ML-101:
Bias vs. Variance, Regularisation, PCA

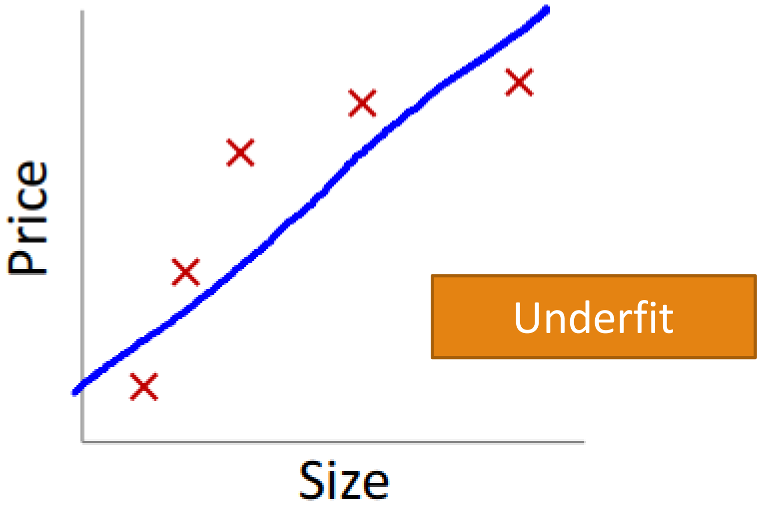
BY SARTHAK CONSUL

Overfitting

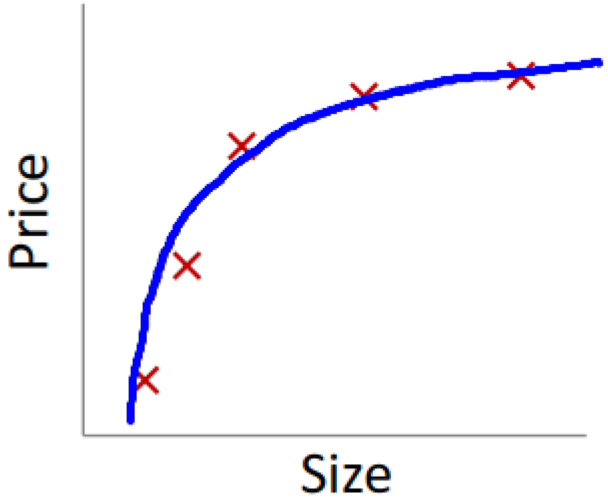
- Model trained does not ‘generalize’ well
- Excellent fit for training set, but poor fitting observed while testing
- Caused due to noise/ random fluctuations present in training data

