Purdue-NCKU program

Lecture 9
Advanced Regression Technique

Dr. Qifan Song

Beyond Normality

generalized linear regression model

To handle data with different distribution and different support

- $E(Y|X) = g^{-1}(X\beta)$, where g is called link function
- $Y \sim f_Y(y \mid \theta, \tau) = h(y, \tau) \exp\left(\frac{b(\theta)T(y) - A(\theta)}{d(\tau)}\right)$
  - examples include families of normal, Bernoulli, Poisson, exponential distributions
  - We reparameterize the model such that $b(\theta) = \theta$. If furthermore that $T(y) = y$ then we can show that $E(Y) = A'(\theta)$ and $var(Y) = A''(\theta)d(\tau)$
  - Canonical link: $X\beta = b(\theta)$

Common choice

Common distributions with typical uses and canonical link functions

Distribution	Support of distribution	Typical uses	Link name	Link function, $\mathbf{X}\boldsymbol{\beta} = g(\mu)$	Mean function
Normal	real: $(-\infty, +\infty)$	Linear-response data	Identity	$\mathbf{X}\boldsymbol{\beta} = \mu$	$\mu = \mathbf{X}\boldsymbol{\beta}$
Exponential Gamma	real: $(0, +\infty)$	Exponential-response data, scale parameters	Negative inverse	$\mathbf{X}\boldsymbol{\beta} = -\mu^{-1}$	$\mu = -(\mathbf{X}\boldsymbol{\beta})^{-1}$
Inverse Gaussian	real: $(0, +\infty)$		Inverse squared	$\mathbf{X}\boldsymbol{\beta} = \mu^{-2}$	$\mu = (\mathbf{X}\boldsymbol{\beta})^{-1/2}$
Poisson	integer: $0, 1, 2, \ldots$	count of occurrences in fixed amount of time/space	Log	$\mathbf{X}\boldsymbol{\beta} = \ln(\mu)$	$\mu = \exp(\mathbf{X}\boldsymbol{\beta})$
Bernoulli	integer: $\{0, 1\}$	outcome of single yes/no occurrence	Logit	$\mathbf{X}\boldsymbol{\beta} = \ln\left(\frac{\mu}{1-\mu}\right)$	$\mu = \frac{\exp(\mathbf{X}\boldsymbol{\beta})}{1+\exp(\mathbf{X}\boldsymbol{\beta})} = \frac{1}{1+\exp(-\mathbf{X}\boldsymbol{\beta})}$
Binomial	integer: $0, 1, \ldots, N$	count of # of "yes" occurrences out of N yes/no occurrences		$\mathbf{X}\boldsymbol{\beta} = \ln\left(\frac{\mu}{n-\mu}\right)$	
Categorical	integer: $[0, K)$ K-vector of integer: $[0, 1]$, where exactly one element in the vector has the value 1	outcome of single K-way occurrence		$\mathbf{X}\boldsymbol{\beta} = \ln\left(\frac{\mu}{1-\mu}\right)$	
Multinomial	*K*-vector of integer: $[0, N]$	count of occurrences of different types (1 .. *K*) out of *N* total *K*-way occurrences			

source: wikipedia.com

Estimation

- MLE estimation: $\beta = \arg\max l(\beta) = \arg\max\{\sum b(\theta_i)T(Y_i) - A(\theta_i)\}$
- Newton-Raphson method: $\beta_{t+1} = \beta_t - (l''(\beta))^{-1}l'(\beta)$. If b and T are identity mapping, then
  - $$\frac{\partial l(\beta)}{\partial \beta} = \sum X_i'(Y_i - A'(\theta_i))$$
  - $$\frac{\partial^2 l(\beta)}{\partial \beta \partial \beta'} = -\mathbf{X}'\mathbf{W}\mathbf{X},$$

    where $\mathbf{W} = diag(w_1, \ldots, w_n) = [diag(A''(\theta_1), \ldots, A''(\theta_n))]^{-1}$
  - Iteratively Reweighted Least Squares

