Support Vector Machine I

Jia-Bin Huang
Virginia Tech

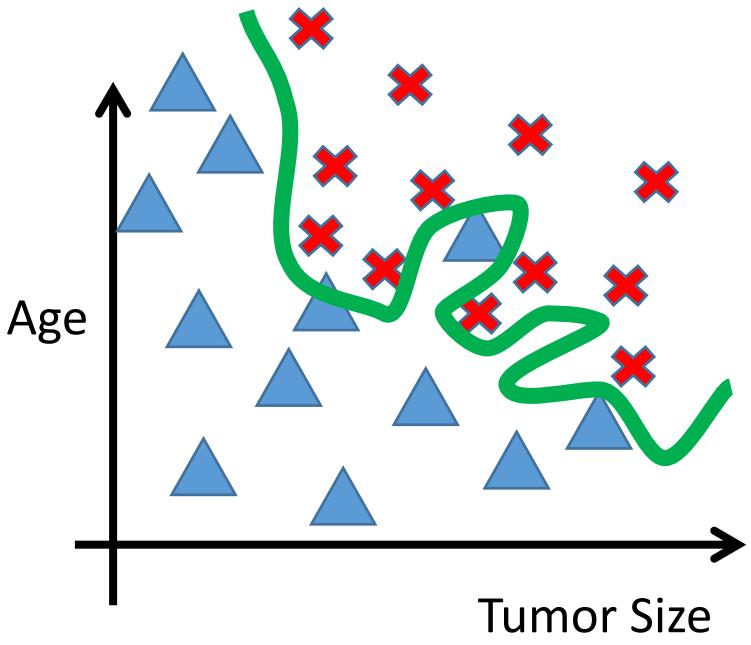
ECE-5424G / CS-5824

Spring 2019

Administrative

- Please use piazza. No emails.
- HW 0 grades are back. Re-grade request for one week.
- HW 1 due soon.
- HW 2 on the way.

Regularized logistic regression



$$h_\theta(x) = g(\theta_0 + \theta_1 x + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2 + \theta_6 x_1^3 x_2 + \theta_7 x_1 x_2^3 + \cdots)$$

- Cost function:

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

Slide credit: Andrew Ng

Gradient descent (Regularized)

Repeat {

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

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

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

}

$$\frac{\partial}{\partial \theta_j} J(\theta)$$

Slide credit: Andrew Ng

$|\theta|_1$: Lasso regularization

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

LASSO: Least Absolute Shrinkage and Selection Operator

Regression shrinkage and selection via the lasso

R Tibshirani - Journal of the Royal Statistical Society. Series B …, 1996 - JSTOR

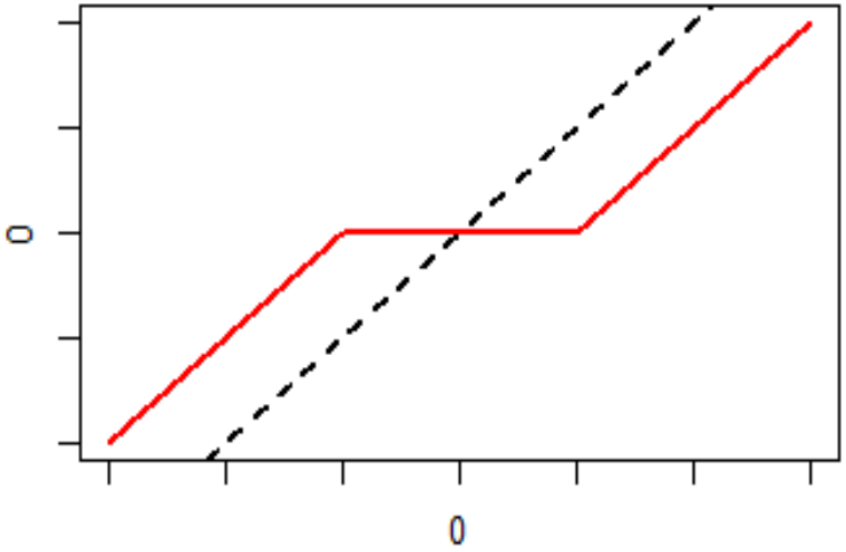
We propose a new method for estimation in linear models. Thelasso'minimizes the residual sum of squares subject to the sum of the absolute value of the coefficients being less than a constant. Because of the nature of this constraint it tends to produce some coefficients that are exactly 0 and hence gives interpretable models. Our simulation studies suggest that the lasso enjoys some of the favourable properties of both subset selection and ridge regression. It produces interpretable models like subset selection and exhibits the stability of …

☆ 99 Cited by 27003 Related articles All 60 versions Web of Science: 12902 Import into BibTeX

Single predictor: Soft Thresholding

- $\text{minimize}_\theta \ \frac{1}{2m}\sum_{i=1}^{m}\left(x^{(i)}\theta - y^{(i)}\right)^2 + \lambda|\theta|_1$

$$\hat{\theta} = \begin{cases} \frac{1}{m} < \boldsymbol{x}, \boldsymbol{y} > \ -\lambda & \text{if} \quad \frac{1}{m} < \boldsymbol{x}, \boldsymbol{y} > \ > \lambda \\ 0 & \text{if} \quad \frac{1}{m} | < \boldsymbol{x}, \boldsymbol{y} > | \ \leq \ \lambda \\ \frac{1}{m} < \boldsymbol{x}, \boldsymbol{y} > \ +\lambda & \text{if} \quad \frac{1}{m} < \boldsymbol{x}, \boldsymbol{y} > \ < -\lambda \end{cases}$$



$$\hat{\theta} = S_\lambda(\frac{1}{m} < \boldsymbol{x}, \boldsymbol{y} >)$$

Soft Thresholding operator $S_\lambda(x) = \text{sign}(x)(|x| - \lambda)_+$

Multiple predictors: : Cyclic Coordinate Descenet

- $\text{minimize}_\theta \ \frac{1}{2m}\sum_{i=1}^{m}\left(x_j^{(i)}\theta_j + \sum_{k\neq j} x_{ij}^{(i)}\theta_k - y^{(i)}\right)^2 + \lambda\sum_{k\neq j}|\theta_k| + \lambda\left|\theta_j\right|_1$

For each j, update θ_j with

$$\text{minimize}_\theta \ \frac{1}{2m}\sum_{i=1}^{m}\left(x_j^{(i)}\theta_j - r_j^{(i)}\right)^2 + \lambda\left|\theta_j\right|_1$$

where $r_j^{(i)} = y^{(i)} - \sum_{k\neq j} x_{ij}^{(i)}\theta_k$

L1 and L2 balls

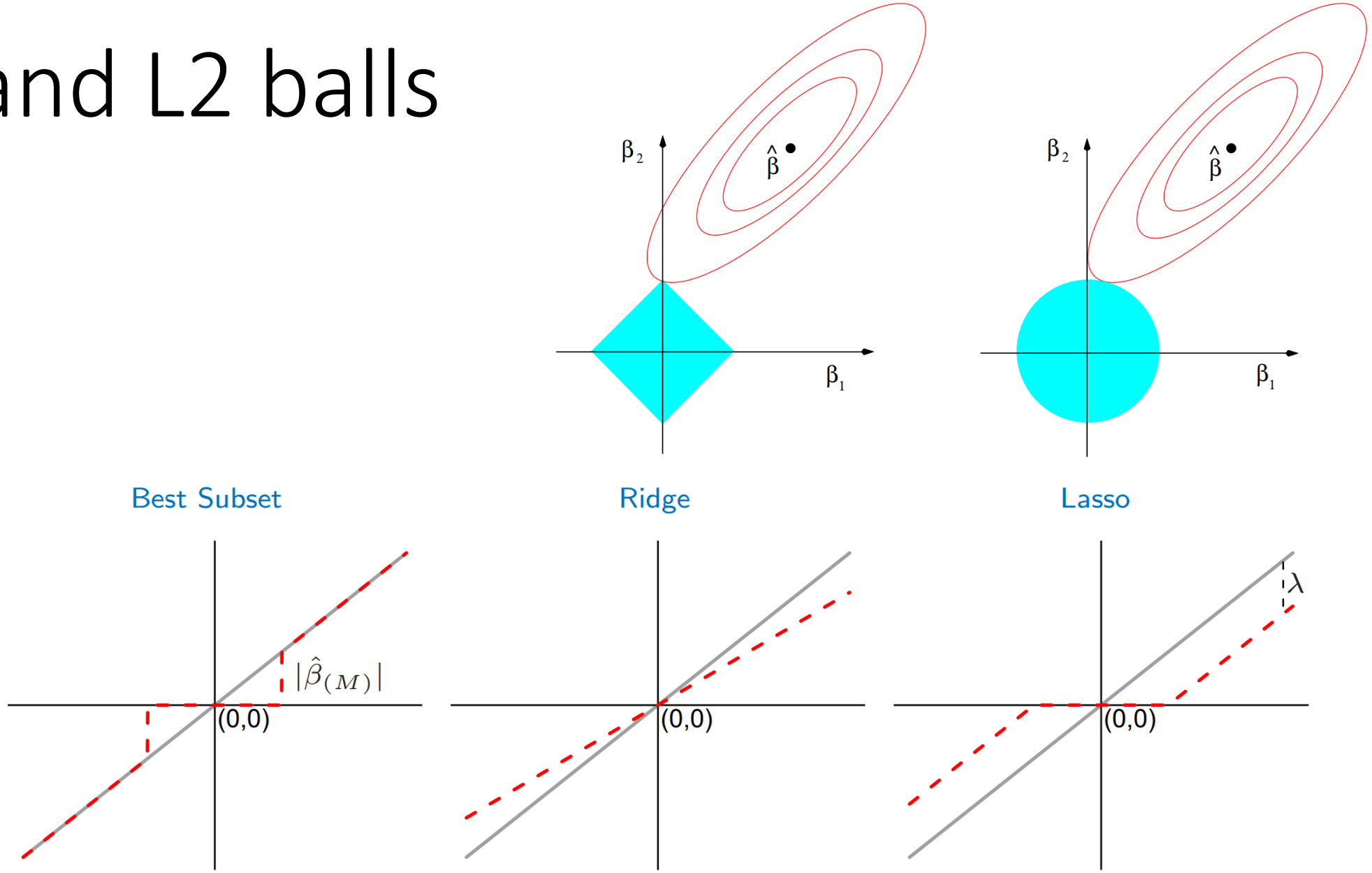


Image credit: https://web.stanford.edu/~hastie/StatLearnSparsity_files/SLS.pdf

Terminology

Regularization function	**Name**	**Solver**
$\|\theta\|_2^2 = \sum_{j=1}^{n} \theta_j^2$	Tikhonov regularization Ridge regression	Close form
$\left\|\|\theta\|\right\|_1 = \sum_{j=1}^{n} \|\theta_j\|$	LASSO regression	Proximal gradient descent, least angle regression
$\alpha\left\|\|\theta\|\right\|_1 + (1-\alpha)\ \|\theta\|_2^2$	Elastic net regularization	Proximal gradient descent

