Support Vector Machines

CSCI 447/547 MACHINE LEARNING

Outline

- Optimization Objective
- Large Margin Intuition
- Math behind Large Margin Classification
- Kernels
- Using an SVM

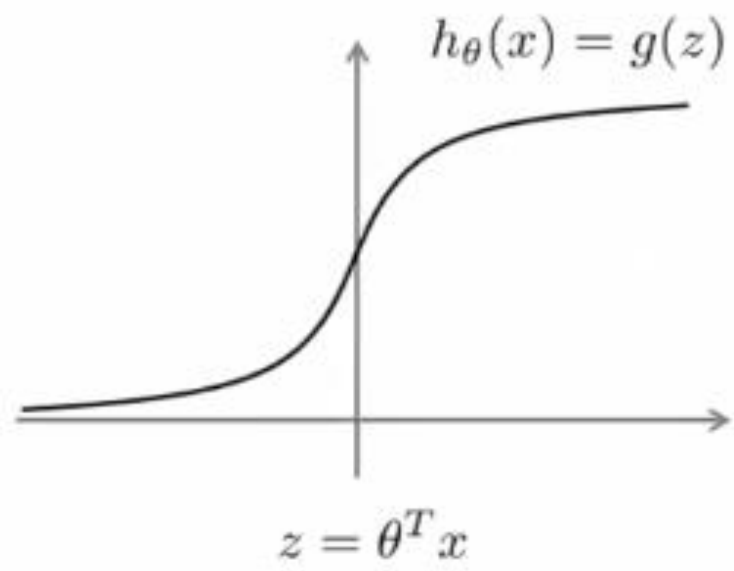
https://www.youtube.com/watch?v=doN5SexZjto

Optimization Objective: Comparison to Logistic Regression

If actual output $y = 1$, we want $h_\theta(x) \approx 1$, $\theta^T \gg 0$

If actual output $y = 0$, we want $h_\theta(x) \approx 0$, $\theta^T \ll 0$

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



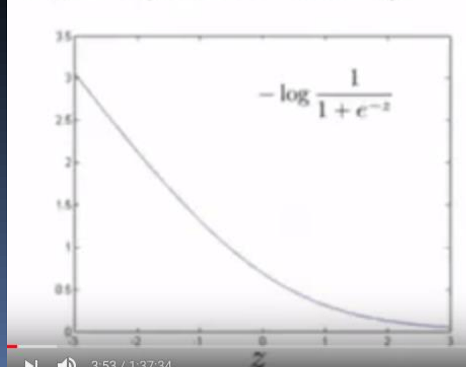
Optimization Objective: Comparison to Logistic Regression

Each example (x,y) contributes a term to the cost function:

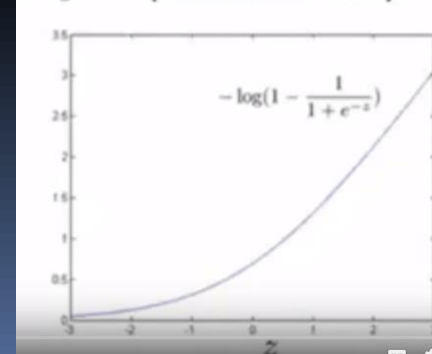
Cost of example: $-(y \log h_\theta(x) + (1-y)\log(1-h_\theta(x)))$

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

If $y=1$ (want $\theta^T x \gg 0$):



If $y=0$ (want $\theta^T x \ll 0$):



Optimization Objective: Comparison to Logistic Regression

Cost function $J(\theta)$:

Logistic regression:

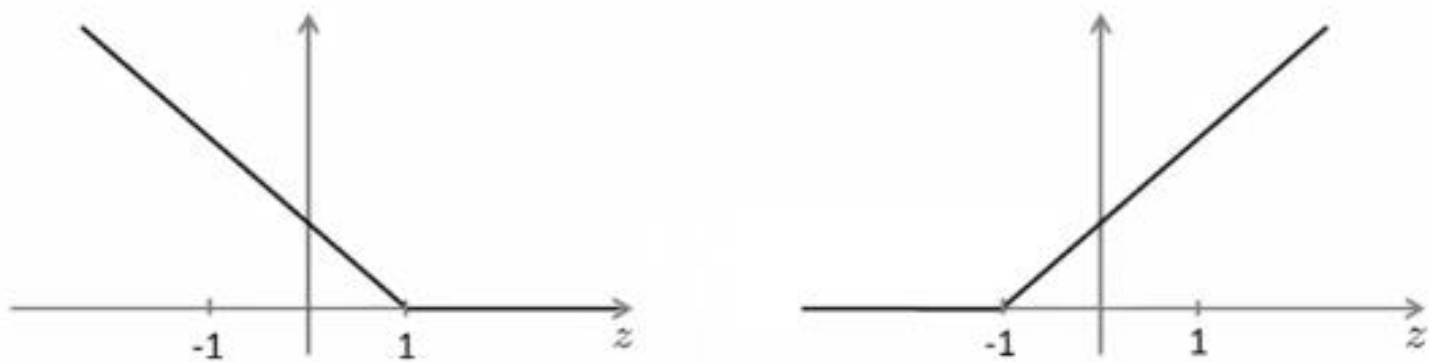
$$\min_\theta \frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)}\left(-\log h_\theta(x^{(i)})\right) + (1-y^{(i)})\left((-\log(1-h_\theta(x^{(i)}))\right)\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

