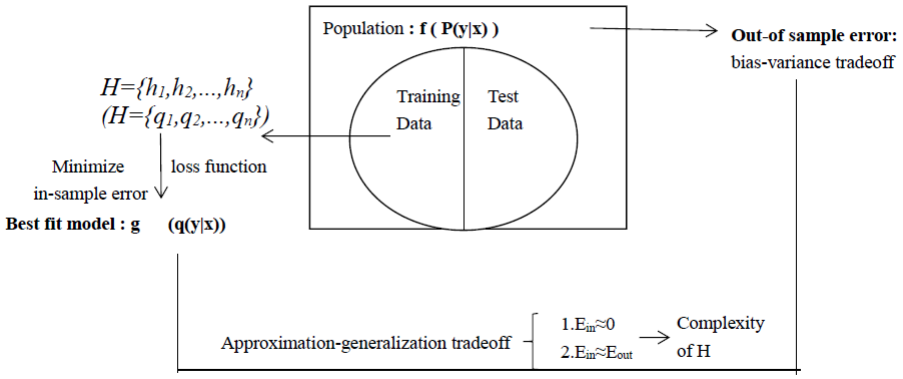
Foundations of Statistical Learning

Rao Sihang 15220162202309

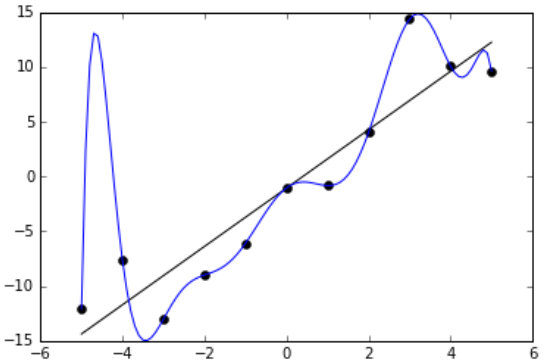
March 17, 2019

General framework of this lecture



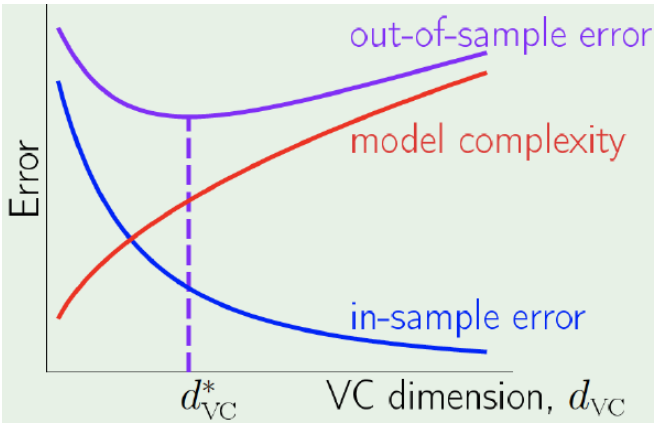
Approximation-generalization tradeoff

In most cases, we cannot see the underlying population and we can only build models based on limited sample data to investigate the correlation or casuality among some variables.Then, the criterion of model we choose is crucial. On the one hand, we hope the model we pick can perform well on those data we can observe, which means the model we structures should minimize the in-sample error as much as possible. On the one hand, as the complexity of model goes up, another problem comes up. If a complicated model can perform extremely well on those sample data with almostly zero in-sample error,it is likely to be unsuitable for other data(out-of-sample error might be considerable large).



In the figure above,although the polynomial function presented by bule line exactly fits those nosiy data, obviously, this model might perform worse than the black straight line on another dataset if the model only "memorize" training data rather than "learning" to generalize from a trend.

Accoding to VC inequality,$E_{out} \leq E_{in} + \Omega_{(d_{vc})}$,$\Omega_{(d_{vc})}$named generalization error can be viewed as the penalty of model complexity, as the model complexity increases, generalization error tends to rise even though the approximation error(in-sample error) decreases. And our goal is to find optimal model complexity with minimun out-of-sample error.



