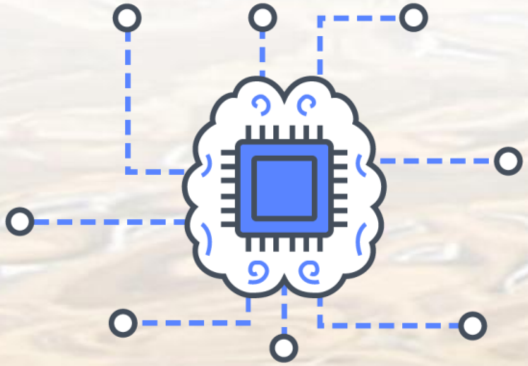


Machine Learning

Chapter 7: Sparse Kernel Machine

林嘉文 (Chia-Wen Lin)
清華大學電機系
cwlin@ee.nthu.edu.tw

Support Vector Machine

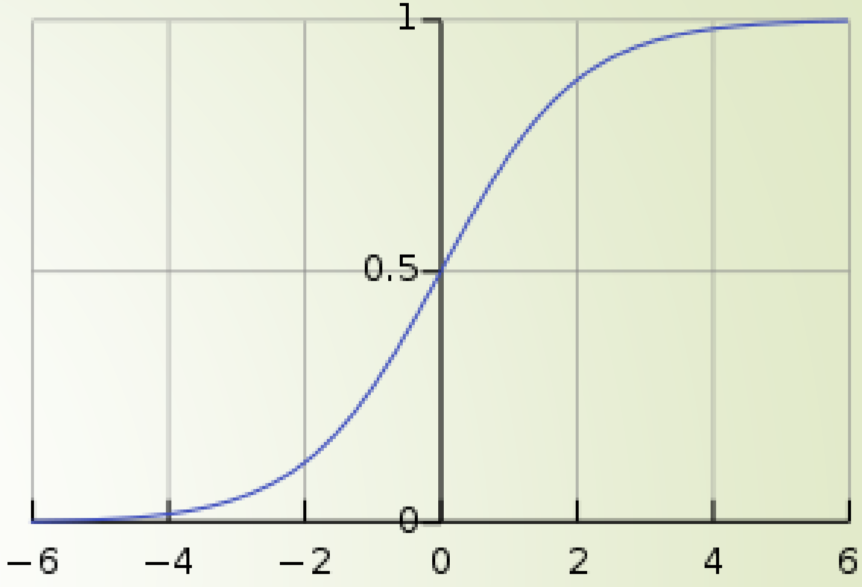
- Cost function
- Large margin classification
- Kernels
- Using an SVM

Support Vector Machine

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

Logistic Regression

$g(z)$



$z = \mathbf{w}^\top \mathbf{x}$

$$h_{\mathbf{w}}(\mathbf{x}) = g(\mathbf{w}^\top \mathbf{x})$$

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

Suppose predict "$y = 1$" if $h_{\mathbf{w}}(\mathbf{x}) \geq 0.5$

$z = \mathbf{w}^\top \mathbf{x} \geq 0$

predict "$y = 0$" if $h_{\mathbf{w}}(\mathbf{x}) < 0.5$

$z = \mathbf{w}^\top \mathbf{x} < 0$

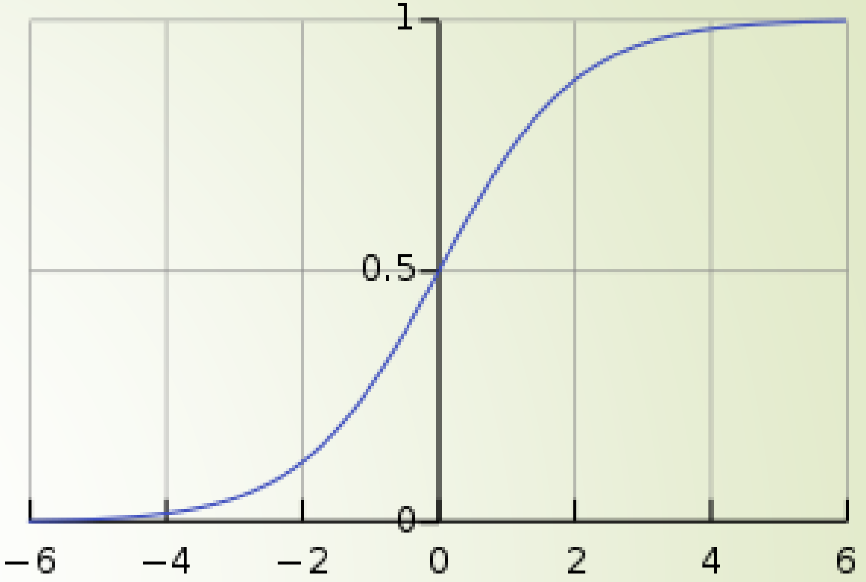
Alternative View

$$h_{\mathbf{w}}(\mathbf{x}) = g(\mathbf{w}^\top \mathbf{x})$$
$$g(z) = \frac{1}{1 + e^{-z}}$$

If "$y = 1$", we want $h_{\mathbf{w}}(\mathbf{x}) \approx 1$

If "$y = 0$", we want $h_{\mathbf{w}}(\mathbf{x}) \approx 0$

$g(z)$



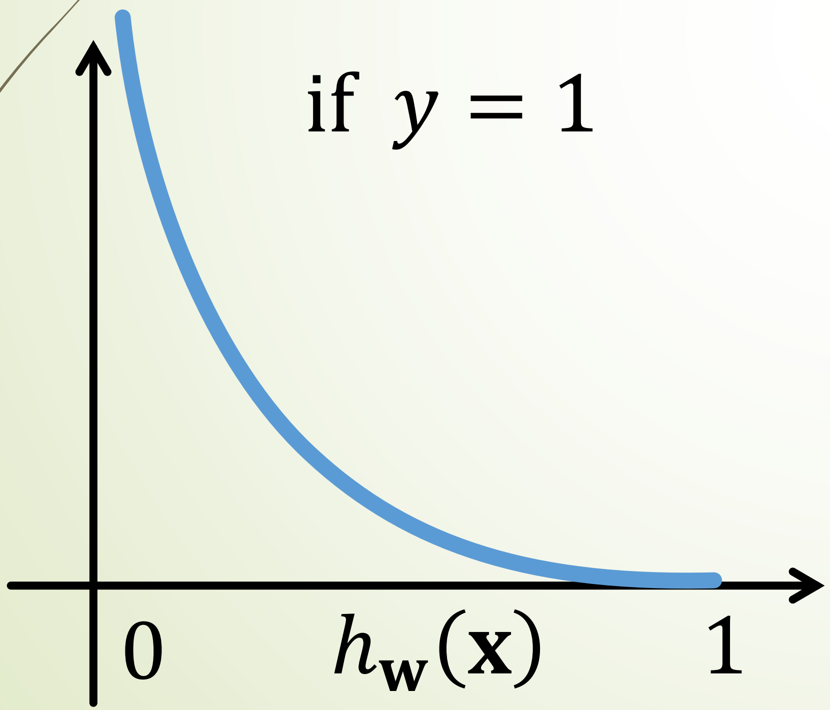
$z = \mathbf{w}^\top \mathbf{x}$

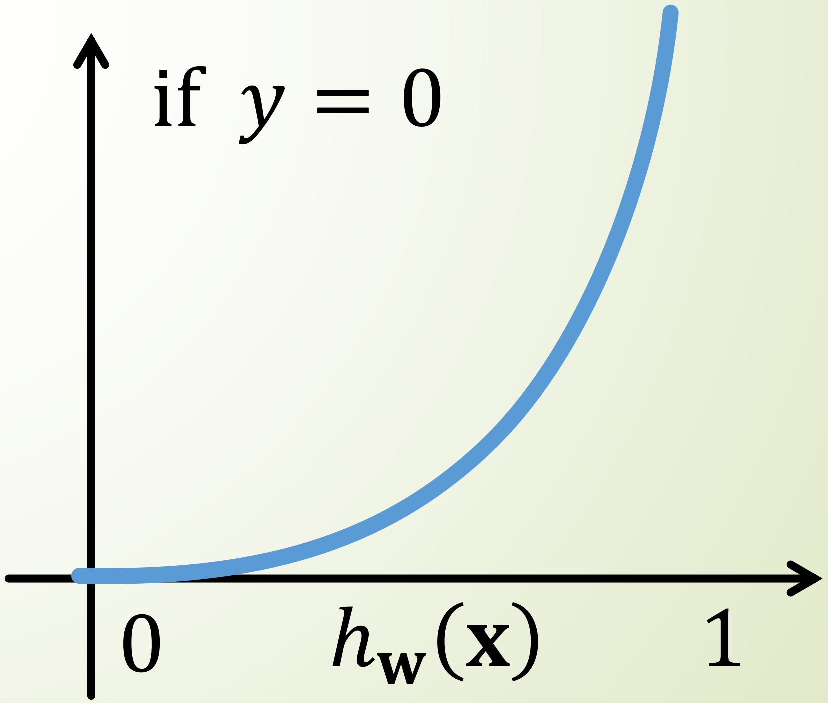
$z = \mathbf{w}^\top \mathbf{x} \gg 0$

$z = \mathbf{w}^\top \mathbf{x} \ll 0$

Cost Function for Logistic Regression

$$\mathrm{Cost}(h_{\mathbf{w}}(\mathbf{x}), y) = \begin{cases} -\log\left(h_{\mathbf{w}}(\mathbf{x})\right) & \text{if } y = 1 \\ -\log\left(1 - h_{\mathbf{w}}(\mathbf{x})\right) & \text{if } y = 0 \end{cases}$$

