Notes of Machine Learning Model

Xiangli Chen

Computer Science Department
University of Illinois at Chicago

September 1, 2017

Loss

Definition

Loss function:

$$L(y, f(x)) \text{ where } f : \mathcal{X} \to \mathbb{R} \quad L : \mathcal{X} \times \mathcal{Y} \times \mathbb{R} \to [0, \infty]$$

measures how good we are on a particular (x, y) pair.

Loss examples

- $0 - 1$ loss: $L(y, f(x)) = \begin{cases} 1 & y \neq f(x) \\ 0 & y = f(x) \end{cases} = 1_{\{f(x) \neq y\}}$
- L_1, L_2 loss: $|y - f(x)|, (y - f(x))^2$
- Hinge loss: $\max(0, 1 - yf(x)) = |1 - yf(x)|_+$
- $\epsilon-$insensitive loss: $\max(0, |y - f(x)| - \epsilon)$

Risk

Definition

Risk of f function (The expected loss):

$$R_{L,p}(f) = \int_{\mathcal{X}\times\mathcal{Y}} L(y, f(x)) d_{p(x,y)} = \mathbb{E}[L(Y, f(X))]$$

Risk examples

- Risk of $0-1$ loss: $\mathbb{E}[1_{\{f(X)\neq Y\}}] = p(f(x) \neq y)$
- Risk of L_2 loss: $\mathbb{E}[(Y - f(X))^2]$

Definition

Bayes Risk (minimum):

$$R^*_{L,P} = \inf_{f:\mathcal{X}\to\mathbb{R}} R_{L,p}(f) = \inf_{f:\mathcal{X}\to\mathbb{R}} \int_{\mathcal{X}\times\mathcal{Y}} L(y, f(x)) d_{p(x,y)}$$

Consistency of learning

In practive, $p(x,y)$ is unknow. Instead, training data $\mathcal{D}$ sampled from $p(x,y)$ is given.
Goal of learning:
Learn a function $f_{\mathcal{D}}$ whose risk is close to the Bayes risk $R^*_{L,p}$.
$\mathcal{D}$ is a random variable, so as $R_{L,p}(f_{\mathcal{D}})$.

Definition

Univerally Consistency:
A learning model is univerally consistent if for all distributions $p(x,y)$, the risk converges to the Bayes risk as the sample size N increases.

$$R_{L,p}(f_{\mathcal{D}_N}) \xrightarrow{p} R^*_{L,p} \text{ as } N \to \infty$$

That $\forall \epsilon > 0 \quad \lim_{N\to\infty} p(|R_{L,p}(f_{\mathcal{D}_N}) - R^*_{L,p}| \leq \epsilon) = 0$

Consistency of learning

Stone's theorem 1977:
Many classification (KNN, SVM), regression algorithms are universally consistent for certain loss functions under certain conditions.
Devroy 1982 (No Free Lunch):
For every consistent learning method and for every fixed convergence rate h_n, $\exists p(x, y)$ such that the convergence rate of this learning method on $p(x, y)$ distributed data is slower than h_n.

Emipirical risk

The distribution $p(x, y)$ of data is unknown, instead we are given training data $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$.

Definition

Empirical Risk:

$$\hat{R}_N(f) = \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i))$$

Empirical risk converges to risk:

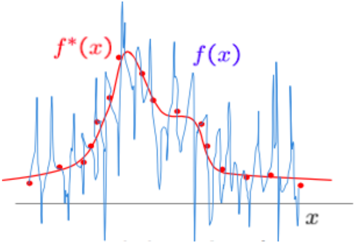
$$\lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i)) = R_{L,p}(f) \quad \text{(Law of large numbers).}$$

Overfitting

Optimize over all possible predictor $f : \mathcal{X} \to \mathbb{R}$ that

$$f_N^* = \arg \inf_{f : \mathcal{X} \to \mathbb{R}} \hat{R}_N(f) \text{ leads to overfitting.}$$

For example, overfitting in regression with ERM.



The empirical risk of $f(x)$ is zero, but the true risk of $f(x)$ is very high (large generalization error).
Prediction performance is very poor on new random test point.

