CS 599 Machine Learning
Lecture 9: Support Vector Machines

Hao Ji
Computer Science Department
Cal Poly Pomona

Courtesy of Dr. Shuiwang Ji at WSC, Dr. Sanja Fidler at UT, and Dr. Hung-yi Lee at NTU

SVM

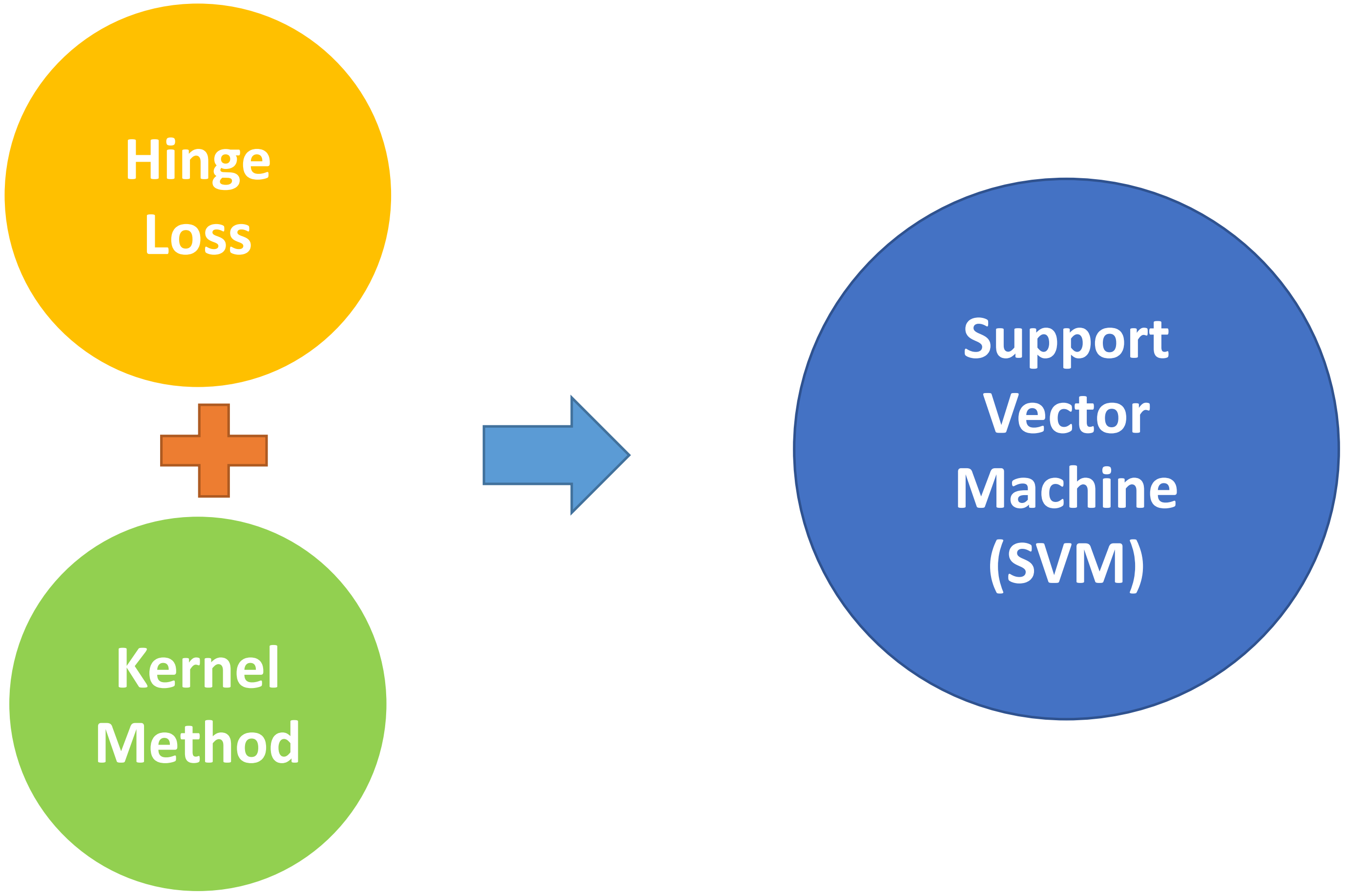
- SVM was inspired from Statistical Learning theory

- SVM was first introduced in 1992
  - Boser, Bernhard E., Isabelle M. Guyon, and Vladimir N. Vapnik. "A training algorithm for optimal margin classifiers." In *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 144-152. ACM, 1992.

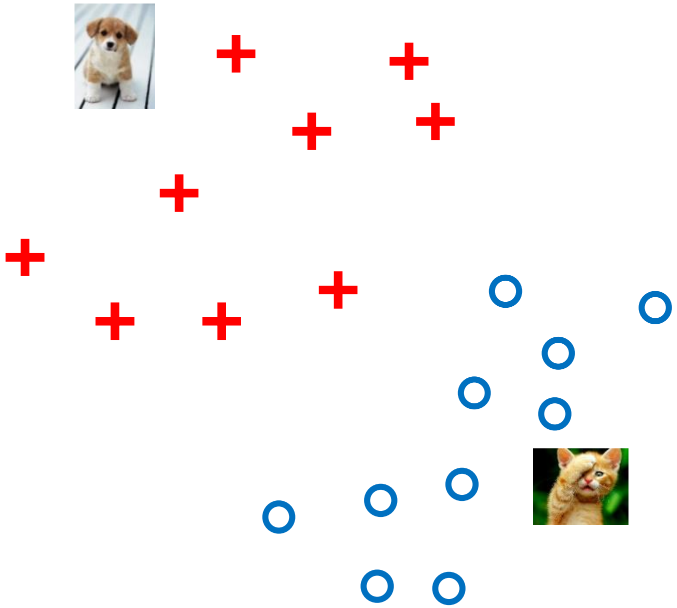
- SVM becomes popular because of its success in handwritten digit recognition
  - Bottou, Léon, Corinna Cortes, John S. Denker, Harris Drucker, Isabelle Guyon, Lawrence D. Jackel, Yann LeCun et al. "Comparison of classifier methods: a case study in handwritten digit recognition." In *Pattern Recognition, 1994. Vol. 2-Conference B: Computer Vision & Image Processing., Proceedings of the 12th IAPR International. Conference on*, vol. 2, pp. 77-82. IEEE, 1994.

SVM

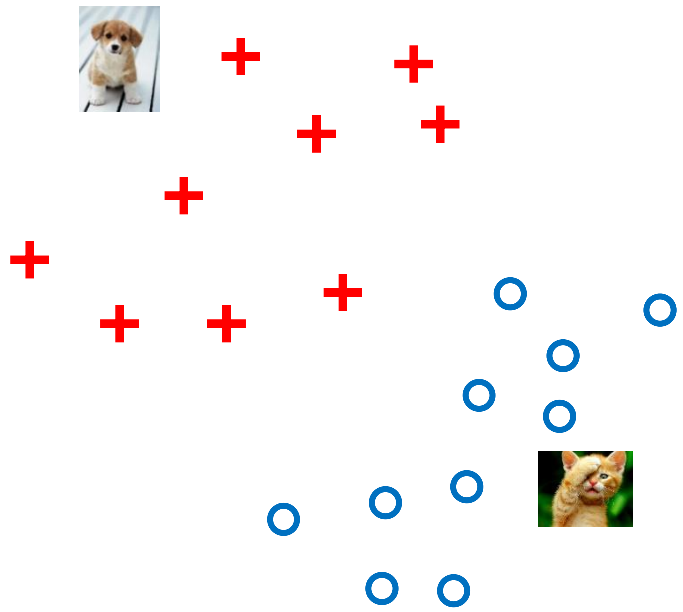
- Linear SVM
- Soft Margin
  - Hinge Loss
- Kernel Trick



Linear Classifier

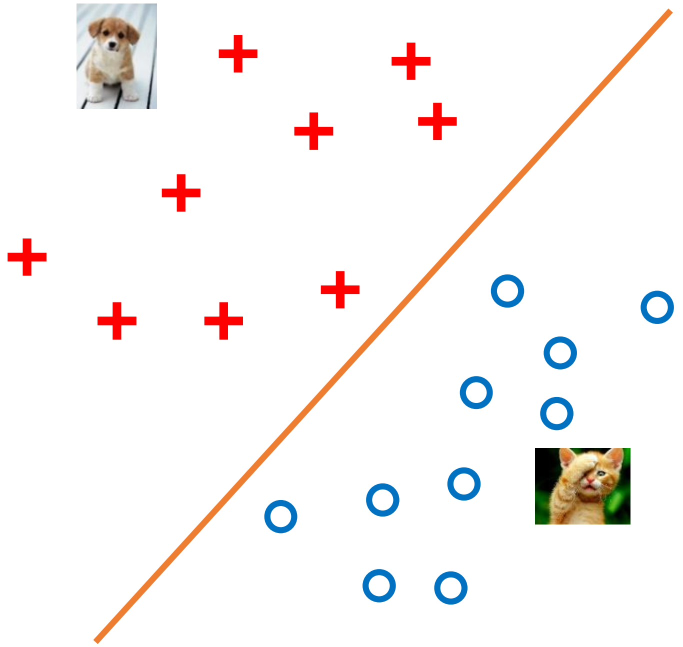


Linear Classifier



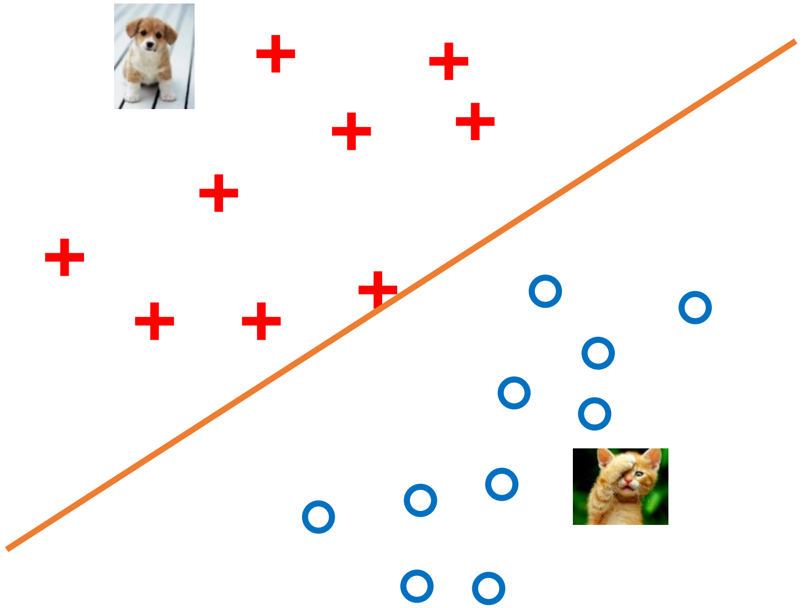
How would you classify this data?

Linear Classifier



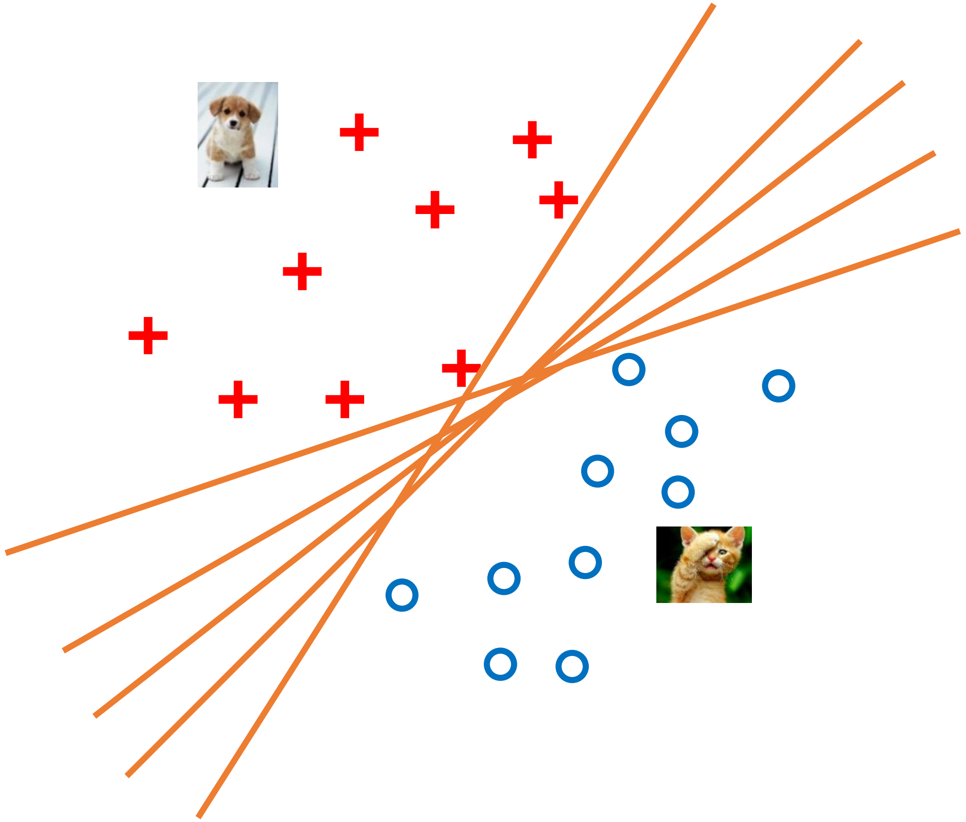
$$f(x) = \begin{cases} 1 & if\ w^T x + b \geq 0 \\ -1 & if\ w^T x + b < 0 \end{cases}$$

Linear Classifier



$$f(x) = \begin{cases} 1 & if\ w^T x + b \geq 0 \\ -1 & if\ w^T x + b < 0 \end{cases}$$

Linear Classifier

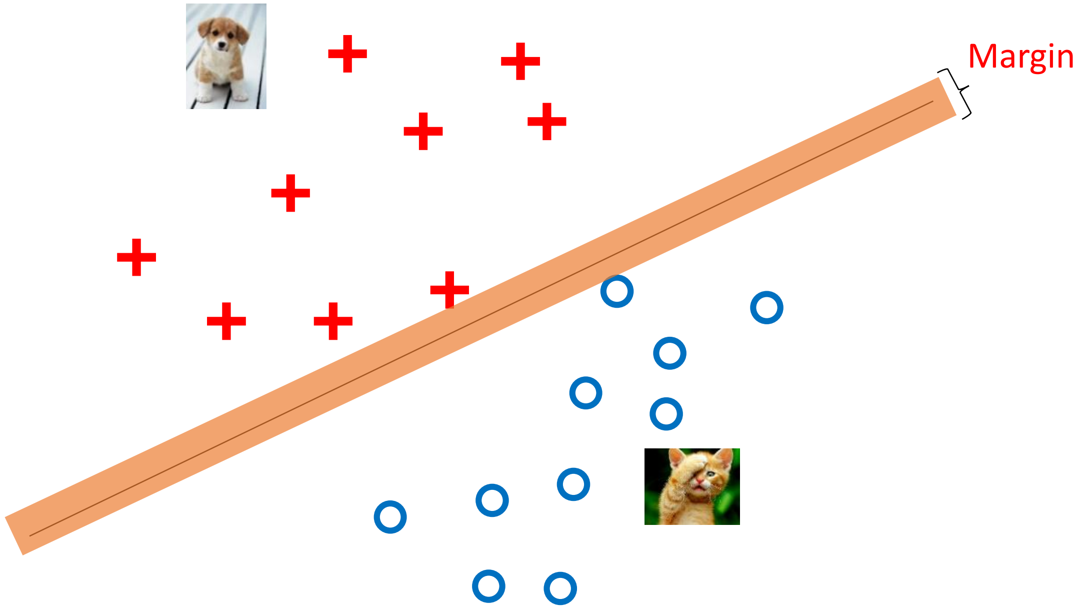


$$f(x) = \begin{cases} 1 & if\ w^T x + b \geq 0 \\ -1 & if\ w^T x + b < 0 \end{cases}$$

Any of these would be fine ...

But, which one is the best?

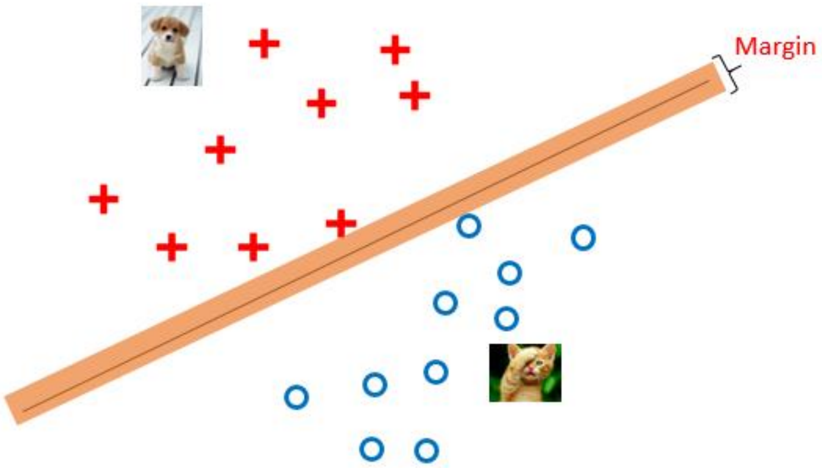
Linear Classifier



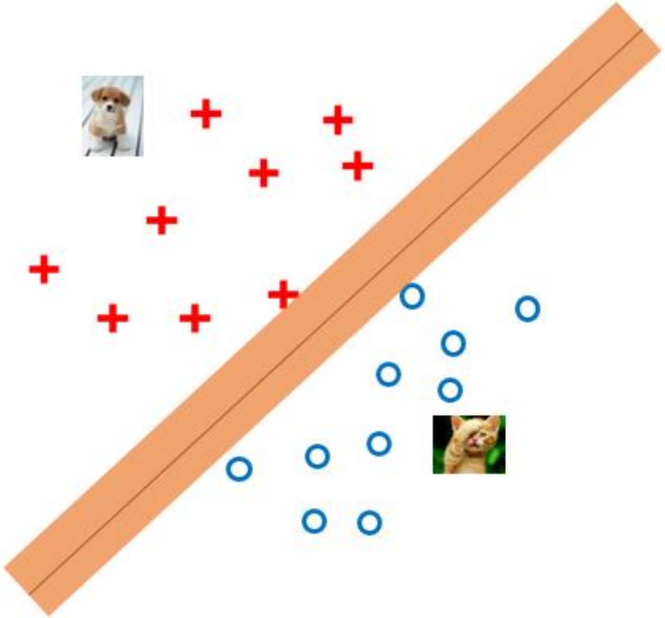
$$f(x) = \begin{cases} 1 & if\ w^T x + b \geq 0 \\ -1 & if\ w^T x + b < 0 \end{cases}$$

Margin: the width that the boundary could be increased by before hitting a datapoint

SVM



VS

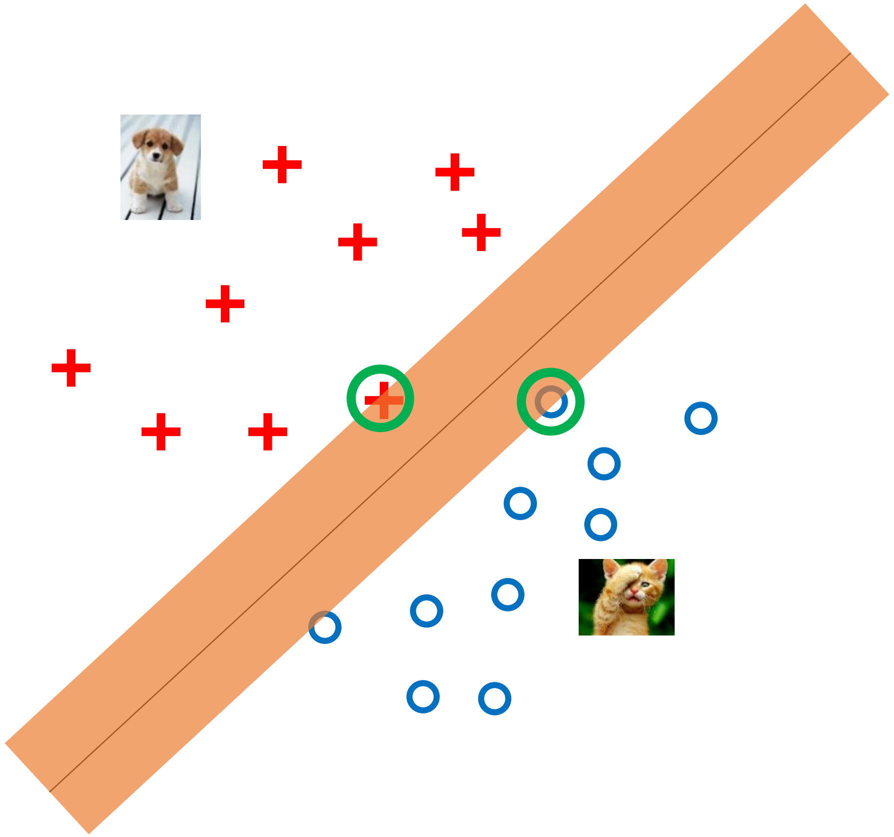


The maximum margin linear classifier: the linear classifier with the maximum margin

The simplest kind of SVM (called Linear SVM)

SVM

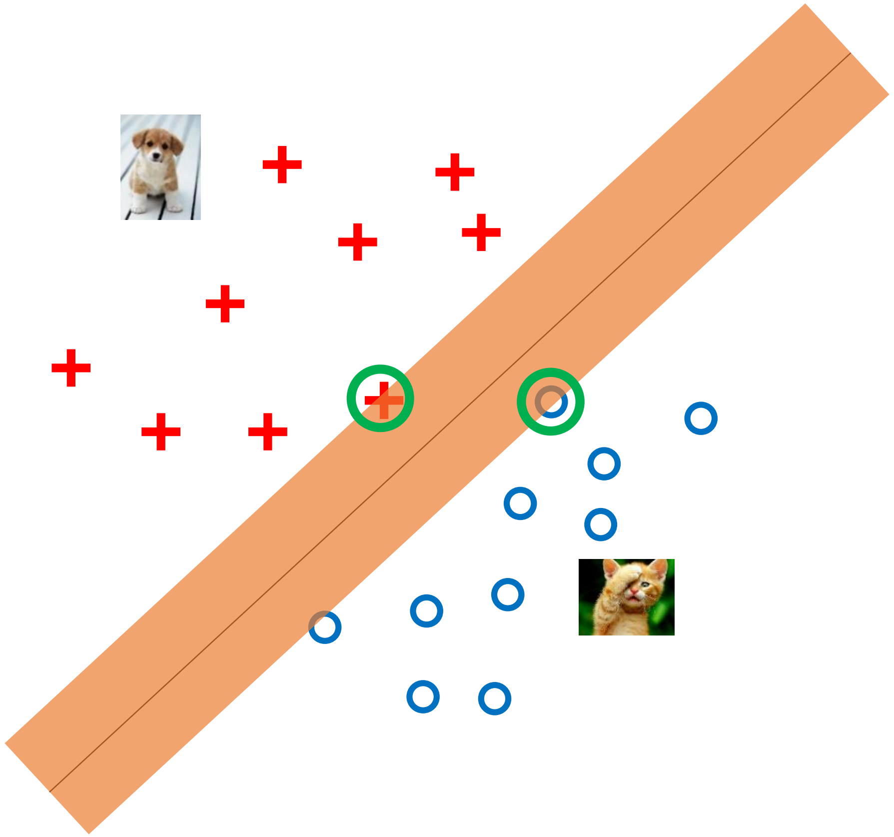
Support Vectors: the data points that the margin pushes up against



SVM

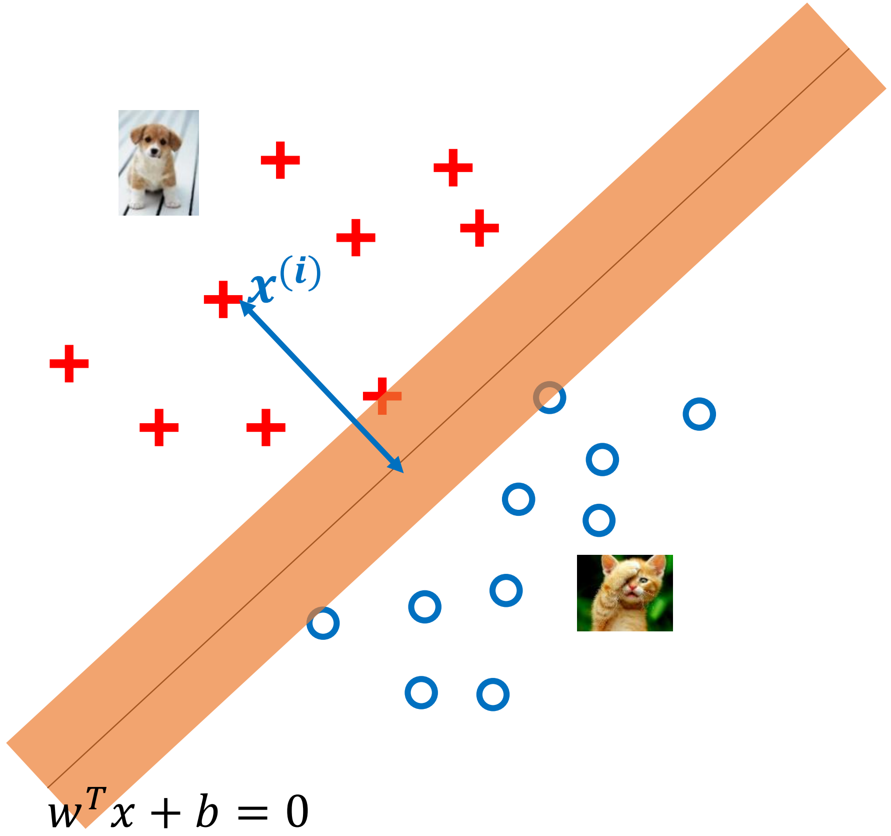
Support Vectors: the data points that the margin pushes up against

- Intuitively this feels safest
  - If we’ve made a small error in the location of the boundary, this gives us least chance of causing a misclassification
- The model is immune to removal of any non-support-vector datapoints
- There’re some theory (using VC dimension) that is related to (but not the same as) the proposition that this is a good thing
- Empirically it works well.



SVM

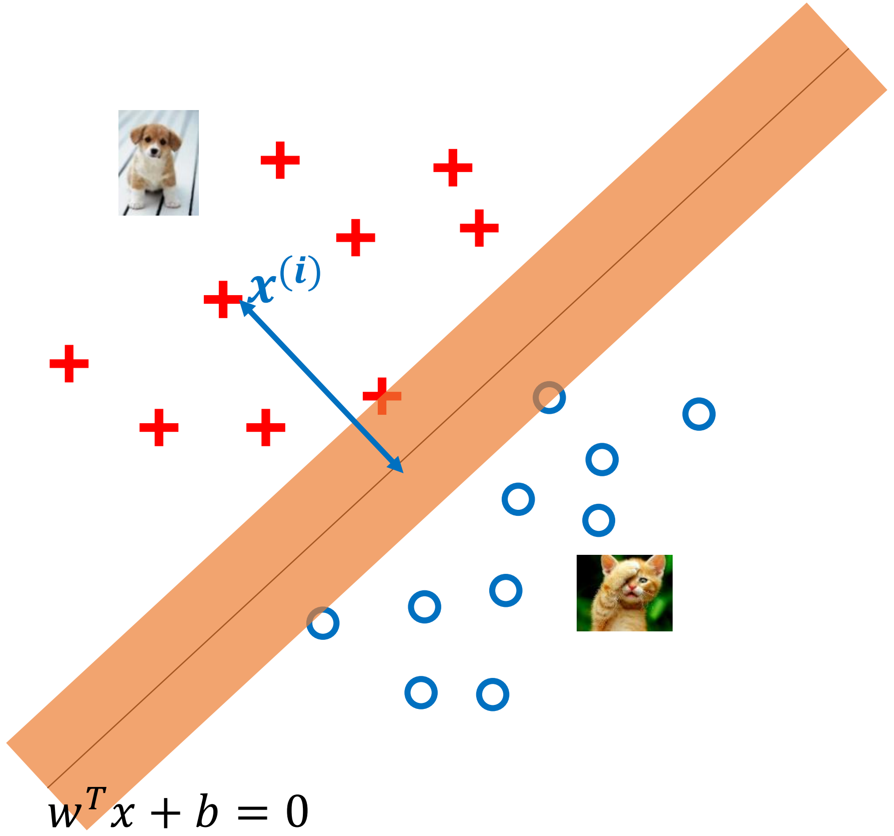
- What is the distance expression for a point $x^{(i)}$ to a line $w^T x + b = 0$?



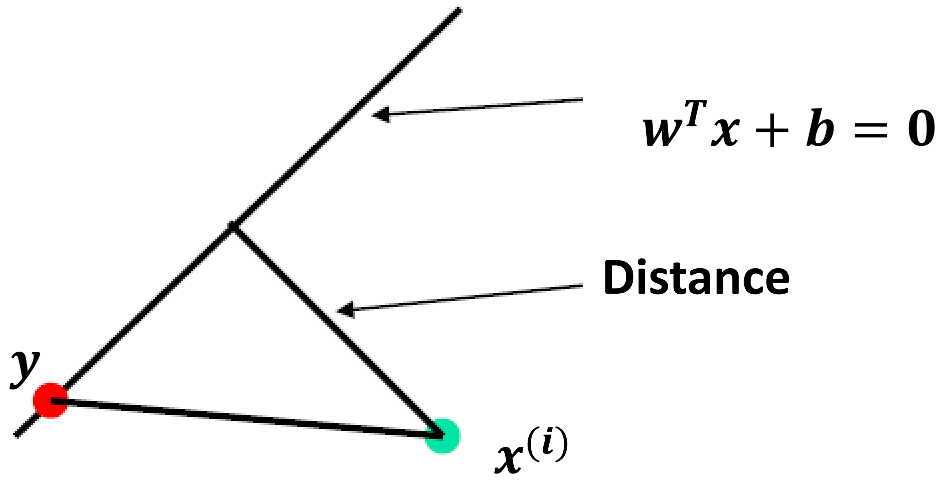
SVM

- What is the distance expression for a point $x^{(i)}$ to a line $w^T x + b = 0$?