- Occurs when too many features are used

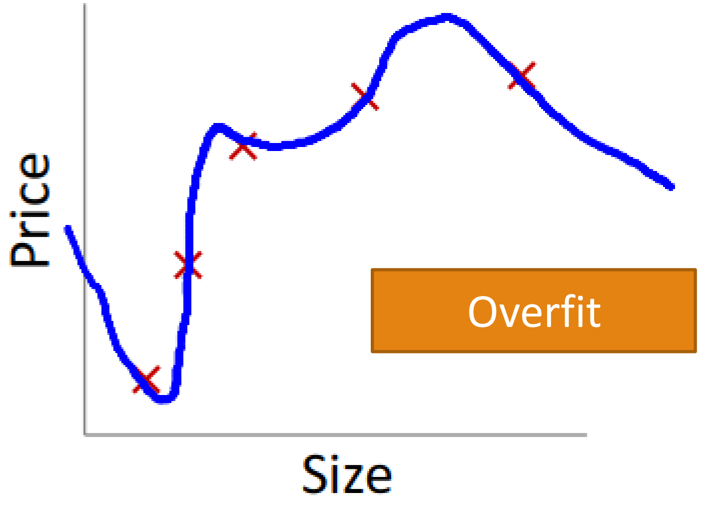
Overfitting in Linear Regression



$$\theta_0 + \theta_1 x$$



$$\theta_0 + \theta_1 x + \theta_2 x^2$$



$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

Overfitting in Polynomial Regression

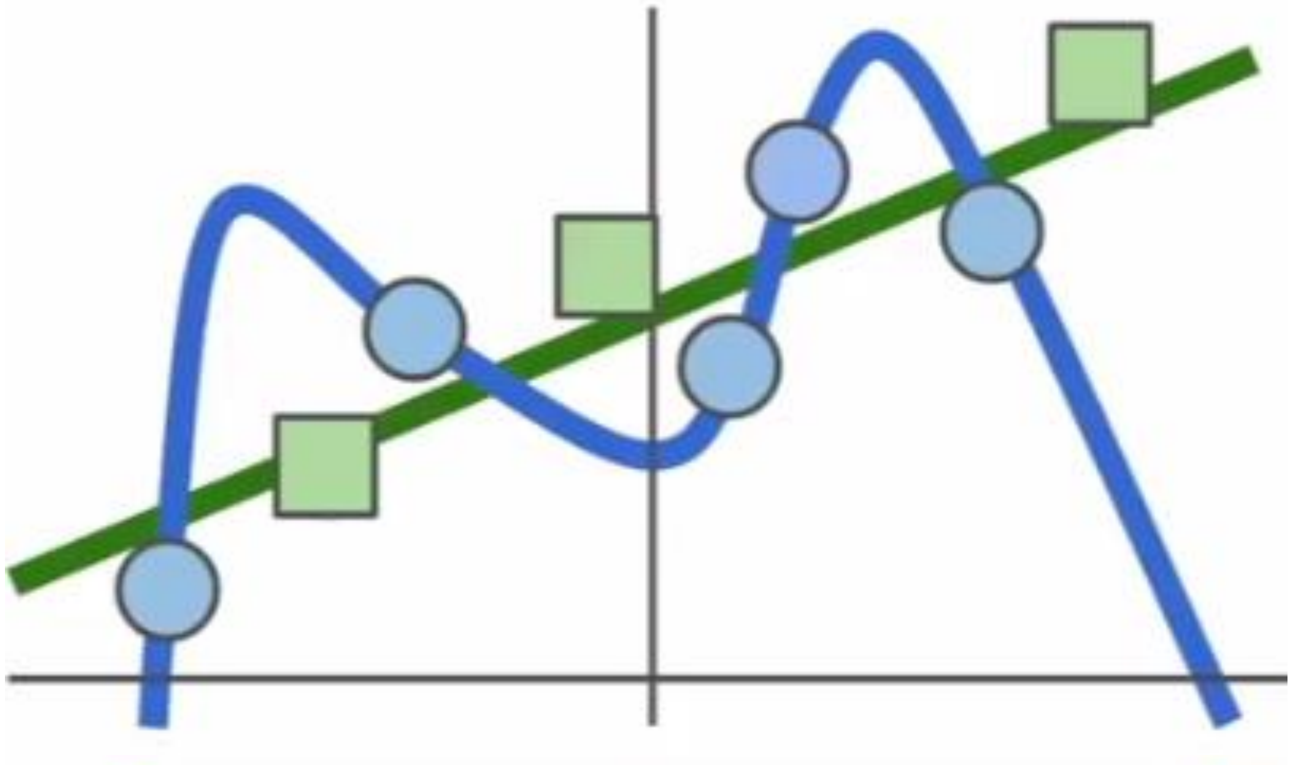
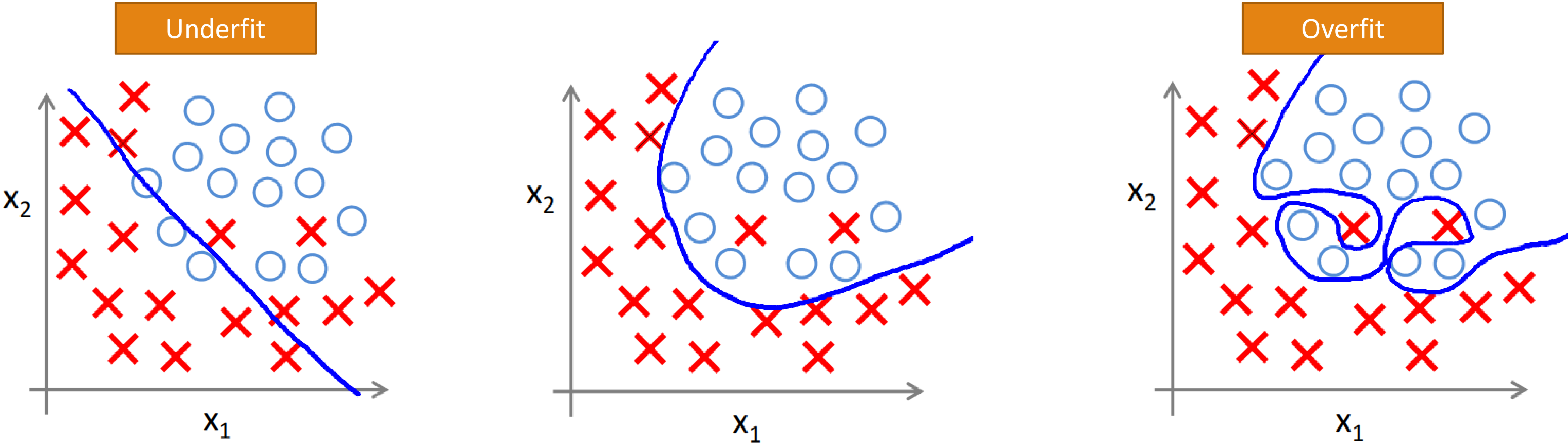


