innopolis university

Machine Learning

Prof. Adil Khan

Objectives

1. A quick recap of last lecture
2. Software defect prediction
   - predicting the number of defects
   - View this task in the context of ML
3. What is linear regression? What is its objective function? How is it motivated?
4. Deriving closed-form solution for linear regression

Recap (1)

What is Machine Learning?

- A subfield of artificial intelligence
- Computer programs that ***improve*** their ***performance*** at some task through ***experience***
- Examples: object recognition, spam detection, disease prediction, weather forecasting, etc.

Goal of Learning

- Learning or inferring a "functional" relationship between predictors and target

$$D = \{\boldsymbol{x}_i, y_i\}_{i=1}^{N}$$

$$\boldsymbol{x} \in \mathbb{R}^d$$

$$\boxed{\hat{f} \approx f \quad \textit{Goal of learning}}$$

$$y = f(x)$$

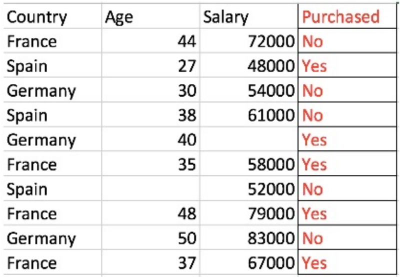
Parametric Models

$$y = f(\boldsymbol{x}; parameters)$$

$$y = f(\boldsymbol{x}; \boldsymbol{w})$$

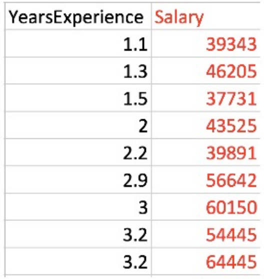
$$y = f(x; w_0, w_1) = w_0 + w_1 x$$

Classification and Regression



Country	Age	Salary	Purchased
France	44	72000	No
Spain	27	48000	Yes
Germany	30	54000	No
Spain	38	61000	No
Germany	40		Yes
France	35	58000	Yes
Spain		52000	No
France	48	79000	Yes
Germany	50	83000	No
France	37	67000	Yes

Classification

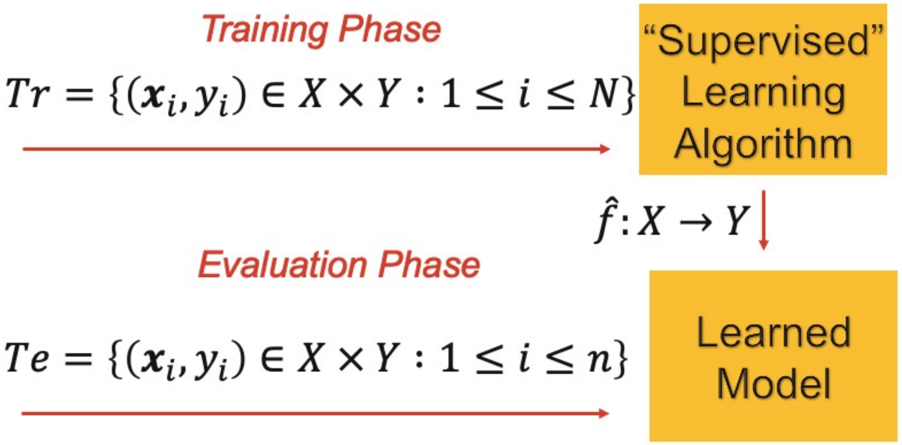


YearsExperience	Salary
1.1	39343
1.3	46205
1.5	37731
2	43525
2.2	39891
2.9	56642
3	60150
3.2	54445
3.2	64445

Regression

Recap (2)

How do we implement it?



Training Phase

$Tr = \{(\boldsymbol{x}_i, y_i) \in X \times Y : 1 \leq i \leq N\}$

"Supervised" Learning Algorithm

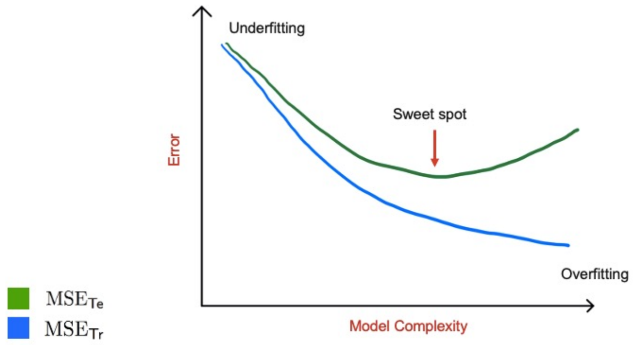
$\hat{f}: X \rightarrow Y$

Evaluation Phase

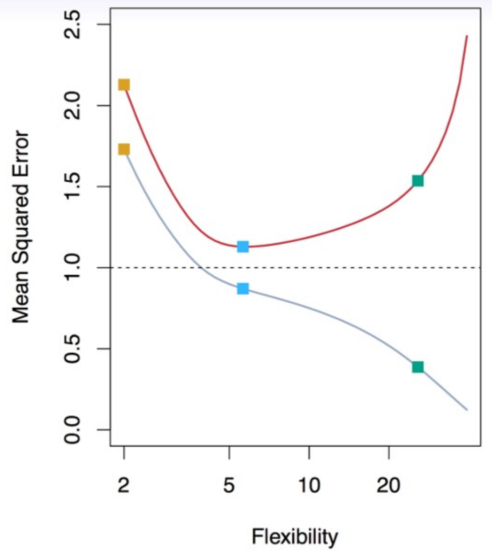
$Te = \{(\boldsymbol{x}_i, y_i) \in X \times Y : 1 \leq i \leq n\}$

