

Statistcs

LLN

$$\lim_{n \rightarrow \infty} \left| \frac{1}{n} \sum_{i=1}^n x_i - \mathbb{E}[x] \right| = 0$$

Unbiasedness

$$\text{Bias}_{\theta} = \mathbb{E} \left[\hat{\theta} \right] - \theta = \mathbb{E} \left[\hat{\theta} - \theta \right] = 0$$

Consistency

$$\lim_{n \rightarrow \infty} \mathbf{P} \left(\left| \hat{X} - \frac{1}{n} \sum_{i=1}^n X_i \right| \leq \epsilon \right) = 1$$

Properties of Gaussians

$$\begin{aligned} X &\sim \mathcal{N}(\mu, \sigma^2) \implies Y = aX + b \sim \mathcal{N}(a\mu + b, a^2\sigma^2) \\ X &\sim \mathcal{N}(\mu_x, \sigma_x^2), \quad Y \sim \mathcal{N}(\mu_y, \sigma_y^2), \implies Z = X + Y \sim \mathcal{N}(\mu_x + \mu_y, \sigma_x^2 + \sigma_y^2) \end{aligned}$$

Information Theory

Information

$$\mathbf{I}(S) = \log_2 \frac{1}{\mathbf{P}(S)}$$

Entropy

$$\begin{aligned} \mathbf{H}[S] &= \sum_{s \in \{S\}} \mathbf{P}(S=s) \cdot \mathbf{I}(S=s) = \sum_{s \in \{S\}} \mathbf{P}(S=s) \log_2 \frac{1}{\mathbf{P}(S=s)} \\ &= \mathbb{E} \left[\log_2 \frac{1}{\mathbf{P}(S=s)} \right] = -\mathbb{E} \left[\log_2 \mathbf{P}(S=s) \right] \\ &= -\sum_{s \in \{S\}} \mathbf{P}(S=s) \log_2 \mathbf{P}(S=s) \end{aligned}$$

Information Gain

$$\mathbf{I}(Y, X_i) = \mathbf{H}[Y] - \mathbf{H}[Y | X_i]$$

Learning Theory

PAC Learning

$$\begin{aligned} \mathbf{P} \left(\left| \hat{\theta} - \theta^* \right| \leq \epsilon \right) &\geq 1 - \delta \\ \mathbf{P} \left(\left| \hat{\theta} - \theta^* \right| > \epsilon \right) &< \delta \end{aligned}$$

Decision Theory

Risk

$$\text{Risk}(f) = R(f) := \mathbb{E}_{(X,Y)}[\text{loss}(Y, f(X))]$$

Bayes Risk

$$R(f^*) \leq R(f), \forall f$$

Bayes Optimal Rule

$$f^*(P) = \arg \min_f \mathbb{E}_{(X,Y) \sim P}[\text{loss}(Y, f(X))]$$

Emperical Risk Minimization

$$\hat{f}_n = \arg \min_f \frac{1}{n} \sum_{i=1}^n [\text{loss}(Y_i, f(x_i))] \xrightarrow[n \rightarrow \infty]{\text{LNN}} \arg \min_f \mathbb{E}_{(X,Y)}[\text{loss}(Y, f(X))]$$

Risk

True Risk

$$R(f) := \mathbb{E}_{(X,Y)}[\ell(f(X), Y)]$$

Emperical Risk

$$\hat{R}_{\mathcal{D}}(f) := \frac{1}{|\mathcal{D}|} \sum_{i \in \mathcal{D}} [\ell(f(X_i), Y_i)]$$

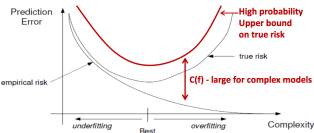
Excess Risk

$$\mathbb{E} \left[R(\hat{f}_n) \right] - R(f^*)$$

Structural Risk

$$\begin{aligned} \left| R(f) - \hat{R}_n(f) \right| &\leq C(f) \quad \forall f \in \mathcal{F} \\ R(f) &\leq \hat{R}_n(f) + C(f), \quad \forall f \in \mathcal{F} \end{aligned}$$

Use $\hat{R}_n(\hat{f}_n) + C(\hat{f}_n)$ as a *pessimistic* estimate of true risk.