Image Source: Coursera - Machine Learning by Andrew Ng

Overfitting in Logistic Regression



$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2)$$

$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \ldots)$$

Addressing Overfitting

- Reducing the no. of features
  - Manual Selection
  - Model Selection Algorithm
- Cross Validation
- Aggregation
- Regularisation

Regularisation

- Additional constraints so that smaller values of parameters
- Implementing Occam's Razor
- Governed by regularisation constant $\lambda(>0)$

- What if λ is too small? Overfitting is not fixed

  What if λ is too large? Underfitting

Regularised Linear Regression

- Smaller values of parameters
- Cost Function

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + \frac{\lambda}{2m} \sum_{j=1}^{n} \theta_j^2$$

- Gradient Descent

$$\theta_j := \theta_j\left(1 - \alpha\frac{\lambda}{m}\right) - \alpha\frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right) x_j^{(i)}$$

Θ_0 is not constrained and so its GD doesn't change

Regularised Logistic Regression

- Cost Function

$$J(\theta) = -\left[\frac{1}{m}\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) log\big(1 - h_\theta(x^{(i)})\big)\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

- Gradient Descent

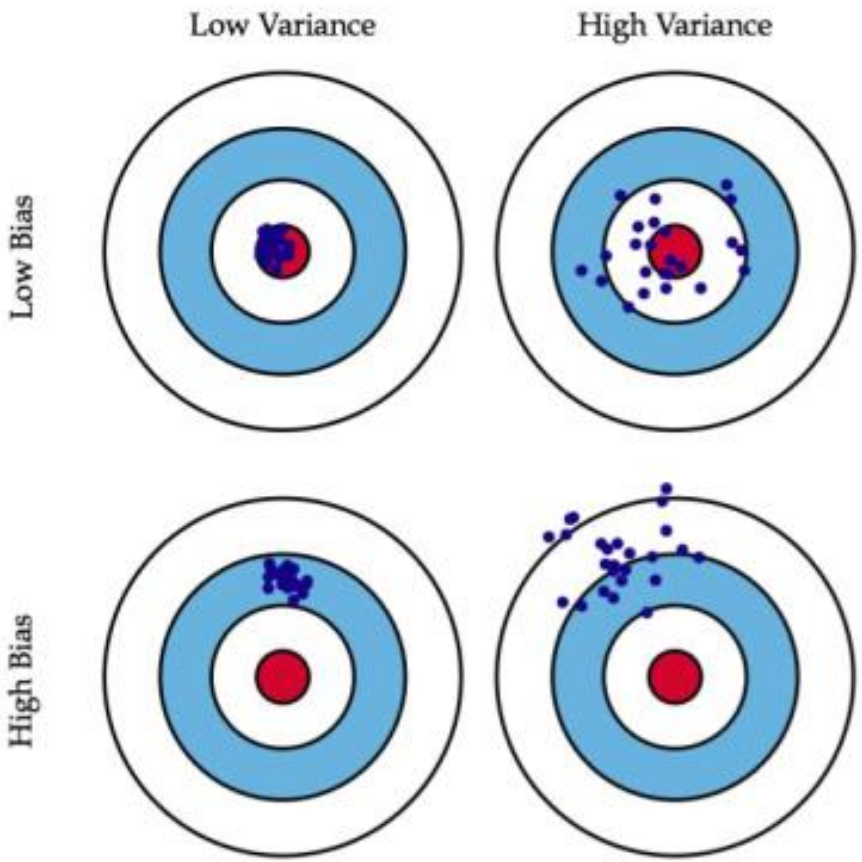
$$\theta_j := \theta_j(1 - \alpha\frac{\lambda}{m}) - \alpha\frac{1}{m}\sum_{i=1}^{m}\big(h_\theta(x^{(i)}) - y^{(i)}\big)x_j^{(i)}$$