Things to remember

- **Overfitting**
  - Complex model: doing well on the training set, but perform poorly on the testing set
- **Cost function**
  - Norm penalty: L1 or L2
- **Regularized linear regression**
  - Gradient decent with weight decay (L2 norm)
- **Regularized logistic regression**
  - Gradient decent with weight decay (L2 norm)

Support Vector Machine

- Cost function
- Large margin classification
- Kernels
- Using an SVM

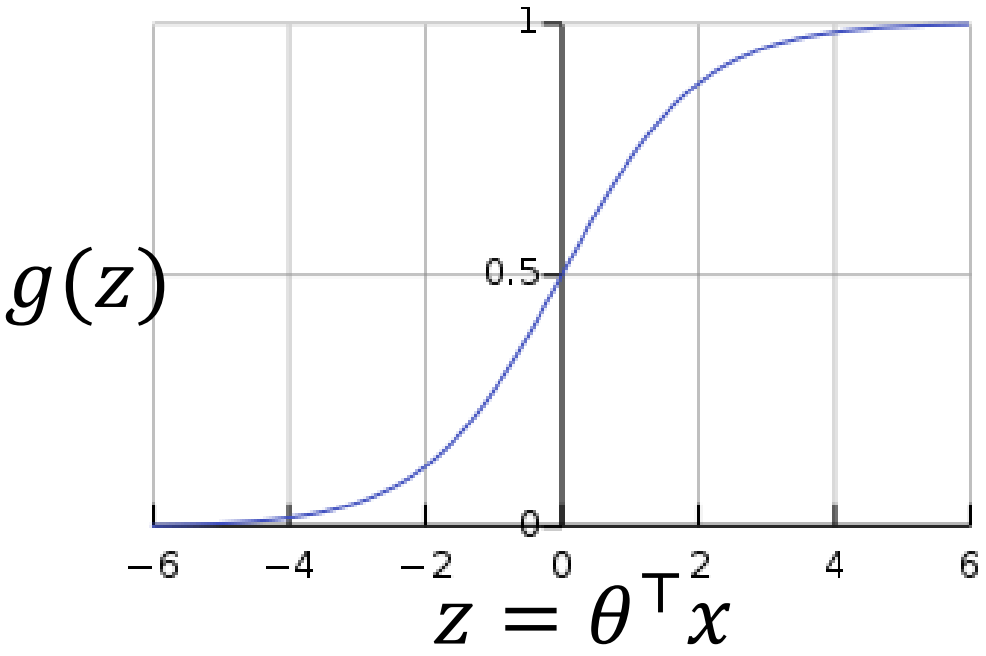
Support Vector Machine

- **Cost function**
- Large margin classification
- Kernels
- Using an SVM

Logistic regression

$$h_\theta(x) = g(\theta^\top x)$$

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



Suppose predict "y = 1" if $h_\theta(x) \geq 0.5$

$$z = \theta^\top x \geq 0$$

predict "y = 0" if $h_\theta(x) < 0.5$

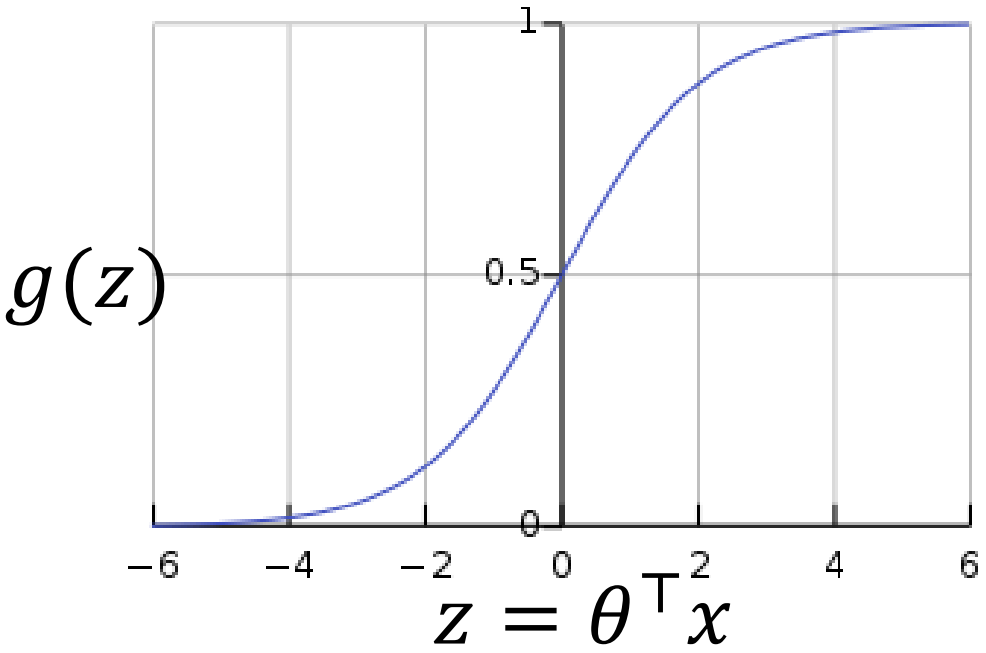
$$z = \theta^\top x < 0$$

Slide credit: Andrew Ng

Alternative view

$$h_\theta(x) = g(\theta^\top x)$$

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



If “y = 1”, we want $h_\theta(x) \approx 1$

$$z = \theta^\top x \gg 0$$

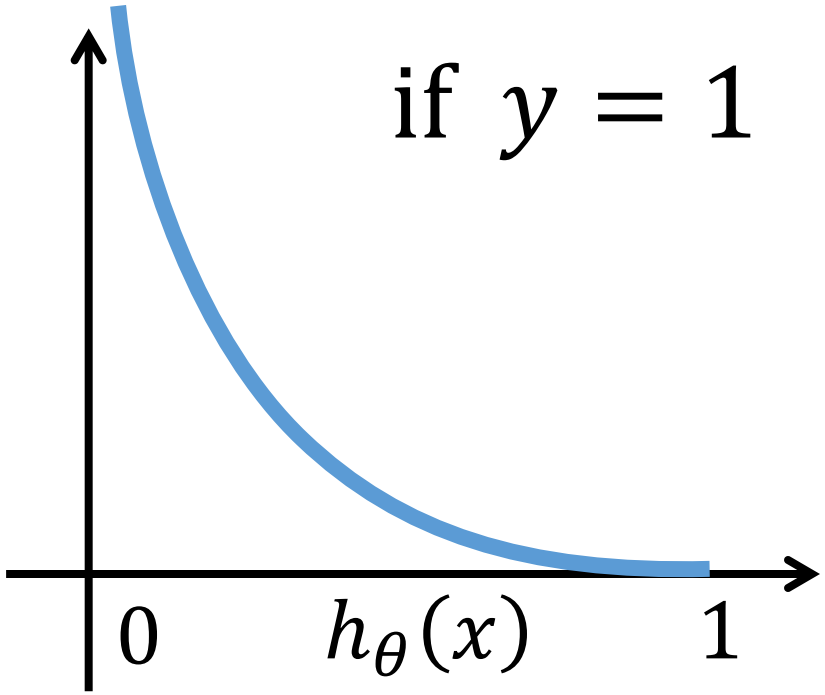
If “y = 0”, we want $h_\theta(x) \approx 0$

