

# Regression: Overfitting and Curse of Dimensionality

---

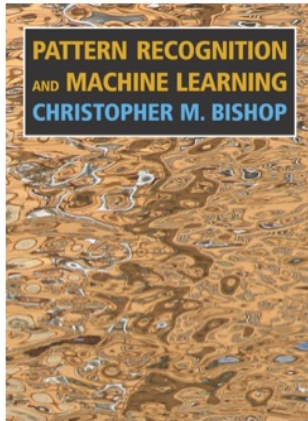
Song Liu ([song.liu@bristol.ac.uk](mailto:song.liu@bristol.ac.uk))

Office Hour: 3-4pm Tuesday

Office: Fry Building GA 18

## Reference

---



Today's class *roughly* follows Chapter 1.

Pattern Recognition and  
Machine Learning

Christopher Bishop, 2006

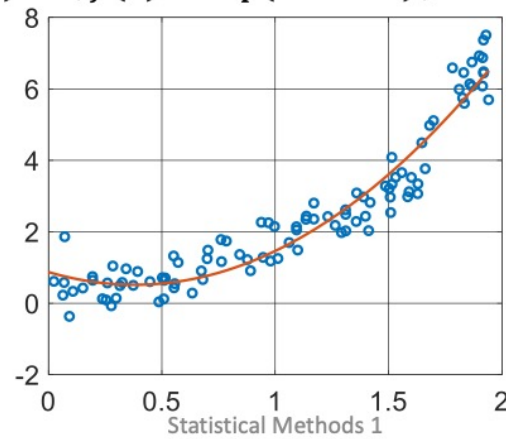
## LS with Feature Transform

$$\mathbf{w}_{\text{LS}} := \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{i \in D_0} [y_i - f(\mathbf{x}_i; \mathbf{w})]^2$$
$$f(\mathbf{x}; \mathbf{w}) := \langle \mathbf{w}_1, \boldsymbol{\phi}(\mathbf{x}) \rangle + w_0, \mathbf{w} := [\mathbf{w}_1, w_0]^\top$$

- $\boldsymbol{\phi}(\mathbf{x})$  can be a collection of polynomial functions:
- $\boldsymbol{\phi}(\mathbf{x}) := [x^1, x^2, x^3 \dots x^b]^\top$ .
- $b$  is called the degree of  $\boldsymbol{\phi}(\mathbf{x})$ .

## LS with Polynomial Transform ( $b = 2$ )

- $x \sim \text{uniform}(0,2)$
- $y = f(x) + \epsilon, f(x) = \exp(1.5x - 1), \epsilon \sim N(0, .64)$



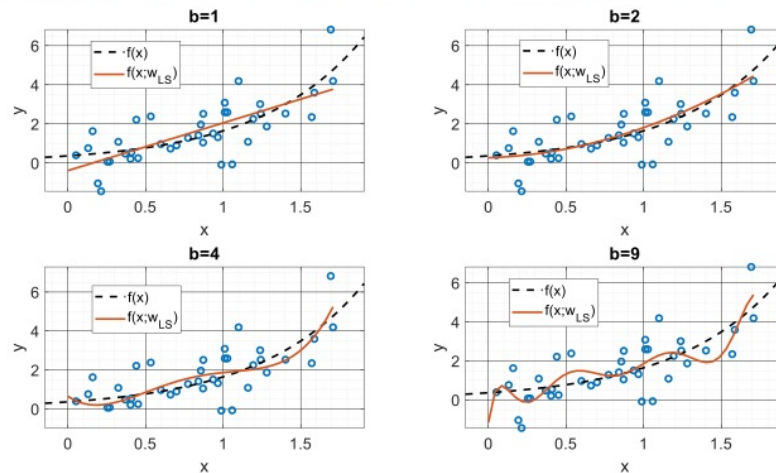
9/28/21

Statistical Methods 1

4

The dataset has a conditional mean of  $f(x)$ , why?

## Poly. Transform with various $b$



9/28/21

Statistical Methods 1

5

Recall, we have learned in the last lecture, LS is supposed to fit the conditional mean using its prediction function, under Normal distributional assumption.

Notice how the fitted curves behave in the case  $b=4$ ,  $b=9$ .

The curve is almost trying **too hard** to pass through all points.

In fact, if you increase  $b$  to be  $n$ , your fitted line will pass through all your data points. Why?