Bias vs Variance Trade-off

- Error is has 3 components:
  i. Bias
  ii. Variance
  iii. Imperfections that can't be dealt with

$$\text{Error} = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error}$$

Bias and Variance



Reasons for bias and variance

- Bias
  - Wrong assumptions
  - High value of λ

- Variance
  - Taking very few samples
  - Too many features, small λ

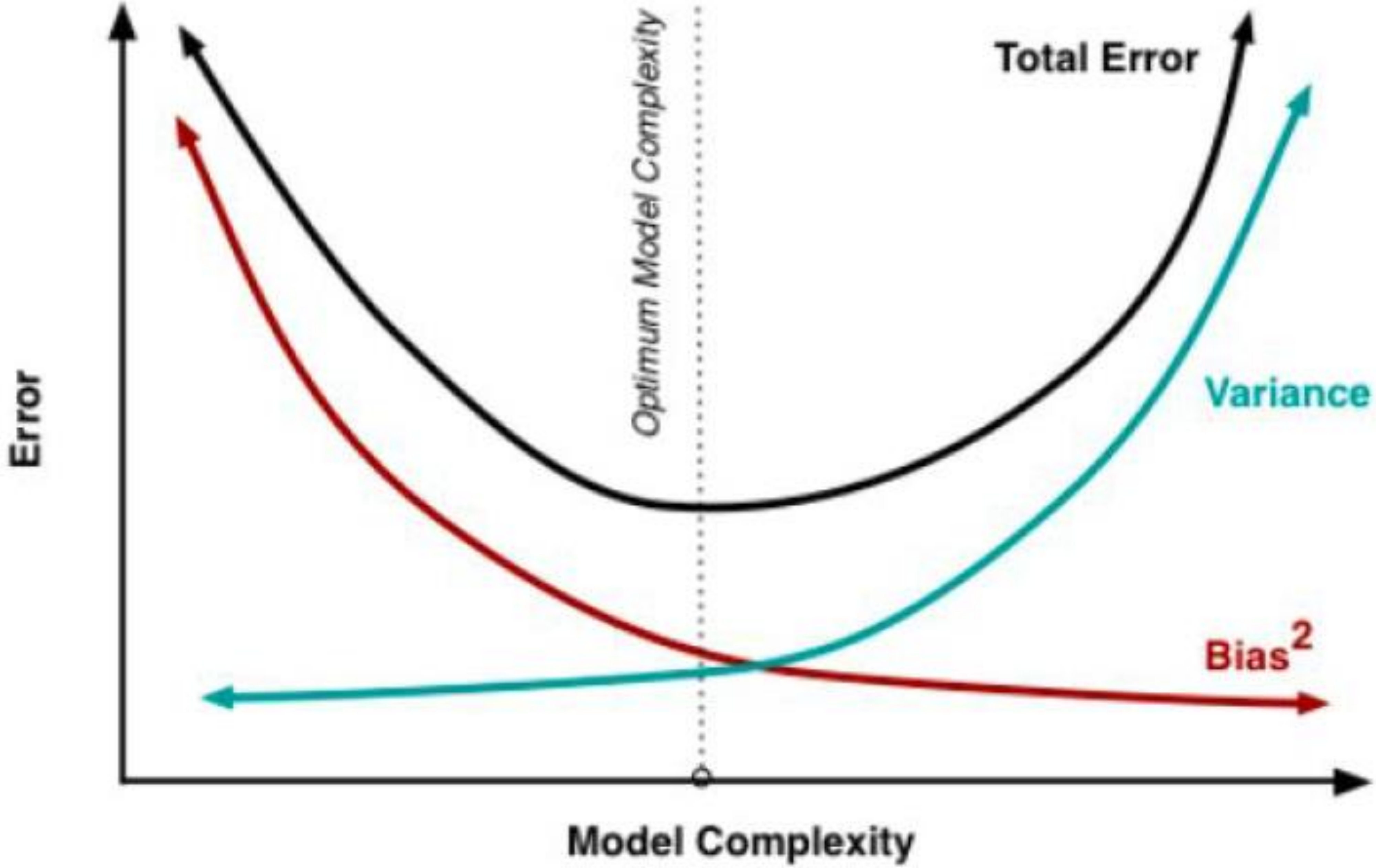

Optimum Model Complexity
Total Error
Error