Overfitting in machine learning

Overfitting is the production of an analysis that corresponds too closely or exactly to a paticular set of data, and may therefore fail to additional data or predict future observations reliably.Overfitting is more common in machine learning, because machine learning can cope with highly complicated problems with ability to fit noise.How to avoid overfitting?

- Collect more training data. Based on a large dataset, model can improve itself.

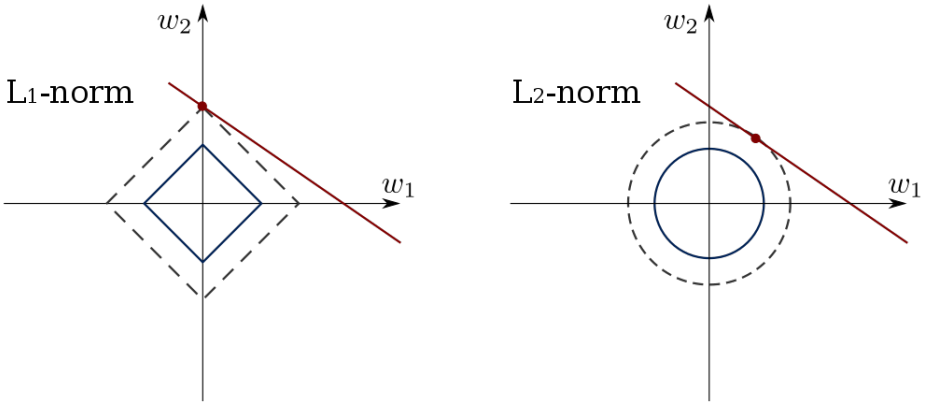
- Reduce the complexity of model.
- Regularization. In order to aviod excessive parameters and complexity, we can add a regularization term to a loss function:

$$min \sum L(h(x_i), f(x_i)) + \lambda h(g)$$

$\lambda$is a parameter controlling the importance of the regularization term.$h(g)$ is typically chosen to impose a penalty on the complexity of hypothesis model. In linear regression model, the model with L_1norm as restriction named LASSO(Least absolute shrinkage and selection operator) regression., with L_2norem as constraint named Ridge regression. The loss function in linear regression is

$$min\ L(\beta) = \frac{1}{N} \sum_{i=1}^{N} [y_i - f(x_i)]^2$$

Then, we can add $||\beta||_1 < \theta$or $||\beta||_2 < \theta$ to restrict those parameters.



- Early stopping. Just simply stop the training before overfitting has a chance to occur.
- Dropout. At each training stage, a unit is either "dropped out "of the net wih probability 1-p or kepy with probability with P. Then, only a reduced network with subsample data is trained on the data in that stage, and the removed units are reinserted into the network with their original weigts.

Bias-variance tradeoff

Bias-variance tradeoff is another way to understand approximation-generalization tradeoff. When we choose the "best"model g from a series of hypotheses H, and apply this model g to the another dataset or population, we face the expected out-of-sample error with bias and variance of g.

$$\begin{aligned} E_{out}(g) &= E\left[(g(x) - f(x))^2\right] \\ &= Var(g(x)) + E\left[(g(x) - f(x))\right]^2 \\ &= Var(g) + [\text{bias}(g)]^2 \end{aligned}$$

Similarly, as the complexity of g increases, bias of g tends to decline,but the variance will increases. So we have to choose the optimal complexity to minimize out-of-sample error.

There are several methods to cope with problem.

- Boosting. Boosting is a machine learning ensemble meta-algorithm for primarily reducing bias.most boosting algorithms consist of iteratively learning weak classifiers with respect to a distribution and adding them to a final strong classifier. When they are added, they are typically weighted in some way that is usually related to the weak learners' accuracy. After a weak learner is added, the data weights are readjusted, known as "reweighting". Misclassified input data gain a higher weight and examples that are classified correctly lose weight.[nb 1] Thus, future weak learners focus more on the examples that previous weak learners misclassified

- Bagging, also called Bootstrap aggregating. This method is designed to reduce the variance of model and ensure the stability of model. Booststrap means randomly and with replacement sampling from training data to form another new dataset.

References

[1] The main reference of those contents above is Wikipedia.