The Basics of Machine Learning

Xingcheng Xu

2021

Learning methods

- Supervised learning
- Unsupervised learning
- Semi-supervised learning
- Reinforcement learning

Framework of Supervised Learning

- Assumption
  - random vectors

$$(X, Y) : (\Omega, \mathcal{F}, \mathbb{P}) \to \mathscr{X} \times \mathscr{Y} \subset \mathbb{R}^{d+1}$$

  - the joint distribution: $\mathbb{P}_{X,Y} = \mathbb{P}$, the joint distribution function: $P(x, y)$. (Unknown in the real world problem)
  - use the information X to predict Y:

$$Y = f(X) + \varepsilon.$$

  - data

$$D = \{(X_1, Y_1), (X_2, Y_2), \cdots, (X_N, Y_N)\},$$

    independent identically distribution (i.i.d.), realization of the random vector (X, Y)

Five central elements in supervised learning:

- Data: $D = \{(X_1, Y_1), (X_2, Y_2), \cdots, (X_N, Y_N)\} \sim$ *Original Space*
- Features: $X = (X^{(1)}, X^{(2)}, \cdots, X^{(d)}) \sim$ *Latent Space*
- Models: $f \in \mathscr{S} \sim$ *Function Space*
- Cost/Loss: $\mathbb{E}[L(Y, f(X))]$
- Algorithm: numerical methods for model training

Questions

- If we know exactly the joint distribution $\mathbb{P}$, What is the best predictor f^*? In which sense?
- In reality, we do not know the joint distribution but the data D. How can we build a model and get an optimal predictor from the data?

The answer to the theoretical problem

The best predictor f^* depends on the loss function $L(\cdot, \cdot)$.

$$f^* = \arg\min_{f \in \mathscr{S}} \mathbb{E}[L(Y, f(X))]$$

- loss function is quadratic,

$$f^*(x) = \mathbb{E}[Y|X = x].$$

- loss function is absolute,

$$f^*(x) = \mathrm{Median}[Y|X = x].$$

- loss function is 0-1 loss,

$$f^*(x) = \mathrm{argmax}_y P(y|X = x).$$

The three main components of the error

The total error: $f^* - \hat{f}$, where

- f^* = the best predictor (Features, Cost/Loss)
- $\hat{f}$ = the output of the ML model.

Decomposition of the error:

$$f^* - \hat{f} = \underbrace{f^* - f_m}_{appr.} + \underbrace{f_m - f_{m,N}}_{estim.} + \underbrace{f_{m,N} - \hat{f}}_{optim.}$$

- $f^* - f_m$ = *approximation error*, due entirely to the choice of the hypothesis space (Model)
- $f_m - f_{m,N}$ = *estimation error*, additional error due to the fact that we only have a finite dataset (Data)
- $f_{m,N} - \hat{f}$ = *optimization error*, additional error caused by training (Algorithm)

Data

- Structured data
- Image, Video, 3D
- Text, Code
- Audio, Speech, Music
- Graph/Network
- Sequential data

Features

- Extracting good features is the most important thing for getting your analysis to work. It is much more important than good machine learning classifiers, fancy statistical techniques, or elegant code.
- Feature extraction is also the most creative part of data science and the one most closely tied to domain expertise.
- The best features are the ones that carefully reflect the thing you are studying.

Models: Which space does a model f live in?

The living space of model is called the hypothesis space such as the set of all measurable functions, the set of linear functions, or polynomial functions, or tree-like functions et al.

- Nonparametric model:

$$f \in \mathscr{S} = \{f \textit{ lives in some function space}\}.$$

- Parametric model:

$$f \in \mathscr{S} = \{f | f = f_\theta,\ \theta \in \mathbb{R}^q\}.$$

Models: Examples

Example

Linear regression:
$f \in \mathscr{S} = \{f_\beta | f_\beta(x) = \beta_0 + \beta_1' x,\ x \in \mathbb{R}^q,\ (\beta_0, \beta_1) \in \mathbb{R}^{q+1}\}$.

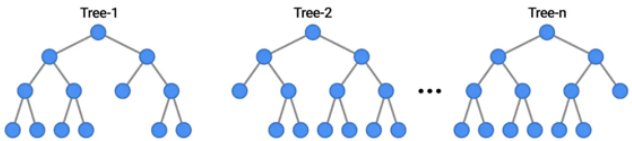
Example

Polynomial regression:
$f \in \mathscr{S} = \{f_\beta | f_\beta(x) = \sum_{j=0}^{q} \beta_j x^j,\ x \in \mathbb{R},\ \beta \in \mathbb{R}^{q+1}\}$.