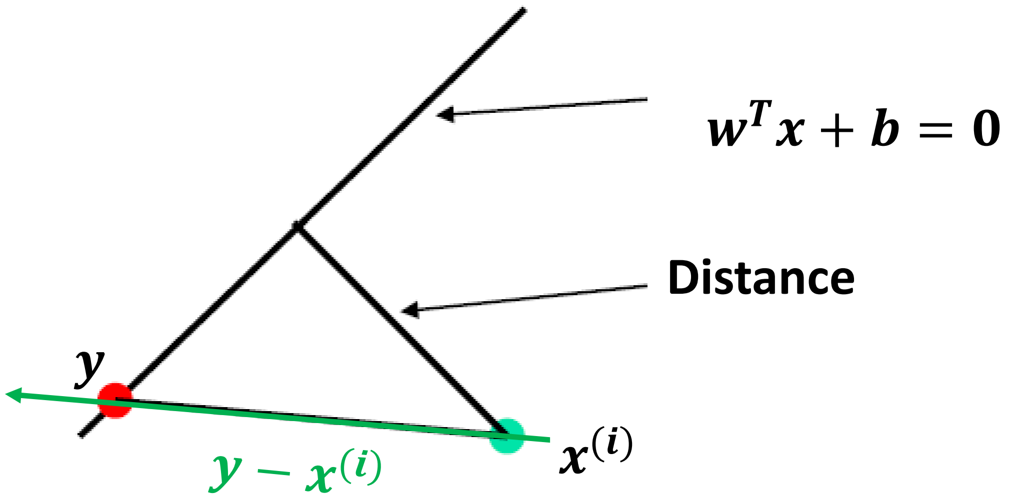
$$d\left(x^{(i)}\right) = \frac{\left|w^T x^{(i)} + b\right|}{\|w\|_2}$$



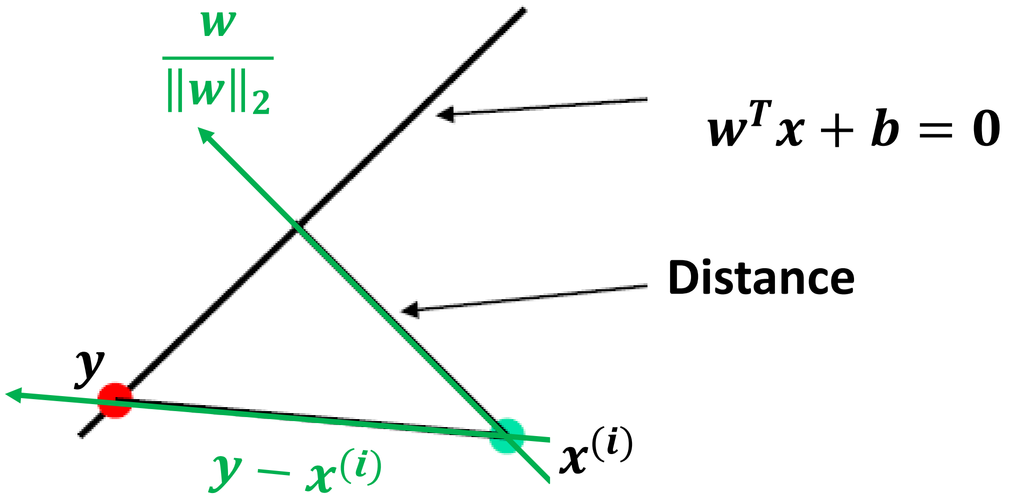
Distance Expression



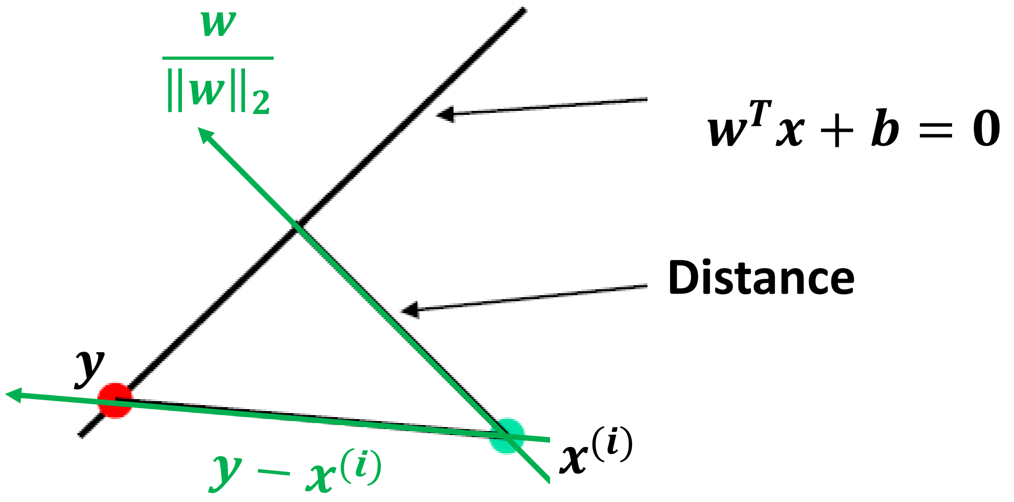
Distance Expression



Distance Expression

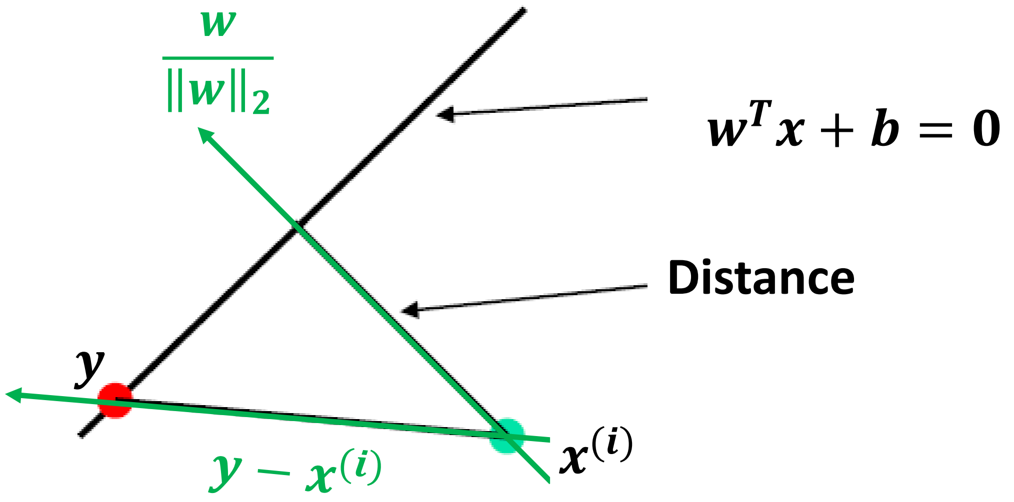


Distance Expression



$$d\left(x^{(i)}\right) = \left|\frac{\boldsymbol{w}^T}{\|\boldsymbol{w}\|_2}\left(\boldsymbol{y} - \boldsymbol{x}^{(i)}\right)\right|$$

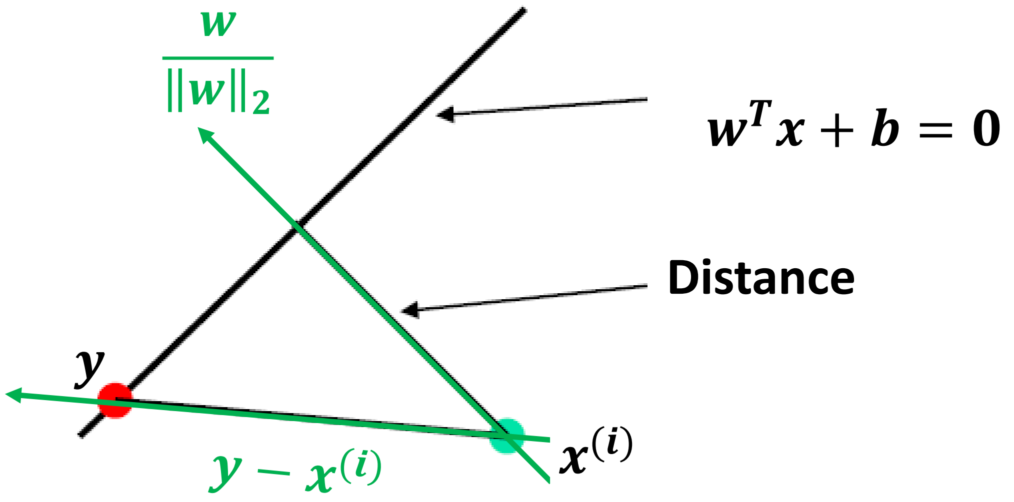
Distance Expression



$$d\left(x^{(i)}\right) = \left|\frac{\boldsymbol{w}^T}{\|\boldsymbol{w}\|_2}\left(\boldsymbol{y} - \boldsymbol{x}^{(i)}\right)\right|$$

$$= \left|\frac{\boldsymbol{w}^T\boldsymbol{y} - \boldsymbol{w}^T\boldsymbol{x}^{(i)}}{\|\boldsymbol{w}\|_2}\right|$$

Distance Expression

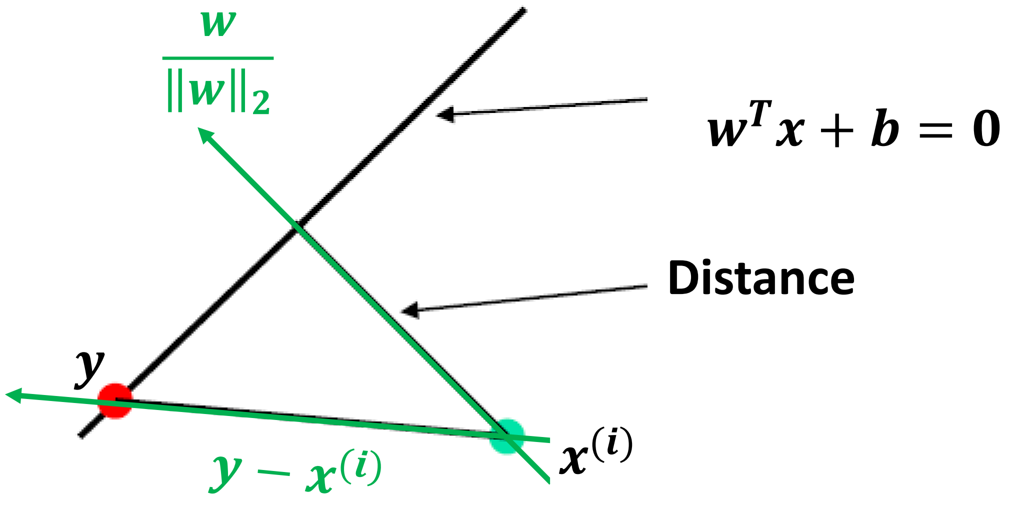


$$d\left(x^{(i)}\right) = \left|\frac{\boldsymbol{w}^T}{\|\boldsymbol{w}\|_2}\left(\boldsymbol{y} - \boldsymbol{x}^{(i)}\right)\right|$$

$$= \left|\frac{\boldsymbol{w}^T\boldsymbol{y} - \boldsymbol{w}^T\boldsymbol{x}^{(i)}}{\|\boldsymbol{w}\|_2}\right|$$

$$= \left|\frac{-\boldsymbol{b} - \boldsymbol{w}^T\boldsymbol{x}^{(i)}}{\|\boldsymbol{w}\|_2}\right|$$

Distance Expression



$$d\left(x^{(i)}\right) = \left|\frac{\boldsymbol{w}^T}{\|\boldsymbol{w}\|_2}\left(\boldsymbol{y} - \boldsymbol{x}^{(i)}\right)\right|$$

$$= \left|\frac{\boldsymbol{w}^T\boldsymbol{y} - \boldsymbol{w}^T\boldsymbol{x}^{(i)}}{\|\boldsymbol{w}\|_2}\right|$$

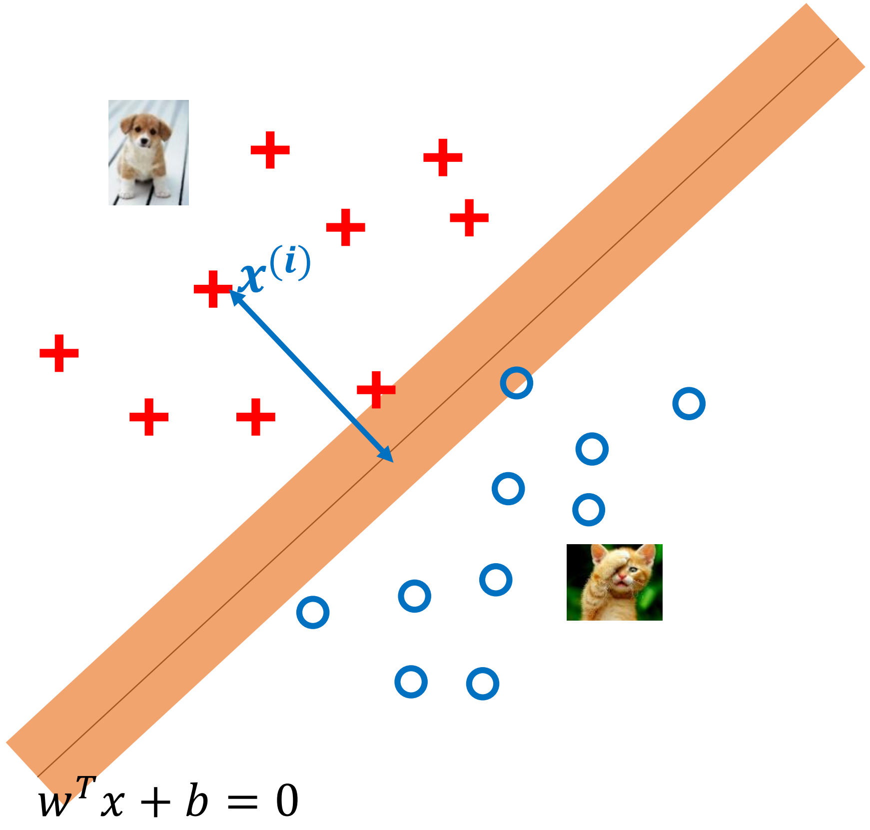
$$= \left|\frac{-\boldsymbol{b} - \boldsymbol{w}^T\boldsymbol{x}^{(i)}}{\|\boldsymbol{w}\|_2}\right|$$

$$= \left|\frac{\boldsymbol{w}^T\boldsymbol{x}^{(i)} + \boldsymbol{b}}{\|\boldsymbol{w}\|_2}\right|$$

SVM

- What is the distance expression for a point $x^{(i)}$ to a line $w^T x + b = 0$?

$$d\left(x^{(i)}\right) = \frac{\left|w^T x^{(i)} + b\right|}{\|w\|_2}$$



SVM

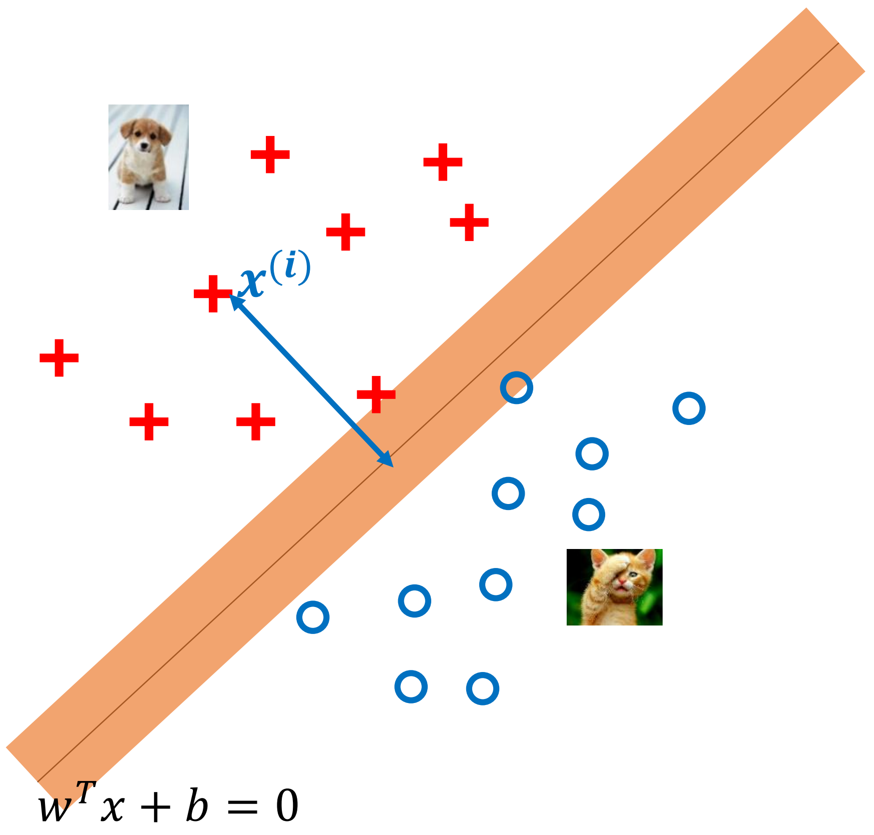
- What is the distance expression for a point $x^{(i)}$ to a line $w^T x + b = 0$?

$$d\left(x^{(i)}\right) = \frac{\left|w^T x^{(i)} + b\right|}{\|w\|_2}$$

- Margin: the width that the boundary could be increased by before hitting a datapoint

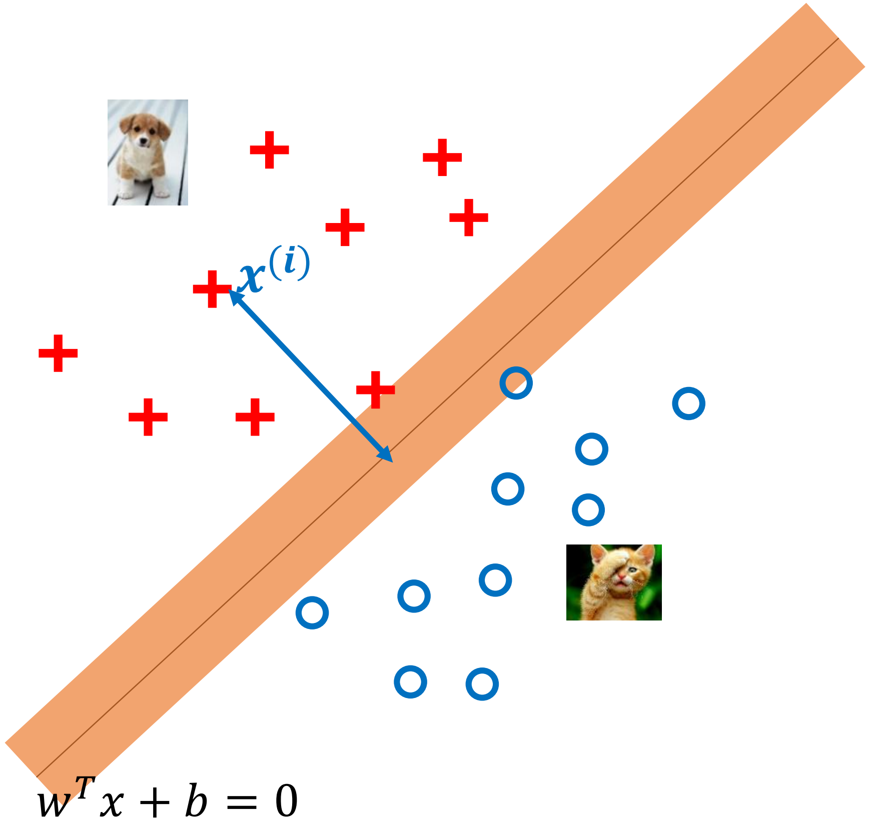
$$margin \equiv 2 * \min_{x^{(i)} \in D} d\left(x^{(i)}\right)$$

$$= 2 * \min_{x^{(i)} \in D} \frac{\left|w^T x^{(i)} + b\right|}{\|w\|_2}$$



SVM

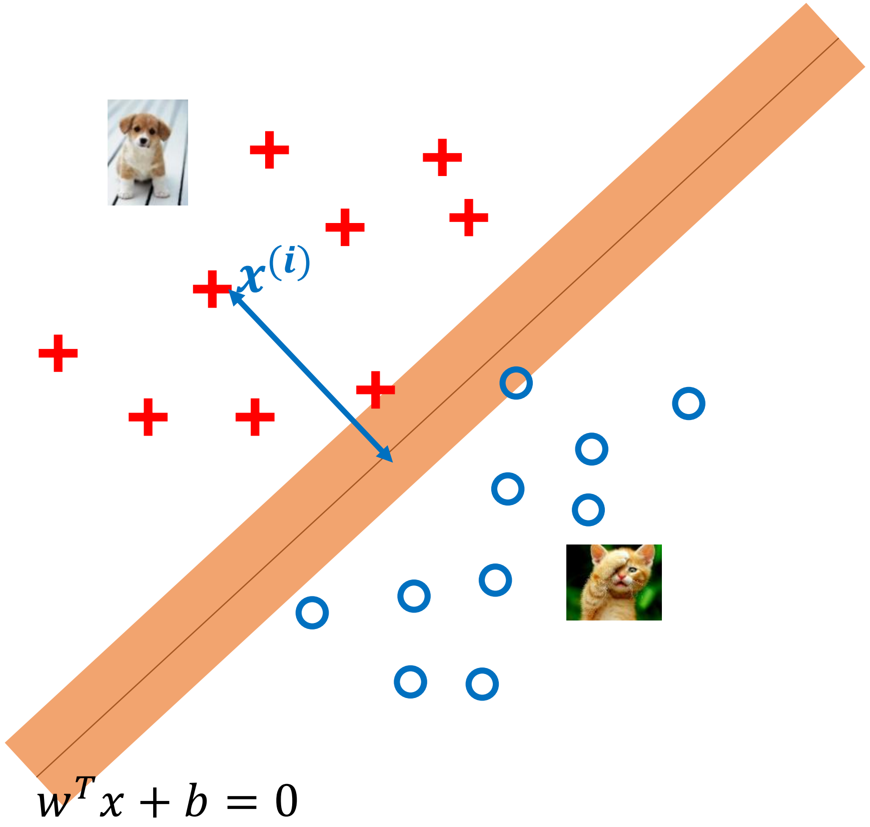
- **Linear SVM**
  - A maximum margin classifier
  - Preventing data points from falling into the margin



SVM

- **Linear SVM**
  - A maximum margin classifier
  - Preventing data points from falling into the margin

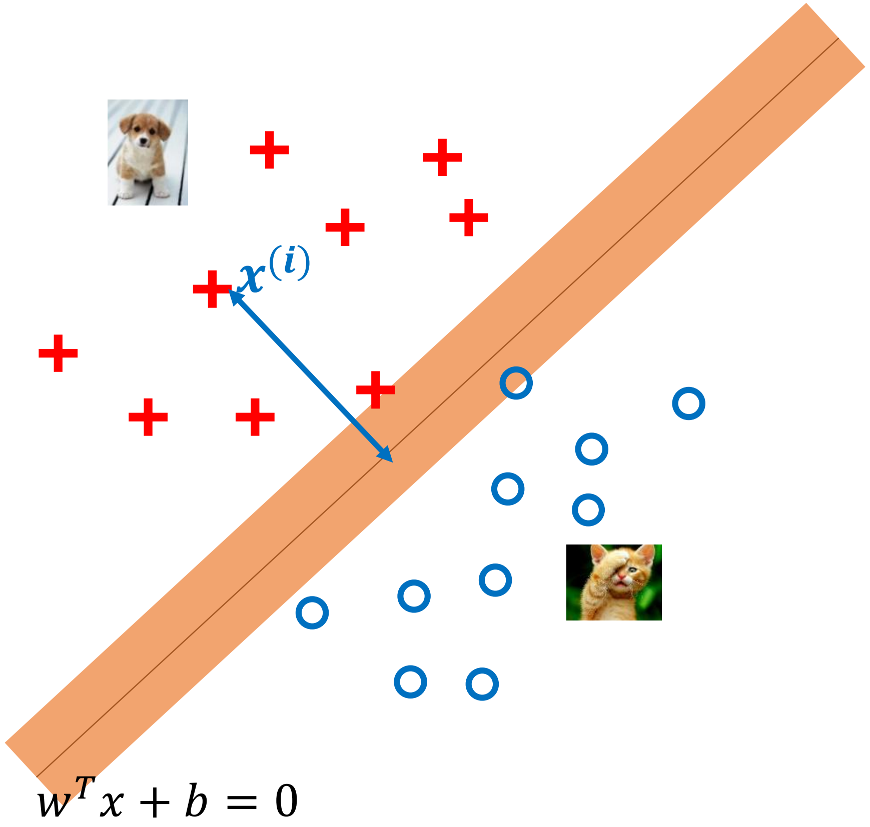
$$\max_{w,b} margin$$



SVM

- **Linear SVM**
  - A maximum margin classifier
  - Preventing data points from falling into the margin

$$\max_{w,b} margin$$

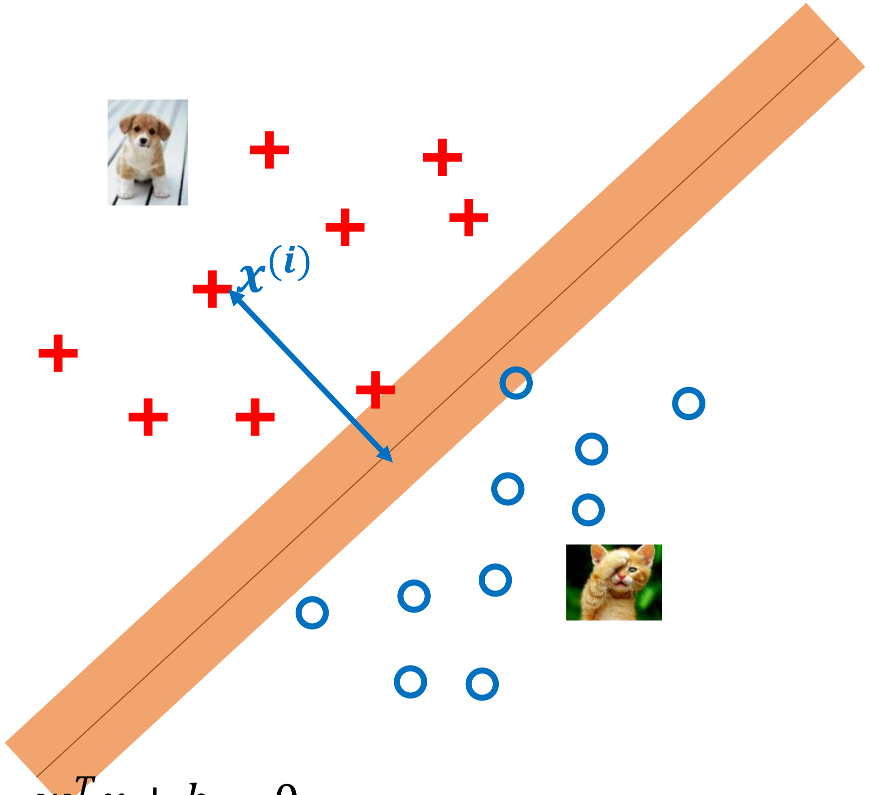


$$margin = \min_{x^{(i)} \in D} \frac{|w^T x^{(i)} + b|}{\|w\|_2}$$

SVM

- **Linear SVM**
  - A maximum margin classifier
  - Preventing data points from falling into the margin

$$\max_{w,b} margin = 2 * \max_{w,b} \min_{x^{(i)} \in D} \frac{|w^T x^{(i)} + b|}{\|w\|_2}$$



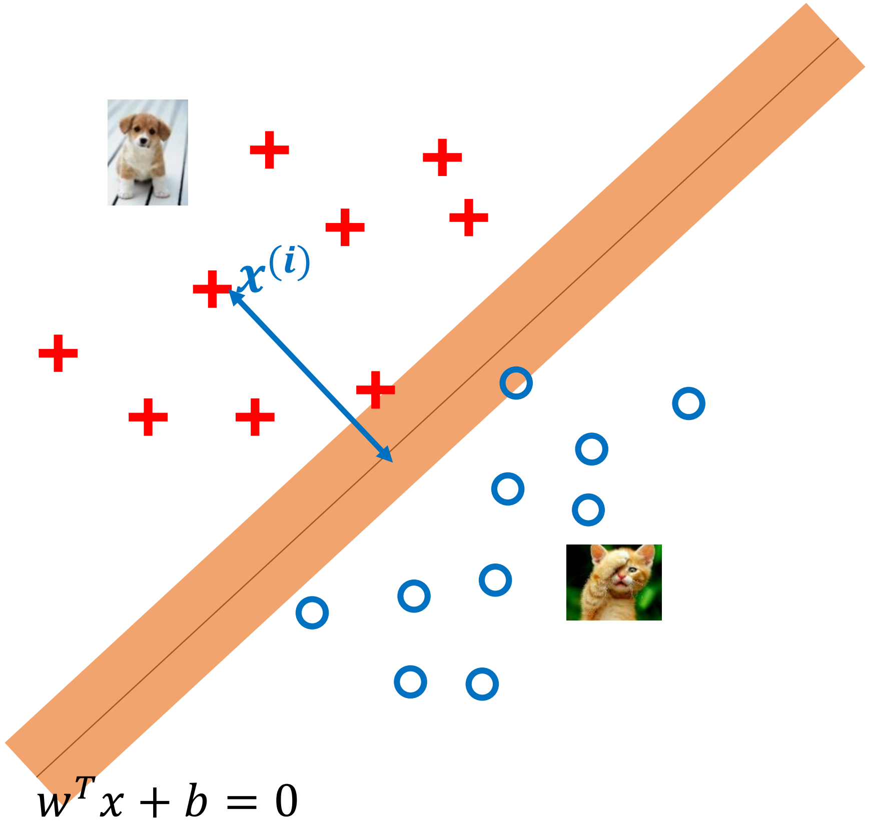
$$w^T x + b = 0$$

$$margin = \min_{x^{(i)} \in D} \frac{|w^T x^{(i)} + b|}{\|w\|_2}$$

SVM

- **Linear SVM**
  - A maximum margin classifier
  - Preventing data points from falling into the margin

$$\max_{w,b} margin = 2 * \max_{w,b} \min_{x^{(i)} \in D} \frac{|w^T x^{(i)} + b|}{\|w\|_2}$$



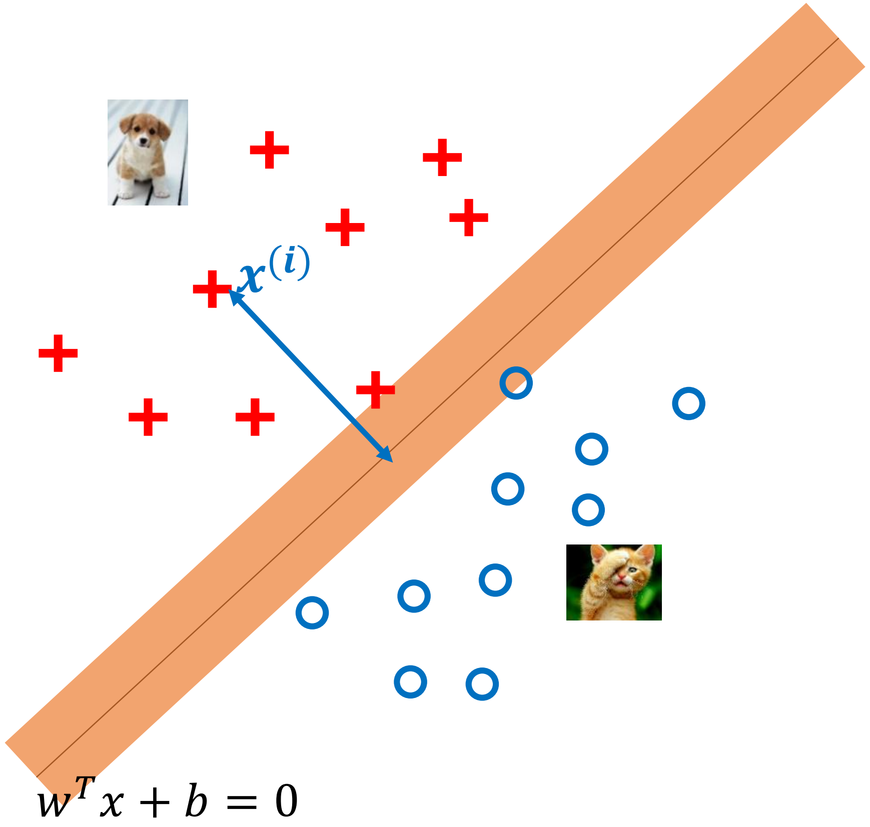
$$margin = \min_{x^{(i)} \in D} \frac{|w^T x^{(i)} + b|}{\|w\|_2}$$

SVM

- **Linear SVM**
  - A maximum margin classifier
  - Preventing data points from falling into the margin

Hard margin

$$\max_{w,b} margin = 2 * \max_{w,b} \min_{x^{(i)} \in D} \frac{|w^T x^{(i)} + b|}{\|w\|_2}$$

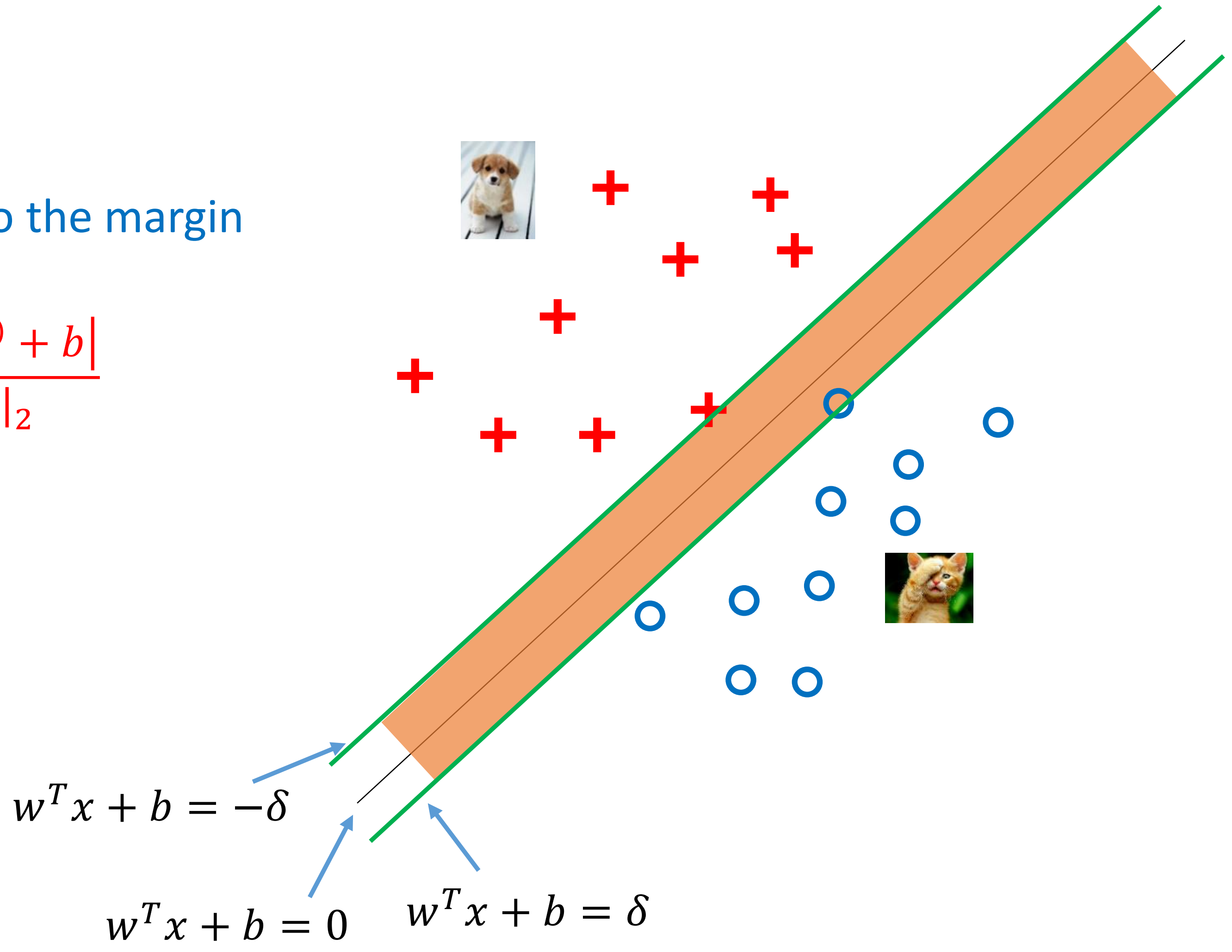


$$margin = \min_{x^{(i)} \in D} \frac{|w^T x^{(i)} + b|}{\|w\|_2}$$

SVM

- **Linear SVM**
  - A maximum margin classifier
  - Preventing data points from falling into the margin

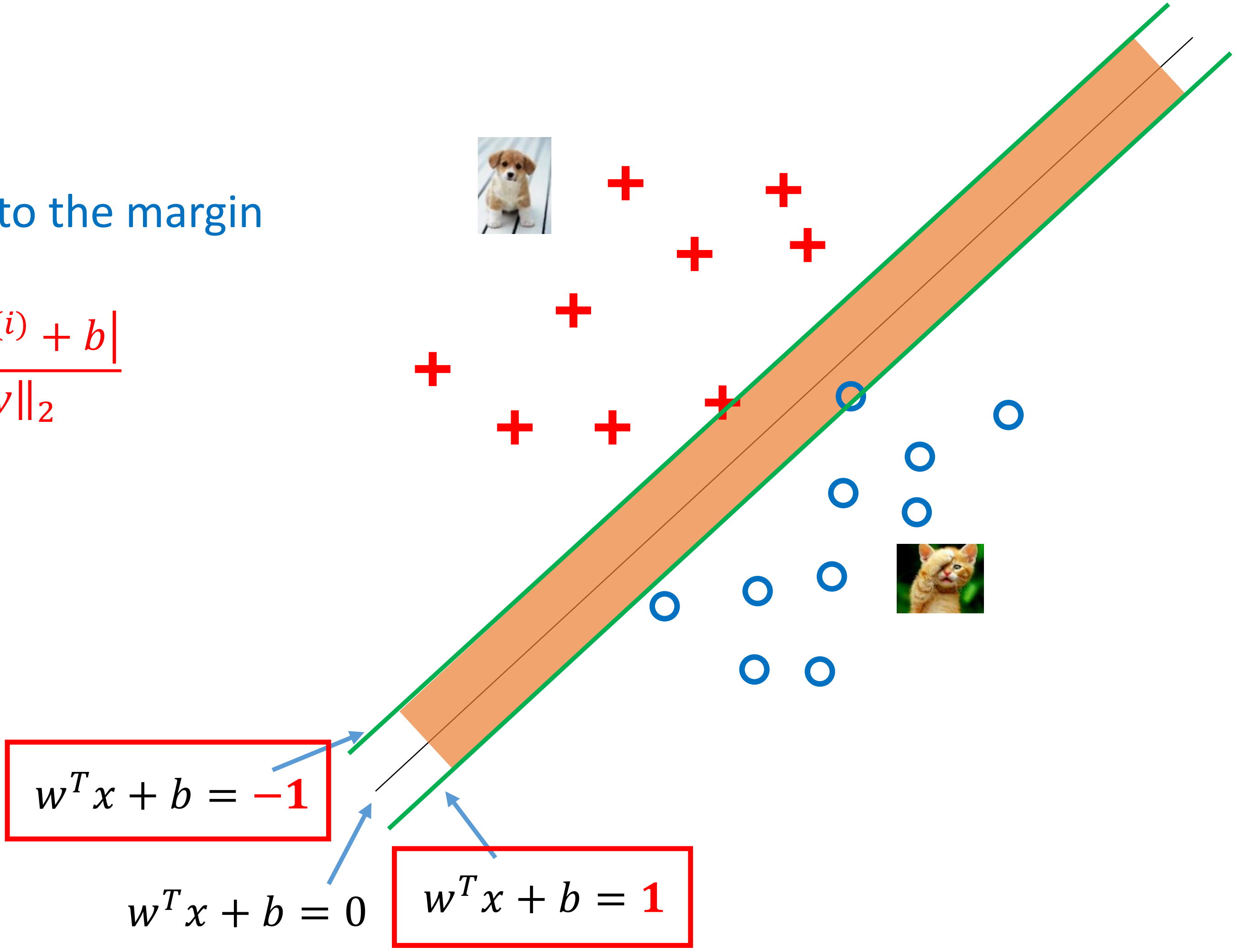
$$\max_{w,b} margin = 2 * \max_{w,b} \min_{x^{(i)} \in D} \frac{|w^T x^{(i)} + b|}{\|w\|_2}$$



