Logistic Regression Part 3

Dr Mehrdad Ghaziasgar

Senior Lecturer in Computer Science
Department of Computer Science
University of the Western Cape
mghaziasgar@uwc.ac.za

Content

- Part 1:
    - Classification With Logistic Regression
    - Model Evaluation For Classification

- Part 2:
    - Overfitting and Underfitting

- Part 3:
    - Practical Issues With Classification

Content – Part 3

- Practical Issues
    - Introduction
    - Polynomial Degree?
    - How Much to Regularize?
    - Training Set Size?

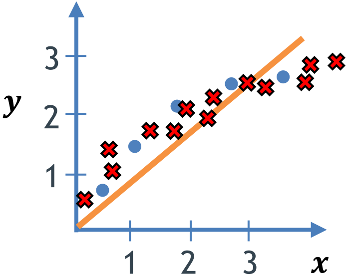
Practical Issues

Introduction

- Overfitting and Underfitting
    - Introduction
    - Polynomial Degree?
    - How Much to Regularize?
    - Training Set Size?

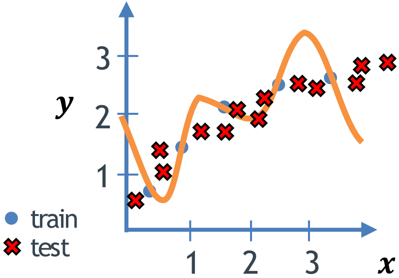
Introduction

- In the previous part we discovered that the degree of polynomial that we use can affect accuracy:
  - Too low: the model will have high bias: too simple
  - Too high: the model will have high variance: too complex and "hugging" the data



$h_\theta(x) = \theta_0 + \theta_1 x$

degree d = 1



$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

degree d = 4

Introduction

- In the previous part we discovered that the degree of polynomial that we use can affect accuracy:
    - Too low: the model will have high bias: too simple
    - Too high: the model will have high variance: too complex and "hugging" the data

- We looked at bias and variance:
    - Bias: the model is too simple
        - The model doesn't fit either the train or test data well
    - Variance: the model is too complex
        - The model fits the train data VERY well but doesn't fit the test data well

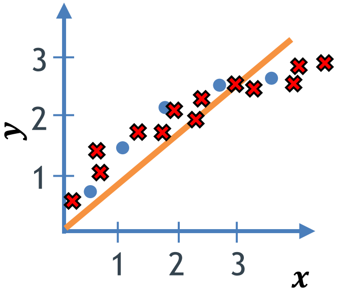
    - In both cases: the model fails to predict the test data well (which is what we're interested in in the first place)

Introduction

- We spoke about regularization using the parameter λ
    - Setting λ too high: high regularization – all weights are reduced to almost zero – high bias
    - Setting λ too low (zero): no regularization – model remains complex – high variance
- Question is: how can we systematically / practically determine:
    - The best polynomial degree?

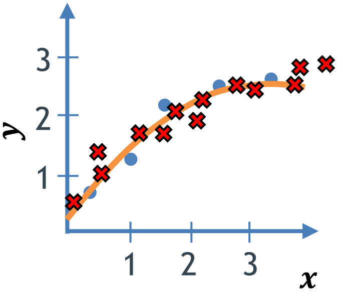
Introduction

- We spoke about regularization using the parameter λ
  - Setting λ too high: high regularization - all weights are reduced to almost zero - high bias
  - Setting λ too low (zero): no regularization - model remains complex - high variance
- Question is: how can we systematically / practically determine:
  - The best polynomial degree?



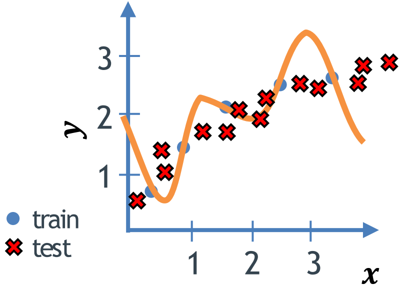
$h_\theta(x) = \theta_0 + \theta_1 x$

degree d = 1



$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$

degree d = 2 ("just right")



$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

degree d = 4

Introduction

- We spoke about regularization using the parameter λ
  - Setting λ too high: high regularization – all weights are reduced to almost zero – high bias
  - Setting λ too low (zero): no regularization – model remains complex – high variance
- Question is: how can we systematically / practically determine:
  - The best polynomial degree?
  - How much to regularize?
- ALL of these issues have to do with:
  - Diagnosing Bias vs Variance
  - Error on the train set vs Error on the CV/test set
- Also: how much data should we train on?
  - Preferable to use the least amount of data that **helps**: how much helps??
  - Is more data **always** better?