## Poly. Feature with various $b$

- The higher the  $b$ , the more flexible our  $f(x; \mathbf{w})$  is.
- However, when increasing  $b$ ,
  - The fit of  $f(x; \mathbf{w}_{LS})$  first got better ( $b = 2$ ).
  - then got worse ( $b = 4, b = 9$ ).
  - $f(x; \mathbf{w}_{LS})$  become too “squiggly”, when  $b$  is large.
  - $f(x; \mathbf{w}_{LS})$  almost tried “too hard” to fit our data.
- Is this a general pattern?
  - We design an experiment to find out.

9/28/21

Statistical Methods 1

6

It is somewhat counter intuitive. If an  $f$  is more flexible, we should expect it to be better right? NO. Flexibility can be a bad thing, as the flexibility indicates the complexity of our prediction function.

## Training Error and Testing Error

- We randomly split our dataset  $D$  into  $D_0$  and  $D_1$ .
  - assuming  $D$  contains IID pairs.
- $\mathbf{w}_{LS}$  is fitted using  $D_0$  only.
- Define an error  $E(D', \mathbf{w}) = \sum_{i \in D'} [y_i - f(\mathbf{x}_i; \mathbf{w})]^2$ .
- It tells how well  $f(\mathbf{x}; \mathbf{w})$  fits a specific dataset  $D'$ .
- We can have **two performance metrics**:
- $E(D_0, \mathbf{w}_{LS})$  is usually referred to as **training error**.
- $E(D_1, \mathbf{w}_{LS})$  is usually referred to as **testing error**.

9/28/21

Statistical Methods 1

7

The error represents the goodness of fitting.

Which error would better indicate the “general performance” of our estimated  $f(\mathbf{x}; \mathbf{w}_{LS})$ ?

## Training Error and Testing Error

- We do not care  $E(D_0, w_{LS})$ !
- We have already seen the output in  $D_0$  during the training.
- We care performance of  $f(\mathbf{x}; \mathbf{w}_{LS})$  on unseen dataset  $D_1$ !
- The ability of getting low  $E(D_1, w_{LS})$  is called **generalization**.
- Generalization is a **key goal** in statistical decision making.
- Go back to the example,
- As  $b$  increases, how  $E(D_0, w_{LS})$  and  $E(D_1, w_{LS})$  change?

9/28/21

Statistical Methods 1

8

You can think of this as a student preparing exam:

Student is given past exam papers.

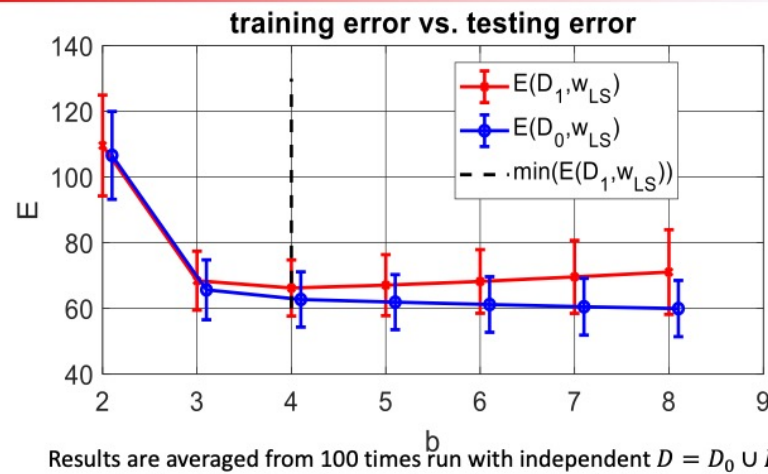
Student does not care the score he got in a past exam paper, as he has already seen all the answers of past papers.

He cares how much he will get in a future, unseen exam, and how well he can generalize his knowledge to this future exam.

Not just in regression problems, in all tasks machine learning tasks, we care about the generalization ability.



# Training Error and Testing Error



9/28/21

Statistical Methods 1

9

## Training Error and Testing Error

- Training error keeps reducing.
  - $f(\mathbf{x}; \mathbf{w}_{LS})$  fit  $D_0$  better and better as  $b$  increases.
  - Testing error drops then goes up again.
  - $f(\mathbf{x}; \mathbf{w}_{LS})$  does not fit unseen  $D_1$  well, when  $b$  is too large.
  - **The problem:**
    - Generalization of  $f(\mathbf{x}; \mathbf{w}_{LS})$  deteriorates when  $b$  is too large.
    - The phenomenon  $f(\mathbf{x}; \mathbf{w}_{LS})$  fits too well on training set while underperforming on unseen datasets, is called
- Overfitting.**

9/28/21

Statistical Methods 1

10

Overfitting is one of the central issue in machine learning. Many methodologies (such as regularization) are developed, specifically for the purpose of reducing overfitting.

