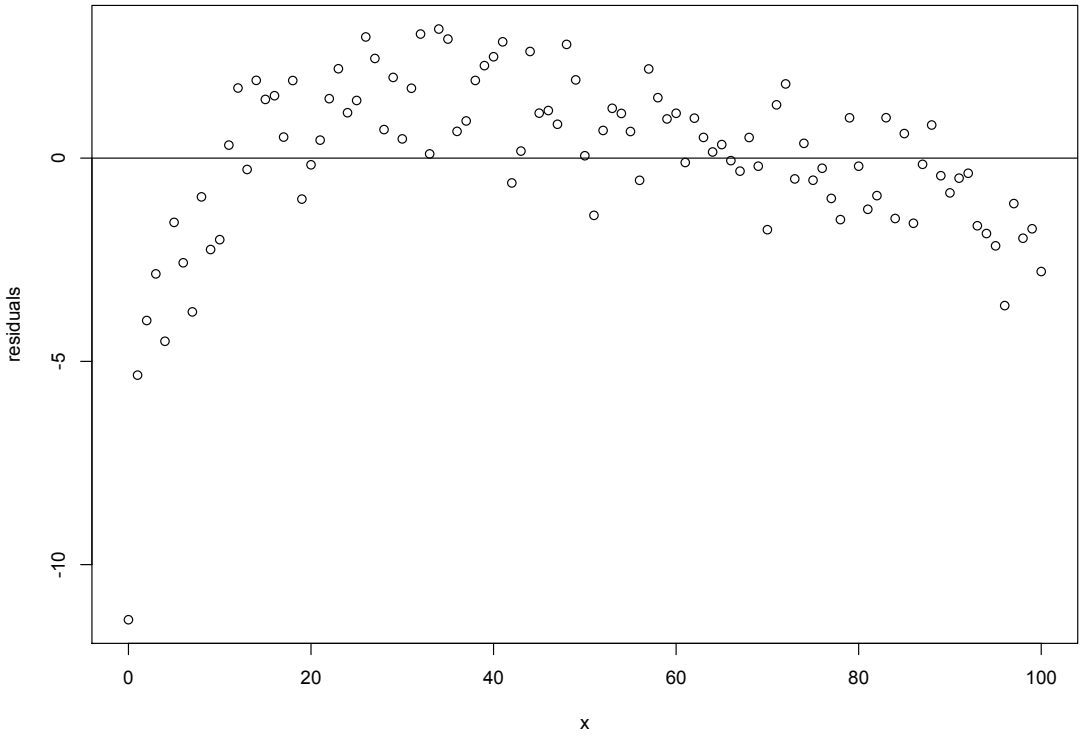
- We can specify the decision function as a polynomial of order p :