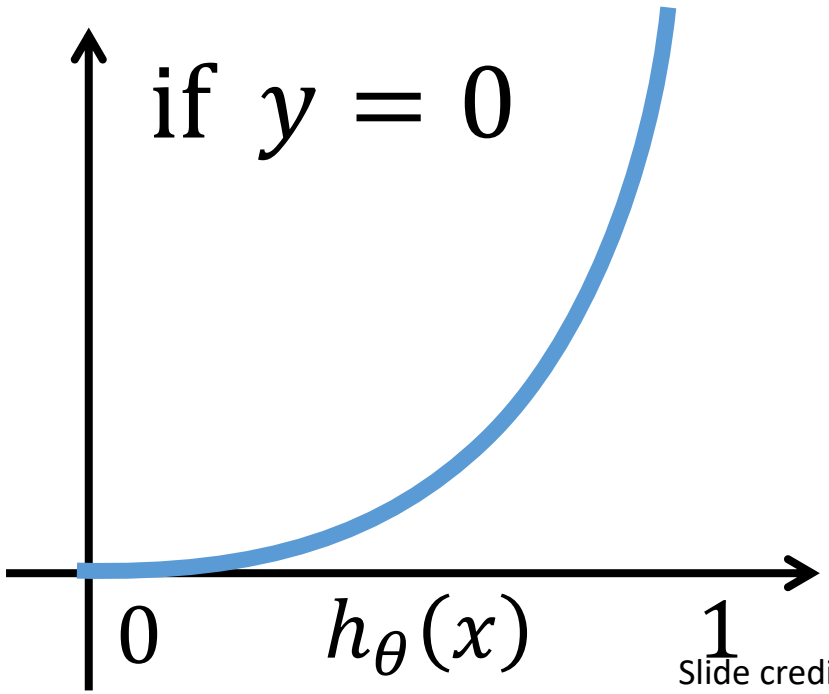
$$z = \theta^\top x \ll 0$$

Slide credit: Andrew Ng

Cost function for **Logistic Regression**

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



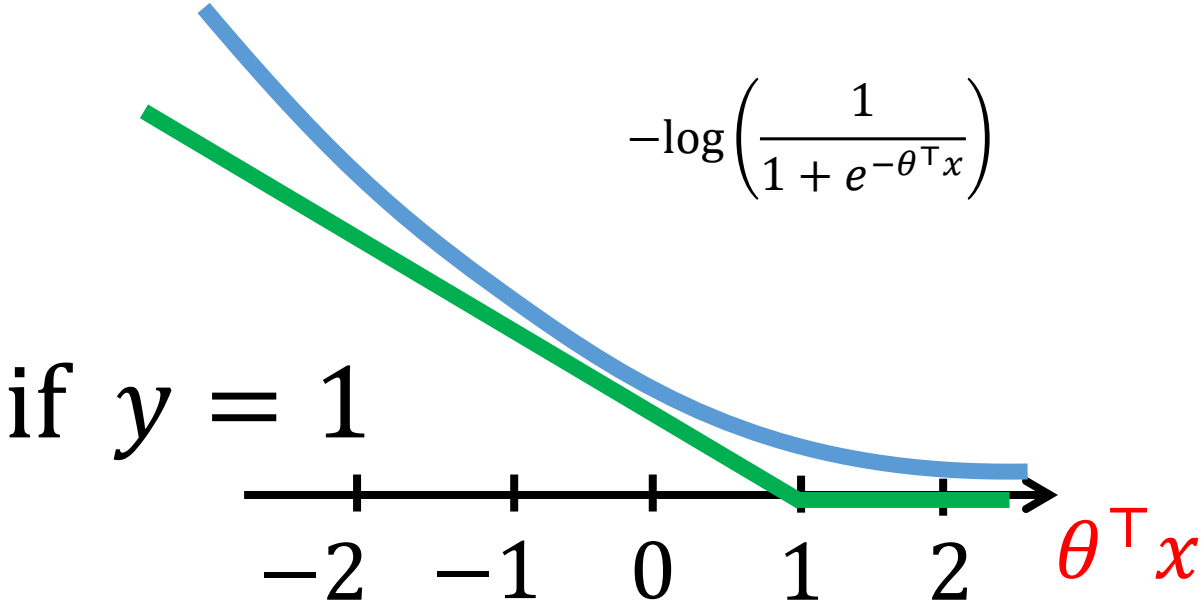


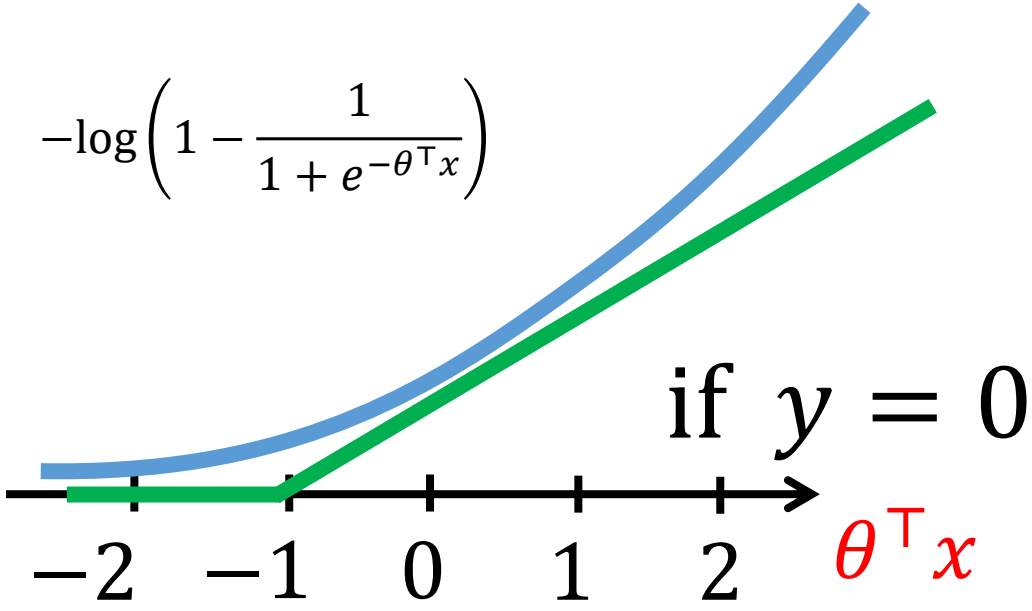
Slide credit: Andrew Ng

Alternative view of logistic regression

- $\mathrm{Cost}(h_\theta(x), y) = -\,y\ \log\big(\mathrm{h}_\theta(\mathrm{x})\big) - (1-\mathrm{y})\log\big(1 - h_\theta(x)\big)$

$$= y\left(-\log\left(\frac{1}{1+e^{-\theta^\top x}}\right)\right) + (1-\mathrm{y})\left(-\log\left(1-\frac{1}{1+e^{-\theta^\top x}}\right)\right)$$



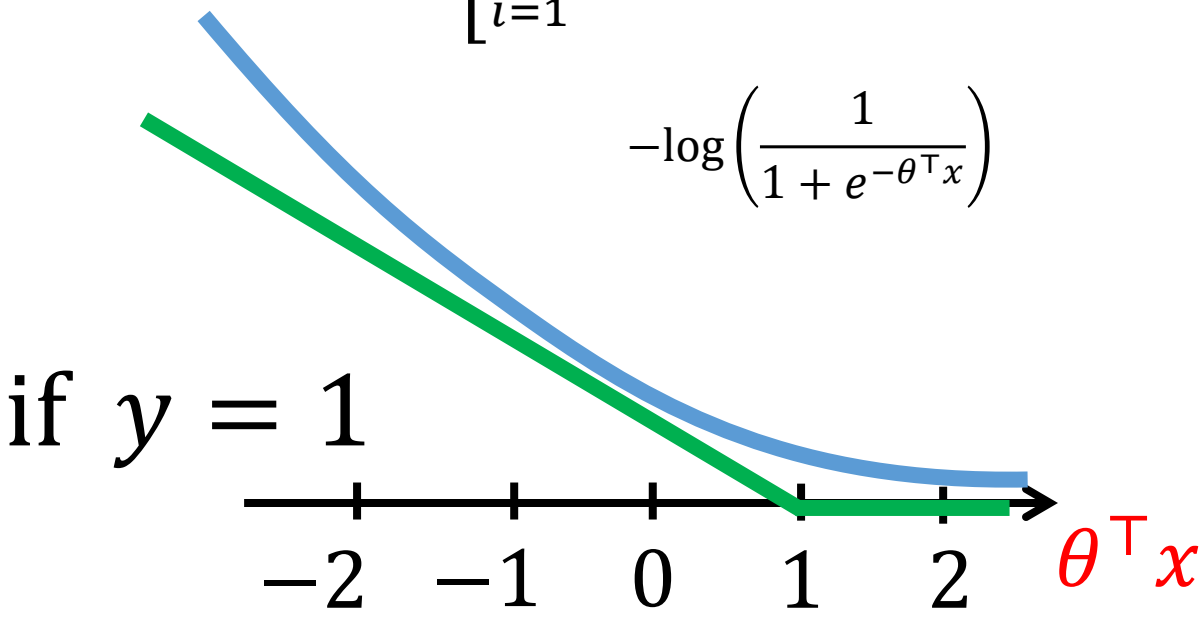


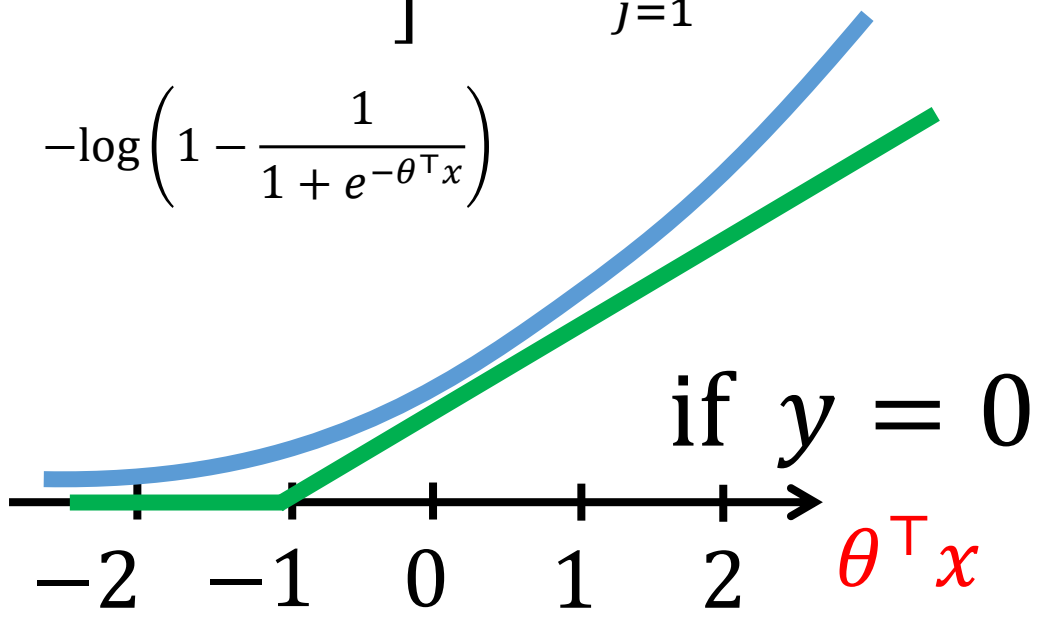
Logistic regression (logistic loss)

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

Support vector machine (hinge loss)

$$\min_{\theta} \frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)}\ \mathrm{cost}_1\left(\theta^\top x^{(i)}\right) + (1-y^{(i)})\ \mathrm{cost}_0\left(\theta^\top x^{(i)}\right)\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$