## Selecting $b$

- $b$  should not be too small, so  $f$  is **flexible enough**!
- $b$  should not be too large, so  $f$  is **not too flexible**!
- How do we select?
- We can split full dataset  $D$  into  $D_0$  and  $D_1$ .
- Use  $D_0$  to fit  $f_{LS}(b)$  and use  $D_1$  to compute  $E(D_1, f_{LS}(b))$ .
- Select a  $b$  such that  $E(D_1, f_{LS}(b))$  is the lowest.
- Fit  $f_{LS}$  again using the selected  $b$  on the full dataset.

9/28/21

Statistical Methods 1

11

It is like doing the experiment on page 9 again, but without the error bar, because we are not allowed to regenerate the dataset (we only have one dataset!).

Selecting a model to get good generalization ability is also an important statistical learning topic.

## Selecting $b$ (Efficiently!)

Problem of splitting  $D$  into  $D_0$  and  $D_1$ :

1. However, we have wasted  $D_1$  for validation.
    - What if  $D_1$  contains info that is beneficial for fitting a good  $f_{LS}$ ?
  2. Only computed  $E(D_1, f_{LS}(b))$  once, result may be random.
- 
- Split  $D$  into  $D_0$  and  $D_1$ , compute  $E(D_1, f_{LS}(b))$
  - Swap the role of  $D_0$  and  $D_1$ , compute  $E(D_0, f'_{LS}(b))$ 
    - $f'_{LS}(b)$  is fitted using  $D_1$
  - Select  $b$  that minimizes  $\frac{E(D_1, f_{LS}(b))}{2} + \frac{E(D_0, f'_{LS}(b))}{2}$

9/28/21

Statistical Methods 1

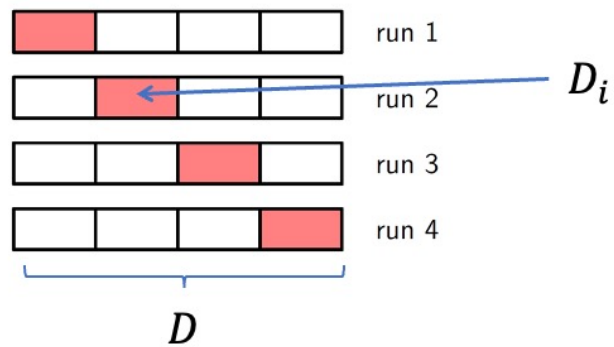
12

We are getting an aggregated error.

## Cross-validation

- The extension of above idea gives rise to a commonly used model selection method: **Cross-validation**.
- Split  $D$  into **disjoint**  $D_0 \dots D_k$ ,
- For  $i = 0$  to  $k$ 
  - Fit  $f_{\text{LS}}^{(i)}(b)$  on all subsets **but**  $D_i$ ,  $\forall b$
  - Compute  $E(D_i, f_{\text{LS}}^{(i)}(b))$ ,  $\forall b$
- Select  $b$  that minimizes  $\frac{\sum_i E^{(i)}}{k+1}$
- $k$  can go as high as  $n - 1$ : **leave-one-out-validation**

## Cross-validation



- PRML, Figure 1.18
- Read Chapter PRML 1.3

## Problem of Cross-validation

- The implementation of cross-validation is easy,
- But the computational cost is high.
  - $f_{LS}^{(i)}(x; \mathbf{w})$  must be fitted and validated for all splits.
- The effectiveness of cross-validation depends on the IID assumption of our dataset  $D$ .
  - Validation set and the training set must be IID!
  - Which may not hold in reality: e.g. stock price dataset.
- Can we avoid overfitting without splitting our dataset for validation? We will discuss this in the future.

9/28/21

Statistical Methods 1

15

The identical distributional assumption is needed as if data comes from different distributions, the random split on the dataset does not make any sense.