$$f(x) = \beta_0 + \sum_{k=1}^{p} \beta_k x^k$$

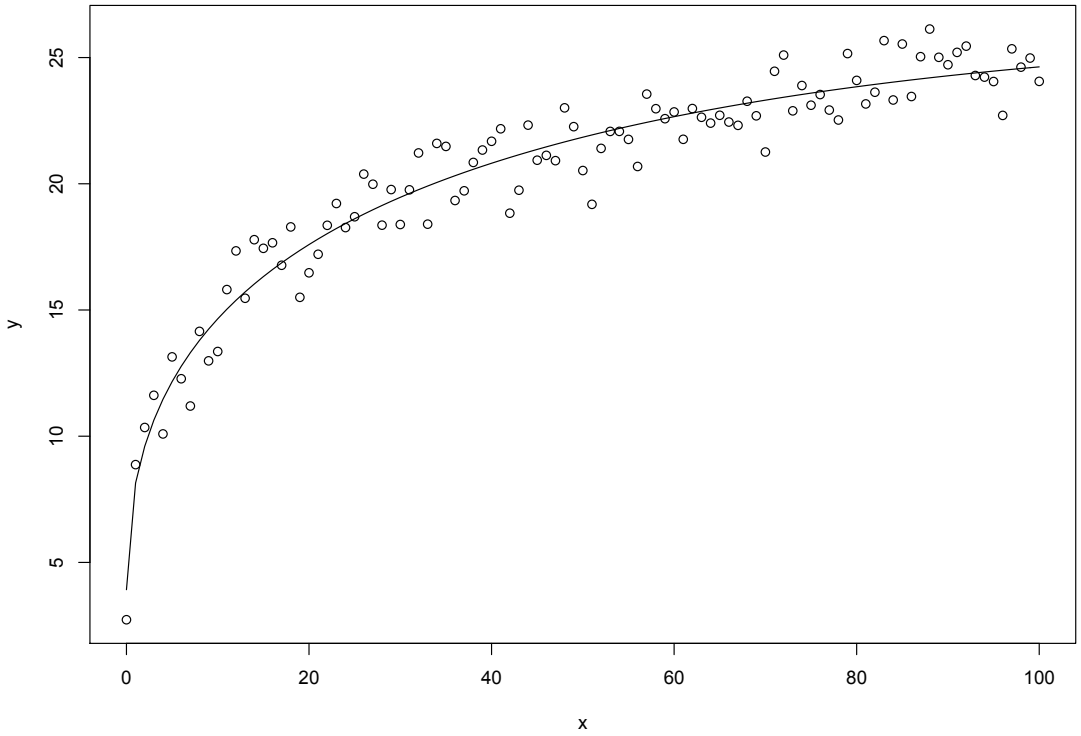
Underfitting