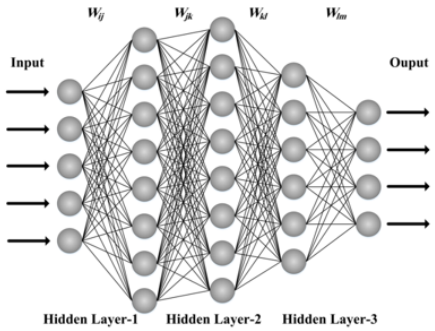
Example

What are the hypothesis spaces of decision trees, random forests, GBDT and neural networks models (MLP, CNN, RNN, Transformer)? $\cdots$

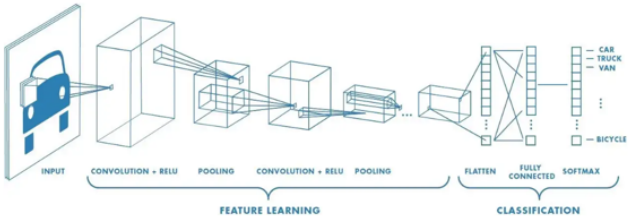
Models: Examples



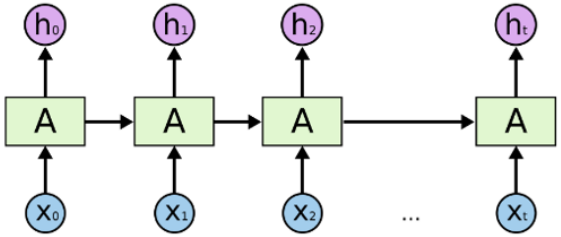
RF



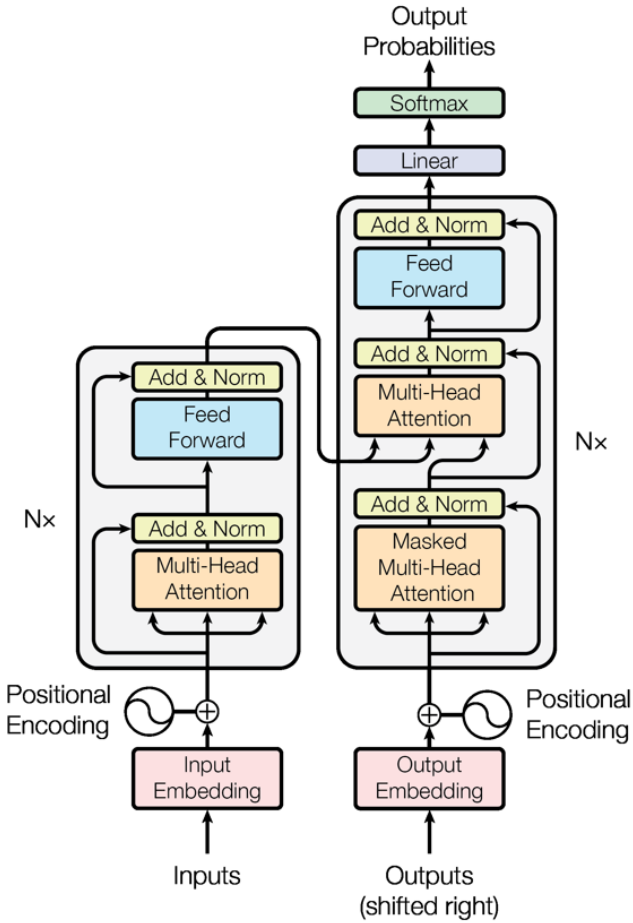
MLP



CNN



RNN



Transformer

Cost/Loss

- Loss function $L(Y, f(X))$
  - quadratic loss function

$$L(Y, f(X)) = (Y - f(X))^2$$

  - absolute loss function

$$L(Y, f(X)) = |Y - f(X)|$$

  - 0-1 loss function (for classification)

$$L(Y, f(X)) = 1_{\{Y \neq f(X)\}}$$

  - log-likelihood loss function (for probabilistic models)

$$L(P(Y|X)) = -\log P(Y|X)$$

  - ...