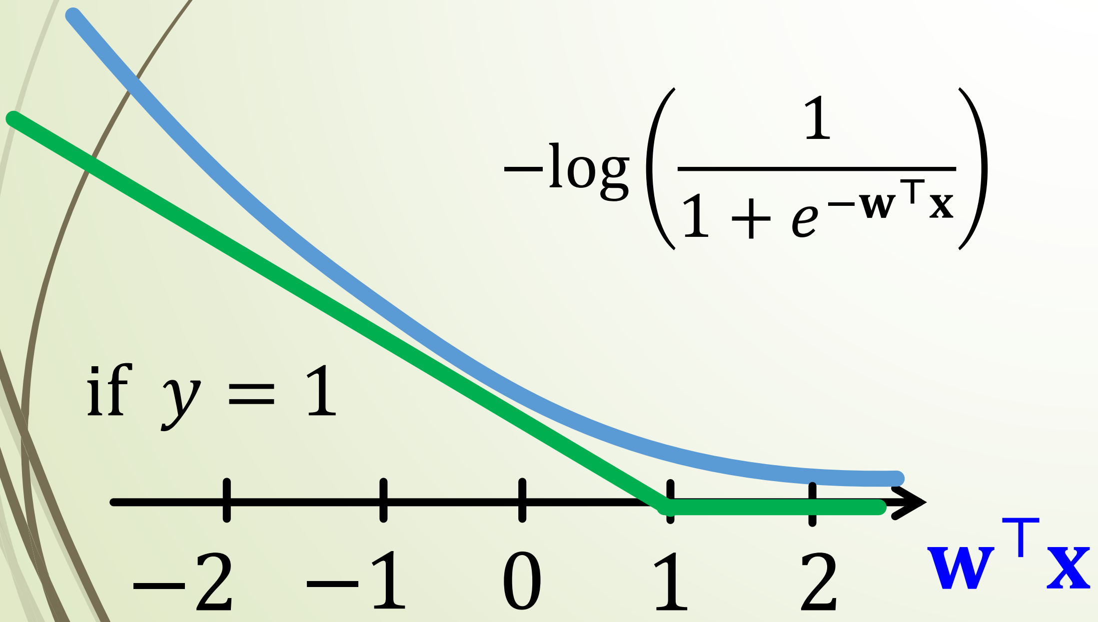


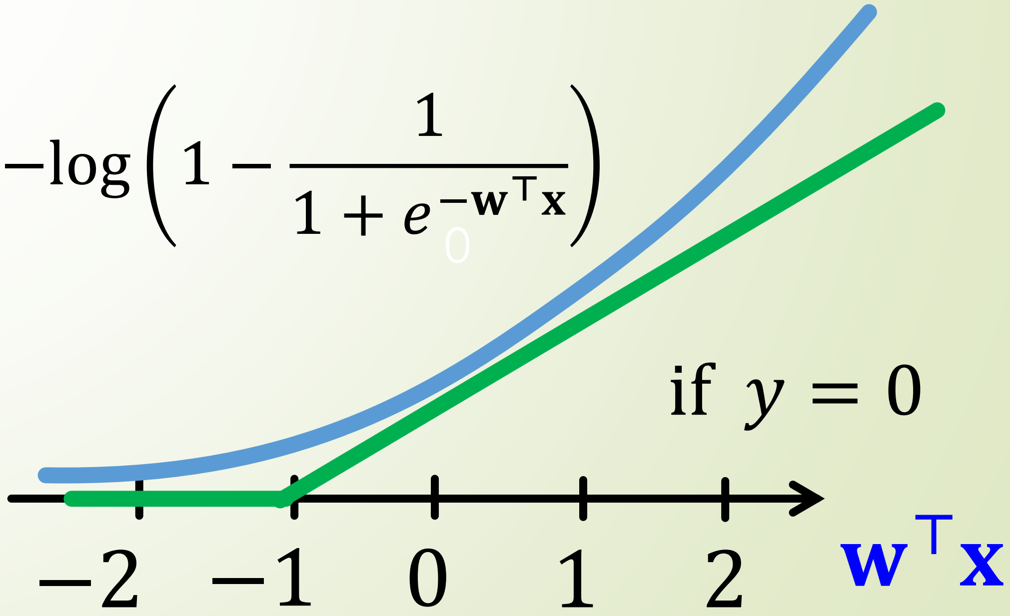


Alternative View of Logistic Regression

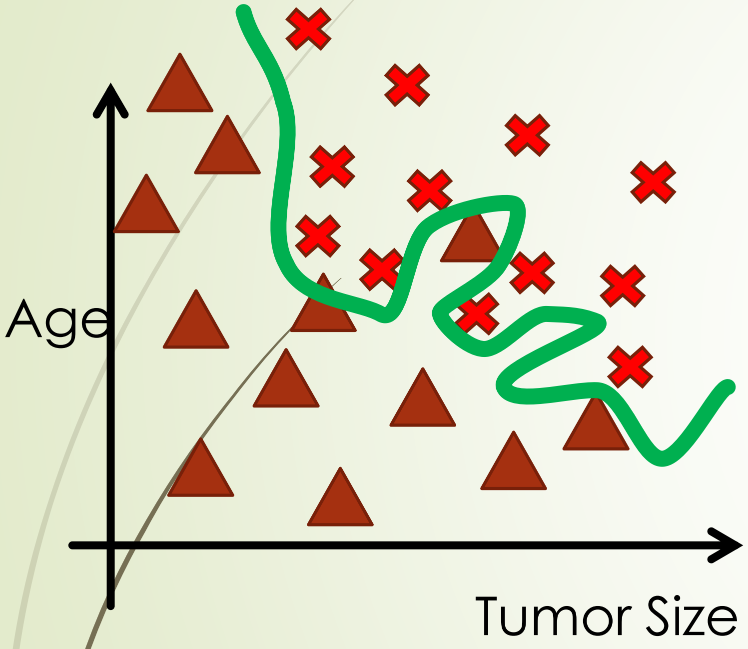
- $\mathrm{Cost}(h_{\mathbf{w}}(\mathbf{x}), y) = -\,y \cdot \log\big(h_{\mathbf{w}}(\mathbf{x})\big) - (1-y) \cdot \log\big(1 - h_{\mathbf{w}}(\mathbf{x})\big)$

$$= \; y \left(-\log\left(\frac{1}{1+e^{-\mathbf{w}^\top \mathbf{x}}} \right) \right) + (1-y) \left(-\log\left(1 - \frac{1}{1+e^{-\mathbf{w}^\top \mathbf{x}}} \right) \right)$$





Regularized Logistic Regression



$$h_{\mathbf{w}}(\mathbf{x}) = g(w_0 + w_1x + w_2x_2 + w_3x_1^2 + w_4x_2^2 + w_5x_1x_2 + w_6x_1^3x_2 + w_7x_1x_2^3 + \cdots)$$

- Cost function:

$$J(\mathbf{w}) = \frac{1}{N}\left[\sum_{i=1}^{N} y^{(i)} \log h_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right) + (1 - y^{(i)}) \log\left(1 - h_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right)\right) + \frac{\lambda}{2}\sum_{j=1}^{D} w_j^2\right]$$

Regularization Schemes

Regularization function	Name	Solver
$\|\|\mathbf{w}\|\|_2^2 = \sum_{j=1}^{D} w_j^2$	Tikhonov regularization Ridge regression	Closed form
$\|\|\mathbf{w}\|\|_1 = \sum_{j=1}^{D} \|w_j\|$	LASSO regression	Proximal gradient descent, least angle regression
$\alpha\|\|\mathbf{w}\|\|_1 + (1-\alpha)\ \|\|\mathbf{w}\|\|_2^2$	Elastic net regularization	Proximal gradient descent

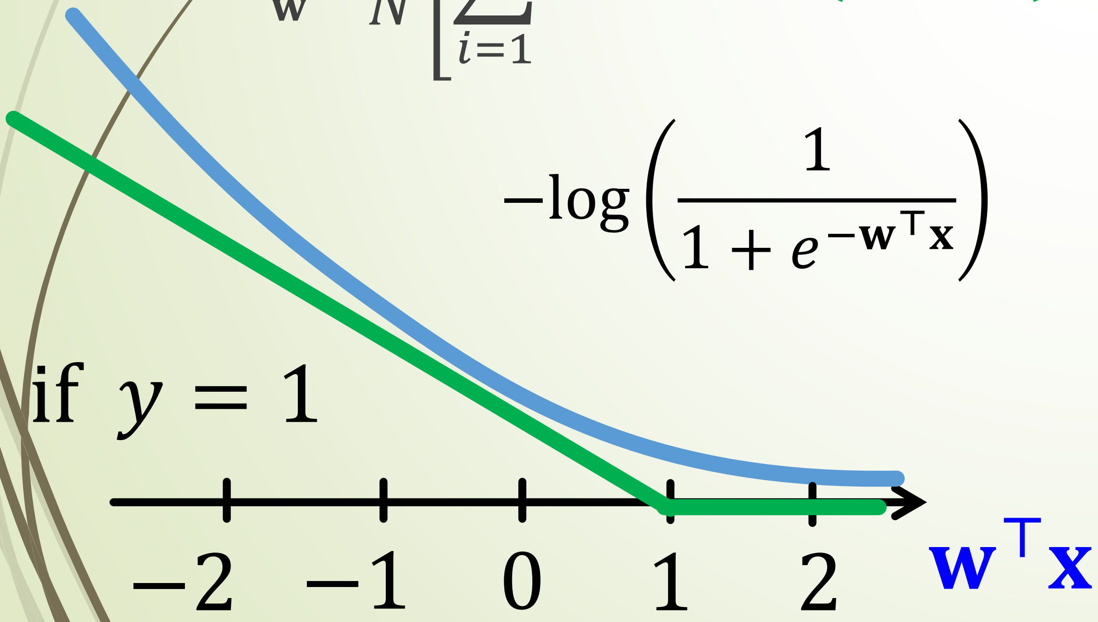
Logistic Regression (Logistic Loss)

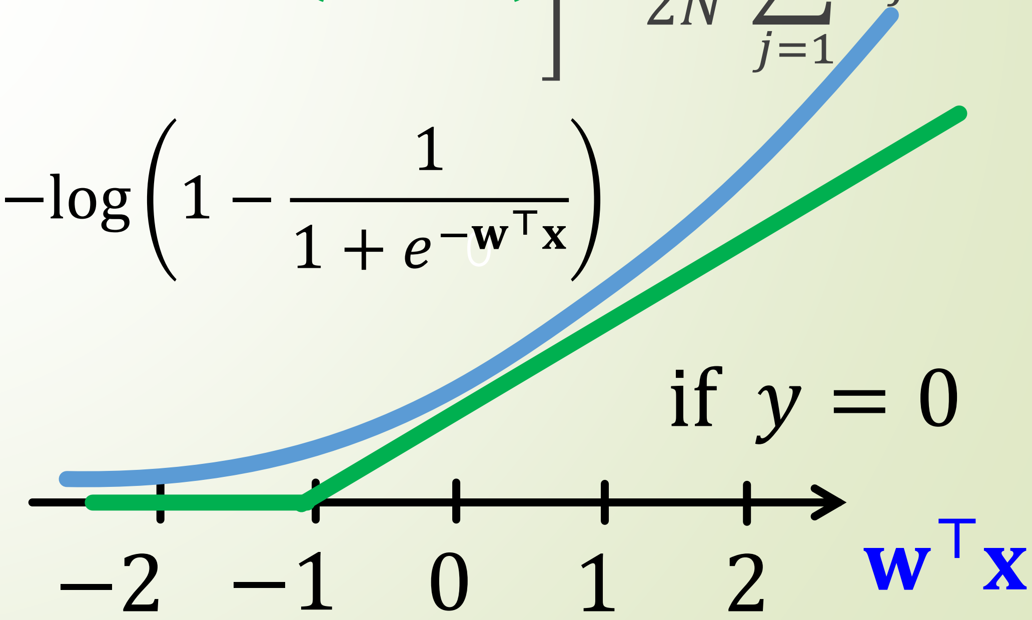
$$\min_{\mathbf{w}} \frac{1}{N}\left[\sum_{i=1}^{N} y^{(i)}\left(-\log\left(h_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right)\right)\right) + (1-y^{(i)})\left(-\log\left(1-h_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right)\right)\right)\right] + \frac{\lambda}{2N}\sum_{j=1}^{D} w_j^2$$

Support vector machine (**hinge loss**)

$$\min_{\mathbf{w}} \frac{1}{N}\left[\sum_{i=1}^{N} y^{(i)}\ \mathrm{cost}_1\left(\mathbf{w}^\top\mathbf{x}^{(i)}\right) + (1-y^{(i)})\ \mathrm{cost}_0\left(\mathbf{w}^\top\mathbf{x}^{(i)}\right)\right] + \frac{\lambda}{2N}\sum_{j=1}^{D} w_j^2$$

$$-\log\left(\frac{1}{1+e^{-\mathbf{w}^\top\mathbf{x}}}\right) \qquad -\log\left(1-\frac{1}{1+e^{-\mathbf{w}^\top\mathbf{x}}}\right)$$





Optimization Objective for SVM

$$\min_{\mathbf{w}} \frac{1}{N}\left[\sum_{i=1}^{N} y^{(i)}\ \mathrm{cost}_1\left(\mathbf{w}^\top \mathbf{x}^{(i)}\right) + (1 - y^{(i)})\ \mathrm{cost}_0\left(\mathbf{w}^\top \mathbf{x}^{(i)}\right)\right] + \frac{\lambda}{2N}\sum_{j=1}^{D} w_j^2$$