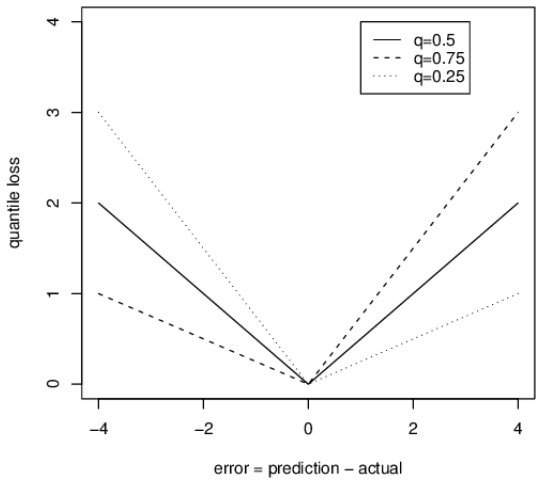
    $$\beta_{t+1} = (\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}\mathbf{z},$$

    where $\mathbf{z} = \mathbf{X}\beta_t + \mathbf{W}^{-1}(\mathbf{Y} - \mathbf{A}'(\theta))$, $\mathbf{A}'(\theta) = (A'(\theta_1), \ldots, A'(\theta_n))'$

quantile regression

The method of least squares estimates the conditional mean of the response variable, while quantile regression estimates the conditional median (or other quantiles) of the response variable.

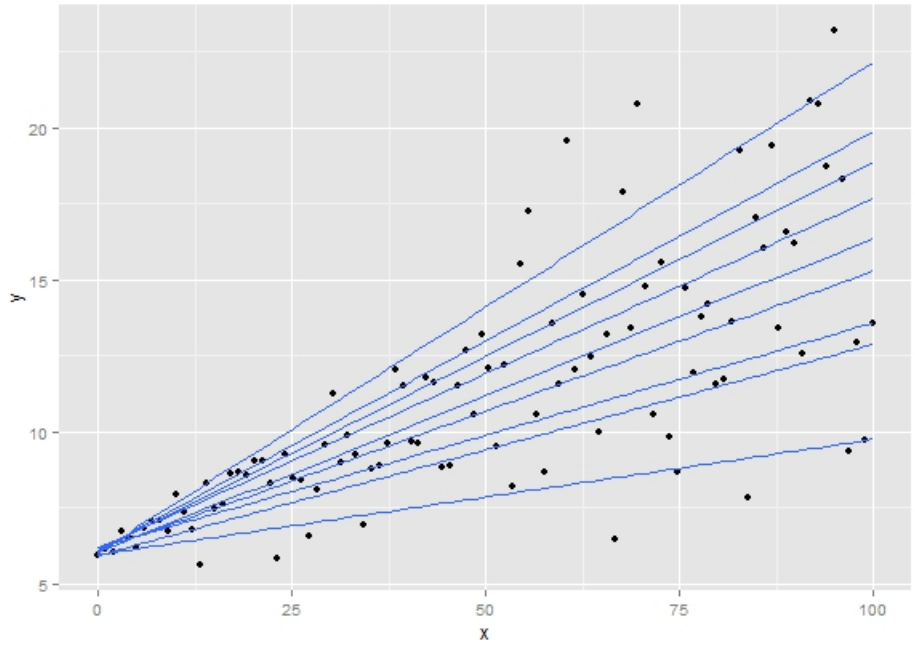
- Square distance leads to mean estimation
- Pinball loss, $\rho_\tau(x) = x(\tau - I_{(x<0)})$ leads to τ-th quantile estimation



quantile linear regression

- $Q = \sum_{i=1}^{n} \rho_\tau(Y_i - X\beta)$
- More robust than OLS to non-normal errors and outliers in y direction)
- Optimization: quadratic majorization or gradient descent

