CS 6316 Machine Learning

Model Selection and Validation

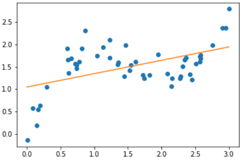
Yangfeng Ji

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

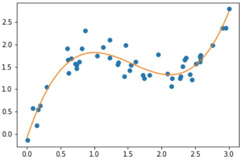
Overview

Overview

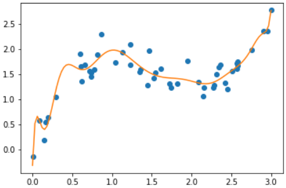
Polynomial regression



(a) $d = 1$



(b) $d = 3$



(c) $d = 15$

Structural Risk Minimization

Take linear regression with ℓ_2 as an example. Let $\mathcal{H}_\lambda$ represents the hypothesis space defined with the following objective function

$$L_{S,\ell_2}(h_{\boldsymbol{w}}) = \frac{1}{m}\sum_{i=1}^{m}(h_{\boldsymbol{w}}(\boldsymbol{x}_i) - y_i)^2 + \lambda\|\boldsymbol{w}\|^2 \tag{1}$$

where λ is the regularization parameter

Structural Risk Minimization

Take linear regression with ℓ_2 as an example. Let $\mathcal{H}_\lambda$ represents the hypothesis space defined with the following objective function

$$L_{S,\ell_2}(h_{\boldsymbol{w}}) = \frac{1}{m}\sum_{i=1}^{m}(h_{\boldsymbol{w}}(\boldsymbol{x}_i) - y_i)^2 + \lambda\|\boldsymbol{w}\|^2 \quad (1)$$

where λ is the regularization parameter

- The basic idea of SRM is to start from a small hypothesis space (e.g., $\mathcal{H}_\lambda$ with a small λ, then gradually increase λ to have a larger $\mathcal{H}_\lambda$

Structural Risk Minimization

Take linear regression with ℓ_2 as an example. Let $\mathcal{H}_\lambda$ represents the hypothesis space defined with the following objective function

$$L_{S,\ell_2}(h_{\boldsymbol{w}}) = \frac{1}{m}\sum_{i=1}^{m}(h_{\boldsymbol{w}}(\boldsymbol{x}_i) - y_i)^2 + \lambda\|\boldsymbol{w}\|^2 \quad (1)$$

where λ is the regularization parameter

- The basic idea of SRM is to start from a small hypothesis space (e.g., $\mathcal{H}_\lambda$ with a small λ, then gradually increase λ to have a larger $\mathcal{H}_\lambda$
- Another example: Support Vector Machines

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
$$V = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_{m_v}, y_{m_v})\} \tag{2}$$
- Evaluate the predictor h on this validation set
$$L_V(h) = \frac{|\{i \in [m_v] : h(\boldsymbol{x}) \neq y_i\}|}{m_v}. \tag{3}$$

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

Theorem

Let h be some predictor and assume that the loss function is in $[0, 1]$. Then, for every $\delta \in (0, 1)$, with probability of at least $1 - \delta$ over the choice of a validation set V of size m_v, we have

$$|L_V(h) - L_{\mathscr{D}}(h)| \leq \sqrt{\frac{\log(2/\delta)}{2m_v}} \tag{4}$$

where

- $L_V(h)$: the validation error
- $L_{\mathscr{D}}(h)$: the true error

[Shalev-Shwartz and Ben-David, 2014, Theorem 11.1]

Sample Complexity

- The fundamental theorem of learning

$$L_{\mathscr{D}}(h) \leq L_S(h) + \sqrt{C\frac{d + \log(1/\delta)}{m}} \tag{5}$$

where d is the VC dimension of the corresponding hypothesis space

Sample Complexity

- The fundamental theorem of learning

$$L_{\mathscr{D}}(h) \leq L_S(h) + \sqrt{C\frac{d + \log(1/\delta)}{m}} \tag{5}$$

  where d is the VC dimension of the corresponding hypothesis space

- On the other hand, from the previous theorem

$$L_{\mathscr{D}}(h) \leq L_V(h) + \sqrt{\frac{\log(2/\delta)}{2m_v}} \tag{6}$$

- A good validation set should have similar number of examples as in the training set

Model Selection

Model Selection Procedure

Given the training set S and the validation set V

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

$$h_c(\boldsymbol{x}, S) = \operatorname*{argmin}_{h' \in \mathcal{H}_c} L_S(h'(\boldsymbol{x}, S)) \tag{7}$$

Model Selection Procedure

Given the training set S and the validation set V

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

$$h_c(\boldsymbol{x}, S) = \operatorname*{argmin}_{h' \in \mathcal{H}_c} L_S(h'(\boldsymbol{x}, S)) \tag{7}$$