1) Divide $\frac{1}{N}$

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

$$\downarrow$$

$$\min_{\mathbf{w}} C\left[\sum_{i=1}^{N} y^{(i)}\ \mathrm{cost}_1\left(\mathbf{w}^\top \mathbf{x}^{(i)}\right) + (1 - y^{(i)})\ \mathrm{cost}_0\left(\mathbf{w}^\top \mathbf{x}^{(i)}\right)\right] + \frac{1}{2}\sum_{j=1}^{D} w_j^2$$

Slide credit: Andrew Ng

Hypothesis of SVM

$$\min_{\mathbf{w}} C\left[\sum_{i=1}^{N} y^{(i)}\ \mathrm{cost}_1\left(\mathbf{w}^\top \mathbf{x}^{(i)}\right) + (1 - y^{(i)})\ \mathrm{cost}_0\left(\mathbf{w}^\top \mathbf{x}^{(i)}\right)\right] + \frac{1}{2}\sum_{j=1}^{D} w_j^2$$

- Hypothesis

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

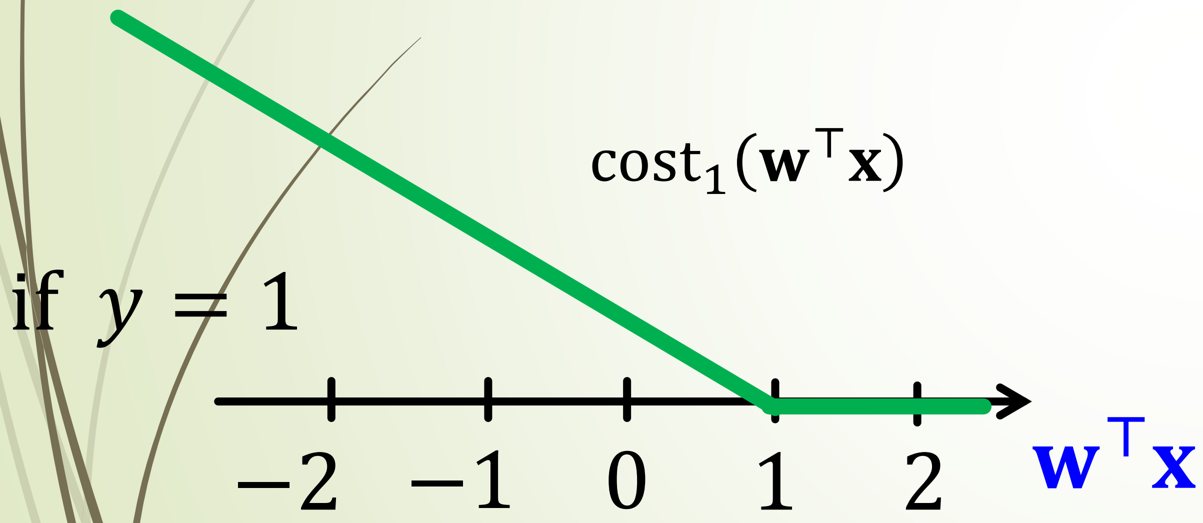
Slide credit: Andrew Ng

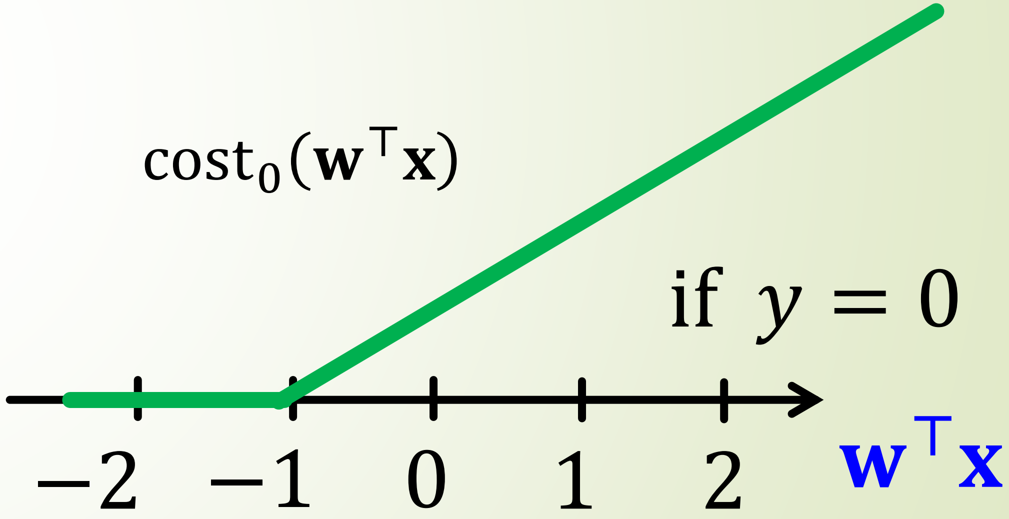
Support Vector Machine

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

Support Vector Machine

$$\min_{\mathbf{w}} C\left[\sum_{i=1}^{N} y^{(i)}\ \mathrm{cost}_1\left(\mathbf{w}^\top \mathbf{x}^{(i)}\right) + \left(1 - y^{(i)}\right) \mathrm{cost}_0\left(\mathbf{w}^\top \mathbf{x}^{(i)}\right)\right] + \frac{1}{2}\sum_{j=1}^{D} w_j^2$$





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

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

Slide credit: Andrew Ng

SVM Decision Boundary

$$\min_{\mathbf{w}} C\left[\sum_{i=1}^{N} y^{(i)}\ \mathrm{cost}_1\left(\mathbf{w}^\top \mathbf{x}^{(i)}\right) + (1-y^{(i)})\ \mathrm{cost}_0\left(\mathbf{w}^\top \mathbf{x}^{(i)}\right)\right] + \frac{1}{2}\sum_{j=1}^{D} w_j^2$$

- Let's say we have a very large C ...

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

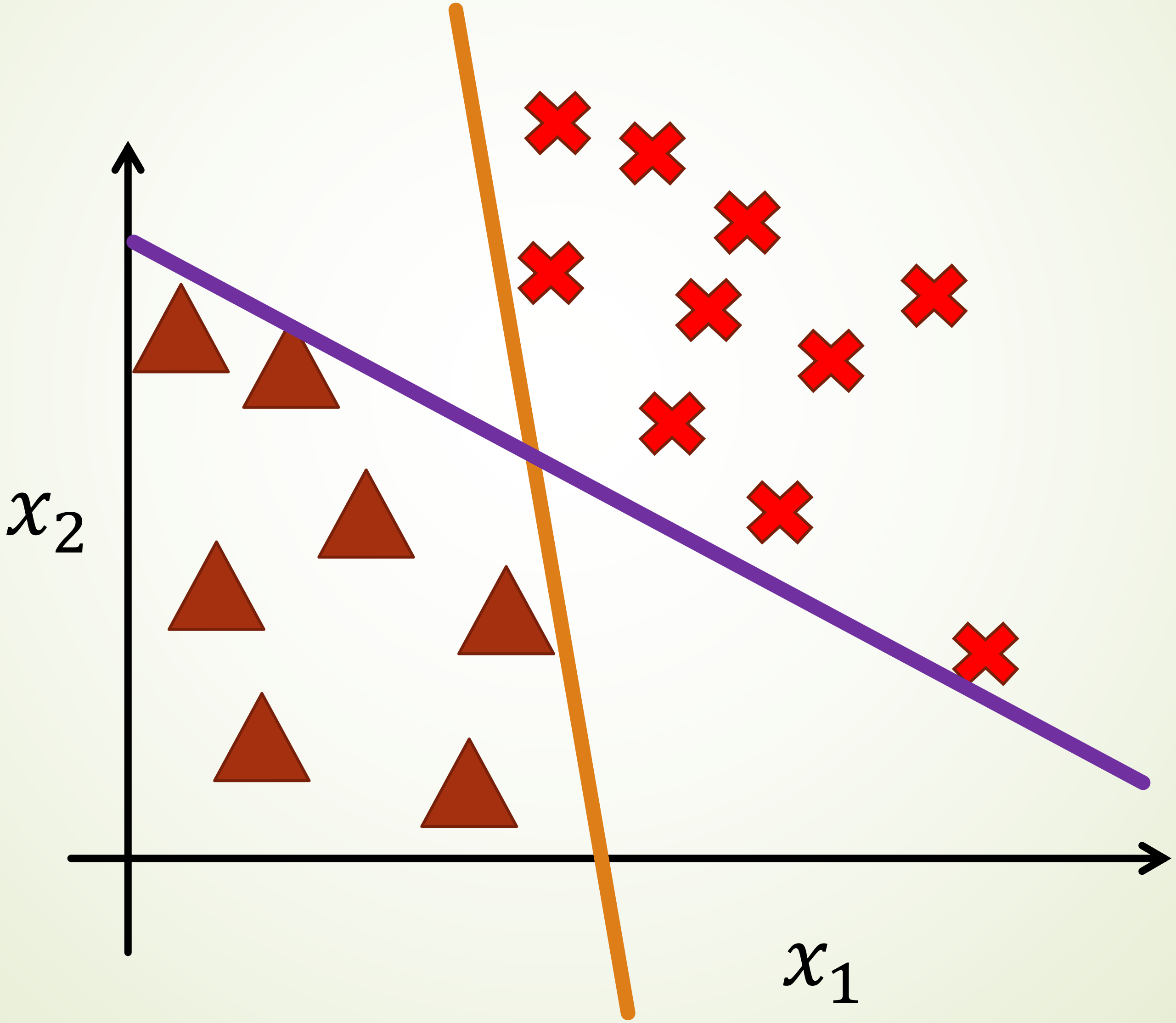
  $\mathbf{w}^\top \mathbf{x}^{(i)} \geq 1$

