LOSS FUNCTION UNDER THE HOOD

Presented by: Shuyu Luo

OUTLINE

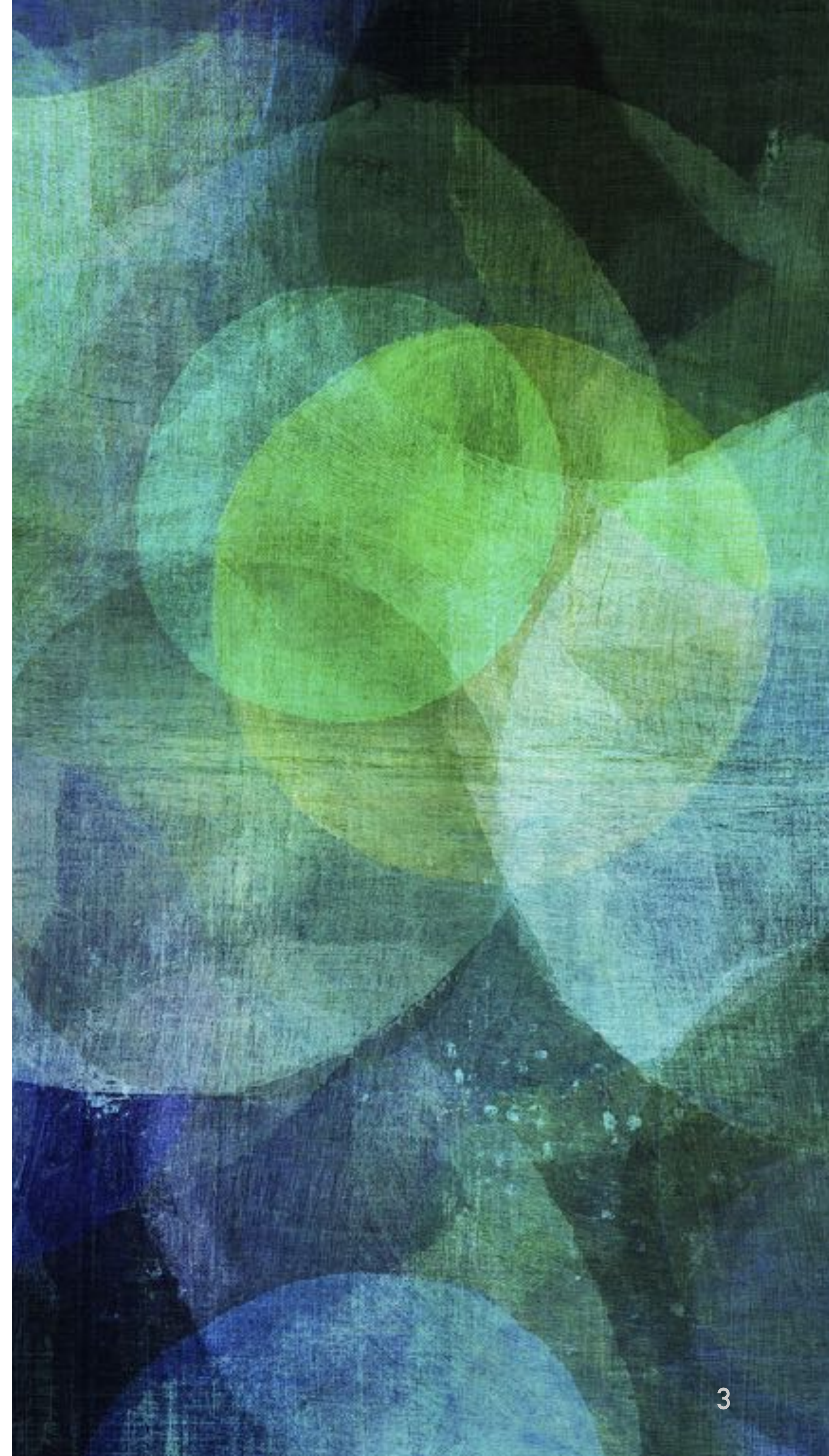
Three commonly used supervised learning model:

- Linear Regression
- Logistic Regression
- Support Vector Machine

For each of above model:

- Hypothesis
- Cost Function
- Optimization

LINEAR REGRESSION

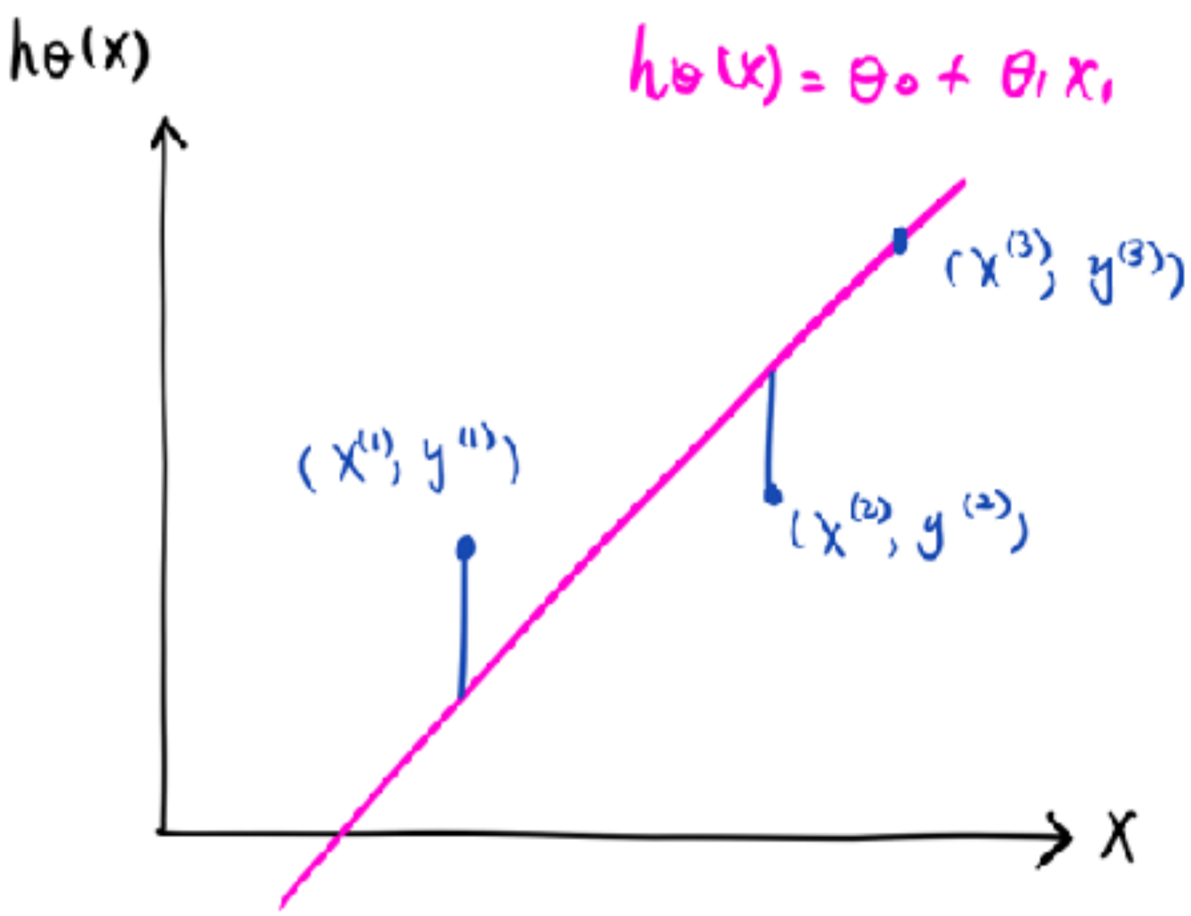


LINEAR REGRESSION

- Hypothesis

$$h_\theta(x^{(i)}) = \theta_0 x_0 + \theta_1 x_1 + \ldots \theta_n x_n = \begin{bmatrix} \theta_0 & \theta_1 & \ldots & \theta_n \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \ldots \\ x_n \end{bmatrix} = \theta^T X^{(i)}$$

