Note for Machine Learning

Chenmi'en Tan
E-mail: chenmientan@outlook.com
Actively Updating (Last update: June 25, 2021)

Abstract

This is a personal note covers some of the topics discussed in *Pattern Recoginition and Machine Learning*. Beyesian approaches as well as their approximations, RVM, graphical models, sampling methods, and LDS et al. are not included.

Contents

0.1 Bias-variance Decomposition

0.2 Cross Validation

1 Linear Regression

The hypothesis of linear regression is that the responses rely on independent Guassian distributions where the means have linear relationship with the predictors. By denoting $\mathbf{X} \in \mathbb{R}^{N\times(p+1)}$ as the predictor matrix where each row is a sample and each column is a predictor (all items in the first column is 1, representing the bias) and $\mathbf{t} \in \mathbb{R}^N$ as the corresponding response vector, the assumption can be mathematically expressed as $\mathbf{t} \sim N(\mathbf{Xw}, \sigma^2\mathbf{I})$. Here we assume that $\mathbf{w}$ is an unknown constant vector. To estimate $\mathbf{w}$, we demand the likelihood function, which is

$$L(\mathbf{w}, \sigma^2|\mathbf{t}) = (2\pi)^{-N/2}\sigma^{-N}\exp(-\frac{1}{2\sigma^2}(\mathbf{Xw}-\mathbf{t})^T(\mathbf{Xw}-\mathbf{t}))$$

It can be observed that maximizing L respects to $\mathbf{w}$ is equivalent to minimizing the empirical residual sum of squares $J(\mathbf{w}) = (\mathbf{Xw}-\mathbf{t})^T(\mathbf{Xw}-\mathbf{t})$. This is a convex optimization problem since $\nabla^2 J = 2\mathbf{X}^T\mathbf{X} \succeq \mathbf{0}$. Thus, the optimal solution is obtained when $\nabla J = 2\mathbf{X}^T(\mathbf{Xw}-\mathbf{t}) = \mathbf{0}$. If $\mathbf{X}$ is full column ranked, the unique solution can be further derived as $\hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{t}$.

The estimation for $\mathbf{w}$ would leads to response estimation for newly observed predictor. For instance, when encountering $\mathbf{x} \in \mathbb{R}^{p+1}$ as the new sample, a natrual estimation for the response is $\mathbf{x}^T\hat{\mathbf{w}}$. The superiority of such an estimation is that it has the smallest expected error among the family of unbiased linear estimations (with the form of $\mathbf{a}^T\mathbf{t}$).

Guassian-Markov theorem Since $\mathbb{E}[\mathbf{x}^T\hat{\mathbf{w}}] = \mathbf{x}^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{X}\mathbf{w} = \mathbf{x}^T\mathbf{w}$ we see $\mathbf{x}^T\hat{\mathbf{w}}$ is unbiased. Assume that $\mathbf{a}^T\mathbf{t}$ is an unbiased linear estimator for $\mathbf{x}^T\mathbf{w}$, then we have $\mathbb{V}[\mathbf{x}^T\hat{\mathbf{w}}] \leq \mathbb{V}[\mathbf{a}^T\mathbf{t}]$. To find this, firstly notice that $\mathbb{E}[\mathbf{a}^T\mathbf{t}] = \mathbf{a}^T\mathbf{Xw} = \mathbf{x}^T\mathbf{w}$ implies $\mathbf{a}^T\mathbf{X} = \mathbf{x}^T$. Hence

$$\begin{aligned}
\mathbb{V}[\mathbf{a}^T\mathbf{t}] &= \mathbb{V}[\mathbf{a}^T\mathbf{t} - \mathbf{x}^T\hat{\mathbf{w}} + \mathbf{x}^T\hat{\mathbf{w}}] \\
&= \mathbb{V}[\mathbf{a}^T\mathbf{t} - \mathbf{x}^T\hat{\mathbf{w}}] + 2\mathrm{Cov}(\mathbf{a}^T\mathbf{t} - \mathbf{x}^T\hat{\mathbf{w}}, \mathbf{x}^T\hat{\mathbf{w}}) + \mathbb{V}[\mathbf{x}^T\hat{\mathbf{w}}] \\
&\geq 2\mathrm{Cov}[(\mathbf{a}^T - \mathbf{x}^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T)\mathbf{t}, \mathbf{x}^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{t}] + \mathbb{V}[\mathbf{x}^T\hat{\mathbf{w}}] \\
&= 2\sigma^2[\mathbf{a}^T - \mathbf{x}^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T]\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x} + \mathbb{V}[\mathbf{x}^T\hat{\mathbf{w}}] = \mathbb{V}[\mathbf{x}^T\hat{\mathbf{w}}]
\end{aligned}$$