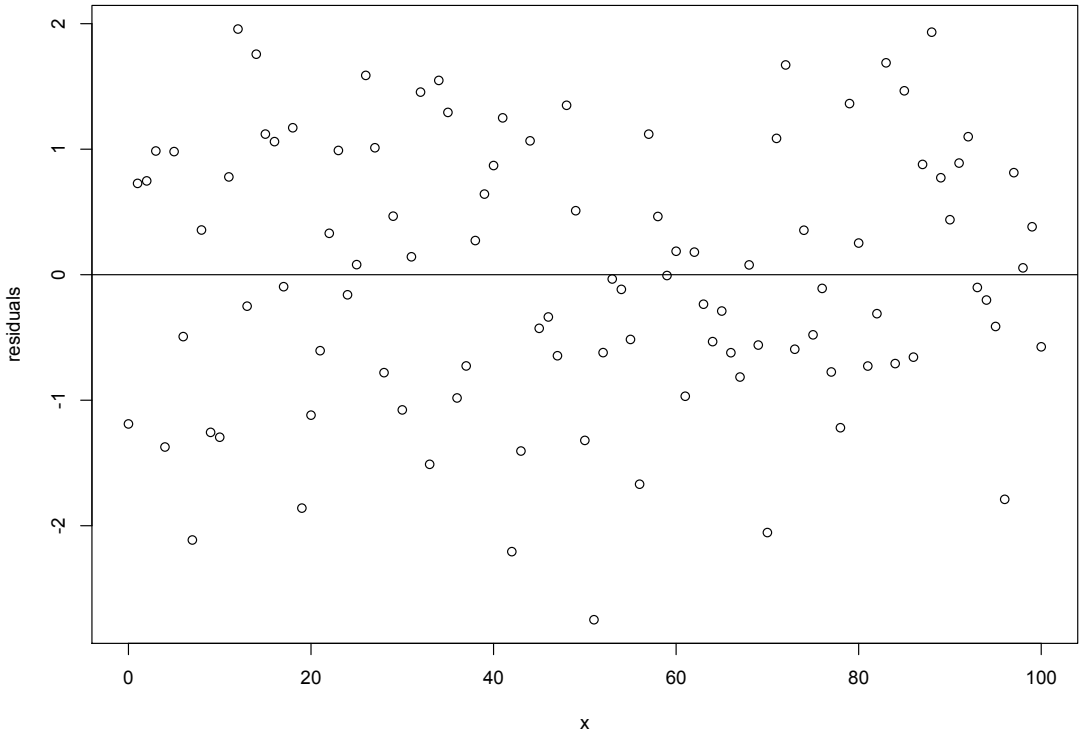
Detect underfitting by residual plot



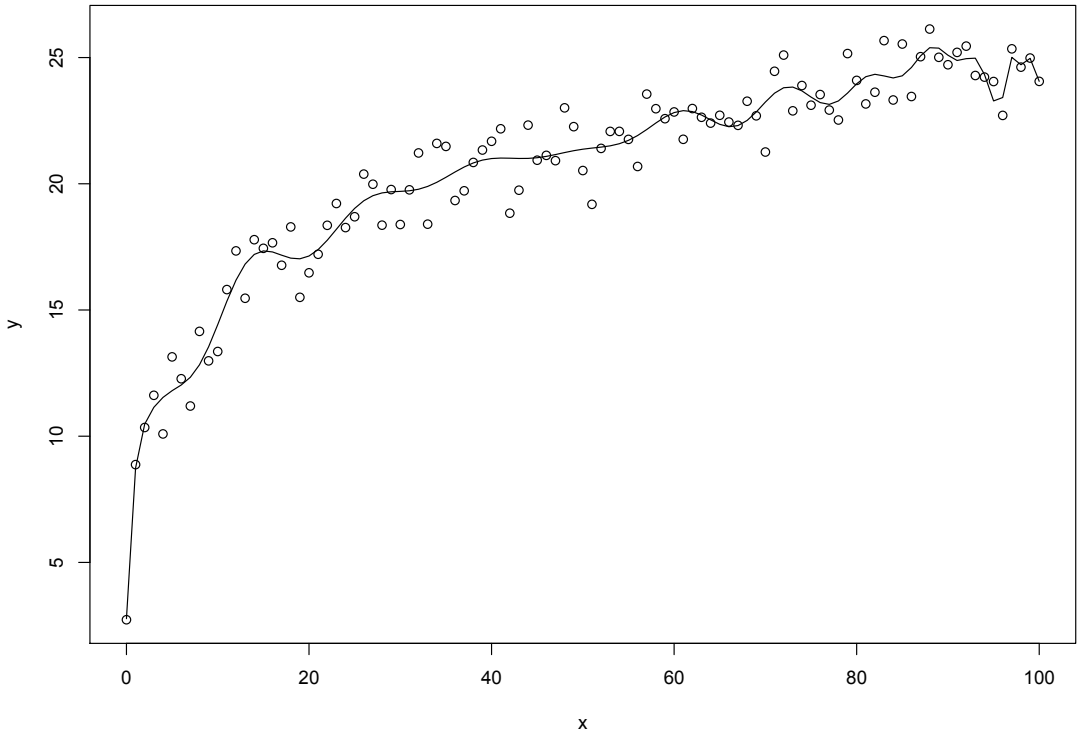
Just right …



