Support Vector Machines

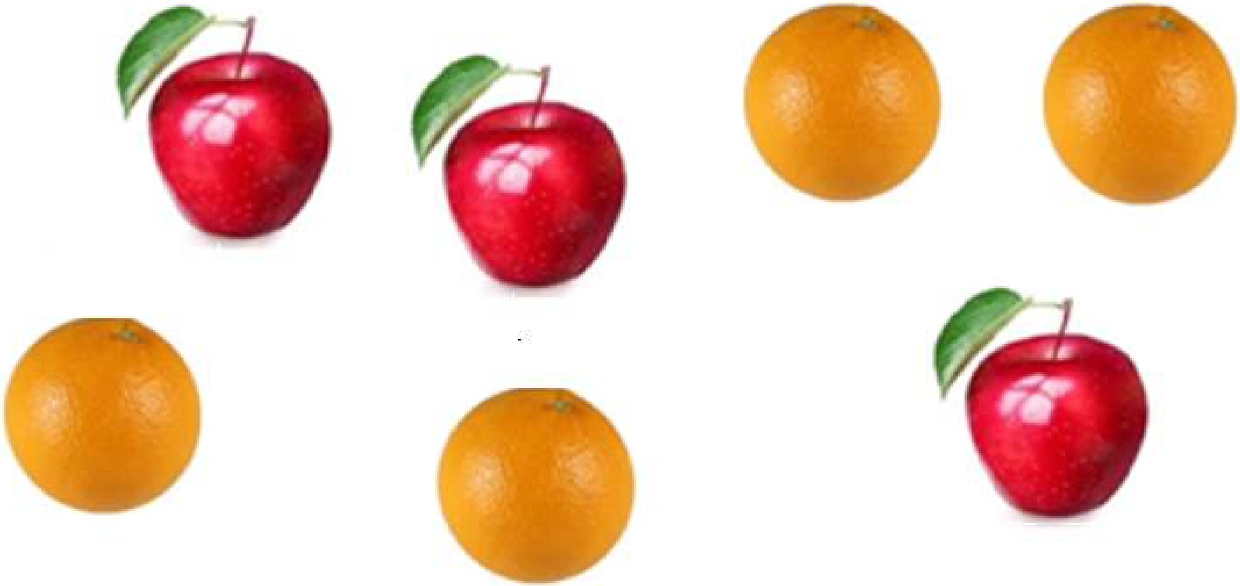
Ms. Sandhya Harikumar,

Department of Computer Science & Engineering,

Amrita Vishwa Vidyapeetham, Amritapuri

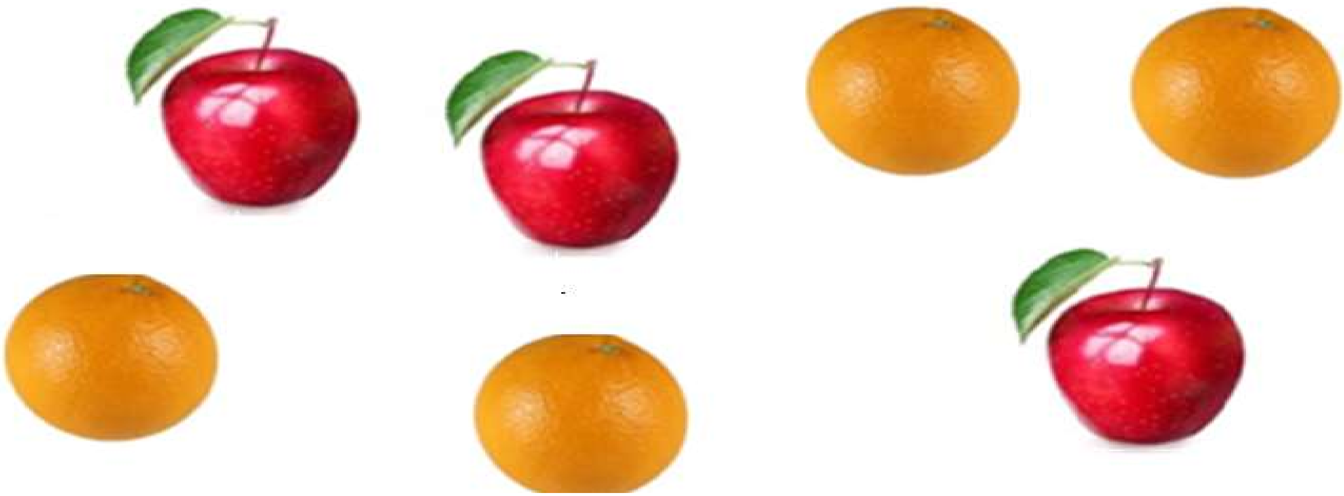