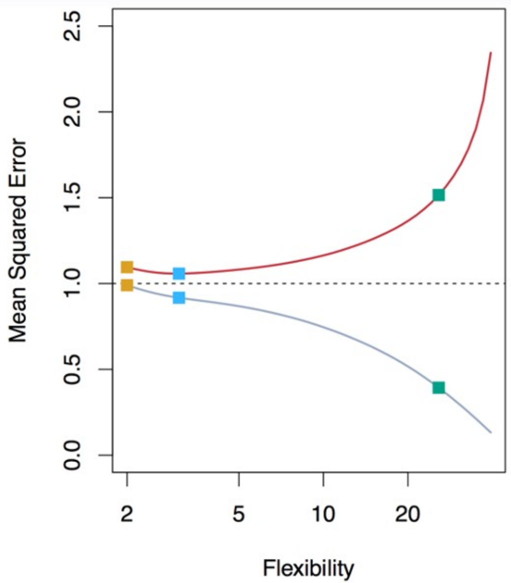
Learned Model

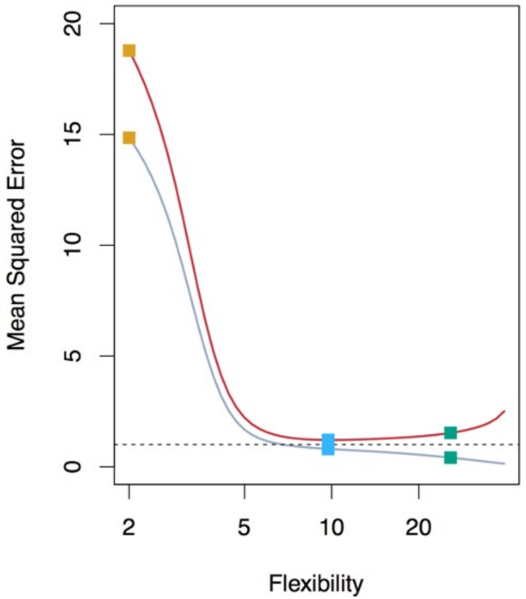
Model Complexity or Flexibility



Underfitting and Overfitting







Software Defects

- Also known as
  - Bugs
  - Problems
  - Error
  - Anomaly
  - …
- We say a software has defects if
  - It does something that it should not
  - It does not do something that it should
  - …

Problem Sources

- Requirements definition
- Design
- Implementation
- Inadequate testing
- …

Adverse Effects of Defected Software

- **Healthcare:** loss of lives, breech of data, etc.
- **Communications:** Loss of data, etc.
- **Defense:** Misidentification of the target, etc.
- **Electric power:** power outages, injuries, etc.
- **Money management:** fraud, shutdown of stockexchange, etc.
- …

Bug-free Software

- Can you gaurantee that the software systems that you or your team will develop would be bug-free?

- Even if we will be extra careful, still it is extremely hard to make software bug-free because
  - As softwares get more features and supports more platform it becomes increasingly difficult to make it bug-free

Detection vs. Prediction

- Software defect detection
  - Identify defects
  - Fix them

- But usually the bugs found later cost more to fix

- Software defect prediction
  - Advance information on likely defects
  - .. Number of defects ..

You Now Know …

1. What are software bugs?
2. What are their sources?
3. What are their adverse effects?
4. How unlikely is it to create bug-free software?
5. How important is it to be able to predict defect’s related information?

❖ Now, let’s see how can we predict ***number of defects*** in a software using machine learning

Predicting Number of Defects From the Point of view of ML

Given a computer program, let’s say p_i

1. What will be the $\boldsymbol{x}_i$?
2. What will be the y_i?

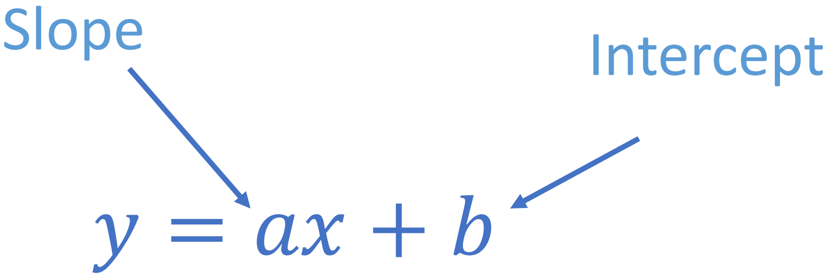
Predicting Number of Defects From the Point of view of ML

Thus, the goal of learning is to estimate following functional relationship

$$\underset{y_i}{\underline{\#\ of\ defects\ in\ p_i}} = f(\underset{\boldsymbol{x}_i}{\underline{features\ of\ p_i}})$$

Let’s take a detour!

Equation of a Straight Line

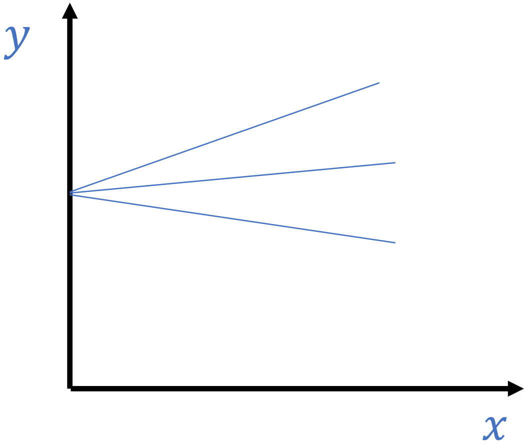


Slope

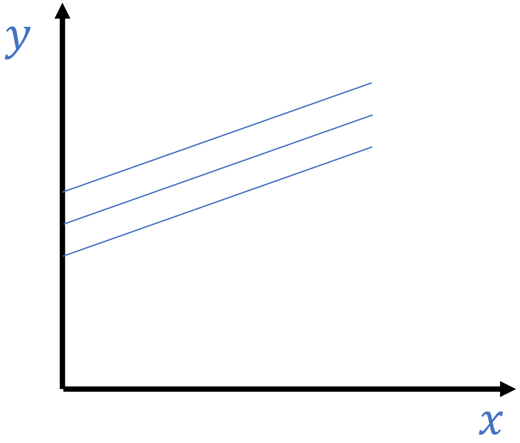
Intercept

$$y = ax + b$$

Different Slopes and Intercepts



Different Slopes



Different Intercepts

Back to our Regression Problem

$$\#\ of\ defects\ in\ p_i = f(features\ or\ behavior\ of\ p_i)$$

- Let’s suppose there is just one feature,
- Then we can write the above expression as

$$\mathrm{y} = w_1 x + w_0$$

- Which is the same equation as that of a straight line
- And that is why, we call it “Simple Linear Regression”