Variance
Bias2
Model Complexity

Image Source: https://www.kaggle.com/residentmario/learning-curves-with-zillow-economics-data/code

Learning Curves

- High Bias: $J_{train}(\theta)$ is high, $J_{train}(\theta) \approx J_{test}(\theta)$
- High Variance: $J_{train}(\theta)$ is low, $J_{train}(\theta) << J_{test}(\theta)$

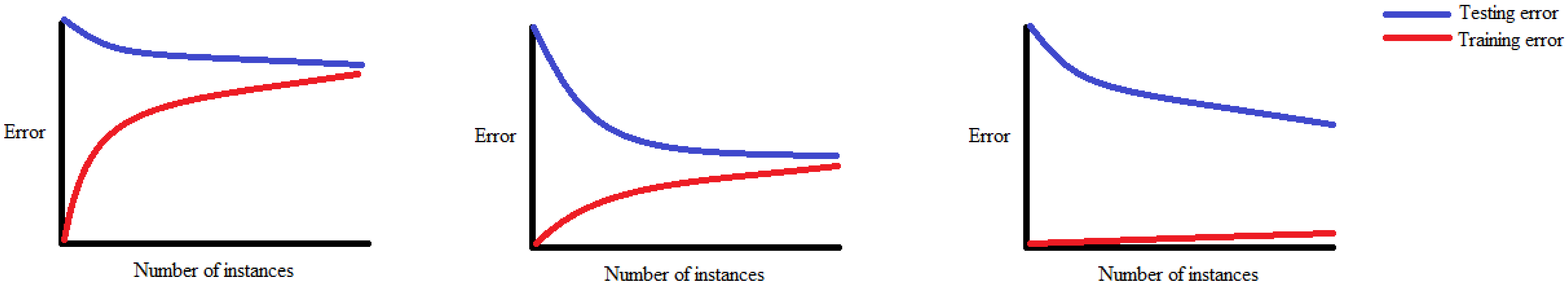


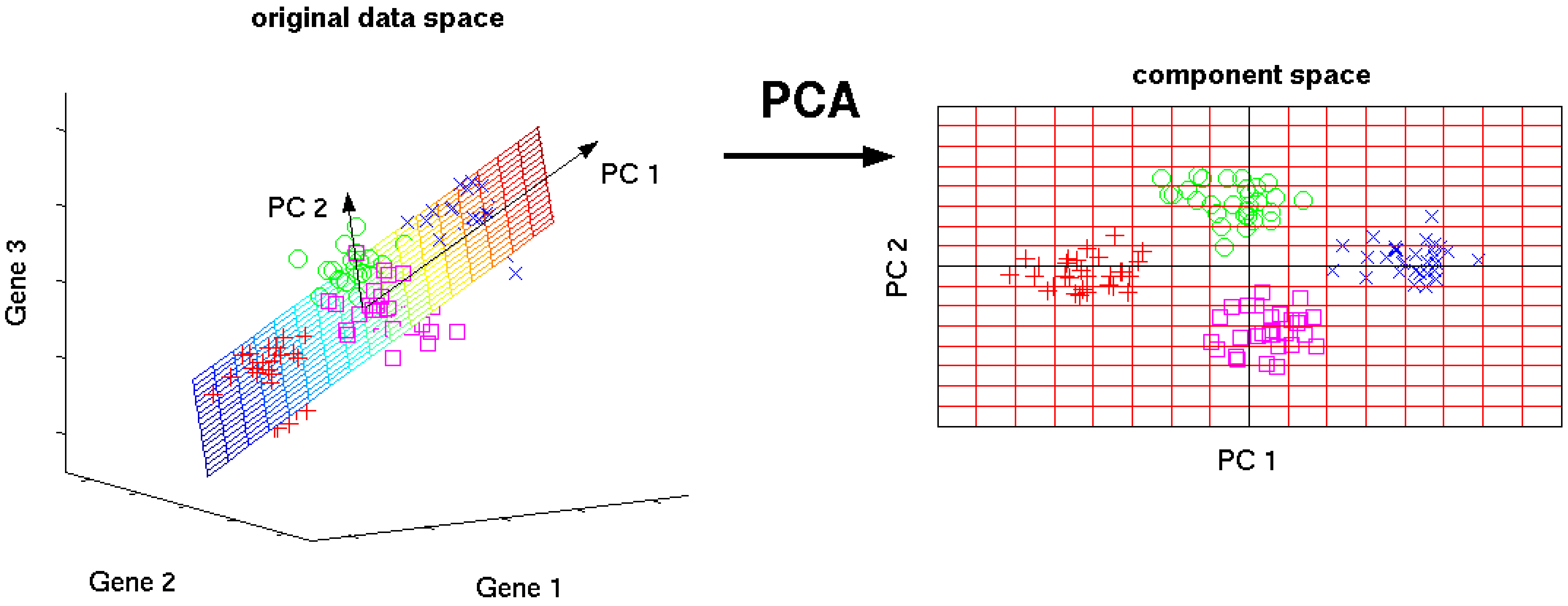
Image #1 (high bias) Image #2 (ideal) Image #3 (high variance)

