CS 4774 Machine Learning

Model Selection and Validation

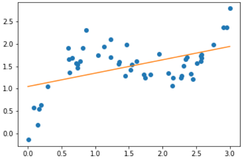
Yangfeng Ji

Information and Language Processing Lab
Department of Computer Science
University of Virginia

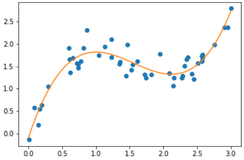
Overview

Overview

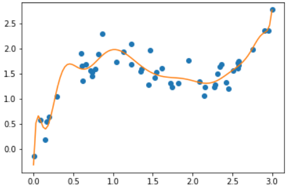
Polynomial regression



(a) $d = 1$



(b) $d = 3$



(c) $d = 15$

Model Evaluation and Selection

Since we cannot compute the true error of any given hypothesis $h \in \mathcal{H}$

- How to evaluate the performance for a given model?
- How to select the best model among a few candidates?

Model Validation

Validation Set

The simplest way to estimate the true error of a predictor h

- Independently sample an additional set of examples V with size m_v

$$V = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_{m_v}, y_{m_v})\} \tag{1}$$

- Evaluate the predictor h on this validation set

$$L_V(h) = \frac{|\{i \in [m_v] : h(\boldsymbol{x}) \neq y_i\}|}{m_v}. \tag{2}$$

Usually, $L_V(h)$ is a good approximation to $L_{\mathcal{D}}(h)$

Model Selection

Model Selection Procedure

Given the training set S and the validation set V

- For each model configuration c, find the best hypothesis $h_c(\boldsymbol{x}, S)$

$$h_c(\boldsymbol{x}, S) = \operatorname*{argmin}_{h' \in \mathcal{H}_c} L_S(h'(\boldsymbol{x}, S)) \tag{3}$$

Model Selection Procedure

Given the training set S and the validation set V

- For each model configuration c, find the best hypothesis $h_c(\boldsymbol{x}, S)$

$$h_c(\boldsymbol{x}, S) = \operatorname*{argmin}_{h' \in \mathcal{H}_c} L_S(h'(\boldsymbol{x}, S)) \tag{3}$$

- With a collection of best models with different configurations $\mathcal{H}' = \{h_{c_1}(\boldsymbol{x}, S), \ldots, h_{c_k}(\boldsymbol{x}, S)\}$, find the overall best hypothesis

$$h(\boldsymbol{x}, S) = \operatorname*{argmin}_{h' \in \mathcal{H}'} L_V(h'(\boldsymbol{x}, S)) \tag{4}$$

Model Selection Procedure

Given the training set S and the validation set V

- For each model configuration c, find the best hypothesis $h_c(\boldsymbol{x}, S)$

$$h_c(\boldsymbol{x}, S) = \underset{h' \in \mathcal{H}_c}{\operatorname{argmin}} L_S(h'(\boldsymbol{x}, S)) \tag{3}$$

- With a collection of best models with different configurations $\mathcal{H}' = \{h_{c_1}(\boldsymbol{x}, S), \ldots, h_{c_k}(\boldsymbol{x}, S)\}$, find the overall best hypothesis

$$h(\boldsymbol{x}, S) = \underset{h' \in \mathcal{H}'}{\operatorname{argmin}} L_V(h'(\boldsymbol{x}, S)) \tag{4}$$

- It is similar to learn with the finite hypothesis space $\mathcal{H}'$

Model Configuration/Hyperparameters

Consider polynomial regression

$$\mathcal{H}_d = \{w_0 + w_1 x + \cdots + w_d x^d : w_0, w_1, \ldots, w_d \in \mathbb{R}\} \tag{5}$$

- the degree of polynomials d
- regularization coefficient λ as in $\lambda \cdot \|\boldsymbol{w}\|_2^2$
- the bias term w_0

Model Configuration/Hyperparameters

Consider polynomial regression

$$\mathscr{H}_d = \{w_0 + w_1 x + \cdots + w_d x^d : w_0, w_1, \ldots, w_d \in \mathbb{R}\} \tag{5}$$

- the degree of polynomials d
- regularization coefficient λ as in $\lambda \cdot \|\boldsymbol{w}\|_2^2$
- the bias term w_0

Additional factors during learning

- Optimization methods
- Dimensionality of inputs, etc.