Overfitting solution

Restrict all possible predictors $f : \mathcal{X} \to \mathbb{R}$ to a function set $\mathcal{F}$.
ERM over the function set $\mathcal{F}$,

$$f^*_{N,\mathcal{F}} = \arg \inf_{f \in \mathcal{F}} \hat{R}_N(f).$$

Let

$$\hat{R}^*_{n,\mathcal{F}} = \inf_{f \in \mathcal{F}} \hat{R}_N(f) = \inf_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^{N} L(y_i, f(x_i)).$$

Approximation and Estimation issues

1st issue

$$R^*_{\mathcal{F}} - R^* \geq 0$$

needs to be smalll. Solution: structural risk minimization (SRM)
Let $\mathcal{F}_n$ increases with the sample size $n \, (\mathcal{F}_{n+1} \supset \mathcal{F}_n)$, and let $\mathcal{F}_{n+1}$ contains more complex functions than $\mathcal{F}_n$.

2nd issue

$$\inf_{f \in \mathcal{F}} \hat{R}_N(f)$$

might be difficult to be optimized in f (may be non-convex).
Solution: use loss function L such that $\hat{R}_N(f)$ is convex in f
e.g. use convex hinge loss or L_2 loss to approximate $0 - 1$ non-convex loss.

Approximation and Estimation Errors

Bayes risk classifier f^*

$$f^*(x) = \arg \inf_{f:\mathcal{X}\to\mathbb{R}} R(f)$$

Risk of a classifier f

$$R(f) - R(f^*) = \underbrace{R(f) - \inf_{f\in\mathcal{F}} R(f)}_{\text{Estimation error}} + \underbrace{\inf_{f\in\mathcal{F}} R(f) - R(f^*)}_{\text{Approximation error}}$$

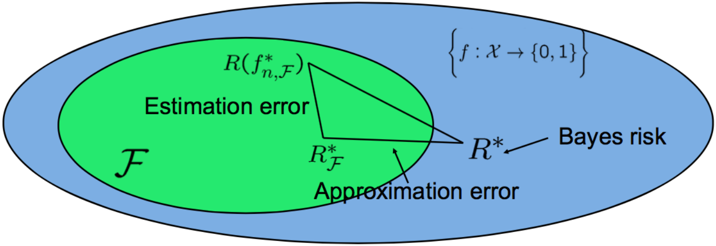
Approximation and Estimation Errors

Ultimate goal: $R(f^*_{n,\mathcal{F}}) - R^* = 0$

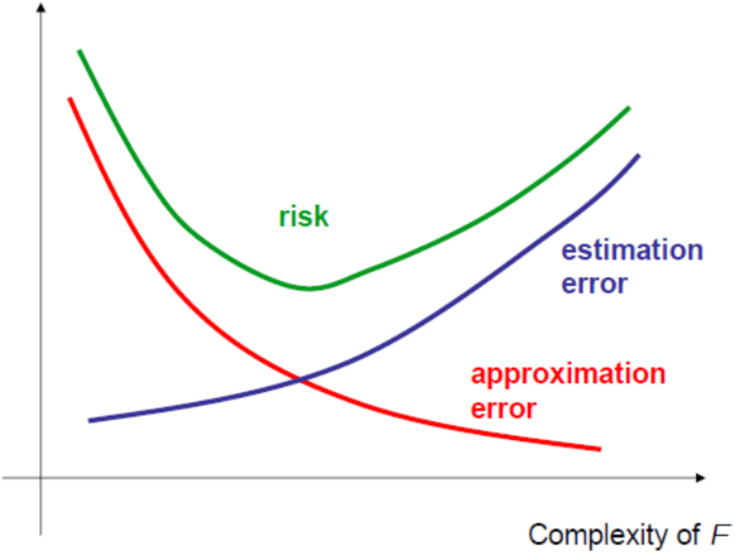
Risk of $f^*_{n,\mathcal{F}}$

$$R(f^*_{n,\mathcal{F}}) - R^* = \underbrace{R(f^*_{n,\mathcal{F}}) - R^*_{\mathcal{F}}}_{\text{Estimation error}} + \underbrace{R^*_{\mathcal{F}} - R^*}_{\text{Approximation error}}$$