Maximum Margin Hyperplane

Classification



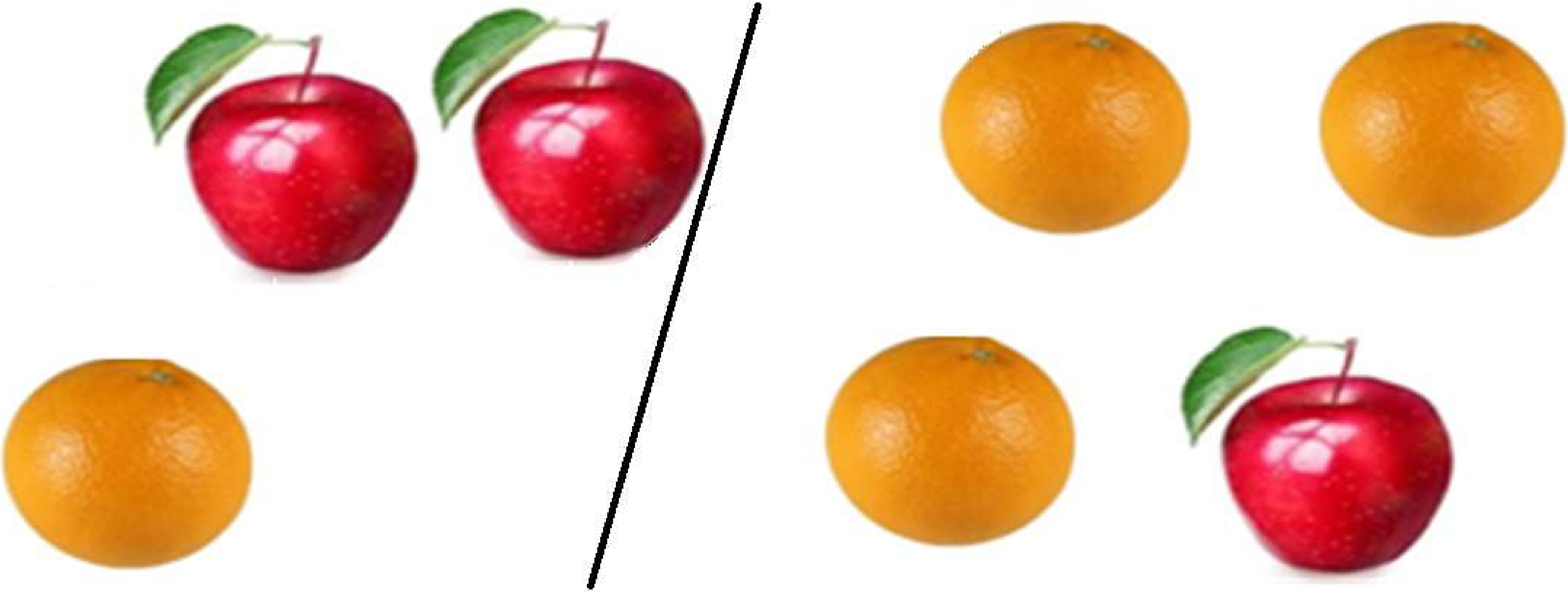
- Generate a model(target function) that predicts a class label for a given instance from pre-classified patterns