- 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

$$h(\boldsymbol{x}, S) = \operatorname*{argmin}_{h' \in \mathcal{H}'} L_V(h'(\boldsymbol{x}, S)) \tag{8}$$

Model Selection Procedure

Given the training set S and the validation set V

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

$$h_c(\boldsymbol{x}, S) = \operatorname*{argmin}_{h' \in \mathcal{H}_c} L_S(h'(\boldsymbol{x}, S)) \tag{7}$$

- 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

$$h(\boldsymbol{x}, S) = \operatorname*{argmin}_{h' \in \mathcal{H}'} L_V(h'(\boldsymbol{x}, S)) \tag{8}$$

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

Model Configuration/Hyperparameters

Consider polynomial regression

$$\mathscr{H}_d = \{w_0 + w_1 x + \cdots + w_d x^d : w_0, w_1, \ldots, w_d \in \mathbb{R}\} \tag{9}$$

- 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

$$\mathscr{H}_d = \{w_0 + w_1 x + \cdots + w_d x^d : w_0, w_1, \ldots, w_d \in \mathbb{R}\} \tag{9}$$

- 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, ..., S_k
for θ_t ∈ Θ do
    for i = 1, ..., k do
        h_{i,θ_t} = A(S\S_i; θ_t)
    end for
    Err(θ_t) = (1/k) Σ_{i=1}^{k} L_{S_i}(h_{i,θ_t})
end for
Output: θ̂ ← argmin_{θ_t ∈ Θ} Err(θ_t)
```

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

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 $\mathscr{H}'$
- 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 $\mathcal{H}'$
- 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

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

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

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

About Large $L_S(h_S)$

Recall that h_S is an ERM hypothesis, aka

$$h_S \in \underset{h' \in \mathcal{H}}{\operatorname{argmin}} L_S(h') \tag{10}$$

About Large $L_S(h_S)$

Recall that h_S is an ERM hypothesis, aka

$$h_S \in \operatorname*{argmin}_{h' \in \mathcal{H}} L_S(h') \tag{10}$$

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

$$h_S \in \operatorname*{argmin}_{h' \in \mathcal{H}} L_S(h') \tag{10}$$

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

$$h_S \in \operatorname*{argmin}_{h' \in \mathcal{H}} L_S(h') \tag{10}$$

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

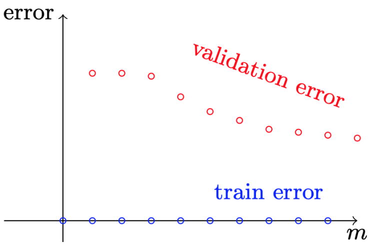
1. the hypothesis space is too large
2. you may not have enough training examples
3. the hypothesis space is inappropriate

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

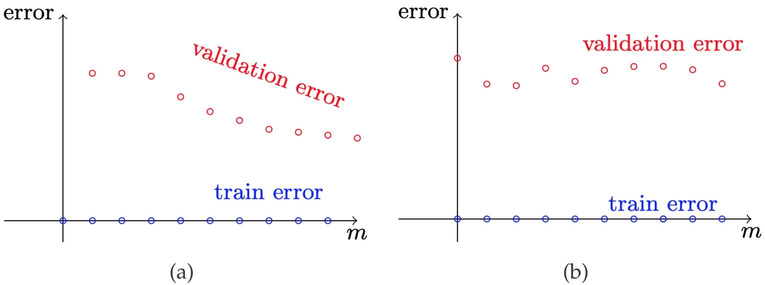


Figure: Examples of learning curves [Shalev-Shwartz and Ben-David, 2014, Page 153].

Final Project

Proposal Deadline

March 6, 11:59 PM (about two weeks from now)

Goals

The goals of the final project

- Provide an opportunity to practice what we learn in class about learning theory and algorithms, such as
  - Bias-variance tradeoff
  - Overfitting vs. underfitting
  - Logistic regression, regularization, boosting, SVMs, etc.

Goals

The goals of the final project

- Provide an opportunity to practice what we learn in class about learning theory and algorithms, such as
  - Bias-variance tradeoff
  - Overfitting vs. underfitting
  - Logistic regression, regularization, boosting, SVMs, etc.
- Encourages students to think about something beyond the course materials, e.g.,
  - Real-world applications
  - Feature representation and selection
  - High-dimensional data

Two Types of Projects

We accept two types of projects

- **Application project**: pick an application problem that interests you and explore how to find the best learning algorithms to solve it.

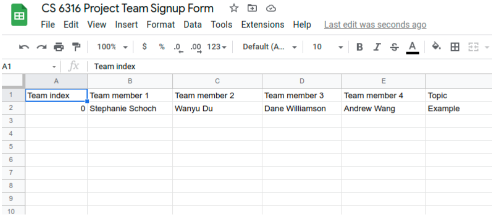
Two Types of Projects

We accept two types of projects

- **Application project**: pick an application problem that interests you and explore how to find the best learning algorithms to solve it.
- **Algorithmic project**: Pick a machine learning problem, then