In General, Linear Regression

$$\boxed{y = w_0 + w_1 x_1 + w_2 x_2 + \cdots w_p x_p}$$

- The response variable is quantitative
- The relationship between response and predictors is assumed to be linear in the inputs
- Thus we are restricting ourselves to a hypothesis space of linear functions

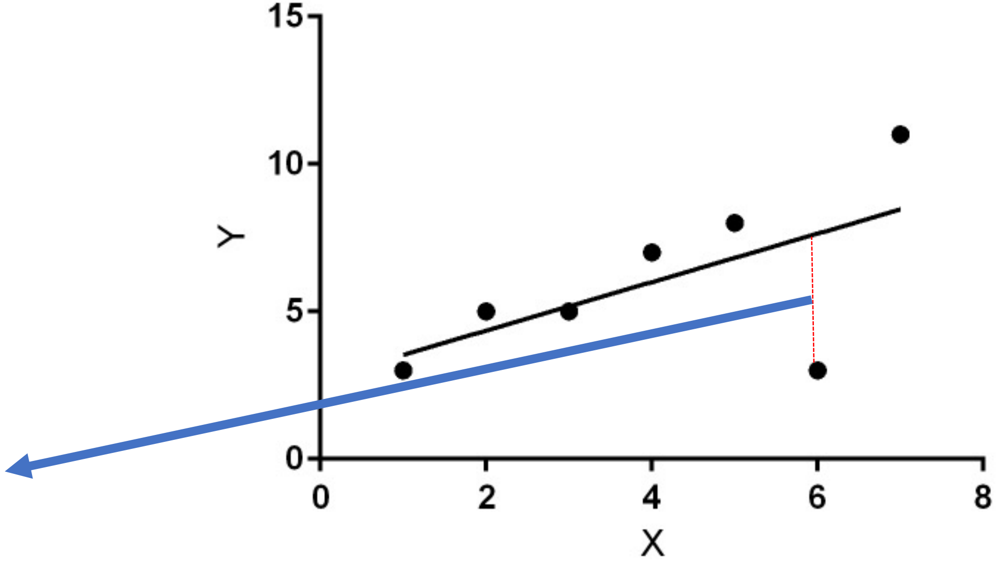
Why Linear Regression

- Although it may seem overly simplistic, linear regression is extremely useful

  - Easy for inferencing
  - Serves as a good jumping point for more powerful and complex approaches

How Do We Train Linear Regression Model?

$$f(x_i) = w_0 + w_1 x_i$$



$$e_i = y_i - f(x_i)$$

Mean Squared Error (MSE)

$$f(x_i) = w_0 + w_1 x_i$$

$$e_i = y_i - f(x_i)$$

$$\mathcal{L}(w_0, w_1) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - (w_0 + w_1 x_i)\right)^2$$

We need to find the value of parameters that minimize this cost or loss function.

Objective Function

$$f(x_i) = w_0 + w_1 x_i$$

$$e_i = y_i - f(x_i)$$

$$\mathcal{L}(w_0, w_1) = \frac{1}{n}\sum_{i=1}^{n}\big(y_i - (w_0 + w_1 x_i)\big)^2$$

$$\underset{w_o, w_1}{\operatorname{argmin}}\ \mathcal{L}(w_0, w_1)$$

The term argmin is the shorthand for “find the argument that minimizes … “

Let’s take a detour, again!

Derivative

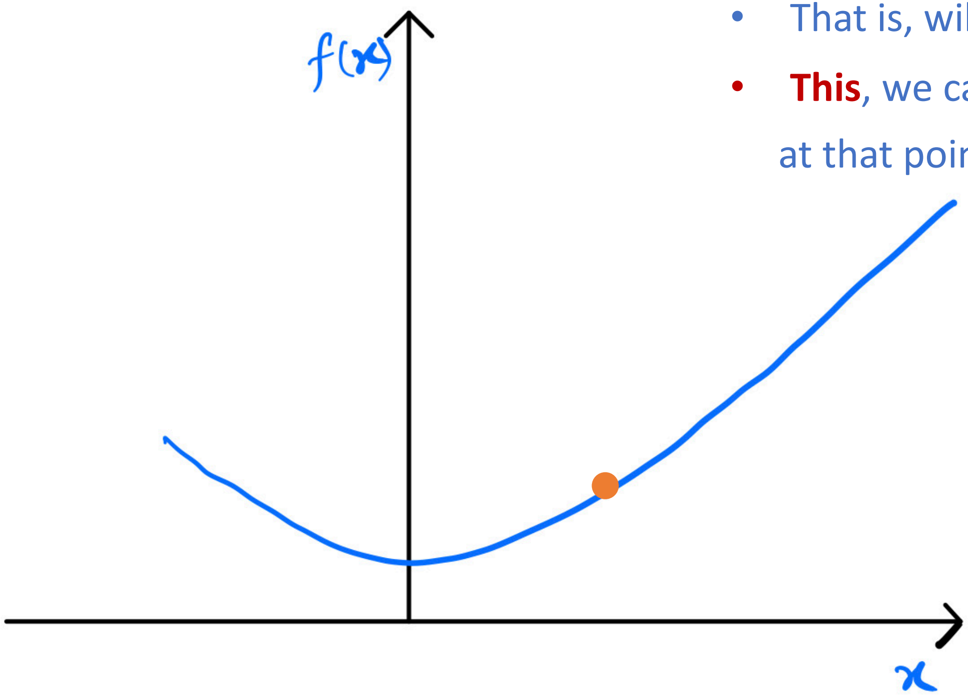
- The derivate is the heart of calculus
- The derivative of a function of a single variable is defined as