SVM

- **Linear SVM**
  - A maximum margin classifier
  - Preventing data points from falling into the margin

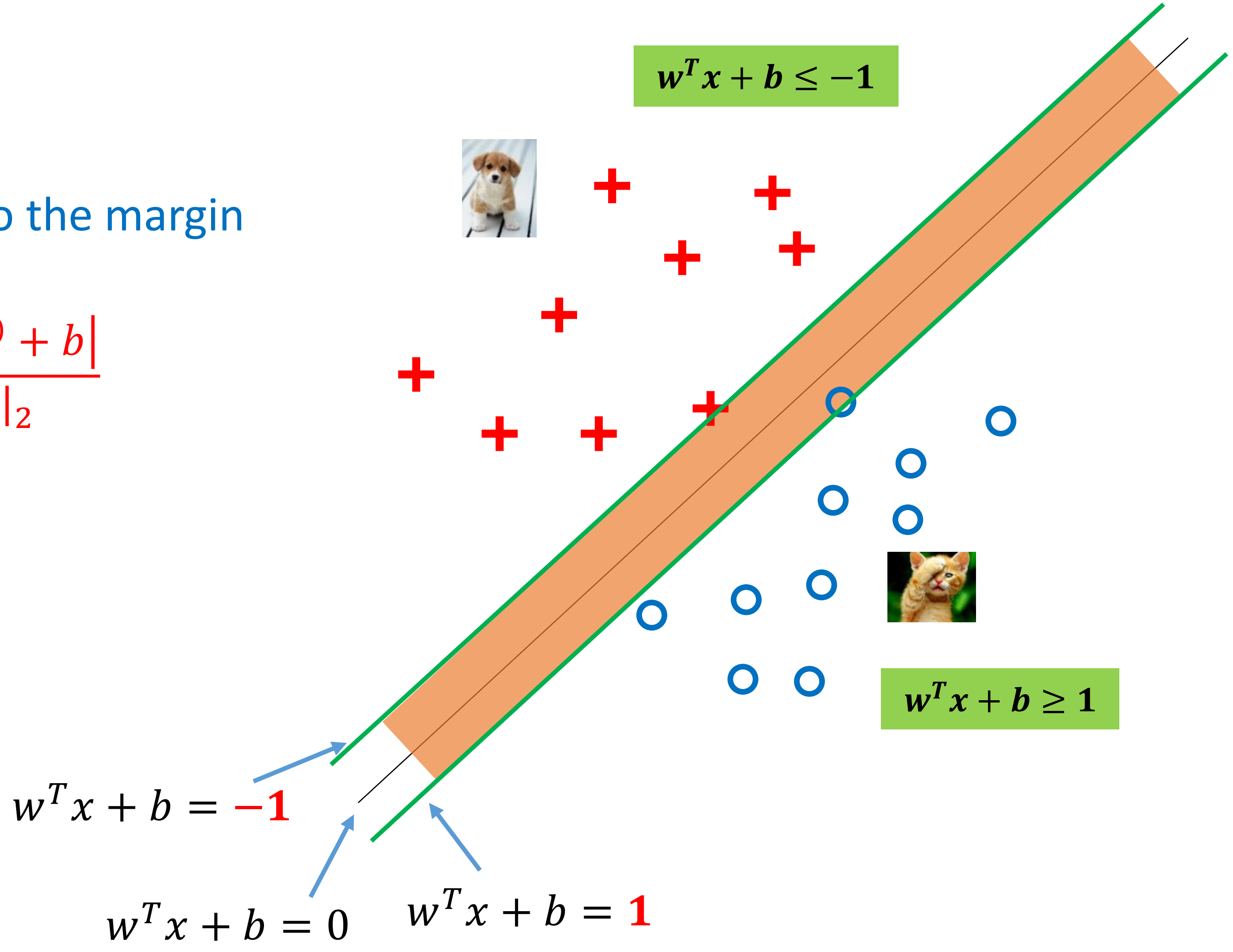
$$\max_{w,b} margin = 2 * \max_{w,b} \min_{x^{(i)} \in D} \frac{|w^T x^{(i)} + b|}{\|w\|_2}$$



SVM

- **Linear SVM**
  - A maximum margin classifier
  - Preventing data points from falling into the margin

$$\max_{w,b} margin = 2 * \max_{w,b} \min_{x^{(i)} \in D} \frac{|w^T x^{(i)} + b|}{\|w\|_2}$$



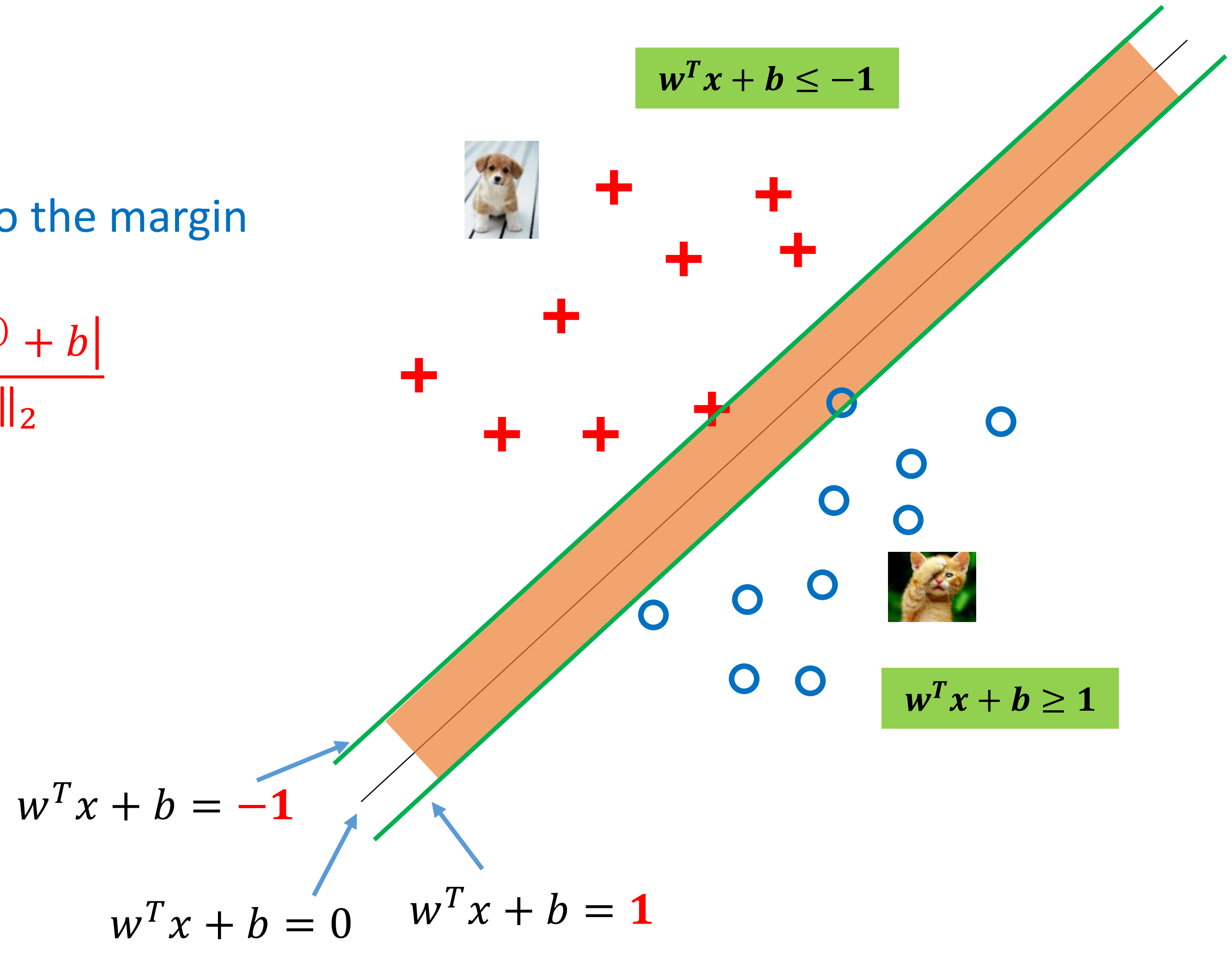
SVM

- **Linear SVM**
  - A maximum margin classifier
  - Preventing data points from falling into the margin

$$\max_{w,b} margin = 2 * \max_{w,b} \min_{x^{(i)} \in D} \frac{|w^T x^{(i)} + b|}{\|w\|_2}$$

For a data point $x^{(i)}$,

- if its target $t^{(i)} = 1$, $\quad w^T x^{(i)} + b \geq 1$
- if its target $t^{(i)} = -1$, $\quad w^T x^{(i)} + b \leq -1$



SVM

- **Linear SVM**
  - A maximum margin classifier
  - Preventing data points from falling into the margin

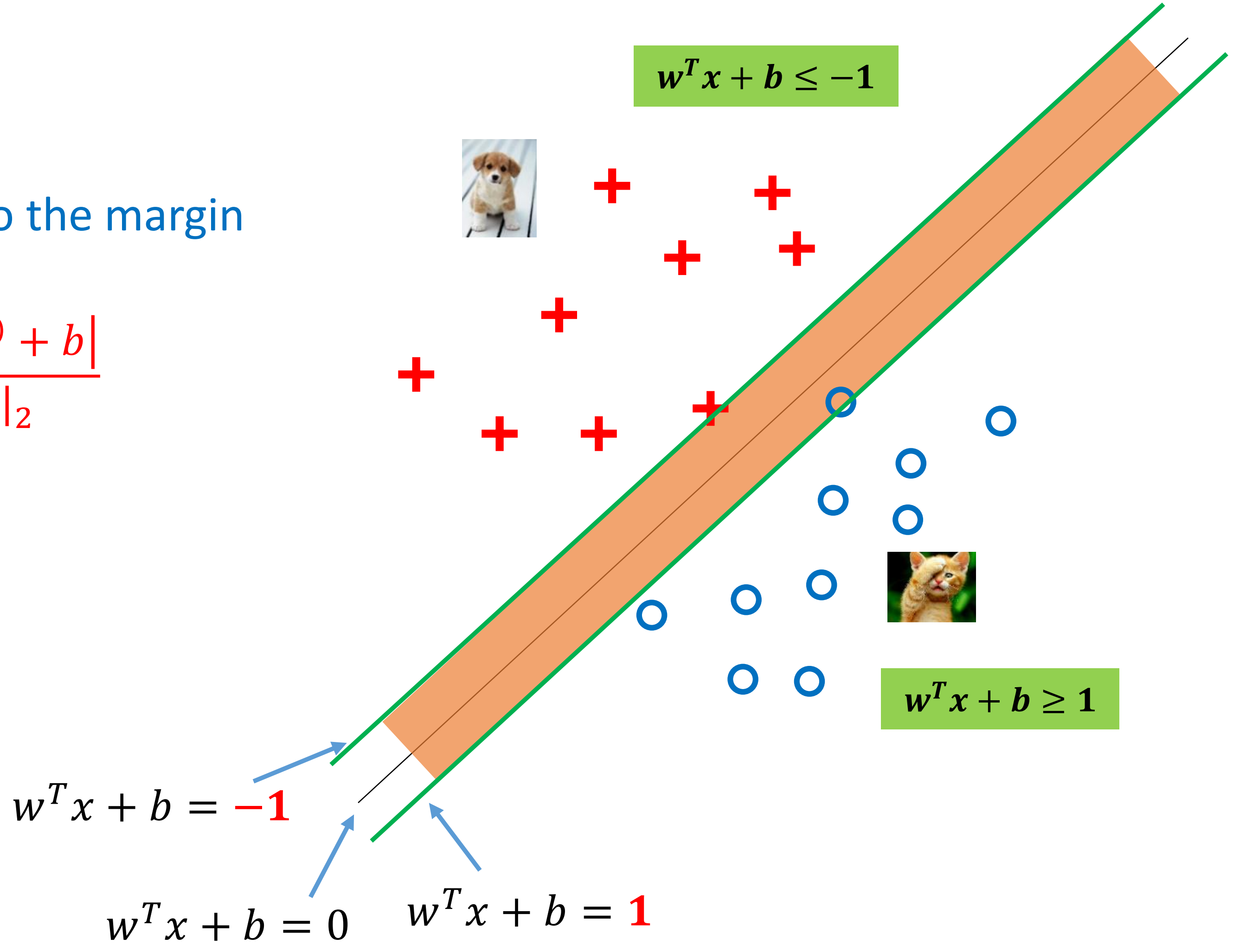
$$\max_{w,b} margin = 2 * \max_{w,b} \min_{x^{(i)}\in D} \frac{|w^T x^{(i)} + b|}{\|w\|_2}$$

For a data point $x^{(i)}$,

- if its target $t^{(i)} = 1$, $\quad w^T x^{(i)} + b \geq 1$
- if its target $t^{(i)} = -1$, $\quad w^T x^{(i)} + b \leq -1$

So we want

$$\left(\boldsymbol{w}^T \boldsymbol{x}^{(i)} + \boldsymbol{b}\right)\boldsymbol{t}^{(i)} \geq \mathbf{1}$$

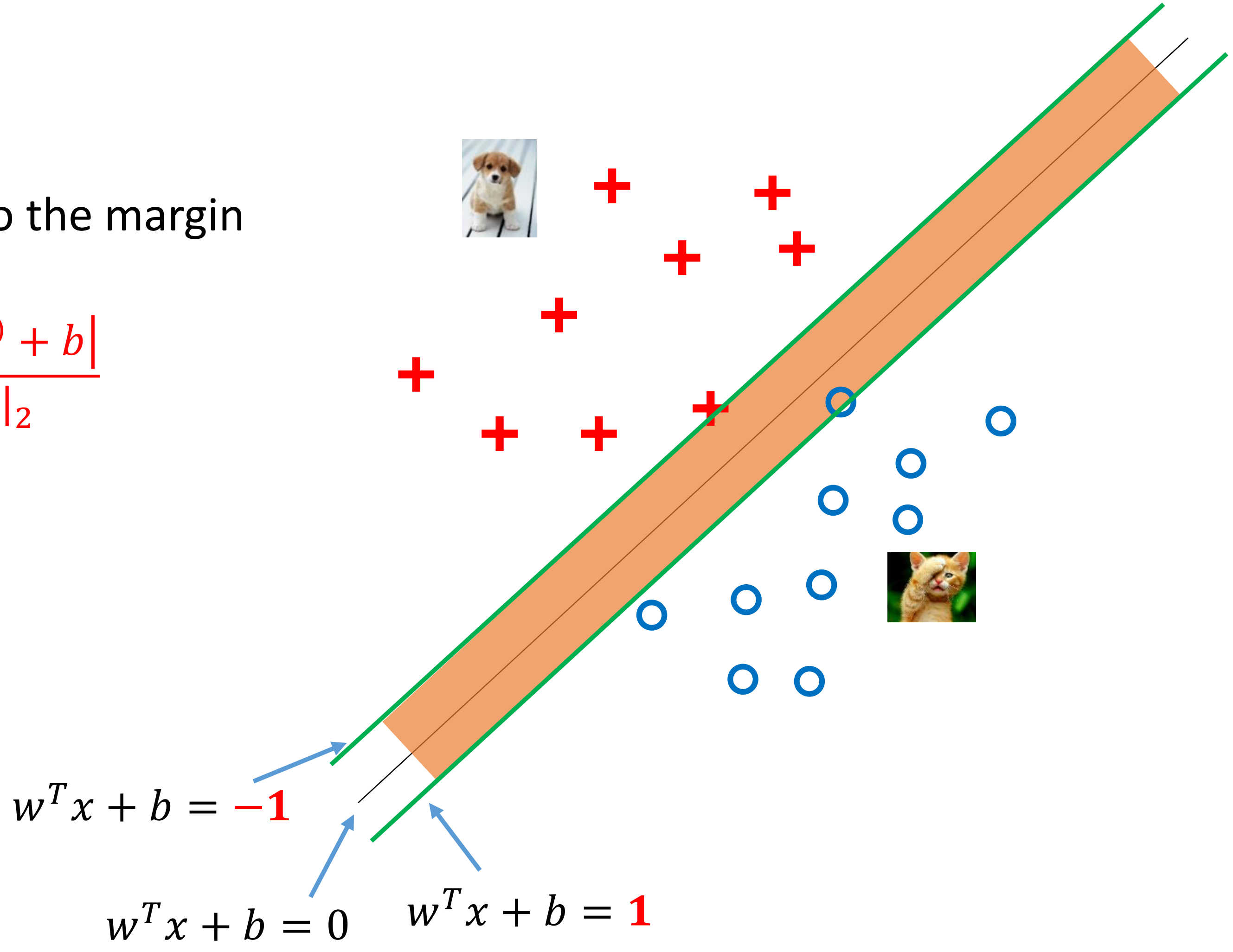


SVM

- **Linear SVM**
  - A maximum margin classifier
  - Preventing data points from falling into the margin

$$\max_{w,b} margin = 2 * \max_{w,b} \min_{x^{(i)} \in D} \frac{\left|w^T x^{(i)} + b\right|}{\|w\|_2}$$

$$s.t. \quad \forall \boldsymbol{i}, (w^T x^{(i)} + b) t^{(i)} \geq 1$$



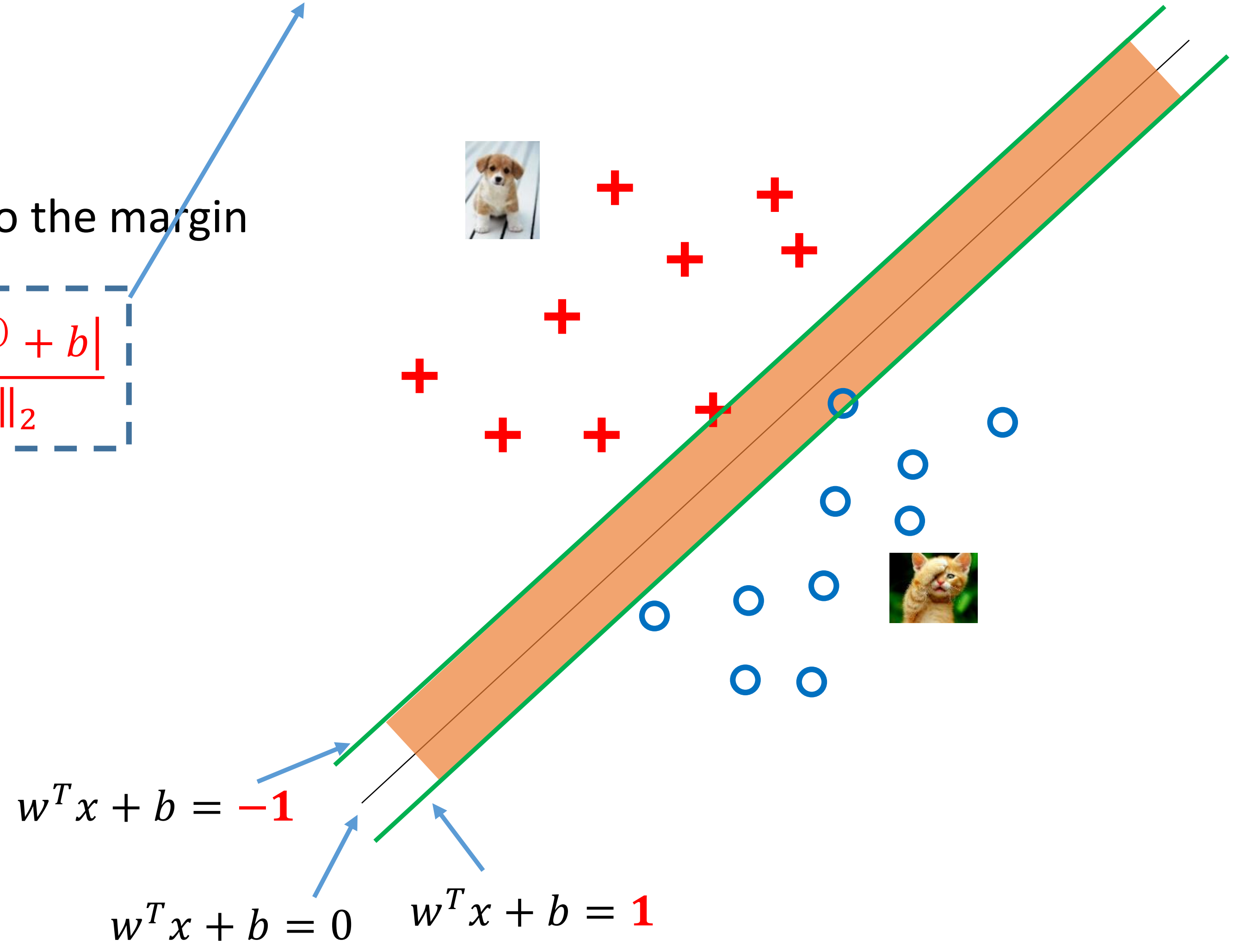
SVM

$$\max_{w,b} \min_{x^{(i)} \in D} \frac{|w^T x^{(i)} + b|}{\|w\|_2} \Rightarrow \max_{w,b} \frac{1}{\|w\|_2}$$

- **Linear SVM**
  - A maximum margin classifier
  - Preventing data points from falling into the margin

$$\max_{w,b} margin = 2 * \max_{w,b} \min_{x^{(i)} \in D} \frac{|w^T x^{(i)} + b|}{\|w\|_2}$$

$$s.t. \quad \forall \boldsymbol{i}, (w^T x^{(i)} + b) t^{(i)} \geq 1$$



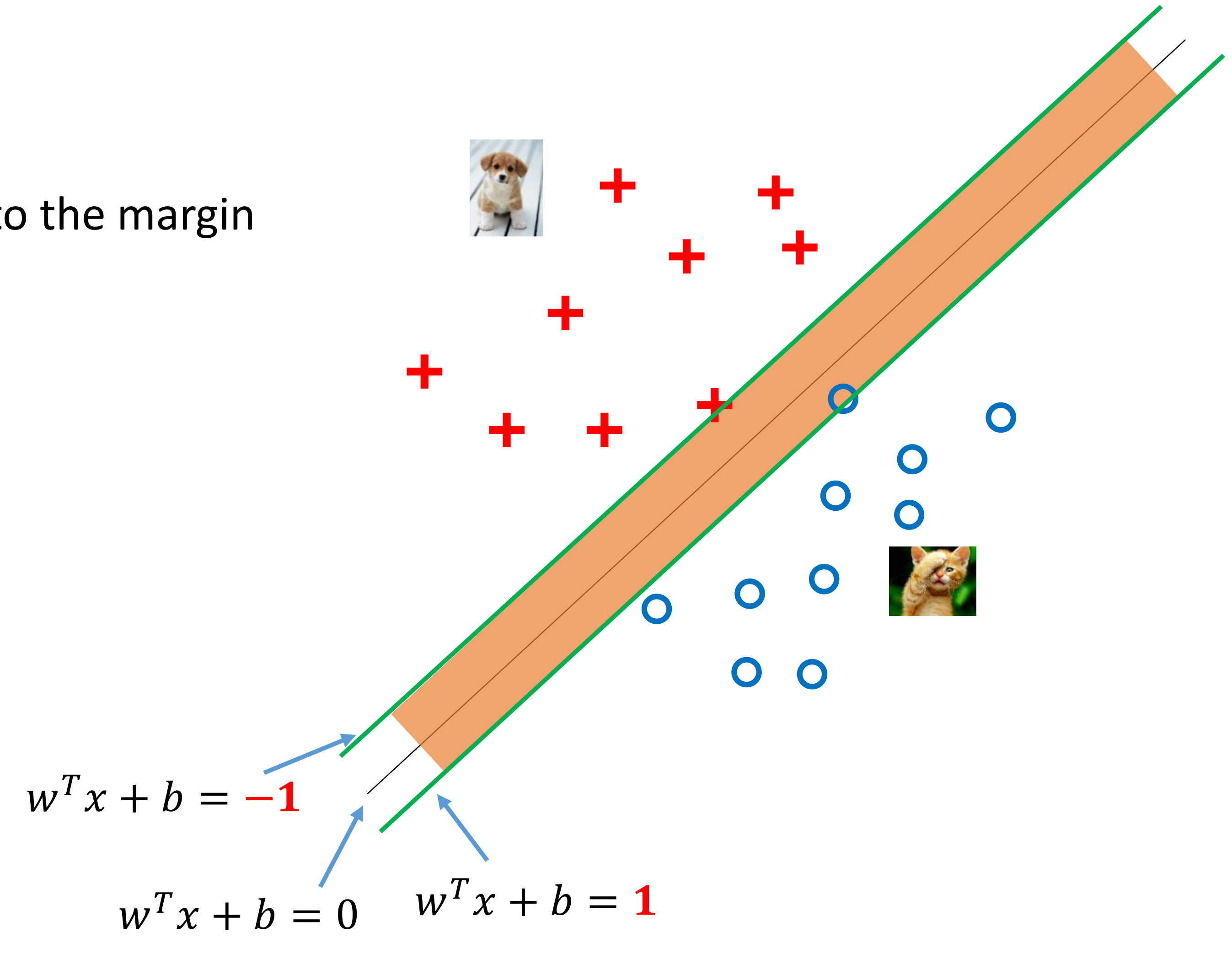
SVM

$$\max_{w,b} \min_{x^{(i)} \in D} \frac{|w^T x^{(i)} + b|}{\|w\|_2} \Rightarrow \max_{w,b} \frac{1}{\|w\|_2}$$

- **Linear SVM**
  - A maximum margin classifier
  - Preventing data points from falling into the margin

$$\max_{w,b} margin = 2 * \max_{w,b} \frac{1}{\|w\|_2}$$

$$s.t. \quad \forall \boldsymbol{i}, (w^T x^{(i)} + b)t^{(i)} \geq 1$$



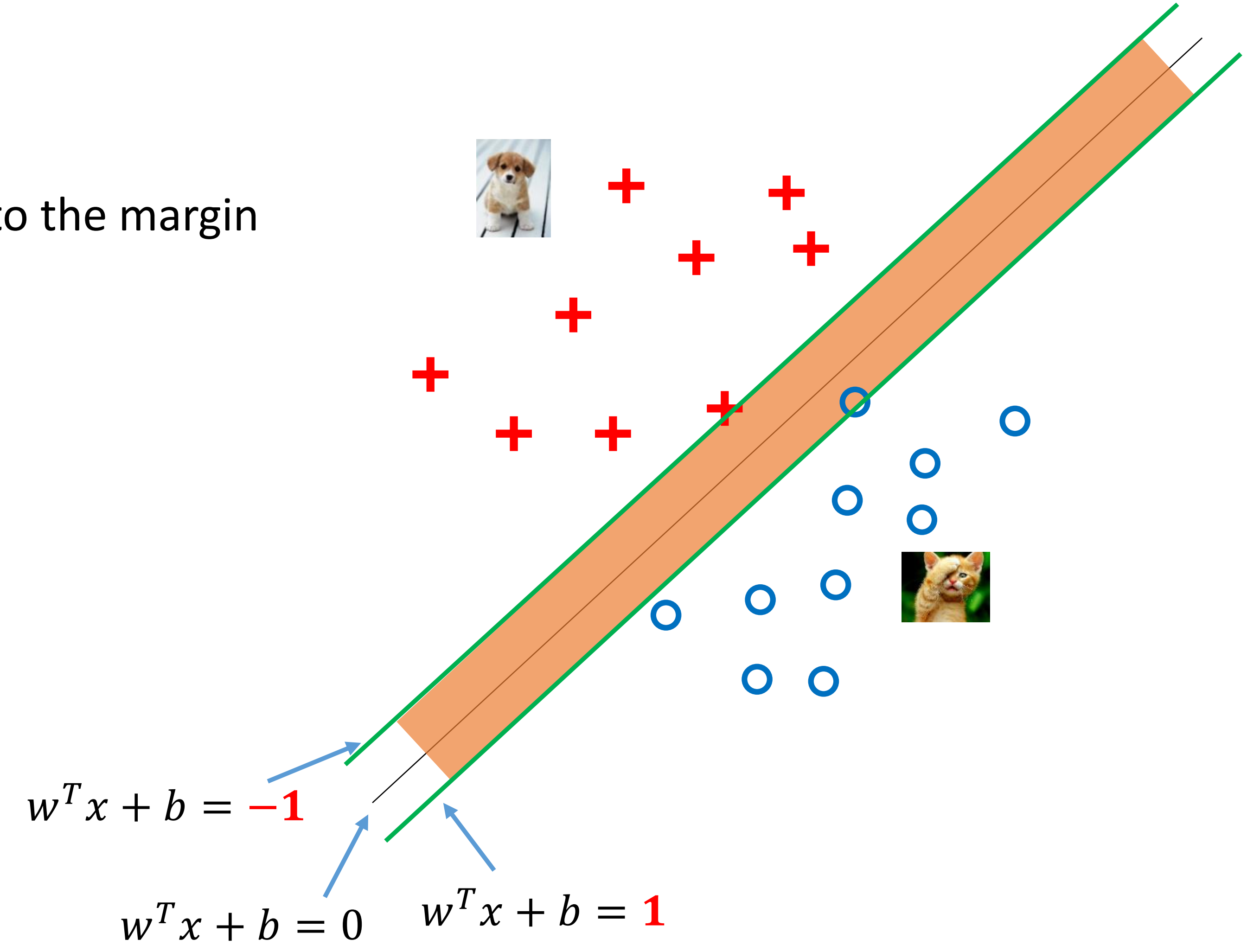
SVM

$$\max_{w,b} \min_{x^{(i)} \in D} \frac{|w^T x^{(i)} + b|}{\|w\|_2} \Rightarrow \max_{w,b} \frac{1}{\|w\|_2} \Rightarrow \min_{w,b} \frac{1}{2}\|w\|_2^2$$

- **Linear SVM**
  - A maximum margin classifier
  - Preventing data points from falling into the margin

$$\min_{w,b} \frac{1}{2}\|w\|_2^2$$

$$s.t. \quad \forall \boldsymbol{i}, (w^T x^{(i)} + b) t^{(i)} \geq 1$$

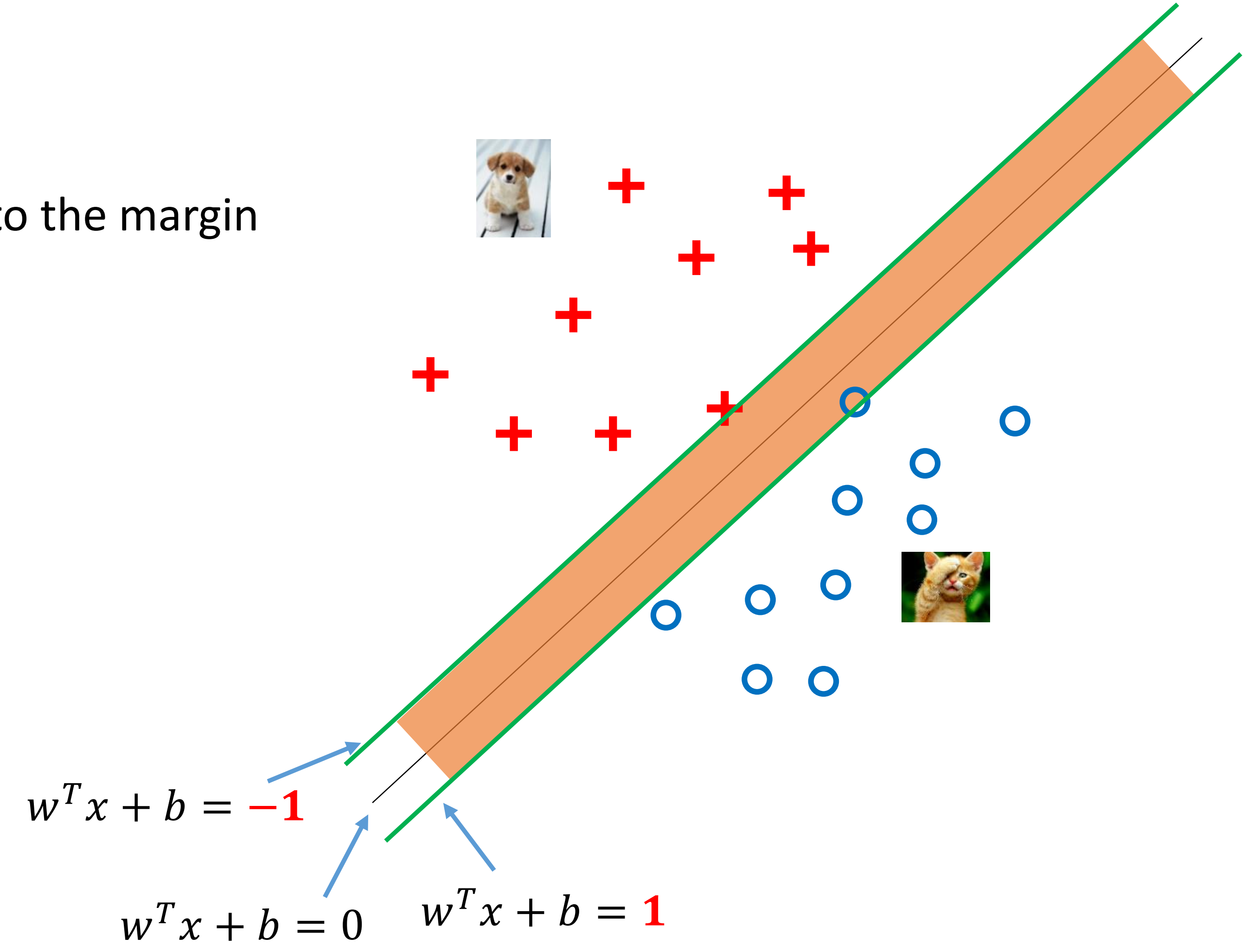


SVM

- **Linear SVM**
  - A maximum margin classifier
  - Preventing data points from falling into the margin

$$\min_{w,b} \frac{1}{2}\|w\|_2^2$$
$$s.t. \quad \forall i, (w^T x^{(i)} + b)t^{(i)} \geq 1$$

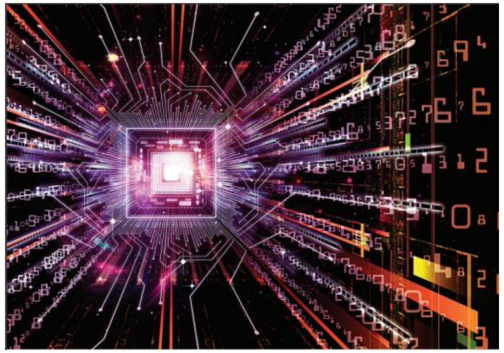
A constrained minimization



Machine Learning

- **Learning**

Representation ⇨ Evaluation ⇨ Optimization



Reference: Domingos, Pedro. "A few useful things to know about machine learning." *Communications of the ACM* 55.10 (2012): 78-87.

