Model selection

Reference: Yangfeng Ji, Machine learning, lecture notes

1 Overview

- Model validation: How to evaluate the performance of a given model?
- Model selection: How to select the best model among a few candidates?

2 Model validation: validation set

- **Idea**: evaluate a model by the validation set V
- **Theorem**: a good validation set V should have a similar number of samples as the training set S.
- **Some issues:**

If the validation set is

- small, then it could be biased and could not give a good approximation to the true error
- large, e.g., the same order of the training set, then we waste the information if do not use the examples for training.

- **Solution**: K-Fold Cross Validation

The basic procedure of k-fold cross validation:

- Split the whole data set into k parts
- For each model configuration, run the learning procedure k times
  - Each time, pick one part as validation set and the rest as training set
- Take the average of k validation errors as the model error

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5

- **Dataset splitting**: Train-Validation-Test Split

- Training set: used for learning with a pre-selected hypothesis space, such as
  - logistic regression for classification
  - polynomial regression with $d = 15$ and $\lambda = 0.1$
- Validation set: used for selecting the best hypothesis across multiple hypothesis spaces
  - Similar to learning with a finite hypothesis space $\mathcal{H}'$
- Test set: only used for evaluating the overall best hypothesis

Typical splits on *all* available data

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Test

3 Model selection in Practice

- **4 directions**: sample space, hypothesis space, feature selection, and optimization algorithm

There are many elements that can help fix the learning procedure

- Get a larger sample
- Change the hypothesis class by
  - Enlarging it
  - Reducing it
  - Completely changing it
  - Changing the parameters you consider
- Change the feature representation of the data (usually domain dependent)
- Change the optimization algorithm used to apply your learning rule (lecture on optimization methods)

[Shalev-Shwartz and Ben-David, 2014, Page 151]

- Error decomposition: training error and validation error

With two additional terms

- $L_V(h_S)$: validation error
- $L_S(h_S)$: empirical (*or* training) error

the true error of h_S can be decomposed as

$$L_{\mathcal{D}}(h_S) = \underbrace{(L_{\mathcal{D}}(h_S) - L_V(h_S))}_{(1)} + \underbrace{(L_V(h_S) - L_S(h_S))}_{(2)} + \underbrace{L_S(h_S)}_{(3)}$$

- Item (1) is bounded by the previous theorem
- Item (2) is large: **overfitting**
- Item (3) is large: **underfitting**

- Large training error:

If $L_S(h_S)$ is large, it is possible that

1. the hypothesis space $\mathcal{H}$ is not large enough
2. the hypothesis space is large enough, but your implementation has some bugs

Q: How to distinguish these two?

A: Find an existing simple baseline model

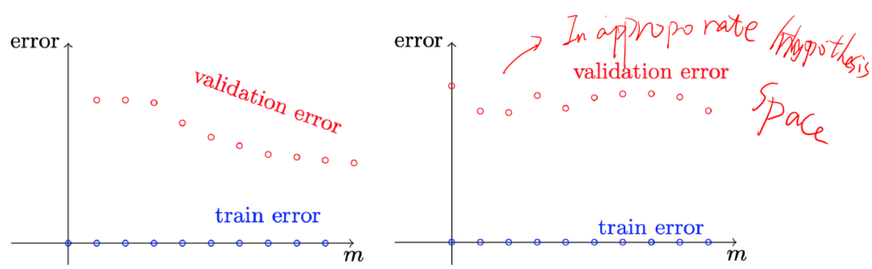
- **Large validation error but small training error**:

... with a small $L_S(h_S)$, it is possible that

1. the hypothesis space is too large
2. you may not have enough training examples
3. the hypothesis space is inappropriate

Comments

- Issue 1 and 2 are easy to fix
  - Get more data if possible, or reduce the hypothesis space
- How to distinguish issue 3 from 1 and 2?



(a) (b)