Cost/Loss

- Expected cost

$$\mathbb{E}[L(Y, f(X))] = \int_{\mathscr{X} \times \mathscr{Y}} L(y, f(x)) P(dx, dy)$$

- Empirical cost

$$\bar{L}_N(D) = \frac{1}{N} \sum_{i=1}^{N} L(Y_i, f(X_i))$$

Why does the empirical cost work? The answer is the law of large number (LLN) (Need assume that $\mathbb{E}|L(Y, f(X))| < \infty$, see Durrett (2019), PTE 5 edition, Theorem 2.2.14 (WLLN) and Theorem 2.4.1(SLLN)), i.e.

$$\frac{1}{N} \sum_{i=1}^{N} L(Y_i, f(X_i)) \to \mathbb{E}[L(Y, f(X))], \ a.s.$$

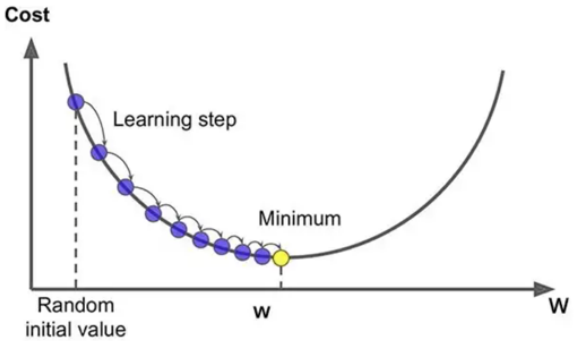
Algorithm

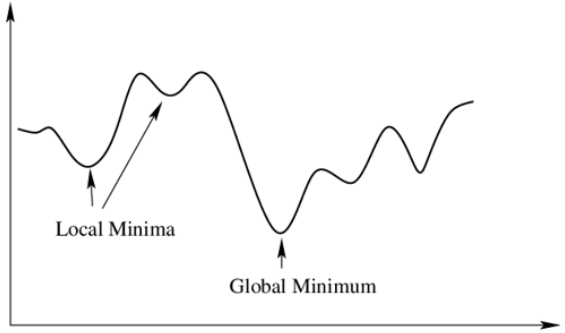
- Solve the optimization problem:

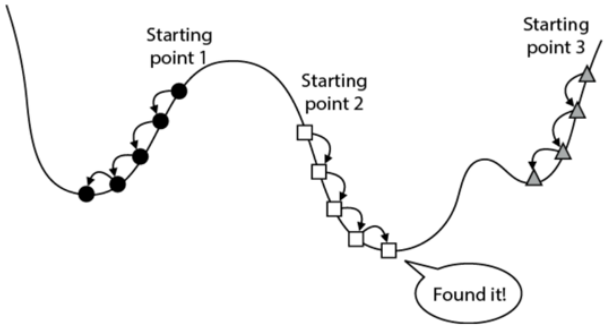
$$\hat{f} = \arg\min_{f \in \mathscr{S}} \frac{1}{N} \sum_{i=1}^{N} L(Y_i, f(X_i))$$

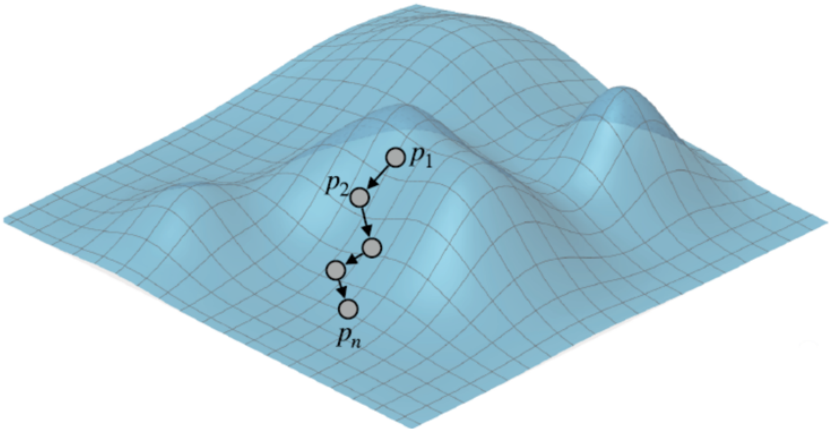
- If the analytic solution exists (e.g. linear regression), the optimizer is easy to calculate.
- If not, we should use some efficient numerical methods, e.g. gradient descent optimization algorithms such as Stochastic Gradient Descent (SGD), Adaptive Moment Estimation (Adam) to find the globally optimal solutions.