Quantile Regression in the Heteroscedastic Error Model



Beyond Linearity

Local regression

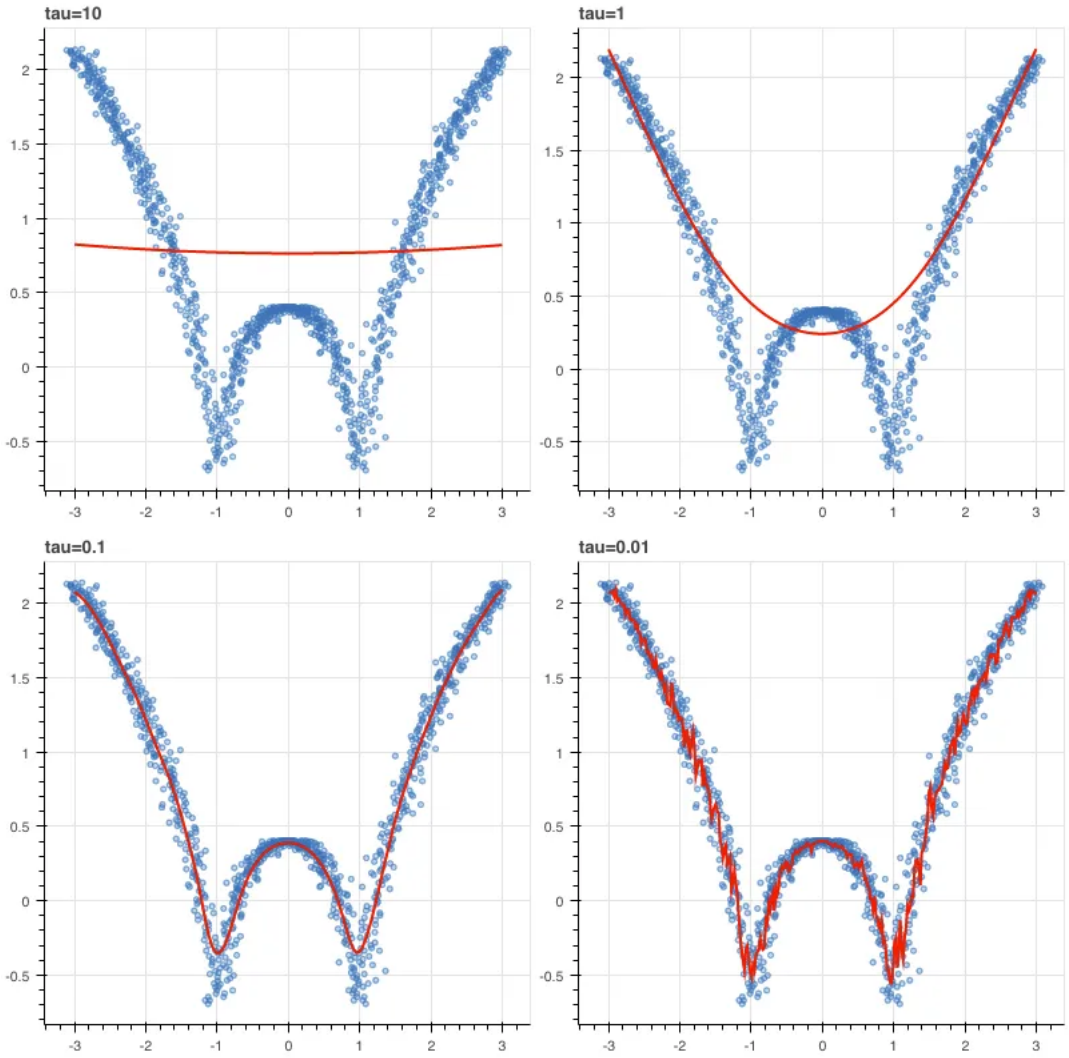
Adaptive to local smoothness via weighting

- $E(Y) = f(X)$ and we estimate f locally
- $f(X_h)$ is estimated by
  - Weighted Least Squared estimator with (X_i, Y_i) associated with weights $w(X_i, X_h)$.
  - w is larger is X_i and X_h is closer. Local data play more important role of estimation
  - Choices of weight function w: Gaussian $\exp\{-\|X_i - X_h\|^2/\sigma^2\}$, uniform $I(\|X_i - X_h\| \leq b)$, triangle function, or k-NN

Local regression

- Choice of Bandwidth can be important (trade-off between smoothness and biasness)
- Biased estimation, especially when $f''(x)$ is large
- Consistent estimation, as local data density goes to infinity and bandwidth goes to zero.
- Locally 0-order (kernel smoother) or high-order (local polynomial regression) extension.

Local regression



source: https://medium.com/100-days-of-algorithms/day-97-locally-weighted-regression-c9cfaff087fb