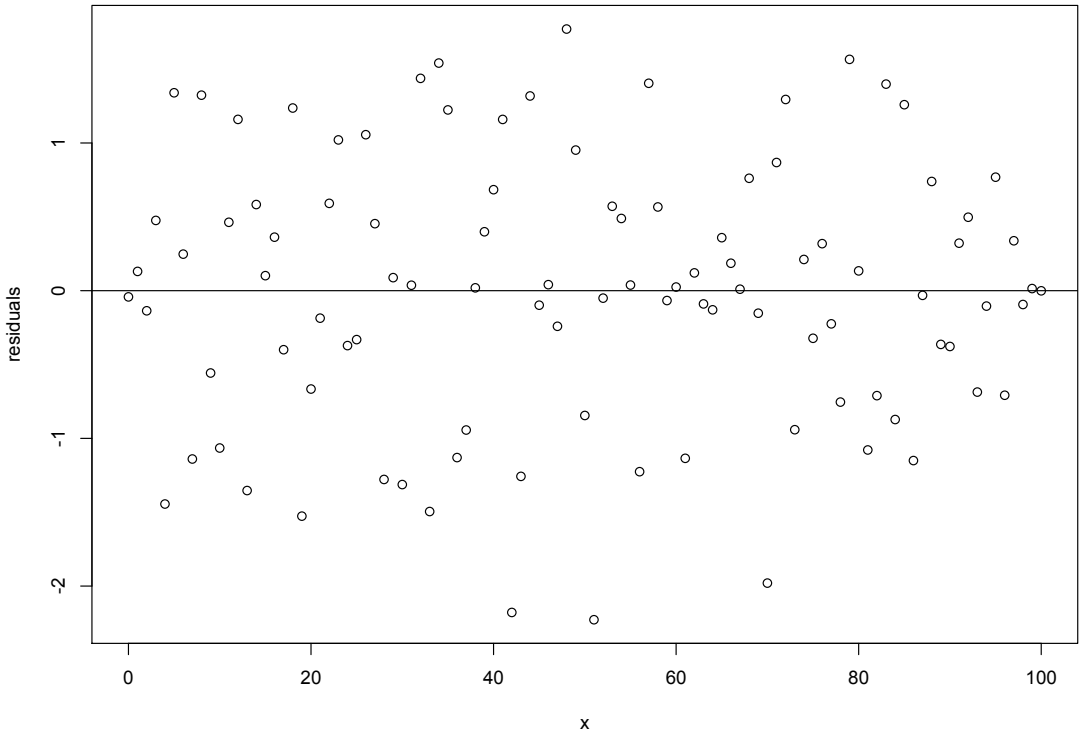
Residual plot



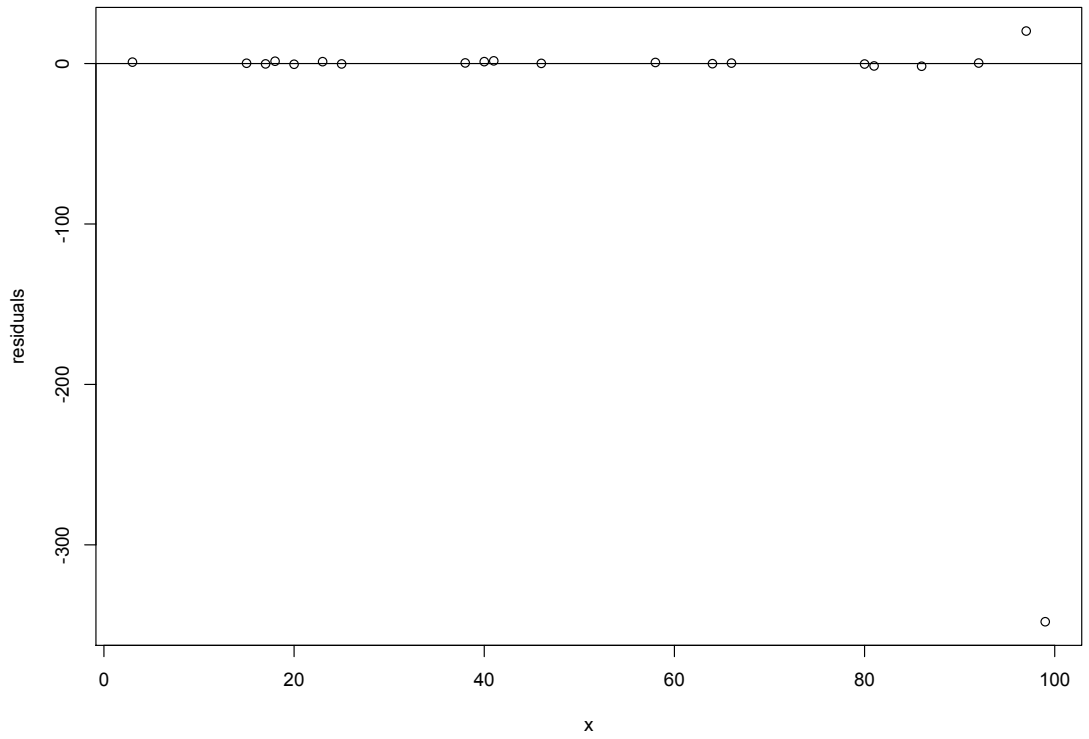
Overfitting



Residual plot ...