- Loss Function



Least Squared Error:

$$\frac{1}{2}(h_\theta^{(i)} - y^{(i)})^2$$

Mean Squared Error:

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

LINEAR REGRESSION

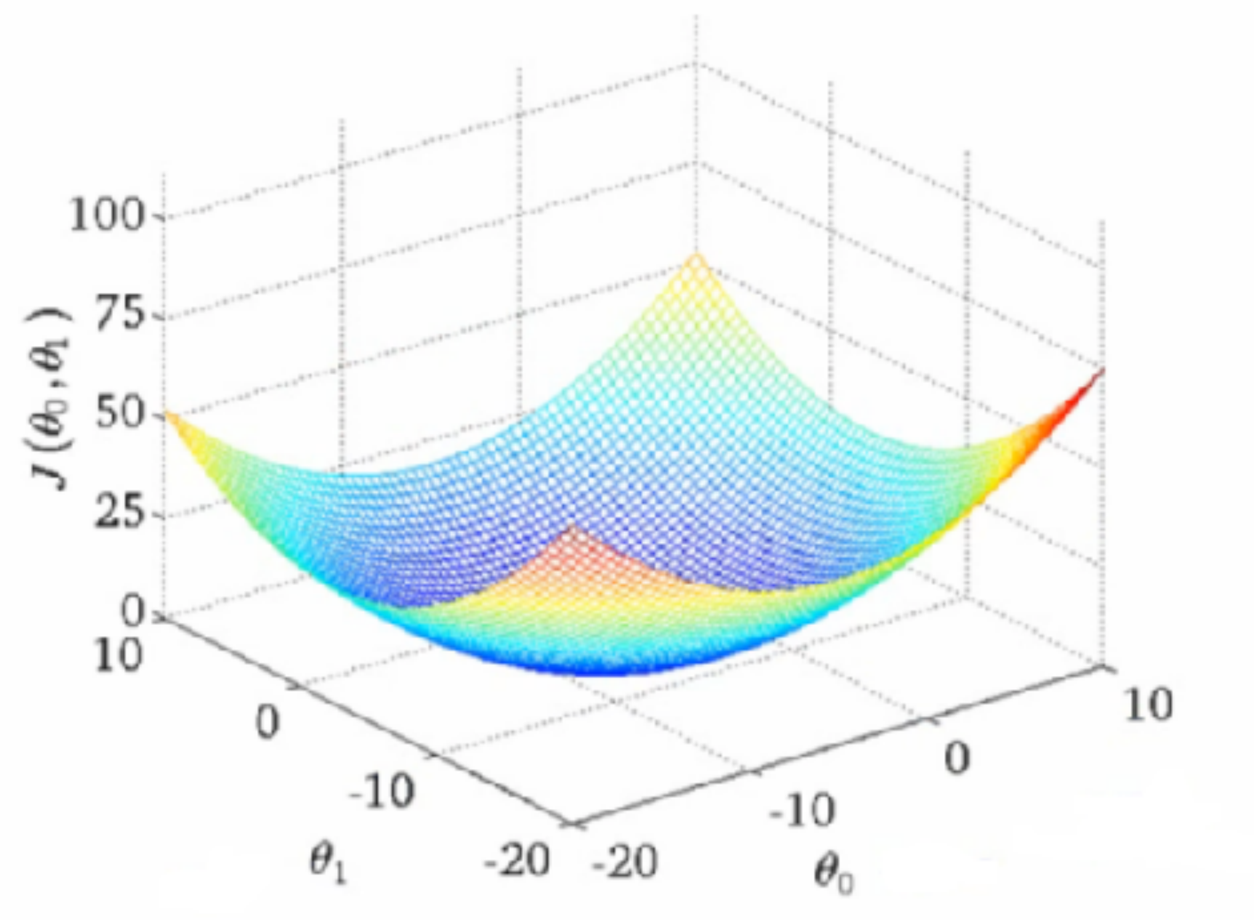
- Optimization
  - Goal:

Find the optimal set of coefficients who can minimize cost function.

  - Approach:

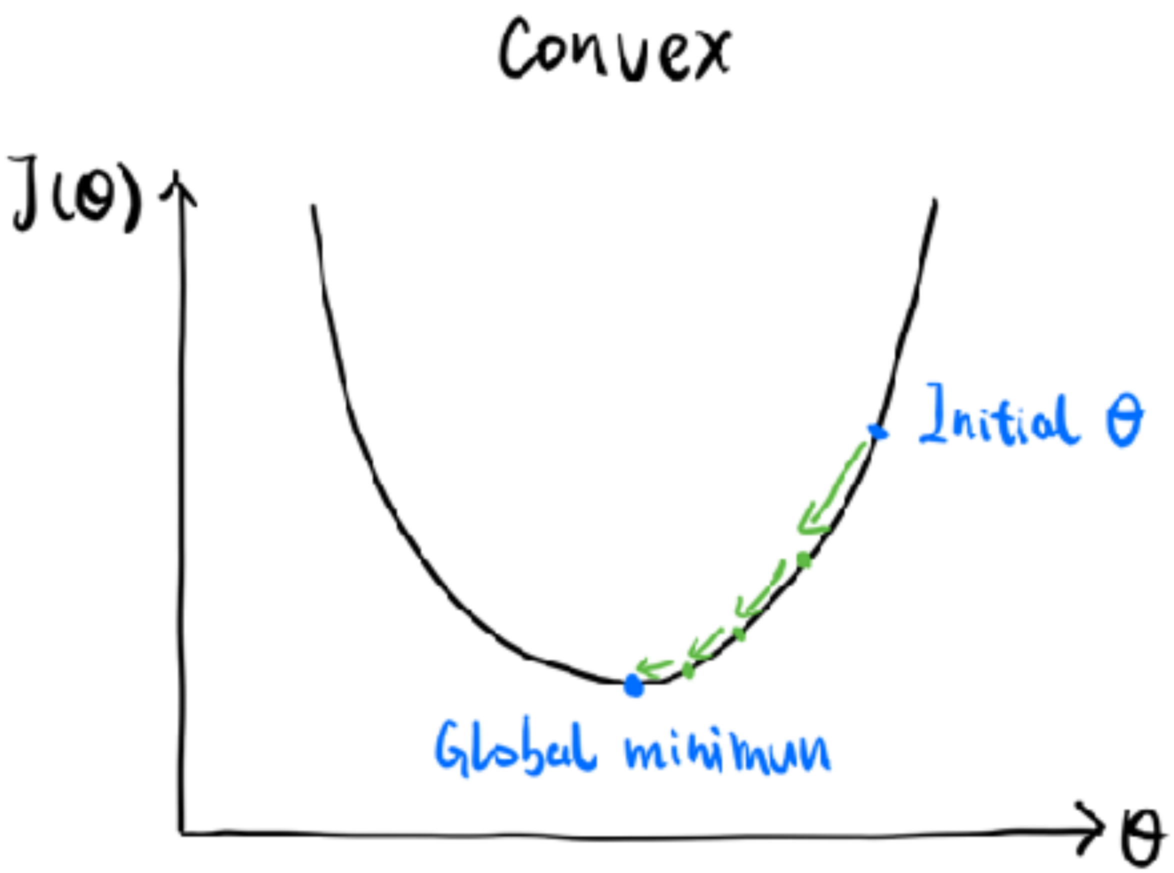
Analytical(Normal Equation)

Programming(Gradient Descent)