Optimization objective for SVM

$$\min_{\theta} \frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)}\ \mathrm{cost}_1\left(\theta^\top x^{(i)}\right) + \left(1 - y^{(i)}\right) \mathrm{cost}_0\left(\theta^\top x^{(i)}\right)\right] + \frac{\lambda}{2m}\sum_{j=1}^{n} \theta_j^2$$

1) Multiply $\frac{1}{m}$

2) Multiply $\mathrm{C} = \frac{1}{\lambda}$

$$\min_{\theta} C\left[\sum_{i=1}^{m} y^{(i)}\ \mathrm{cost}_1\left(\theta^\top x^{(i)}\right) + \left(1 - y^{(i)}\right) \mathrm{cost}_0\left(\theta^\top x^{(i)}\right)\right] + \sum_{j=1}^{n} \theta_j^2$$

Slide credit: Andrew Ng

Hypothesis of SVM

$$\min_{\theta} C\left[\sum_{i=1}^{m} y^{(i)}\ \operatorname{cost}_1\left(\theta^\top x^{(i)}\right) + \left(1 - y^{(i)}\right) \operatorname{cost}_0\left(\theta^\top x^{(i)}\right)\right] + \sum_{j=1}^{n} \theta_j^2$$

- Hypothesis

$$h_\theta(x) = \begin{cases} 1 & \text{if } \theta^\top x \geq 0 \\ 0 & \text{if } \theta^\top x < 0 \end{cases}$$

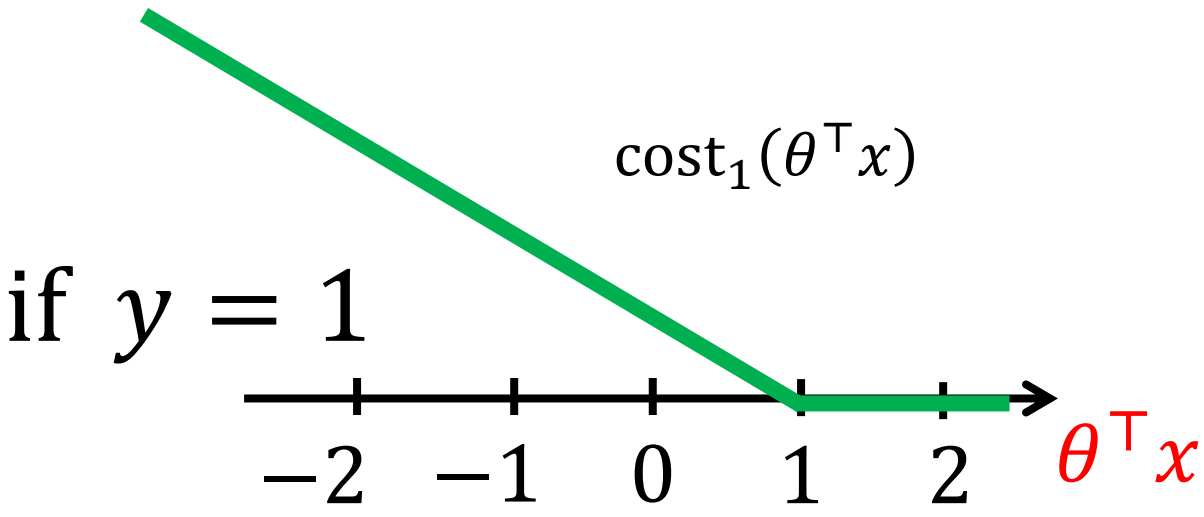
Slide credit: Andrew Ng

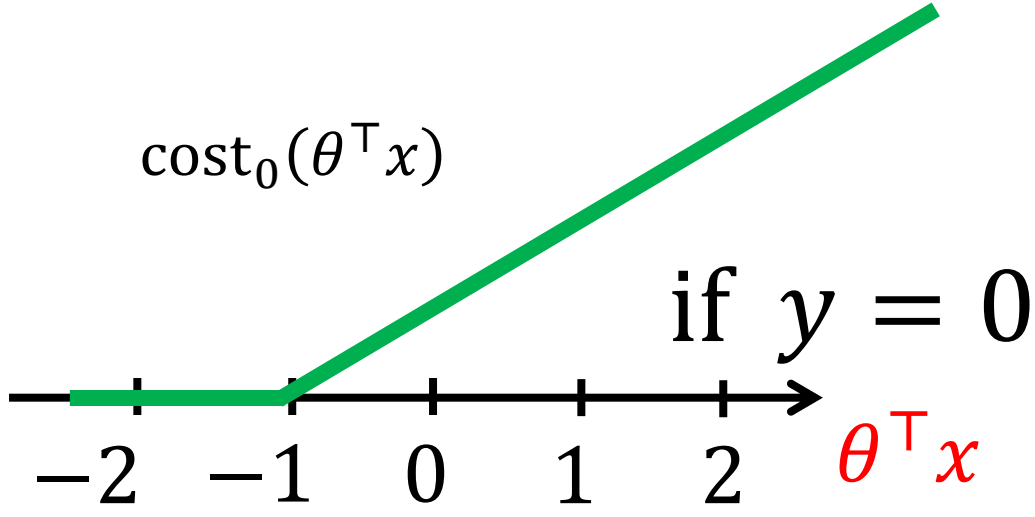
Support Vector Machine

- Cost function
- **Large margin classification**
- Kernels
- Using an SVM

Support vector machine

$$\min_{\theta} C\left[\sum_{i=1}^{m} y^{(i)}\ \mathrm{cost}_1\left(\theta^\top x^{(i)}\right) + \left(1 - y^{(i)}\right) \mathrm{cost}_0\left(\theta^\top x^{(i)}\right)\right] + \sum_{j=1}^{n} \theta_j^2$$





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

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

Slide credit: Andrew Ng

SVM decision boundary

$$\min_{\theta} C\left[\sum_{i=1}^{m} y^{(i)}\ \mathrm{cost}_1\left(\theta^\top x^{(i)}\right) + \left(1 - y^{(i)}\right)\mathrm{cost}_0\left(\theta^\top x^{(i)}\right)\right] + \frac{1}{2}\sum_{j=1}^{n}\theta_j^2$$

- Let's say we have a very large C …

- Whenever $y^{(i)} = 1$:

  $\theta^\top x^{(i)} \geq 1$

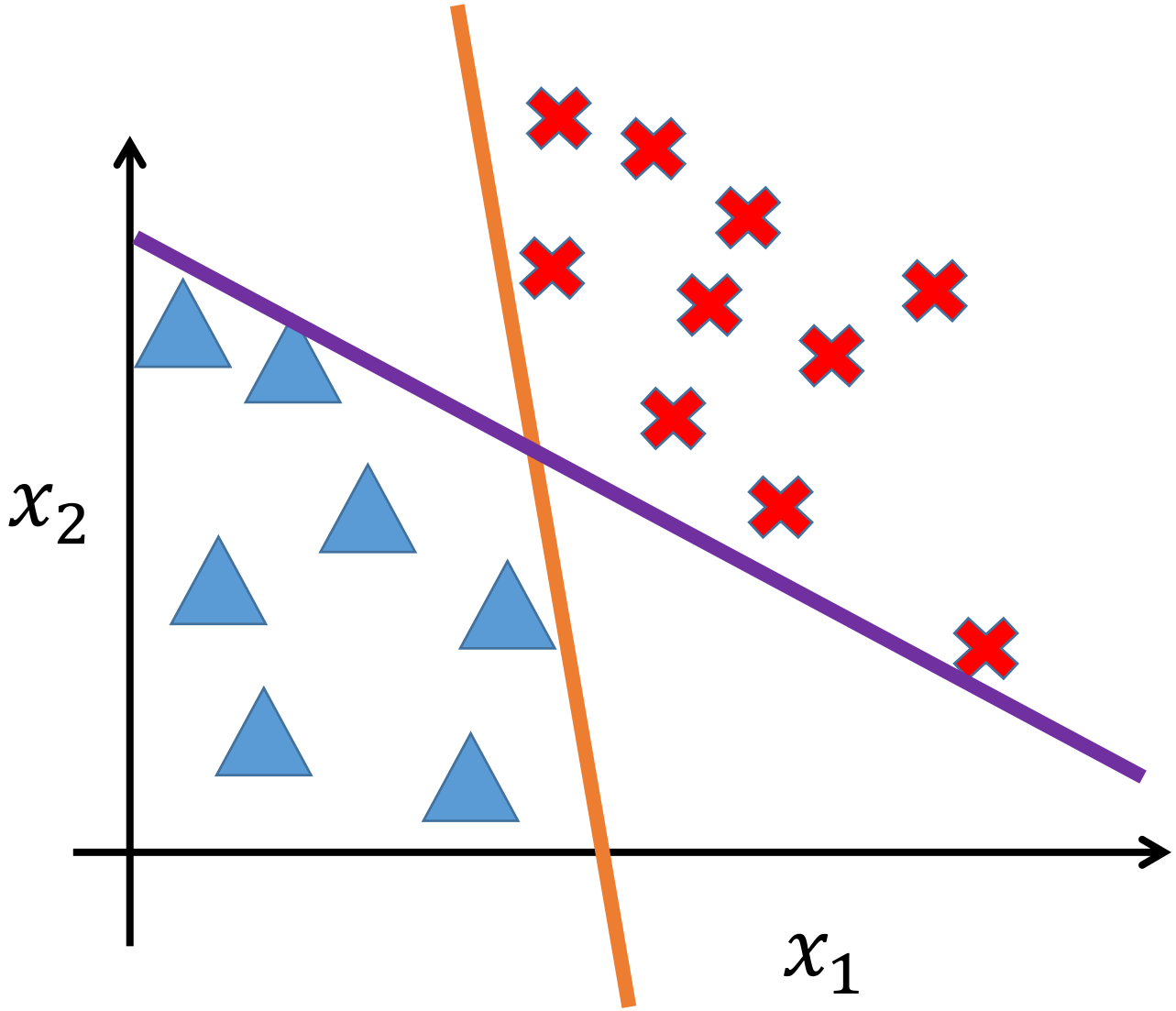
- Whenever $y^{(i)} = 0$:

