Model Selection and Validation

Peng Lingwei

April 17, 2019

Contents

11 Model Selection and Validation

In this Chapter we will present two approaches for model selection.

- Structural Risk Minimization;
- Validation.

In this Chapter, we also consider WHAT TO DO IF LEARNING FAILS.

11.1 MODEL SELECTION USING SRM

The SRM paradigm has been described and analyzed in Section 7.2.

Consider a countable sequence of hypothesis classes $\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, \ldots$. $\forall d$, the $\mathcal{H}_d$ enjoys the uniform convergence property

$$m^{UC}_{\mathcal{H}_d}(\epsilon, \delta)(\epsilon, \delta) \leq \frac{g(d)\log(1/\delta)}{\epsilon^2} \tag{11.1}$$

We reuse $w(n) = \frac{6}{n^2\pi^2}$ in chapter 7, we get

$$L_{\mathcal{D}}(h) \leq L_S(h) + \sqrt{\frac{g(d)(\log(1/\delta) + 2\log(d) + \log(\pi^2/6))}{m}} \tag{11.2}$$

The upper bound given in Equation (11.2) is pessimistic.

11.2 VALIDATION

11.2.1 Hold Out Set

Formally, let $V = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{m_v}, y_{m_v})\}$ be a set of validation set. We have

Theorem 11.1. *Let h be some predictor and assume that the loss function is in $[0, 1]$. Then, $\forall \delta \in (0, 1)$, we have,*

$$\mathbb{P}\left\{ |L_V(h) - L_{\mathcal{D}}(h)| \leq \sqrt{\frac{\log(2/\delta)}{2m_v}} \right\} \geq 1 - \delta.$$

This is tighter than the usual bounds we have seen so far. The reason for the tightness of this bound is that it is in terms of an estimate on a fresh validation set that is independent of the way h was generated. (Compare with theorem 6.8)

$$L_{\mathcal{D}}(h) \leq L_S(h) + \sqrt{C\frac{d + \log(1/\delta)}{m}}.$$

For validation set can be seen as partitioning random set into two parts, so we often refer it as a *hold out set*.

11.2.2 Validation for Model Selection

Validatiun can be naturally used for model selection.

Theorem 11.2. *Let $\mathcal{H} = h_1, \ldots, h_r$ be an arbitrary set of predictors and assume that the loss function is in $[0, 1]$ be arbitrary set of predictors and assume that the loss function is in $[0, 1]$. Assume that a validation set V of size m_v is sampled independent of $\mathcal{H}$. Then,*

$$\forall h \in \mathcal{H}, \mathbb{P}\left\{ |L_{\mathcal{D}} - L_V(h)| \leq \sqrt{\frac{\log(2|\mathcal{H}/\delta)}{2m_v}} \right\} \geq 1 - \delta.$$

11.2.3 The Model-Selection Curve

In polynomial fitting problem, the training error is monotonically decreasing as we increase the polynomial degree. On the other hand, the validation error first decreases but then starts to increase, which indicates that we are starting to suffer from overfitting.

11.2.4 k-Fold Cross Validation

leave-one-out (LOO). k-Fold cross validation is often used for model selection (or parameter tuning).

- Rigorously understanding the exact behavior of cross validation is still an open problem;
- Rogers and Wagner have shown that for k local rules (kNN) the cross validation gives a very good estimate of the true error;
- Other paper show that cross validation works for stable algorithms.

Algorithm 1 k-Fold Cross Validation for Model Selection

Require: training set $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$, set of parameter values Θ, learning algorithm A, integer k
  partition S into $S_1, S_2, \ldots, S_k$
  for $\theta \in \Theta$ **do**
    for i=1 ... k **do**
      $h_i, \theta = A(S\S_i; \theta)$
    end for.
    $error(\theta) = \frac{1}{k} \sum_{i=1}^{k} L_{S_i}(h_{i,\theta})$
  end for.
  return $\theta^* = \arg\min_\theta [error(\theta)], h_{\theta^*} = A(S; \theta^*)$

11.2.5 Train-Validation-Test Split

In most practical applications, we split the avaliable examples into three sets.

- Training set
- Validation set
- Test set

11.3 WHAT TO DO IF LEARNING FAILS

Main approaches for fixing:

- Get a larger sample
- Change the hypothesis class: enlarging it; reducing it; completely changing it; changing the parameters you consider.
- Change the feature representation of the data
- Change the optimization algorithm used to apply your learning rule

Approximation error: $L_{\mathcal{D}}(h^*)$, for $h^* \in \arg\min_{h \in \mathcal{H}} L_{\mathcal{D}}(h)$
Estimation error: $L_{\mathcal{D}}(h_S) - L_{\mathcal{D}}(h^*)$

Large approximation error: enlarge the hypothesis class or completely change it; change the feature representation of the data.

Large estimation error: enlarge sample set; reducing the hypothesis class.

A different error decomposition.

$$L_{\mathcal{D}}(h_S) = (L_{\mathcal{D}}(h_S) - L_V(h_S)) + (L_V(h_S) - L_S(h_S)) + L_S(h_S).$$

- $(L_{\mathcal{D}}(h_S) - L_V(h_S))$ can be bounded quite tightly using Theorem 11.1.
- $(L_V(h_S) - L_S(h_S))$ is large we say that our algorithm suffers from "overfitting". (note good estimate)
- $L_S(h_S)$ is large we say that our algorithm suffers from "underfitting". (not good estimate)

We write

$$L_S(h_S) = (L_S(h_S) - L_S(h^*)) + (L_S(h^*) - L_{\mathcal{D}}(h^*)) + L_{\mathcal{D}}(h^*).$$

- $L_S(h_S) - L_S(h^*) \leq 0$ for $ERM_{\mathcal{H}}$ hypothesis.
- $(L_S(h_S) - L_{\mathcal{D}}(h^*))$ can be bounded quite tightly (as in Theorem 11.1).
- $L_{\mathcal{D}}(h^*)$ is approximation error.

So $L_S(h_S)$ is a necessary but not sufficient estimator for $L_{\mathcal{D}}(h^*)$. For example, when $m < VCdim(\mathcal{H})$, we have $L_S(h_S)$ is 0, but $L_{\mathcal{D}}(h^*)$ is high.

By learning curve:

- If we see that the validation error is high and decreases with the training set but training error is zero, we increase the number of examples or decrease the complexity of the hypothesis class.
- If the validation error is kept around 1/2 then we have no evidence that the approximation error of $\mathcal{H}$ is good. so we need increase the complexity of the hypothesis class or completely change it.

11.4 SUMMARY

1. Plot the model-selection curve for parameter tuning.
2. Large training error: enlarging the hypothesis class, completely change it, or change the feature representation of the data.
3. Training error is small: plot learning curves and try to deduce from them whether the problem is estimation error or approximation error.
4. Large estimation error and small approximation error: more training data or reducing the complexity of the hypothesis class.
5. Large approximation error: change the hypothesis class or the feature representation of the data completely.