LINEAR REGRESSION

- Convex vs. Non-Convex



$$Find\ coefficient\ subjects\ to\ \frac{\partial J(\theta)}{\partial \theta_j} = 0, \quad J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

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

LINEAR REGRESSION

- Gradient Descent:

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

$$\theta_1 = \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_1^{(i)}$$

$$\cdots\cdots$$

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

- Learning Rate α:

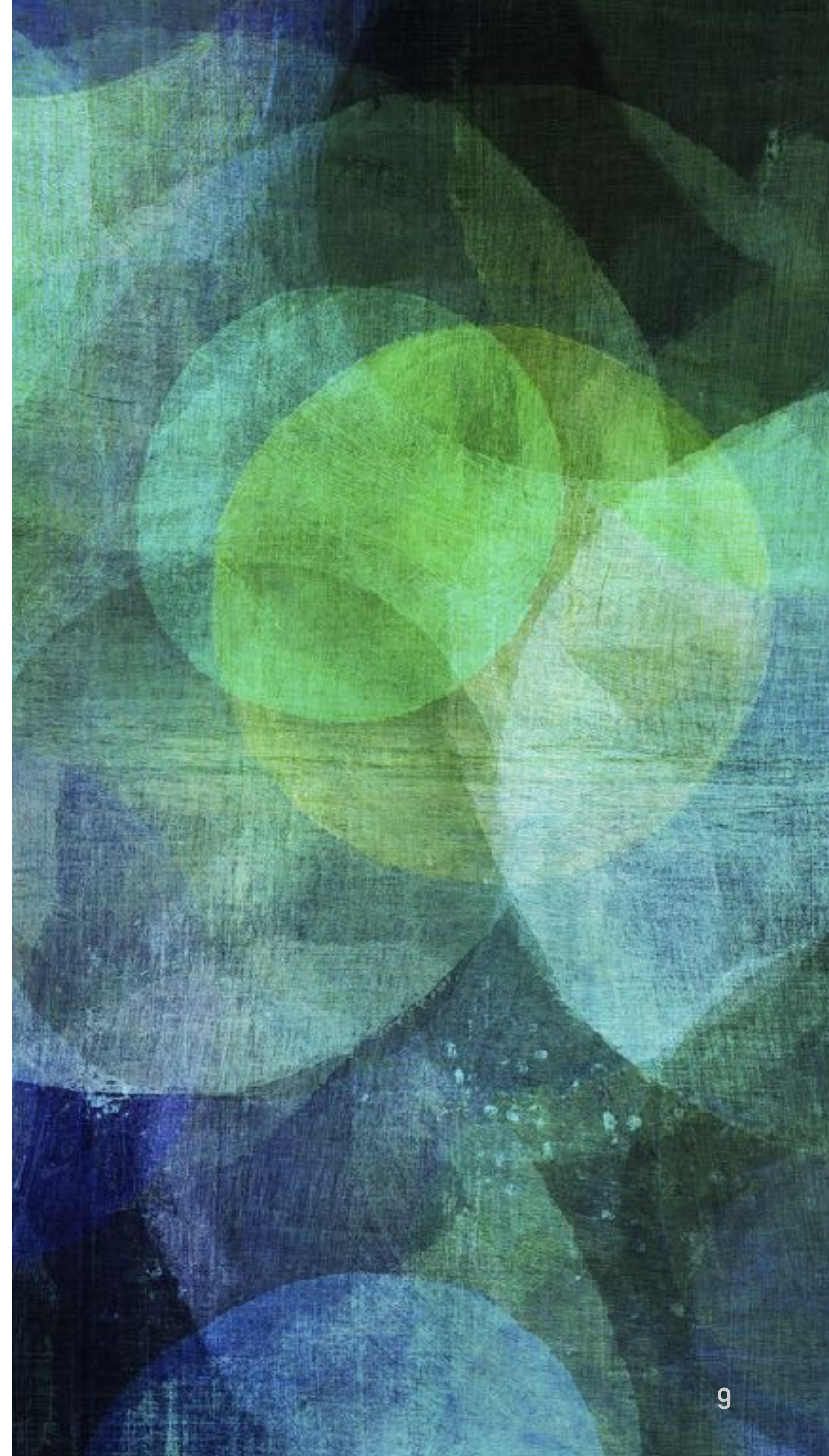
Large α → Converge quickly. Less training time required. Cost function may not minimized enough.

Small α → Converge slowly. More training time required. Cost function minimized better.

LINEAR REGRESSION

- Example (Jupyter Notebook)
  - Boston Housing Prediction Dataset
  - Gradient Descent
  - Vectorized Implementation

LOGISTIC REGRESSION



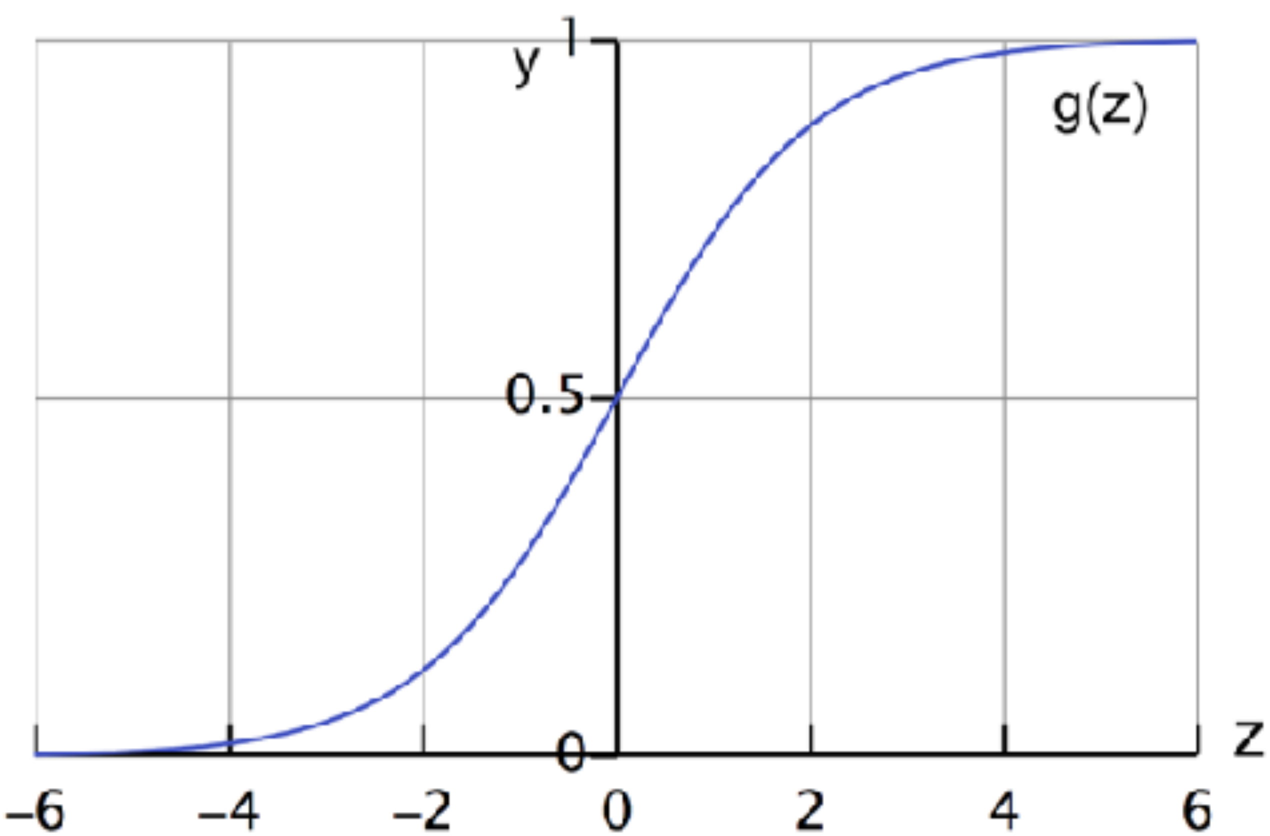
LOGISTIC REGRESSION

- Linear Regression Hypothesis:

$$\theta_0 x_0 + \theta_1 x_1 + \ldots \theta_n x_n = \theta^T X^{(i)}$$

$\longrightarrow$ Raw Model Output

- Hypothesis



Sigmoid Function:

$$g(z) = \frac{1}{1 + e^{(-z)}}$$

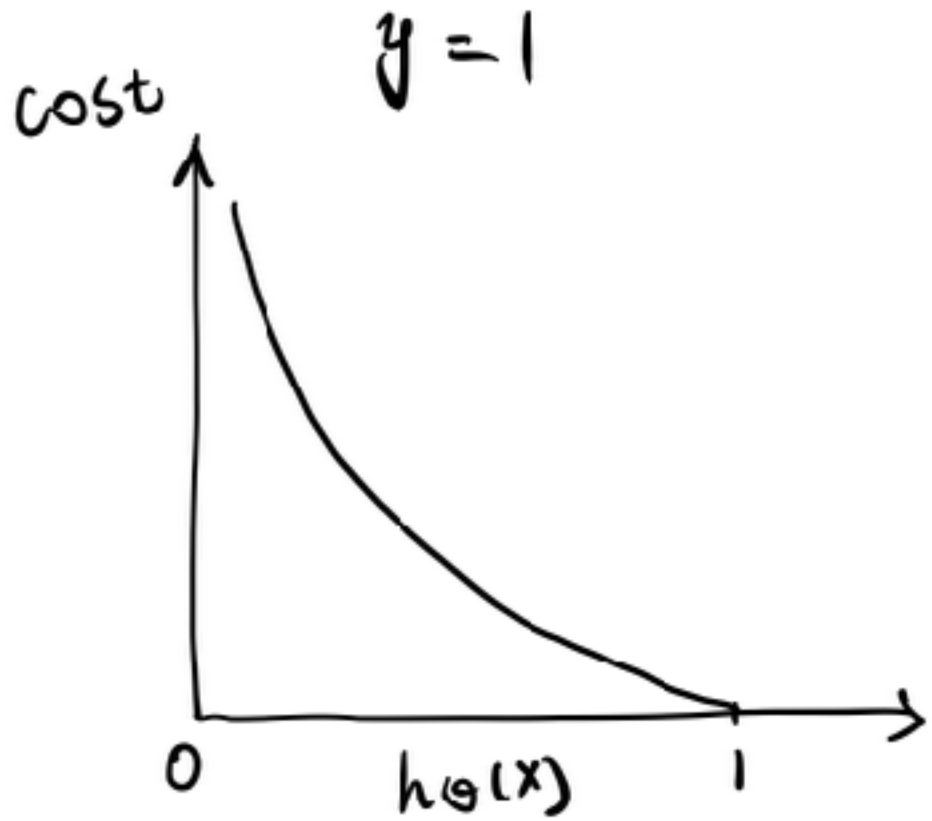
Hypothesis:

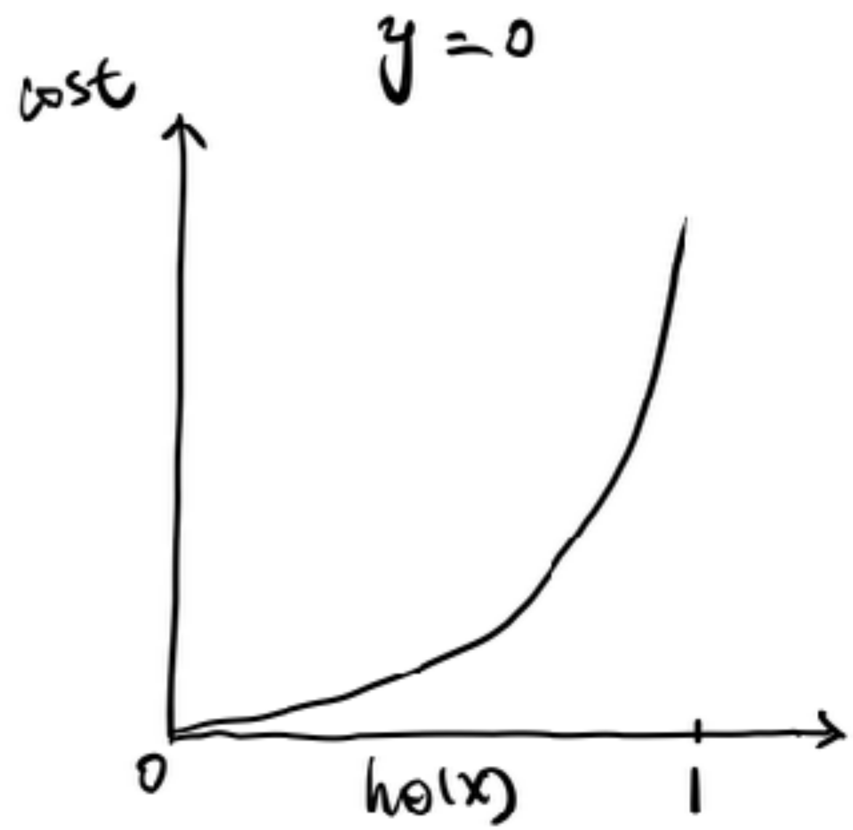
$$h_\theta(x^{(i)}) = \frac{1}{1 + e^{(-\theta^T X^{(i)})}}$$

$$P(y^{(i)} = 1 \mid X^{(i)}; \theta)$$

LOGISTIC REGRESSION

- Loss Function





Logistic Loss:

$$Cost(h_\theta(x^{(i)}), y^{(i)}) = \begin{cases} -log(h_\theta(x^{(i)})) & if\ y^{(i)} = 1 \\ -log(1 - h_\theta(x^{(i)})) & if\ y^{(i)} = 0 \end{cases}$$

$$Cost(h_\theta(x^{(i)}), y^{(i)}) = -y^{(i)}log(h_\theta(x^{(i)})) - (1 - y^{(i)})log(1 - h_\theta(x^{(i)}))$$

LOGISTIC REGRESSION

- Cost Function

$$J(\theta) = \frac{1}{m}[\sum_{i=1}^{m} y^{(i)}log(h_\theta(x^{(i)})) + (1 - y^{(i)})log(1 - h_\theta(x^{(i)}))]$$