  $\theta^\top x^{(i)} \leq -1$

$$\boxed{\begin{aligned} &\min_{\theta} \tfrac{1}{2}\textstyle\sum_{j=1}^{n}\theta_j^2 \\ &\text{s.t.} \quad \theta^\top x^{(i)} \geq 1 \quad \text{if} \quad \mathrm{y}^{(i)} = 1 \\ &\qquad\ \ \theta^\top x^{(i)} \leq -1 \quad \text{if} \quad \mathrm{y}^{(i)} = 0 \end{aligned}}$$

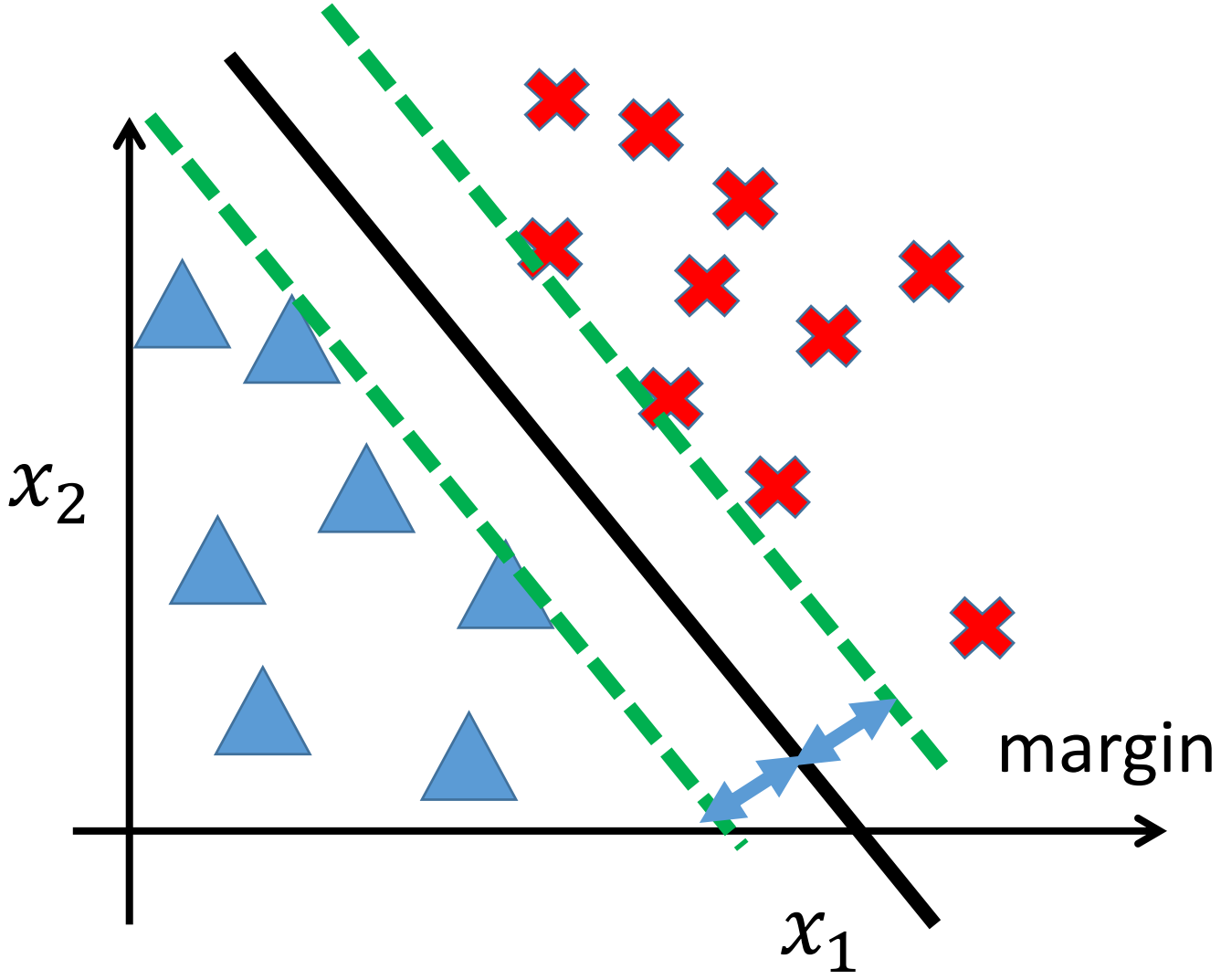
Slide credit: Andrew Ng

SVM decision boundary: Linearly separable case



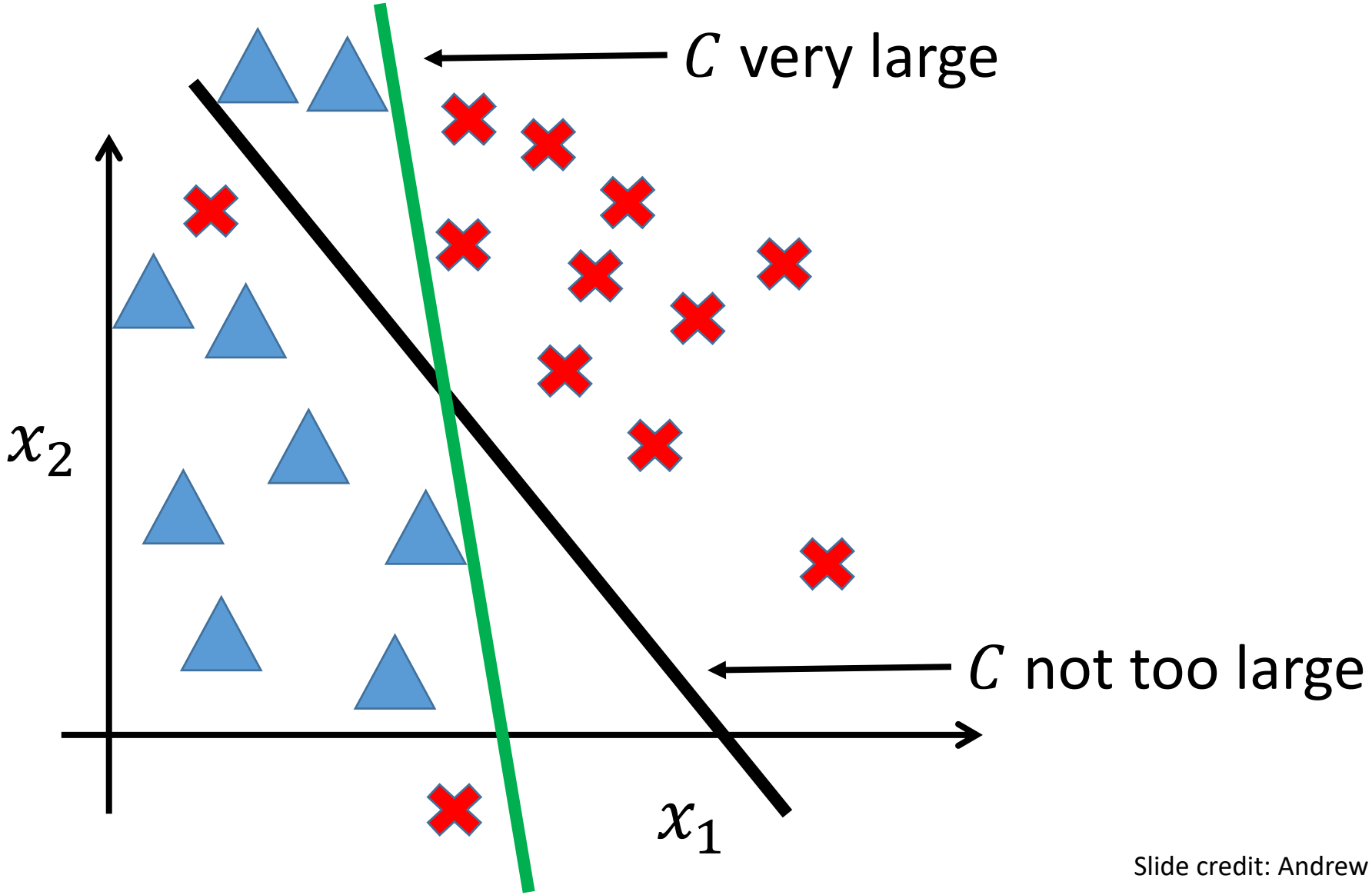
Slide credit: Andrew Ng

SVM decision boundary: Linearly separable case



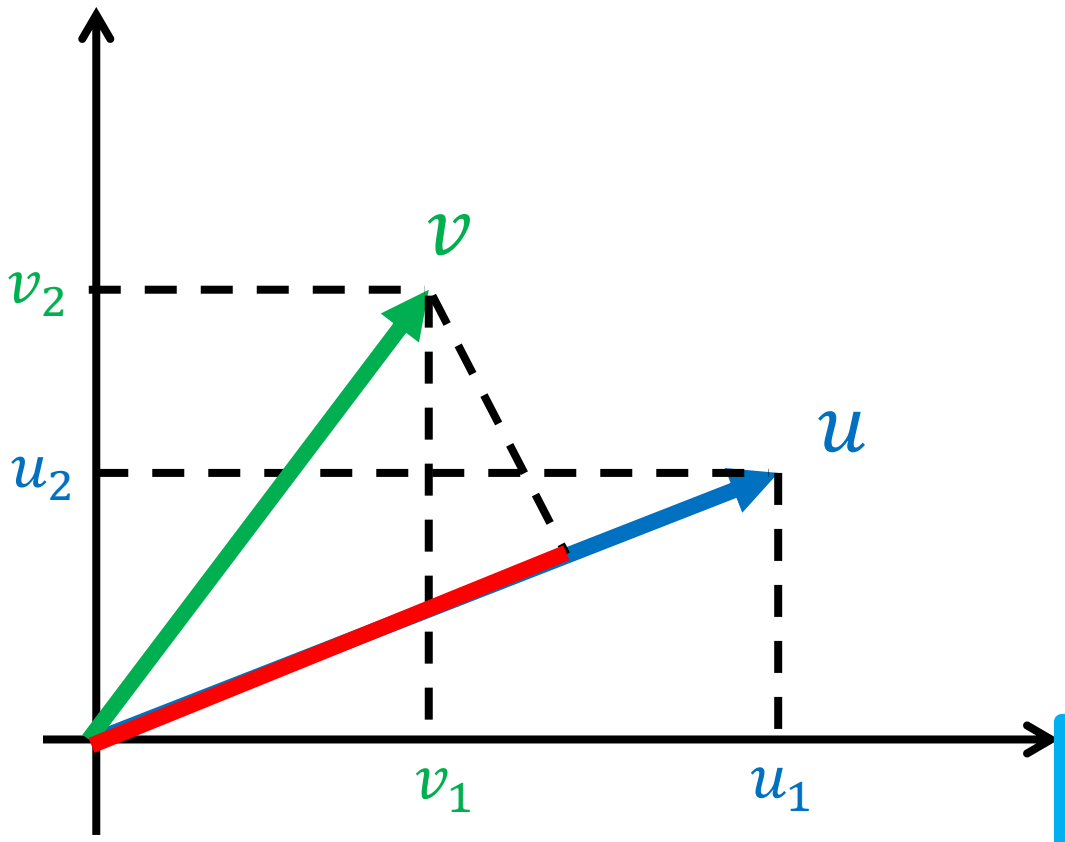
Slide credit: Andrew Ng

Large margin classifier in the presence of outlier



Slide credit: Andrew Ng

Vector inner product



$$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \qquad v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

$\|u\|$ = length of vector u

$$= \sqrt{u_1^2 + u_2^2} \in \mathbb{R}$$

p = length of projection of v onto u