$$f'(x) = \lim_{dx \longrightarrow 0} \frac{f(x + dx) - f(x)}{dx}$$

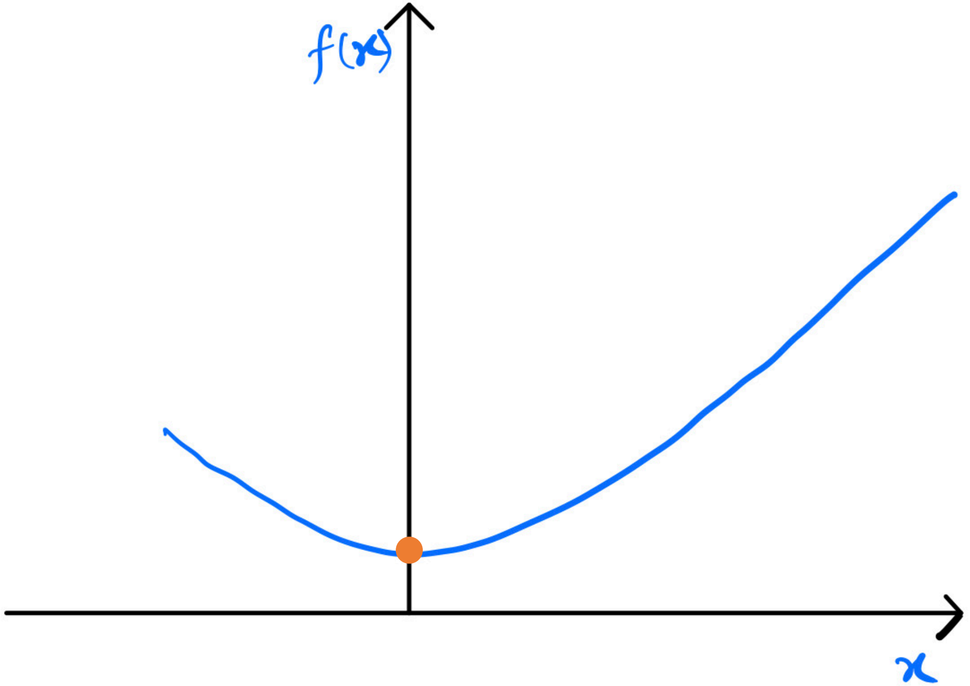
- But the question is, what can we use it for?

Use of Derivative

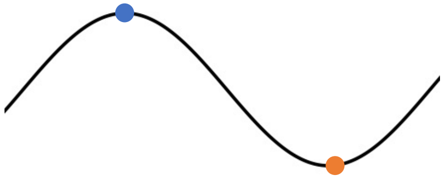
- Let’s say we are standing at this point
- We would like to know, what will happen if we increased x
- That is, will the function’s value increase or decrease?
- **This**, we can find by taking the derivate of the function at that point!



What will be $f'(x)$ at this point?

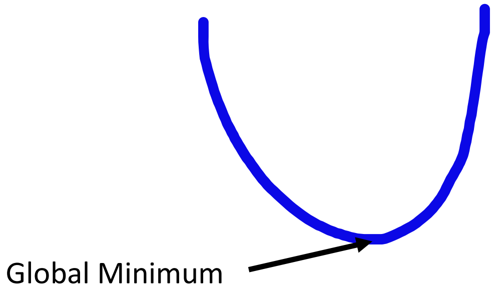


Maximum and Minimum

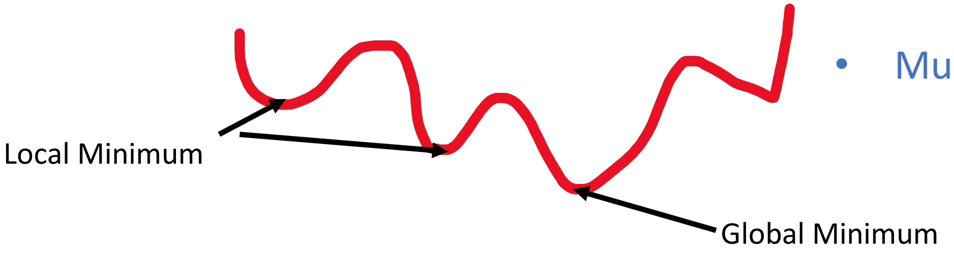


Point x	f' slightly left to x	f' at x	f' slightly right to x
Maximum ●	> 0	0	< 0
Minimum ●	< 0	0	> 0

Convex vs Non-convex



- Unique minimum – its global minimum



- Multiple minimum points – local and global minimum

Recap

1. Objective function (which we want to optimize)

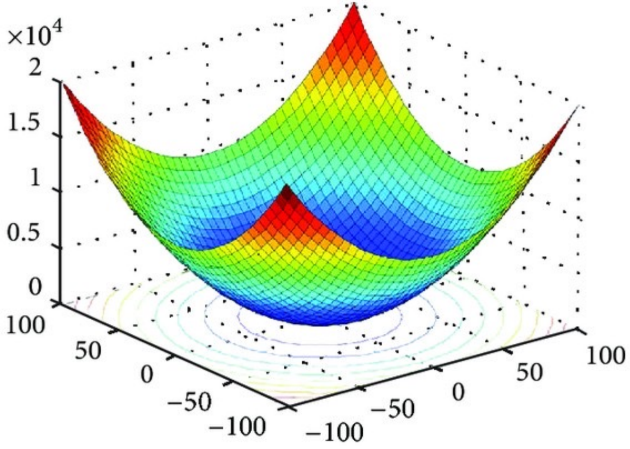
2. Derivative (a mathematical tool that can tell us whether a function increases or decreases as we slightly increase its input)