- Regularized Cost Function
  - L1 (Lasso) Regularization

    Prevent overfitting; Mitigate multicollinearity; Feature selection

  - L2 (Ridge) Regularization

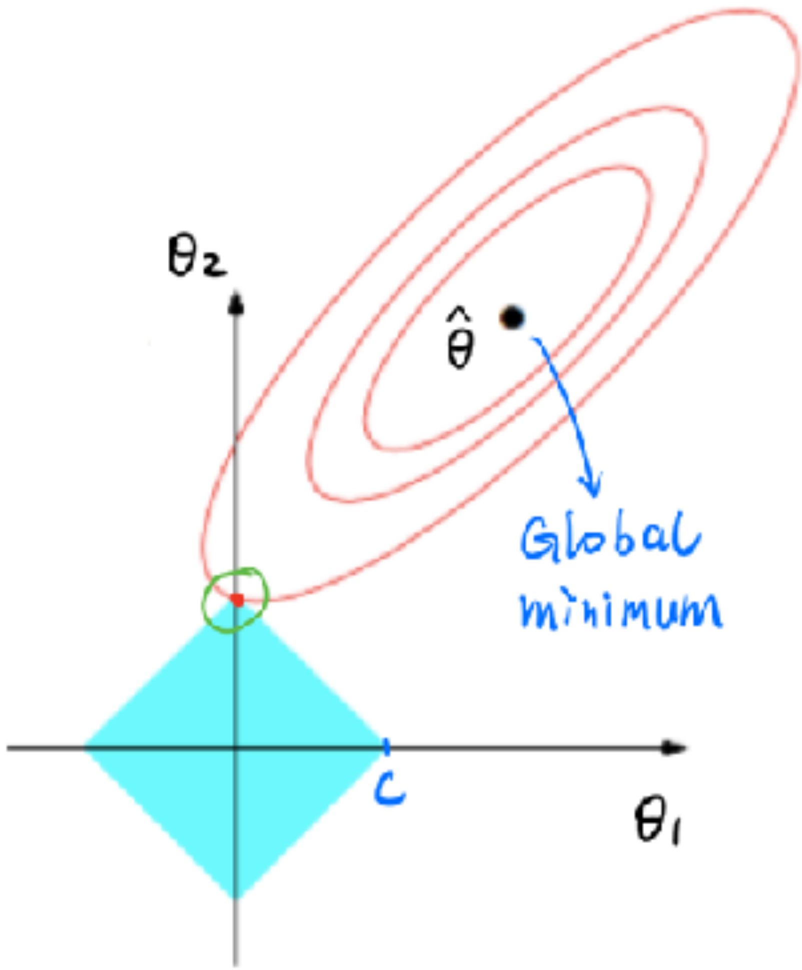
    Prevent overfitting; Mitigate multicollinearity

LOGISTIC REGRESSION

- Regularization

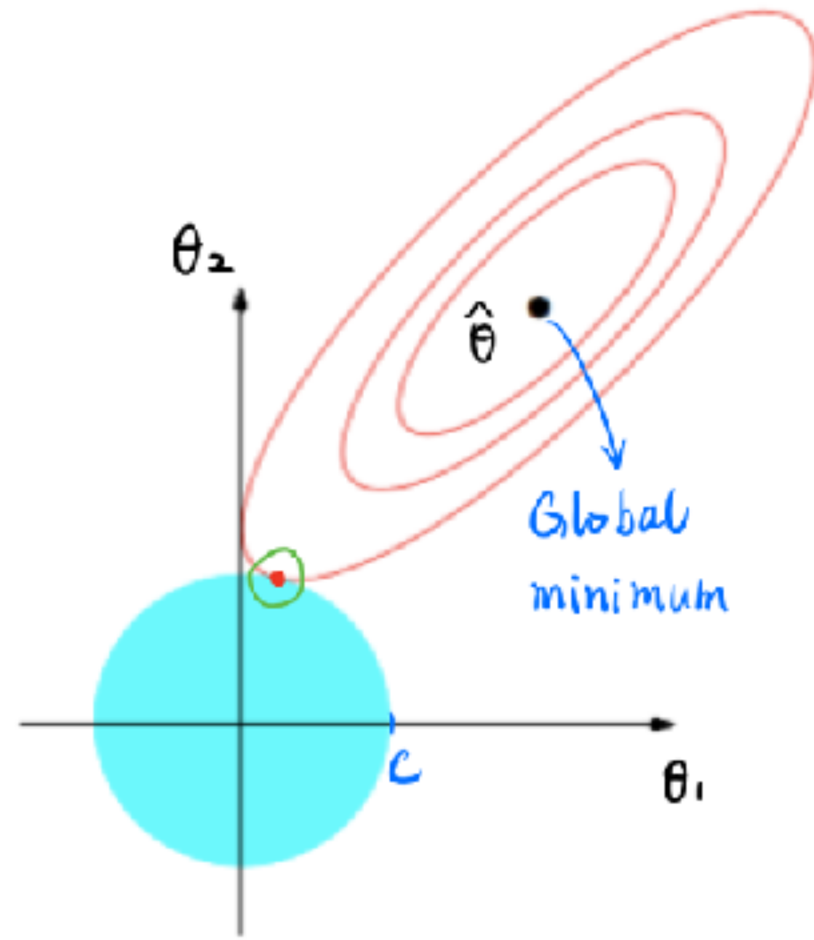
L1

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} Cost(h_\theta(x^{(i)}), y^{(i)}) \quad s.t. \quad ||\theta|| <= C$$



L2

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} Cost(h_\theta(x^{(i)}), y^{(i)}) \quad s.t. \quad ||\theta||^2 <= C^2$$



LOGISTIC REGRESSION

➤ Regularization

L1

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} Cost(h_\theta(x^{(i)}), y^{(i)}) + \frac{\lambda}{m}\sum_{j=1}^{n} |\theta_j|$$

L2

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

➤ Regularized Parameter ƛ

- Plays a role similar to 1/C

- Large ƛ → more weight on regularization, smaller coefficients, may lead to underfitting

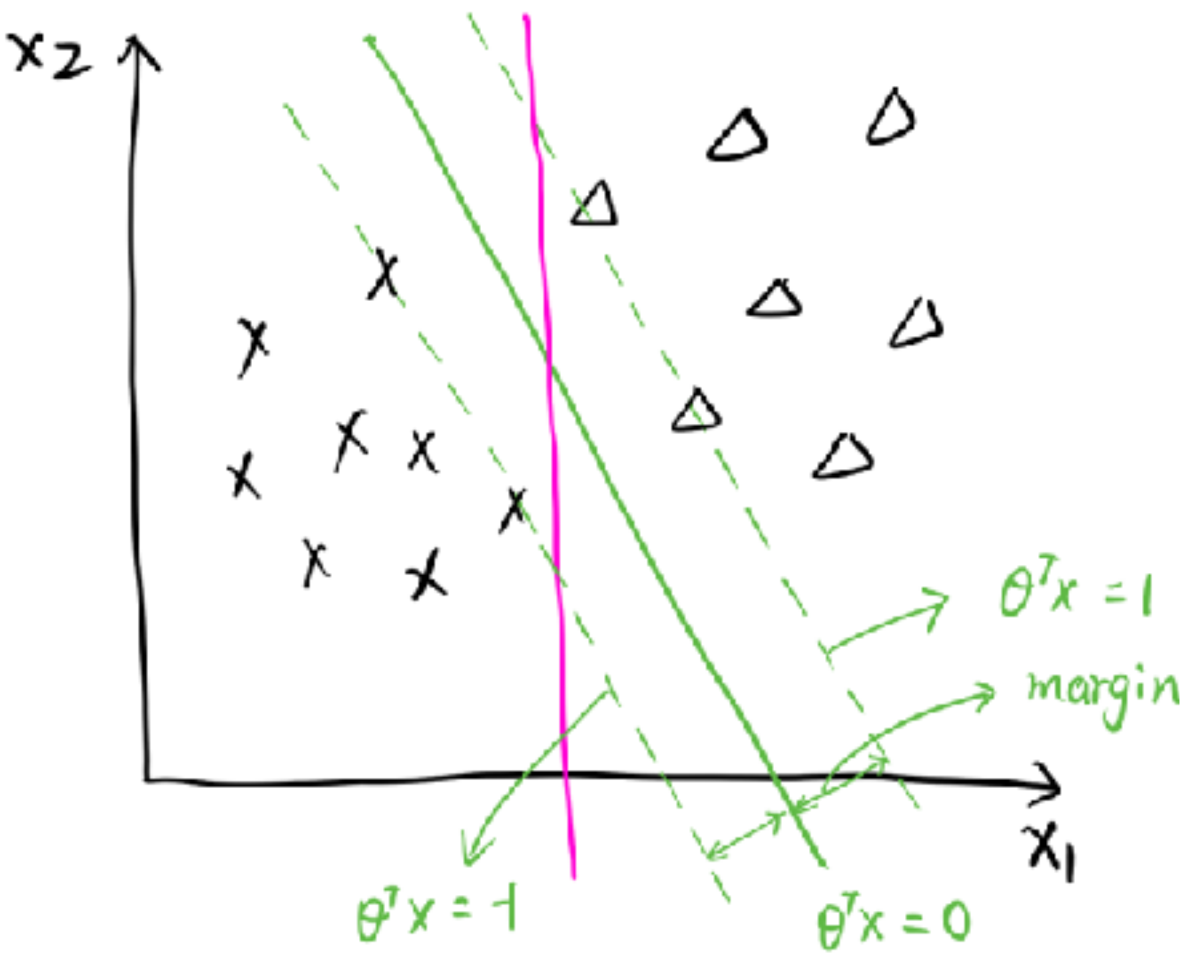
- Small ƛ → more weight on 'fit', larger coefficients, may lead to overfitting