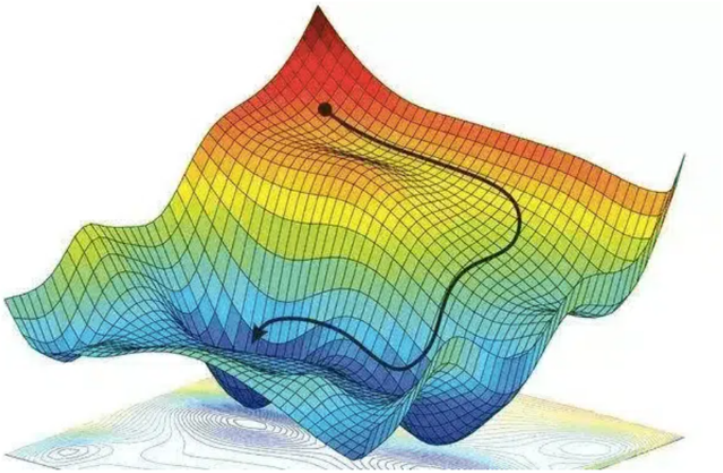
Algorithm: GD











Geometric interpretation under quadratic loss function

$L^2(\Omega, \mathcal{F}, \mathbb{P}) = \{Z \in \mathcal{F} : \ \mathbb{E}_{\mathbb{P}}[Z^2] < \infty\}$ is a Hilbert space where $\mathcal{F} = \sigma((X, Y))$ and $L^2(\Omega, \mathcal{F}_0, \mathbb{P})$ is a closed subspace with $\mathcal{F}_0 = \sigma(X)$. The conditional expectation $\mathbb{E}[Y|X]$ is the projection of Y onto $L^2(\Omega, \mathcal{F}_0, \mathbb{P})$. We have $Y = \mathbb{E}[Y|X] + \varepsilon$ and $\mathbb{E}[Y|X]$ is orthogonal with ε.