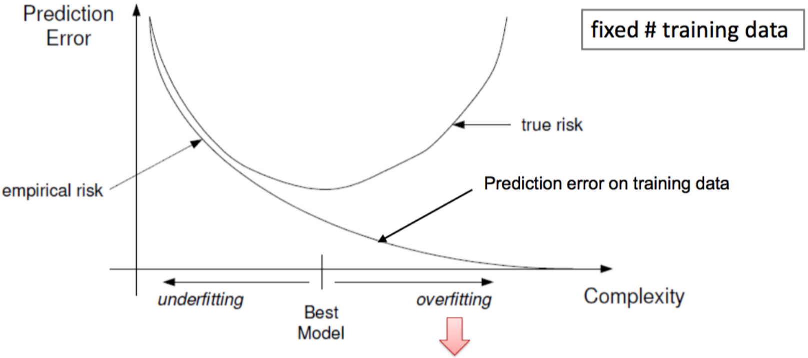
Big picture-$\mathcal{F}$ controls the model complexity



Effect of Model Complexity



Effect of Model Complexity



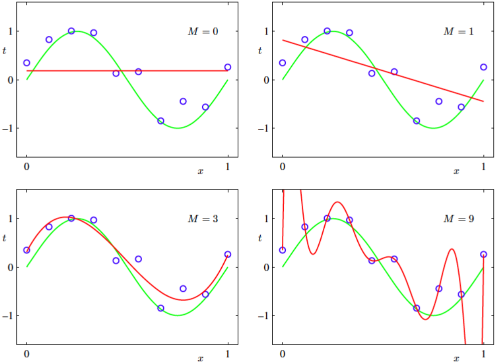
Empirical risk is no longer a good indicator of true risk

Overfitting and Underfitting-Regerssion

Target value $\{t_n\}$ are obtained via:

$$t(x) = f^*(x) + \mathcal{N}(0, 0.3^2)$$

where $f^*(x) = sin(2\pi x)$.
Assume polynomail predictor for learning: $f_{\mathcal{F}}(x) = \sum_{j=0}^{M} \omega_j x^j$.



Risk Bound Analysis

Come in future

Loss Function

Risk of $f(x)$ under L_2 (square) loss:

$$\mathbb{E}[(Y - f(X))^2]$$

The function minimize the risk is:

$$f^*(x) = \arg\inf_f \int (y - f(x))^2 d_{p(x,y)}$$

Using calculus of variation to give

$$\frac{\partial \mathbb{E}}{\partial f(x)} = 2 \int (y - f(x)) p(x, y) d_y = 0$$

Solving for $f(x)$, we obtain

$$f^*(x) = \frac{\int y p(x, y) dy}{p(x)} = \mathbb{E}[Y|x]$$

Expected Loss

We can expand the square loss as:

$$\begin{aligned}(f(x)-y)^2 &= (f(x)-f^*(x)+f^*(x)-y)^2\\ &= (f(x)-f^*(x))^2 + 2(f(x)-f^*(x))(f^*(x)-y)\\ &\qquad + (f^*(x)-y)^2\end{aligned}$$

Substituting into the loss function and performing the integral over y, we obtain (the cross-term vanishes):

$$R(f) = \mathbb{E}[L] = \mathbb{E}[(f(X)-f^*(X))^2] + \underbrace{\mathbb{E}[(f^*(X)-Y)^2]}_{R^*}$$

The Bayes risk R^* is independent of $f(x)$ which represents the irreducible minimum value of the loss function.

Biase and Variance

Let $\hat{f}_{\mathcal{D},\mathcal{F}}$ (denoted by $\hat{f}$) learned from training data $\mathcal{D}$ over class $\mathcal{F}$. $\mathcal{D}$ is a random variable so as $\hat{f}_{\mathcal{D},\mathcal{F}}$

$$(\hat{f}(x) - f^*(x))^2 = (\hat{f}(x) - \mathbb{E}_{\mathcal{D}}[\hat{f}(x)] + \mathbb{E}_{\mathcal{D}}[\hat{f}(x)] - f^*(x))^2$$

Take the expectation with respect to $\mathcal{D}$ gives the bias and variance of a single input x

$$\mathbb{E}_{\mathcal{D}}[(\hat{f}(x) - f^*(x))^2] = \underbrace{(\mathbb{E}_{\mathcal{D}}[\hat{f}(x)] - f^*(x))^2}_{(\text{bias})^2} + \underbrace{\mathbb{E}_{\mathcal{D}}[(\hat{f}(x) - \mathbb{E}_{\mathcal{D}}[\hat{f}(x)])^2]}_{\text{variance}}$$