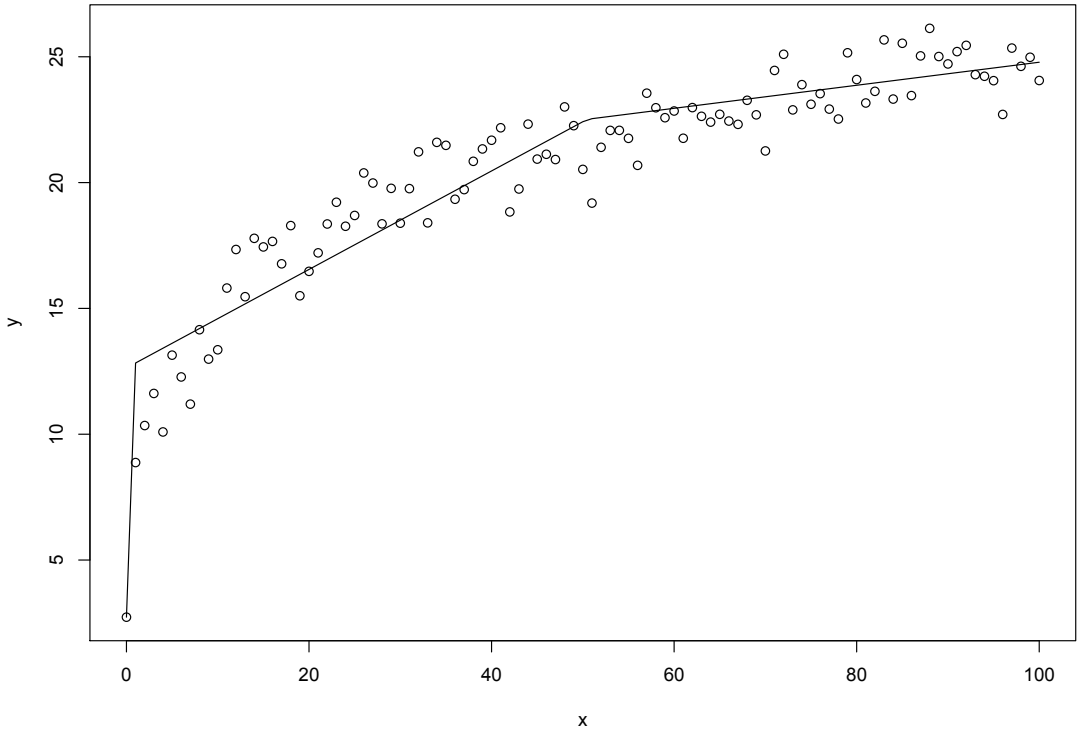


Training/testing split

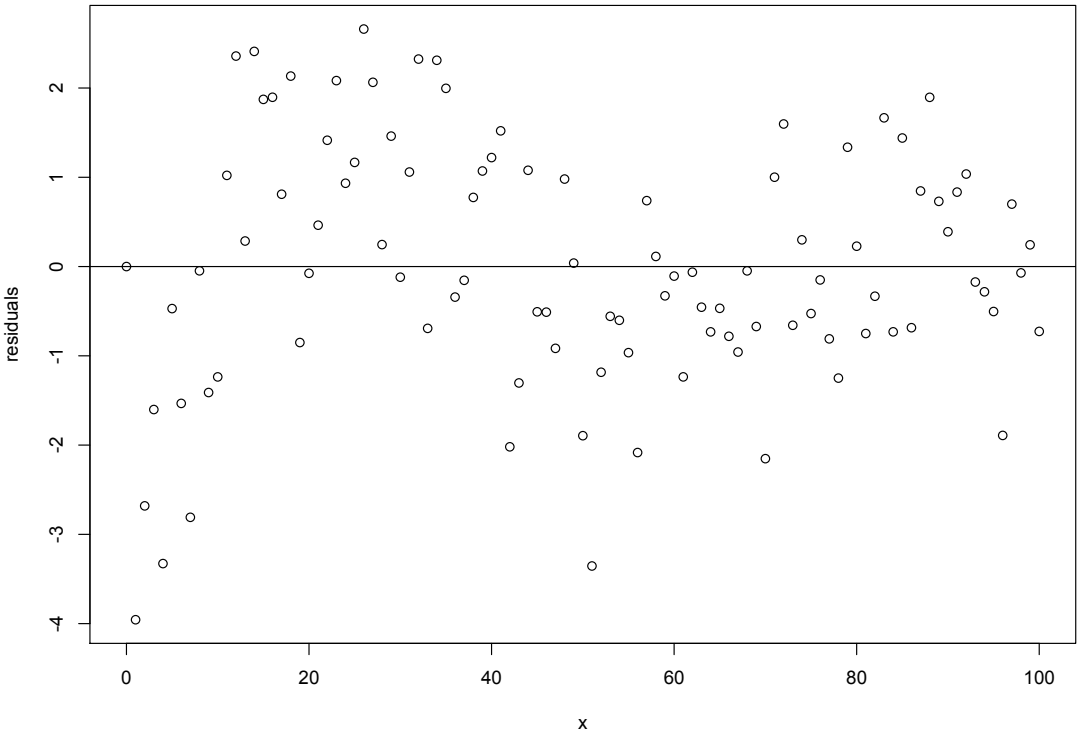


Piecewise linear regression