3. Maximum and Minimum points (where derivative is 0)

4. Convex functions (we love them as they have only one global minimum point making them easier to use in optimization problems)

Back to Our Objective Function

$$\mathcal{L}(w_0, w_1) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - (w_0 + w_1 x_i)\right)^2$$



MSE is convex: at the unique minimum of our loss function, its “partial” derivative with respect to w_0 and w_1 will be zero!

The Least Square Solution

$$\mathcal{L}(w_0, w_1) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - (w_0 + w_1 x_i)\right)^2$$

1. Compute partial derivatives of the loss function with respect to w_0 and w_1
2. Set them to 0
3. And solve for w_0 and w_1

The Least Square Solution (2)

$$\mathcal{L}(w_0, w_1) = \frac{1}{n}\sum_{i=1}^{n}\left(w_1^2 x_i^2 + 2w_1 x_i(w_o - y_i) + w_0^2 - 2w_0 y_i + y_i^2\right)$$

- Let’s take the partial derivatives of the loss function with respect to w_0,
- We can start by removing the terms that do not include w_0

$$\frac{1}{n}\sum_{i=1}^{n}(w_0^2 + 2w_1 x_i w_0 - 2w_0 y_i)$$

The Least Square Solution (3)

$$\frac{1}{n}\sum_{i=1}^{n}(w_0^2 + 2w_1 x_i w_0 - 2w_0 y_i)$$

- Rearrange the terms not indexed by n outside of the summation,

$$= w_0^2 + 2w_1 w_0 \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right) - 2w_0 \frac{1}{n}\left(\sum_{i=1}^{n} y_i\right)$$

Basic Properties and Formulas

If $f(x)$ and $g(x)$ are differentiable functions (the derivative exists), c and n are any real numbers,

1. $(cf)' = c\,f'(x)$
2. $(f \pm g)' = f'(x) \pm g'(x)$
3. $(f\,g)' = f'\,g + f\,g'$ – **Product Rule**
4. $\left(\dfrac{f}{g}\right)' = \dfrac{f'\,g - f\,g'}{g^2}$ – **Quotient Rule**
5. $\dfrac{d}{dx}(c) = 0$
6. $\dfrac{d}{dx}\left(x^n\right) = n\,x^{n-1}$ – **Power Rule**
7. $\dfrac{d}{dx}\left(f\left(g(x)\right)\right) = f'\left(g(x)\right)g'(x)$

   This is the **Chain Rule**

Common Derivatives

$$\frac{d}{dx}(x) = 1$$

$$\frac{d}{dx}(\sin x) = \cos x$$

$$\frac{d}{dx}(\cos x) = -\sin x$$

$$\frac{d}{dx}(\tan x) = \sec^2 x$$

$$\frac{d}{dx}(\sec x) = \sec x \tan x$$

$$\frac{d}{dx}(\csc x) = -\csc x \cot x$$

$$\frac{d}{dx}(\cot x) = -\csc^2 x$$

$$\frac{d}{dx}\left(\sin^{-1} x\right) = \frac{1}{\sqrt{1-x^2}}$$

$$\frac{d}{dx}\left(\cos^{-1} x\right) = -\frac{1}{\sqrt{1-x^2}}$$

$$\frac{d}{dx}\left(\tan^{-1} x\right) = \frac{1}{1+x^2}$$

$$\frac{d}{dx}\left(a^x\right) = a^x \ln(a)$$

$$\frac{d}{dx}\left(\mathbf{e}^x\right) = \mathbf{e}^x$$

$$\frac{d}{dx}\left(\ln(x)\right) = \frac{1}{x}, \quad x > 0$$

$$\frac{d}{dx}\left(\ln|x|\right) = \frac{1}{x}, \quad x \neq 0$$

$$\frac{d}{dx}\left(\log_a(x)\right) = \frac{1}{x \ln a}, \quad x > 0$$

The Least Square Solution (4)

$$w_0^2 + 2w_1 w_0 \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right) - 2w_0 \frac{1}{n}\left(\sum_{i=1}^{n} y_i\right)$$

- Now, Let’s take the partial derivative with respect to w_0,

$$\frac{\partial \mathcal{L}}{\partial w_0} = 2w_0 + 2w_1 \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right) - 2\frac{1}{n}\left(\sum_{i=1}^{n} y_i\right)$$

The Least Square Solution (5)

$$\frac{\partial \mathcal{L}}{\partial w_0} = 2w_0 + 2w_1 \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right) - 2\frac{1}{n}\left(\sum_{i=1}^{n} y_i\right)$$

- Now equate the partial derivative to zero,

$$2w_0 + 2w_1 \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right) - 2\frac{1}{n}\left(\sum_{i=1}^{n} y_i\right) = 0$$

$$2w_0 = 2\frac{1}{n}\left(\sum_{i=1}^{n} y_i\right) - 2w_1 \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)$$