Multiple outputs The outcome remains the same for the case of multiple responses, even if these responses are correlated. Notice that $\mathbf{t}_i \sim$

$N(\mathbf{x}_i^T\mathbf{W}, \Sigma), \forall i = 1, \ldots, N$, thus the likelihood function is

$$L(\mathbf{W}, \Sigma|\mathbf{T}) = (2\pi)^{-NK/2} |\Sigma|^{-N/2} \exp(-\frac{1}{2}\sum_{i=1}^{N}(\mathbf{x}_i^T\mathbf{W} - \mathbf{t}_i)\Sigma^{-1}(\mathbf{x}_i^T\mathbf{W} - \mathbf{t}_i)^T)$$

Again, maximizing L respects to $\mathbf{W}$ is equivalent to minimize

$$J(\mathbf{W}) = \sum_{i=1}^{N}(\mathbf{x}_i^T\mathbf{W} - \mathbf{t}_i)\Sigma^{-1}(\mathbf{x}_i^T\mathbf{W} - \mathbf{t}_i)^T$$

Here we have

$$\begin{aligned}
\mathrm{d}J &= 2\sum_{i=1}^{N}\mathbf{x}_i^T\mathrm{d}\mathbf{W}\Sigma^{-1}(\mathbf{x}_i^T\mathbf{W} - \mathbf{t}_i)^T = 2\sum_{i=1}^{N}\mathrm{tr}[\mathbf{x}_i^T\mathrm{d}\mathbf{W}\Sigma^{-1}(\mathbf{x}_i^T\mathbf{W} - \mathbf{t}_i)^T] \\
&= 2\sum_{i=1}^{N}\mathrm{tr}[\Sigma^{-1}(\mathbf{x}_i^T\mathbf{W} - \mathbf{t}_i)^T\mathbf{x}_i^T\mathrm{d}\mathbf{W}] = 2\mathrm{tr}\{\Sigma^{-1}\sum_{i=1}^{N}[(\mathbf{x}_i^T\mathbf{W} - \mathbf{t}_i)^T\mathbf{x}_i^T]\mathrm{d}\mathbf{W}\} \\
&= 2\mathrm{tr}[\Sigma^{-1}(\mathbf{XW} - \mathbf{T})^T\mathbf{X}\mathrm{d}\mathbf{W}]
\end{aligned}$$

Thus $\nabla J = 2\Sigma^{-1}(\mathbf{XW} - \mathbf{T})^T\mathbf{X} = 0 \Leftrightarrow \hat{\mathbf{W}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{T}$, as we desire.

1.1 Regularization: Ridge and Lasso

L^2 and L^1 regularization items can be introduced to relieve overfitting, where the objective functions are

$$J^{\mathrm{Ridge}}(\mathbf{w}) = (\mathbf{Xw} - \mathbf{t})^T(\mathbf{Xw} - \mathbf{t}) + \lambda\|\mathbf{w}\|_2^2$$

$$J^{\mathrm{Lasso}}(\mathbf{w}) = (\mathbf{Xw} - \mathbf{t})^T(\mathbf{Xw} - \mathbf{t}) + \lambda\|\mathbf{w}\|_1$$

respectively. Both of these two optimization problems are convex. However, only the ridge regression one can be solved analytically, i.e., $\hat{\mathbf{w}}^{\mathrm{Ridge}} = (\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{t}$. From the perspective of

1.2 Data Reduction: PCR and PLS

2 Linear Classification

2.1 Generative Model: Discriminant Analysis

2.2 Discriminant Model: Logistic Regression

2.3 Data Reduction: Fisher Discriminant Analysis

2.4 Perceptron

3 Kernel Methods

3.1 Moving into Higher Dimension: Kernel Ridge Regression

3.2 Nadaraya-Watson Model

3.3 Sparsity: Support Vector Machine

3.4 Gaussian Process Regression

4 Expectation Maximization

4.1 Gaussian Mixture

4.2 Hidden Markov Chain

5 Pricipal Component Analysis

6 Neural Networks

7 Ensemble Learning

Algorithm 1: AdaBoost

Initialize sample weight $w_n^{(1)} = 1/N, \forall n = 1, \ldots, N$;
for $m = 1, \ldots, M$ **do**
  Train classifier $y_m(\cdot)$ by minimizing $J_m = \sum_{n=1}^N w_n^{(m)} 1_{y_m(\mathbf{x}_n) \neq t_n}$
  Compute $\epsilon_m = J_m / \sum_{n=1}^N w_n^{(m)}$ and $\alpha_m = \eta \log \frac{1-\epsilon_m}{\epsilon_m}$
  Update sample weight $w_n^{(m+1)} = w_n^{(m)} \exp(\alpha_m 1_{y_m(\mathbf{x}_n) \neq t_n})$
end

The weight remains unchanged if the sample is correctly classified and increases if the sample is misclassified.