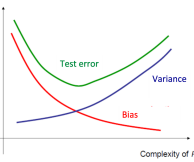
Risk Estimation

Bias–Variance Tradeoff

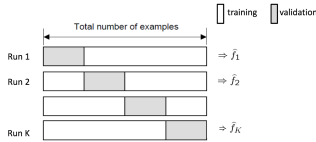
$$\text{Bias} = \mathbb{E} [f(X)] - f^*(X)$$

$$\text{Variance} = \mathbb{E} \left[(f(X) - \mathbb{E} [f(X)])^2 \right]$$

$$\mathbb{E} \left[(f(X) - f^*(X))^2 \right] = \text{Bias}^2 + \text{Variance}$$

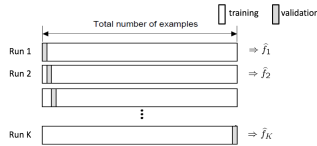


K–Fold Cross–Validation



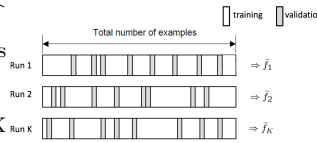
LOO Cross–Validation

Special Case of K–Fold with $K = n$ partitions.



Random Subsampling

Randomly subsample a fixed fraction αn ($0 < \alpha < 1$) of the dataset for validation. Form hold–out predictor with remaining data as training data. Repeat K times. Final Predictor is average / majority vote over K hold–out estimates



Hold–out Method

1. Split into two sets

$$\mathcal{D}_T = \{(X_i, Y_i)\}_{i=1}^m$$

training set

$$\mathcal{D}_V = \{(X_i, Y_i)\}_{i=m+1}^n$$

holdout set

2. use \mathcal{D}_T to train a predictor $\hat{f}_{\mathcal{D}_T}$
3. use \mathcal{D}_V to evaluate the predictor $\hat{R}_{\mathcal{D}_V}(\hat{f}_{\mathcal{D}_T})$

Estimating True Risk

Estimate the error of a predictor on n data points. If K is large (close to n), bias of error estimate is small since each training set has close to n data points.

However, variance of error estiamtes is high since each validation set has fewer data points and might deviate a lot from the mean.

$$\text{Error estimate} = \frac{1}{K} \sum_{k=1}^K \hat{R}_{V_k}(\hat{f}_{T_k})$$



Risk Minimization

Emperical Risk Minimization

$$\hat{f}_n = \arg \min_f \frac{1}{n} \sum_{i=1}^n n[\text{loss}(Y_i, f(x_i))]$$

Structural Risk Minimization

$$\hat{f}_n = \arg \min_{f \in \mathcal{F}} [\hat{R}_n(f) + \lambda C(f)]$$

Overfitting

Regularization

Complexity Regularization

$$\hat{f}_n = \arg \min_{f \in \mathcal{F}} \{\hat{R}_n(f) + C(f)\}$$

Information Criteria

AIC (Akiake IC) $C(f) = \#parameters$

Allows $\#$ parameters to be infinite as $\#$ training data n becomes large

BIC (Bayesian IC) $C(f) = \#parameters * \log n$

Penalizes complex models more heavily – limits complexity of models as $\#$ training data n becomes large

Model Selection

- define a finite set of model classes
- estimate true risk for each model class
- select model class with lowest estimated true risk

Model classes $\{\mathcal{F}_\lambda\}$ of increasing complexity $\mathcal{F}_1 < \mathcal{F}_2 < \dots$

$$\min_{\lambda} \min_{f \in \mathcal{F}_\lambda} J(f, \lambda)$$

- given λ estimate \hat{f}_λ using *empercal* / *structural* / *complexity regularized* risk minimization
- select λ for which \hat{f}_λ has minimum true risk estimated using *cross–validation* / *hold–out* / *information criteria*