SVM hypothesis

$$\min_\theta C\sum_{i=1}^{m}\left[y^{(i)}cost_1(\theta^T x^{(i)}) + (1-y^{(i)})cost_0(\theta^T x^{(i)})\right] + \frac{1}{2}\sum_{i=1}^{n}\theta_j^2$$

Large Margin Intuition

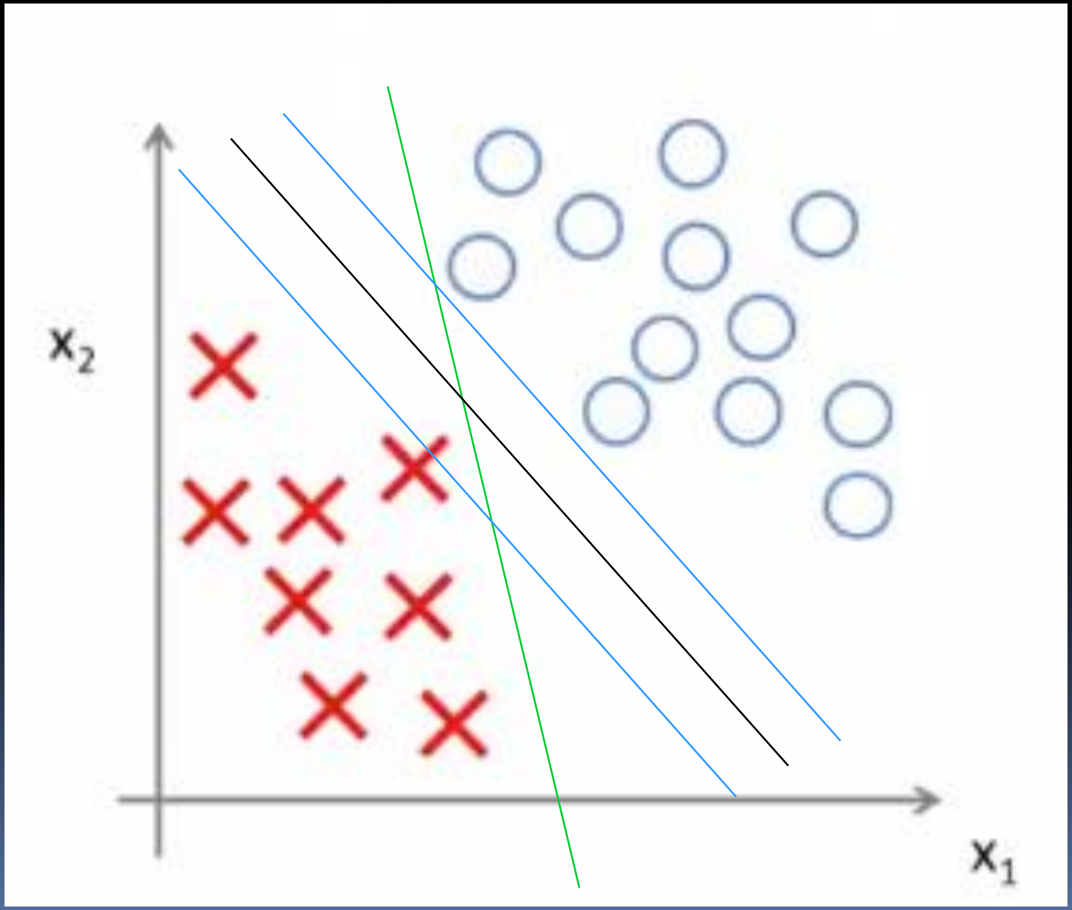
$$\min_{\theta} C \sum_{i=1}^{m} \left[y^{(i)} cost_1(\theta^T x^{(i)}) + (1 - y^{(i)}) cost_0(\theta^T x^{(i)}) \right] + \frac{1}{2} \sum_{i=1}^{n} \theta_j^2$$



If $y = 1$, we want $\theta^T x \geq 1$ (not just ≥ 0)

If $y = 0$, we want $\theta^T x \leq -1$ (not just < 0)