Limitation of Keeping a Validation Set

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.

k-Fold Cross Validation

The basic procedure of k-fold cross validation:

- Split the whole data set into k parts

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

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5

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

Cross-Validation Algorithm

Input: (1) training set S; (2) set of parameter values Θ; (3) learning algorithm A, and (4) integer k

Partition S into $S_1, S_2, \ldots, S_k$

for $\theta_t \in \Theta$ **do**

  for $i = 1, \ldots, k$ **do**

    $h_{i,\theta_t} = A(S \backslash S_i; \theta_t)$

  end for

  $\mathrm{Err}(\theta_t) = \frac{1}{k} \sum_{i=1}^{k} L_{S_i}(h_{i,\theta_t})$

end for

Output: $\hat{\theta} \leftarrow \operatorname{argmin}_{\theta_t \in \Theta} \mathrm{Err}(\theta_t)$

In practice, k is usually 5 or 10.

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 $\mathscr{H}'$
- Test set: only used for evaluating the overall best hypothesis

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

Train	Val	Test

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 $\mathscr{H}'$
- 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

Model Selection in Practice

What To Do If A Learning Fails

There are many elements that can help fix the learning procedure

- Get a larger sample

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

What To Do If A Learning Fails

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

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

What To Do If A Learning Fails

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)

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

What To Do If A Learning Fails

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 Using Validation

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

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

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

About Large $L_S(h_S)$

Recall that h_S is an ERM hypothesis, aka

$$h_S \in \operatorname*{argmin}_{h' \in \mathcal{H}} L_S(h') \tag{6}$$

About Large $L_S(h_S)$

Recall that h_S is an ERM hypothesis, aka

$$h_S \in \operatorname*{argmin}_{h' \in \mathcal{H}} L_S(h') \tag{6}$$

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

About Large $L_S(h_S)$

Recall that h_S is an ERM hypothesis, aka

$$h_S \in \operatorname*{argmin}_{h' \in \mathcal{H}} L_S(h') \tag{6}$$

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?

About Large $L_S(h_S)$

Recall that h_S is an ERM hypothesis, aka

$$h_S \in \operatorname*{argmin}_{h' \in \mathcal{H}} L_S(h') \tag{6}$$

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

About Large $L_V(h_S)$

... 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

About Large $L_V(h_S)$

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

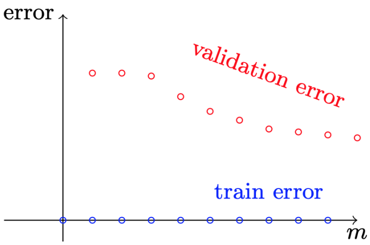
1. the hypothesis space is too large
2. you may not have enough training examples
3. the hypothesis space is inappropriate

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?

Learning Curves

With different proportions of training examples, we can plot the training and validation errors

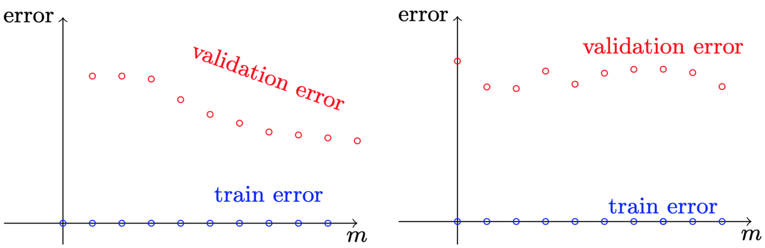


(a)

Figure: Examples of learning curves [Shalev-Shwartz and Ben-David, 2014, Page 153].

Learning Curves

With different proportions of training examples, we can plot the training and validation errors



(a) (b)

Figure: Examples of learning curves [Shalev-Shwartz and Ben-David, 2014, Page 153].

Reference

Shalev-Shwartz, S. and Ben-David, S. (2014).
Understanding machine learning: From theory to algorithms.
Cambridge university press.