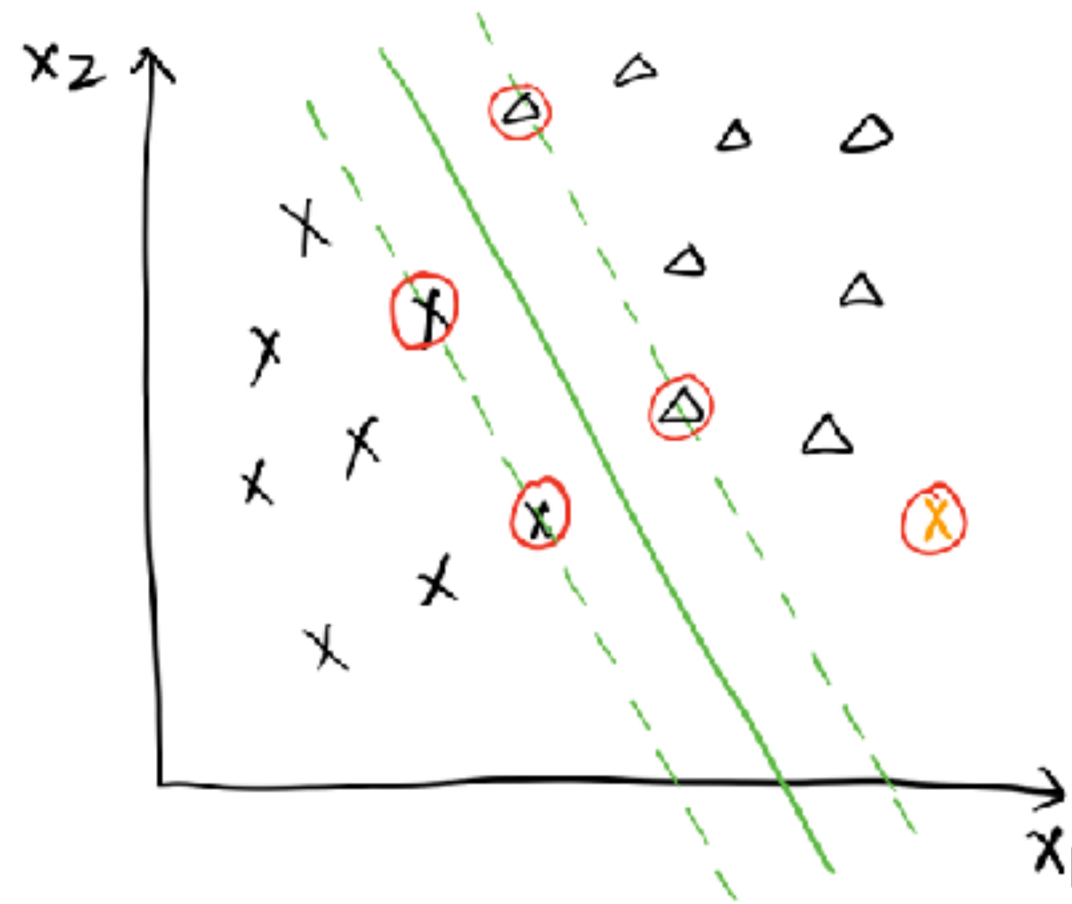
SUPPORT VECTOR MACHINE

LINEAR SVM

- Raw Model Output:

$$\theta_0 x_0 + \theta_1 x_1 + \ldots \theta_n x_n = \theta^T X^{(i)}$$



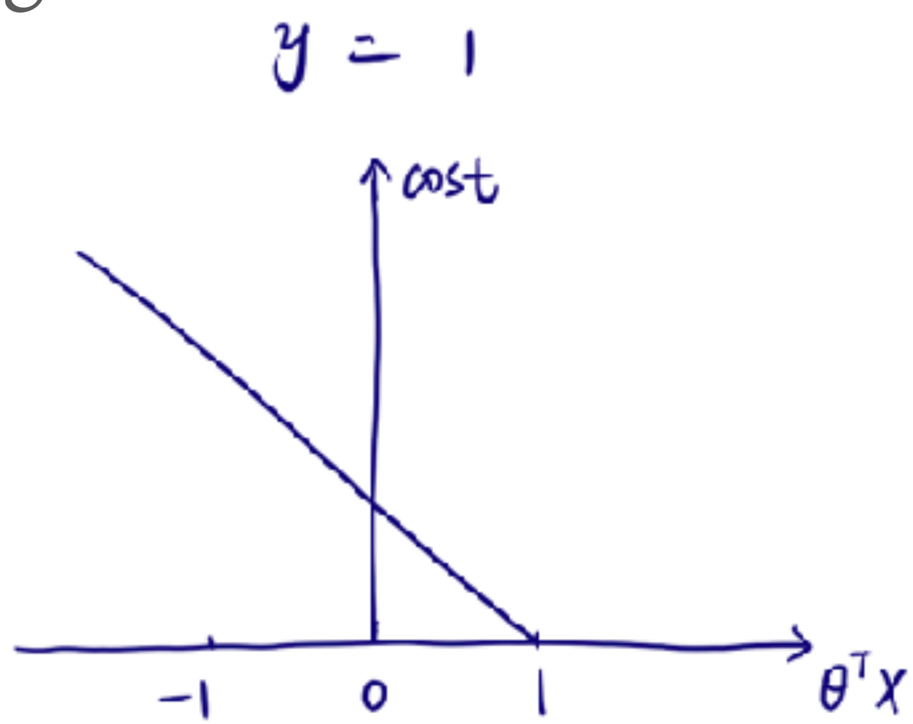


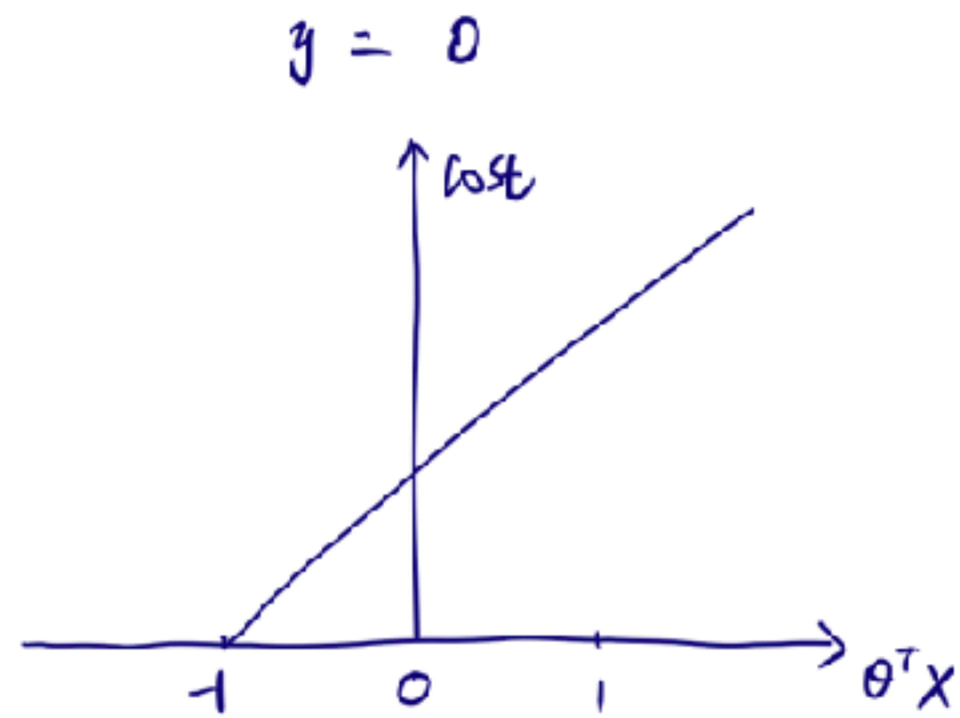
LINEAR SVM

- Hypothesis

$$h_\theta(x^{(i)}) = \begin{cases} 1 & if\ \theta^T X^{(i)} >= 0 \\ 0 & otherwise \end{cases}$$

