Lecture 17

03/22/2022

$\boxed{\text{REGRESSION}} \approx \boxed{\text{FITTING A LINE TO DATA}}$

- 2D linear regression

  ↳ we already have equations and techniques from MLE for homo/heteroscedastic uncertainties.

⟹ much easier to generalize to different LINEAR fitting models using matrix math.

e.g. $$y_i = \theta_0 + \theta_1 x_i$$

$$\Downarrow$$

$$\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_N \end{pmatrix} = \begin{pmatrix} 1 & x_0 \\ 1 & x_1 \\ \vdots & \vdots \\ 1 & x_N \end{pmatrix} \begin{pmatrix} \theta_0 \\ \theta_1 \end{pmatrix}$$

$$\Downarrow$$

$$\vec{Y} = M \vec{\theta}$$

model

model design matrix

linear model parameters

$$\ln L = -\frac{1}{2}(Y - M\theta)^T C^{-1} (Y - M\theta) - \frac{1}{2}\ln[\det(C)]$$

where $C = \begin{pmatrix} \sigma_0^2 & & & 0 \\ & \sigma_1^2 & & \\ & & \ddots & \\ 0 & & & \sigma_N^2 \end{pmatrix}$

$\underbrace{\phantom{\begin{pmatrix} \sigma_0^2 & \sigma_1^2 & \sigma_N^2 \end{pmatrix}}}_{\text{measurment uncertainty covariance matrix}}$

$\Longrightarrow$ use linear algebra to analytically maximize $\ln \mathcal{L}$ simultaneously over all θ components.

$$\hat{\theta}_{ML} = (M^T C^{-1} M)^{-1} M^T C^{-1} Y$$

$$\Sigma_{\hat{\theta}} = \underbrace{(M^T C^{-1} M)}_{M^T C^{-1} M \,=\, \text{Fisher matrix}}{}^{-1} = \begin{pmatrix} \sigma_{\theta_0}^2 & \sigma_{\theta_0}\sigma_{\theta_1} \\ \sigma_{\theta_0}\sigma_{\theta_1} & \sigma_{\theta_1}^2 \end{pmatrix}$$

Multivariate Linear Regression

↳ what if we have more than one data feature?

↳ we fit a HYPERPLANE, not a line.

$$y_i = \theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2} + \theta_{ik} x_{ik}$$

$\Longrightarrow$ $M = \begin{pmatrix} 1 & x_{01} & x_{02} & \cdots & x_{0k} \\ 1 & x_{11} & x_{12} & \cdots & x_{1k} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{N1} & x_{N2} & \cdots & x_{Nk} \end{pmatrix}$

$Y = M\theta$

straight-line regression just a special case of polynomial regression

- **Polynomial Regression**

  ↳ what if we want to fit a polynomial, not just a straight line?

$$y_i = \theta_0 + \theta_1 x_i + \theta_2 x_i^2 + \dots + \theta_N x_i^N$$

$$\Rightarrow \quad Y = M\theta \qquad M = \begin{pmatrix} 1 & x_0 & x_0^2 & x_0^3 \\ 1 & x_1 & x_1^2 & x_1^3 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_N & x_N^2 & x_N^3 \end{pmatrix}$$

  ⇒ still LINEAR REGRESSION, because coefficients θ appear linearly.

- **Basis function regression**

polynomial regression just a special case of basis function regression

  ↳ what if our model has some weird basis functions?

$$y_i = \theta_0 + \theta_1 \sin(x_i)$$

$$\Rightarrow \quad Y = M\theta \qquad M = \begin{pmatrix} 1 & \sin(x_1) \\ 1 & \sin(x_2) \\ \vdots & \vdots \\ 1 & \sin(x_N) \end{pmatrix}$$

also "Nadarya-Watson Regression"

Kernel regression

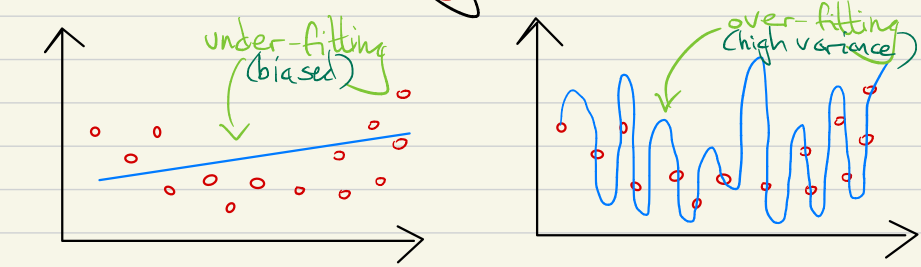
↳ replace each data point by a kernel.

$$f(x|K) = \frac{\sum_{i=1}^{N} K\left(\frac{\|x_i - x\|}{h}\right) y_i}{\sum_{i=1}^{N} K\left(\frac{\|x_i - x\|}{h}\right)}$$

→ essentially a weighted sum of y_i

$$w_i(x) = \frac{K\left(\frac{\|x_i - x\|}{h}\right)}{\sum_{i=1}^{N} K\left(\frac{\|x_i - x\|}{h}\right)}$$

Over/Under-fitting



↳ use cross-validation! data → training, validation, test

* model parameters learned from training set
* hyper-parameters (e.g. polynomial degree) learned from validation set.
* test set then assesses error for a new set of data.