Practical Issues

Polynomial Degree?

- Overfitting and Underfitting
    - Introduction
    - Polynomial Degree?
    - How Much to Regularize?
    - Training Set Size?

Determining Polynomial Degree

- We split the data into Train-CV-Test sets

- For each polynomial of degree $\boldsymbol{d}$ we're comparing:
  - We train a hypothesis $\boldsymbol{h_\theta}(\boldsymbol{x})$ of degree $\boldsymbol{d}$ on the Train set e.g.
    - for $\boldsymbol{d} = 1$, $\boldsymbol{h_\theta}(\boldsymbol{x}) = \boldsymbol{\theta_0} + \boldsymbol{\theta_1 x}$,
    - for $\boldsymbol{d} = 2$, $\boldsymbol{h_\theta}(\boldsymbol{x}) = \boldsymbol{\theta_0} + \boldsymbol{\theta_1 x} + \boldsymbol{\theta_2 x^2}$
    - etc.
  - Training gives us the best $\boldsymbol{\theta}$ for $\boldsymbol{h_\theta}(\boldsymbol{x})$ on the Train set (that minimizes the cost)
  - We pass in the (same) Train set to the cost function $\boldsymbol{J}(\boldsymbol{\theta})$ to get the training error $\boldsymbol{J}_{\text{train}}$
  - We pass in the CV set to the cost function $\boldsymbol{J}(\boldsymbol{\theta})$ to get the CV error $\boldsymbol{J}_{\text{cv}}$
  - We plot these values on a graph (we'll see it later)

- Pick the model with $\boldsymbol{d}$ that produces the best (lowest) $\boldsymbol{J}_{\text{cv}}$
- Compute the error of this model on the Test set $\boldsymbol{J}_{\text{test}}$ which is taken as the generalization performance of the selected model

Determining Polynomial Degree

- E.g.

1. $\boldsymbol{d} = 1$ $\rightarrow$ $\min_{\boldsymbol{\theta}} \boldsymbol{J}(\boldsymbol{\theta})$ to get $\boldsymbol{\theta}^{(1)}$ $\rightarrow$ plot $\boldsymbol{J}_{\text{train}}(\boldsymbol{\theta}^{(1)})$, $\boldsymbol{J}_{\text{cv}}(\boldsymbol{\theta}^{(1)})$
2. $\boldsymbol{d} = 2$ $\rightarrow$ $\min_{\boldsymbol{\theta}} \boldsymbol{J}(\boldsymbol{\theta})$ to get $\boldsymbol{\theta}^{(2)}$ $\rightarrow$ plot $\boldsymbol{J}_{\text{train}}(\boldsymbol{\theta}^{(2)}), \boldsymbol{J}_{\text{cv}}(\boldsymbol{\theta}^{(2)})$

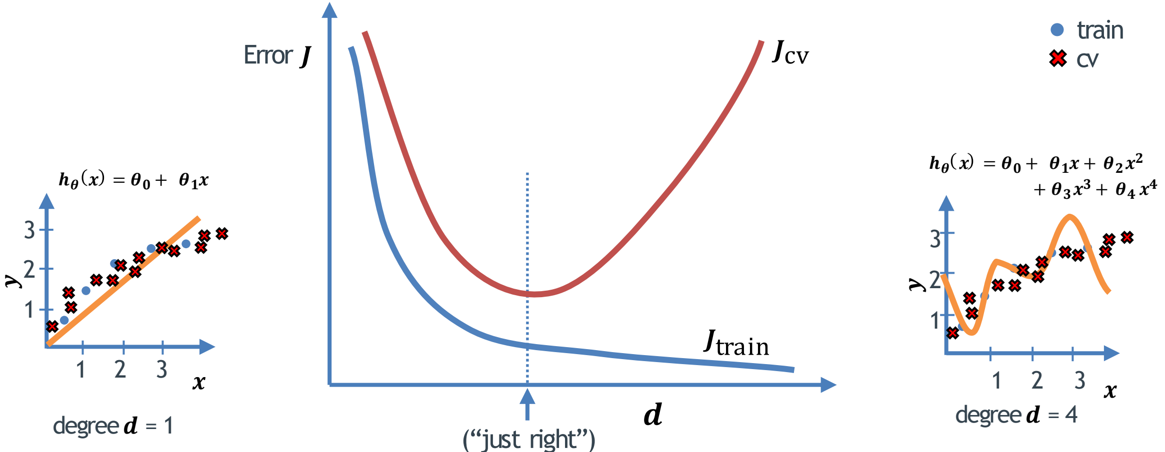
...