Three Steps for SVM

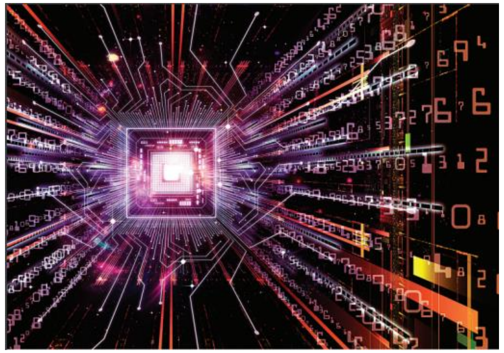
- **Learning**

Representation ⇨ **Evaluation** ⇨ **Optimization**

$y(x) = w^T x + b$

$$\min_{w,b} \frac{1}{2}\|w\|_2^2$$

$$s.t. \quad \forall i, (w^T x^{(i)} + b)t^{(i)} \geq 1$$



Reference: Domingos, Pedro. "A few useful things to know about machine learning." *Communications of the ACM* 55.10 (2012): 78-87.

Three Steps for SVM

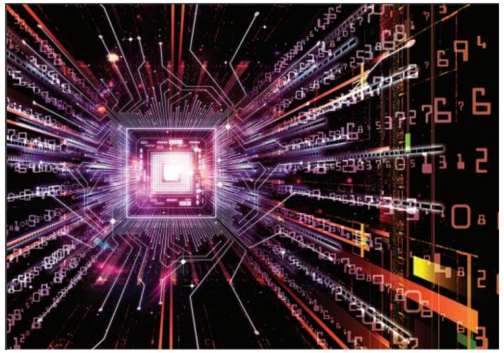
- **Learning**

Representation ⇨ **Evaluation** ⇨ **Optimization**

$y(x) = w^T x + b$

$$\min_{w,b} \frac{1}{2}\|w\|_2^2$$

$$s.t. \quad \forall i, \left(w^T x^{(i)} + b\right) t^{(i)} \geq 1$$



Reference: Domingos, Pedro. "A few useful things to know about machine learning." *Communications of the ACM* 55.10 (2012): 78-87.

$$\min_{w,b} \frac{1}{2}\|w\|_2^2$$

$$s.t. \quad \forall i\,,\left(w^T x^{(i)} + b\right)t^{(i)} \geq 1$$

$$\min_{w,b} \frac{1}{2}\|w\|_2^2$$

$$s.t. \quad \forall i\,, \left(w^T x^{(i)} + b\right) t^{(i)} \geq 1$$

Dual problem

Lagrange function

$$L(w,b,\alpha) = \frac{1}{2}\|w\|_2^2 + \sum_{i=1}^{N} \alpha^{(i)}\left(1 - \left(w^T x^{(i)} + b\right)t^{(i)}\right)$$

where $\alpha^{(i)}$'s are Lagrange multipliers

$$\min_{w,b} \frac{1}{2}\|w\|_2^2$$

$$s.t. \quad \forall i, \left(w^T x^{(i)} + b\right)t^{(i)} \geq 1$$

Dual problem

Lagrange function

$$L(w,b,\alpha) = \frac{1}{2}\|w\|_2^2 + \sum_{i=1}^{N} \alpha^{(i)}\left(1 - \left(w^T x^{(i)} + b\right)t^{(i)}\right)$$

where $\alpha^{(i)}$'s are Lagrange multipliers

First, minimize function L w.r.t. w, b for fixed Lagrange multipliers

$$\frac{\partial L(w,b,\alpha)}{\partial w} = w - \sum_{i=1}^{N} \alpha^{(i)} t^{(i)} x^{(i)} = 0$$

$$\frac{\partial L(w,b,\alpha)}{\partial b} = \sum_{i=1}^{N} \alpha^{(i)} t^{(i)} = 0$$

$$\boldsymbol{\min_{w,b} \frac{1}{2}\|w\|_2^2}$$

$$\boldsymbol{s.t.\quad \forall i, \left(w^T x^{(i)} + b\right)t^{(i)} \geq 1}$$

Dual problem

Lagrange function

$$L(w,b,\alpha) = \frac{1}{2}\|w\|_2^2 + \sum_{i=1}^{N} \alpha^{(i)}\left(1 - \left(w^T x^{(i)} + b\right)t^{(i)}\right)$$

where $\alpha^{(i)}$'s are Lagrange multipliers

First, minimize function L w.r.t. w, b for fixed Lagrange multipliers

$$\frac{\partial L(w,b,\alpha)}{\partial w} = w - \sum_{i=1}^{N} \alpha^{(i)} t^{(i)} x^{(i)} = 0 \quad \Rightarrow \quad w = \sum_{i=1}^{N} \alpha^{(i)} t^{(i)} x^{(i)}$$

$$\frac{\partial L(w,b,\alpha)}{\partial b} = \sum_{i=1}^{N} \alpha^{(i)} t^{(i)} = 0$$

$$\min_{w,b} \frac{1}{2}\|w\|_2^2$$

$$s.t. \quad \forall i, \left(w^T x^{(i)} + b\right)t^{(i)} \geq 1$$

Dual problem

Lagrange function

$$L(w,b,\alpha) = \frac{1}{2}\|w\|_2^2 + \sum_{i=1}^{N} \alpha^{(i)}\left(1 - \left(w^T x^{(i)} + b\right) t^{(i)}\right)$$

where $\alpha^{(i)}$'s are Lagrange multipliers

$$\min_{w,b} \frac{1}{2}\|w\|_2^2$$

$$s.t. \quad \forall i, \left(w^T x^{(i)} + b\right) t^{(i)} \geq 1$$

First, minimize function L w.r.t. w, b for fixed Lagrange multipliers

$$\frac{\partial L(w,b,\alpha)}{\partial w} = w - \sum_{i=1}^{N} \alpha^{(i)} t^{(i)} x^{(i)} = 0 \quad \Rightarrow \quad w = \sum_{i=1}^{N} \alpha^{(i)} t^{(i)} x^{(i)}$$

$$\frac{\partial L(w,b,\alpha)}{\partial b} = \sum_{i=1}^{N} \alpha^{(i)} t^{(i)} = 0$$

Then, substitute w back to function L

$$L(\alpha) = \sum_{i=1}^{N} \alpha^{(i)} - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} t^{(i)} t^{(j)} \alpha^{(i)} \alpha^{(j)} \left(x^{(i)^T} x^{(j)}\right)$$

Dual problem

Lagrange function

$$L(w,b,\alpha) = \frac{1}{2}\|w\|_2^2 + \sum_{i=1}^{N} \alpha^{(i)}\left(1 - \left(w^T x^{(i)} + b\right)t^{(i)}\right)$$

where $\alpha^{(i)}$'s are Lagrange multipliers

First, minimize function L w.r.t. w, b for fixed Lagrange multipliers

$$\frac{\partial L(w,b,\alpha)}{\partial w} = w - \sum_{i=1}^{N} \alpha^{(i)} t^{(i)} x^{(i)} = 0 \quad \Rightarrow \quad w = \sum_{i=1}^{N} \alpha^{(i)} t^{(i)} x^{(i)}$$

$$\frac{\partial L(w,b,\alpha)}{\partial b} = \sum_{i=1}^{N} \alpha^{(i)} t^{(i)} = 0$$

Then, substitute w back to function L

$$L(\alpha) = \sum_{i=1}^{N} \alpha^{(i)} - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} t^{(i)} t^{(j)} \alpha^{(i)} \alpha^{(j)} \left({x^{(i)}}^T x^{(j)}\right)$$

Next, we can obtain $\alpha^{(i)}$'s by solving the following optimization problem

$$\max_{\alpha^{(i)}} L(\alpha) = \max_{\alpha^{(i)}} \sum_{i=1}^{N} \alpha^{(i)} - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} t^{(i)} t^{(j)} \alpha^{(i)} \alpha^{(j)} \left({x^{(i)}}^T x^{(j)}\right)$$

s.t. $\alpha^{(i)} \geq 0$ and $\sum_{i=1}^{N} \alpha^{(i)} t^{(i)} = 0$

$$\min_{w,b} \frac{1}{2}\|w\|_2^2$$

$$s.t. \quad \forall i, \left(w^T x^{(i)} + b\right)t^{(i)} \geq 1$$

Dual problem

Lagrange function

$$L(w,b,\alpha) = \frac{1}{2}\|w\|_2^2 + \sum_{i=1}^{N} \alpha^{(i)}\left(1 - \left(w^T x^{(i)} + b\right)t^{(i)}\right)$$

where $\alpha^{(i)}$'s are Lagrange multipliers

First, minimize function L w.r.t. w, b for fixed Lagrange multipliers

$$\frac{\partial L(w,b,\alpha)}{\partial w} = w - \sum_{i=1}^{N} \alpha^{(i)} t^{(i)} x^{(i)} = 0 \quad \Rightarrow \quad w = \sum_{i=1}^{N} \alpha^{(i)} t^{(i)} x^{(i)}$$

$$\frac{\partial L(w,b,\alpha)}{\partial b} = \sum_{i=1}^{N} \alpha^{(i)} t^{(i)} = 0$$

Then, substitute w back to function L

$$L(\alpha) = \sum_{i=1}^{N} \alpha^{(i)} - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} t^{(i)} t^{(j)} \alpha^{(i)} \alpha^{(j)} \left({x^{(i)}}^T x^{(j)}\right)$$

Next, we can obtain $\alpha^{(i)}$'s by solving the following optimization problem

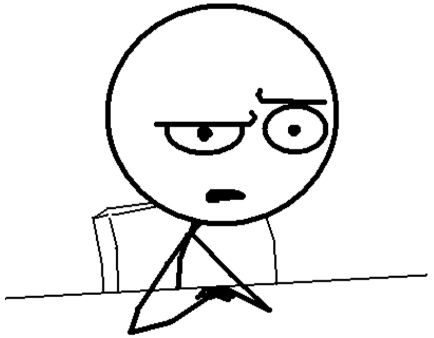
$$\max_{\alpha^{(i)}} L(\alpha) = \max_{\alpha^{(i)}} \sum_{i=1}^{N} \alpha^{(i)} - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} t^{(i)} t^{(j)} \alpha^{(i)} \alpha^{(j)} \left({x^{(i)}}^T x^{(j)}\right)$$

s.t. $\alpha^{(i)} \geq 0$ and $\sum_{i=1}^{N} \alpha^{(i)} t^{(i)} = 0$

$$\min_{w,b} \frac{1}{2}\|w\|_2^2$$

$$s.t. \quad \forall i, \left(w^T x^{(i)} + b\right) t^{(i)} \geq 1$$

Dual problem



$$\min_{w,b} \frac{1}{2}\|w\|_2^2$$

$$s.t. \quad \forall i, (w^T x^{(i)} + b)t^{(i)} \geq 1$$

Lagrange function

$$L(w,b,\alpha) = \frac{1}{2}\|w\|_2^2 + \sum_{i=1}^{N} \alpha^{(i)}\left(1 - (w^T x^{(i)} + b)t^{(i)}\right)$$

where $\alpha^{(i)}$'s are Lagrange multipliers

First, minimize function L w.r.t. w, b for fixed Lagrange multipliers

$$\frac{\partial L(w,b,\alpha)}{\partial w} = w - \sum_{i=1}^{N} \alpha^{(i)} t^{(i)} x^{(i)} = 0 \Rightarrow w = \sum_{i=1}^{N} \alpha^{(i)} t^{(i)} x^{(i)}$$

$$\frac{\partial L(w,b,\alpha)}{\partial b} = \sum_{i=1}^{N} \alpha^{(i)} t^{(i)} = 0$$

Then, substitute w back to function L

$$L(\alpha) = \sum_{i=1}^{N} \alpha^{(i)} - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} t^{(i)} t^{(j)} \alpha^{(i)} \alpha^{(j)} \left({x^{(i)}}^T x^{(j)}\right)$$

Next, we can obtain $\alpha^{(i)}$'s by solving the following optimization problem

$$\max_{\alpha^{(i)}} L(\alpha) = \max_{\alpha^{(i)}} \sum_{i=1}^{N} \alpha^{(i)} - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} t^{(i)} t^{(j)} \boldsymbol{\alpha}^{(i)} \boldsymbol{\alpha}^{(j)} \left({x^{(i)}}^T x^{(j)}\right)$$

s.t. $\alpha^{(i)} \geq 0$ and $\sum_{i=1}^{N} \alpha^{(i)} t^{(i)} = 0$

Dual problem

Quadratic Programming

Find $\underset{\mathbf{u}}{\arg\max} \quad c + \mathbf{d}^T\mathbf{u} + \dfrac{\mathbf{u}^T R \mathbf{u}}{2}$ Quadratic criterion

Subject to

$$\left.\begin{aligned}
a_{11}u_1 + a_{12}u_2 + \ldots + a_{1m}u_m &\le b_1 \\
a_{21}u_1 + a_{22}u_2 + \ldots + a_{2m}u_m &\le b_2 \\
&\vdots \\
a_{n1}u_1 + a_{n2}u_2 + \ldots + a_{nm}u_m &\le b_n
\end{aligned}\right\}$$

n additional linear <u>in</u>equality constraints

And subject to

$$\left.\begin{aligned}
a_{(n+1)1}u_1 + a_{(n+1)2}u_2 + \ldots + a_{(n+1)m}u_m &= b_{(n+1)} \\
a_{(n+2)1}u_1 + a_{(n+2)2}u_2 + \ldots + a_{(n+2)m}u_m &= b_{(n+2)} \\
&\vdots \\
a_{(n+e)1}u_1 + a_{(n+e)2}u_2 + \ldots + a_{(n+e)m}u_m &= b_{(n+e)}
\end{aligned}\right\}$$

e additional linear <u>e</u>quality constraints

SVM

- Training a SVM model by solving

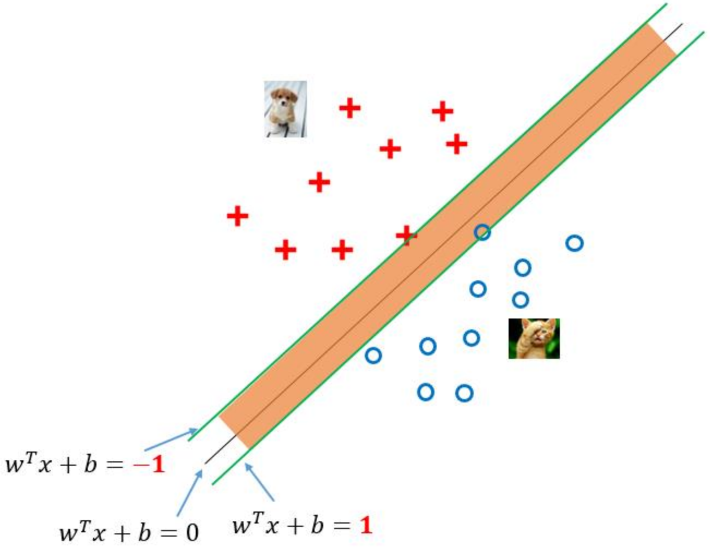
$$\max_{\alpha^{(i)}} L(\alpha) = \max_{\alpha^{(i)}} \sum_{i=1}^{N} \alpha^{(i)} - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} t^{(i)} t^{(j)} \alpha^{(i)} \alpha^{(j)} \left(x^{(i)^T} x^{(j)} \right)$$

s.t. $\alpha^{(i)} \geq 0$ and $\sum_{i=1}^{N} \alpha^{(i)} t^{(i)} = 0$

$$\min_{w,b} \frac{1}{2} \|w\|_2^2$$

$$s.t. \quad \forall i, \left(w^T x^{(i)} + b \right) t^{(i)} \geq 1$$

A constrained minimization



SVM

- Training a SVM model by solving

$$\max_{\alpha^{(i)}} L(\alpha) = \max_{\alpha^{(i)}} \sum_{i=1}^{N} \alpha^{(i)} - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} t^{(i)} t^{(j)} \alpha^{(i)} \alpha^{(j)} \left({x^{(i)}}^{T} x^{(j)} \right)$$

s.t. $\alpha^{(i)} \geq 0$ and $\sum_{i=1}^{N} \alpha^{(i)} t^{(i)} = 0$

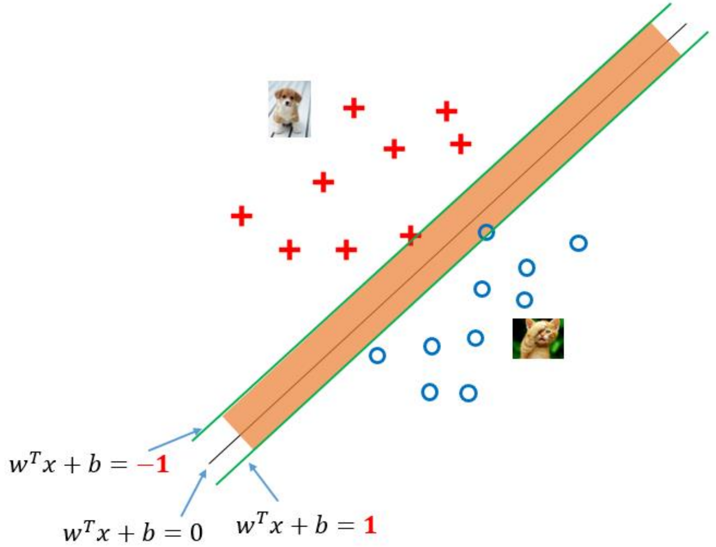
$$\min_{w,b} \frac{1}{2} \|w\|_2^2$$

$$s.t. \quad \forall i, (w^T x^{(i)} + b) t^{(i)} \geq 1$$

A constrained minimization

- Once $\alpha^{(i)}$ is obtained
  - The weights are

$$w = \sum_{i=1}^{N} \alpha^{(i)} t^{(i)} x^{(i)}$$



SVM

- Training a SVM model by solving

$$\max_{\alpha^{(i)}} L(\alpha) = \max_{\alpha^{(i)}} \sum_{i=1}^{N} \alpha^{(i)} - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} t^{(i)} t^{(j)} \alpha^{(i)} \alpha^{(j)} \left(x^{(i)^T} x^{(j)} \right)$$

s.t. $\alpha^{(i)} \geq 0$ and $\sum_{i=1}^{N} \alpha^{(i)} t^{(i)} = 0$

$$\min_{w,b} \frac{1}{2} \|w\|_2^2$$

$$s.t. \quad \forall i, \left(w^T x^{(i)} + b \right) t^{(i)} \geq 1$$

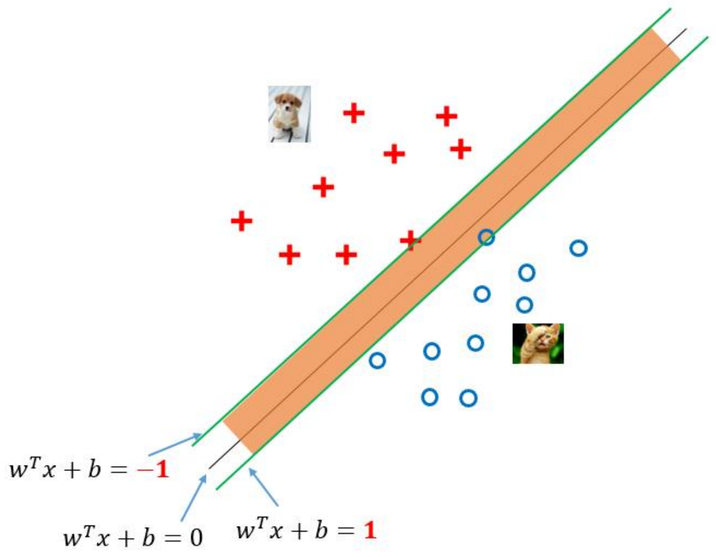
A constrained minimization

- Once $\alpha^{(i)}$ is obtained
  - The weights are

$$w = \sum_{i=1}^{N} \alpha^{(i)} t^{(i)} x^{(i)}$$

  - Prediction on a new example:

$$y^{(new)} = sgn\left(w^T x^{(new)} + b\right)$$
$$= sgn\left(\sum_{i=1}^{N} \alpha^{(i)} t^{(i)} \left(x^{(i)^T} x^{(new)} \right) + b\right)$$



Only a small subset of $\alpha^{(i)}$'s will be nonzero, and the corresponding $x^{(i)}$'s are the **Support Vectors**.

SVM

- Training a SVM model by solving

$$\max_{\alpha^{(i)}} L(\alpha) = \max_{\alpha^{(i)}} \sum_{i=1}^{N} \alpha^{(i)} - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} t^{(i)}t^{(j)}\alpha^{(i)}\alpha^{(j)}\left({x^{(i)}}^T x^{(j)}\right)$$

s.t. $\alpha^{(i)} \geq 0$ and $\sum_{i=1}^{N} \alpha^{(i)} t^{(i)} = 0$

$$\min_{w,b} \frac{1}{2}\|w\|_2^2$$

$$s.t.\quad \forall i, \left(w^T x^{(i)} + b\right) t^{(i)} \geq 1$$

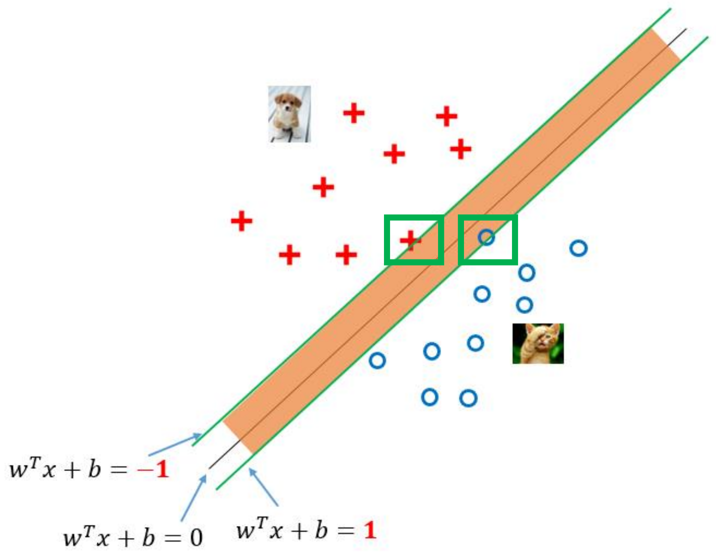
A constrained minimization

- Once $\alpha^{(i)}$ is obtained
  - The weights are

$$\boxed{w} = \sum_{i=1}^{N} \boxed{\alpha^{(i)}} t^{(i)} x^{(i)}$$

  - Prediction on a new example:

$$y^{(new)} = sgn\left(w^T x^{(new)} + b\right)$$
$$= sgn\left(\sum_{i=1}^{N} \alpha^{(i)} t^{(i)} \left({x^{(i)}}^T x^{(new)}\right) + b\right)$$



SVM

Only a small subset of $\alpha^{(i)}$'s will be nonzero, and the corresponding $x^{(i)}$'s are the **Support Vectors**.

Note that both the learning objective and the decision function depend only on ***dot products*** between datapoints

- Training a SVM model by solving

$$\max_{\alpha^{(i)}} L(\alpha) = \max_{\alpha^{(i)}} \sum_{i=1}^{N} \alpha^{(i)} - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} t^{(i)} t^{(j)} \alpha^{(i)} \alpha^{(j)} \left({x^{(i)}}^{T} x^{(j)} \right)$$

s.t. $\alpha^{(i)} \geq 0$ and $\sum_{i=1}^{N} \alpha^{(i)} t^{(i)} = 0$

$$\min_{w,b} \frac{1}{2} \|w\|_2^2$$
$$s.t. \quad \forall i, \left(w^T x^{(i)} + b \right) t^{(i)} \geq 1$$

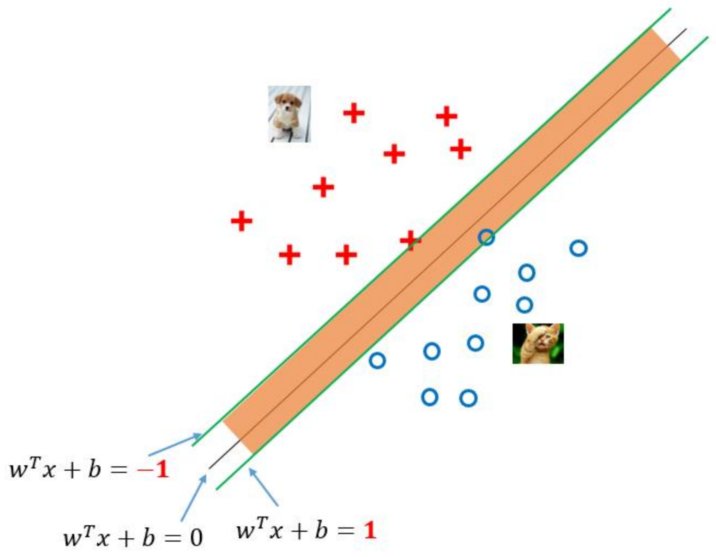
A constrained minimization

- Once $\alpha^{(i)}$ is obtained
  - The weights are

$$w = \sum_{i=1}^{N} \alpha^{(i)} t^{(i)} x^{(i)}$$

  - Prediction on a new example:

$$y^{(new)} = sgn\left(w^T x^{(new)} + b \right)$$
$$= sgn\left(\sum_{i=1}^{N} \alpha^{(i)} t^{(i)} \left({x^{(i)}}^{T} x^{(new)} \right) + b \right)$$



SVM

Only a small subset of $\alpha^{(i)}$'s will be nonzero, and the corresponding $x^{(i)}$'s are the **Support Vectors**.

Note that both the learning objective and the decision function depend only on ***dot products*** between datapoints

- Training a SVM model by solving

$$\max_{\alpha^{(i)}} L(\alpha) = \max_{\alpha^{(i)}} \sum_{i=1}^{N} \alpha^{(i)} - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} t^{(i)} t^{(j)} \alpha^{(i)} \alpha^{(j)} \left({x^{(i)}}^T x^{(j)} \right)$$

s.t. $\alpha^{(i)} \geq 0$ and $\sum_{i=1}^{N} \alpha^{(i)} t^{(i)} = 0$

svm.fit()

$$\min_{w,b} \frac{1}{2} \|w\|_2^2$$

$$s.t. \quad \forall i, \left(w^T x^{(i)} + b\right) t^{(i)} \geq 1$$

A constrained minimization

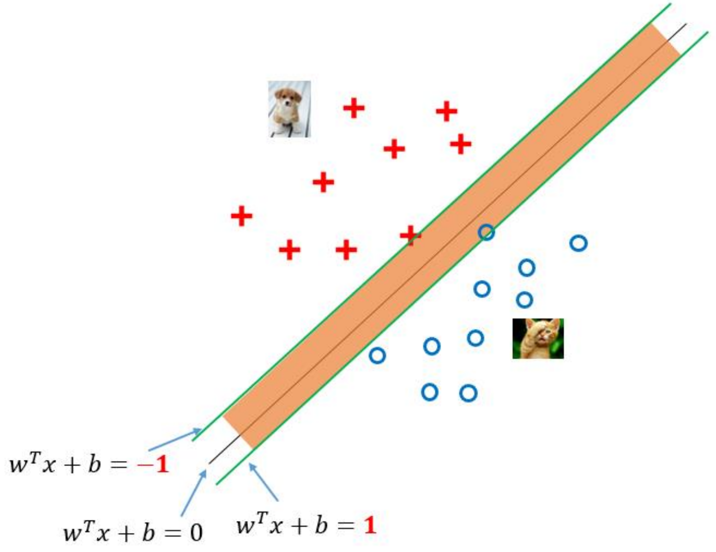
- Once $\alpha^{(i)}$ is obtained
  - The weights are

$$w = \sum_{i=1}^{N} \alpha^{(i)} t^{(i)} x^{(i)}$$

  - Prediction on a new example:

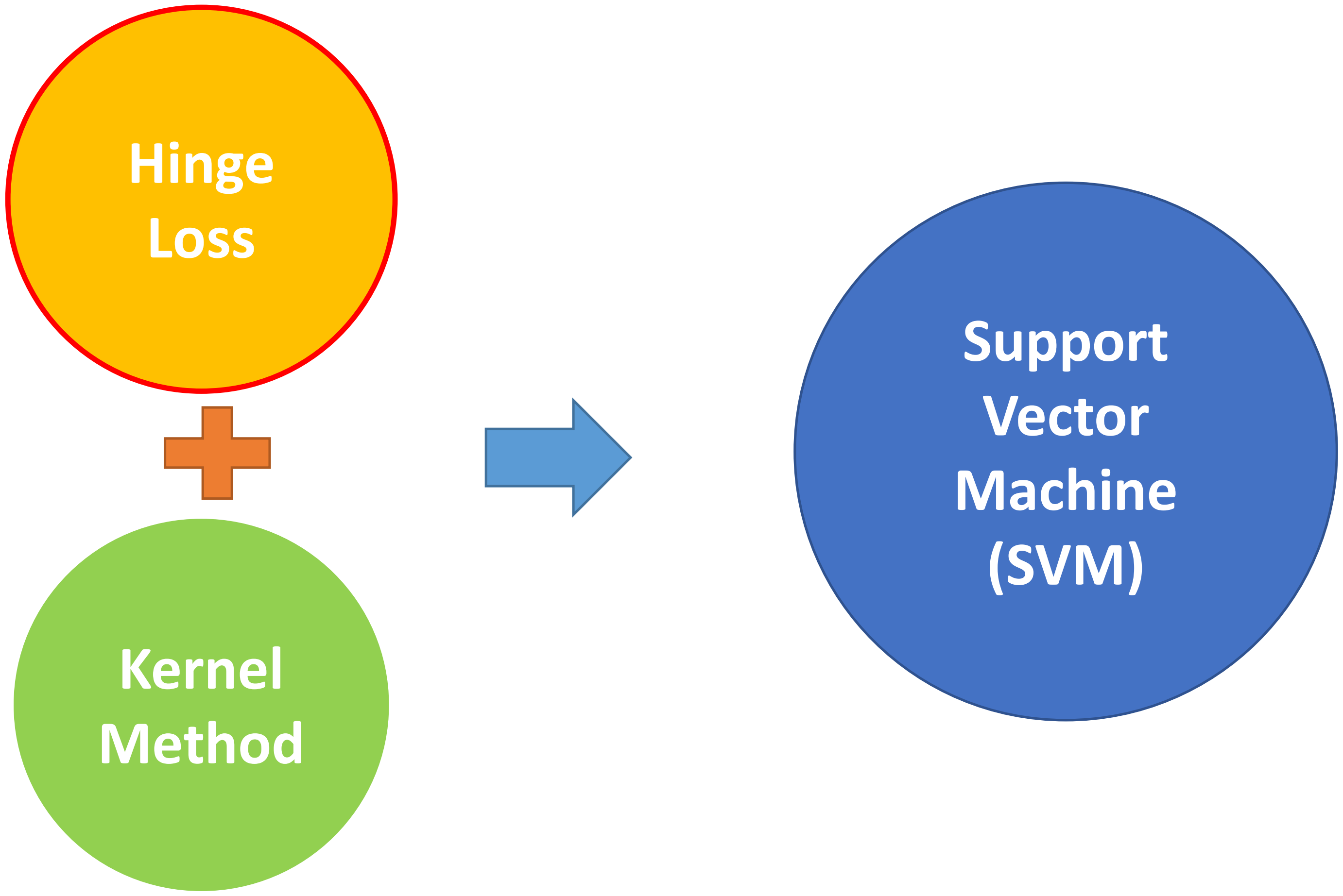
$$y^{(new)} = sgn\left(w^T x^{(new)} + b\right)$$
$$= sgn\left(\sum_{i=1}^{N} \alpha^{(i)} t^{(i)} \left({x^{(i)}}^T x^{(new)} \right) + b\right)$$

svm.predict()

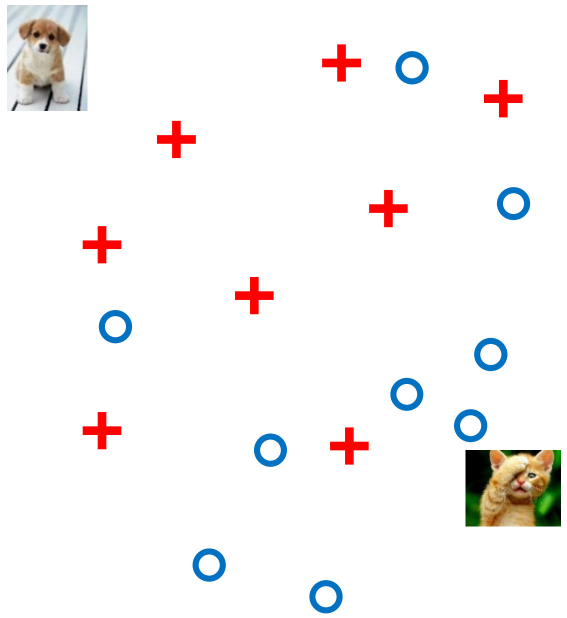


SVM

- Linear SVM
- Soft Margin (Non-linearly separable)
  - Hinge Loss
- Kernel Trick

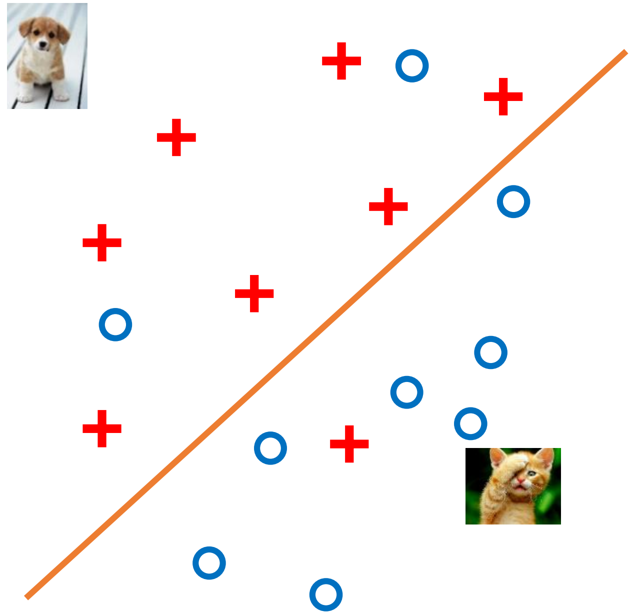


Non-Separable



Non-Separable

- What should we do?