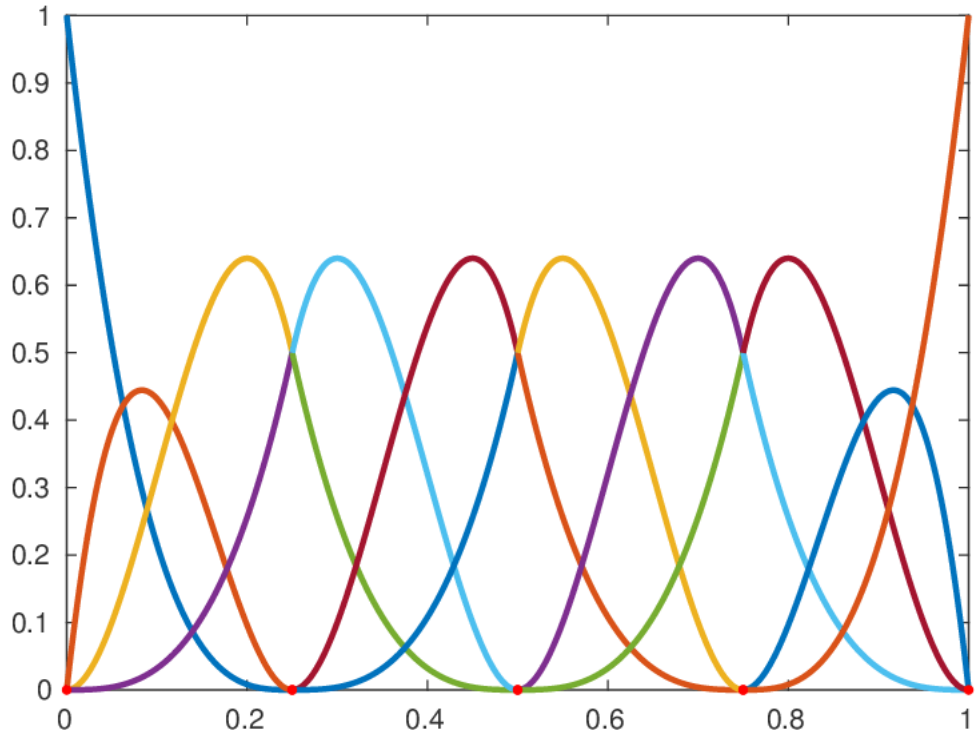
Beyond Linearity

Sieve/Spline regression

- Sieve expansion: $f_m(x) = \sum_{j=1}^{K_m} z_{jm}(x)\beta_{jm}$ approximates true f^*
- z_{jm} is pre-designed nonlinear basis function.
- A larger K_m (hopefully) leads to a better approximation
- Example: polynomial basis, wavelet basis and spline basis
- Estimation follows MLR

Spline regression

- Spline function: piece-wise polynomial functions with smooth connection
- B-spline or basis spline. Any spline function of given degree can be expressed as a linear combination of B-splines of that degree, under same knots setting.