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

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

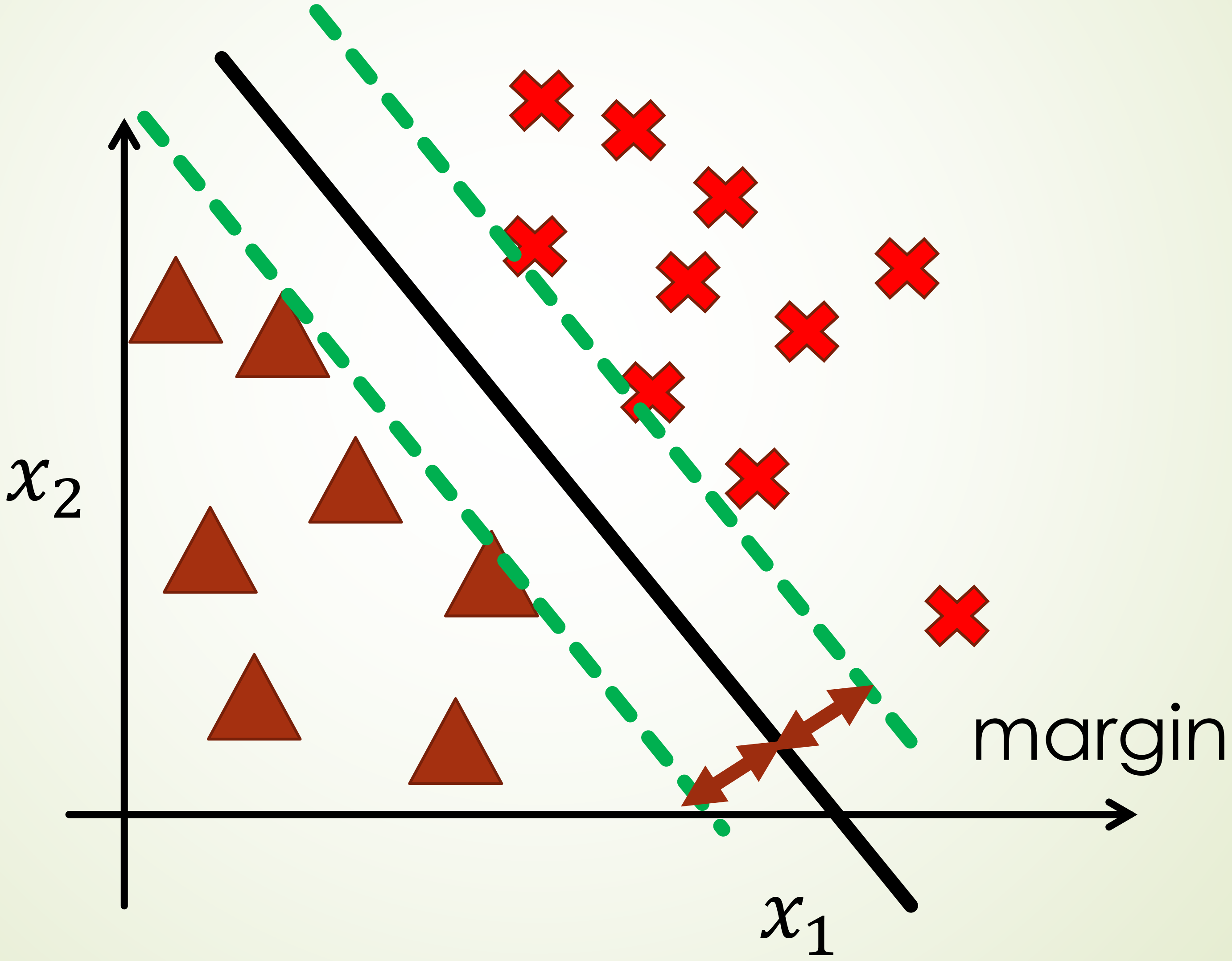
$$\boxed{\begin{aligned} &\min_{\mathbf{w}} \frac{1}{2}\textstyle\sum_{j=1}^{D} w_j^2 \\ &\quad \text{s.t.} \quad \mathbf{w}^\top \mathbf{x}^{(i)} \geq 1 \quad \text{if} \quad y^{(i)} = 1 \\ &\qquad\quad\ \ \mathbf{w}^\top \mathbf{x}^{(i)} \leq -1 \ \text{if} \quad y^{(i)} = 0 \end{aligned}}$$

SVM Decision Boundary: Linearly Separable Case



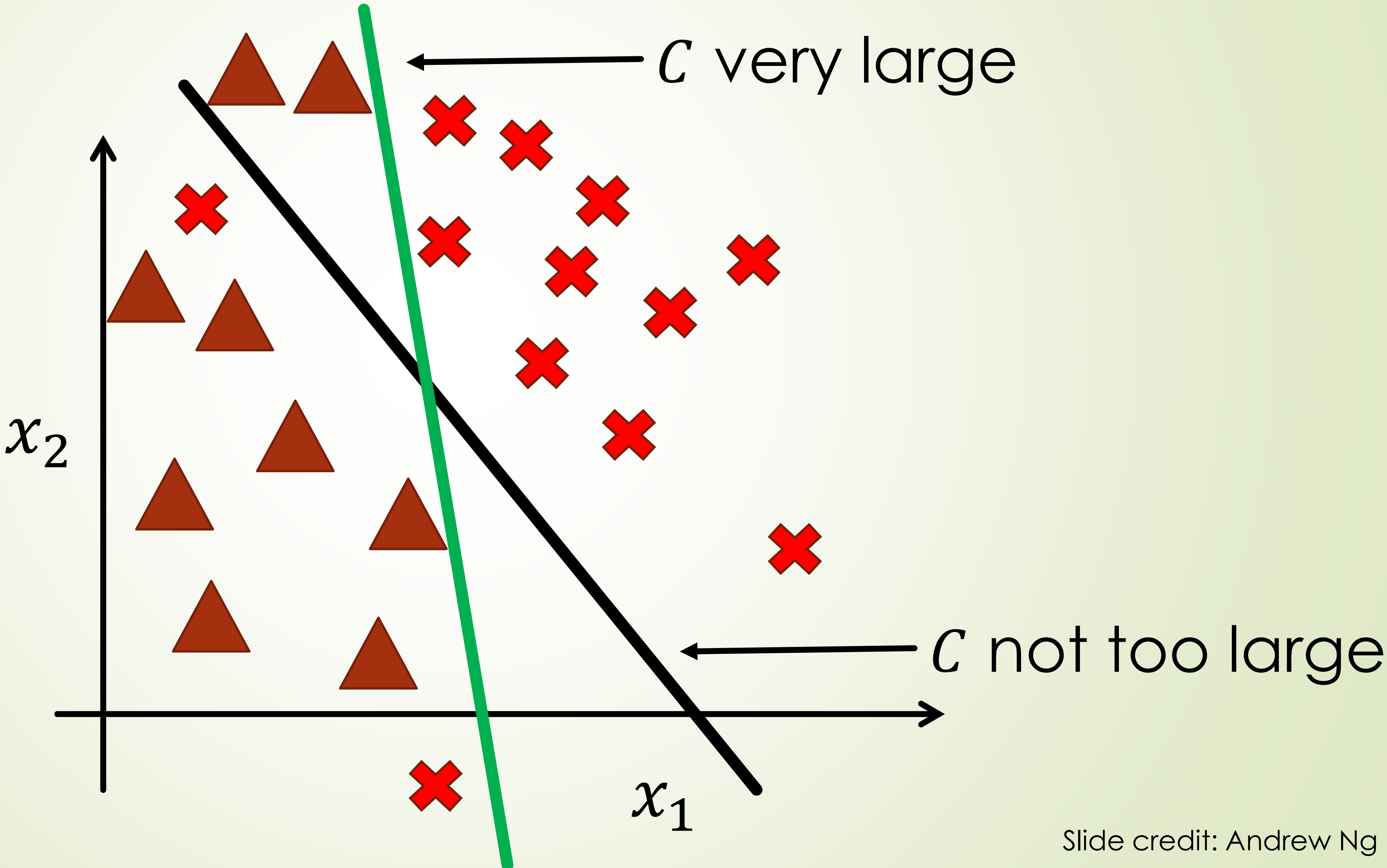
Slide credit: Andrew Ng

SVM Decision Boundary: Linearly Separable Case



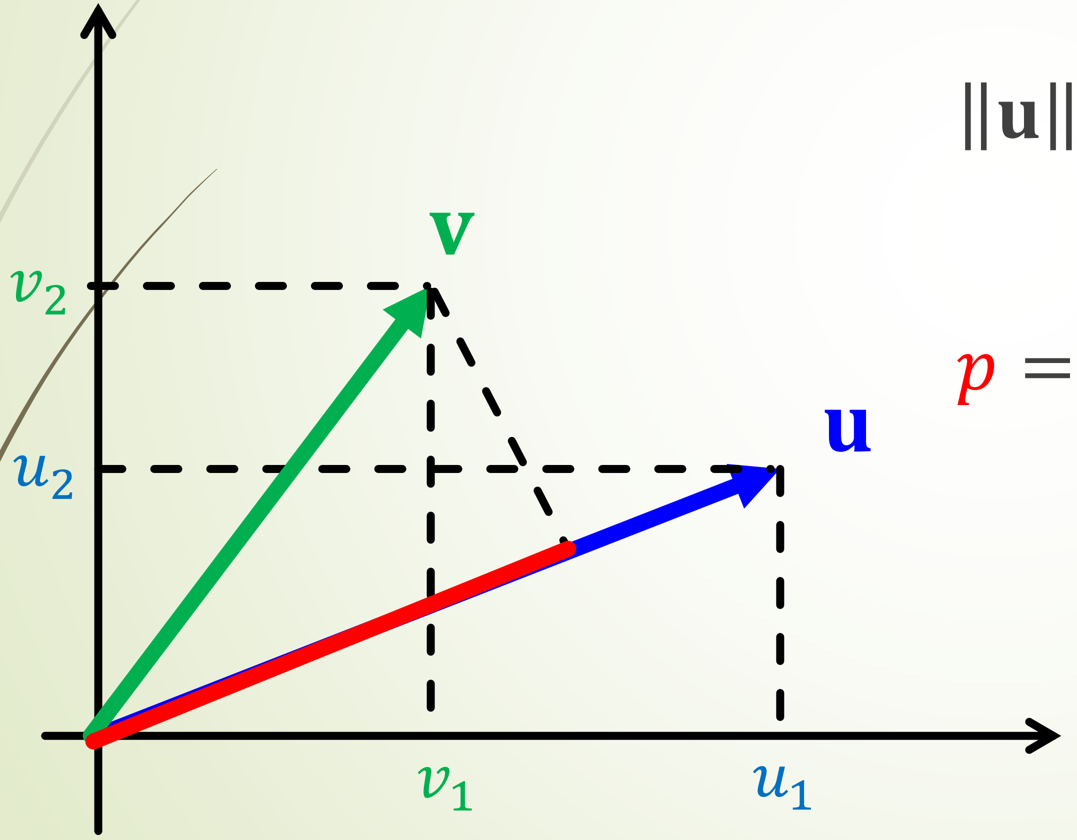
Slide credit: Andrew Ng

Large Margin Classifier in the Presence of Outlier



Slide credit: Andrew Ng

Vector Inner Product



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

$\|\mathbf{u}\|$ = length of vector $\mathbf{u}$

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

p = length of projection of $\mathbf{v}$ onto $\mathbf{u}$

$$\boxed{\begin{aligned} \mathbf{u}^{\top}\mathbf{v} &= p \cdot \|\mathbf{u}\| \\ &= \mathrm{u}_1 \mathrm{v}_1 + \mathrm{u}_2 \mathrm{v}_2 \end{aligned}}$$