$$\boxed{\begin{aligned} u^\top v &= p \cdot \|u\| \\ &= u_1 v_1 + u_2 v_2 \end{aligned}}$$

Slide credit: Andrew Ng

SVM decision boundary

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

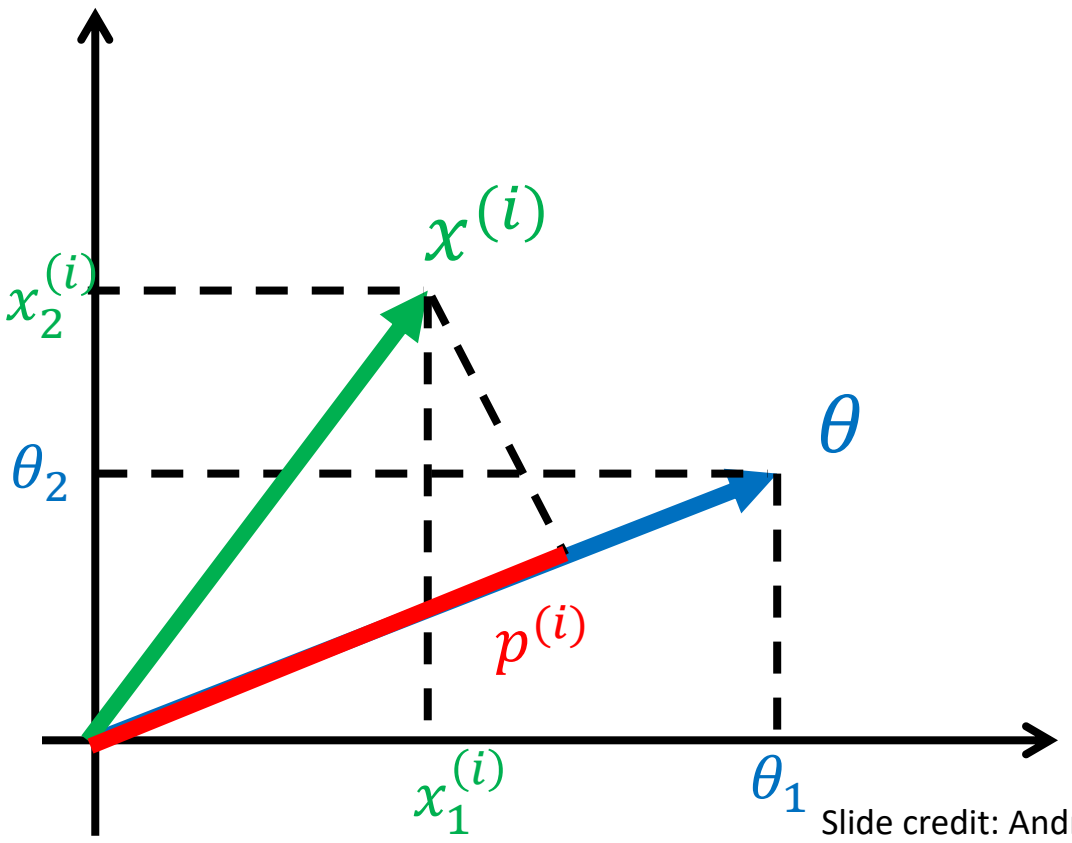
$$\text{s.t.} \quad \theta^\top x^{(i)} \geq 1 \quad \text{if} \quad y^{(i)} = 1$$
$$\theta^\top x^{(i)} \leq -1 \quad \text{if} \quad y^{(i)} = 0$$

$$\frac{1}{2}\sum_{j=1}^{n}\theta_j^2 = \frac{1}{2}\left(\theta_1^2 + \theta_2^2\right) = \frac{1}{2}\left(\sqrt{\theta_1^2 + \theta_2^2}\right)^2 = \frac{1}{2}\|\theta\|^2$$

Simplication: $\theta_0 = 0, n = 2$

What's $\theta^\top x^{(i)}$?

$$\theta^\top x^{(i)} = p^{(i)}\|\theta\|^2$$



Slide credit: Andrew Ng

SVM decision boundary

$$\min_{\theta} \frac{1}{2}\|\theta\|^2$$

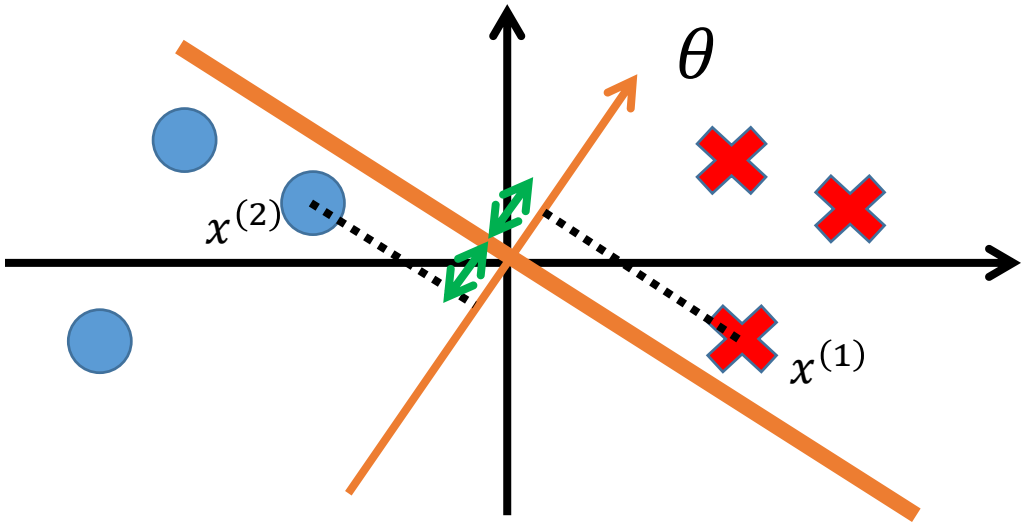
$$\text{s.t.} \quad p^{(i)}\|\theta\|^2 \geq 1 \quad \text{if} \quad y^{(i)} = 1$$

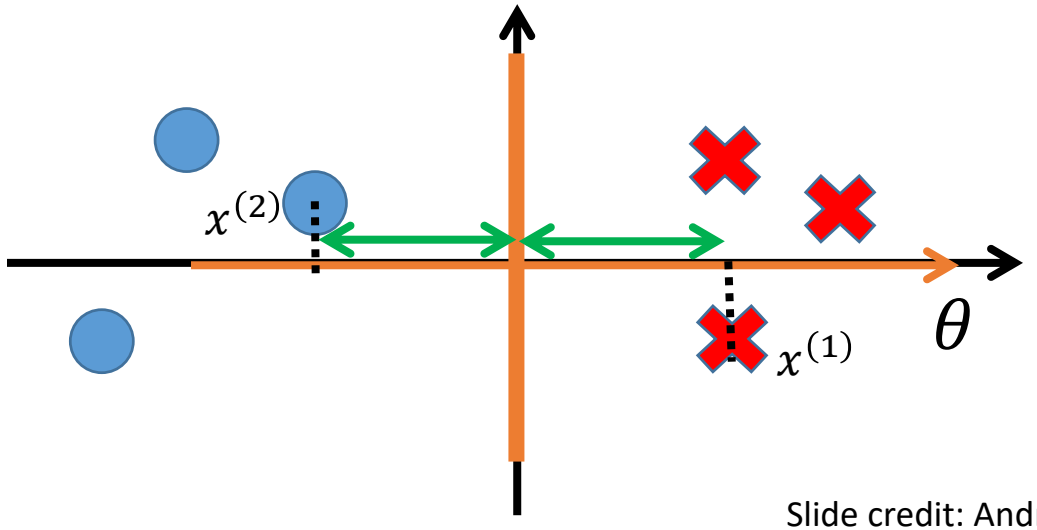
$$p^{(i)}\|\theta\|^2 \leq -1 \quad \text{if} \quad y^{(i)} = 0$$