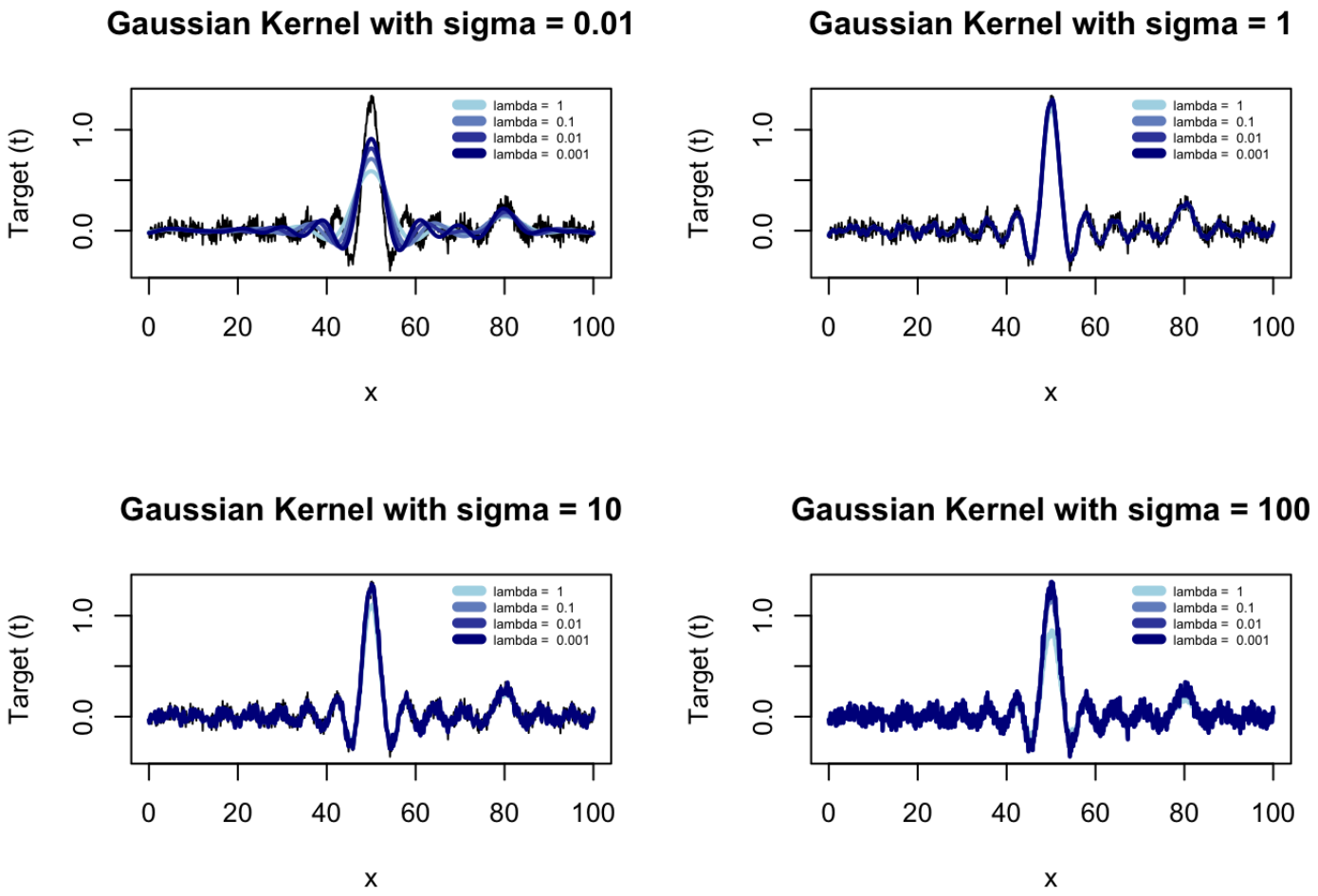
Spline regression

- Choice of degree (usually 3)
- Choice of knots (usually uniform grid)
- Smoothing spline: minimizing $\sum_{i=1}^{n}(Y_i - f(X_i))^2 + \lambda \int_0^2 (f''(x))^2 dx$
- The solution to this minimization problem is a cubic spline with knots set being the X_i's.
- λ controls the smoothness level

Kernel Ridge regression

- Mapping all X_i to feature space $(\phi_1(X_i), \ldots, \phi_k(X_i), \ldots) = \phi(X_i)$
- Linear regression on feature space, or view ϕ_i's as (infinite) basis functions
- Denote Φ be the $n \times \infty$ dimension design matrix, we consider ridge regression
- $\beta = (\Phi^T\Phi + \lambda I_\infty)\Phi^T Y = \Phi^T(\Phi\Phi^T + \lambda I_n)Y$
- Fitted value $\phi(X_h)^T\Phi^T(\Phi\Phi^T + \lambda I_n)Y$
- Kernel $< \phi(X_i), \phi(X_j) >= K(X_i, X_j)$
- Requires inverting n by n matrix

Choice of kernel and λ



Source: https://rpubs.com/Saulabrm/210788

Beyond Linearity

Gaussian Process

- View function f as a random process (GP)
- For any vector $(X_1, \ldots, X_n)$, $(f(X_1), \ldots, f(X_n))$ follows multivariate normal with covariance matrix $k(X_i, X_j)$.
- $Y_i = f(X_i) + N(0, \sigma^2)$
- What is the joint distribution of $Y_1, \ldots, Y_n, Y_h$, or $Y_1, \ldots, Y_n, f(X_h)$,?

- Conditional distribution of multivariate normal (source: wikipedia)

Conditional distributions [edit]

If *N*-dimensional $\mathbf{x}$ is partitioned as follows

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \text{ with sizes } \begin{bmatrix} q \times 1 \\ (N-q) \times 1 \end{bmatrix}$$

and accordingly $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are partitioned as follows

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix} \text{ with sizes } \begin{bmatrix} q \times 1 \\ (N-q) \times 1 \end{bmatrix}$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{bmatrix} \text{ with sizes } \begin{bmatrix} q \times q & q \times (N-q) \\ (N-q) \times q & (N-q) \times (N-q) \end{bmatrix}$$

then the distribution of $\mathbf{x}_1$ conditional on $\mathbf{x}_2 = \mathbf{a}$ is multivariate normal $(\mathbf{x}_1 \mid \mathbf{x}_2 = \mathbf{a}) \sim N(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\Sigma}})$ where

$$\bar{\boldsymbol{\mu}} = \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\mathbf{a} - \boldsymbol{\mu}_2)$$