10. $\boldsymbol{d} = 10$ $\rightarrow$ $\min_{\boldsymbol{\theta}} \boldsymbol{J}(\boldsymbol{\theta})$ to get $\boldsymbol{\theta}^{(10)}$ $\rightarrow$ plot $\boldsymbol{J}_{\text{train}}(\boldsymbol{\theta}^{(10)}), \boldsymbol{J}_{\text{cv}}(\boldsymbol{\theta}^{(10)})$

- Assume that $\boldsymbol{d} = 4$ gets the lowest $\boldsymbol{J}_{\text{cv}}$
- Compute the error of this model on the Test set $\boldsymbol{J}_{\text{test}}(\boldsymbol{\theta}^{(4)})$ which is taken as the generalization performance of the selected model

Determining Polynomial Degree

- The plot of the two errors J_{train} and J_{cv} versus the parameter we're optimizing i.e. $\boldsymbol{d}$:

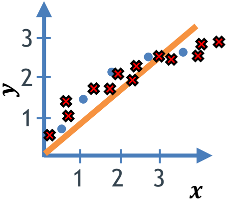


Determining Polynomial Degree

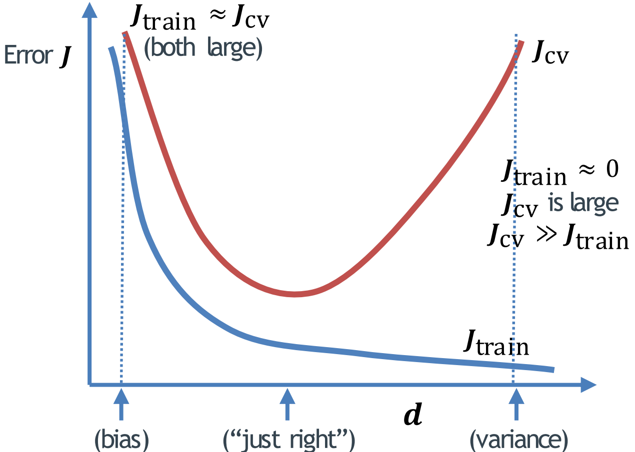
- Bias:
- $J_{\text{train}} \approx J_{\text{cv}}$ and both errors are large

- Variance:
- $J_{\text{train}} \approx 0$ and J_{cv} is large so
- $J_{\text{cv}} \gg J_{\text{train}}$

$h_\theta(x) = \theta_0 + \theta_1 x$

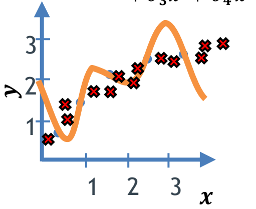


degree d = 1



- train
- cv

$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$



degree d = 4

Practical Issues
How Much to Regularize?

- Overfitting and Underfitting
  - Introduction
  - Polynomial Degree?
  - How Much to Regularize?
  - Training Set Size?

Determining How Much to Regularize

- We're using a high-order polynomial e.g. $\boldsymbol{h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4}$
- We split the data into Train-CV-Test sets

- For a range of λ values that we're comparing e.g. 0, 0.02, 0.04, 0.08 ... 5.12, 10.24:
  - We train the regularized hypothesis $\boldsymbol{h_\theta(x)}$ using λ on the Train set e.g.
    - Training gives us the best $\boldsymbol{\theta}$ for $\boldsymbol{h_\theta(x)}$ on the Train set (that minimizes the cost)
  - We pass in the (same) Train set to the cost function $\boldsymbol{J(\theta)}$ to get the training error $\boldsymbol{J}_{\text{train}}$
  - We pass in the CV set to the cost function $\boldsymbol{J(\theta)}$ to get the CV error $\boldsymbol{J}_{\text{CV}}$
  - We plot these values on a graph (we'll see it later)

- Pick the model with λ that produces the best (lowest) $\boldsymbol{J}_{\text{CV}}$
- Compute the error of this model on the Test set $\boldsymbol{J}_{\text{test}}$ which is taken as the generalization performance of the selected model

Determining How Much to Regularize

- E.g.