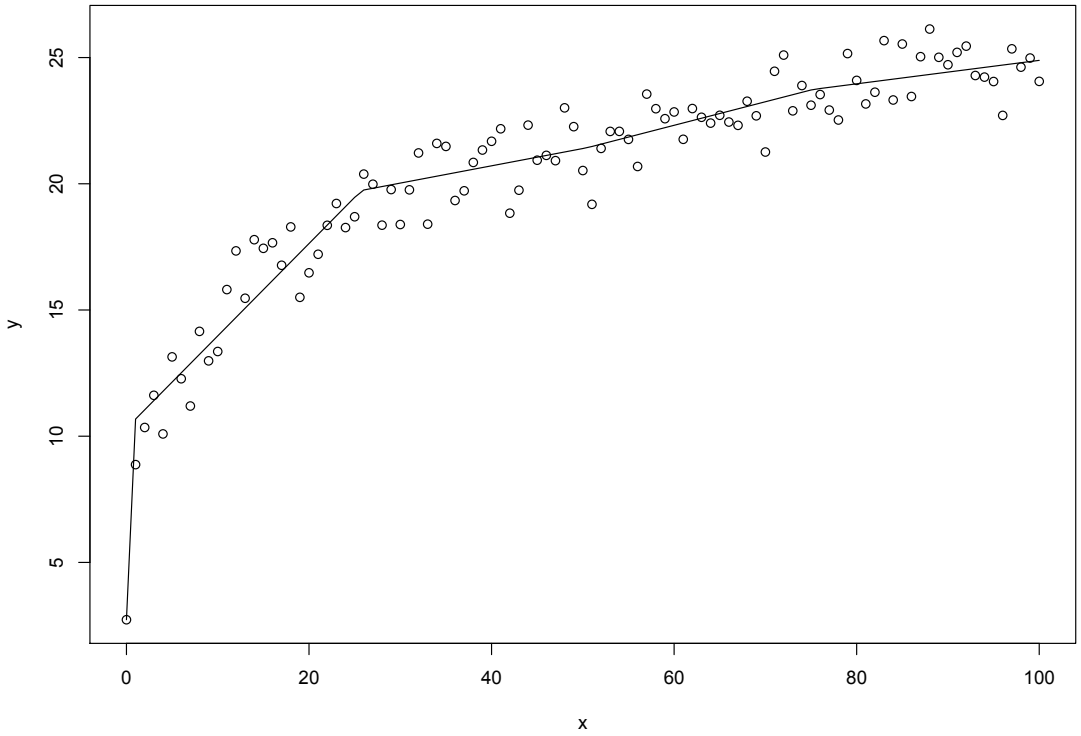
- $f(x) = \beta_0 + \beta_1 x + \beta_2 x \cdot I(x > 10) + \ldots$