Biase and Variance

Expected square loss = $(\text{bias})^2$+variance+noise

$$\begin{aligned} R(\hat{f}_{\mathcal{D},\mathcal{F}}) &= \mathbb{E}_{X,Y,\mathcal{D}}[(\hat{f}_{\mathcal{D},\mathcal{F}}(X) - Y)^2] \\ &= \mathbb{E}_{X,\mathcal{D}}[(\hat{f}_{\mathcal{D},\mathcal{F}}(X) - f^*(X))^2] + \underbrace{\mathbb{E}_{X,Y}[(f^*(X) - Y)^2]}_{noise} \\ &= \underbrace{\mathbb{E}_X[\mathbb{E}_{\mathcal{D}}[(\hat{f}_{\mathcal{D},\mathcal{F}}(X)] - f^*(X))^2]}_{(\text{bias})^2} \\ &\qquad + \underbrace{\mathbb{E}_{X,\mathcal{D}}[(\hat{f}_{\mathcal{D},\mathcal{F}}(X) - \mathbb{E}_{\mathcal{D}}[(\hat{f}_{\mathcal{D},\mathcal{F}}(X)])^2]}_{variance} + \text{noise} \end{aligned}$$

- Large bias, small variance-poor approximation but stable
- Small bias, large variance-good approximation but instable

Biase and Variance

Generate data

$$t = \sin(2\pi x) + \mathcal{N}(0, 0.3^2) \quad x \sim \mathit{uniform}(0, 1)$$

Regularized least squares with Gaussian basis functions

$$\hat{R}_{N,\mathcal{F}} = \sum_{i=1}^{N} (y_i - \omega^T \Phi(x_i))^2 + \frac{\lambda}{2} \omega^T \omega \text{ where } \Phi_i(x) = e^{-\frac{(x-\mu_i)^2}{2s^2}}$$

Biase and Variance

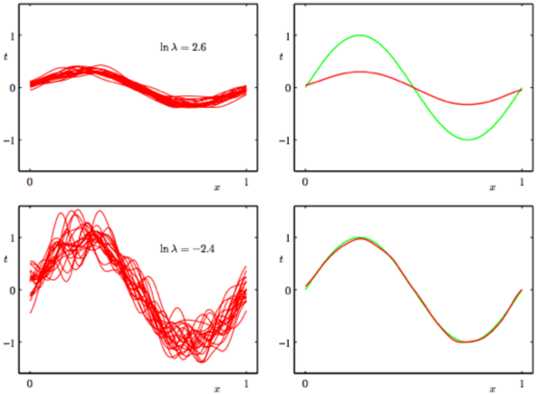


Figure 1: 100 data sets, each having 25 data points. 24 Gaussian basis functions (25 parameters including the bias one). The left column shows the result of fitting the model for various values of $\ln \lambda$ (only 20 of the 100 fits are shown). The right column shows the corresponding average of the 100 fits (red) along with the sinusoidal function from which the data sets were generated (green).

Illustration

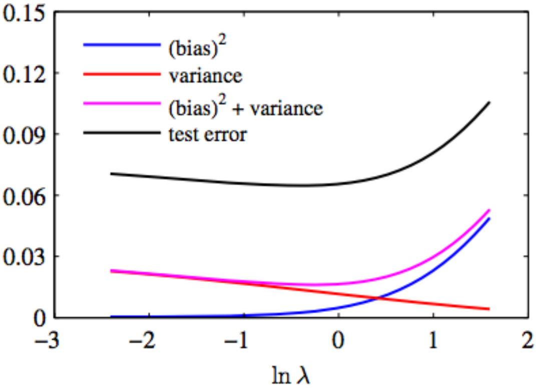


Figure 2: Plot of squared bias and variance, together with their sum, corresponding to the result shown in last figure. Also shown is the average test set error for a test data size of 1000 points. The minimum value of $\text{biase}^2 + \text{variance}$ occurs around $\ln \lambda = -0.31$, which is close to the value that gives the minimum error on the test data.