kNN Classifier



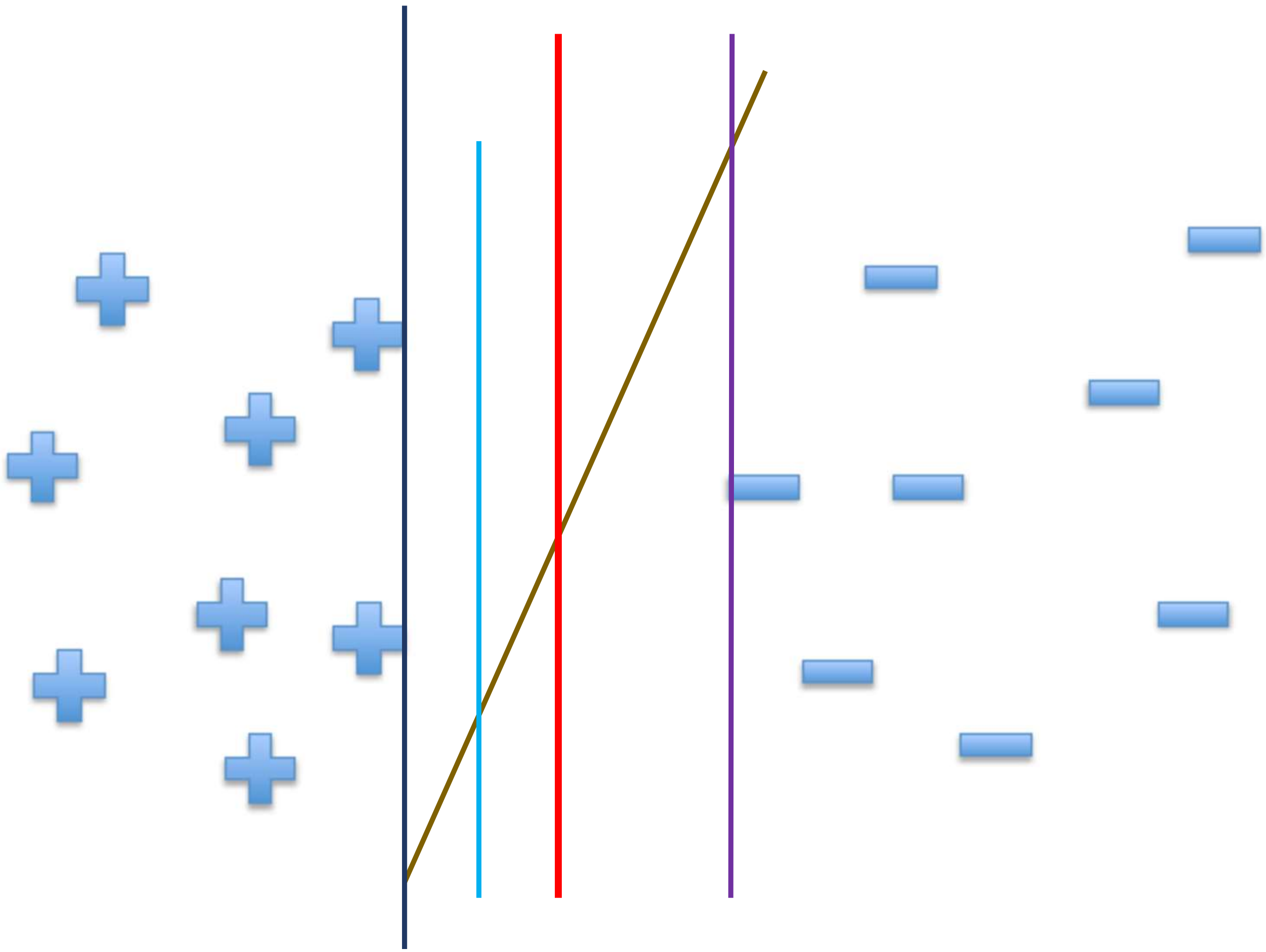
- Computationally expensive
- Generally suffers from high variance