  (1) develop a new algorithm to solve this problem, or
  (2) design a variant of an existing algorithm that can provide a better solution

Team and Proposal

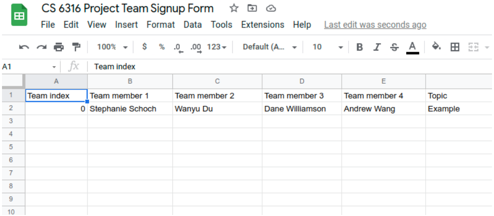
- Each team will have up to four students
  - Final project signup form on the course webpage



	A	B	C	D	E	
1	Team index	Team member 1	Team member 2	Team member 3	Team member 4	Topic
2	0	Stephanie Schoch	Wanyu Du	Dane Williamson	Andrew Wang	Example
3						
4						
5						
6						
7						
8						
9						
10						

Team and Proposal

- Each team will have up to four students
  - Final project signup form on the course webpage


CS 6316 Project Team Signup Form

File Edit View Insert Format Data Tools Extensions Help Last edit was seconds ago

A1 Team index

	A	B	C	D	E	
1	Team index	Team member 1	Team member 2	Team member 3	Team member 4	Topic
2	0	Stephanie Schoch	Wanyu Du	Dane Williamson	Andrew Wang	Example
3						
4						
5						
6						
7						
8						
9						
10						

- Project proposal
  - Follow the ICLR 2022 template
  - Include the following items in the author list
    - Team No.
    - All team members
  - Page limits: 2 – 3 pages, including references

Project Proposal Format: Main Content

The proposal should include the following five sections, eight points in total

1. Problem definition (2 point)
2. Proposed idea (2.5 points)
3. Related work (2 point)
4. Datasets (1 point)
5. Timeline (0.5 point)

Project Proposal Format: Main Content

The proposal should include the following five sections, eight points in total

1. Problem definition (2 point)
2. Proposed idea (2.5 points)
3. Related work (2 point)
4. Datasets (1 point)
5. Timeline (0.5 point)

Each team only submit one proposal, the same points of the proposal are shared by all team members.

Problem Definition

... should include at least the following information

- Input domain of the problem
- Output domain of the problem
- Some specific examples to explain the input/output domains
- The *motivation* of choosing this problem

Proposed Idea

Depending the type of your project

- Application projects
  - provide a concrete plan of exploring some learning algorithms to solve the problem
  - show some justifications
  - explain the expected outcome

Proposed Idea

Depending the type of your project

- Application projects
  - provide a concrete plan of exploring some learning algorithms to solve the problem
  - show some justifications
  - explain the expected outcome
- Algorithmic projects
  - identify the challenge of solving the machine learning problem and discuss the common limitation of existing algorithms
  - propose an idea to address the limitation
  - explain the expected outcome

Related Work

The related work section should include

- At least *three* papers on the related work
- For each paper
  - Application projects: describe why the work in this paper can be used to solve the proposed problem
  - Algorithmic projects: describe what is the specific issues of the work proposed in this paper and how your proposed idea can address these issues

Datasets

Please include a brief description about the dataset(s) that you will use in this section, for example,

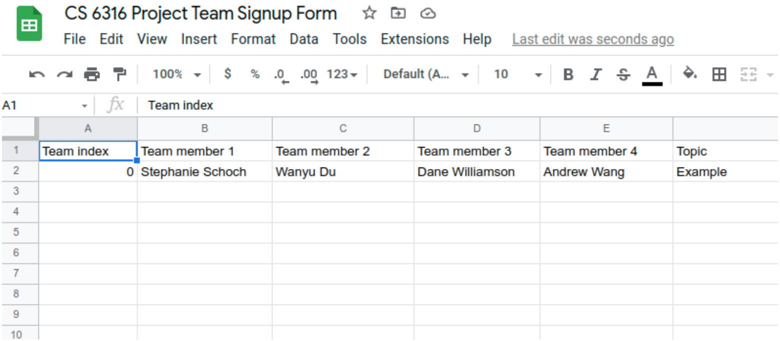
- Size of the dataset
- Dimensionality
- Data splits
  - Train-validation-test split
  - Train-test split for cross-validation

Timeline

What you plan to do in each two weeks till the end of the semester.

Signup Online

You can find the link on the course webpage



	A	B	C	D	E	
1	Team index	Team member 1	Team member 2	Team member 3	Team member 4	Topic
2	0	Stephanie Schoch	Wanyu Du	Dane Williamson	Andrew Wang	Example
3						
4						
5						
6						
7						
8						
9						
10						

- Proposal deadline: March 6, 11:59 PM

Reference

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