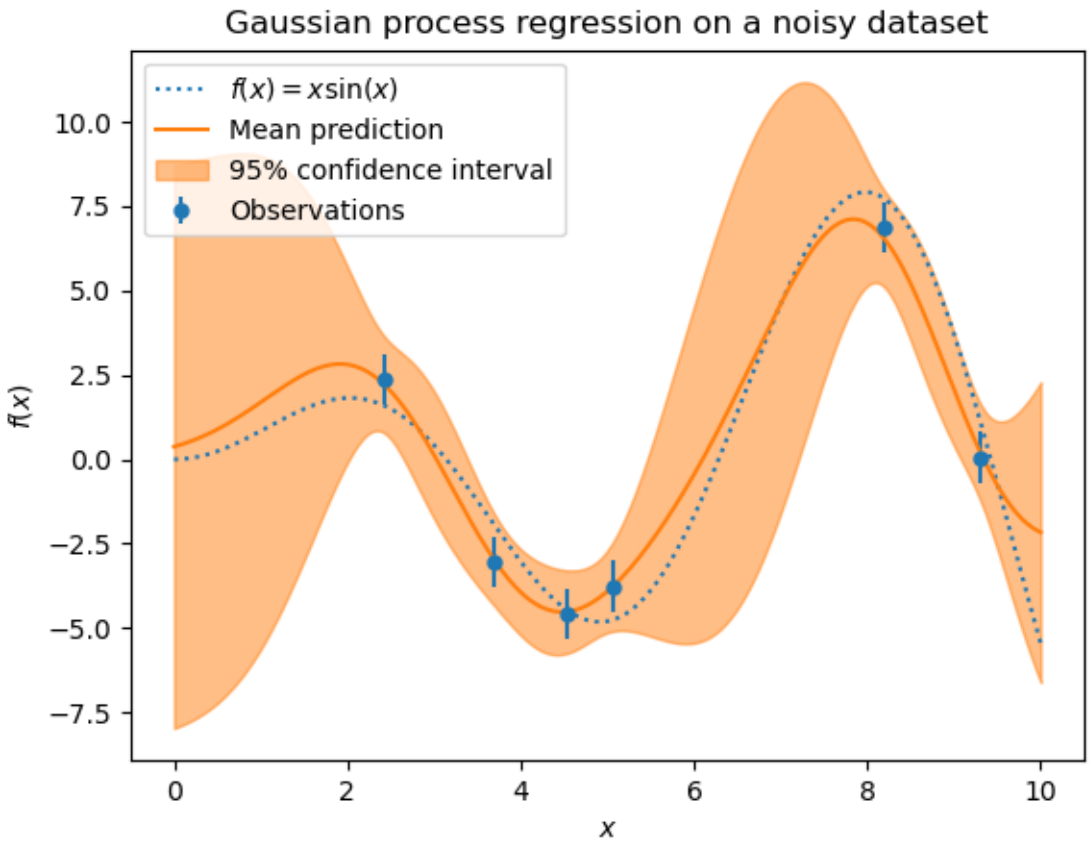
and covariance matrix

$$\bar{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}.$$ [20]

- Estimate (predicted) $f(X_h)$ or Y_h by the conditional distribution.