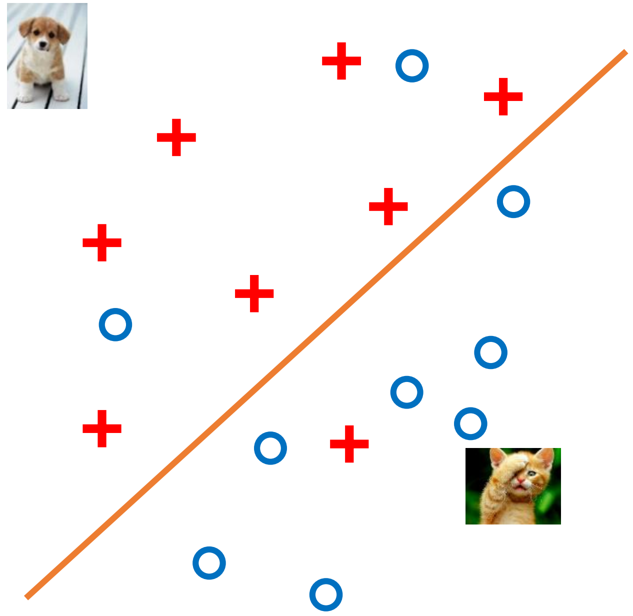


Non-Separable

- What should we do?

$$\min_{w,b} \frac{1}{2}\|w\|_2^2 + \color{red}{penalty}$$

Balance the tradeoff between margin and classification errors



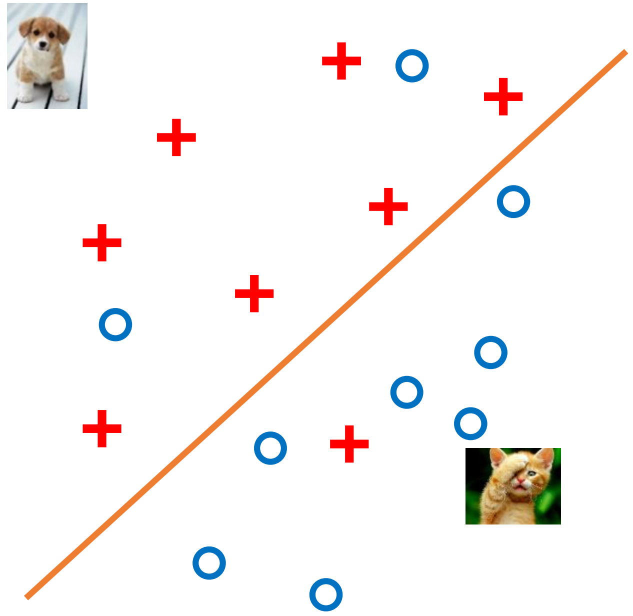
Non-Separable

- What should we do?
  - Idea 1:

$$\min_{w,b} \frac{1}{2}\|w\|_2^2 + C \times (\#train\ errors)$$

  - Idea 2:

$$\min_{w,b} \frac{1}{2}\|w\|_2^2 + C \times (distance\ of\ error\ points\ to\ their\ corret\ place)$$



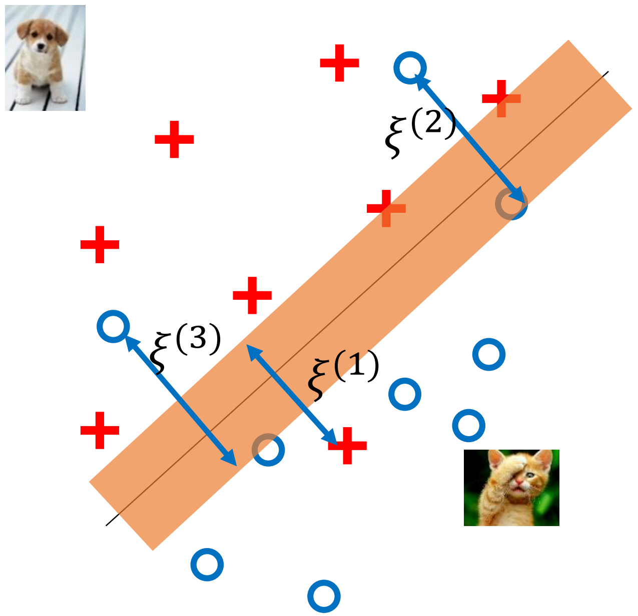
Non-Separable

- What should we do?
  - Idea 1:

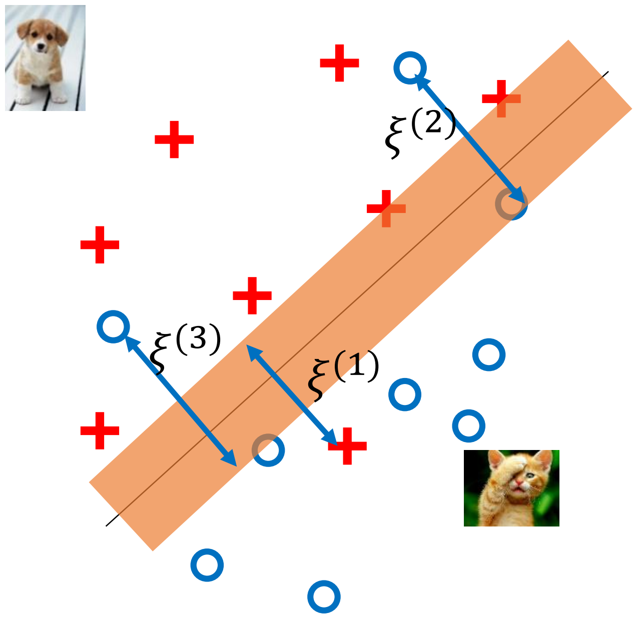
$$\min_{w,b} \frac{1}{2}\|w\|_2^2 + C \times (\#train\ errors)$$

  - Idea 2:

$$\min_{w,b} \frac{1}{2}\|w\|_2^2 + C \times (distance\ of\ error\ points\ to\ their\ corret\ place)$$



Non-Separable



- What should we do?
  - Idea 1:

$$\min_{w,b} \frac{1}{2}\|w\|_2^2 + C \times (\#train\ errors)$$

  - Idea 2:

$$\min_{w,b} \frac{1}{2}\|w\|_2^2 + C \sum_{i=1}^{N} \xi^{(i)}$$

$$s.t.\ \left(w^T \boldsymbol{x}^{(1)} + b\right)t^{(1)} \geq 1 - \xi^{(1)}, \xi^{(1)} \geq 0$$

$$\left(w^T \boldsymbol{x}^{(2)} + b\right)t^{(2)} \geq 1 - \xi^{(2)}, \xi^{(2)} \geq 0$$

$$\vdots$$

$$\left(w^T \boldsymbol{x}^{(N)} + b\right)t^{(N)} \geq 1 - \xi^{(N)}, \xi^{(N)} \geq 0$$

Non-Separable

- What should we do?
  - Idea 1:

$$\min_{w,b} \frac{1}{2}\|w\|_2^2 + C \times (\#train\ errors)$$

  - Idea 2:

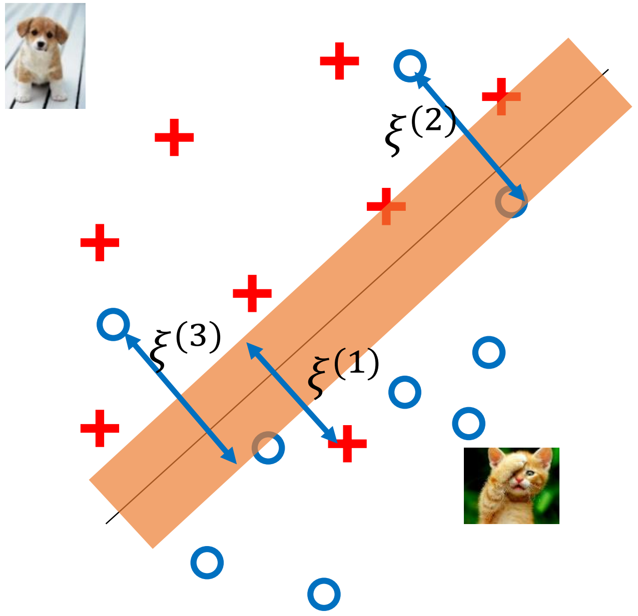
$$\min_{w,b} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N}\xi^{(i)}$$

$$s.t.\ \left(w^T\boldsymbol{x}^{(1)} + b\right)t^{(1)} \geq 1 - \xi^{(1)}, \xi^{(1)} \geq 0$$

$$\left(w^T\boldsymbol{x}^{(2)} + b\right)t^{(2)} \geq 1 - \xi^{(2)}, \xi^{(2)} \geq 0$$

$$\vdots$$

$$\left(w^T\boldsymbol{x}^{(N)} + b\right)t^{(N)} \geq 1 - \xi^{(N)}, \xi^{(N)} \geq 0$$



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

Non-Separable

- What should we do?
  - Idea 1:

$$\min_{w,b} \frac{1}{2}\|w\|_2^2 + C \times (\#train\ errors)$$

  - Idea 2:

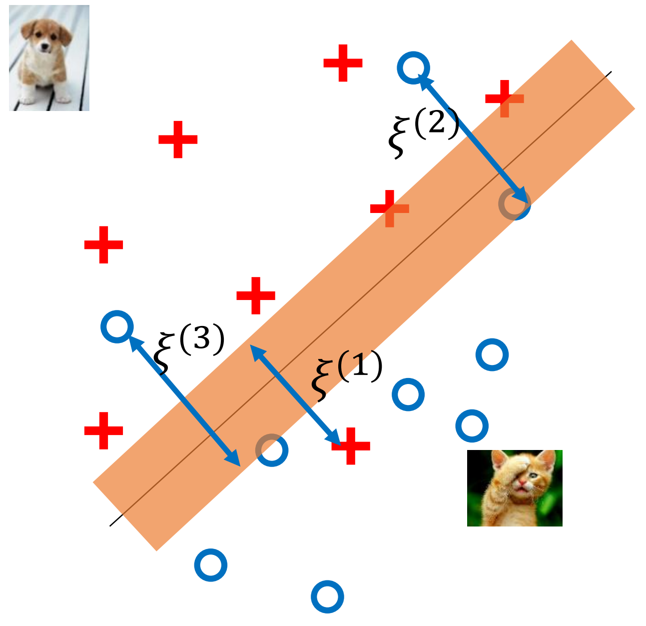
$$\min_{w,b} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N}\xi^{(i)}$$

$$s.t.\ \left(w^T\boldsymbol{x}^{(1)} + b\right)t^{(1)} \geq 1 - \xi^{(1)}, \xi^{(1)} \geq 0$$

$$\left(w^T\boldsymbol{x}^{(2)} + b\right)t^{(2)} \geq 1 - \xi^{(2)}, \xi^{(2)} \geq 0$$

$$\vdots$$

$$\left(w^T\boldsymbol{x}^{(N)} + b\right)t^{(N)} \geq 1 - \xi^{(N)}, \xi^{(N)} \geq 0$$



Soft margin

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

SVM

- **Linear SVM with slack variables**
  - A maximum margin classifier
  - Considering the tradeoff between margin and classification errors

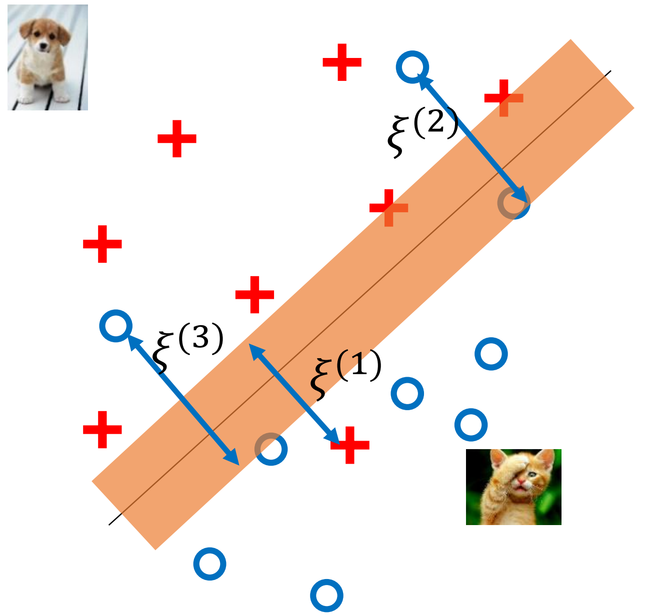
$$\min_{w,b} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N} \xi^{(i)}$$

$$s.t.\ \ \forall i\,, \left(w^T x^{(i)} + b\right) t^{(i)} \geq 1 - \xi^{(i)}$$

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

A constrained minimization

Soft margin



Three Steps for SVM

- **Learning**

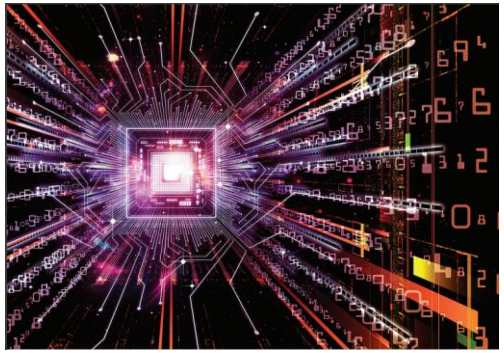
Representation ⇨ **Evaluation** ⇨ **Optimization**

$y(x) = w^T x + b$

$$\min_{w,b} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N} \xi^{(i)}$$

$$s.t.\quad \forall i\,,\, (w^T x^{(i)} + b\,)t^{(i)} \geq 1 - \xi^{(i)}$$

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



Reference: Domingos, Pedro. "A few useful things to know about machine learning." *Communications of the ACM* 55.10 (2012): 78-87.

Dual problem

$$\min_{w,b} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N} \xi^{(i)}$$

$$s.t.\ \ \forall i, \left(w^T x^{(i)} + b\right) t^{(i)} \geq 1 - \xi^{(i)}$$

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

A constrained minimization

Lagrange function

$$L(w,b,\xi,\alpha,\eta)=\frac{1}{2}\|w\|_2^2+C\sum_{i=1}^{N}\xi^{(i)}+\sum_{i=1}^{N}\alpha^{(i)}\left(1-\xi^{(i)}-\left(w^Tx^{(i)}+b\right)t^{(i)}\right)-\sum_{i=1}^{N}\eta^{(i)}\xi^{(i)}$$

where $\alpha^{(i)}$'s and $\eta^{(i)}$'s are Lagrange multipliers

Dual problem

$$\min_{w,b}\frac{1}{2}\|w\|_2^2+C\sum_{i=1}^{N}\xi^{(i)}$$

$$s.t.\quad \forall i,\left(w^Tx^{(i)}+b\right)t^{(i)}\geq 1-\xi^{(i)}$$

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

A constrained minimization

Lagrange function

$$L(w,b,\xi,\alpha,\eta) = \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N}\xi^{(i)} + \sum_{i=1}^{N}\alpha^{(i)}\left(1-\xi^{(i)} - \left(w^T x^{(i)} + b\right)t^{(i)}\right) - \sum_{i=1}^{N}\eta^{(i)}\xi^{(i)}$$

where $\alpha^{(i)}$'s and $\eta^{(i)}$'s are Lagrange multipliers

First, minimize function L w.r.t. $w, b, \xi^{(i)}$ for fixed Lagrange multipliers

$$\frac{\partial L(w,b,\xi,\alpha,\eta)}{\partial w} = w - \sum_{i=1}^{N}\alpha^{(i)}t^{(i)}x^{(i)} = 0$$

$$\frac{\partial L(w,b,\xi,\alpha,\eta)}{\partial b} = \sum_{i=1}^{N}\alpha^{(i)}t^{(i)} = 0 \qquad \frac{\partial L(w,b,\xi,\alpha,\eta)}{\partial \xi^{(i)}} = C - \alpha^{(i)} - \eta^{(i)} = 0$$

Dual problem

$$\min_{w,b} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N}\xi^{(i)}$$

$$s.t. \quad \forall i, \left(w^T x^{(i)} + b\right)t^{(i)} \geq 1 - \xi^{(i)}$$

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

A constrained minimization

Lagrange function

$$L(w,b,\xi,\alpha,\eta) = \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N}\xi^{(i)} + \sum_{i=1}^{N}\alpha^{(i)}\left(1-\xi^{(i)}-\left(w^T x^{(i)}+b\right)t^{(i)}\right) - \sum_{i=1}^{N}\eta^{(i)}\xi^{(i)}$$

where $\alpha^{(i)}$'s and $\eta^{(i)}$'s are Lagrange multipliers

First, minimize function L w.r.t. $w, b, \xi^{(i)}$ for fixed Lagrange multipliers

$$\frac{\partial L(w,b,\xi,\alpha,\eta)}{\partial w} = w - \sum_{i=1}^{N}\alpha^{(i)}t^{(i)}x^{(i)} = 0 \quad \Rightarrow \quad w = \sum_{i=1}^{N}\alpha^{(i)}t^{(i)}x^{(i)}$$

$$\frac{\partial L(w,b,\xi,\alpha,\eta)}{\partial b} = \sum_{i=1}^{N}\alpha^{(i)}t^{(i)} = 0 \qquad \frac{\partial L(w,b,\xi,\alpha,\eta)}{\partial \xi^{(i)}} = C - \alpha^{(i)} - \eta^{(i)} = 0$$

Dual problem

$$\min_{w,b} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N}\xi^{(i)}$$

$$s.t.\ \ \forall i, \left(w^T x^{(i)}+b\right)t^{(i)} \ge 1-\xi^{(i)}$$

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

A constrained minimization

Lagrange function

$$L(w,b,\xi,\alpha,\eta)=\frac{1}{2}\|w\|_2^2+C\sum_{i=1}^{N}\xi^{(i)}+\sum_{i=1}^{N}\alpha^{(i)}\left(1-\xi^{(i)}-\left(w^T x^{(i)}+b\right)t^{(i)}\right)-\sum_{i=1}^{N}\eta^{(i)}\xi^{(i)}$$

where $\alpha^{(i)}$'s and $\eta^{(i)}$'s are Lagrange multipliers

First, minimize function L w.r.t. $w, b, \xi^{(i)}$ for fixed Lagrange multipliers

$$\frac{\partial L(w,b,\xi,\alpha,\eta)}{\partial w}=w-\sum_{i=1}^{N}\alpha^{(i)}t^{(i)}x^{(i)}=0 \quad \Rightarrow \quad w=\sum_{i=1}^{N}\alpha^{(i)}t^{(i)}x^{(i)}$$

$$\frac{\partial L(w,b,\xi,\alpha,\eta)}{\partial b}=\sum_{i=1}^{N}\alpha^{(i)}t^{(i)}=0 \qquad \frac{\partial L(w,b,\xi,\alpha,\eta)}{\partial \xi^{(i)}}=C-\alpha^{(i)}-\eta^{(i)}=0$$

Then, substitute w back to function L

$$L(\alpha)=\sum_{i=1}^{N}\alpha^{(i)}-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}t^{(i)}t^{(j)}\alpha^{(i)}\alpha^{(j)}\left({x^{(i)}}^T x^{(j)}\right)$$

Dual problem

$$\min_{w,b}\ \frac{1}{2}\|w\|_2^2+C\sum_{i=1}^{N}\xi^{(i)}$$

$$s.t.\ \ \forall i,\ \left(w^T x^{(i)}+b\right)t^{(i)}\ge 1-\xi^{(i)}$$

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

A constrained minimization

Lagrange function

$$L(w, b, \xi, \alpha, \eta) = \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N} \xi^{(i)} + \sum_{i=1}^{N} \alpha^{(i)}\left(1 - \xi^{(i)} - \left(w^T x^{(i)} + b\right)t^{(i)}\right) - \sum_{i=1}^{N} \eta^{(i)}\xi^{(i)}$$

where $\alpha^{(i)}$'s and $\eta^{(i)}$'s are Lagrange multipliers

First, minimize function L w.r.t. $w, b, \xi^{(i)}$ for fixed Lagrange multipliers

$$\frac{\partial L(w, b, \xi, \alpha, \eta)}{\partial w} = w - \sum_{i=1}^{N} \alpha^{(i)} t^{(i)} x^{(i)} = 0 \quad \Rightarrow \quad w = \sum_{i=1}^{N} \alpha^{(i)} t^{(i)} x^{(i)}$$

$$\frac{\partial L(w, b, \xi, \alpha, \eta)}{\partial b} = \sum_{i=1}^{N} \alpha^{(i)} t^{(i)} = 0 \qquad \frac{\partial L(w, b, \xi, \alpha, \eta)}{\partial \xi^{(i)}} = C - \alpha^{(i)} - \eta^{(i)} = 0$$

Then, substitute w back to function L

$$L(\alpha) = \sum_{i=1}^{N} \alpha^{(i)} - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} t^{(i)} t^{(j)} \alpha^{(i)} \alpha^{(j)} \left({x^{(i)}}^T x^{(j)}\right)$$

Next, we can obtain $\alpha^{(i)}$'s by solving the following optimization problem

$$\max_{\alpha^{(i)}} L(\alpha) = \max_{\alpha^{(i)}} \sum_{i=1}^{N} \alpha^{(i)} - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} t^{(i)} t^{(j)} \alpha^{(i)} \alpha^{(j)} \left({x^{(i)}}^T x^{(j)}\right)$$

s.t. $\alpha^{(i)} \in [0, C]$ and $\sum_{i=1}^{N} \alpha^{(i)} t^{(i)} = 0$

Dual problem

$$\min_{w,b} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N} \xi^{(i)}$$

$$s.t.\ \ \forall i, \left(w^T x^{(i)} + b\right)t^{(i)} \geq 1 - \xi^{(i)}$$

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

A constrained minimization

SVM

- Let's view SVM in another way

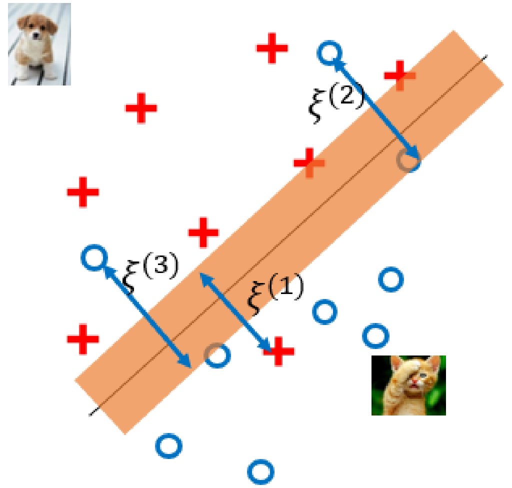
$$\min_{w,b} \frac{1}{2}\|w\|_2^2 + penalty$$

$$\min_{w,b} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N}\xi^{(i)}$$

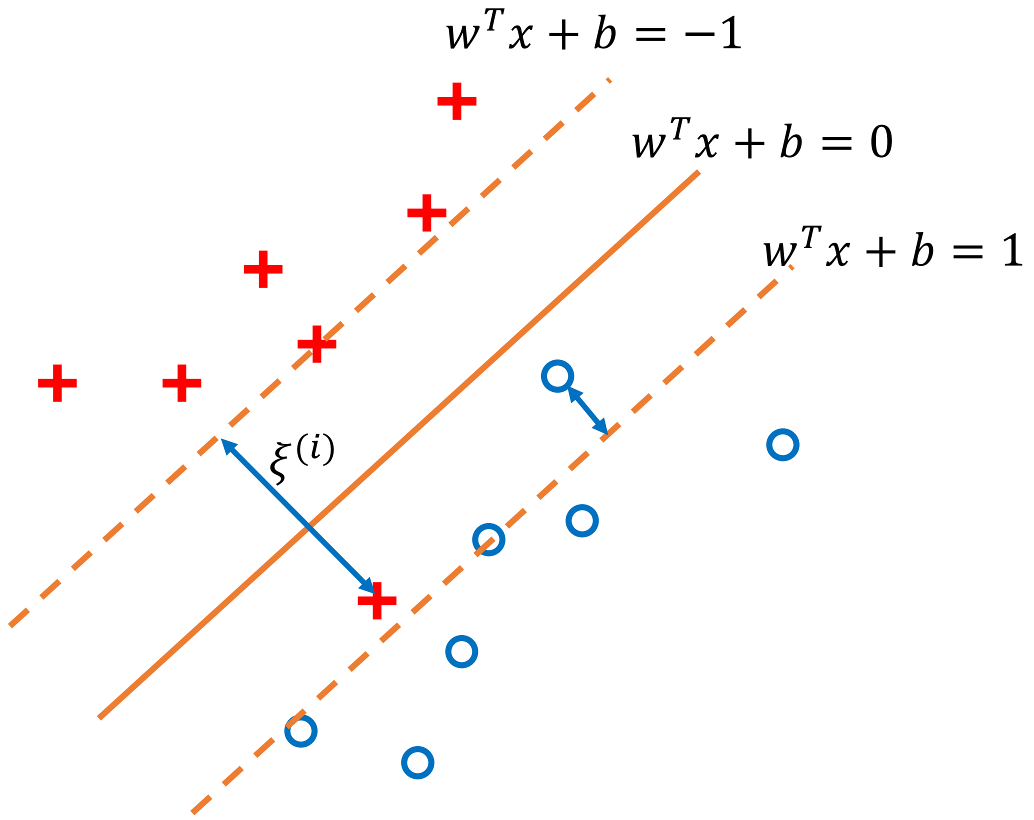
$$s.t.\ \ \forall i,\ (w^T x^{(i)} + b)t^{(i)} \geq 1 - \xi^{(i)}$$

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

A constrained minimization



Non-Separable



- **Case 1:** $\boldsymbol{\xi^{(i)} \geq 1}$

  $Misclassification$

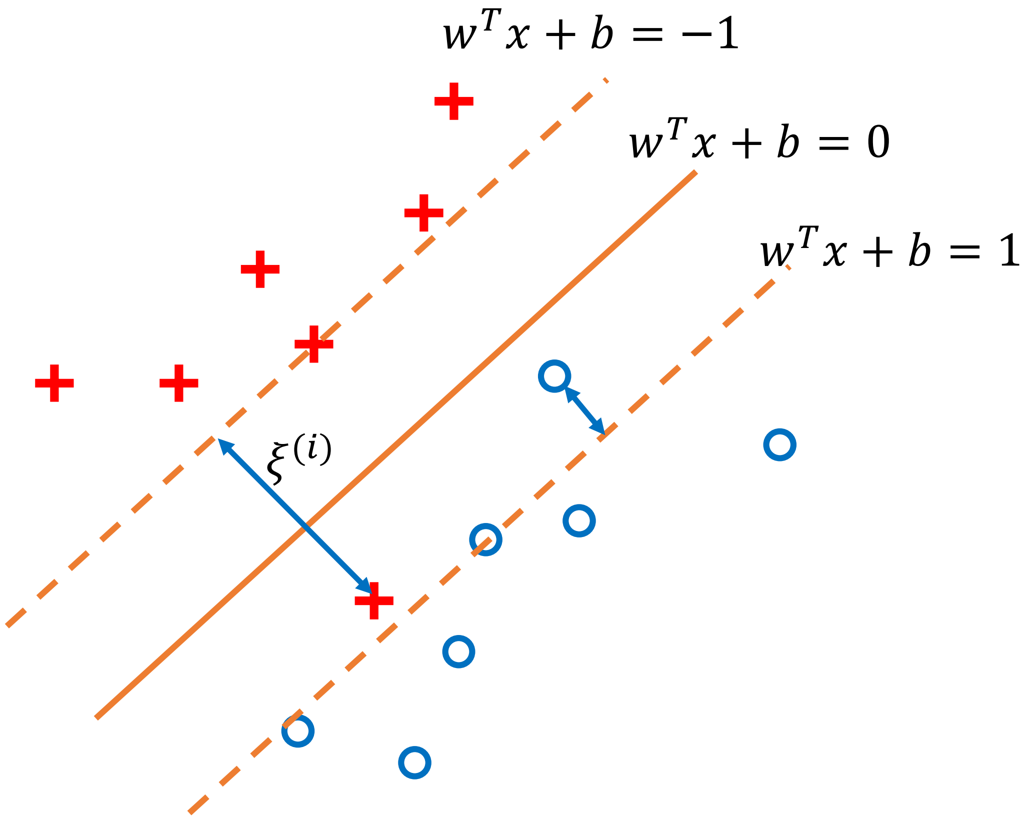
- **Case 2:** $\boldsymbol{0 < \xi^{(i)} < 1}$

  $x^{(i)}$ $is\ correctly\ classificed, but\ lies\ inside\ the\ margin$

- **Case 3:** $\boldsymbol{\xi^{(i)} = 0}$

  $x^{(i)}$ $is\ correctly\ classificed, but\ lies\ outside\ the\ margin$

Non-Separable



Hinge loss: $max(0, 1 - t \cdot y)$

$$\varepsilon^{(i)} = max\left(0, 1 - \left(w^T x^{(i)} + b\right)t^{(i)}\right)$$

- **Case 1:** $\xi^{(i)} \geq 1$

  $Misclassification$

- **Case 2:** $0 < \xi^{(i)} < 1$

  $x^{(i)}$ *is correctly classificed, but lies inside the margin*

- **Case 3:** $\xi^{(i)} = 0$

  $x^{(i)}$ *is correctly classificed, but lies outside the margin*

SVM

- Let's view SVM in another way

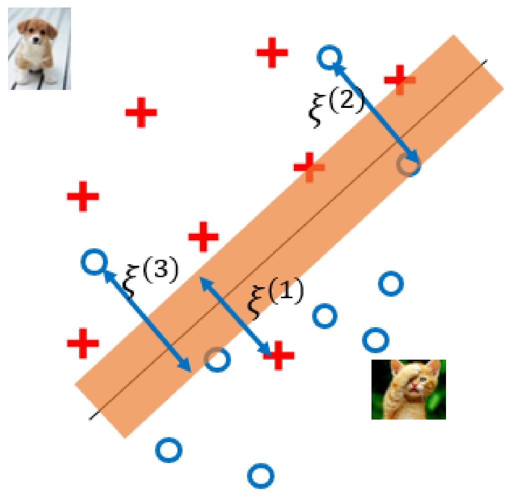
$$\min_{w,b} \frac{1}{2}\|w\|_2^2 + penalty$$

$$\min_{w,b} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N} \xi^{(i)}$$

$$s.t.\quad \forall i, (w^T x^{(i)} + b)t^{(i)} \geq 1 - \xi^{(i)}$$

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

A constrained minimization



SVM

- Let's view SVM in another way

$$\min_{w,b} \frac{1}{2}\|w\|_2^2 + penalty$$

- For training data $\{(x^{(i)}, t^{(i)})\}_{i=1}^{N}$, use the following penalty

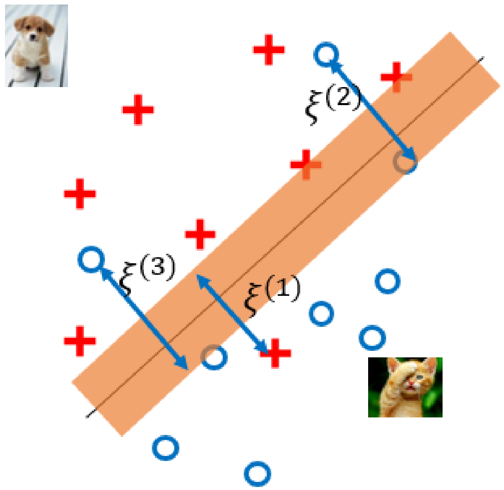
$$max\left(0, 1-\left(w^T x^{(i)} + b\right)t^{(i)}\right)$$

$$\min_{w,b} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N} \xi^{(i)}$$

$$s.t.\quad \forall i, \left(w^T x^{(i)} + b\right)t^{(i)} \geq 1 - \xi^{(i)}$$

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

A constrained minimization



SVM

$$\min_{w,b} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N}\xi^{(i)}$$

$$s.t.\quad \forall i,\ (w^T x^{(i)} + b)t^{(i)} \geq 1 - \xi^{(i)}$$

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

A constrained minimization

- Let's view SVM in another way

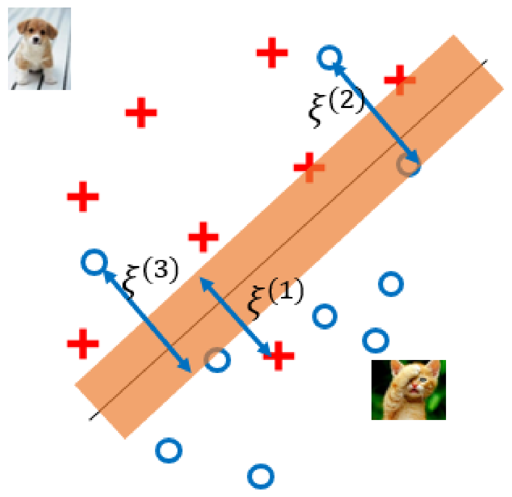
$$\min_{w,b} \frac{1}{2}\|w\|_2^2 + penalty$$

- For training data $\{(x^{(i)}, t^{(i)})\}_{i=1}^{N}$, use the following penalty

$$max(0, 1 - (w^T x^{(i)} + b)t^{(i)})$$

- So we can define a loss function as

$$\ell(w,b) = \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N} max(0, 1 - (w^T x^{(i)} + b)t^{(i)})$$



Logistic Regression

- Training datasets
  $\mathcal{D} = \{(x^{(1)}, t^{(1)}), \dots, (x^{(i)}, t^{(i)}), \dots, (x^{(N)}, t^{(N)})\}$

  (target $t^{(i)}$: 0 or 1)

- Linear model
  $y(x) = \sigma(w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_M x_M)$

- Parameters

  $w_0, w_1, w_2, \dots, w_M$

- Loss function

$$\ell(w) = -\frac{1}{N}\sum_{i=1}^{N}\left[t^{(i)}\log\left(y\left(x^{(i)}\right)\right) + \left(1 - t^{(i)}\right)\log\left(1 - y\left(x^{(i)}\right)\right)\right]$$

- Goal: minimize $\ell(w)$

Steps:
- Initialize w (e.g., randomly)
- Repeatedly update w based on the gradient

$$w = w - \epsilon\, \nabla_w \ell(w)$$

where ϵ is the learning rate.

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

Logistic Regression

- Training datasets
  $\mathcal{D} = \{(x^{(1)}, t^{(1)}), \dots, (x^{(i)}, t^{(i)}), \dots, (x^{(N)}, t^{(N)})\}$
  (target $t^{(i)}$: 0 or 1)
- Linear model
  $y(x) = \sigma(w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_M x_M)$
- Parameters
  $w_0, w_1, w_2, \dots, w_M$
- Loss function

$$\ell(w) = -\frac{1}{N}\sum_{i=1}^{N}\left[t^{(i)}\log\left(y\left(x^{(i)}\right)\right) + \left(1 - t^{(i)}\right)\log\left(1 - y\left(x^{(i)}\right)\right)\right]$$

- Goal: minimize $\ell(w)$

Steps:
- Initialize w (e.g., randomly)
- Repeatedly update w based on the gradient
  $$w = w - \epsilon \nabla_w \ell(w)$$
  where ϵ is the learning rate.