SVM Decision Boundary

$$\frac{1}{2}\sum_{j=1}^{D} w_j^2 = \frac{1}{2}\left(w_1^2 + w_2^2\right) = \frac{1}{2}\left(\sqrt{w_1^2 + w_2^2}\right)^2 = \frac{1}{2}\|\mathbf{w}\|^2$$

$$\min_{\mathbf{w}} \frac{1}{2}\sum_{j=1}^{D} w_j^2$$

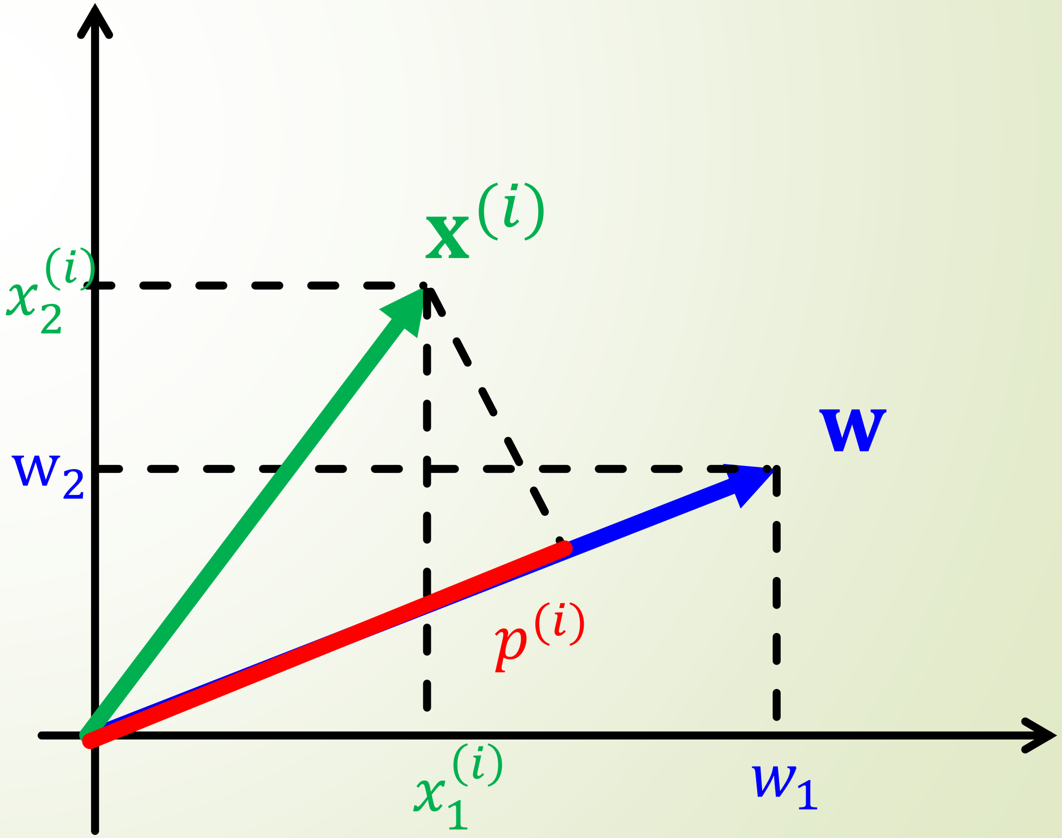
$$\text{s.t.} \quad \mathbf{w}^\top \mathbf{x}^{(i)} \geq 1 \quad \text{if} \quad \mathrm{y}^{(i)} = 1$$

$$\mathbf{w}^\top \mathbf{x}^{(i)} \leq -1 \quad \text{if} \quad \mathrm{y}^{(i)} = 0$$

Simplification: $w_0 = 0, N = 2$

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

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



SVM Decision Boundary

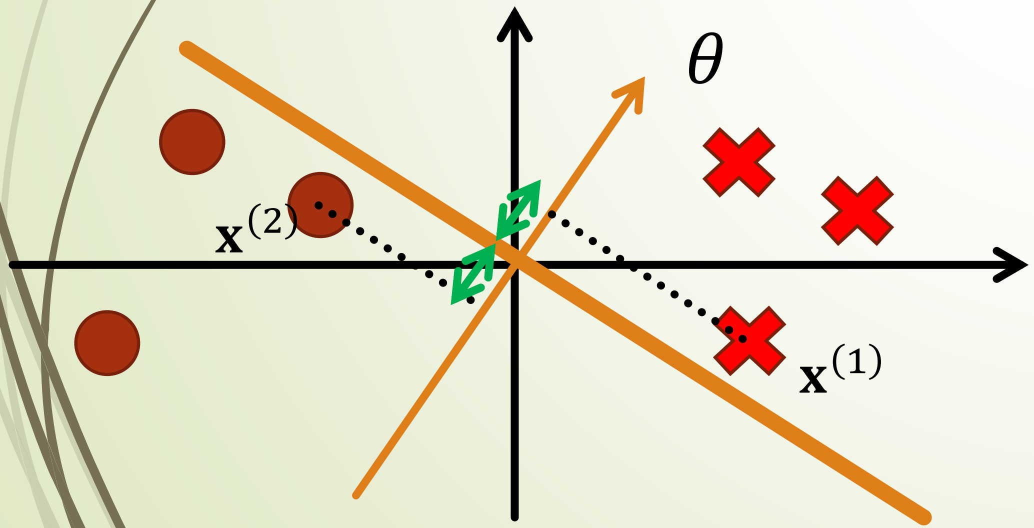
$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2$$

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

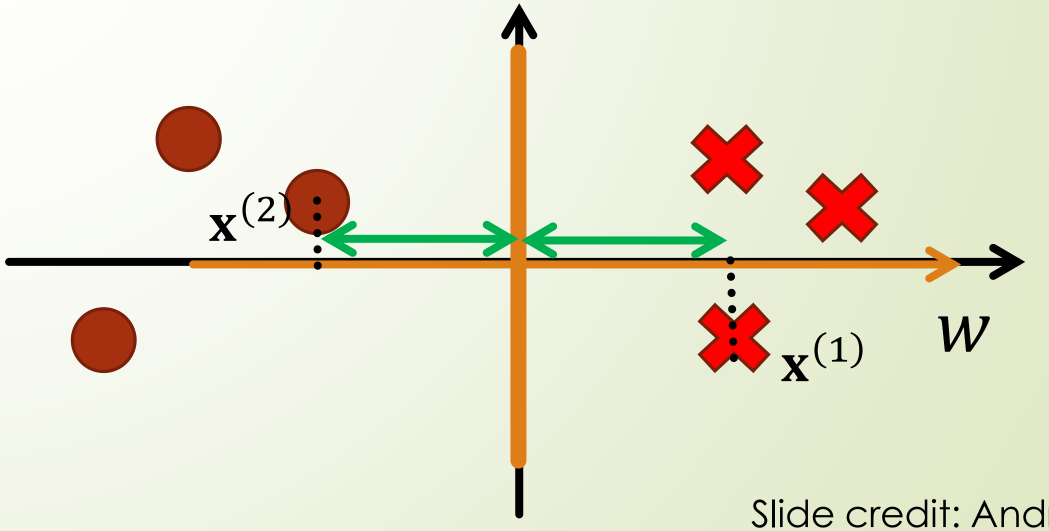
$$p^{(i)}\|\mathbf{w}\|^2 \leq -1 \quad \text{if} \quad y^{(i)} = 0$$

Simplification: $w_0 = 0, N = 2$

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

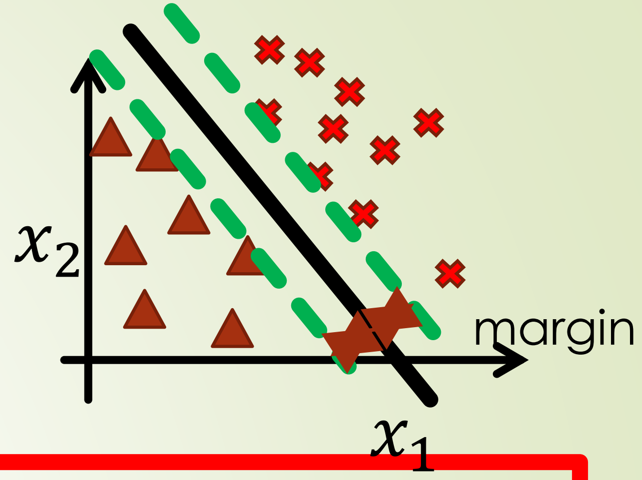


$p^{(1)}, p^{(2)}$ large $\rightarrow \|\mathbf{w}\|^2$ can be small



Slide credit: Andrew Ng

Rewrite the Formulation



Let $w_j = w'_j/\gamma$, with $\|\mathbf{w}'\|^2 = 1$

$$\min_{\mathbf{w}} \frac{1}{2}\sum_{j=1}^{D} w_j^2$$

$$\text{s.t.} \quad \mathbf{w}^\top \mathbf{x}^{(i)} \geq 1 \quad \text{if} \quad y^{(i)} = 1$$

$$\mathbf{w}^\top \mathbf{x}^{(i)} \leq -1 \quad \text{if} \quad y^{(i)} = 0$$

$$\max_{w',\gamma} \quad \gamma$$

$$\text{s.t.} \quad \|\mathbf{w}'\|^2 = 1$$

$$\mathbf{w}'^\top \mathbf{x}^{(i)} \geq \gamma \quad \text{if} \quad y^{(i)} = 1$$

$$\mathbf{w}'^\top \mathbf{x}^{(i)} \leq -\gamma \quad \text{if} \quad y^{(i)} = 0$$

$$\min_{\mathbf{w}} \frac{1}{2}\sum_{j=1}^{D} w_j^2 = \min_{w',\gamma} \frac{1}{2}\sum_{j=1}^{D} \frac{{w'_j}^2}{\gamma^2} = \max_{\gamma} \ \gamma$$

$$\because \sum_{j=1}^{D} {w'_j}^2 = 1$$

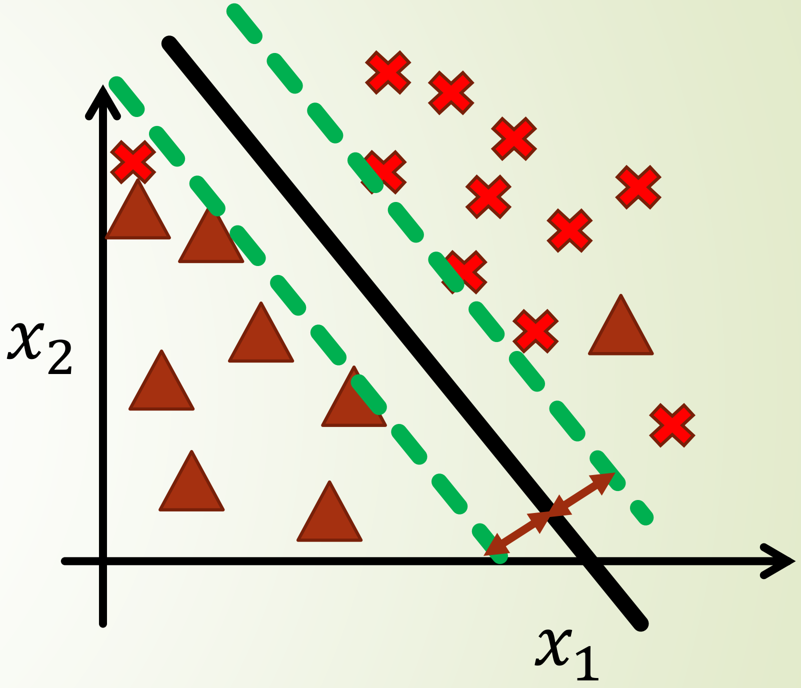
$$\mathbf{w}^\top \mathbf{x}^{(i)} \geq 1 \quad \text{if} \quad y^{(i)} = 1 \quad \rightarrow \quad \mathbf{w}'^\top \mathbf{x}^{(i)} \geq \gamma \quad \text{if} \quad y^{(i)} = 1$$

$$\mathbf{w}^\top \mathbf{x}^{(i)} \leq -1 \quad \text{if} \quad y^{(i)} = 0 \quad \rightarrow \quad \mathbf{w}'^\top \mathbf{x}^{(i)} \leq -\gamma \quad \text{if} \quad y^{(i)} = 0$$

Data Not Linearly Separable?

$$\min_{\mathbf{w}} \frac{1}{2}\sum_{j=1}^{D} w_j^2$$

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



$$\min_{\mathbf{w}} \frac{1}{2}\sum_{j=1}^{D} w_j^2 + C(\#\text{misclassification})$$