1. $\lambda = 0$ $\rightarrow$ $\min_{\boldsymbol{\theta}} \boldsymbol{J}(\boldsymbol{\theta})$ to get $\boldsymbol{\theta}^{(1)}$ $\rightarrow$ plot $J_{\text{train}}(\boldsymbol{\theta}^{(1)})$, $J_{\text{cv}}(\boldsymbol{\theta}^{(1)})$
2. $\lambda = 0.02$ $\rightarrow$ $\min_{\boldsymbol{\theta}} \boldsymbol{J}(\boldsymbol{\theta})$ to get $\boldsymbol{\theta}^{(2)}$ $\rightarrow$ plot $J_{\text{train}}(\boldsymbol{\theta}^{(2)}), J_{\text{cv}}(\boldsymbol{\theta}^{(2)})$

...

6. $\lambda = 0.64$ $\rightarrow$ $\min_{\boldsymbol{\theta}} \boldsymbol{J}(\boldsymbol{\theta})$ to get $\boldsymbol{\theta}^{(6)}$ $\rightarrow$ plot $J_{\text{train}}(\boldsymbol{\theta}^{(6)}), J_{\text{cv}}(\boldsymbol{\theta}^{(6)})$

...

10. $\lambda = 10.24$ $\rightarrow$ $\min_{\boldsymbol{\theta}} \boldsymbol{J}(\boldsymbol{\theta})$ to get $\boldsymbol{\theta}^{(10)}$ $\rightarrow$ plot $J_{\text{train}}(\boldsymbol{\theta}^{(10)}), J_{\text{cv}}(\boldsymbol{\theta}^{(10)})$

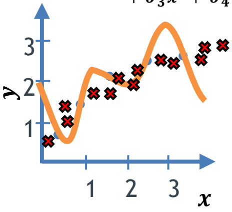
- Assume that $\lambda = 0.64$ gets the lowest J_{cv}
- Compute the error of this model on the Test set $J_{\text{test}}(\boldsymbol{\theta}^{(6)})$ which is taken as the generalization performance of the selected model

- Generally good idea to increase λ by doubling/tripling it everytime

Determining How Much to Regularize

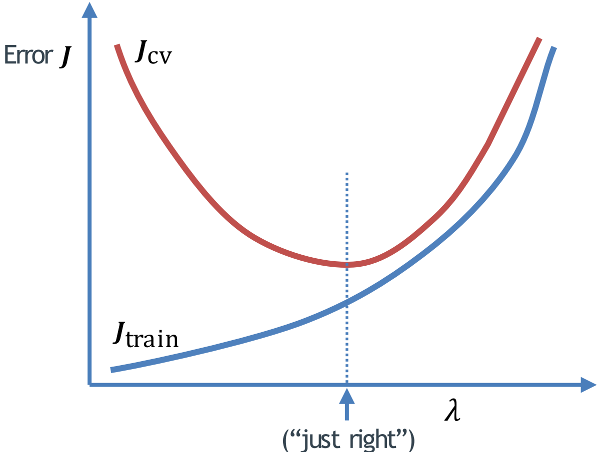
- The plot of the two errors J_{train} and J_{cv} versus the parameter we're optimizing i.e. λ:

$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

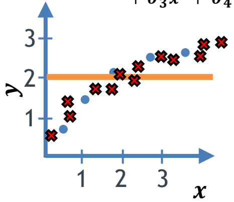


$\lambda = 0$

No regularization



$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$



$\lambda = 1000$ (very large)