The Least Square Solution (6)

$$2w_0 = 2\frac{1}{n}\left(\sum_{i=1}^{n} y_i\right) - 2w_1\frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)$$

$$w_0 = \frac{1}{n}\left(\sum_{i=1}^{n} y_i\right) - w_1\frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)$$

$$\boxed{w_0 = \overline{y} - w_1\overline{x}}$$

The Least Square Solution (7)

$$w_0 = \overline{y} - w_1\overline{x}$$

- Now, we must do the same process for w_1

The Least Square Solution (8)

$$\mathcal{L}(w_0, w_1) = \frac{1}{n}\sum_{i=1}^{n}\left(w_1^2 x_i^2 + 2w_1 x_i (w_o - y_i) + w_0^2 - 2w_0 y_i + y_i^2\right)$$

- We will now take the partial derivatives of the loss function with respect to w_1,
- We can start by removing the terms that do not include w_1

$$\frac{1}{n}\sum_{i=1}^{n}\left(w_1^2 x_i^2 + 2w_1 x_i w_0 - 2w_1 x_i y_i\right)$$

The Least Square Solution (9)

$$\frac{1}{n}\sum_{i=1}^{n}\left(w_1^2 x_i^2 + 2w_1 x_i w_0 - 2w_1 x_i y_i\right)$$

- Rearrange the terms not indexed by n outside of the summation,

$$= w_1^2 \frac{1}{n}\left(\sum_{i=1}^{n} x_i^2\right) + 2w_1 \frac{1}{n}\left(\sum_{i=1}^{n} x_i \left(w_0 - y_i\right)\right)$$

The Least Square Solution (10)

$$w_1^2 \frac{1}{n}\left(\sum_{i=1}^{n} x_i^2\right) + 2w_1 \frac{1}{n}\left(\sum_{i=1}^{n} x_i\,(w_0 - y_i)\right)$$

- Now, Let’s take the partial derivative with respect to w_1,

$$\frac{\partial \mathcal{L}}{\partial w_1} = 2w_1 \frac{1}{n}\left(\sum_{i=1}^{n} x_i^2\right) + 2\frac{1}{n}\left(\sum_{i=1}^{n} x_i\,(w_0 - y_i)\right)$$

The Least Square Solution (11)

$$w_1^2 \frac{1}{n}\left(\sum_{i=1}^{n} x_i^2\right) + 2w_1 \frac{1}{n}\left(\sum_{i=1}^{n} x_i \left(w_0 - y_i\right)\right) \qquad w_0 = \overline{y} - w_1\overline{x}$$

- Now, Let’s take the partial derivative with respect to w_1,

$$\frac{\partial \mathcal{L}}{\partial w_1} = 2w_1 \frac{1}{n}\left(\sum_{i=1}^{n} x_i^2\right) + 2\frac{1}{n}\left(\sum_{i=1}^{n} x_i \left(\overline{y} - w_1\overline{x} - y_i\right)\right)$$

The Least Square Solution (12)

$$\frac{\partial \mathcal{L}}{\partial w_1} = 2w_1 \frac{1}{n}\left(\sum_{i=1}^{n} x_i^2\right) + 2\frac{1}{n}\left(\sum_{i=1}^{n} x_i\left(\overline{y} - w_1\overline{x} - y_i\right)\right)$$

- Let’s expand the right hand side

$$= 2w_1 \frac{1}{n}\left(\sum_{i=1}^{n} x_i^2\right) + 2\overline{y}\frac{1}{n}\left(\sum_{i=1}^{n} x_i\right) - 2w_1\overline{x}\frac{1}{n}\left(\sum_{i=1}^{n} x_i\right) - 2\frac{1}{n}\left(\sum_{i=1}^{n} x_i y_i\right)$$

The Least Square Solution (13)

$$= 2w_1 \frac{1}{n}\left(\sum_{i=1}^{n} x_i^2\right) + 2\overline{y}\frac{1}{n}\left(\sum_{i=1}^{n} x_i\right) - 2w_1\overline{x}\frac{1}{n}\left(\sum_{i=1}^{n} x_i\right) - 2\frac{1}{n}\left(\sum_{i=1}^{n} x_i y_i\right)$$

- We can rewrite it as

$$= 2w_1\overline{x^2} + 2\overline{y}\ \overline{x} - 2w_1\overline{x}\ \overline{x} - 2\overline{xy}$$

$$= 2w_1\left(\overline{x^2} - (\ \overline{x}\ )^2\right) + 2\overline{y}\ \overline{x} - 2\overline{xy}$$

The Least Square Solution (14)

$$\frac{\partial \mathcal{L}}{\partial w_1} = 2w_1\left(\overline{x^2} - (\,\overline{x}\,)^2\right) + 2\overline{y}\;\overline{x} - 2\overline{xy}$$

- Let’s set it to 0 and solve for w_1

$$2w_1\left(\overline{x^2} - (\,\overline{x}\,)^2\right) = 2\overline{xy} - 2\overline{y}\;\overline{x}$$

$$w_1\left(\overline{x^2} - (\,\overline{x}\,)^2\right) = \overline{xy} - \overline{y}\;\overline{x}$$

$$w_1 = \frac{\overline{xy} - \overline{x}\;\overline{y}}{\overline{x^2} - (\,\overline{x}\,)^2}$$

The Least Square Solution (Summary)

$$\mathcal{L}(w_0, w_1) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - (w_0 + w_1 x_i)\right)^2$$

$$\boxed{\begin{aligned} w_0 &= \bar{y} - w_1\bar{x} \\[2em] w_1 &= \frac{\overline{xy} - \bar{x}\,\bar{y}}{\overline{x^2} - (\bar{x})^2} \end{aligned}}$$