Simplication: $\theta_0 = 0, n = 2$

$p^{(1)}, p^{(2)}$ small $\rightarrow \|\theta\|^2$ large

$p^{(1)}, p^{(2)}$ large $\rightarrow \|\theta\|^2$ can be small



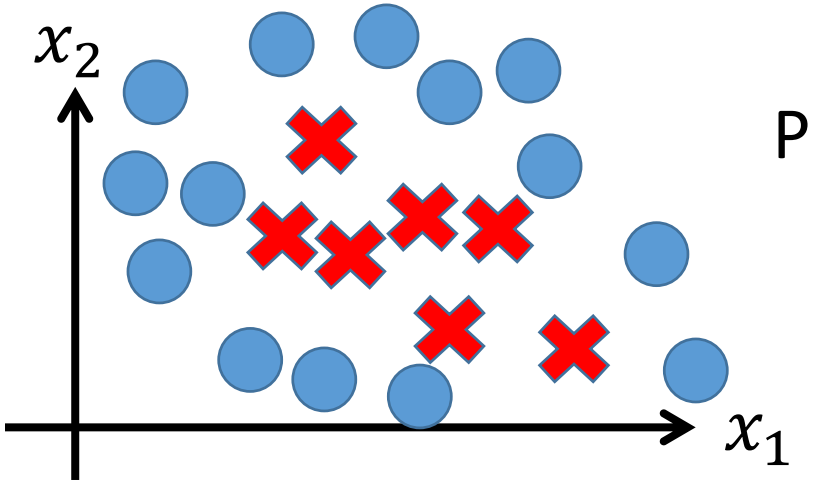


Slide credit: Andrew Ng

Support Vector Machine

- Cost function
- Large margin classification
- **Kernels**
- Using an SVM

Non-linear decision boundary



Predict $y = 1$ if

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

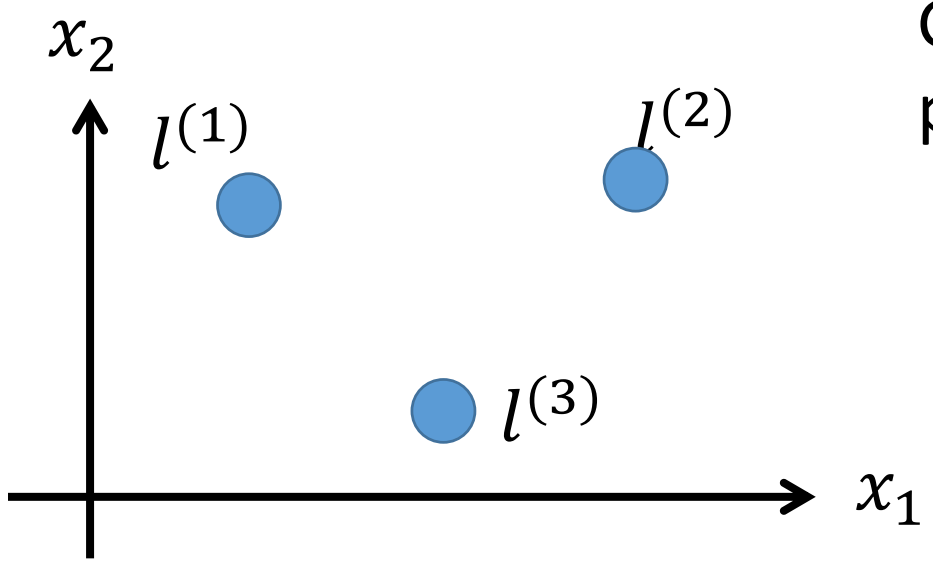
$$\theta_0 + \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 + \cdots$$

$$f_1 = x_1, f_2 = x_2, f_3 = x_1 x_2, \cdots$$

Is there a different/better choice of the features $f_1, f_2, f_3, \cdots$?

Slide credit: Andrew Ng

Kernel



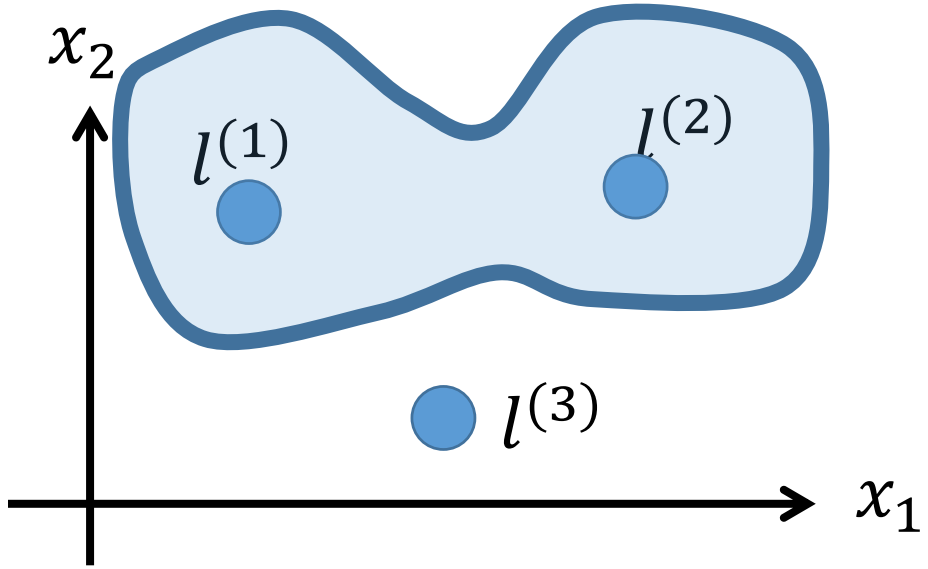
Give x, compute new features depending on proximity to landmarks $l^{(1)}$, $l^{(2)}$, $l^{(3)}$

$$f_1 = \text{similarity}(x, l^{(1)})$$
$$f_2 = \text{similarity}(x, l^{(2)})$$
$$f_3 = \text{similarity}(x, l^{(3)})$$

Gaussian kernel

$$\text{similarity}\left(x, l^{(i)}\right) = \exp\left(-\frac{\left\|x - l^{(i)}\right\|^2}{2\sigma^2}\right)$$

Slide credit: Andrew Ng



Predict $y = 1$ if

$$\theta_0 + \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 \geq 0$$

Ex: $\theta_0 = -0.5, \theta_1 = 1, \theta_2 = 1, \theta_3 = 0$

$$f_1 = \text{similarity}(x, l^{(1)})$$
$$f_2 = \text{similarity}(x, l^{(2)})$$
$$f_3 = \text{similarity}(x, l^{(3)})$$

Slide credit: Andrew Ng

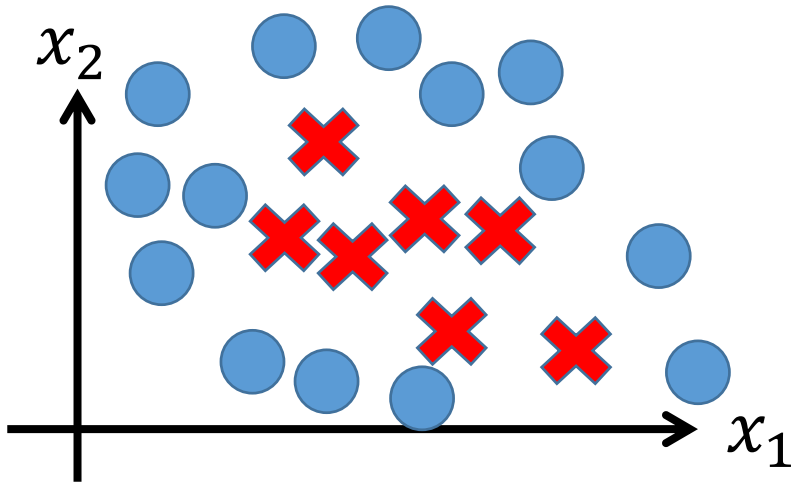
Choosing the landmarks

- Given x

$$f_i = \text{similarity}\left(x, l^{(i)}\right) = \exp\left(-\frac{\left\|x - l^{(i)}\right\|^2}{2\sigma^2}\right)$$

Predict $y = 1$ if $\theta_0 + \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 \geq 0$

Where to get $l^{(1)}, l^{(2)}, l^{(3)}, \cdots$?



$l^{(1)}$
$l^{(2)}$
$l^{(3)}$
$\vdots$
$l^{(m)}$

Slide credit: Andrew Ng

SVM with kernels

- Given $\left(x^{(1)}, y^{(1)}\right), \left(x^{(2)}, y^{(2)}\right), \cdots, \left(x^{(m)}, y^{(m)}\right)$
- Choose $l^{(1)} = x^{(1)}, l^{(2)} = x^{(2)}, l^{(3)} = x^{(3)}, \cdots, l^{(m)} = x^{(m)}$
- Given example x:
  - $f_1 = \text{similarity}\left(x, l^{(1)}\right)$
  - $f_2 = \text{similarity}\left(x, l^{(2)}\right)$
  - $\cdots$
- For training example $\left(x^{(i)}, y^{(i)}\right)$:
  - $x^{(i)} \rightarrow f^{(i)}$

$$f = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_m \end{bmatrix}$$

Slide credit: Andrew Ng

SVM with kernels

- Hypothesis: Given x, compute features $f \in \mathbb{R}^{m+1}$
  - Predict $y = 1 \quad \text{if} \quad \theta^\top f \geq 0$

- **Training (original)**

$$\min_\theta C\left[\sum_{i=1}^{m} y^{(i)}\ \text{cost}_1\left(\theta^\top x^{(i)}\right) + \left(1 - y^{(i)}\right)\text{cost}_0\left(\theta^\top x^{(i)}\right)\right] + \frac{1}{2}\sum_{j=1}^{n}\theta_j^2$$

- **Training (with kernel)**

$$\min_\theta C\left[\sum_{i=1}^{m} y^{(i)}\ \text{cost}_1\left(\theta^\top f^{(i)}\right) + \left(1 - y^{(i)}\right)\text{cost}_0\left(\theta^\top f^{(i)}\right)\right] + \frac{1}{2}\sum_{j=1}^{m}\theta_j^2$$

SVM parameters

- $C\left(=\frac{1}{\lambda}\right)$
  Large C: Lower bias, high variance.
  Small C: Higher bias, low variance.