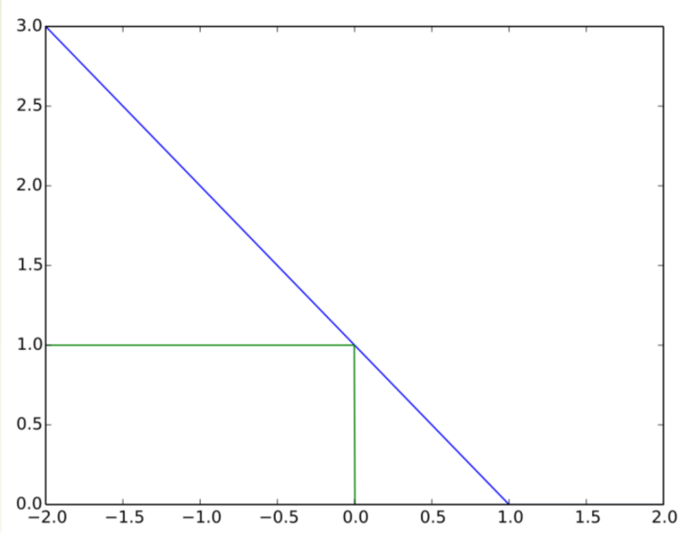
$$\text{s.t.} \quad \mathbf{w}^\top \mathbf{x}^{(i)} \geq 1 \quad \text{if} \quad y^{(i)} = 1$$
$$\mathbf{w}^\top \mathbf{x}^{(i)} \leq -1 \quad \text{if} \quad y^{(i)} = 0$$

NP-hard ☹

Convex Relaxation

$$\min_{\mathbf{w}} \frac{1}{2}\sum_{j=1}^{D} w_j^2 + C(\#\text{misclassification})$$

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

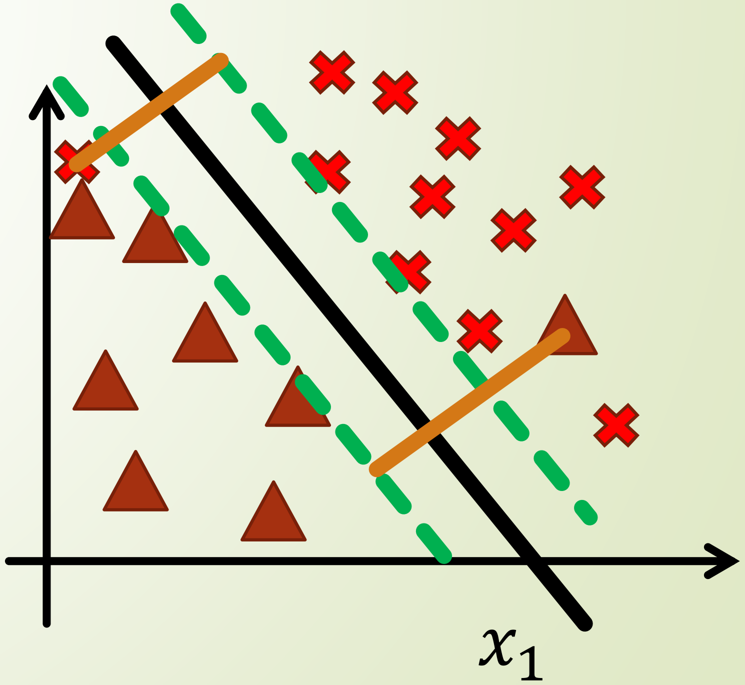


NP-hard ☹

$\xi^{(i)}$: slack variables

$$\min_{\mathbf{w}} \frac{1}{2}\sum_{j=1}^{D} w_j^2 + C\sum_i \xi^{(i)}$$

$$\text{s.t.} \quad \mathbf{w}^\top \mathbf{x}^{(i)} \geq 1 - \xi^{(i)} \quad \text{if} \quad y^{(i)} = 1$$
$$\mathbf{w}^\top \mathbf{x}^{(i)} \leq -1 + \xi^{(i)} \quad \text{if} \quad y^{(i)} = 0$$
$$\xi^{(i)} \geq 0 \quad \forall\, i$$



Hinge Loss

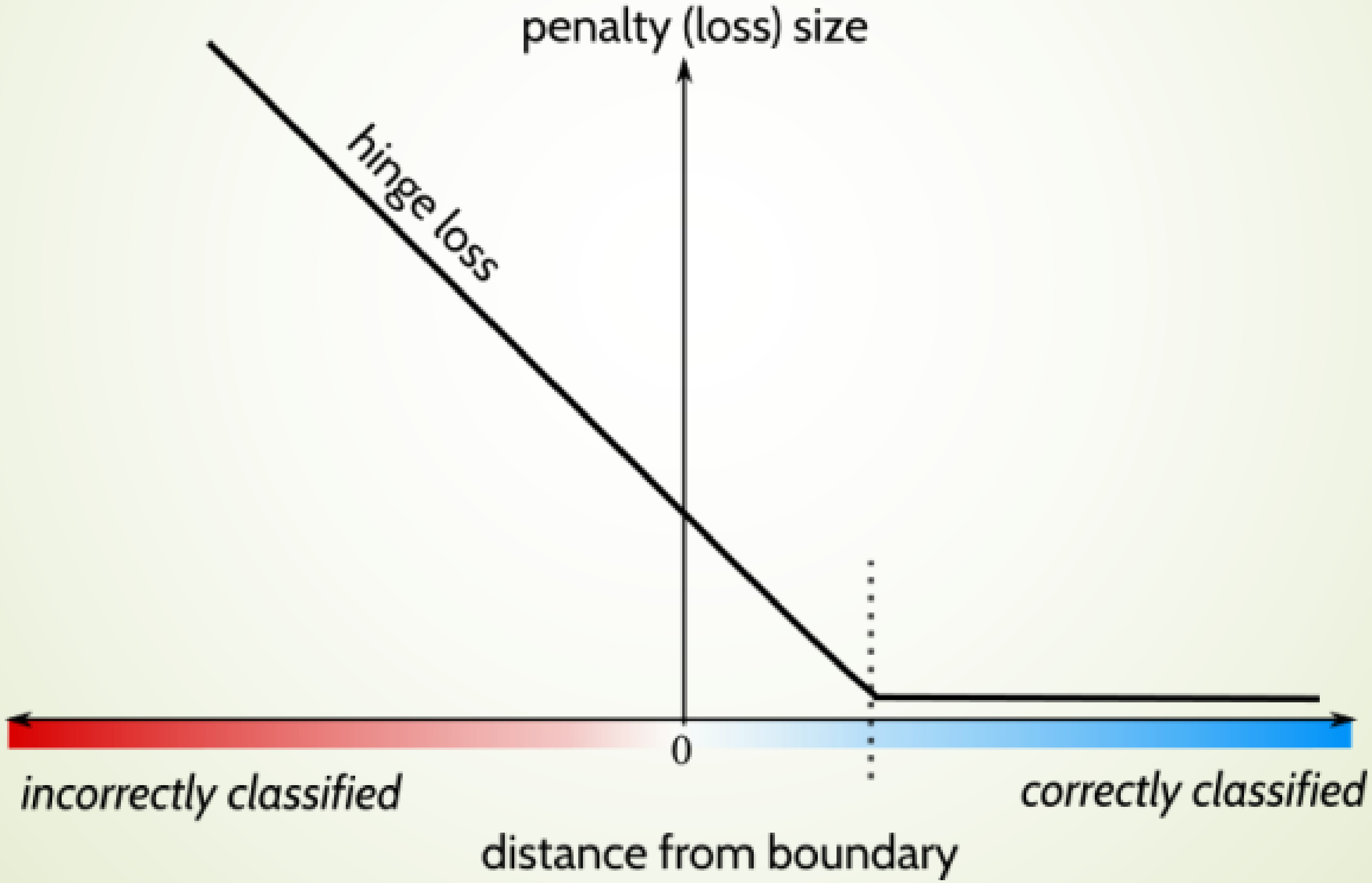


Image credit: https://math.stackexchange.com/questions/782586/how-do-you-minimize-hinge-loss

SVM Formulations

Hard-margin SVM formulation

$$\min_{\mathbf{w}} \frac{1}{2}\sum_{j=1}^{D} w_j^2$$

$$\text{s.t.} \quad \mathbf{w}^\top \mathbf{x}^{(i)} \geq 1 \quad \text{if} \quad y^{(i)} = 1$$

$$\mathbf{w}^\top \mathbf{x}^{(i)} \leq -1 \quad \text{if} \quad y^{(i)} = 0$$

Soft-margin SVM formulation

$$\min_{\mathbf{w}} \frac{1}{2}\sum_{j=1}^{D} w_j^2 + C\sum_i \xi^{(i)}$$

$$\text{s.t.} \quad \mathbf{w}^\top \mathbf{x}^{(i)} \geq \ 1 - \xi^{(i)} \quad \text{if} \quad y^{(i)} = 1$$

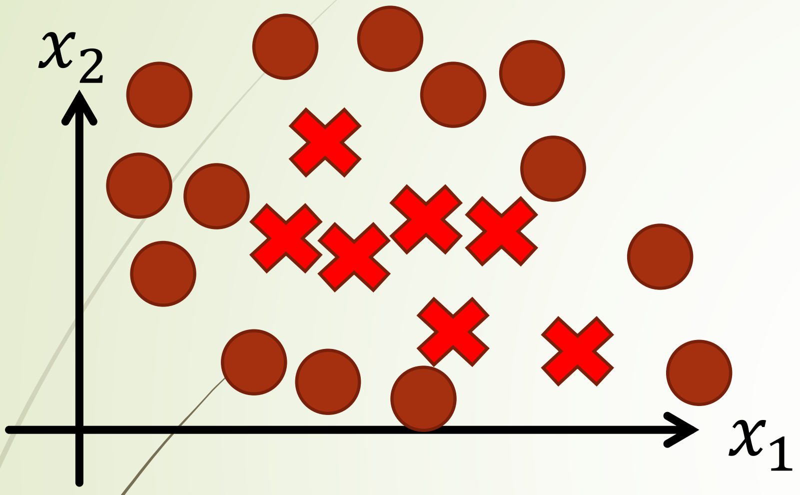
$$\mathbf{w}^\top \mathbf{x}^{(i)} \leq -1 + \xi^{(i)} \quad \text{if} \quad y^{(i)} = 0$$

$$\xi^{(i)} \geq 0 \qquad \forall\, i$$

Support Vector Machine

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

Non-Linear Decision Boundary



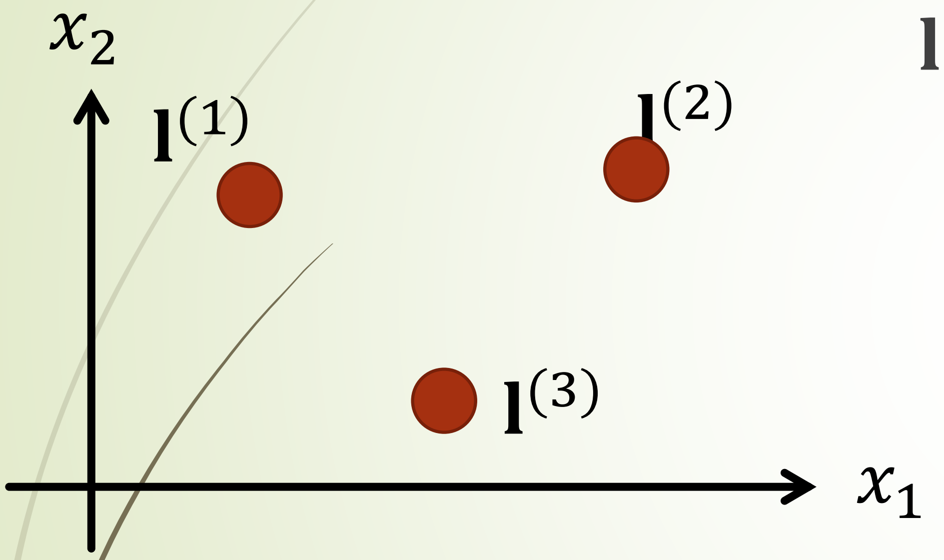
Predict $y = 1$ if

$$w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1 x_2 + w_4 x_1^2 + w_5 x_2^2 + \cdots \geq 0$$

$$w_0 + w_1 f_1 + w_2 f_2 + w_3 f_3 + \cdots$$
$$f_1 = x_1, f_2 = x_2, f_3 = x_1 x_2, \cdots$$