- Bayesian interval inference

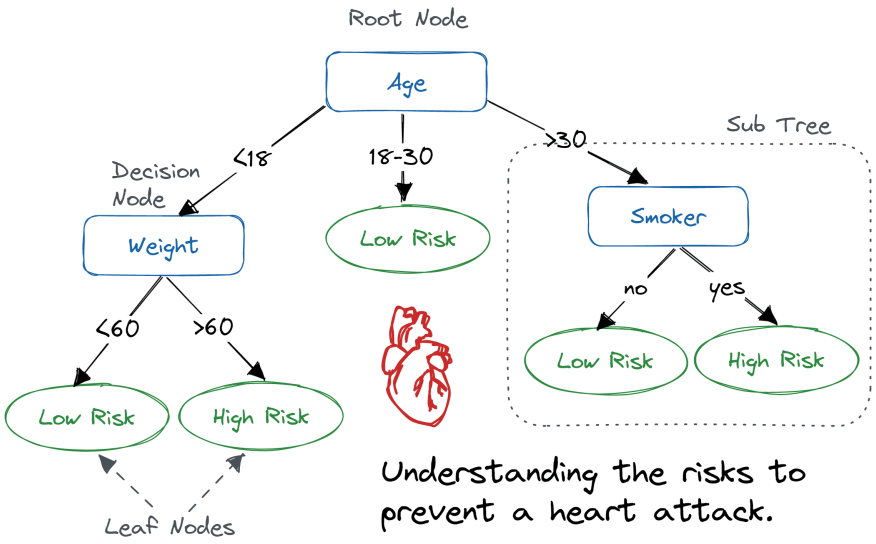
Example



source: https://scikit-learn.org

Tree Model and Random Forest

Constant prediction over a rectangle region of the predictors



Source: https://www.datacamp.com/tutorial/decision-tree-classificatic python

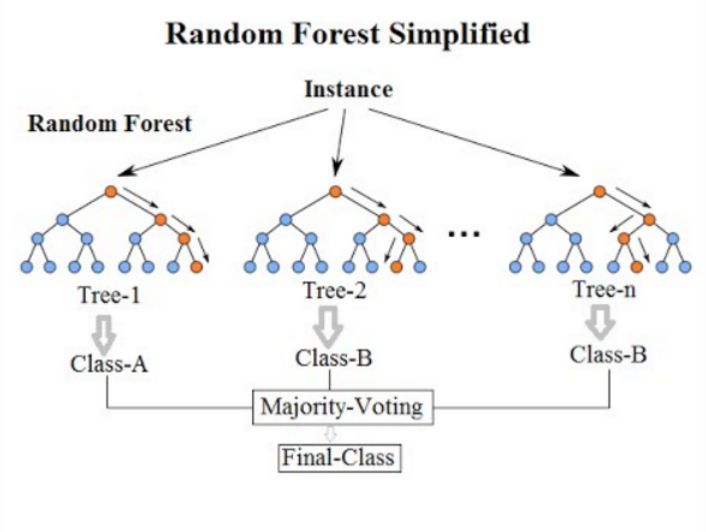
Constructing Trees

- Piecewise constant regression function
- Basically partition the X space into rectangles
- Predicted value is mean of responses in rectangle
- Minimize SSE via greedy search (sequentially partitioning)
- Trade off between minimizing SSE and complexity; proper stopping rule (e.g., never yield a node that contains less than 2% of the data)
- Generalize to non-rectangle split

Random Forest

trees that are grown very deep tend to learn highly irregular patterns. Overfitting occurs

- Average result of many many small trees.
- Each tree is based on a subset of data set; reduce variance
- Each tree is based on a subset of predictors; reduce tree-to-tree correlation



Source: Wikipedia

Beyond Linearity

Neural Network

- Combination of affine mapping and nonlinear elementwise mapping

$$y = B_1 \circ \sigma_1 \circ B_2 \circ \sigma_2 \ldots B_m \circ x$$

- Choice of affine mapping: fully connected layer, CNN layer
- Choice of nonlinear mapping: activation function, Batch normalization
- Other special structure: skip connection, Recurrent neural networks

DNN is not only for regression or classification. Generative model, unsupervised learning, "longitudinal" data

Deep learning and A.I.

- Surprising performance due to explosion of the scale of structure, and massive of training data
- How to use non-supervised data
- How to use pre-trained model
- How to improve adversarial robustness
- How to improve fairness
- How to introduce inference such as C.I. or hypothesis testing?

Neural Network

- Universality of approximation power

  Relu networks means all piecewise linear functions.

- (Stochastic) Gradient Descent Chain rule and Back propagation algorithm (GPU computing)

- Non-convex optimization problem, non-unique solution

- Required experienced research to train a good DNN

- Other special structure: skip connection, Recurrent neural networks

Sparse Model

High dimensional regression $E(Y) = X\beta$ where $\beta \in R^p$ and $p \gg n$

- $\mathbf{X}^T\mathbf{X}$ is not invertible, and columns of $\mathbf{X}$ is always linear dependent
- OLS estimation is not unique, with SSE $= 0$.
- People are willing to believe that there exist a "sparse truth", i.e., most entries of true β^* is almost zero
- Restricted estimation: finding a best sparse β that minimizes the SSE

LASSO

- LASSO (least absolute shrinkage and selection operator):

$$\mathbf{b} = \arg\min \|\mathbf{Y} - \mathbf{X}\mathbf{b}\|, \text{ subject to } \sum |b_i| \leq t$$

- Equivalent estimation:

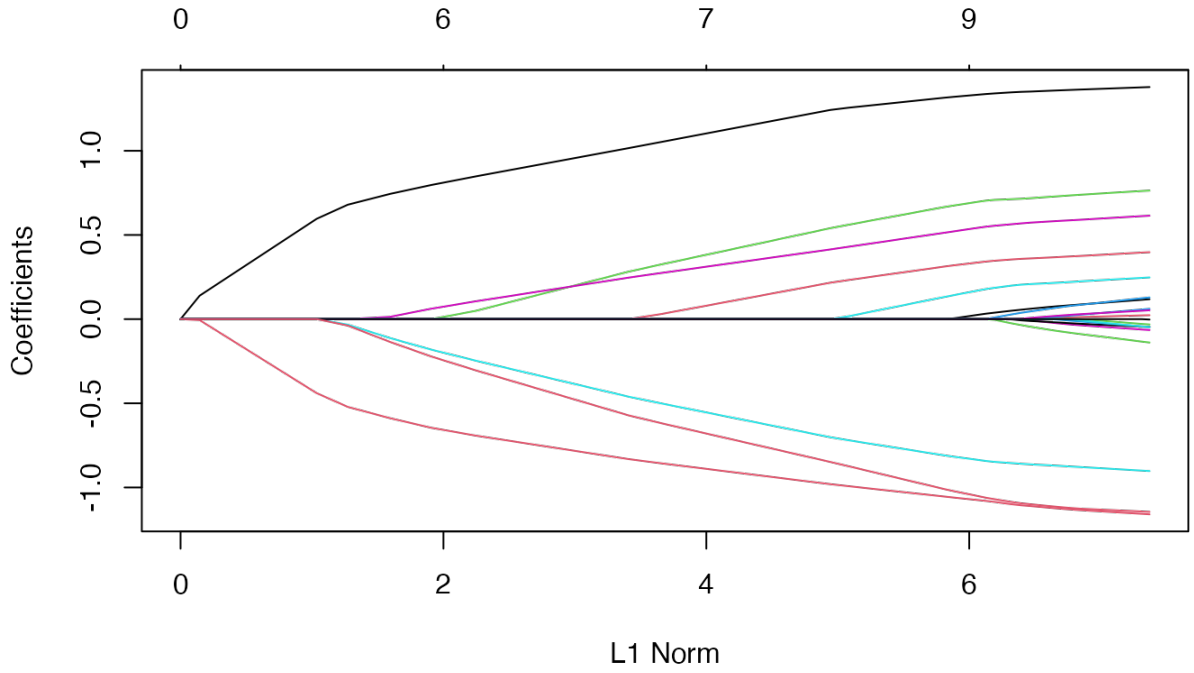
$$\mathbf{b} = \arg\min \|\mathbf{Y} - \mathbf{X}\mathbf{b}\| + \lambda \sum |b_i|$$