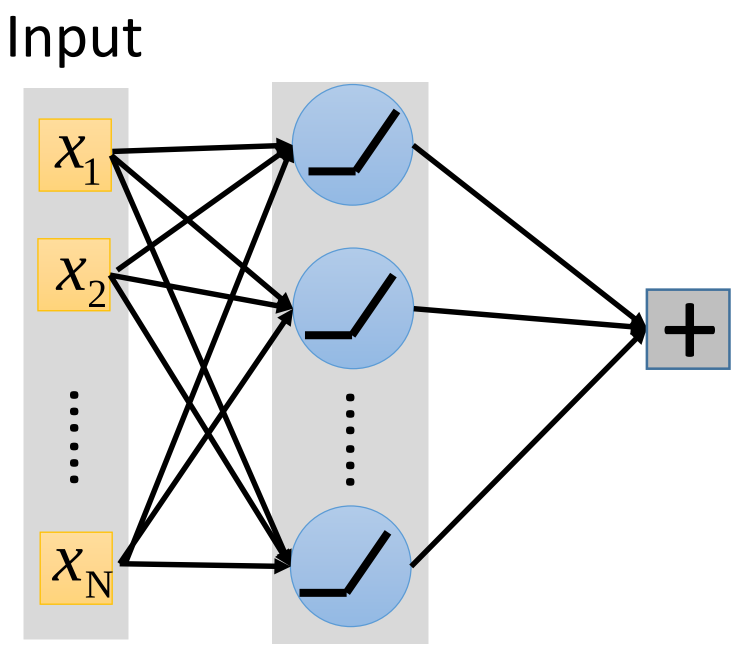
$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Linear SVM

- Training datasets
  $\mathcal{D} = \{(x^{(1)}, t^{(1)}), \dots, (x^{(i)}, t^{(i)}), \dots, (x^{(N)}, t^{(N)})\}$
  (target $t^{(i)}$: 0 or 1)
- Linear model
  $y(x) = b + w_1 x_1 + w_2 x_2 + \cdots + w_M x_M$
- Parameters
  $b, w_1, w_2, \dots, w_M$
- Loss function

$$\ell(w, b) = \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N} max\left(0, 1 - \left(w^T x^{(i)} + b\right)t^{(i)}\right)$$

- Goal: minimize $\ell(w)$

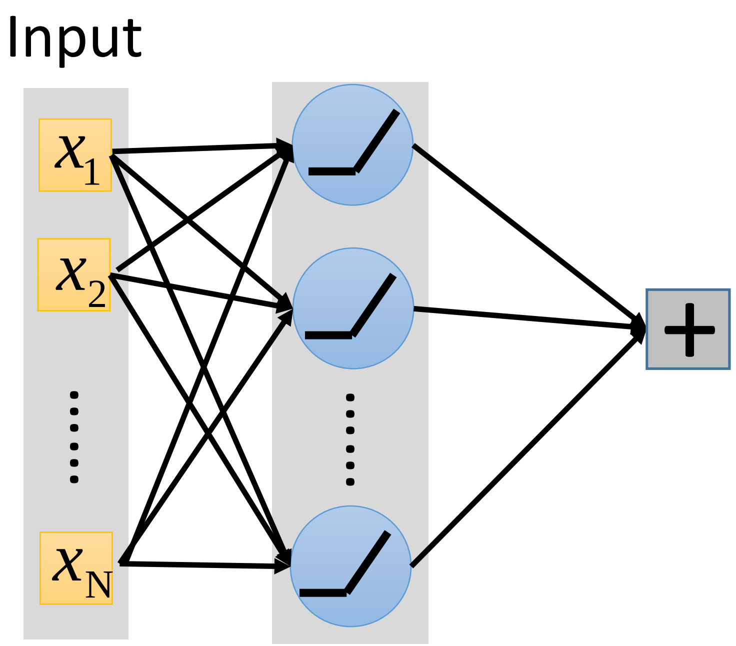


Linear SVM

- Training datasets
  $\mathcal{D} = \{(x^{(1)}, t^{(1)}), \dots, (x^{(i)}, t^{(i)}), \dots, (x^{(N)}, t^{(N)})\}$
  (target $t^{(i)}$: 0 or 1)
- Linear model
  $y(x) = b + w_1 x_1 + w_2 x_2 + \cdots + w_M x_M$
- Parameters
  $b, w_1, w_2, \dots, w_M$
- Loss function

$$\ell(w, b) = \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N} max\left(0, 1 - \left(w^T x^{(i)} + b\right)t^{(i)}\right)$$

- Goal: minimize $\ell(w)$



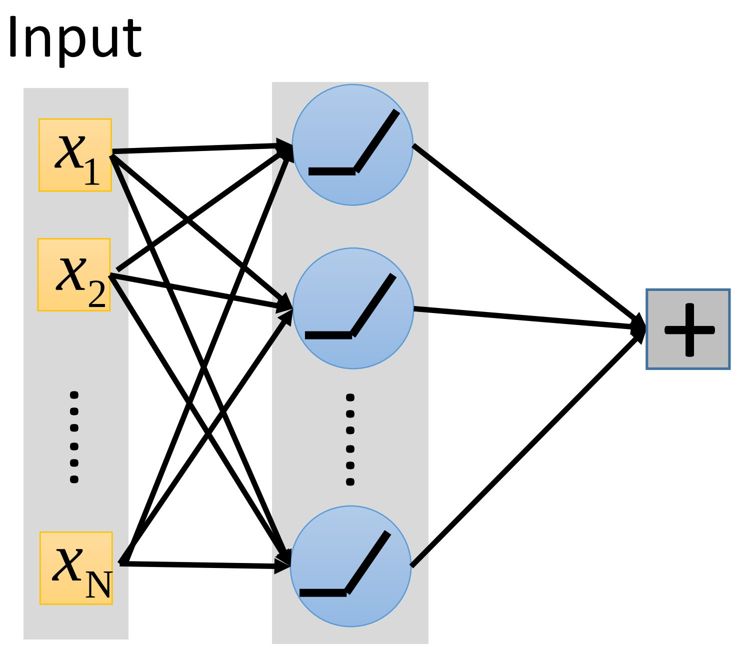
Recall ReLU, Maxout Network

Linear SVM

- Training datasets
  $\mathcal{D} = \{(x^{(1)}, t^{(1)}), \dots, (x^{(i)}, t^{(i)}), \dots, (x^{(N)}, t^{(N)})\}$
  (target $t^{(i)}$: 0 or 1)
- Linear model
  $y(x) = b + w_1 x_1 + w_2 x_2 + \cdots + w_M x_M$
- Parameters
  $b, w_1, w_2, \dots, w_M$
- Loss function

$$\ell(w, b) = \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N} max\left(0, 1 - \left(w^T x^{(i)} + b\right)t^{(i)}\right)$$

- Goal: minimize $\ell(w)$



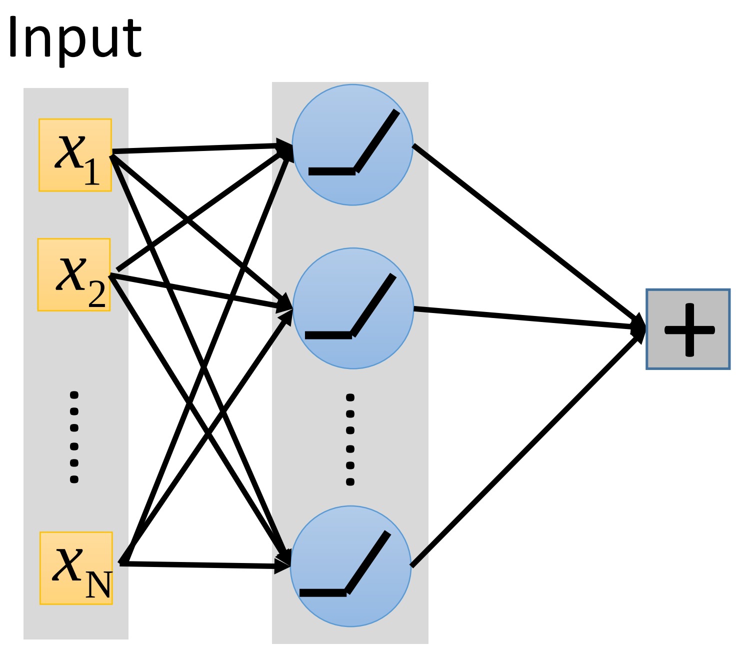
Recall ReLU, Maxout Network

Linear SVM

- Training datasets
  $\mathcal{D} = \{(x^{(1)}, t^{(1)}), \dots, (x^{(i)}, t^{(i)}), \dots, (x^{(N)}, t^{(N)})\}$
  (target $t^{(i)}$: 0 or 1)
- Linear model
  $y(x) = b + w_1 x_1 + w_2 x_2 + \cdots + w_M x_M$
- Parameters
  $b, w_1, w_2, \dots, w_M$
- Loss function

$$\ell(w,b) = \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N} max\left(0, 1 - \left(w^T x^{(i)} + b\right)t^{(i)}\right)$$

- Goal: minimize $\ell(w)$



Recall ReLU, Maxout Network

Linear SVM

- Training datasets

  $$\mathcal{D} = \{(x^{(1)}, t^{(1)}), \dots, (x^{(i)}, t^{(i)}), \dots, (x^{(N)}, t^{(N)})\}$$

  (target $t^{(i)}$: 0 or 1)

- Linear model

  $$y(x) = b + w_1 x_1 + w_2 x_2 + \cdots + w_M x_M$$

- Parameters

  $$b, w_1, w_2, \dots, w_M$$

- Loss function

$$\ell(w, b) = \frac{1}{2}\|w\|_2^2 + C \sum_{i=1}^{N} max\left(0, 1 - \left(w^T x^{(i)} + b\right) t^{(i)}\right)$$

- Goal: minimize $\ell(w)$

Tang, Yichuan. "Deep learning using linear support vector machines." *arXiv preprint arXiv:1306.0239* (2013).

Logistic Regression

- Training datasets
  $\mathcal{D} = \{(x^{(1)}, t^{(1)}), \dots, (x^{(i)}, t^{(i)}), \dots, (x^{(N)}, t^{(N)})\}$

  (target $t^{(i)}$: 0 or 1)
- Linear model
  $y(x) = \sigma(w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_M x_M)$
- Parameters

  $w_0, w_1, w_2, \dots, w_M$
- Loss function

$$\ell(w) = -\frac{1}{N}\sum_{i=1}^{N}\left[t^{(i)}\log\left(y\left(x^{(i)}\right)\right) + \left(1 - t^{(i)}\right)\log\left(1 - y\left(x^{(i)}\right)\right)\right]$$

- Goal: minimize $\ell(w)$

Steps:
- Initialize w (e.g., randomly)
- Repeatedly update w based on the gradient

$$w = w - \epsilon \nabla_w \ell(w)$$

where ϵ is the learning rate.

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

Linear SVM

- Training datasets
  $\mathcal{D} = \{(x^{(1)}, t^{(1)}), \dots, (x^{(i)}, t^{(i)}), \dots, (x^{(N)}, t^{(N)})\}$

  (target $t^{(i)}$: 0 or 1)
- Linear model
  $y(x) = b + w_1 x_1 + w_2 x_2 + \cdots + w_M x_M$
- Parameters

  $b, w_1, w_2, \dots, w_M$
- Loss function

$$\ell(w, b) = \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N} max\left(0, 1 - \left(w^T x^{(i)} + b\right)t^{(i)}\right)$$

- Goal: minimize $\ell(w)$

Quadratic Programming

Logistic Regression

- Training datasets
  $\mathcal{D} = \{(x^{(1)}, t^{(1)}), \dots, (x^{(i)}, t^{(i)}), \dots, (x^{(N)}, t^{(N)})\}$
  (target $t^{(i)}$: 0 or 1)
- Linear model
  $y(x) = \sigma(w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_M x_M)$
- Parameters
  $w_0, w_1, w_2, \dots, w_M$
- Loss function

$$\ell(w) = -\frac{1}{N}\sum_{i=1}^{N}\left[t^{(i)}\log\left(y\left(x^{(i)}\right)\right) + \left(1 - t^{(i)}\right)\log\left(1 - y\left(x^{(i)}\right)\right)\right]$$

- Goal: minimize $\ell(w)$

Steps:
- Initialize w (e.g., randomly)
- Repeatedly update w based on the gradient
  $$w = w - \epsilon \nabla_w \ell(w)$$
  where ϵ is the learning rate.

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

Linear SVM

- Training datasets
  $\mathcal{D} = \{(x^{(1)}, t^{(1)}), \dots, (x^{(i)}, t^{(i)}), \dots, (x^{(N)}, t^{(N)})\}$
  (target $t^{(i)}$: 0 or 1)
- Linear model
  $y(x) = b + w_1 x_1 + w_2 x_2 + \cdots + w_M x_M$
- Parameters
  $b, w_1, w_2, \dots, w_M$
- Loss function

$$\ell(w, b) = \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N} max\left(0, 1 - \left(w^T x^{(i)} + b\right)t^{(i)}\right)$$

- Goal: minimize $\ell(w)$

But, we can also use gradient descent
- Initialize w, b (e.g., randomly)
- Repeatedly update w based on the gradient
  $$w = w - \epsilon \nabla_w \ell(w, \mathrm{b})$$
  $$\mathrm{b} = \mathrm{b} - \epsilon \nabla_{\mathrm{b}} \ell(w, b)$$
  where ϵ is the learning rate.

$$\ell(w,b) = \frac{1}{2}\|w\|_2^2 \;\; + C\sum_{i=1}^{N} max\left(0, 1 - \left(w^T x^{(i)} + b\right)t^{(i)}\right)$$

$$\ell(w,b) = \frac{1}{2}\|w\|_2^2 \; + C\sum_{i=1}^{N} max\left(0, 1 - \left(w^T x^{(i)} + b\right)t^{(i)}\right)$$

$$\Rightarrow \quad \frac{\partial \ell(w,b)}{\partial w} = w - C\sum_{i=1}^{N} \alpha^{(i)} t^{(i)} x^{(i)}$$

where $\alpha^{(i)} = \begin{cases} -1 & if\left(w^T x^{(i)} + b\right)t^{(i)} < 1 \\ 0 & otherwise \end{cases}$

$$\ell(w,b) = \frac{1}{2}\|w\|_2^2 \ + C\sum_{i=1}^{N} max\left(0, 1 - \left(w^T x^{(i)} + b\right)t^{(i)}\right)$$

$$\frac{\partial \ell(w,b)}{\partial w} = w - C\sum_{i=1}^{N} \alpha^{(i)} t^{(i)} x^{(i)}$$

where $\alpha^{(i)} = \begin{cases} -1 & if\left(w^T x^{(i)} + b\right)t^{(i)} < 1 \\ 0 & otherwise \end{cases}$

Let $\frac{\partial \ell(w,b)}{\partial w} = 0$

$$w^* = C\sum_{i=1}^{N} \alpha^{(i)} t^{(i)} x^{(i)}$$

Only a small subset of $\alpha^{(i)}$'s will be nonzero, and the corresponding $x^{(i)}$'s are the **Support Vectors**.

$$\ell(w,b) = \frac{1}{2}\|w\|_2^2 \;+ C\sum_{i=1}^{N} max\left(0, 1-\left(w^T x^{(i)} + b\right)t^{(i)}\right)$$

$$\frac{\partial \ell(w,b)}{\partial w} = w - C\sum_{i=1}^{N}\alpha^{(i)}t^{(i)}x^{(i)}$$

where $\alpha^{(i)} = \begin{cases} -1 & if\left(w^T x^{(i)} + b\right)t^{(i)} < 1 \\ 0 & otherwise \end{cases}$

Let $\frac{\partial \ell(w,b)}{\partial w} = 0$

$$w^* = C\sum_{i=1}^{N}\alpha^{(i)}t^{(i)}x^{(i)}$$

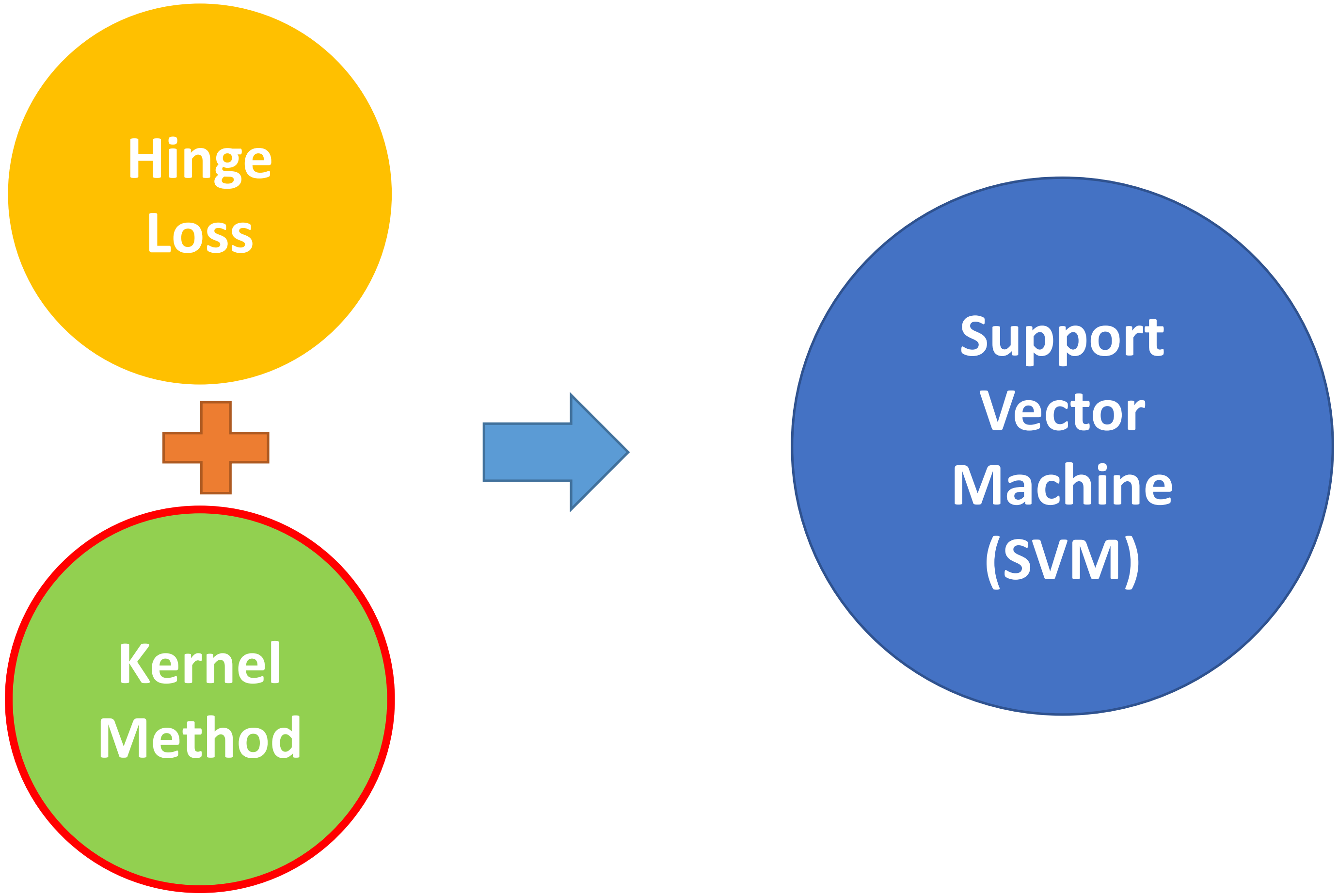
Only a small subset of $\alpha^{(i)}$'s will be nonzero, and the corresponding $x^{(i)}$'s are the **Support Vectors**.

Prediction on a new example:

$$\begin{aligned} y^{(new)} &= sgn\left(w^T x^{(new)} + b\right) \\ &= sgn\left(\sum_{i=1}^{N}\alpha^{(i)}t^{(i)}\left({x^{(i)}}^T x^{(new)}\right) + b\right) \end{aligned}$$

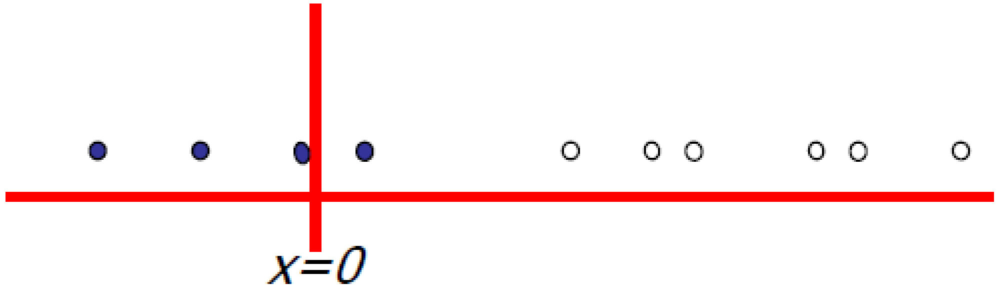
SVM

- Linear SVM
- Soft Margin
  - Hinge Loss
- Kernel Trick



SVM

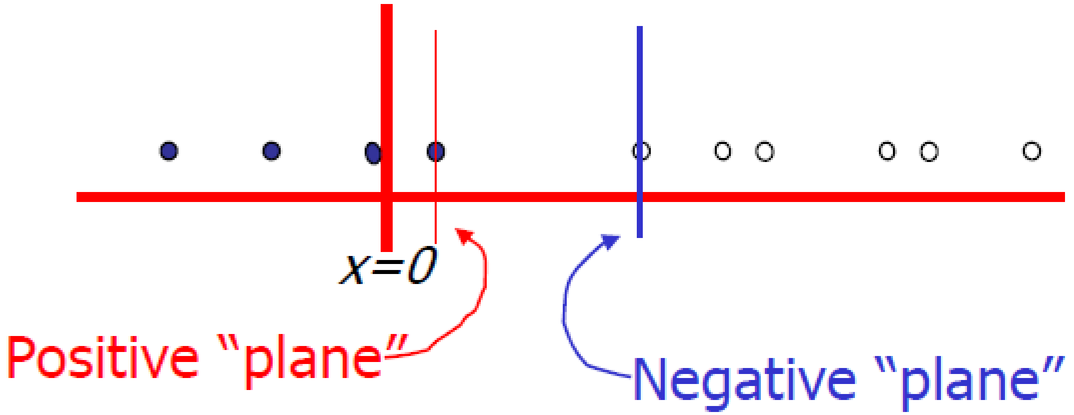
- Suppose we’re in 1-dimension



What would SVMs do with this data?

SVM

- Suppose we're in 1-dimension



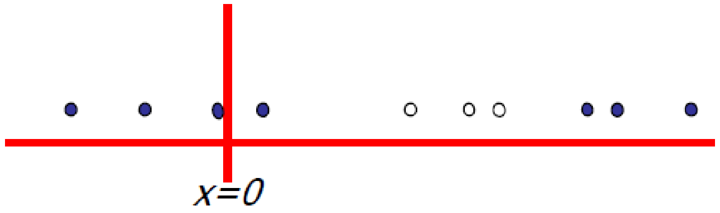
Not a big surprise

SVM

- Suppose we’re in 1-dimension

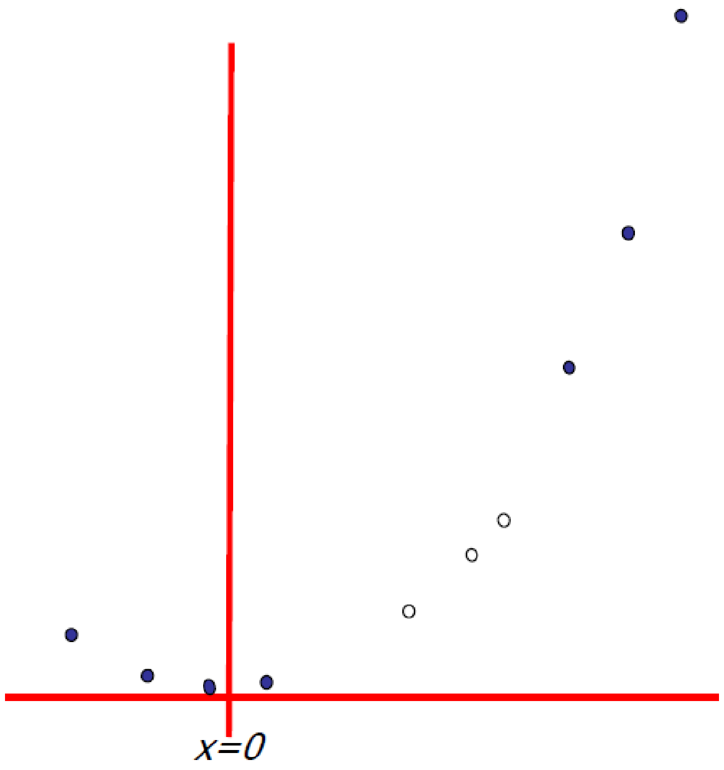
Apply the following map?

$$x_k \quad \Rightarrow \quad (x_k\,, x_k^2)$$



SVM

- Suppose we're in 1-dimension

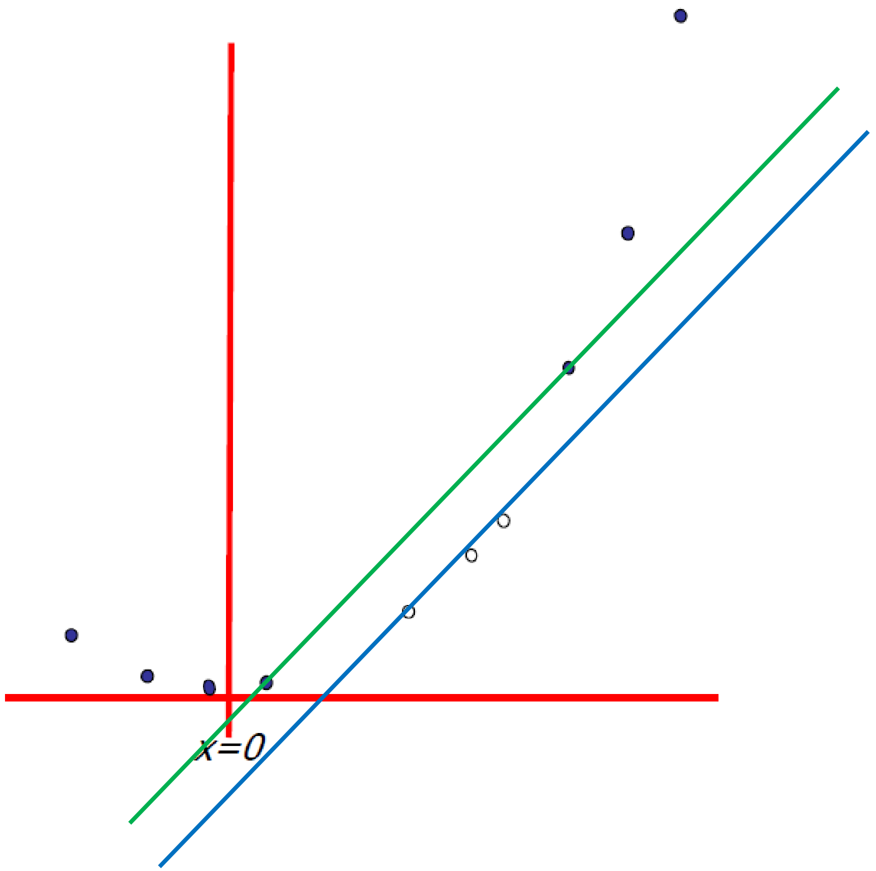


Apply the following map?

$$x_k \quad \Rightarrow \quad (x_k\,, x_k^2)$$

SVM

- Suppose we're in 1-dimension

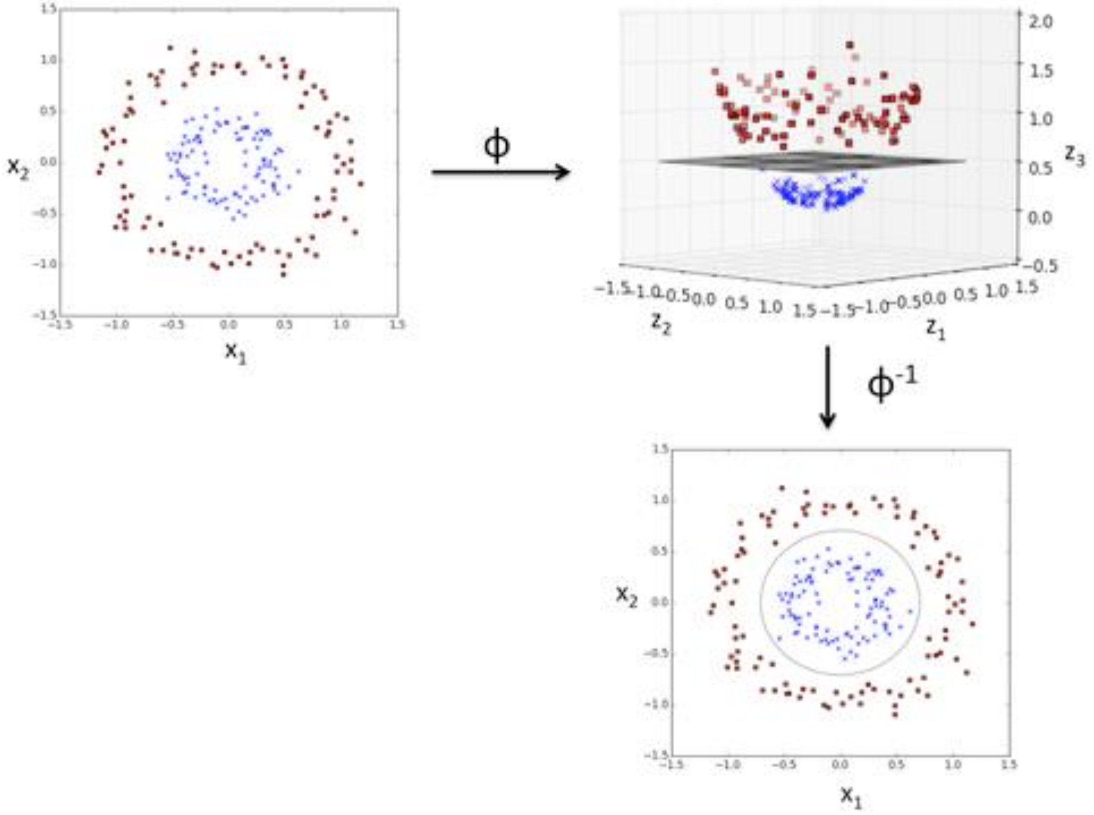


Apply the following map?

$$x_k \quad \Rightarrow \quad (x_k\,, x_k^2)$$

SVM

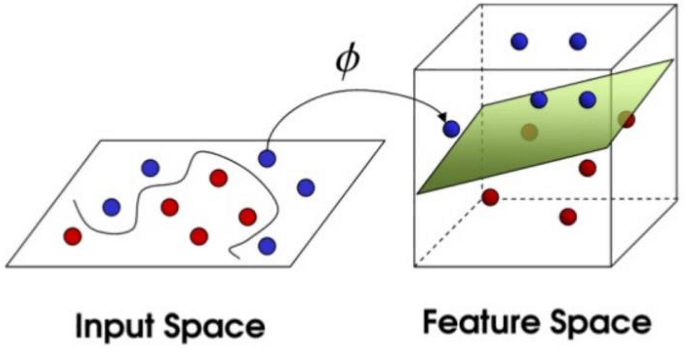
- Suppose we're in 2-dimension



https://sebastianraschka.com/faq/docs/select_svm_kernels.html

Feature Transformation

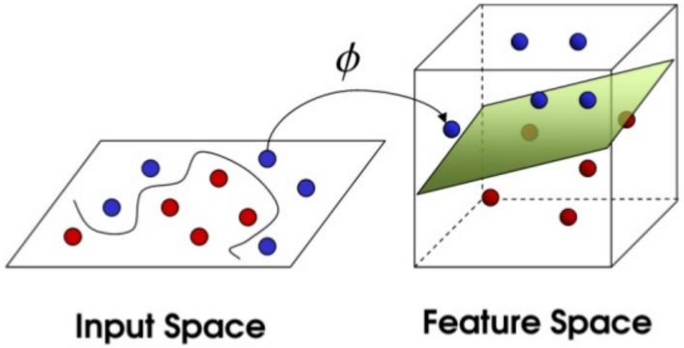
- To form non-linear decision boundaries in input space
  - Map data into feature space $x \to \phi(x)$
  - Replace dot products between inputs with feature points
$$x^{(i)^T} x^{(j)} \to \phi\left(x^{(i)}\right)^T \phi\left(x^{(j)}\right)$$
  - Find a linear decision boundary in feature space



Feature Transformation

- To form non-linear decision boundaries in input space
  - Map data into feature space $x \to \phi(x)$
  - Replace dot products between inputs with feature points
$${x^{(i)}}^T x^{(j)} \to \phi\left(x^{(i)}\right)^T \phi\left(x^{(j)}\right)$$
  - Find a linear decision boundary in feature space

It is not easy to find a good transformation



SVM

Note that both the learning objective and the decision function depend only on ***dot products*** between datapoints

- Training a SVM model is to maximize

$$\max_{\alpha^{(i)}} L(\alpha) = \max_{\alpha^{(i)}} \sum_{i=1}^{N} \alpha^{(i)} - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} t^{(i)} t^{(j)} \alpha^{(i)} \alpha^{(j)} \left({x^{(i)}}^{T} x^{(j)} \right)$$

s.t. $\alpha^{(i)} \geq 0$ and $\sum_{i=1}^{N} \alpha^{(i)} t^{(i)} = 0$

$$\min_{w,b} \frac{1}{2} \|w\|_2^2$$

$$s.t. \quad \forall i, \left(w^T x^{(i)} + b\right) t^{(i)} \geq 1$$

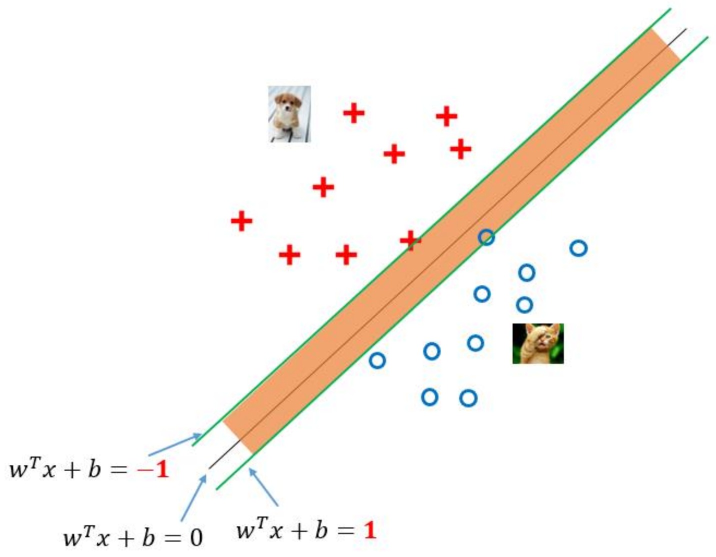
A constrained minimization

- Once $\alpha^{(i)}$ is obtained
  - The weights are

$$w = \sum_{i=1}^{N} \alpha^{(i)} t^{(i)} x^{(i)}$$

  - Prediction on a new example:

$$y^{(new)} = sgn\left(w^T x^{(new)} + b\right)$$
$$= sgn\left(\sum_{i=1}^{N} \alpha^{(i)} t^{(i)} \left({x^{(i)}}^{T} x^{(new)} \right) + b\right)$$



SVM

Note that both the learning objective and the decision function depend only on ***dot products*** between datapoints

- Training a SVM model is to maximize

$$\max_{\alpha^{(i)}} L(\alpha) = \max_{\alpha^{(i)}} \sum_{i=1}^{N} \alpha^{(i)} - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} t^{(i)}t^{(j)}\alpha^{(i)}\alpha^{(j)} \boldsymbol{K}\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}\right)$$

s.t. $\alpha^{(i)} \geq 0$ and $\sum_{i=1}^{N} \alpha^{(i)} t^{(i)} = 0$

$$\min_{w,b} \frac{1}{2}\|w\|_2^2$$

$$s.t.\quad \forall i, \left(w^T x^{(i)} + b\right) t^{(i)} \geq 1$$

A constrained minimization

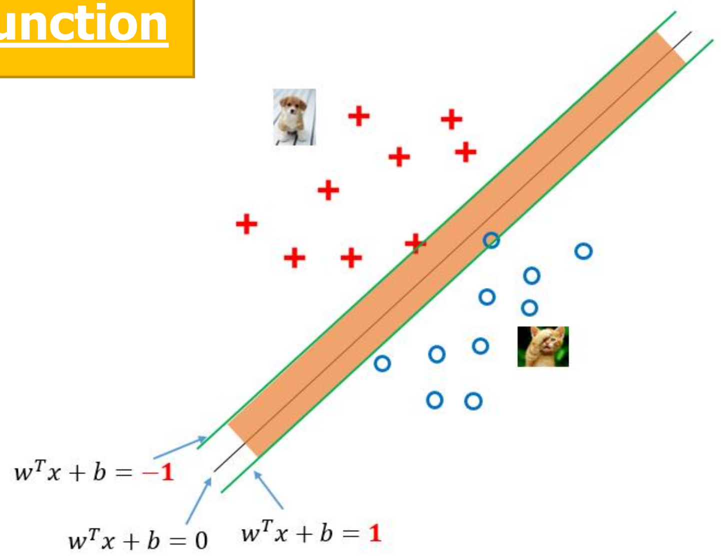
- Once $\alpha^{(i)}$ is obtained
  - The weights are

$$w = \sum_{i=1}^{N} \alpha^{(i)} t^{(i)} x^{(i)}$$

  - Prediction on a new example:

$$y^{(new)} = sgn\left(w^T x^{(new)} + b\right)$$
$$= sgn\left(\sum_{i=1}^{N} \alpha^{(i)} t^{(i)} \boldsymbol{K}\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(new)}\right) + b\right)$$

Replacing dot product with a kernel function



SVM

- Training datasets

$$\mathcal{D} = \{(x^{(1)}, t^{(1)}), \dots, (x^{(i)}, t^{(i)}), \dots, (x^{(N)}, t^{(N)})\}$$

(target $t^{(i)}$: 0 or 1)

- Linear model

$$y(x) = b + w_1 x_1 + w_2 x_2 + \cdots + w_M x_M$$

- Parameters

$$b, w_1, w_2, \dots, w_M$$

Note that both the learning objective and the decision function depend only on ***dot products*** between datapoints

Replacing dot product with a kernel function

SVM

- Training datasets

  $\mathcal{D} = \{(x^{(1)}, t^{(1)}), \ldots, (x^{(i)}, t^{(i)}), \ldots, (x^{(N)}, t^{(N)})\}$

  (target $t^{(i)}$: 0 or 1)

- Linear model

  $y(x) = b + w_1x_1 + w_2x_2 + \cdots + w_Mx_M$

- Parameters

  $b, w_1, w_2, \ldots, w_M$

$$w = \sum_{i=1}^{N} \alpha^{(i)} t^{(i)} x^{(i)}$$

$$\begin{aligned} y(x) &= w^T x + b \\ &= \textstyle\sum_{i=1}^{N} \alpha^{(i)} t^{(i)} K(x^{(i)}, x) + b \end{aligned}$$

Note that both the learning objective and the decision function depend only on ***dot products*** between datapoints

Replacing dot product with a <u>kernel function</u>

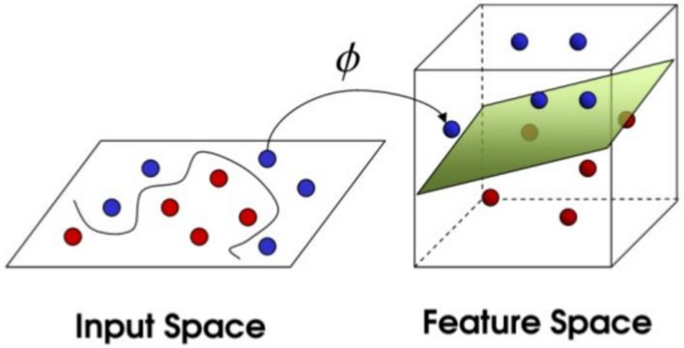
Kernel Trick

- To form non-linear decision boundaries in input space
  - Map data into feature space $x \to \phi(x)$
  - Replace dot products between inputs with feature points

$$x^{(i)^T} x^{(j)} \to \phi\left(x^{(i)}\right)^T \phi\left(x^{(j)}\right)$$

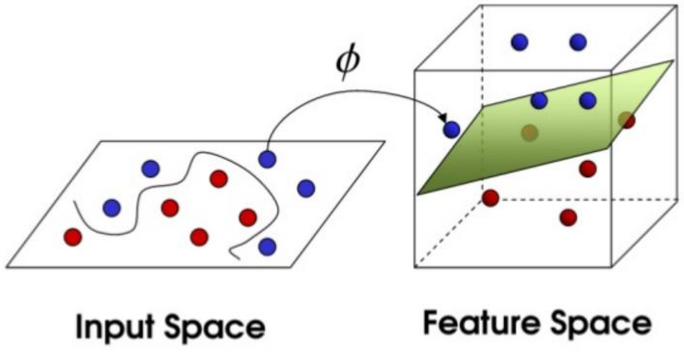
  - Find a linear decision boundary in feature space

- In SVM, we use Kernel Tricks
  - (Pro) Introduce nonlinearity into the model
  - (Pro) Computational cheap
  - (Con) Potential overfitting problem



Kernel Trick

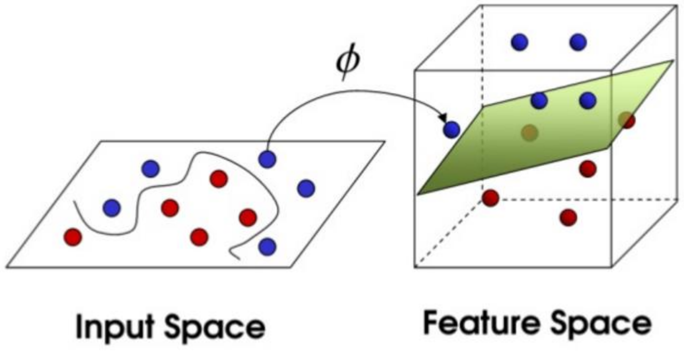
- Directly computing $K\left(x^{(i)}, x^{(j)}\right)$ can be faster than “feature transformation + inner product” sometimes.