Debugging

- High Bias
  1. Add more features/polynomial features
  2. Decrease λ

- High Variance
  1. Get more training data
  2. Choose smaller features
  3. Increase λ
  4. Aggregation and K-fold Cross Validation

Principal Component Analysis

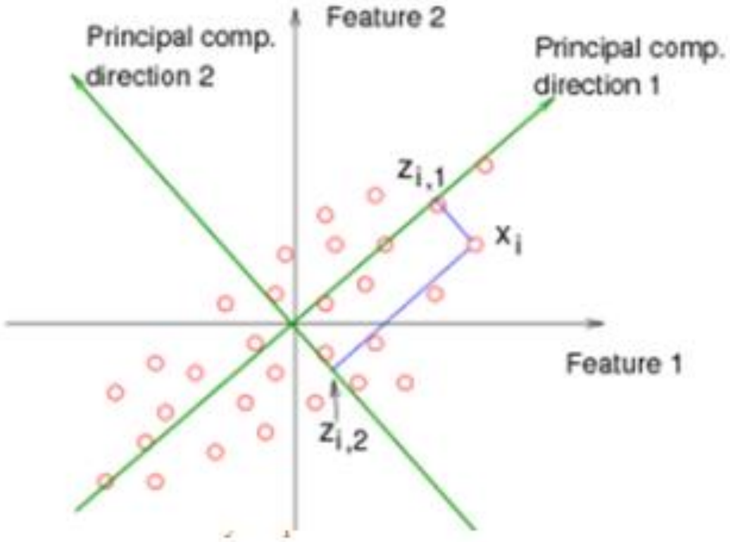


Principal Component Analysis contd.

- Reduces dimensions while returning info about original data
  - Change of basis
  - Best 'subspace' that captures as much data variance as possible
- Based on eigen-value decomposition of covariance matrix

$$\Sigma = \frac{1}{m}\sum_{i=1}^{n}(x^{(i)})(x^{(i)})^T$$

```
[U,D,V] = svd (Sigma)
numpy.linalg.svd
```



- $x(i) \in \mathbb{R}^n \to z(i) \in \mathbb{R}^k$

Do's and Don'ts of PCA

- Do's
  1. Data Compression (k chosen to keep 99% variance)
  2. Speed Up (chosen to keep 99% variance)
  3. Visualization (k=2 or 3)
  4. Feature selection with functional PCA
- Don'ts
  1. Use to fix overfitting
  2. Jump straight to PCA

$$\frac{\frac{1}{m}\sum_{i=1}^{m}||x^{(i)} - x_{approx}^{(i)}||^2}{\frac{1}{m}\sum_{i=1}^{m}||x^{(i)}||^2} \leq 0.01$$