The independent assumption is needed as the average of cross validation would not converge (law of large numbers)

## Polynomial Transform on Higher Dimensional Dataset

- So far, we only considered polynomial transform on one dimensional dataset, i.e.,  $x \in R$
- What about  $x \in R^d$ , when the output  $y$  depends on multiple inputs?
- When  $x \in R^d$ ,
  - $\phi(x) := [h(x^{(1)}), h(x^{(2)}), \dots, h(x^{(d)})]^\top$ .
  - $h(t) := [t^1, t^2, \dots, t^b] \in R^b$ .
  - $\phi(x) \in R^{db}$ , which means  $w_1 \in R^{db}$ .
- This does **not** include cross-dimension polynomials.
  - e.g.,  $x^{(1)} x^{(2)}, x^{(1)} x^{(2)} x^{(3)}, \dots$
  - These can be useful as the output value may depends jointly on several inputs. e.g. blood pressure <- (weight,height)

9/28/21

Statistical Methods 1

16

BMI index,

Wei/Hei<sup>2</sup>

cannot be represented as weighted sum of weight and height.



## Polynomial Transform on Higher Dimensional Dataset

- To include **pairwise** cross-dimension polynomials, we can slightly redesign  $\phi(x)$ :
  - $\phi(x) := [h(x^{(1)}), \dots, h(x^{(d)}), \forall_{u < v} x^{(u)} x^{(v)}]$
  - $\phi(x) \in R^{db + \binom{d}{2}}$ ,
- Similarly, we can include all the **triplets**:
  - $\phi(x) := [h(x^{(1)}), \dots, h(x^{(d)}), \forall_{u < v} x^{(u)} x^{(v)}, \forall_{u < v < w} x^{(u)} x^{(v)} x^{(w)}]$
  - $\phi(x) \in R^{db + \binom{d}{2} + \binom{d}{3}}$ ,
- and we can go on to include **quadruplets**...

## Curse of Dimensionality

---

- We can include cross terms all the way up to  $d$ -plets.
- Unfortunately, we know
  - $\binom{d}{1} + \binom{d}{2} + \binom{d}{3} + \binom{d}{4} + \dots + \binom{d}{d} = 2^d$
- We have not yet included cross terms like:
  - $[x^{(u)}]^2 x^{(v)} \dots$
- The output dimension of  $\phi(\mathbf{x})$  can grow exponentially with dimensionality  $d$  and this is a bad news...

## Curse of Dimensionality

- We have seen in yesterday's homework, the number of observations  $n$ , needs to at least match the output dimension of  $\phi(x)$ , otherwise, we cannot obtain  $w_{LS}$ !
- It means we need to grow  $n$  exponentially with  $d$ !
- Imagine a problem with  $d = 100$ .
  - A terabyte-data on hard-drive contains  $2^{40}$  bytes.

9/28/21

Statistical Methods 1

19

Gene expression datasets usually contain thousands of dimensions with merely dozens of observations.

[https://en.wikipedia.org/wiki/Gene\\_expression\\_profiling](https://en.wikipedia.org/wiki/Gene_expression_profiling)

## Curse of Dimensionality

- The phenomenon, that the number of observations needed to solve a problem grows exponentially with  $d$  exists in many statistical learning tasks.
- They are collectively called “Curse of Dimensionality”.
- This phenomenon forbids us solving high-dimensional problems.

9/28/21

Statistical Methods 1

20

We will see CoD problem again, when introducing non-parametric methods.

Read PRML 1.4, Bishop has a different example.

To solve high dimensional problems, we have to either

Reduce the dimension of  $\mathbf{x}$

Limit the complexity of our model  $\phi(\mathbf{x})$ , e.g., pairwise term only.

## Conclusion

---

- We introduce poly. transform to our prediction func.  $f$ .
- This increases the flexibility of  $f$ , but we also see this additional complexity caused two major problems:
- **Overfitting**
  - The generalization of  $f$  is poor.
- **Curse of Dimensionality**
  - $n$  needs to grows exponentially with the dimensionality of  $x$ .
- Next week, we will introduce a way to reduce the flexibility of  $f$  to combat overfitting and the probabilistic idea behind it.

## Computing Lab

---

- Download “Prostate Cancer dataset”, [description](#), [dataset](#).
- Implement a Least-square solver using R. Do not use built-in functions.
- Fit  $f(\mathbf{x}; \mathbf{w})$  using classic linear least squares.
- Calculate the cross-validation error.
- How does the cross-validation testing error change if you **remove one of the features**?
  - How do you explain this using what we have learned today?