Large Margin Intuition: Linearly Separable Case



Large Margin Classifiers

- Large margin approach sensitive to outliers
- Support vector machines will “do the right thing” in the presence of outliers
  - If constant C is not given too much weight

Large Margin Classifier Math

SVM Decision Boundary

$$\min_{\theta} \frac{1}{2} \sum_{j=1}^{n} \theta_j^2$$

$$\text{s.t.} \quad \theta^T x^{(i)} \geq 1 \quad \text{if } y^{(i)} = 1$$

$$\theta^T x^{(i)} \leq -1 \quad \text{if } y^{(i)} = 0$$

Large Margin Classifier Math

SVM Decision Boundary

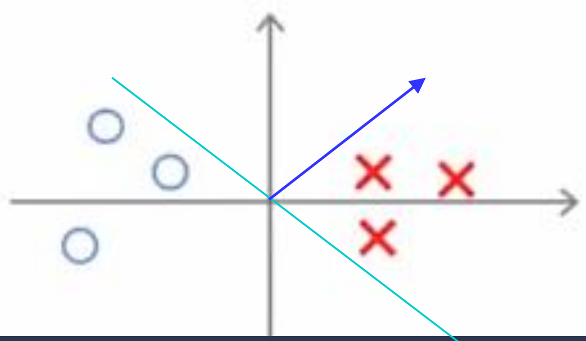
$$\min_{\theta} \frac{1}{2} \sum_{j=1}^{n} \theta_j^2$$

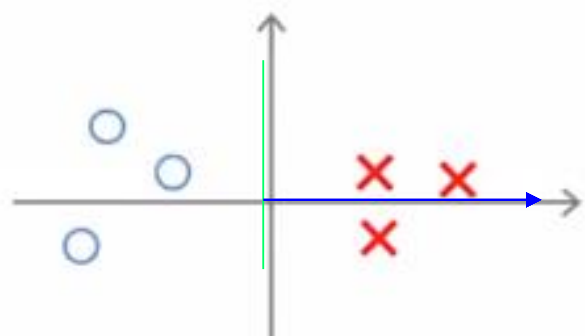
$$\text{s.t.} \quad p^{(i)} \cdot \|\theta\| \geq 1 \qquad \text{if } y^{(i)} = 1$$

$$\qquad p^{(i)} \cdot \|\theta\| \leq -1 \quad \text{if } y^{(i)} = 1$$

where $p^{(i)}$ is the projection of $x^{(i)}$ onto the vector θ.

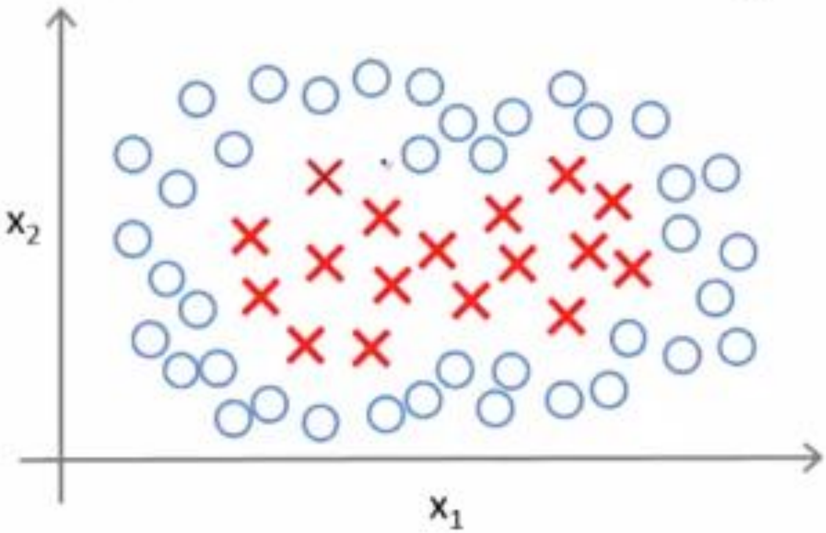
Simplification: $\theta_0 = 0$





Kernels

Non-linear Decision Boundary

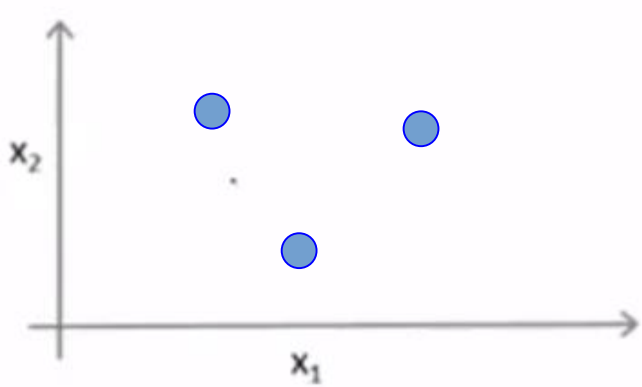


Predict $y = 1$ if

$$\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \theta_4 x_1^2 + \theta_5 x_2^2 + \cdots \geq 0$$