- σ^2
- Large σ^2: features f_i vary more smoothly.
  - Higher bias, lower variance
- Small σ^2: features f_i vary less smoothly.
  - Lower bias, higher variance

Slide credit: Andrew Ng

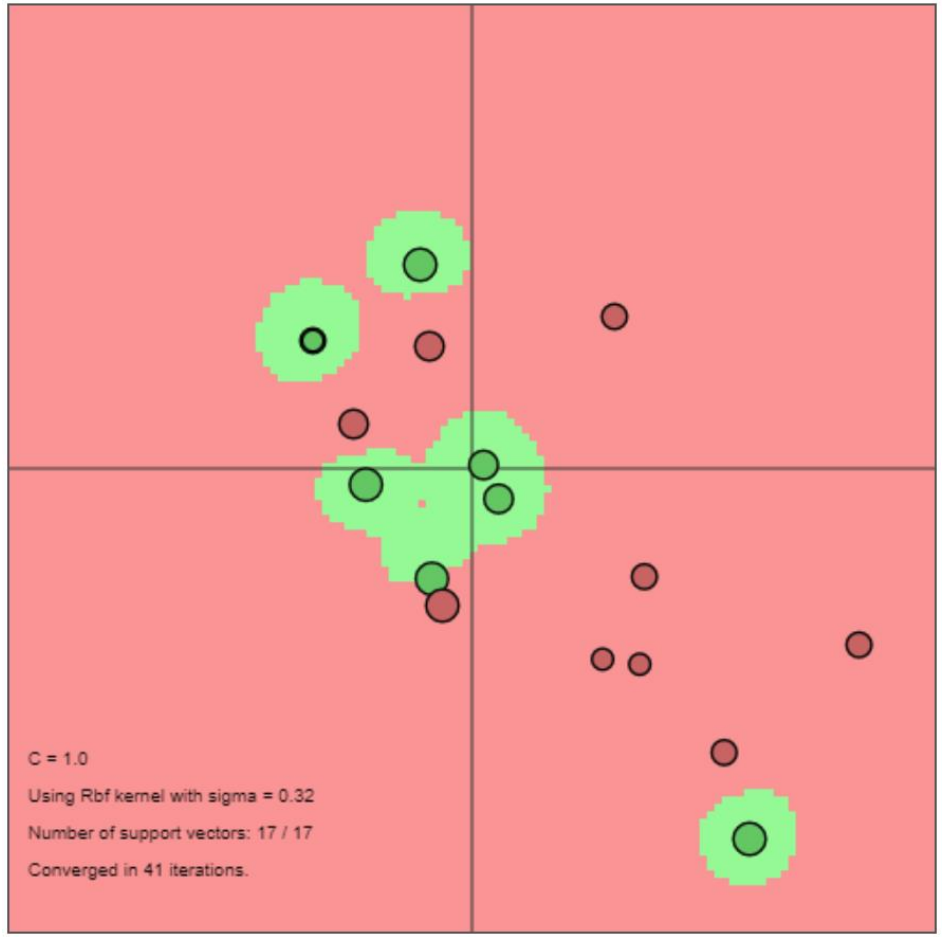
SVM song

- https://www.youtube.com/watch?v=g15bqtyidZs

Video source:

SVM Demo

- https://cs.stanford.edu/people/karpathy/svmjs/demo/



Support Vector Machine

- Cost function
- Large margin classification
- Kernels
- **Using an SVM**

Using SVM

- SVM software package (e.g., liblinear, libsvm) to solve for θ

- Need to specify:
  - Choice of parameter C.
  - Choice of kernel (similarity function):

- Linear kernel: Predict $y = 1 \quad \text{if} \quad \theta^\top x \geq 0$
- Gaussian kernel:
  - $f_i = \exp(-\frac{\|x - l^{(i)}\|^2}{2\sigma^2})$, where $l^{(i)} = x^{(i)}$
  - Need to choose σ^2. Need proper feature scaling

Slide credit: Andrew Ng

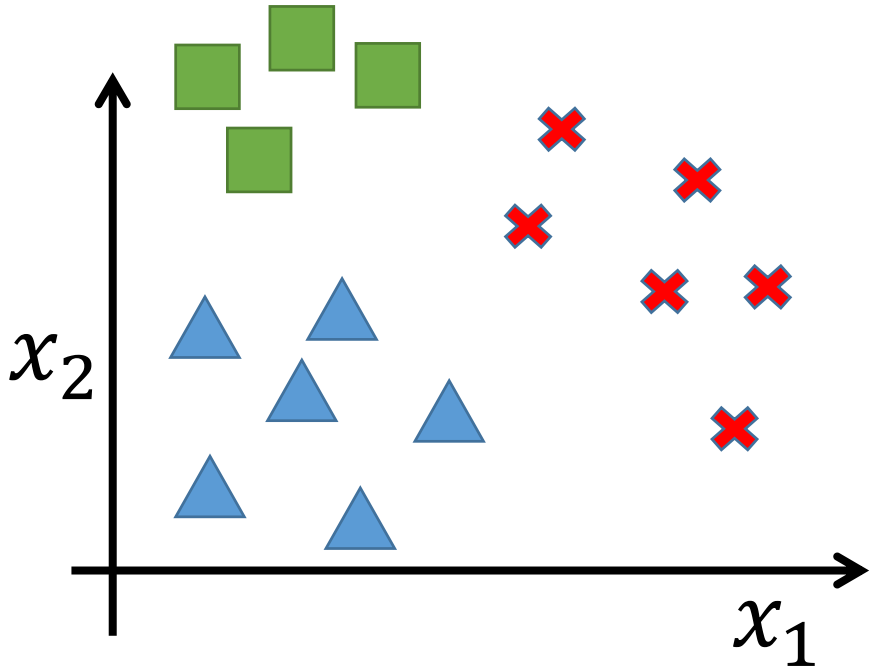
Kernel (similarity) functions

- Note: not all similarity functions make valid kernels.

- Many off-the-shelf kernsl available:
  - Polynomial kernel
  - String kernel
  - Chi-square kernel
  - Histogram intersection kernel

Slide credit: Andrew Ng

Multi-class classification



- Use one-vs.-all method. Train K SVMs, one to distinguish $y = i$ from the rest, get $\theta^{(1)}, \theta^{(2)}, \cdots, \theta^{(K)}$

- Pick class i with the largest ${\theta^{(i)}}^{\top} x$

Slide credit: Andrew Ng

Logistic regression vs. SVMs

- n = number of features ($x \in \mathbb{R}^{n+1}$), m = number of training examples

1. **If $\boldsymbol{n}$ is large (relative to $\boldsymbol{m}$)**: $(n = 10{,}000, \mathrm{m} = 10 - 1000)$
   → Use logistic regression or SVM without a kernel ("linear kernel")
2. **If $\boldsymbol{n}$ is small, $\boldsymbol{m}$ is intermediate**: $(n = 1 - 1000, \mathrm{m} = 10 - 10{,}000)$
   → Use SVM with Gaussian kernel
3. **If $\boldsymbol{n}$ is small, $\boldsymbol{m}$ is large**: $(n = 1 - 1000, \mathrm{m} = 50{,}000+)$
   → Create/add more features, then use logistic regression of linear SVM

Neural network likely to work well for most of these case,
but slower to train

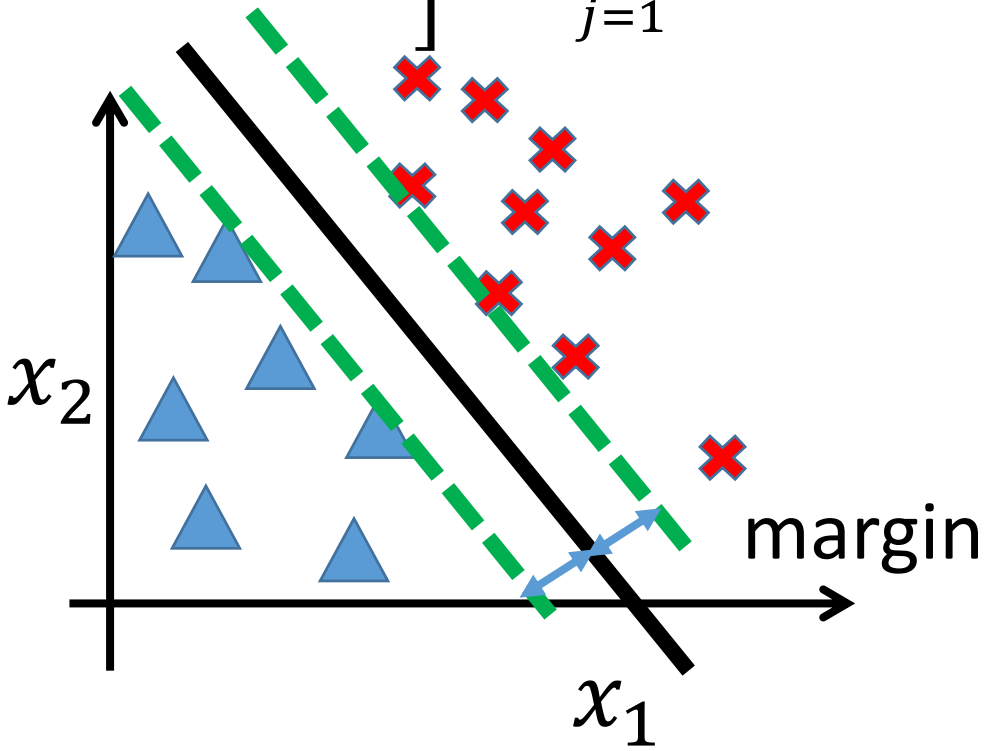
Slide credit: Andrew Ng

Things to remember

- **Cost function**

$$\min_{\theta} C\left[\sum_{i=1}^{m} y^{(i)}\ \mathrm{cost}_1\left(\theta^\top f^{(i)}\right) + \left(1 - y^{(i)}\right) \mathrm{cost}_0\left(\theta^\top f^{(i)}\right)\right] + \frac{1}{2}\sum_{j=1}^{m} \theta_j^2$$

- **Large margin classification**



- **Kernels**

- **Using an SVM**