Kernel Trick

- Directly computing $K\left(x^{(i)}, x^{(j)}\right)$ can be faster than "feature transformation + inner product" sometimes.

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \Rightarrow \phi(x) = \begin{bmatrix} {x_1}^2 \\ \sqrt{2}x_1x_2 \\ {x_2}^2 \end{bmatrix}$$

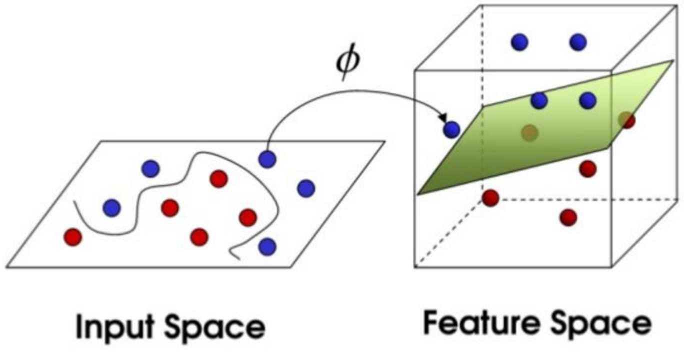


Kernel Trick

- Directly computing $K\left(x^{(i)}, x^{(j)}\right)$ can be faster than "feature transformation + inner product" sometimes.

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \Rightarrow \phi(x) = \begin{bmatrix} {x_1}^2 \\ \sqrt{2} x_1 x_2 \\ {x_2}^2 \end{bmatrix}$$

$$K(x, z) = (x \cdot z)^2$$

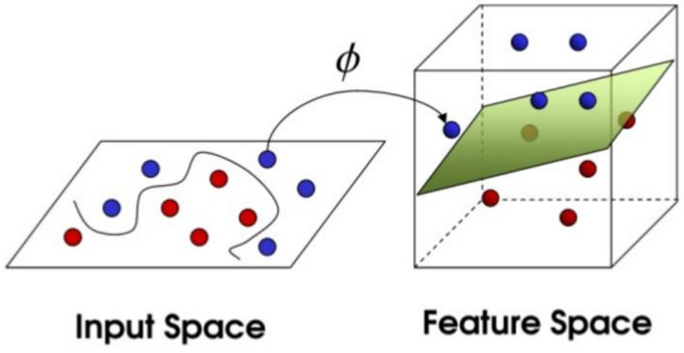


Kernel Trick

- Directly computing $K\left(x^{(i)}, x^{(j)}\right)$ can be faster than "feature transformation + inner product" sometimes.

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \Rightarrow \phi(x) = \begin{bmatrix} {x_1}^2 \\ \sqrt{2}x_1x_2 \\ {x_2}^2 \end{bmatrix}$$

$$K(x, z) = (x \cdot z)^2$$



$$K(x, z) = \phi(x) \cdot \phi(z) = \begin{bmatrix} {x_1}^2 \\ \sqrt{2}x_1x_2 \\ {x_2}^2 \end{bmatrix} \cdot \begin{bmatrix} {z_1}^2 \\ \sqrt{2}z_1z_2 \\ {z_2}^2 \end{bmatrix}$$

$$= {x_1}^2{z_1}^2 + 2x_1x_2z_1z_2 + {x_2}^2{z_2}^2$$

$$= (x_1z_1 + x_2z_2)^2 = \left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \cdot \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}\right)^2$$

$$= (x \cdot z)^2$$

Kernel Trick

- Directly computing $K\left(x^{(i)}, x^{(j)}\right)$ can be faster than "feature transformation + inner product" sometimes.

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_k \end{bmatrix} \quad z = \begin{bmatrix} z_1 \\ \vdots \\ z_k \end{bmatrix}$$

$$\phi(x) = \begin{bmatrix} {x_1}^2 \\ \vdots \\ {x_k}^2 \\ \sqrt{2}x_1x_2 \\ \sqrt{2}x_1x_3 \\ \vdots \\ \sqrt{2}x_2x_3 \\ \vdots \end{bmatrix}$$

Kernel Trick

- Directly computing $K\left(x^{(i)}, x^{(j)}\right)$ can be faster than "feature transformation + inner product" sometimes.

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_k \end{bmatrix} \quad z = \begin{bmatrix} z_1 \\ \vdots \\ z_k \end{bmatrix}$$

$$K(x, z) = (x \cdot z)^2$$

$$= (x_1 z_1 + x_2 z_2 + \cdots + x_k z_k)^2$$

$$= \underline{x_1{}^2}\,\underline{z_1{}^2} + \underline{x_2{}^2}\,\underline{z_2{}^2} + \cdots + \underline{x_k{}^2}\,\underline{z_k{}^2}$$

$$+2\underline{x_1 x_2}\,\underline{z_1 z_2} + 2\underline{x_1 x_3}\,\underline{z_1 z_3} + \cdots$$

$$+2\underline{x_2 x_3}\,\underline{z_2 z_3} + 2\underline{x_2 x_4}\,\underline{z_2 z_4} + \cdots$$

$$= \phi(x) \cdot \phi(z)$$

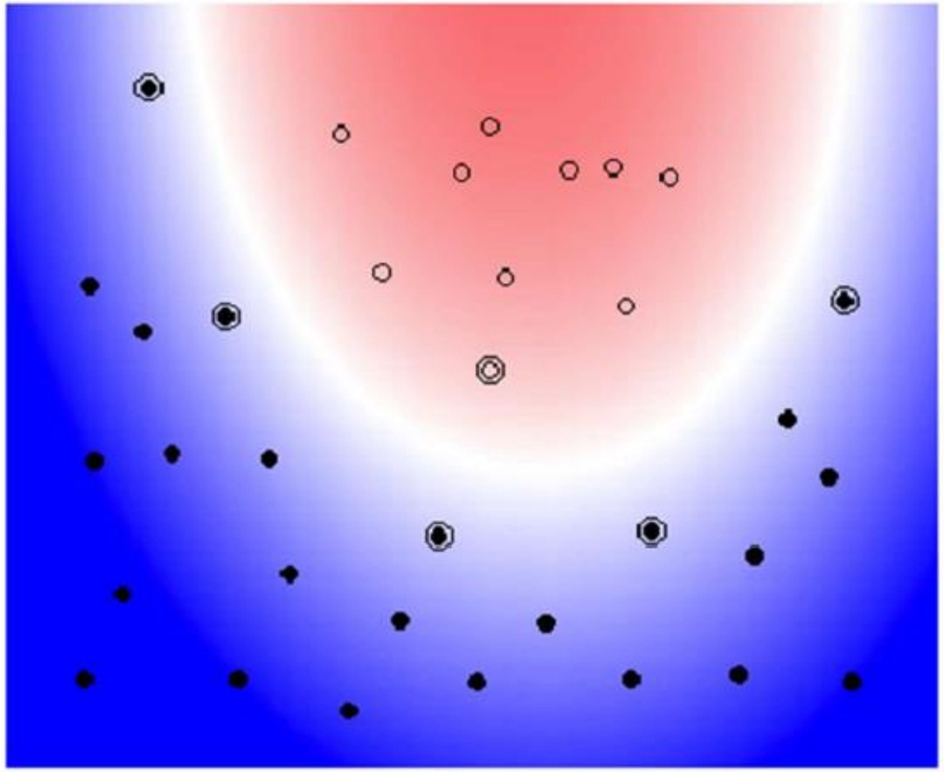
$$\phi(x) = \begin{bmatrix} x_1{}^2 \\ \vdots \\ x_k{}^2 \\ \sqrt{2} x_1 x_2 \\ \sqrt{2} x_1 x_3 \\ \vdots \\ \sqrt{2} x_2 x_3 \\ \vdots \end{bmatrix}$$

Kernel Trick

- SVM with polynomial of degree 2

Kernel: $K\left(x^{(i)}, x^{(j)}\right) = \left(x^{(i)^T} x^{(j)} + 1\right)^2$

plot by Bell SVM applet



Kernel Trick

- Polynomial Kernel

$$K(x, z) = (x^T z + 1)^d$$

- Gaussian Kernel

$$K(x, z) = e^{\left(-\frac{\|x-z\|^2}{2\sigma^2}\right)}$$

- Sigmoid Kernel

$$K(x, z) = tanh(\beta x^T z + a)$$

- …

Kernel Trick

- Radial Basis Function Kernel

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \end{bmatrix} \quad z = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \end{bmatrix}$$

$$K(x,z) = exp\left(-\frac{1}{2}\|x-z\|_2\right)$$

Kernel Trick

- Radial Basis Function Kernel

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \end{bmatrix} \quad z = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \end{bmatrix}$$

$$K(x, z) = exp\left(-\frac{1}{2}\|x - z\|_2\right) = \phi(x) \cdot \phi(z)?$$

Kernel Trick

- Radial Basis Function Kernel

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \end{bmatrix} \quad z = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \end{bmatrix}$$

$$K(x,z) = exp\left(-\frac{1}{2}\|x-z\|_2\right) = \phi(x)\cdot\phi(z)?$$

$$= exp\left(-\frac{1}{2}\|x\|_2 - \frac{1}{2}\|z\|_2 + x\cdot z\right)$$

Kernel Trick

- Radial Basis Function Kernel

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \end{bmatrix} \quad z = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \end{bmatrix}$$

$$K(x, z) = exp\left(-\frac{1}{2}\|x - z\|_2\right) = \phi(x) \cdot \phi(z)?$$

$$= exp\left(-\frac{1}{2}\|x\|_2 - \frac{1}{2}\|z\|_2 + x \cdot z\right)$$

$$= exp\left(-\frac{1}{2}\|x\|_2\right) exp\left(-\frac{1}{2}\|z\|_2\right) exp(x \cdot z) \quad = C_x C_z exp(x \cdot z)$$

Kernel Trick

- Radial Basis Function Kernel

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \end{bmatrix} \quad z = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \end{bmatrix}$$

$$K(x,z) = exp\left(-\frac{1}{2}\|x - z\|_2\right) = \phi(x) \cdot \phi(z)?$$

$$= exp\left(-\frac{1}{2}\|x\|_2 - \frac{1}{2}\|z\|_2 + x \cdot z\right)$$

$$= exp\left(-\frac{1}{2}\|x\|_2\right) exp\left(-\frac{1}{2}\|z\|_2\right) exp(x \cdot z) \quad = C_x C_z exp(x \cdot z)$$

$$= C_x C_z \sum_{i=0}^{\infty} \frac{(x \cdot z)^i}{i!} = C_x C_z + C_x C_z (x \cdot z) + C_x C_z \frac{1}{2} (x \cdot z)^2 \cdots$$

Kernel Trick

- Radial Basis Function Kernel

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \end{bmatrix} \quad z = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \end{bmatrix}$$

$$K(x,z) = exp\left(-\frac{1}{2}\|x-z\|_2\right) = \phi(x)\cdot\phi(z)?$$

$$= exp\left(-\frac{1}{2}\|x\|_2 - \frac{1}{2}\|z\|_2 + x\cdot z\right)$$

$$= exp\left(-\frac{1}{2}\|x\|_2\right) exp\left(-\frac{1}{2}\|z\|_2\right) exp(x\cdot z) \quad = C_x C_z exp(x\cdot z)$$

$$= C_x C_z \sum_{i=0}^{\infty} \frac{(x\cdot z)^i}{i!} = C_x C_z + C_x C_z (x\cdot z) + C_x C_z \frac{1}{2}(x\cdot z)^2 \cdots$$

$$[C_x]\cdot[C_z] \qquad \begin{bmatrix} C_x x_1 \\ C_x x_2 \\ \vdots \end{bmatrix}\cdot\begin{bmatrix} C_z z_1 \\ C_z z_2 \\ \vdots \end{bmatrix} \qquad \frac{1}{\sqrt{2}}\begin{bmatrix} C_x {x_1}^2 \\ \vdots \\ \sqrt{2}C_x x_1 x_2 \\ \vdots \end{bmatrix}\cdot\frac{1}{\sqrt{2}}\begin{bmatrix} C_z {z_1}^2 \\ \vdots \\ \sqrt{2}C_z z_1 z_2 \\ \vdots \end{bmatrix}$$

Kernel Trick

- Radial Basis Function Kernel

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \end{bmatrix} \quad z = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \end{bmatrix}$$

$$K(x, z) = exp\left(-\frac{1}{2}\|x - z\|_2\right) = \phi(x) \cdot \phi(z)?$$

$\phi(*)$ has inf dim!!!

$$= exp\left(-\frac{1}{2}\|x\|_2 - \frac{1}{2}\|z\|_2 + x \cdot z\right)$$

$$= exp\left(-\frac{1}{2}\|x\|_2\right) exp\left(-\frac{1}{2}\|z\|_2\right) exp(x \cdot z) \quad = C_x C_z exp(x \cdot z)$$

$$= C_x C_z \sum_{i=0}^{\infty} \frac{(x \cdot z)^i}{i!} = C_x C_z + C_x C_z (x \cdot z) + C_x C_z \frac{1}{2}(x \cdot z)^2 \cdots$$

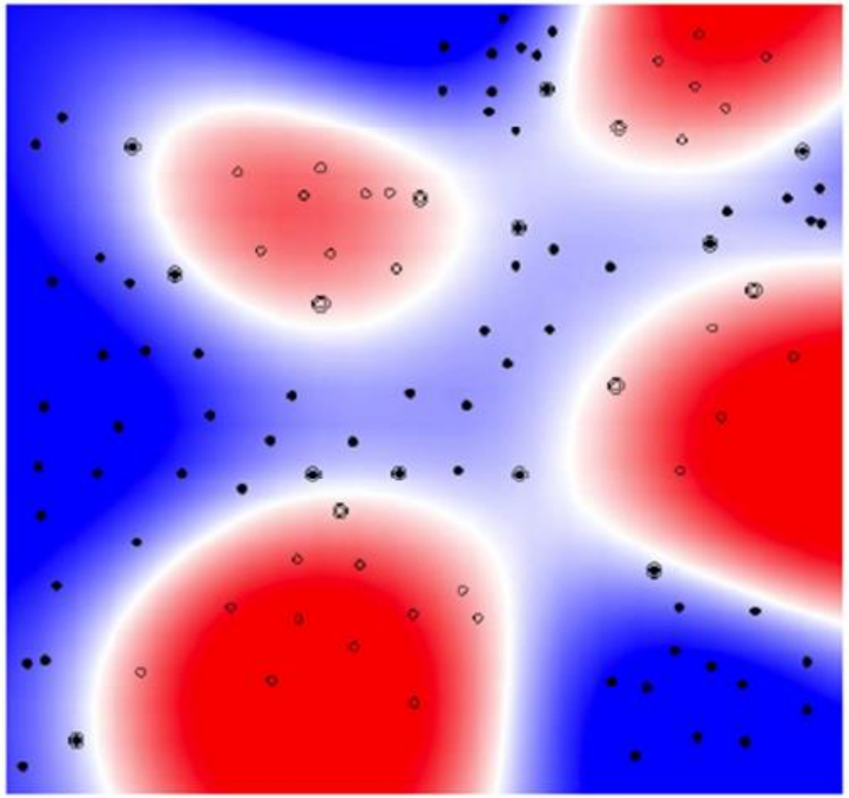
$$[C_x] \cdot [C_z] \qquad \begin{bmatrix} C_x x_1 \\ C_x x_2 \\ \vdots \end{bmatrix} \cdot \begin{bmatrix} C_z z_1 \\ C_z z_2 \\ \vdots \end{bmatrix} \qquad \frac{1}{\sqrt{2}} \begin{bmatrix} C_x {x_1}^2 \\ \vdots \\ \sqrt{2} C_x x_1 x_2 \\ \vdots \end{bmatrix} \cdot \frac{1}{\sqrt{2}} \begin{bmatrix} C_z {z_1}^2 \\ \vdots \\ \sqrt{2} C_z z_1 z_2 \\ \vdots \end{bmatrix}$$

Kernel Trick

- SVM with Radial Basis Function Kernel (RBF)

Kernel: $K\left(x^{(i)}, x^{(j)}\right) = e^{\left(-\frac{\left\|x^{(i)}-x^{(j)}\right\|^2}{\sigma^2}\right)}$

plot by Bell SVM applet



Multi-Class Classification

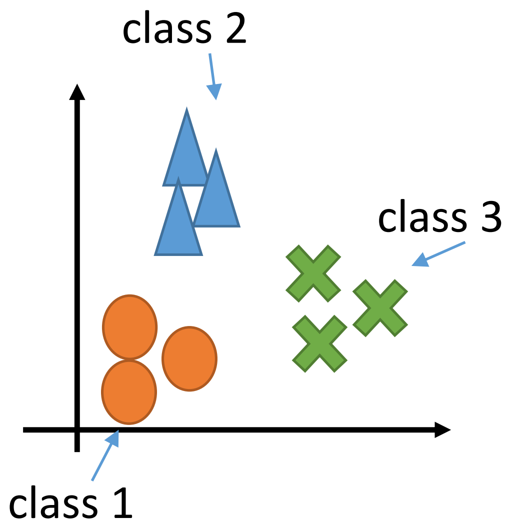
- SVMs are inherently two-class classifiers
- What can be done?

Multi-Class Classification

- SVMs are inherently two-class classifiers
- What can be done?
  - One vs All
  - Other approaches
    - Pair-wise SVM
      - Brunner, Carl, Andreas Fischer, Klaus Luig, and Thorsten Thies. "Pairwise support vector machines and their application to large scale problems." *Journal of Machine Learning Research* 13, no. Aug (2012): 2279-2292.
    - Multi-category SVM
      - Duan, Kai-Bo, and S. Sathiya Keerthi. "Which is the best multiclass SVM method? An empirical study." In *International Workshop on Multiple Classifier Systems*, pp. 278-285. Springer Berlin Heidelberg, 2005.

