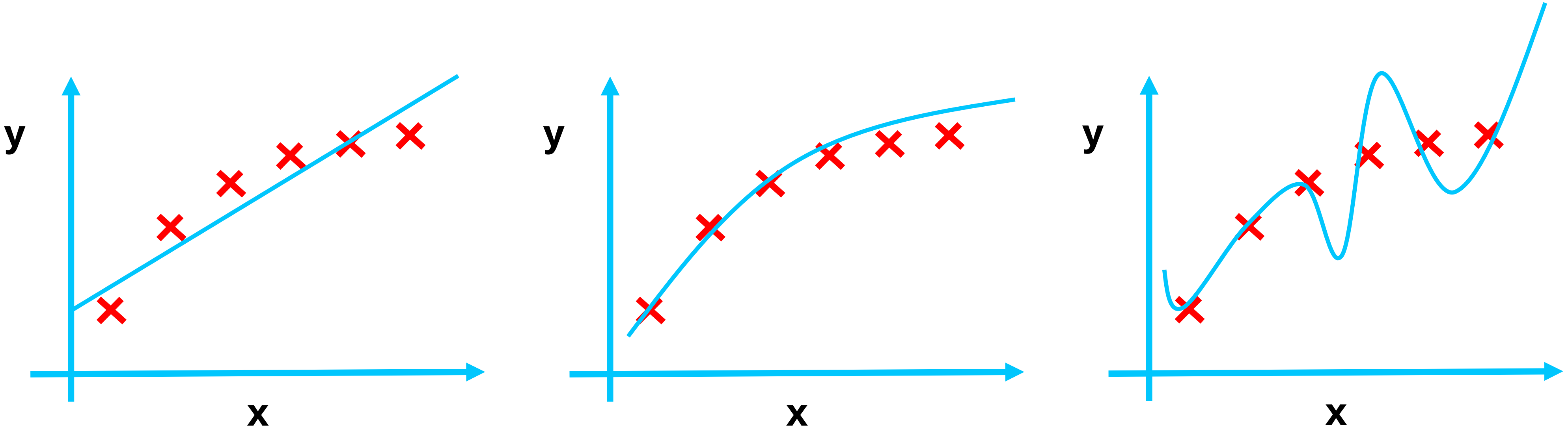
Machine Learning

Regularization

Pramote Kuacharoen

Linear Regression Overfitting

- If we have too many features, the learned hypothesis may fit the training set very well, but fails to generalize to new data



$\theta_0 + \theta_1 x$

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

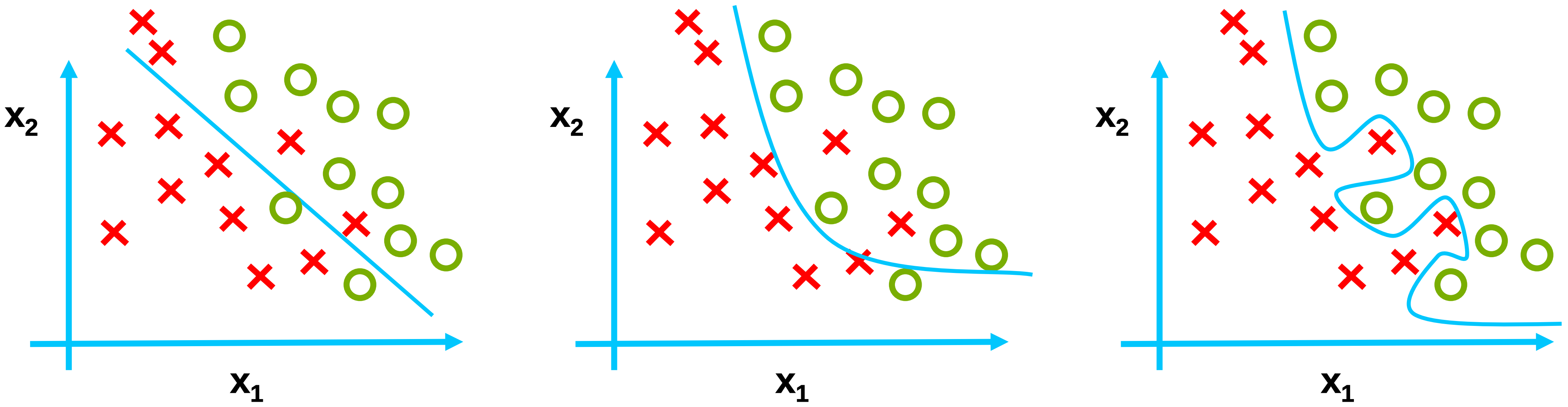
$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