$\theta_1 \approx \theta_2 \approx \theta_3 \approx \theta_4 \approx 0$

So $h_\theta(x) \approx \theta_0$

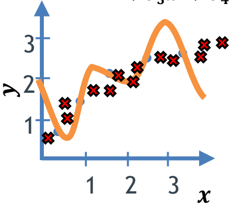
Determining How Much to Regularize

- Bias:
- $J_{\text{train}} \approx J_{\text{cv}}$ and both errors are large

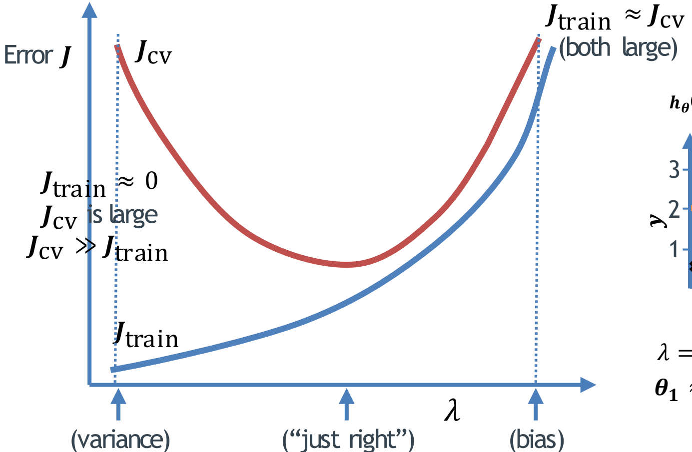
- Variance:
- $J_{\text{train}} \approx 0$ and J_{cv} is large so
- $J_{\text{cv}} \gg J_{\text{train}}$

● train
✖ cv

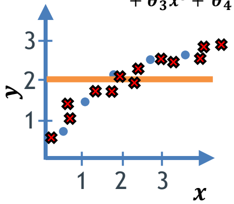
$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$



$\lambda = 0$
No regularization



$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$



$\lambda = 1000$ (very large)

$\boldsymbol{\theta_1 \approx \theta_2 \approx \theta_3 \approx \theta_4 \approx 0}$

So $h_\theta(x) \approx \theta_0$

Practical Issues

Training Set Size?

- Overfitting and Underfitting
    - Introduction
    - Polynomial Degree?
    - How Much to Regularize?
    - Training Set Size?

Determining How Much Training Data To Use

- In this case, we've got a specific model (possible with a given regularization value) that we want to analyse; we're going to vary the number of samples we use for training $\boldsymbol{m}_{\text{train}}$ i.e. Train set size

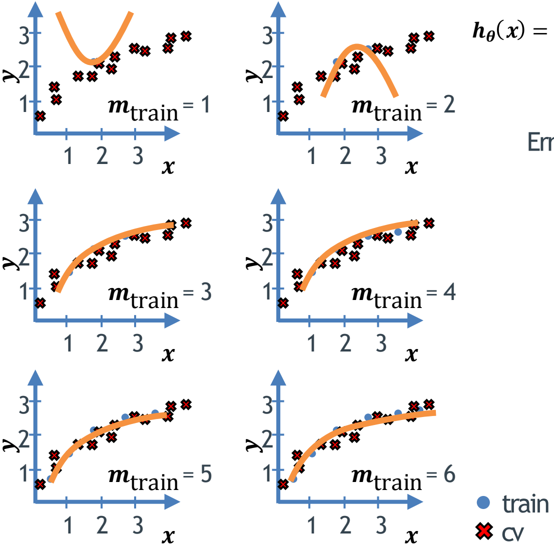
- For $\boldsymbol{m}_{\text{train}}$ ranging from (some appropriate minimum) to (some maximum) in steps of (reasonable step size):
  - We train a hypothesis $\boldsymbol{h}_{\boldsymbol{\theta}}(\boldsymbol{x})$ using only $\boldsymbol{m}_{\text{train}}$ examples in the Train set
    - Training gives us the best $\boldsymbol{\theta}$ for $\boldsymbol{h}_{\boldsymbol{\theta}}(\boldsymbol{x})$ on the Train set (that minimizes the cost)
  - We pass in the (same) Train set (same $\boldsymbol{m}_{\text{train}}$) to the cost function $\boldsymbol{J}(\boldsymbol{\theta})$ to get the training error $\boldsymbol{J}_{\text{train}}$
  - We pass in the entire CV set to the cost function $\boldsymbol{J}(\boldsymbol{\theta})$ to get the CV error $\boldsymbol{J}_{\text{cv}}$
  - We plot these values on a graph (we'll see it later)