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

Kernel

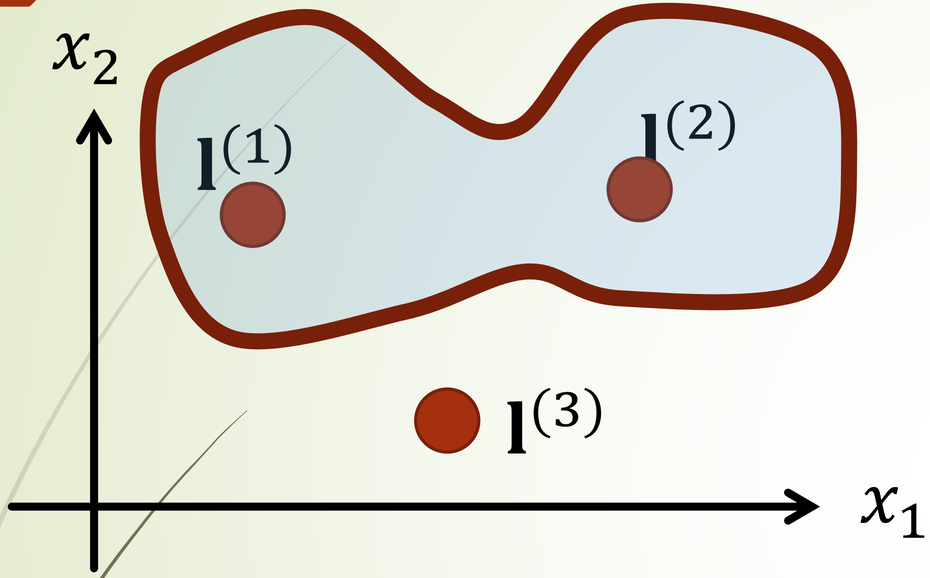


Give $\mathbf{x}$, compute new features depending on proximity to landmarks $\mathbf{l}^{(1)}, \mathbf{l}^{(2)}, \mathbf{l}^{(3)}$

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

Gaussian kernel

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



Predict $y = 1$ if

$$w_0 + w_1 f_1 + w_2 f_2 + w_3 f_3 \geq 0$$

Ex: $w_0 = -0.5, w_1 = 1, w_2 = 1, w_3 = 0$

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

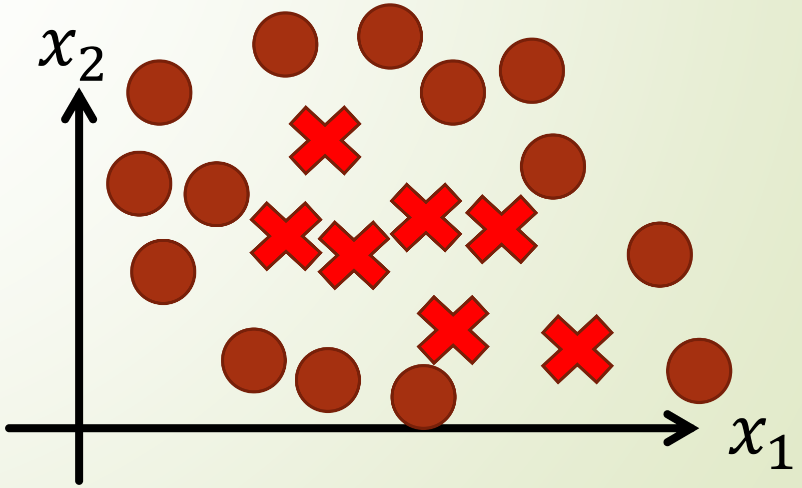
Choosing the Landmarks

- Given $\mathbf{x}$

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

Predict $y = 1$ if $w_0 + w_1 f_1 + w_2 f_2 + w_3 f_3 \geq 0$

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



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

SVM with Kernels

- Given $(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \cdots, (\mathbf{x}^{(N)}, y^{(N)})$
- Choose $\mathbf{l}^{(1)} = \mathbf{x}^{(1)}, \mathbf{l}^{(2)} = \mathbf{x}^{(2)}, \mathbf{l}^{(3)} = \mathbf{x}^{(3)}, \cdots, \mathbf{l}^{(N)} = \mathbf{x}^{(N)}$
- Given example $\mathbf{x}$:
  - $f_1 = \text{similarity}(\mathbf{x}, \mathbf{l}^{(1)})$
  - $f_2 = \text{similarity}(\mathbf{x}, \mathbf{l}^{(2)})$
  - …

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

- For training example $(\mathbf{x}^{(i)}, y^{(i)})$:
  - $\mathbf{x}^{(i)} \to \mathbf{f}^{(i)}$

Slide credit: Andrew Ng

SVM with Kernels

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

- **Training (original)**

$$\min_{\mathbf{w}} C \left[\sum_{i=1}^{N} y^{(i)} \ \mathrm{cost}_1\left(\mathbf{w}^\top \mathbf{x}^{(i)}\right) + \left(1 - y^{(i)}\right) \mathrm{cost}_0\left(\mathbf{w}^\top \mathbf{x}^{(i)}\right) \right] + \frac{1}{2} \sum_{j=1}^{D} w_j^2$$

- **Training (with kernel)**

$$\min_{\mathbf{w}} C \left[\sum_{i=1}^{N} y^{(i)} \ \mathrm{cost}_1\left(\mathbf{w}^\top \mathbf{f}^{(i)}\right) + \left(1 - y^{(i)}\right) \mathrm{cost}_0\left(\mathbf{w}^\top \mathbf{f}^{(i)}\right) \right] + \frac{1}{2} \sum_{j=1}^{D} w_j^2$$

Support Vector Machines (Primal/Dual)

- Primal form

$$\min_{\mathbf{w},\xi^{(i)}} \frac{1}{2}\sum_{j=1}^{D} w_j^2 + C\sum_i \xi^{(i)}$$

$$\begin{aligned} \text{s.t.} \quad & \mathbf{w}^\top \mathbf{x}^{(i)} \geq 1 - \xi^{(i)} \quad && \text{if} \quad y^{(i)} = 1 \\ & \mathbf{w}^\top \mathbf{x}^{(i)} \leq -1 + \xi^{(i)} && \text{if} \quad y^{(i)} = 0 \\ & \xi^{(i)} \geq 0 \qquad \forall\, i \end{aligned}$$

- Lagrangian dual form

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\sum_i \sum_j y^{(i)} y^{(j)} \alpha^{(i)} \alpha^{(j)} \mathbf{x}^{(i)\top} \mathbf{x}^{(j)} - \sum_i \alpha^{(i)}$$

$$\begin{aligned} \text{s.t.} \quad & 0 \leq \alpha^{(i)} \leq C_i \\ & \sum_i y^{(i)} \alpha^{(i)} = 0 \end{aligned}$$

SVM (Lagrangian Dual)

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\sum_i \sum_j y^{(i)} y^{(j)} \alpha^{(i)} \alpha^{(j)} \boxed{\mathbf{x}^{(i)\top}\mathbf{x}^{(j)}} - \sum_i \alpha^{(i)}$$

$$\text{s.t.} \quad 0 \le \alpha^{(i)} \le C_i$$

$$\sum_i y^{(i)} \alpha^{(i)} = 0$$

Replace $\mathbf{x}^{(i)\top}\mathbf{x}^{(j)}$ with $k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$

Classifier: $\mathbf{w} = \sum_i \alpha^{(i)} y^{(i)} \mathbf{w}^{(i)}$

- The points $\mathbf{x}^{(i)}$ for which $\alpha^{(i)} \neq 0 \rightarrow$ Support Vectors

SVM parameters

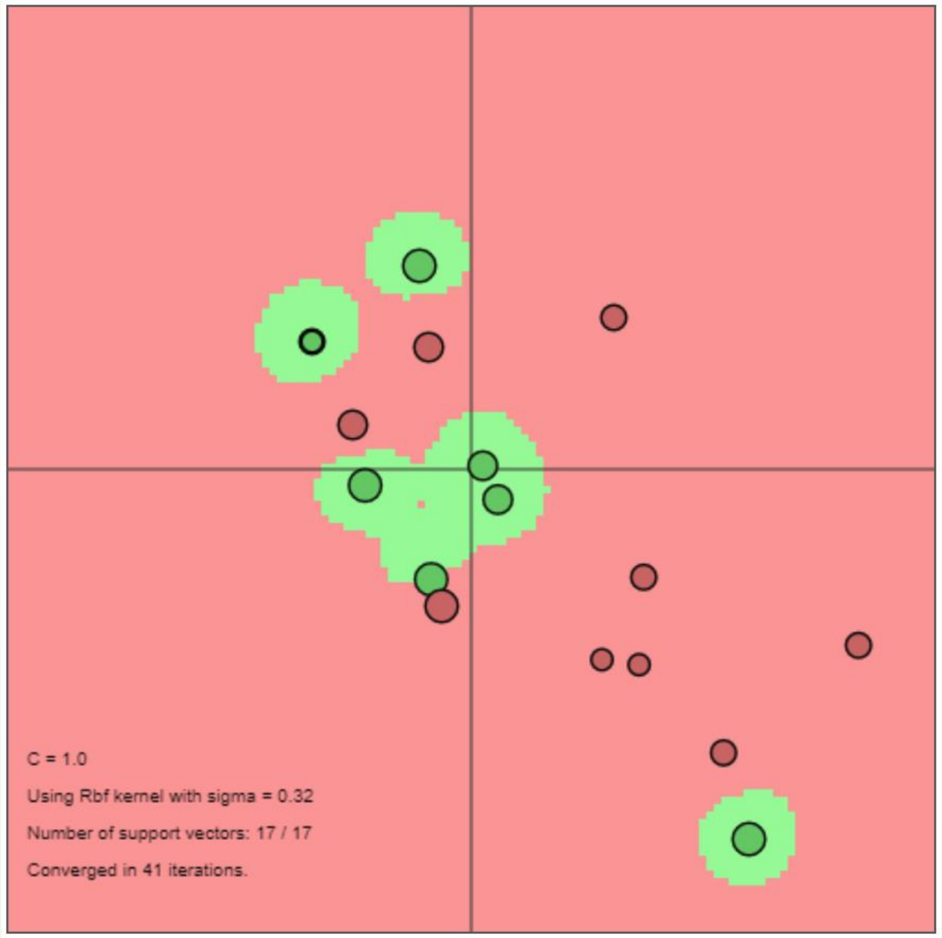
- $C \left(= \frac{1}{\lambda}\right)$
  Large C: Lower bias, high variance.
  Small C: Higher bias, low variance.