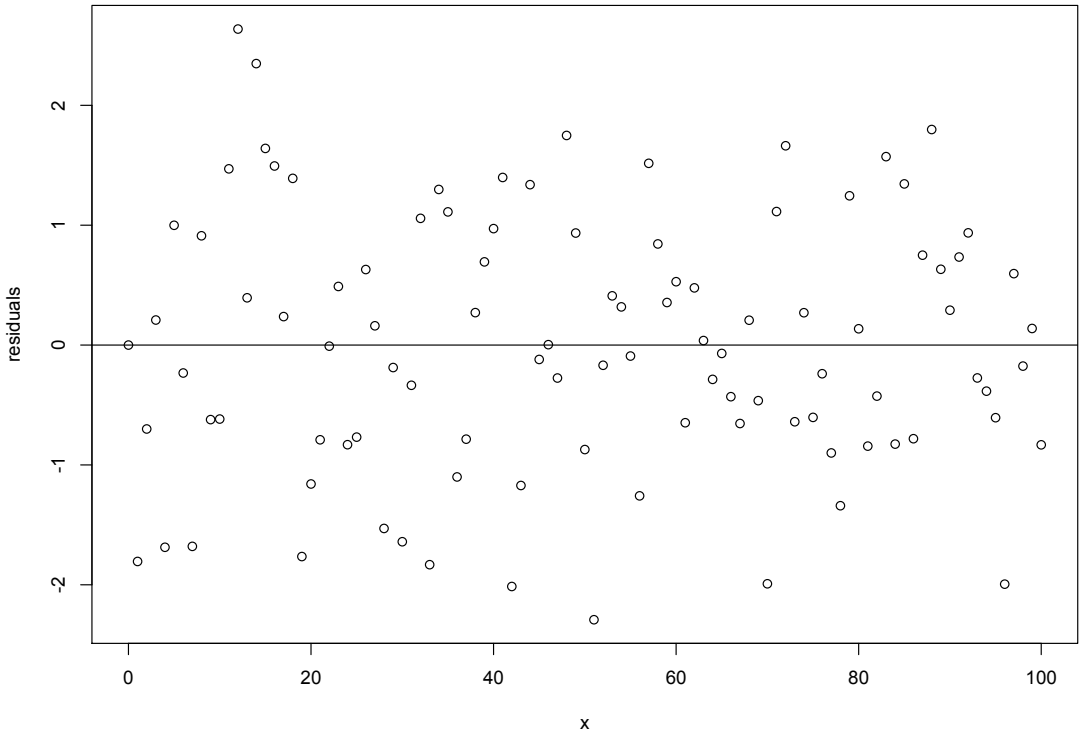
Somehow underfitted …



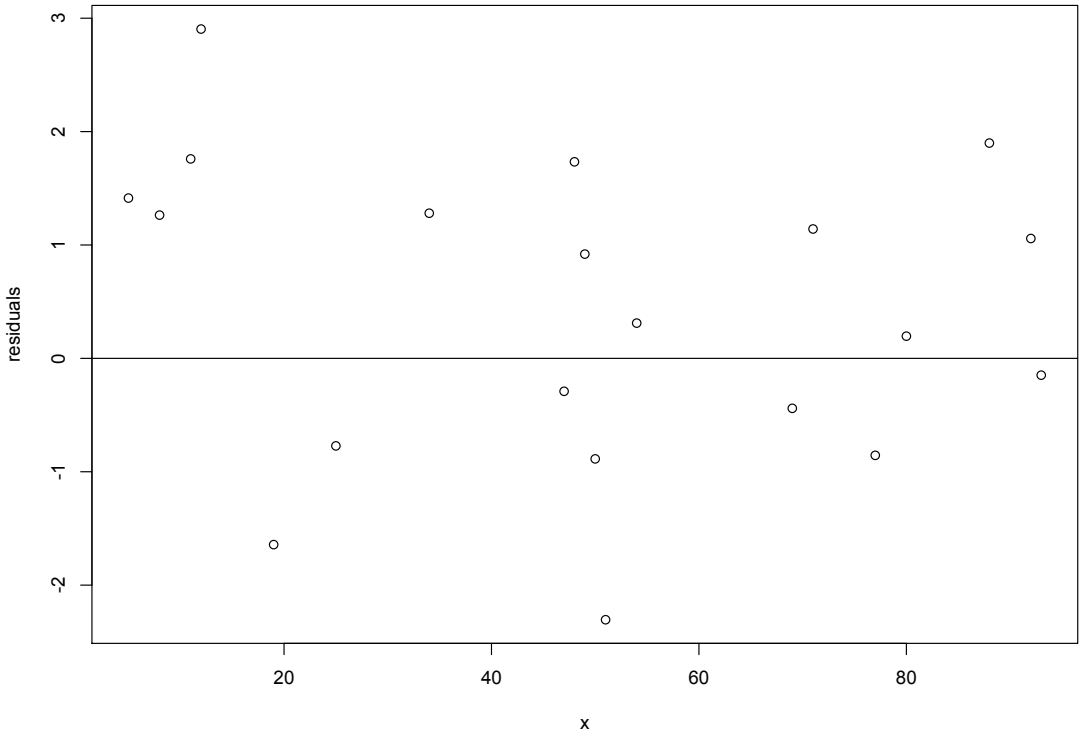
Try to use more knots



Almost just right



Training/testing split



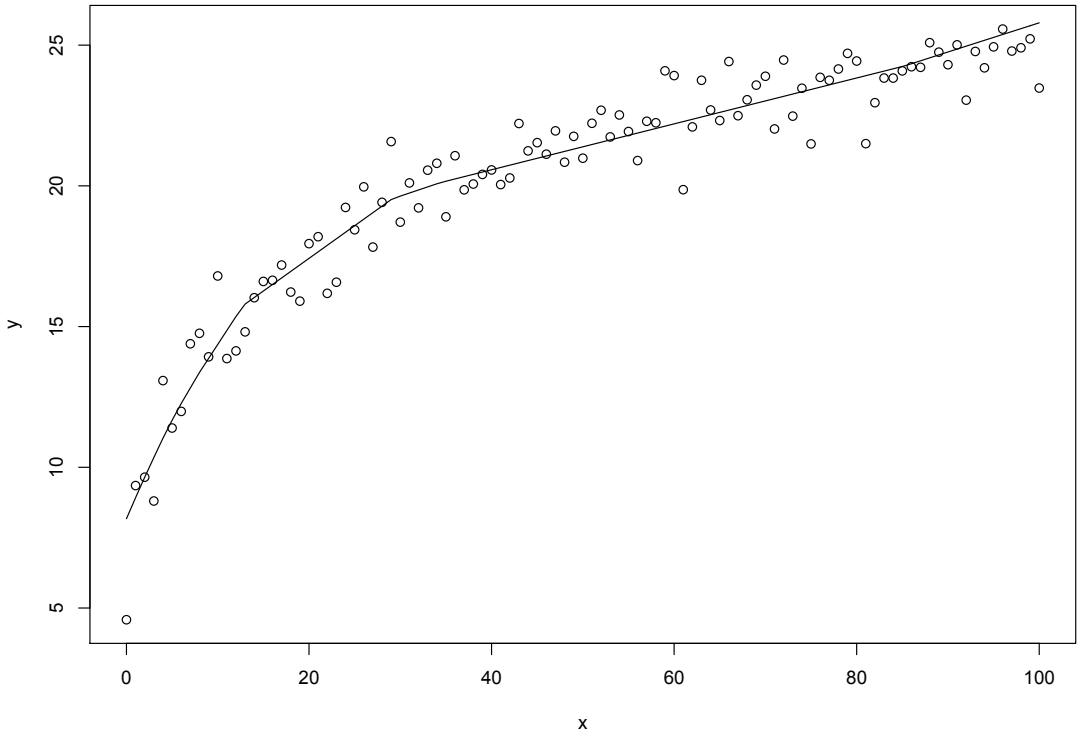
Piecewise linear regression by ReLu function

- Rectifier activation function:

$$\mathrm{ReLu}(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$$

- $f(x) = \beta_0 + \beta_1 x + \beta_2 x \cdot I(x > 10) + \ldots$
  $= \beta_0' + \beta_1 x + \beta_2 \cdot \mathrm{ReLu}(x - 10) + \ldots$

ReLu activation



Homework: diabetes data

1. Use sklearn.linear_model.SGDRegressor to find the best linear model (in terms of minimum cross-validation error) for the diabetes dataset
2. Is “Average blood pressure” an important factor for diabetes disease?