Kernels

Kernel



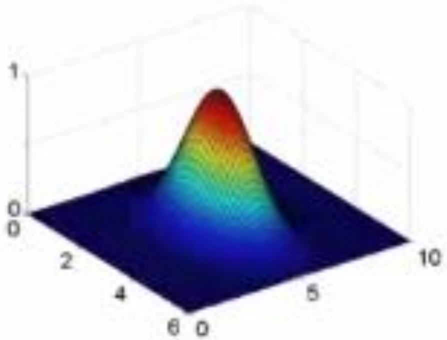
Given x, compute new feature depending on proximity to landmarks $l^{(1)}, l^{(2)}, l^{(3)}$

Kernels

Example:

$$l^{(1)} = \begin{bmatrix} 3 \\ 5 \end{bmatrix}, \quad f_1 = \exp\left(-\frac{\|x - l^{(1)}\|^2}{2\sigma^2}\right)$$

$$\sigma^2 = 1$$



Kernels

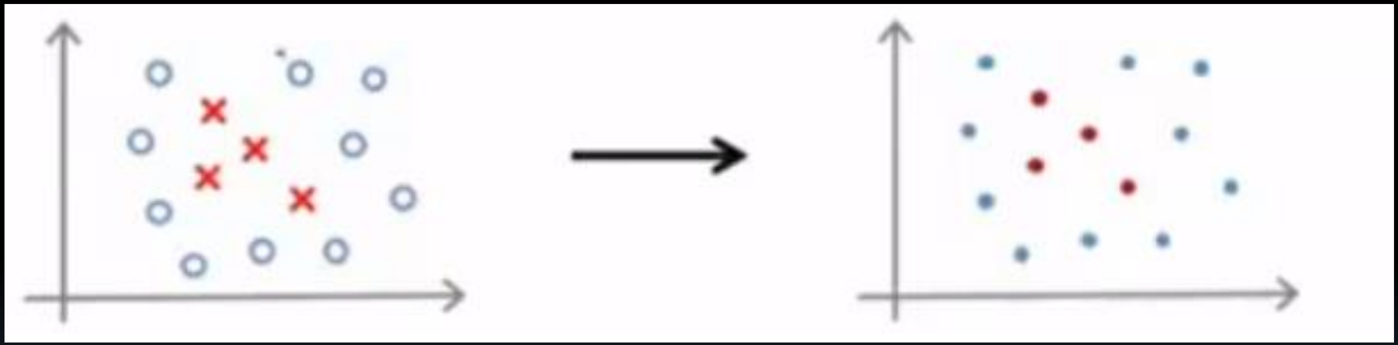
How to choose the landmarks?

What other functions can we use for kernel / similarity besides Gaussian?

Kernels

Landmarks

- Use example data as landmarks



Kernels

To get coefficients, solve for:

Training:

$$\min_\theta C \sum_{i=1}^{m} y^{(i)} cost_1(\theta^T f^{(i)}) + (1 - y^{(i)}) cost_0(\theta^T f^{(i)}) + \frac{1}{2} \sum_{j=1}^{n} \theta_j^2$$

Parameters

- Parameter C=1/λ
  - Large C: Lower bias, high variance
    - Prone to overfitting
  - Small C: Higher bias, low variance
    - Prone to underfitting
- σ^2
  - Large: Features vary more smoothly
    - Higher bias, lower variance
  - Small: Features vary less smoothly
    - Lower bias, higher variance

Using SVMs

- Use software package
  - Has built in optimization
  - Still need to specify parameters, plus the kernel you want to use
    - No kernel = linear
    - Gaussian kernel
    - Define your own kernel function
  - Important to scale features
    - Otherwise features with greater range will dominate

Using SVMs

- Multi-Class Classification
  - Use a One vs. All strategy
    - Train k SVMs to pick between k classes
    - Usually built in to software packages

Using SVMs

- Using logistic regression vs. SVM
  - Depends on n = number of features and m = number of examples
  - If n is large relative to m (n = 10,000; m = 10...1000)
    - Use logistic or SVM with linear kernel
  - If n is small, m is intermediate (n = 1...1000; m = 10...10,000)
    - Use SVM with Gaussian kernel
  - If n is small, m large (n = 1...1000; m = 50,000+)
    - Create more features manually, then use logistic or SVM with linear kernel

Outline

- Optimization Objective
- Large Margin Intuition
- Mathematics of Large Margin Classification
- Kernels
- Using an SVM

Summary

- Optimization Objective
- Large Margin Intuition
- Mathematics of Large Margin Classification
- Kernels
- Using an SVM