Alternatives

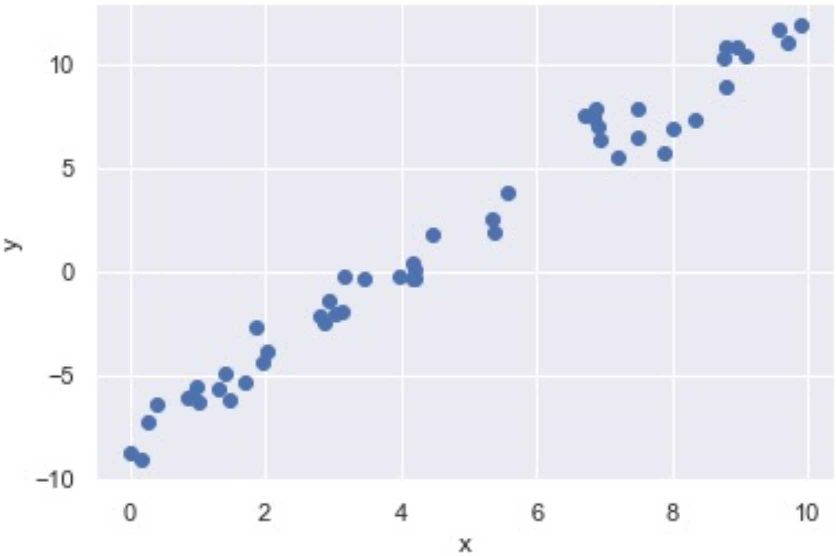
- You just learned how to estimate the parameters of LR using the method of least square

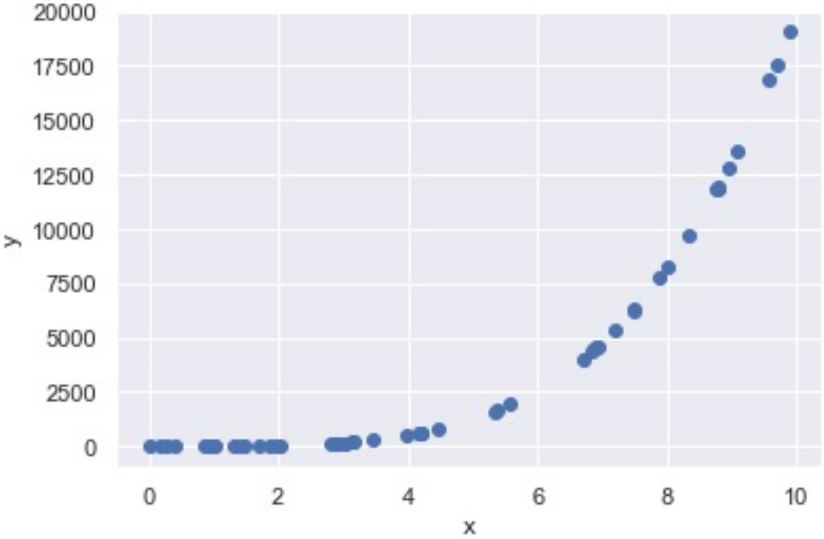
- But there are other ways to do this, especially when we are dealing with data that cannot fit in the memory

- One such, and a very important method, is *Gradient Descent*

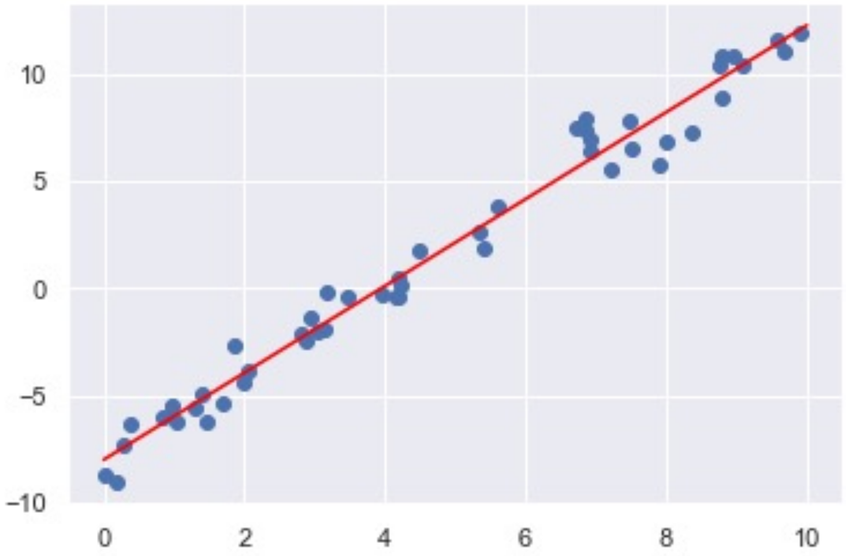
Extending Linear Regression

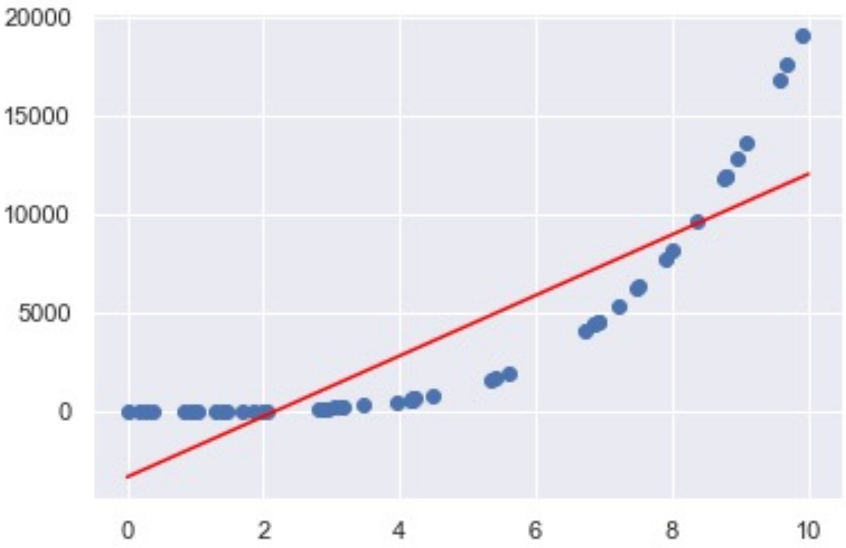
Non-Linear Relationship between Predictors and Response