- Hinge Loss

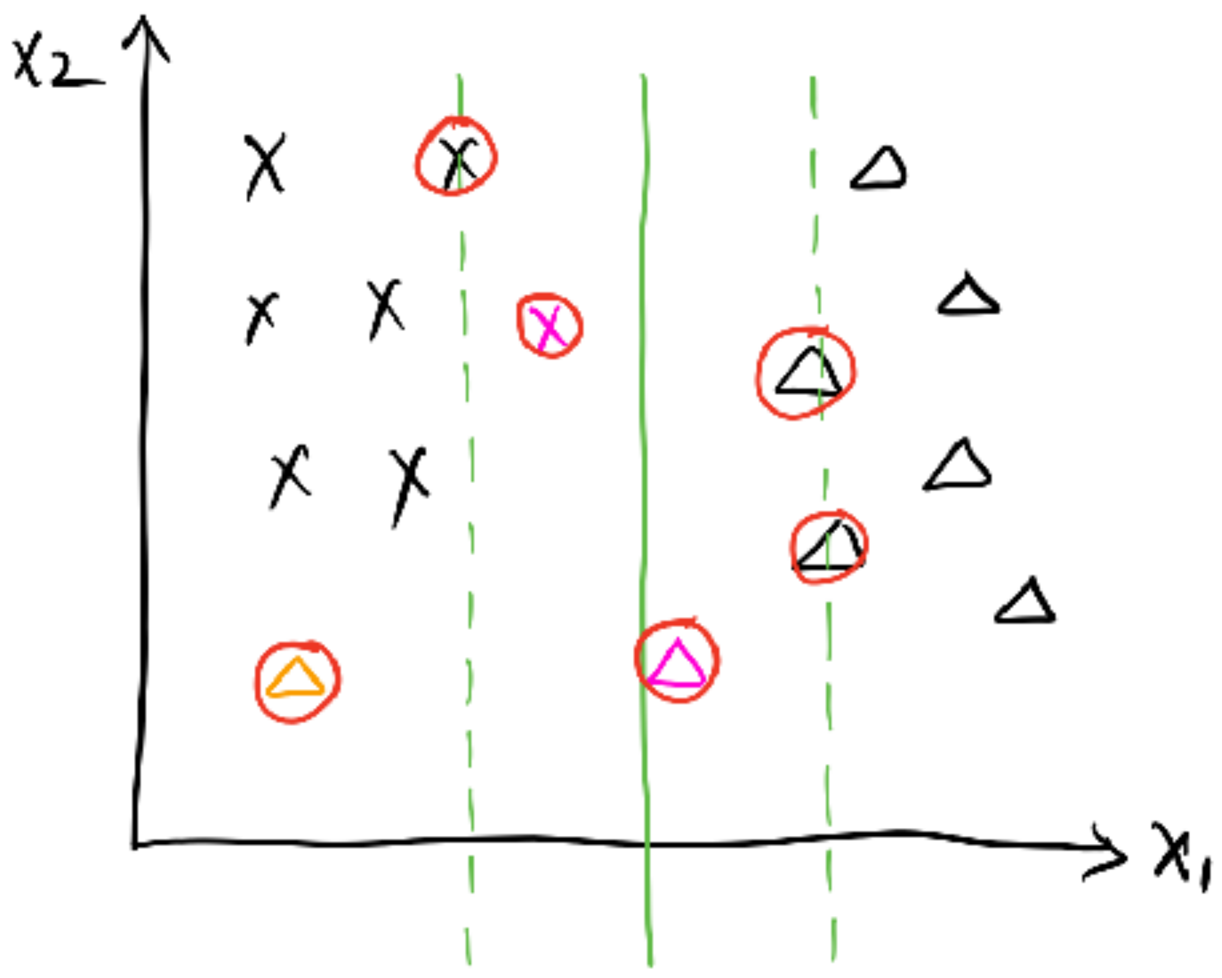


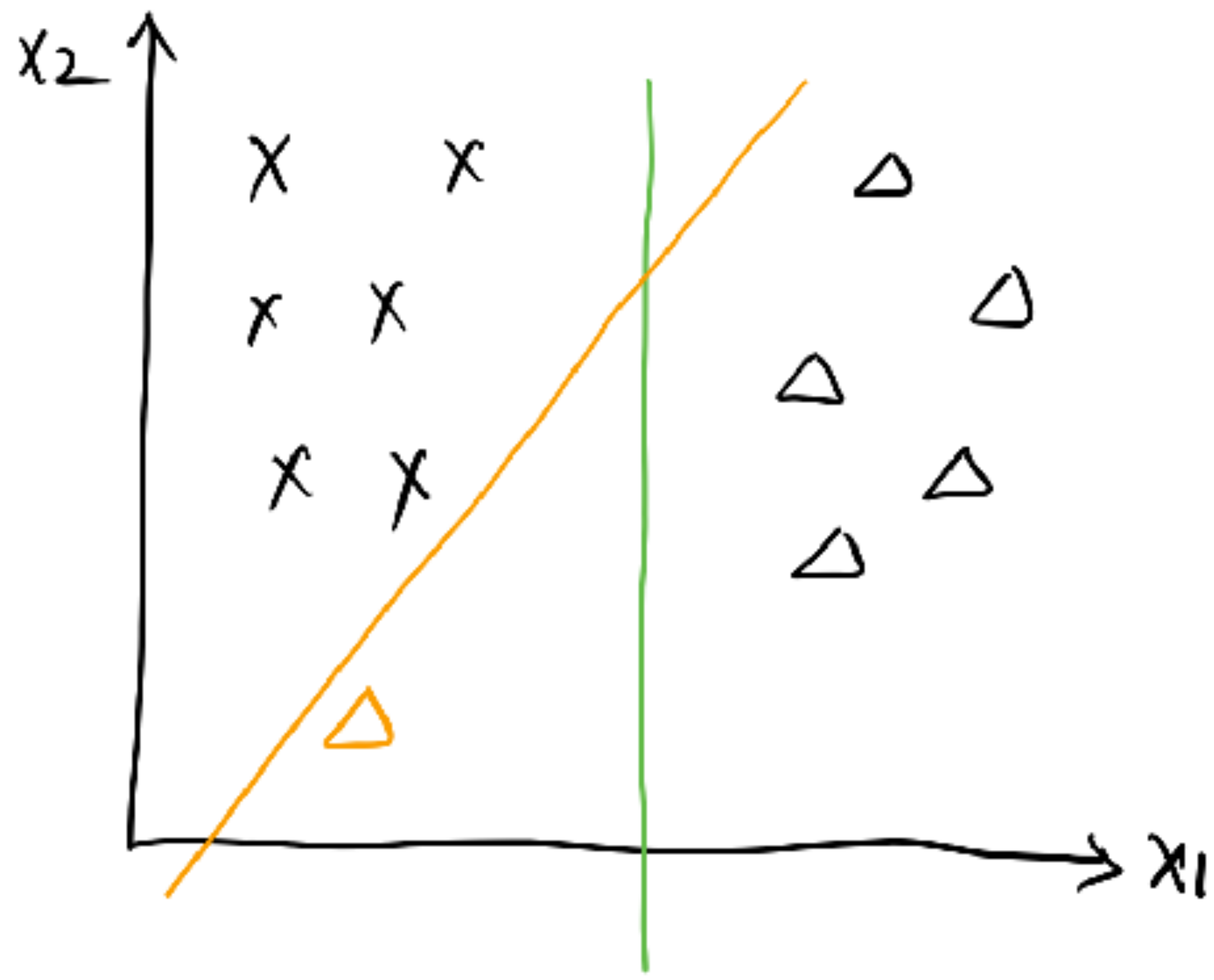


$$Cost(h_\theta(x^{(i)}), y^{(i)}) = \begin{cases} max(0,1 - \theta^T X^{(i)}) & if\ y^{(i)} = 1 \\ max(0,1 + \theta^T X^{(i)}) & if\ y^{(i)} = 0 \end{cases}$$

LINEAR SVM

- Regularization





LINEAR SVM

- Regularized Cost Function

L1

$$J(\theta) = C[\sum_{i=1}^{m} y^{(i)}Cost_1(\theta^T(x^{(i)}) + (1 - y^{(i)})Cost_0(\theta^T(x^{(i)})] + \frac{1}{2}\sum_{j=1}^{n} |\theta_j|$$

L2

$$J(\theta) = C[\sum_{i=1}^{m} y^{(i)}Cost_1(\theta^T(x^{(i)}) + (1 - y^{(i)})Cost_0(\theta^T(x^{(i)})] + \frac{1}{2}\sum_{j=1}^{n} \theta_j^2$$

- Regularized Parameter C

- Plays a role similar to $1/\lambda$

- Small C → more weight on regularization, smaller coefficients, may lead to underfitting

- Large C → more weight on ‘fit’, larger coefficients, may lead to overfitting

SVM WITH KERNEL

- Hypothesis

$$h_\theta(x^{(i)}) = \begin{cases} 1 & if\ \theta^T f^{(i)} >= 0 \\ 0 & otherwise \end{cases}$$

- Loss Function

$$Cost(h_\theta(x^{(i)}), y) = \begin{cases} max(0{,}1 - \theta^T f^{(i)}) & if\ y^{(i)} = 1 \\ max(0{,}1 + \theta^T f^{(i)}) & if\ y^{(i)} = 0 \end{cases}$$

- Cost Function

$$J(\theta) = C[\sum_{i=1}^{m} y^{(i)} Cost_1(\theta^T f^{(i)}) + (1 - y^{(i)}) Cost_0(\theta^T f^{(i)})] + Regularized\ Term$$

SVM WITH KERNEL

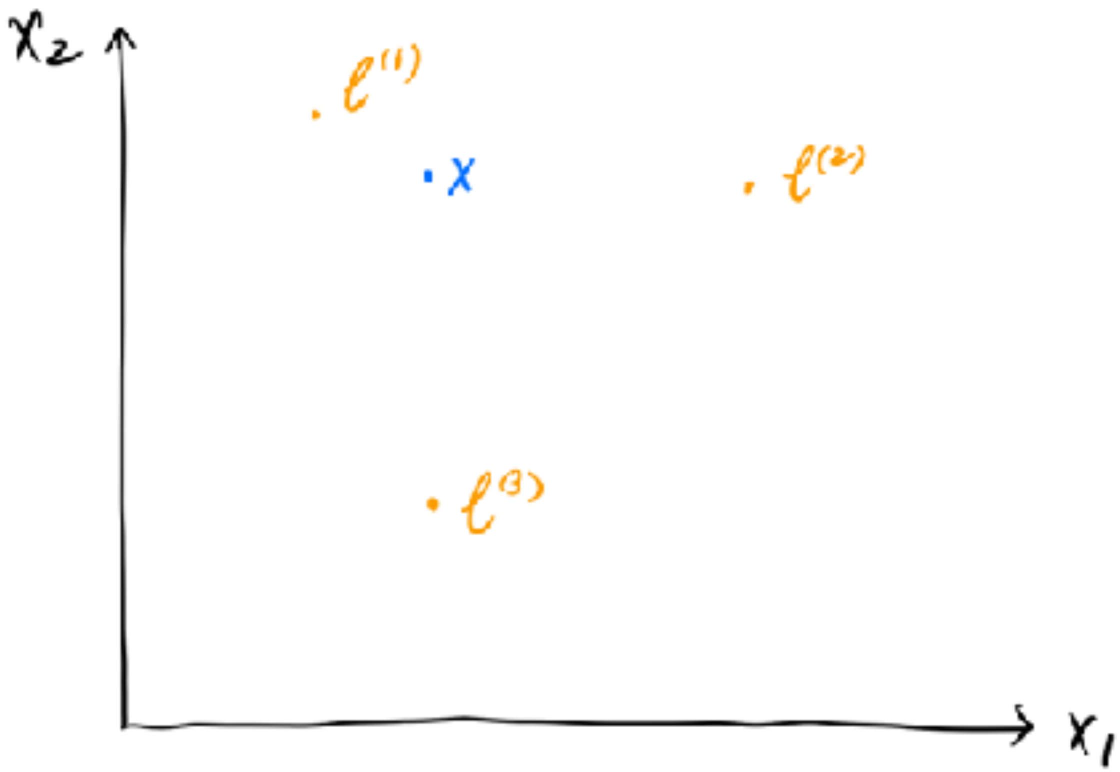
- Polynomial Regression

  Example: $y = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_1 x_2^2$

  $$f_0 = x_0, \quad f_1 = x_1, \quad f_2 = x_2, \quad f_3 = x_1^2, \quad f_4 = x_1 x_2^2$$

- Gaussian Kernel/Radial Basis Function

SVM WITH KERNEL



$$f_1 = Similarity(x, l^{(1)}) = exp(-\frac{||x - l^{(1)}||^2}{2\sigma^2})$$

$$f_2 = Similarity(x, l^{(2)}) = exp(-\frac{||x - l^{(2)}||^2}{2\sigma^2})$$

$$f_3 = Similarity(x, l^{(3)}) = exp(-\frac{||x - l^{(3)}||^2}{2\sigma^2})$$

$$Raw\ Model\ Output : \theta_0 f_0 + \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3, \quad f_0 = 1$$

SVM WITH KERNEL

- Recreate Features :

The number of features equals to the number of training samples.

Given the i^{th} *sample* $x^{(i)}$:

$$f_1^{(i)} = k(x^{(i)}, l^{(1)})$$

$$f_2^{(i)} = k(x^{(i)}, l^{(2)})$$

$$\cdots\cdots$$

$$f_i^{(i)} = k(x^{(i)}, l^{(i)})$$

$$\cdots\cdots$$

$$f_m^{(i)} = k(x^{(i)}, l^{(m)})$$

where $x^{(i)} = l^{(i)},\ f_i^{(i)} = 1$

Hypothesis:

$$h_\theta(x^{(i)}) = \begin{cases} 1 & if\ \theta^T f^{(i)} >= 0 \\ 0 & otherwise \end{cases}$$

THANK YOU!