Examples of Model Spaces

Model spaces with increasing complexity:

- Nearest neighbor classifiers with varying neighborhood sizes $k = 1, 2, 3, \cdots$

  Small neighborhood → Higher complexity

- Decision trees with depth k or with k leaves

  Higher depth/more leaves → Higher complexity

- Regression with polynomials of order $k = 0, 1, 2, \cdots$

  Higher degree → Higher complexity

- Kernel regression with bandwidth h

  Small bandwidth → Higher complexity

Model Selection

Setup:
Model classes $\{\mathcal{F}_\lambda\}_{\lambda \in \Lambda}$ of increasing complexity $\mathcal{F}_1 \subset \mathcal{F}_2 \subset \cdots$
Select:
Right complexity model in a data-driven/adaptive way:

- Hold-out
- Cross-validation
- Information criteria-AIC, BIC, minimum description length (MDL)

Hold-out

Hold-out procedure:
Suppose we are given dataset $\mathcal{D}$

1. Split dataset $\mathcal{D}$ into two sets Training set $\mathcal{D}_T$ and Validation set $\mathcal{D}_v$
2. Use $\mathcal{D}_T$ for training a predictor from each model class

$$\hat{f}_\lambda = \arg\min_{f \in \mathcal{F}_\lambda} \hat{R}_T(f) \quad \lambda \in \Lambda$$

3. Use $\mathcal{D}_v$ to select the model class giving the smallest empirical error

$$\hat{\lambda} = \arg\min_{\lambda \in \Lambda} \hat{R}_V(\hat{f}_\lambda)$$

4. Hold-out predictor

$$\hat{f} = \hat{f}_{\hat{\lambda}}$$

Hold-out Drawbacks

Drawbacks:

- Data is not sufficient to afford setting one subset aside for getting a sense of generalization abilities
- Validation error may be misleading if we get an "unforunate" split

Limitations of hold-out can be overcome by a family of random sub-sampling methods at the expense of more computation

Cross Validation

K-fold cross validation

- Create K-fold partition of the dataset
- Form K hold-out predictors, each time using one partition as validation and the rest K-1 as training datasets
- Final predictor is average/majority vote over the K hold-out estimates

Leave-one-out (LOO) cross-validation (special case of K-fold with $k = n$ partitions)

- Train on $n - 1$ samples and validate on only one sample per run for n runs

Random subsampling

Random sbusampling

- Randomly subsample a fixed fraction $\alpha n (0 < \alpha < 1)$ 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 the K hold-out estimate.

Optimal Encoding

Optimal prefix-free encoding of sending messages a, b, c, d:

distribution	1/2	1/4	1/8	1/8
prefix-free code	0	10	110	111

Length of optimal prefix-free encoding is close to

$$-\log p(x)$$

Minimum expected prefix-free encoding length

$$\mathbb{E}_p[-\log p(X)] \leq \underbrace{\mathbb{E}_p[-\log q(X)]}_{\text{expected log loss}}$$

Expected log loss measures the "distance" of p and q

Conditional Log Loss

Let $p(y|x)$ be the real distribution and $q(y|x)$ be the estimator $x \sim p(x)$
Expected conditional log loss (can extend to continuous case)

$$R_{L,p}\mathbb{E}[-\log q(Y|X)] = -\sum_x \sum_y p(x)p(y|x)\log q(y|x)$$

Empirical conditional log loss given training data $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$

$$\hat{R}_{L,p} = \frac{1}{N}\sum_{i=1}^{N} -\log q(y_i|x_i)$$

Optimal estimator over a set $\mathcal{Q}$ of estimators (minimize loss)

$$q^* = \arg\min_{q\in\mathcal{Q}} \hat{R}_{L,p}(q)$$

It is equivalent to maximizing conditional log-likelihood

Conditional Log Likelihood

Assume random sample (i.i.d) $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$ and the estimator $p(y|x,\theta)$ takes a parametric form.
Conditional log-likelihood

$$L(\theta) = \log p(y_{t=1}^N | x_{i=1}^N, \theta) = \sum_{i=1}^N \log p(y_i | x_i, \theta)$$

Maximize conditional log likelihood - discrimimative approach

$$\arg\max L(\theta) = \arg\min -L(\theta)$$

where $-L(\theta)$ is the empirical conditional log loss.

Example

- logistic regression
- linear regression under Gaussian assumption