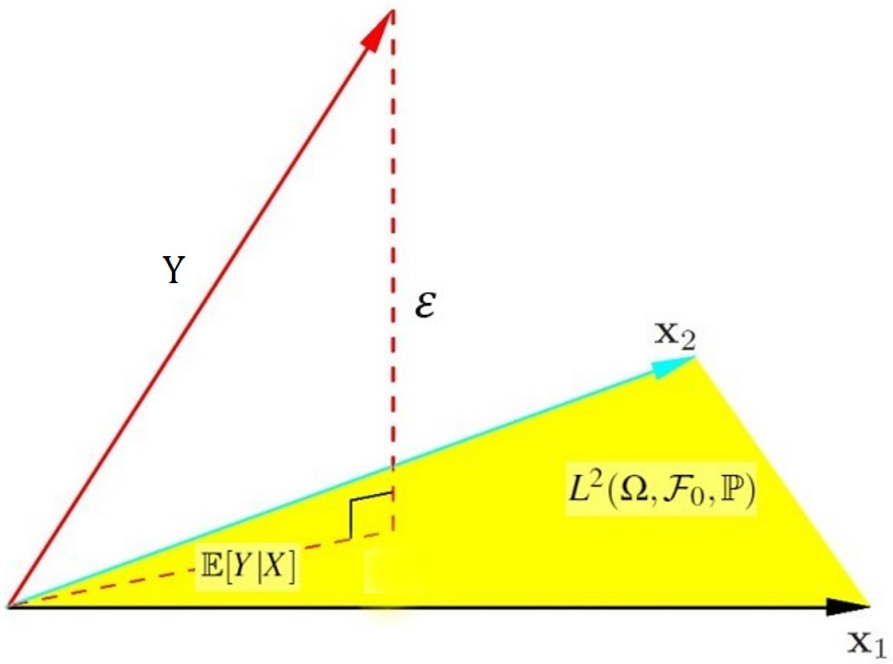


Figure: Conditional expectation as projection in L^2 space

Model Assessment

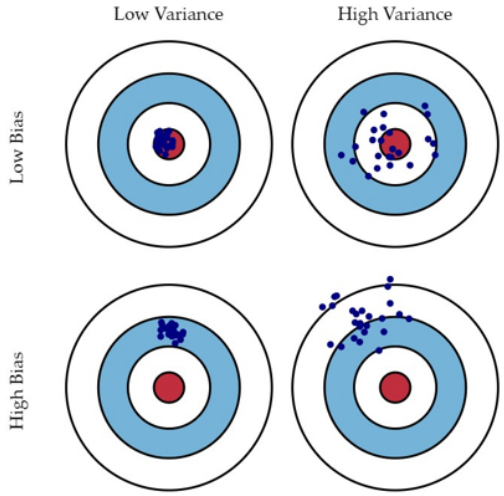
- Assessment
  - Training error
  - Test error
  - Accuracy (classification)
  - Generalization error (Generalization ability)

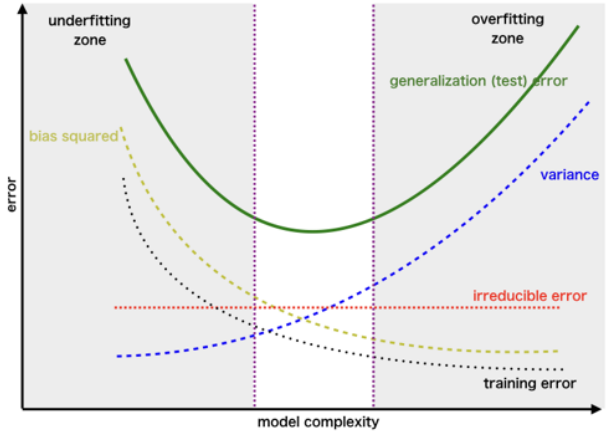
$$\mathbb{E}[L(Y, \hat{f}(X))] = \int_{\mathscr{X} \times \mathscr{Y}} L(y, \hat{f}(x)) P(dx, dy)$$

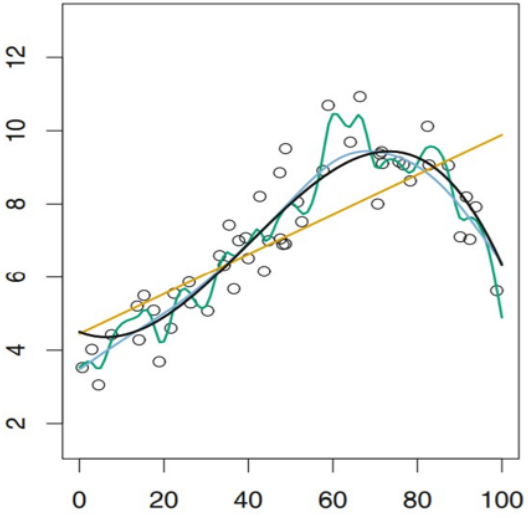
  - Completeness (Fudenberg et al. (2022), JPE)
- Bias-Variance Tradeoff

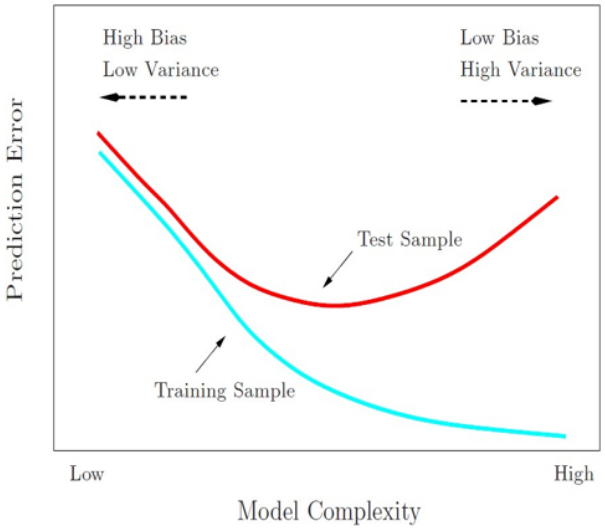
$$\begin{aligned}
&\mathbb{E}[(Y - \hat{f}(X))^2] \\
&= \mathrm{Var}(\hat{f}(X)) + \mathbb{E}[(f(X) - \mathbb{E}\hat{f}(X))^2] + \mathrm{Var}(\varepsilon) \\
&= \mathrm{Var}(\hat{f}(X)) + \mathbb{E}[Bias^2(\hat{f}(X))] + \mathrm{Var}(\varepsilon)
\end{aligned}$$