- a sequence of models, from null to full, as t increases (or λ decrease)

- How to choose λ?

- One can pick one among them, depending on some criterion (e.g. testing accuracy).

The non-smoothness of absolute function introduces sparsity in the solution.



Source: glmnet package in R.

The penalty can be generalized to any modeling.

Model Complexity and Generalization bound

- Candidates family of functions $\mathcal{F}$, and loss function L
- Optimization problem:

$$R_{emp}(f) = \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i))$$

$$\widehat{f} = \arg\min_{f\in\mathcal{F}} R_{emp}(f)$$

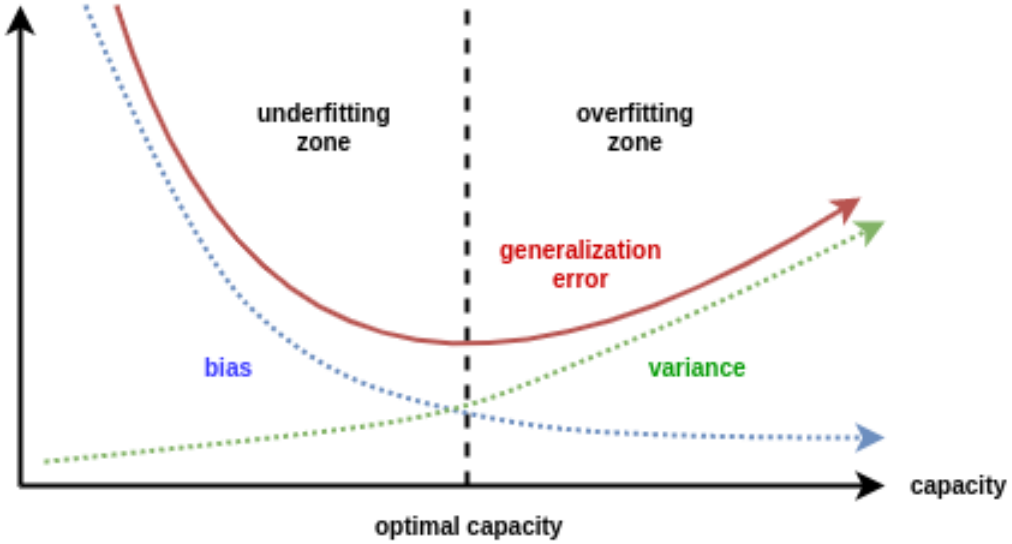
- Generalization error: the discrepancy between training accuracy and testing accuracy.

  If all $f \in \mathcal{F}$ are bounded functions, then

$$R(\widehat{f}) \leq R_{emp}(\widehat{f}) + \sqrt{\frac{\log|\mathcal{F}| + \log(2/\delta)}{2n}}$$

  with probability $1-\delta$, where $R(f) = EL(y, f(x))$

Balance between model complexity and fitting perfor



source: https://djsaunde.wordpress.com/2017/07/17/the-bias-variance-tradeoff/

Explain

- Bias: $f^* \notin \mathcal{F}$, or $\min_{f \in \mathcal{F}} R(f) > \min_f R(f)$

  $\mathcal{F}$ is incapable to model the true function

- Variance: the estimation is too difficult, hard to search a overly large space

- The theorem can be generalized to other complexity measure (e.g. V.C dimension, Rademacher complexity)

- The theorem only gives an upper bound, fails to explain model DNN behavior.