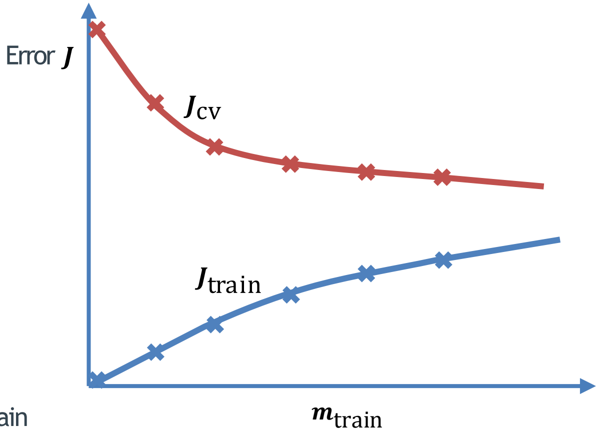
- Determine if the model has high Bias or high Variance using the graph (later):
  - If it has high Bias: try adding more features e.g. new features, higher-order terms, etc.
  - If it has high Variance: try:
    - Regularizing
    - Collecting more training data

Determining How Much Training Data To Use

- The plot of the two errors J_{train} and J_{cv} versus the parameter we're optimizing i.e. m_{train} :

$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$$

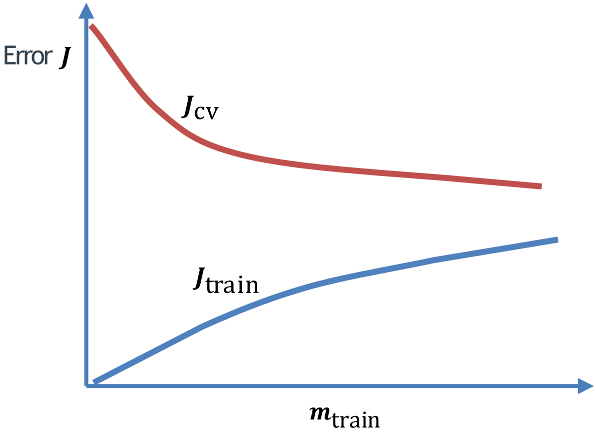




Determining How Much Training Data To Use

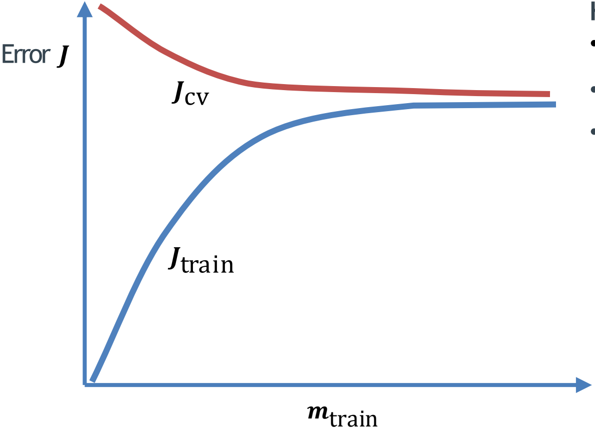
- The plot of the two errors J_{train} and J_{cv} versus the parameter we're optimizing i.e. m_{train} :

- This kind of a plot is called a "**learning curve**"
- It gives some indication of how the model is "learning" as training examples are increasing
- **In general,** this is what a learning curve looks like
- If J_{cv} is low, then the model performs well: **call it a day**. If not:
- The learning curve will take a specific shape depending on whether the model has bias or variance (next slides)
- We can then react accordingly



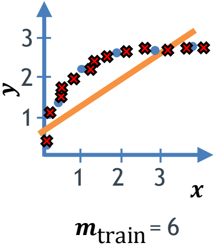
Determining How Much Training Data To Use – High Bias

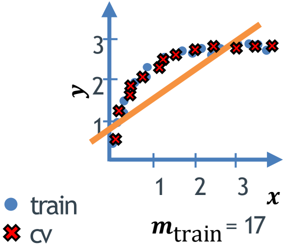
- If the model suffers from high bias, what will the learning curve will look like?
- E.g. Imagine using a straight line $\boldsymbol{h_\theta(x) = \theta_0 + \theta_1 x}$ to fit non-linear data as below



High bias:

- $J_{\text{train}} \approx J_{\text{cv}}$
- Both are very large
- Both appear to stabilize