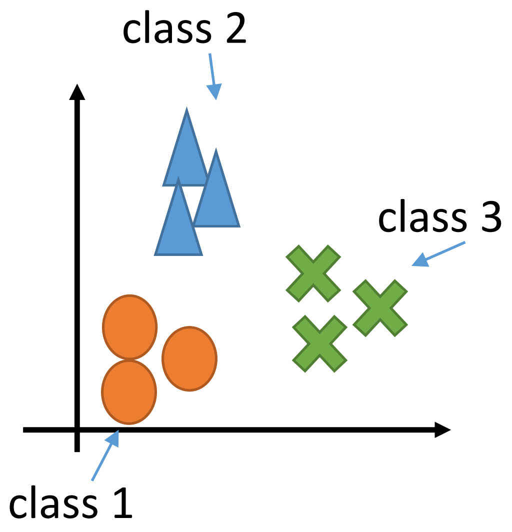
Multi-Class Classification

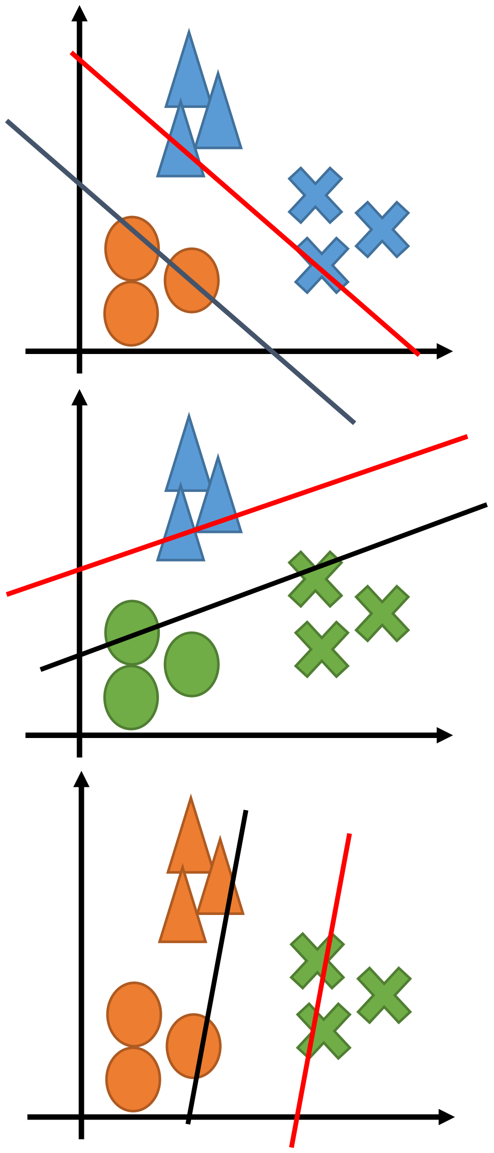
- One vs All



Multi-Class Classification

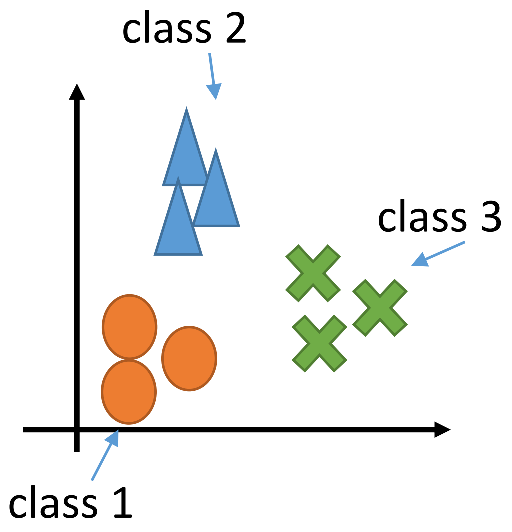
- One vs All

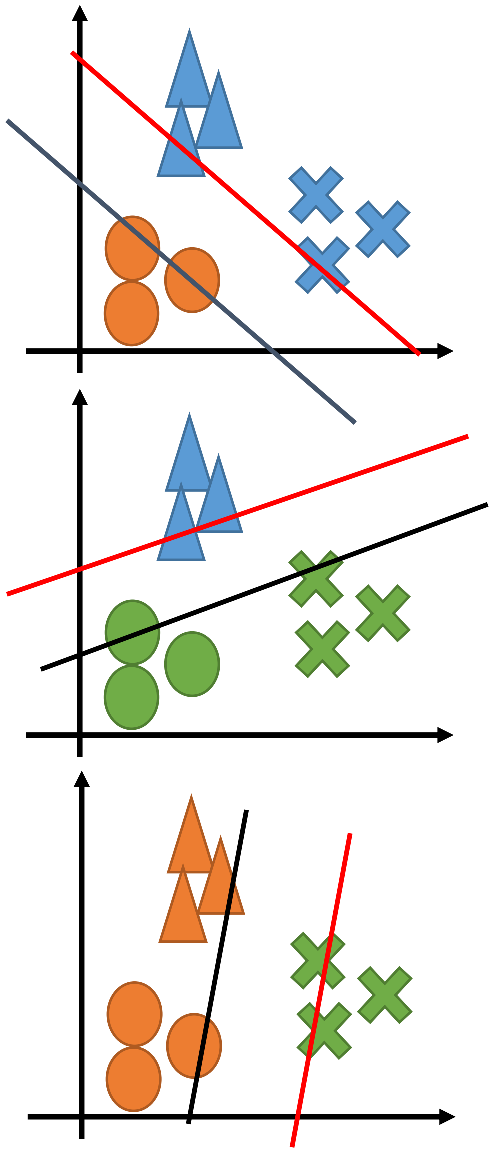


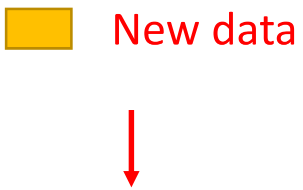


Multi-Class Classification

- One vs All



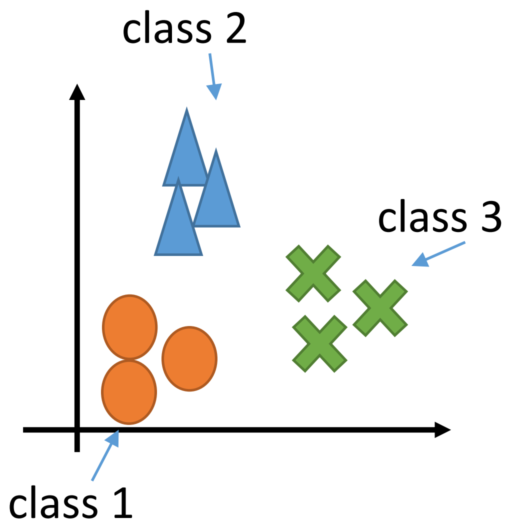


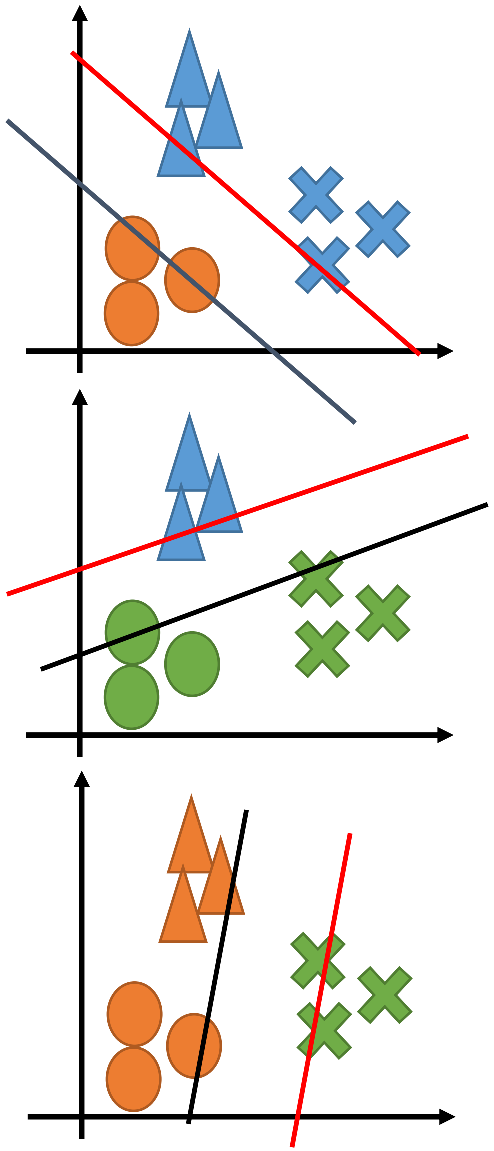


Multi-Class Classification

New data

- One vs All





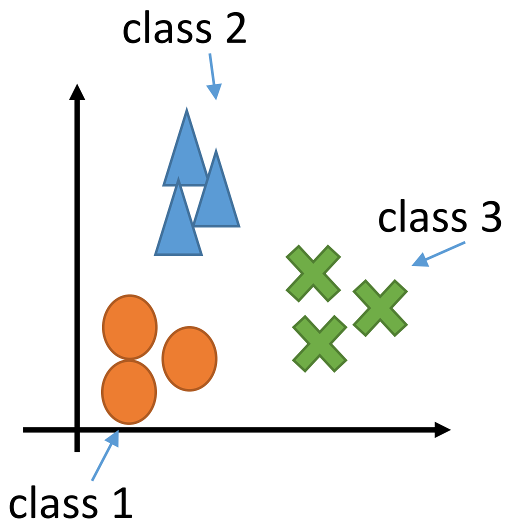
SVM 1: decision value in class 1

SVM 2: decision value in class 2

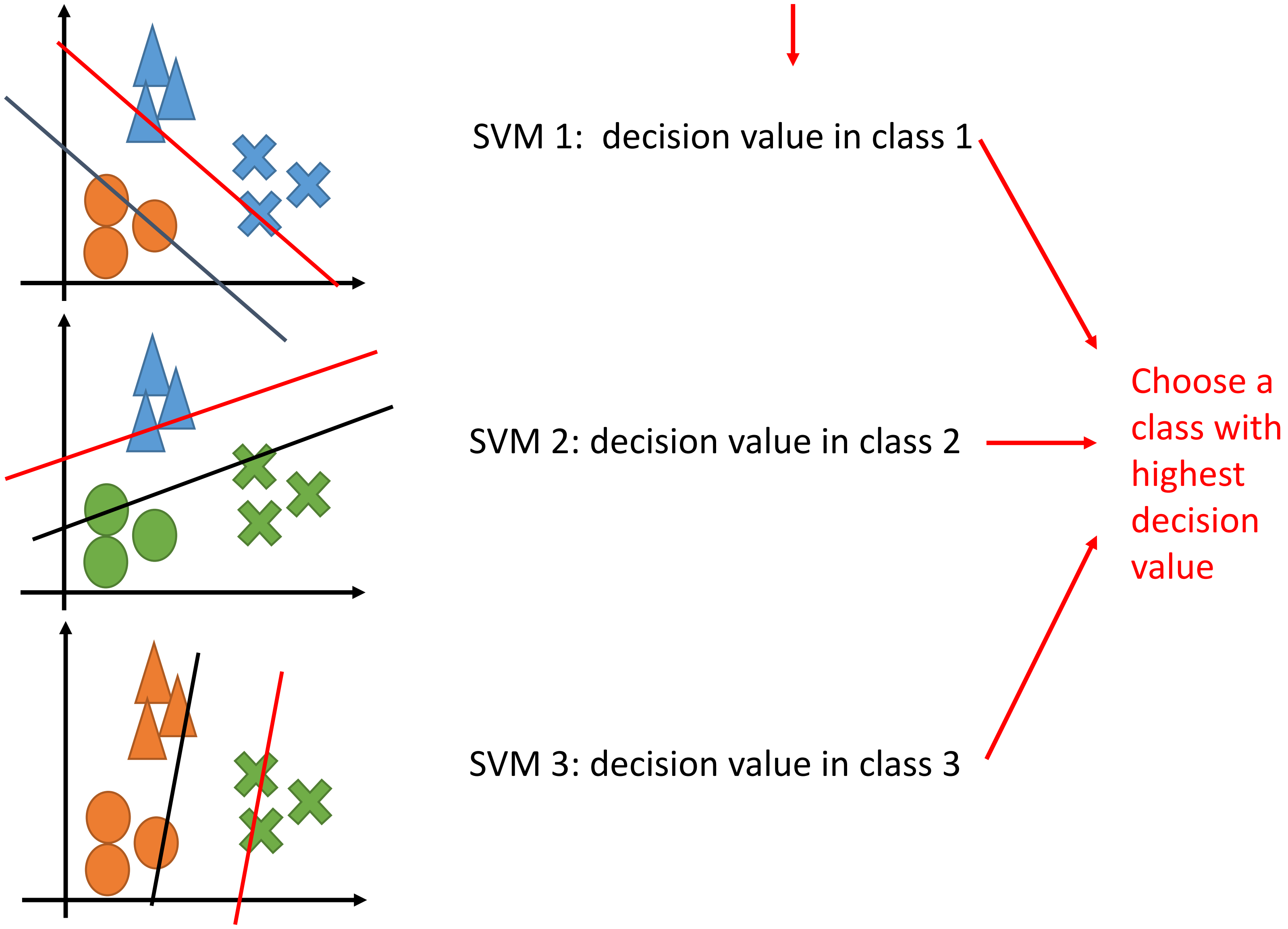
SVM 3: decision value in class 3

Multi-Class Classification

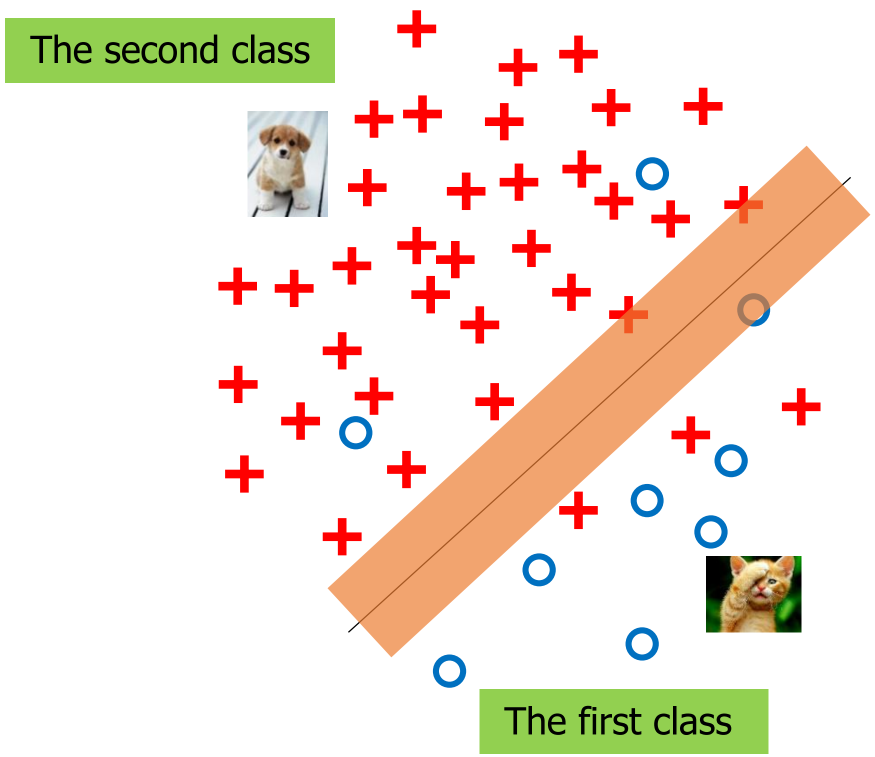
- One vs All



New data



SVM for Unbalanced Data

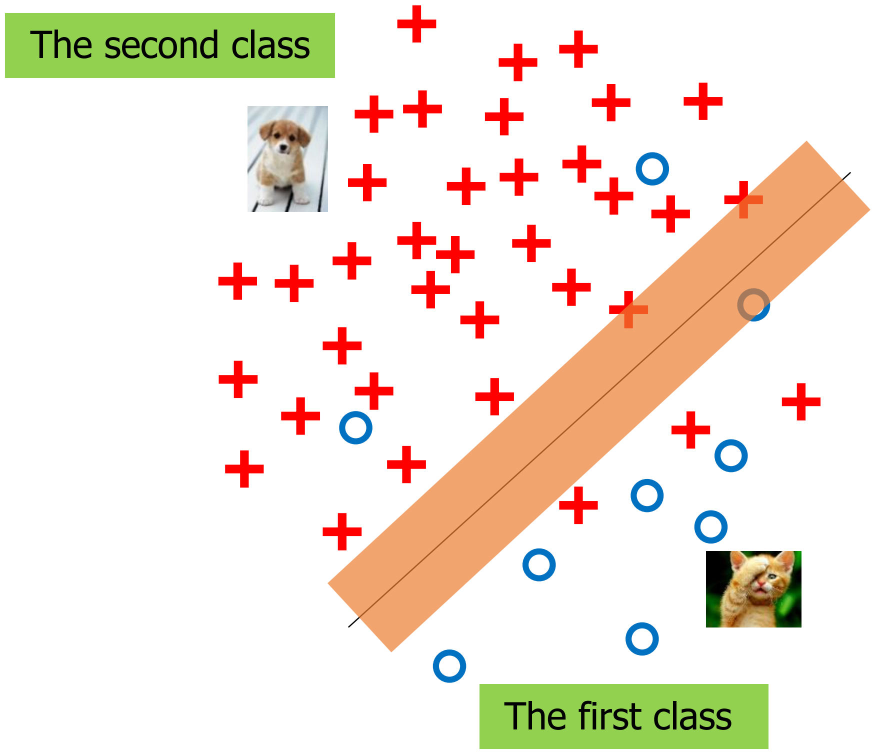


$$\min_{w,b} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N} \xi^{(i)}$$

$$s.t.\quad \forall i,\left(w^T x^{(i)} + b\right)t^{(i)} \geq 1 - \xi^{(i)}$$

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

SVM for Unbalanced Data



$$\min_{w,b} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N} \xi^{(i)}$$

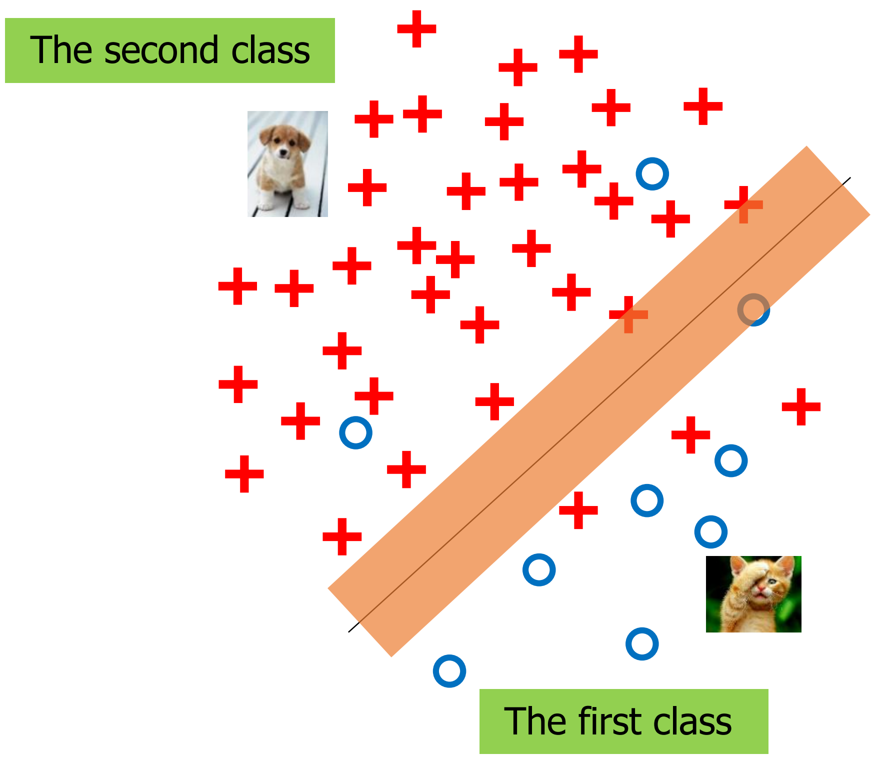
$$s.t.\ \ \forall i,\ (w^T x^{(i)} + b)t^{(i)} \geq 1 - \xi^{(i)}$$

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

If the first class has much smaller size than the second class,

- apply different weights to the two classes: $C_1 > C_2$

SVM for Unbalanced Data



$$\min_{w,b} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N} \xi^{(i)}$$

$$s.t.\quad \forall i, \left(w^T x^{(i)} + b\right) t^{(i)} \geq 1 - \xi^{(i)}$$

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

If the first class has much smaller size than the second class,

- apply different weights to the two classes: $C_1 > C_2$

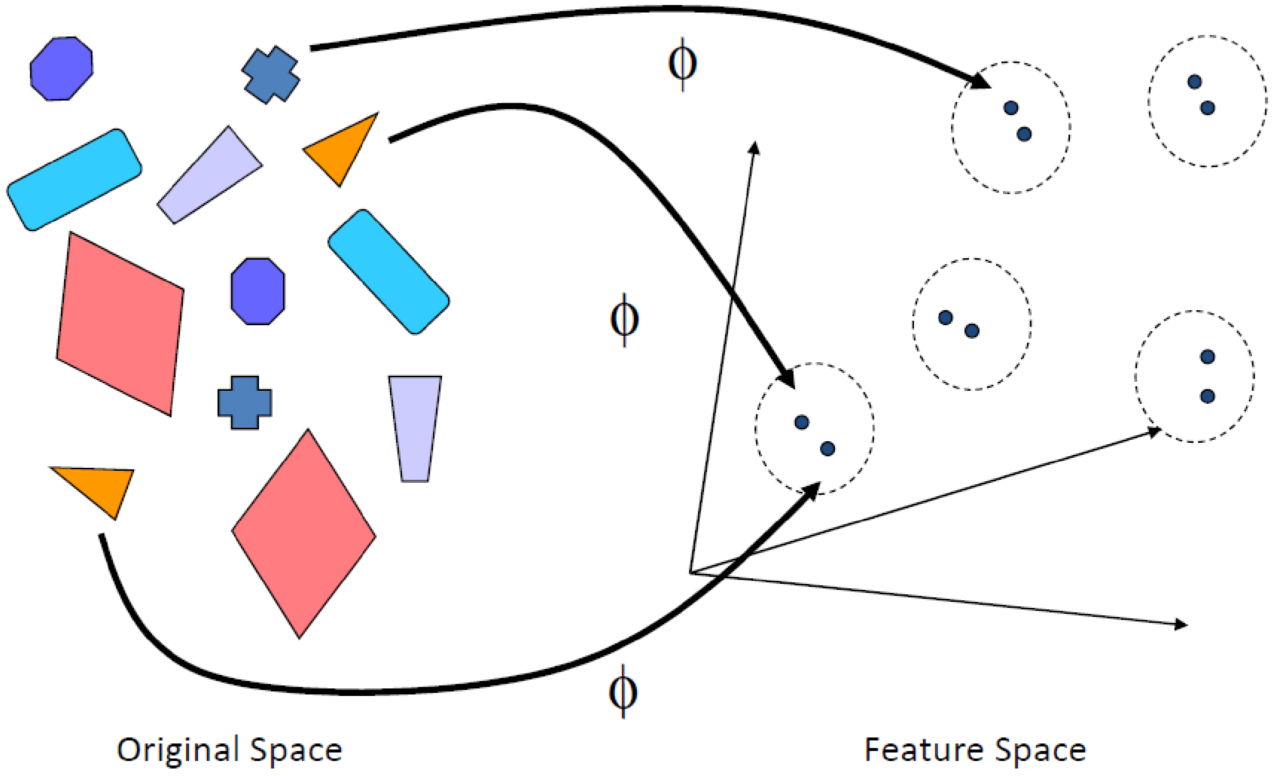
$$\min_{w,b} \frac{1}{2}\|w\|_2^2 + C_1 \sum_{x^{(i)} \in Class\ 1} \xi^{(i)} + C_2 \sum_{x^{(i)} \in Class\ 2} \xi^{(i)}$$

$$s.t.\quad \forall i, \left(w^T x^{(i)} + b\right) t^{(i)} \geq 1 - \xi^{(i)}$$

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

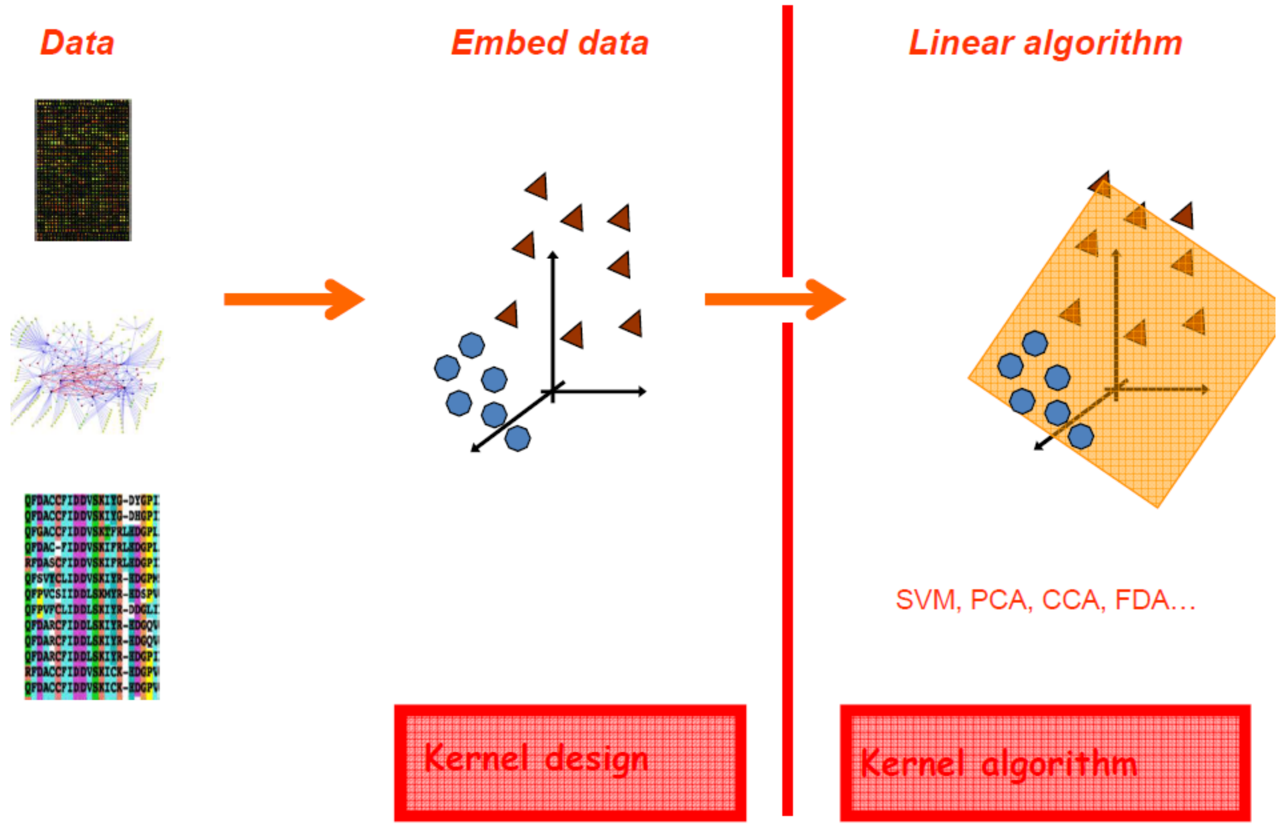
Kernel-based Learning

- Kernels are measures of similarity



Kernel-based Learning

- Kernels are measures of similarity



SVM

- Linear Regression
  - Linear function + Square loss

$$\ell(w) = \frac{1}{2N}\sum_{i=1}^{N}[t^{(i)} - y(x^{(i)})]^2$$

- Logistic Regression
  - Sigmoid function + Cross entropy loss

$$\ell(w) = -\frac{1}{N}\sum_{i=1}^{N}\left[t^{(i)}\log\left(y\left(x^{(i)}\right)\right) + \left(1 - t^{(i)}\right)\log\left(1 - y\left(x^{(i)}\right)\right)\right]$$

- SVM
  - Linear function + Hinge loss + L2 norm

$$\ell(w, b) = \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N} max\left(0, 1 - \left(w^T x^{(i)} + b\right)t^{(i)}\right)$$

SVM

We want to have a classifier with $f\left(x^{(i)}\right)t^{(i)} \geq 1$ for all data points

- Linear Regression
  - Linear function + Square loss

$$\ell(w) = \frac{1}{2N}\sum_{i=1}^{N}[t^{(i)} - y(x^{(i)})]^2$$

- Logistic Regression
  - Sigmoid function + Cross entropy loss

$$\ell(w) = -\frac{1}{N}\sum_{i=1}^{N}\left[t^{(i)}\log\left(y(x^{(i)})\right) + (1-t^{(i)})\log\left(1-y(x^{(i)})\right)\right]$$

- SVM
  - Linear function + Hinge loss + L2 norm

$$\ell(w,b) = \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N} max\left(0, 1-\left(w^T x^{(i)} + b\right)t^{(i)}\right)$$

SVM

We want to have a classifier with $f\left(x^{(i)}\right)t^{(i)} \geq 1$ for all data points

- Linear Regression
  - Linear function + Square loss

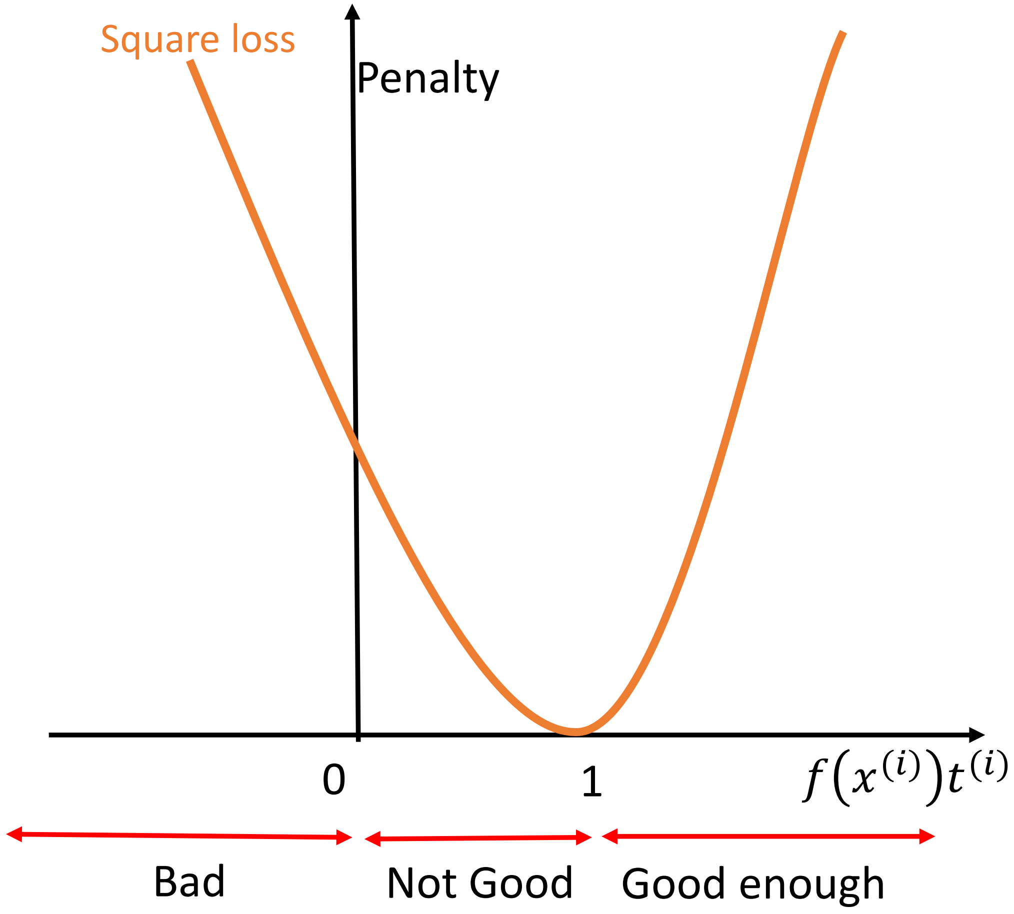
$$\ell(w) = \frac{1}{2N}\sum_{i=1}^{N}[t^{(i)} - y(x^{(i)})]^2$$

- Logistic Regression
  - Sigmoid function + Cross entropy loss

$$\ell(w) = -\frac{1}{N}\sum_{i=1}^{N}\left[t^{(i)}\log\left(y\left(x^{(i)}\right)\right) + \left(1 - t^{(i)}\right)\log\left(1 - y\left(x^{(i)}\right)\right)\right]$$

- SVM
  - Linear function + Hinge loss + L2 norm

$$\ell(w,b) = \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N} max\left(0, 1 - \left(w^T x^{(i)} + b\right)t^{(i)}\right)$$



SVM

We want to have a classifier with $f\left(x^{(i)}\right)t^{(i)} \geq 1$ for all data points

- Linear Regression
  - Linear function + Square loss

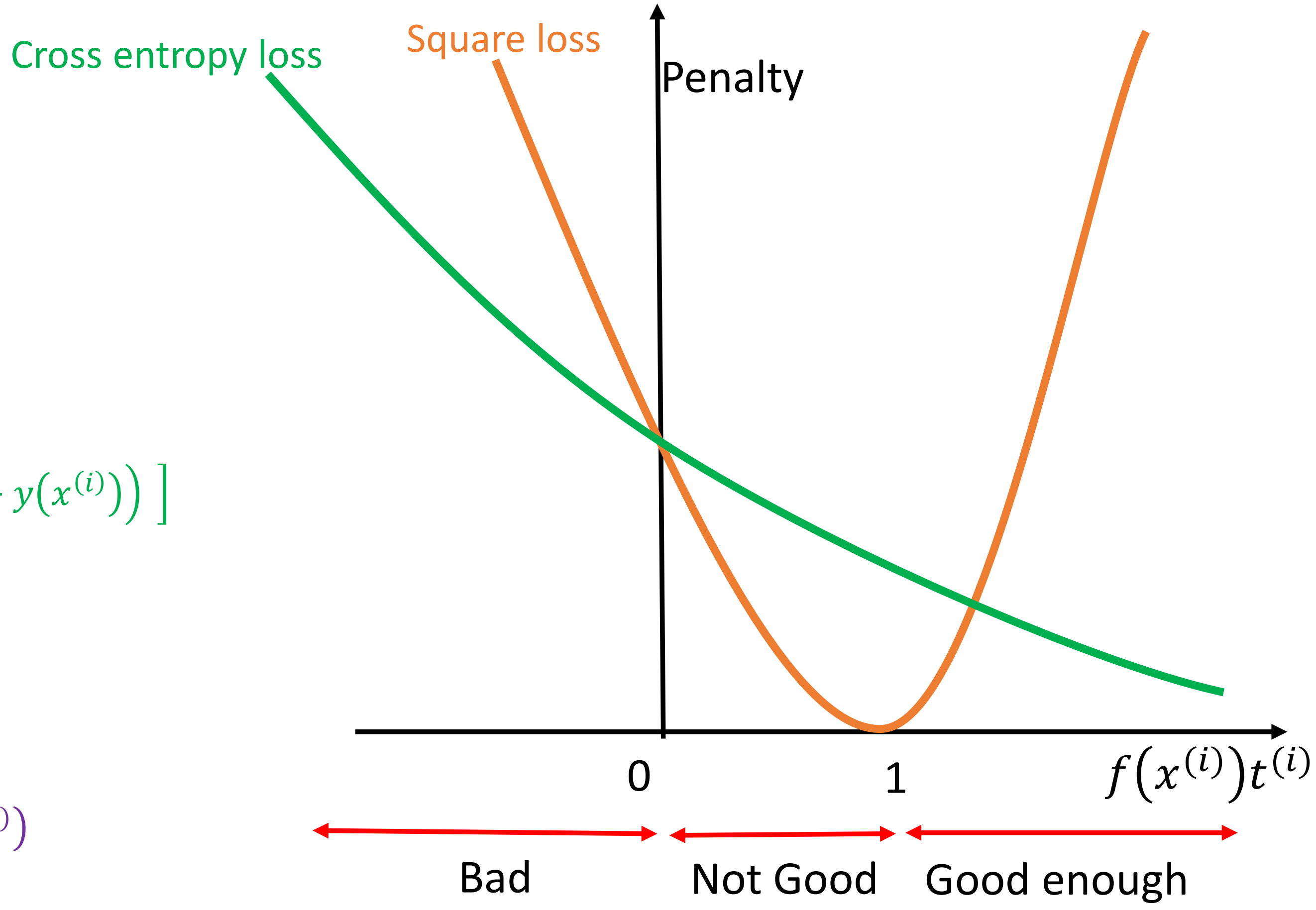
$$\ell(w) = \frac{1}{2N}\sum_{i=1}^{N}[t^{(i)} - y(x^{(i)})]^2$$

- Logistic Regression
  - Sigmoid function + Cross entropy loss

$$\ell(w) = -\frac{1}{N}\sum_{i=1}^{N}\left[t^{(i)}\log\left(y\left(x^{(i)}\right)\right) + \left(1 - t^{(i)}\right)\log\left(1 - y\left(x^{(i)}\right)\right)\right]$$

- SVM
  - Linear function + Hinge loss + L2 norm

$$\ell(w,b) = \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N} max\left(0, 1 - \left(w^T x^{(i)} + b\right)t^{(i)}\right)$$



SVM

We want to have a classifier with $f\left(x^{(i)}\right)t^{(i)} \geq 1$ for all data points

- Linear Regression
  - Linear function + Square loss

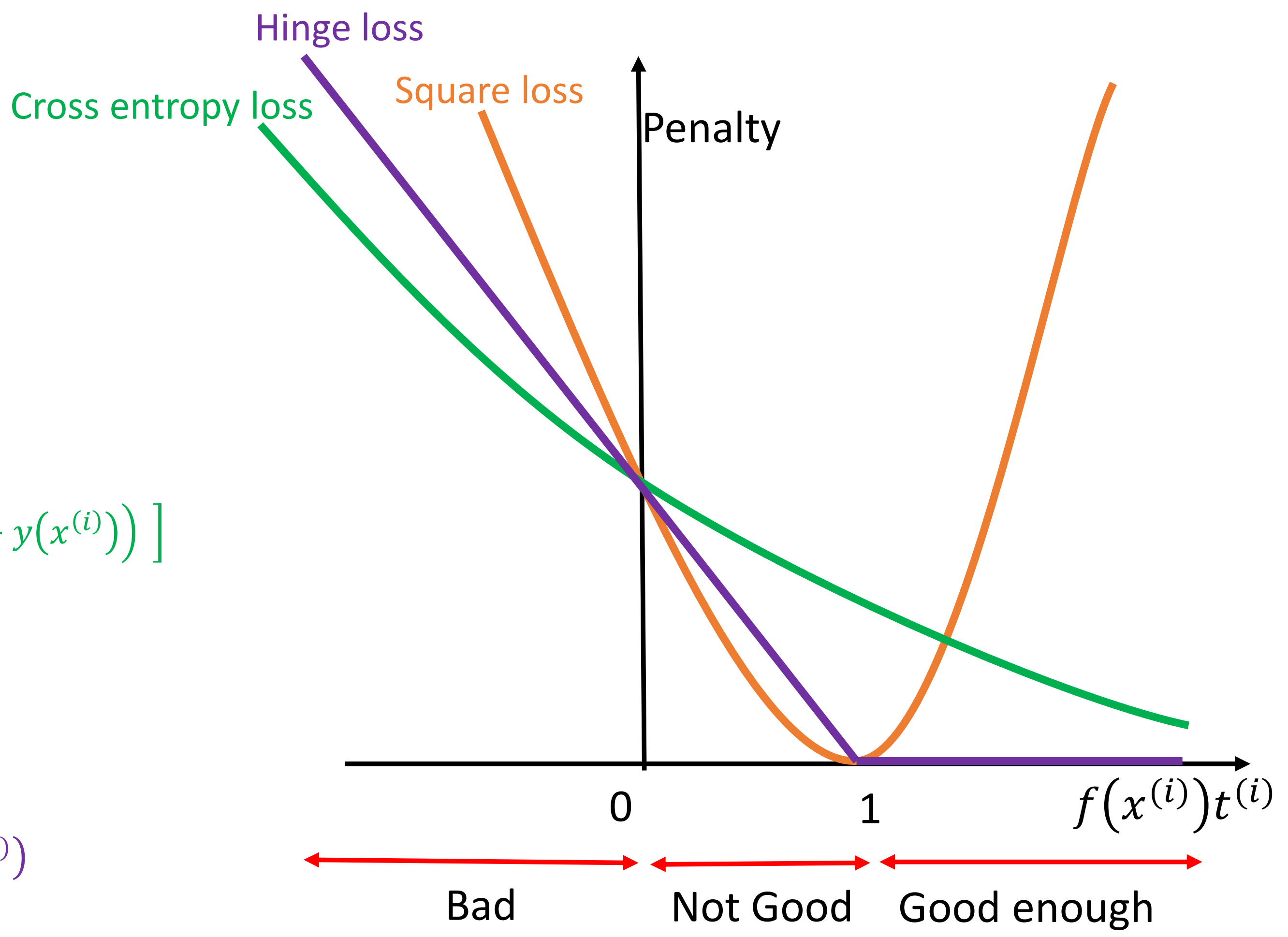
$$\ell(w) = \frac{1}{2N}\sum_{i=1}^{N}[t^{(i)} - y(x^{(i)})]^2$$

- Logistic Regression
  - Sigmoid function + Cross entropy loss

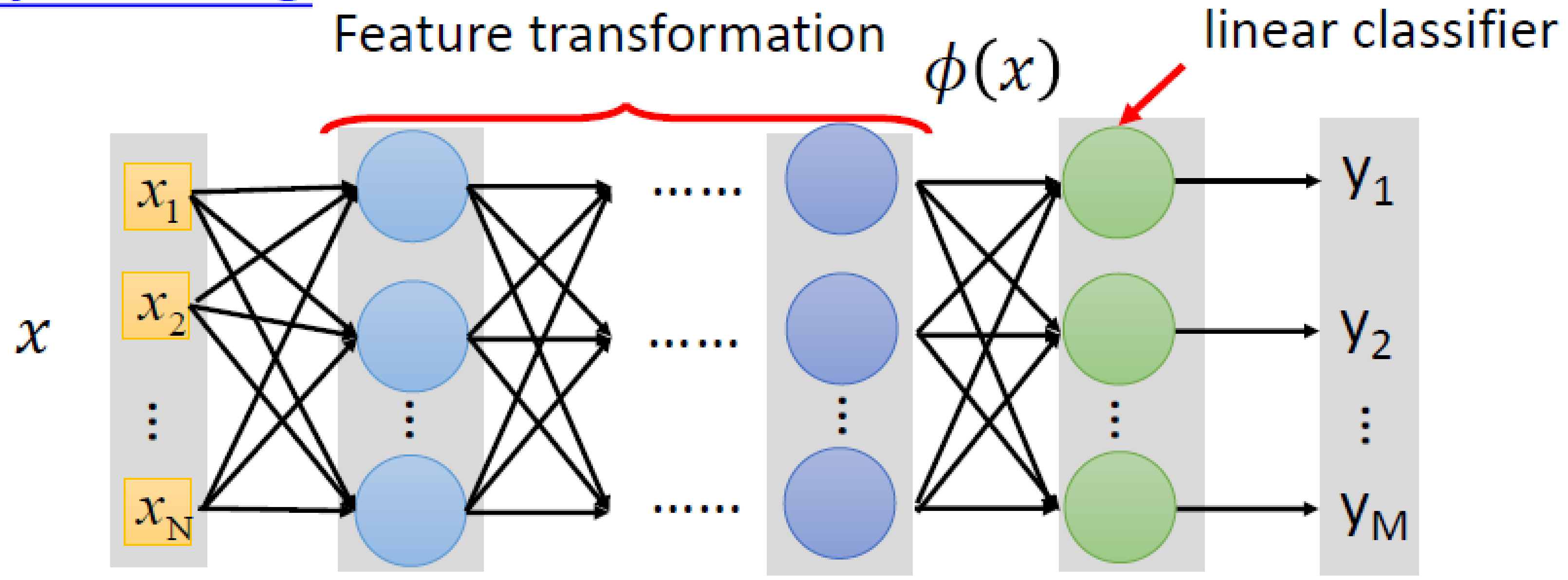
$$\ell(w) = -\frac{1}{N}\sum_{i=1}^{N}\left[t^{(i)}\log\left(y\left(x^{(i)}\right)\right) + \left(1 - t^{(i)}\right)\log\left(1 - y\left(x^{(i)}\right)\right)\right]$$

- SVM
  - Linear function + Hinge loss + L2 norm

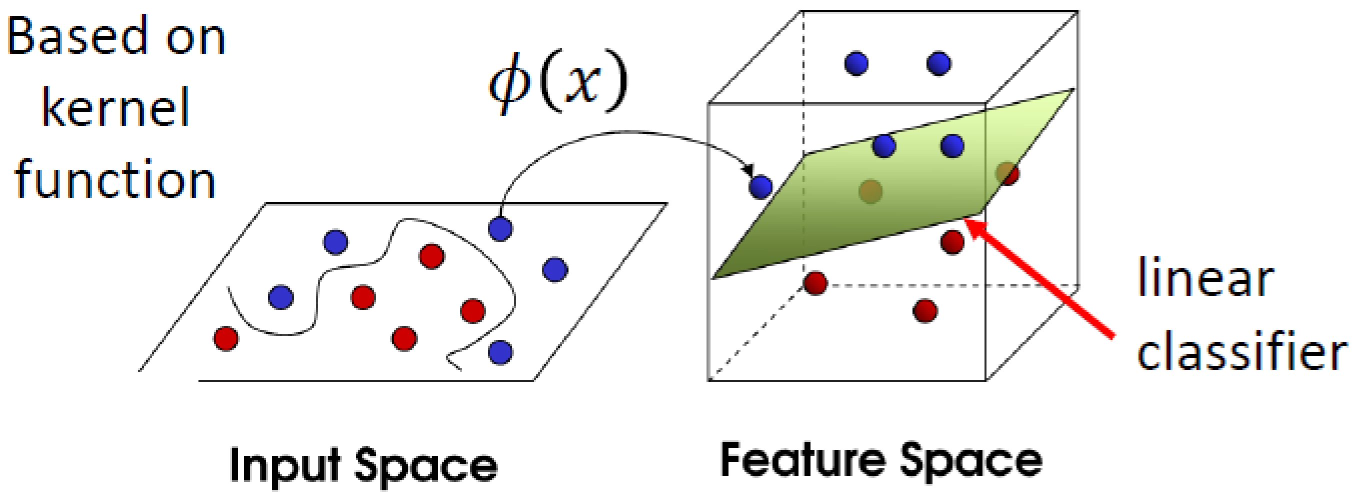
$$\ell(w,b) = \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N} max\left(0, 1 - \left(w^T x^{(i)} + b\right)t^{(i)}\right)$$



Deep Learning



SVM



Readings

- SVM implementations
  - http://www.kernel-machines.org/software
- Burges, Christopher JC. "A tutorial on support vector machines for pattern recognition." Data mining and knowledge discovery 2.2 (1998): 121-167.
- Sections 6.1-6.2 in the book “Pattern Recognition and Machine Learning”, by Christopher M. Bishop, Springer, 2006.
- SVM - Understanding the math - Duality and Lagrange multipliers
  - https://www.svm-tutorial.com/2016/09/duality-lagrange-multipliers/
- MIT OpenCourseWare: Learning: Support Vector Machines
  - https://www.youtube.com/watch?v=_PwhiWxHK8o
- Support Vector Regression (SVR): Section 7.1.4 in the book “Pattern Recognition and Machine Learning”, by Christopher M. Bishop, Springer, 2006.