Overfitting problem







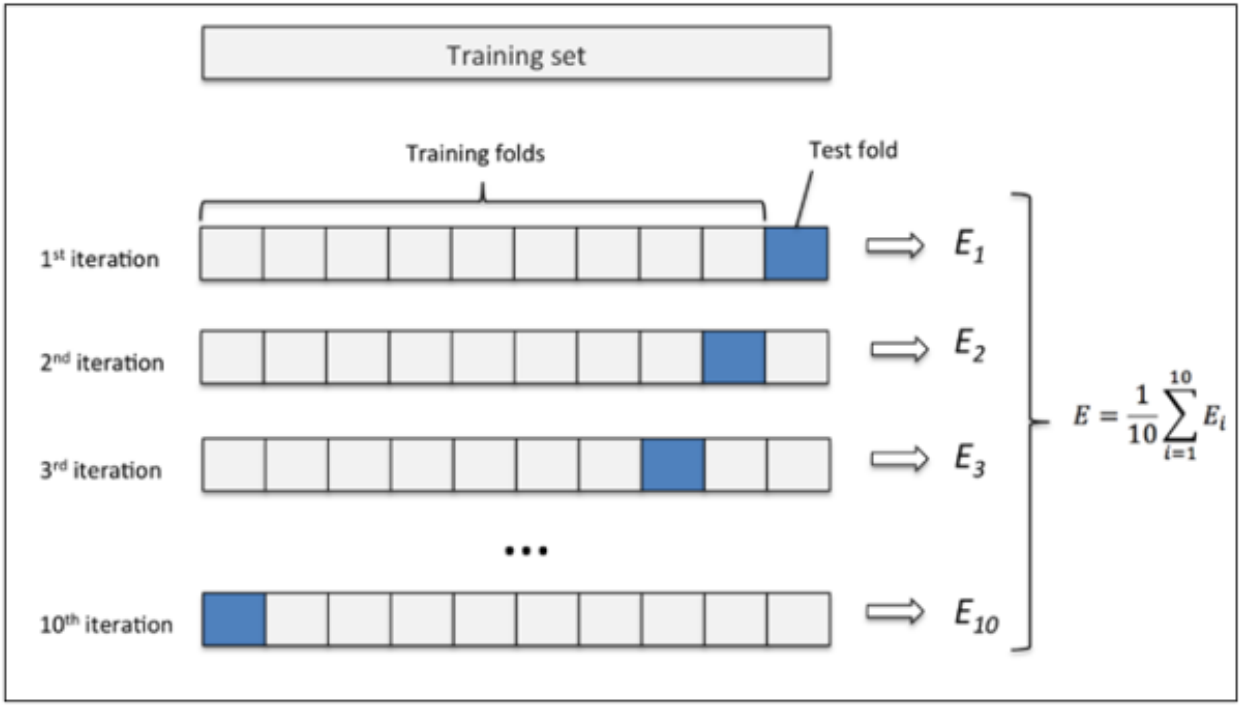


Model Selection

- Regularization (Occam's razor)

$$\min_{f \in \mathscr{S}} \frac{1}{N} \sum_{i=1}^{N} L(Y_i, f(X_i)) + \lambda J(f), \ \ \lambda \geq 0.$$

- Cross validation



Example: regularization and cross-validation

LASSO (Least Absolute Shrinkage And Selection Operator).

- Deal with high dimensional regression problem. The empirical loss is

$$EL_N(D) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \beta' x_i)^2 + \lambda \sum_{j=1}^{q} |\beta_j|, \ \lambda \geq 0.$$

- The idea is to penalize model complexity.
  - this induces bias but can reduce variance.
- LASSO sets many β to zero and shrinks remaining towards zero.
- Tuning parameter λ is most often determined by cross-validation or AIC or BIC.

Reference

- Field Cady: The Data Science Handbook. Wiley, 2017.
- Trevor Hastie, Robert Tibshirani, Jerome Friedman: The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition. Springer, 2009.
- Gareth James, Daniela Witten, Trevor Hastie and Robert Tibsharani: An Introduction to Statistical Learning: with Applications in R. Springer, 2013.
- Hang Li: Statistical Learning Methods. Tsinghua University Press, 2012. (Chinese)
- Zhihua Zhou: Machine Learning. Tsinghua University Press, 2016. (Chinese)