Generalization Error

Terms

estimated predictor $:= \hat{f}_n$

optimal predictor $:= f^*$

risk of estimated predictor $:= R(\hat{f}_n)$

expected risk $:= \mathbb{E} \left[R(\hat{f}_n) \right]$

risk of optimal predictor $:= R(f^*)$

excess risk $:= \mathbb{E} \left[R(\hat{f}_n) \right] - R(f^*)$

True Risk Decomposition

$$\mathbb{E} \left[R(\hat{f}_n) \right] - R^* = \underbrace{\left(\mathbb{E} \left[R(\hat{f}_n) \right] - \inf_{f \in \mathcal{F}} R(f) \right)}_{\text{estimation error}} + \underbrace{\left(\inf_{f \in \mathcal{F}} R(f) - R^* \right)}_{\text{approximation error}}$$

Estimation Error : due to randomness of training data
Approximation Error : due to restriction of model class

A geometric diagram illustrating the decomposition of true risk. A gray shaded region represents the model class \mathcal{F} . Inside this region, a point represents the expected risk of the estimated model, $R(\hat{f}_n)$. Another point within the region represents the best model in the class, $\inf_{f \in \mathcal{F}} R(f)$. A point outside the region represents the true minimum risk, R^* . A blue line segment connects $R(\hat{f}_n)$ and $\inf_{f \in \mathcal{F}} R(f)$, labeled "Estimation error". A red line segment connects $\inf_{f \in \mathcal{F}} R(f)$ and R^* , labeled "Approx. error". A green line segment connects $R(\hat{f}_n)$ and R^* , labeled "Excess risk".

A graph showing the relationship between the complexity of the model class \mathcal{F} and three risk components: risk, estimation error, and approximation error. The x-axis is labeled "Complexity of \mathcal{F} ". The y-axis represents the risk. The "risk" curve (green) is U-shaped, representing the sum of estimation and approximation errors. The "estimation error" curve (blue) increases as complexity increases. The "approximation error" curve (red) decreases as complexity increases. The graph is divided into two stages: "Stage I: better risk estimators" on the left, where estimation error is high and approximation error is high, and "Stage II: model selection" on the right, where estimation error is low and approximation error is low.

Regression

Linear Regression

Ridge Regression

$$\hat{\theta}_{\text{MAP}} \arg \min_{\theta} \sum_{i=1}^n (Y_i - X_i \theta)^2 + \lambda \|\theta\|_2^2$$

Lasso Regression

$$\hat{\theta}_{\text{MAP}} \arg \min_{\theta} \sum_{i=1}^n (Y_i - X_i \theta)^2 + \lambda \|\theta\|_1$$

Polynomial Regression

Classification

Logistic Regression

Naive Bayes

Boosting

Decision Trees

$$\begin{aligned} \arg \max_{X_i} \left[\mathbf{H} \left[Y \right] - \mathbf{H} \left[Y \mid X_i \right] \right] &= \arg \min_{X_i} \mathbf{H} \left[Y \mid X_i \right] \\ &= \arg \min_{X_i} \sum_{x \in \{X_i\}} \left[\mathbf{P} \left(X_i = x \right) \mathbf{H} \left[Y \mid X_i = x \right] \right] \\ &= \arg \min_{X_i} - \sum_{x \in \{X_i\}} \left[\mathbf{P} \left(X_i = x \right) \sum_{y \in \{Y\}} \left[\mathbf{P} \left(Y = y \mid X_i = x \right) \log_2 \mathbf{P} \left(Y = y \mid X_i = x \right) \right] \right] \end{aligned}$$

Support Vector Machines

Deep Learning

Common Activation Functions

Backpropagation

Gradient Descent

Perceptron

MLP

CNN

RNN

LSTM

Clustering

KNN

Kernel Regression

Kernel Trick