Underfit or High bias

Overfit or High Variance

Logistic Regression Overfitting

- If we have too many features, the learned hypothesis may fit the training set very well, but fails to generalize to new data



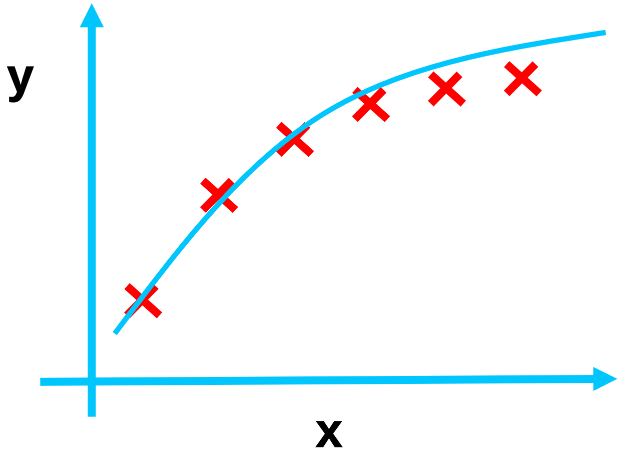
Underfit or High bias

Overfit or High Variance

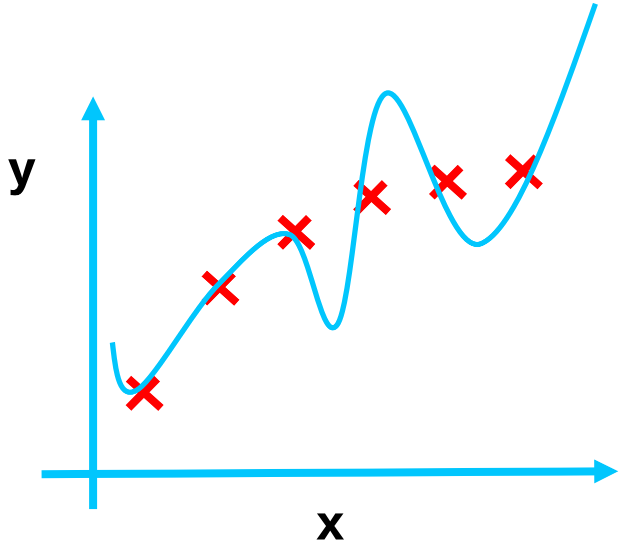
Mitigating Overfitting

- Reduce number of features
  - Manually select which features to eliminate
  - Model selection algorithm
- Regularization
  - Keep all features, but reduce magnitudes/values of parameters θ_j
  - Works well when we have many features, each of which contributes a little bit to predicting y

Intuition



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



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

Suppose we make θ_3 and θ_4 very small, the hypothesis is less influenced by the higher order terms

$$\min_{\theta} \frac{1}{2m} \sum_{i=1}^{m} \left(h\left(x^{(i)}\right) - y^{(i)}\right)^2 + 1000\theta_3 + 1000\theta_4$$

Regularization

- Small values of parameters $\theta_0, \theta_1, \dots, \theta_n$
  - Simpler hypothesis
  - Less prone to overfitting

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

$$\min_{\theta} J(\theta)$$

Regularized Linear Regression

- What if λ is set to an extremely large value
  - Algorithm works fine
  - Algorithm fails to eliminate overfitting
  - Algorithm results in underfitting
  - Gradient descent fails to converge

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

Gradient Descent

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

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

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

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

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

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m}\left[\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x_j^{(i)} + \lambda\theta_j\right]$$

Repeat:

$$\theta_0 := \theta_j - \alpha\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x_0^{(i)}$$

$$\theta_j := \theta_j\left(1 - \alpha\frac{\lambda}{m}\right) - \alpha\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x_j^{(i)} \qquad \text{(j=1, 2, …, n)}$$

$$1 - \alpha\frac{\lambda}{m} < 1$$

Regularized Logistic Regression

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

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

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

$$\theta_j := \theta_j - \alpha\frac{\partial}{\partial \theta_j}J(\theta)$$

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

Repeat:

$$\theta_0 := \theta_0 - \alpha\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x_0^{(i)}$$

$$\theta_j := \theta_j(1 - \alpha\frac{\lambda}{m}) - \alpha\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x_j^{(i)}$$ **(j=1, 2, …, n)**

L1 vs L2 Regularization

Linear Regularization

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

Logistic Regularization

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

L1 Regularization	L2 Regularization
Computational inefficient on non-sparse cases	Computational efficient due to having analytical solutions
Sparse outputs	Non-sparse outputs
Built-in feature selection	No feature selection

Question?