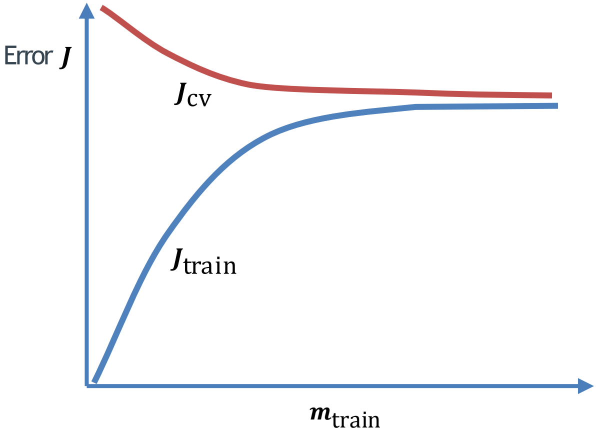


Determining How Much Training Data To Use – High Bias

- If the model suffers from high bias, what will the learning curve will look like?
- E.g. Imagine using a straight line $\boldsymbol{h_\theta(x) = \theta_0 + \theta_1 x}$ to fit non-linear data as below

- A high bias model is about the worst you can have
- In this case: no amount of data will help
    - Don't waste time/money collecting more data
- The model is just too simple

- Only one solution: back to the drawing board to make the model more complex
    - Add new features
    - Add higher-order features
    - Add combinations of features

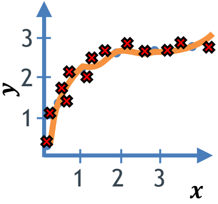


Determining How Much Training Data To Use – High Variance

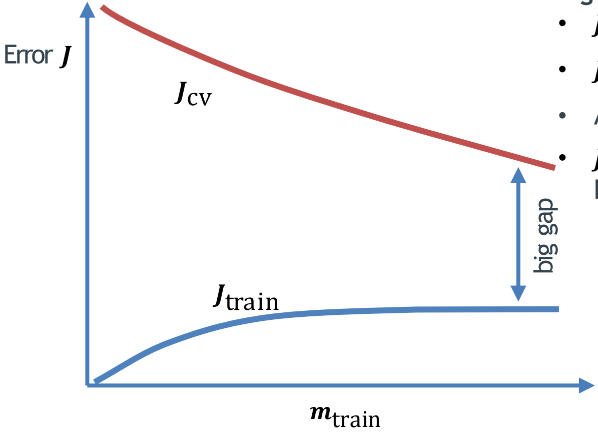
- If the model suffers from high variance, what will the learning curve will look like?
- E.g. Imagine using an extremely high-order polynomial $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_{50} x^{50}$ to fit data as below!

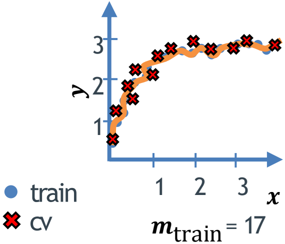
High variance:

- J_{train} is possibly very low
- $J_{\text{cv}} \gg J_{\text{train}}$
- A big gap between them
- J_{cv} doesn't appear to have stabilized



$m_{\text{train}} = 6$



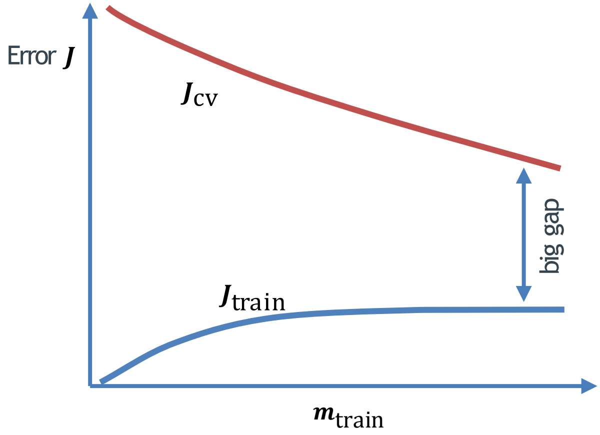


Determining How Much Training Data To Use – High Variance

- If the model suffers from high variance, what will the learning curve will look like?
- E.g. Imagine using an extremely high-order polynomial $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_{50} x^{50}$ to fit data as below!

- A high variance model is can be worked on
- In this case: more data can help
    - Extrapolating the curves to the right will bring J_{CV} lower and lower

- Before collecting more data, consider first:
    - Regularizing the model (as seen before)
    - If it doesn't work: try reducing the polynomial order (as seen before)
    - If those don't work: try using/collecting more data



THE END
Of Logistic Regression