Non-Linear Relationship between Predictors and Response (2)





Polynomial Regression

- Using the same framework that we learned, to fit a family of more complex models through a transformation of predictors

- Linear model has the following form

$$y = w_0 + w_1 x$$

- It is linear in both predictor (x) and parameters (w_0, w_1)

- Let’s keep it linear in parameters, but make it quadratic in predictors

Polynomial Regression (2)

- That is,

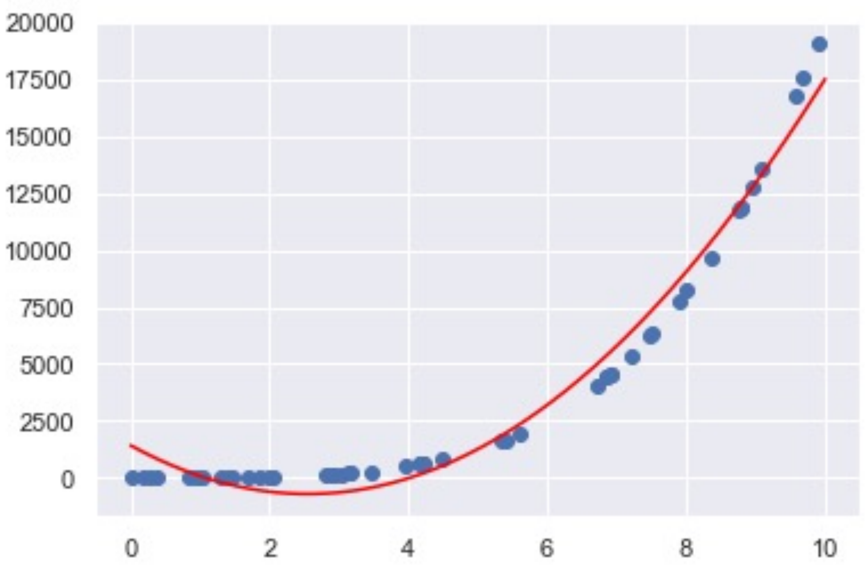
$$y = w_0 + w_1 x + w_2 x^2$$

- More generally,

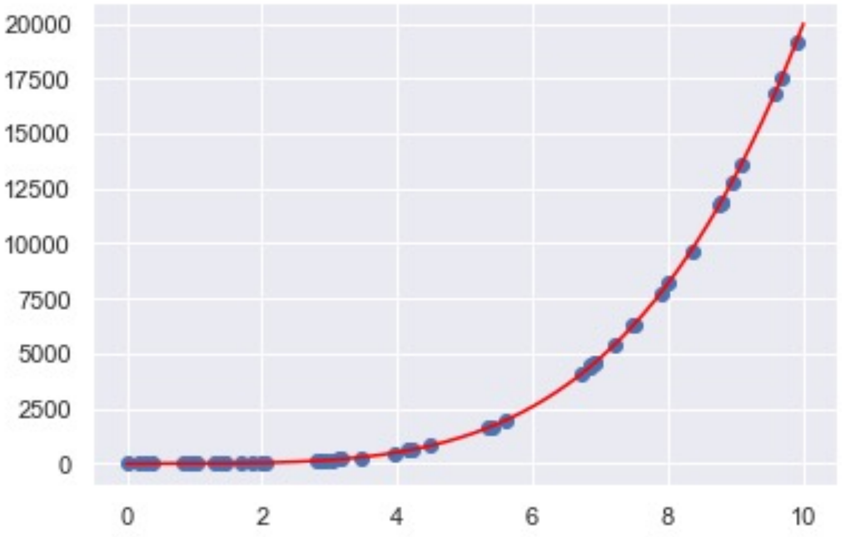
$$y = w_0 + w_1 x + w_2 x^2 + \cdots + w_d x^d$$

- Do not forget, “the model is still linear in parameters”

Polynomial Regression (3)



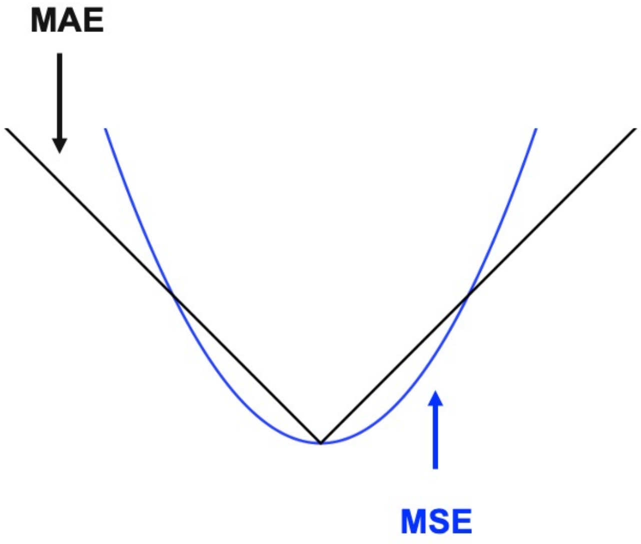
Order or Degree 2



Order or Degree 4

MSE vs MAE

- MAE
  - Good: when there could be outliers in the data
  - Bad: because its derivative is the same everywhere
- MSE
  - Bad: when there are outliers in the data
  - Good: gradient is large for large loss and decreases as loss approaches 0
- Then there is Huber loss
  - Goods of both MSE and MAE
  - But has extra hyperparameter that needs tuning



Summary

- Importance of prediting (number of) defects in software
- Analyzing the task from the point of view of ML – to see that it’s a regression task
- Formulating the learning objective
- Solving the objective
  - Least Square Solution
- Next Lecture:
  - Gradient Descent