- σ^2
- Large σ^2: features $\mathbf{f}_i$ vary more smoothly.
  - Higher bias, lower variance
- Small σ^2: features $\mathbf{f}_i$ vary less smoothly.
  - Lower bias, higher variance

Slide credit: Andrew Ng

SVM Demo

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



Support Vector Machine

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

Using SVM

- SVM software package (e.g., liblinear, libsvm) to solve for $\mathbf{w}$
- Need to specify:
  - Choice of parameter C.
  - Choice of kernel (similarity function):
- Linear kernel: Predict $y = 1 \quad \text{if} \quad \mathbf{w}^\top \mathbf{x} \geq 0$
- Gaussian kernel:
  - $f_i = \exp(-\frac{\|\mathbf{x}-\mathbf{l}^{(i)}\|^2}{2\sigma^2})$, where $\mathbf{l}^{(i)} = \mathbf{x}^{(i)}$
  - Need to choose σ^2. Need proper feature scaling

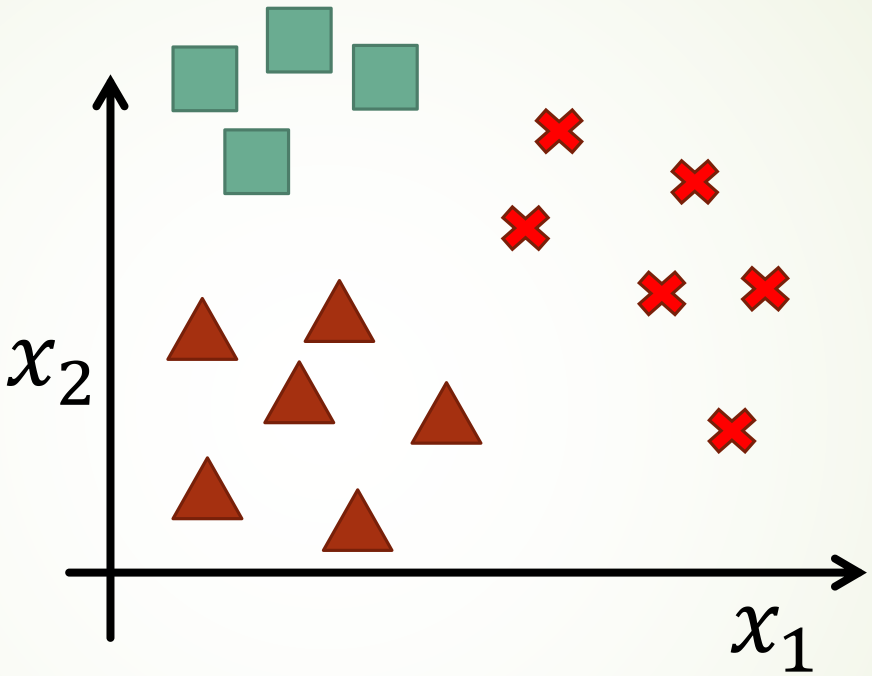
Kernel (Similarity) Functions

- Note: not all similarity functions make valid kernels.

- Many off-the-shelf kernels 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 $\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \dots, \mathbf{w}^{(K)}$
- Pick class i with the largest $\mathbf{w}^{(i)^\top}\mathbf{x}$

Logistic Regression vs. SVMs

- D = # of features ($\mathbf{x} \in \mathbb{R}^{D+1}$), N = # of training examples

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

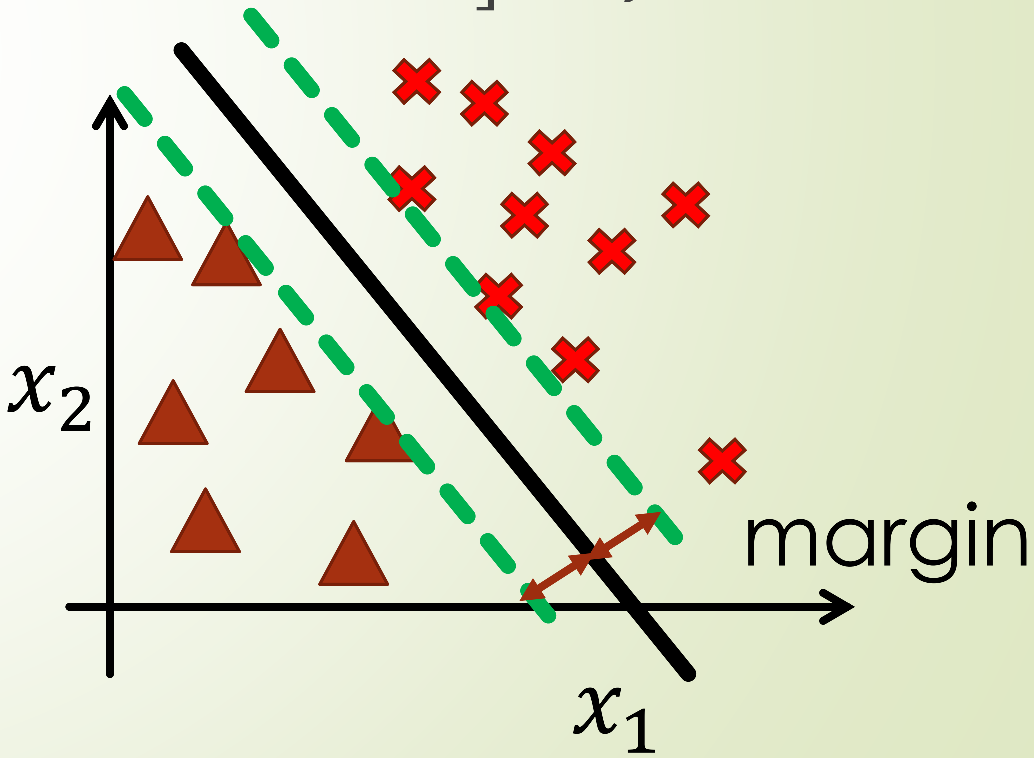
Neural network likely to work well for most of these cases, but slower to train

Things to Remember

- **Cost function**

$$\min_{\mathbf{w}} C\left[\sum_{i=1}^{N} y^{(i)}\ \mathrm{cost}_1\left(\mathbf{w}^\top \mathbf{f}^{(i)}\right) + (1-y^{(i)})\ \mathrm{cost}_0\left(\mathbf{w}^\top \mathbf{f}^{(i)}\right)\right] + \frac{1}{2}\sum_{j=1}^{D} w_j^2$$

- **Large margin classification**
- **Kernels**
- **Using SVM**



Support Vector Regression (SVR)

- Goal: Extending SVMs to regression problems while at the same time preserving the property of sparseness

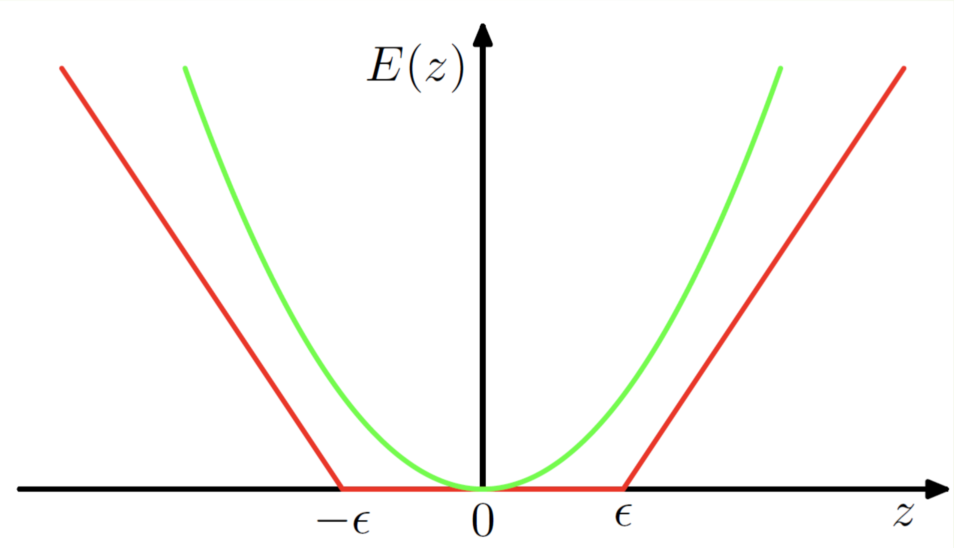
$$\min_{\mathbf{w}} \frac{1}{2} \sum_{i=1}^{N} \{y_n - t_n\}^2 + \frac{\lambda}{2} \|\mathbf{w}\|^2$$

- To obtain sparse solutions, the quadratic error function is replaced by an ϵ-insensitive error function which gives zero error $|y(\mathbf{x}) - t| < \epsilon$ where $\epsilon > \mathbf{0}$.

Support Vector Regression (SVR)

- A simple example of an ϵ-insensitive error function, having a linear cost associated with errors outside the insensitive region, is given by

$$E_\epsilon(y(\mathbf{x}) - t) = \begin{cases} 0, & \text{if } |y(\mathbf{x}) - t| < \epsilon \\ |y(\mathbf{x}) - t| - \epsilon, & \text{otherwise} \end{cases}$$



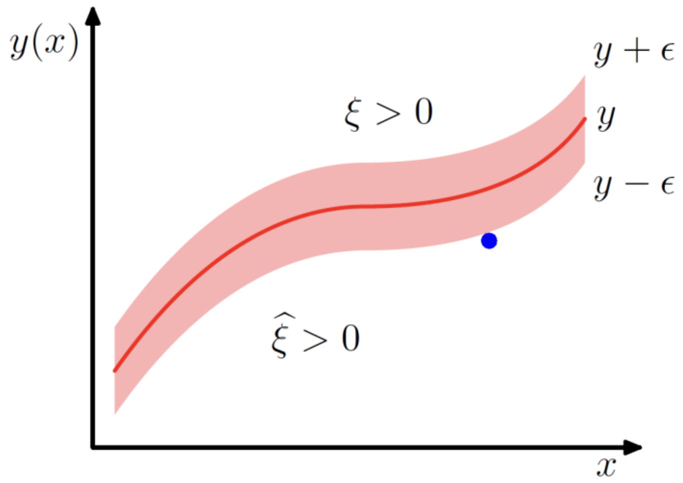
Support Vector Regression (SVR)

- We therefore minimize a regularized error function given by

$$\min_{\mathbf{w}} C \sum_{i=1}^{N} E_\epsilon(y(\mathbf{x}_n) - t_n) + \frac{1}{2}\|\mathbf{w}\|^2$$

- This introduces two slack variables $\xi_n \geq 0$ and $\hat{\xi}_n \geq 0$, where $\xi_n > 0$ corresponds to a point for which $t_n > y(\mathbf{x}_n) + \epsilon$ and $\hat{\xi}_n < 0$ corresponds to a point for which $t_n < y(\mathbf{x}_n) - \epsilon$

Illustration of SVM regression, showing the regression curve together with the ϵ-insensitive 'tube'. Also shown are examples of the slack variables ξ and $\widehat{\xi}$. Points above the ϵ-tube have $\xi > 0$ and $\widehat{\xi} = 0$, points below the ϵ-tube have $\xi = 0$ and $\widehat{\xi} > 0$, and points inside the ϵ-tube have $\xi = \widehat{\xi} = 0$.



Support Vector Regression (SVR)

- The error function for support vector regression can then be written as

$$\min_{\mathbf{w}} C \sum_{i=1}^{N} (\xi_n + \hat{\xi}_n) + \frac{1}{2} \|\mathbf{w}\|^2$$

subject to $\xi_n \geq 0$ and $\hat{\xi}_n \geq 0$ as well as

$$\begin{aligned} t_n &\leqslant y(\mathbf{x}_n) + \epsilon + \xi_n \\ t_n &\geqslant y(\mathbf{x}_n) - \epsilon - \widehat{\xi}_n. \end{aligned}$$

- Introducing Lagrange multipliers, we minimize the Lagrangian

$$\begin{aligned} L \;=\; & C \sum_{n=1}^{N} (\xi_n + \widehat{\xi}_n) + \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{n=1}^{N} (\mu_n \xi_n + \widehat{\mu}_n \widehat{\xi}_n) \\ & - \sum_{n=1}^{N} a_n (\epsilon + \xi_n + y_n - t_n) - \sum_{n=1}^{N} \widehat{a}_n (\epsilon + \widehat{\xi}_n - y_n + t_n) \end{aligned}$$