Logistic Regression

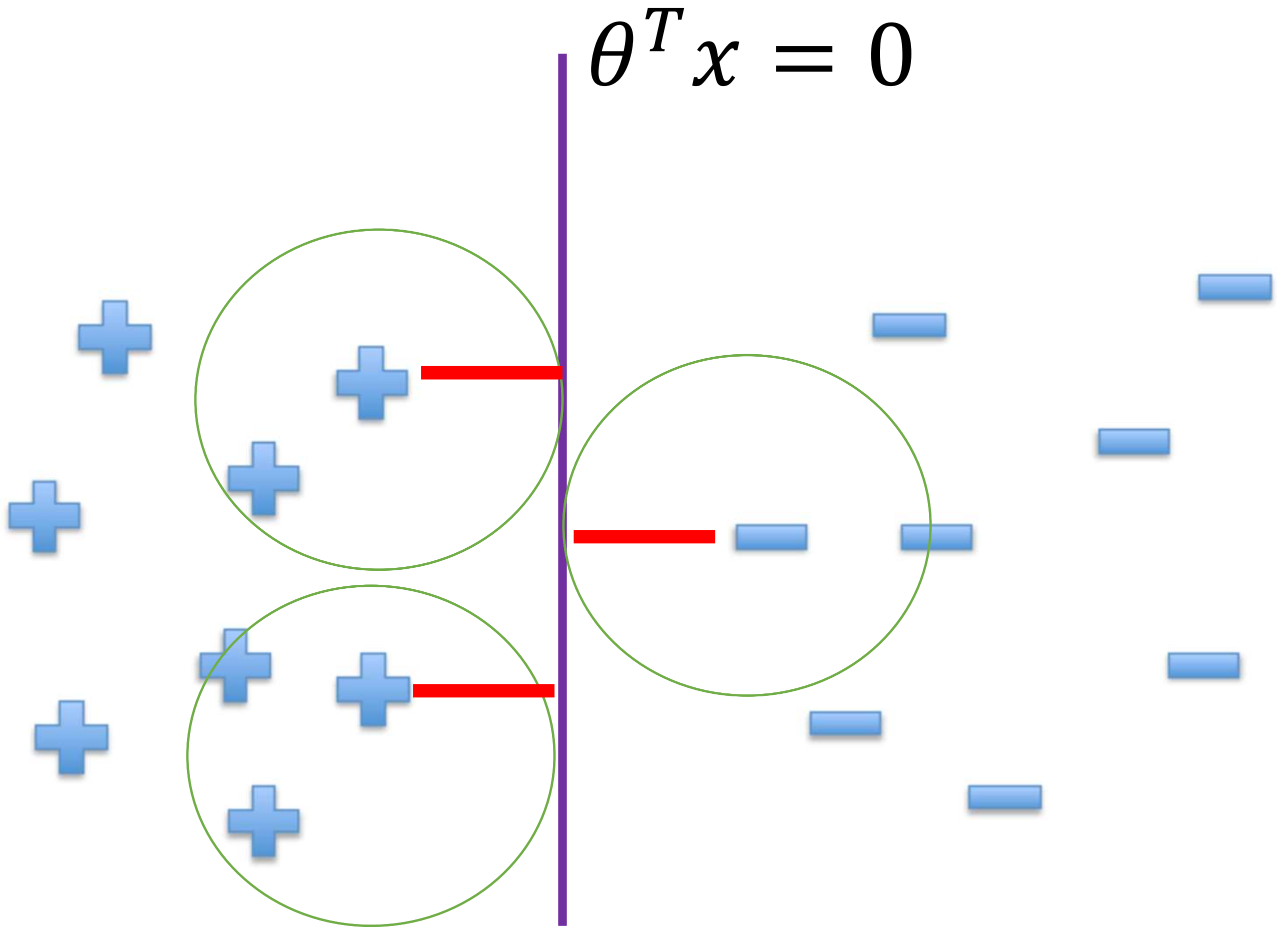


- $h_\theta(x) = g(\theta^T x) = \frac{1}{1+e^{-\theta^T x}}$

Linear Separators

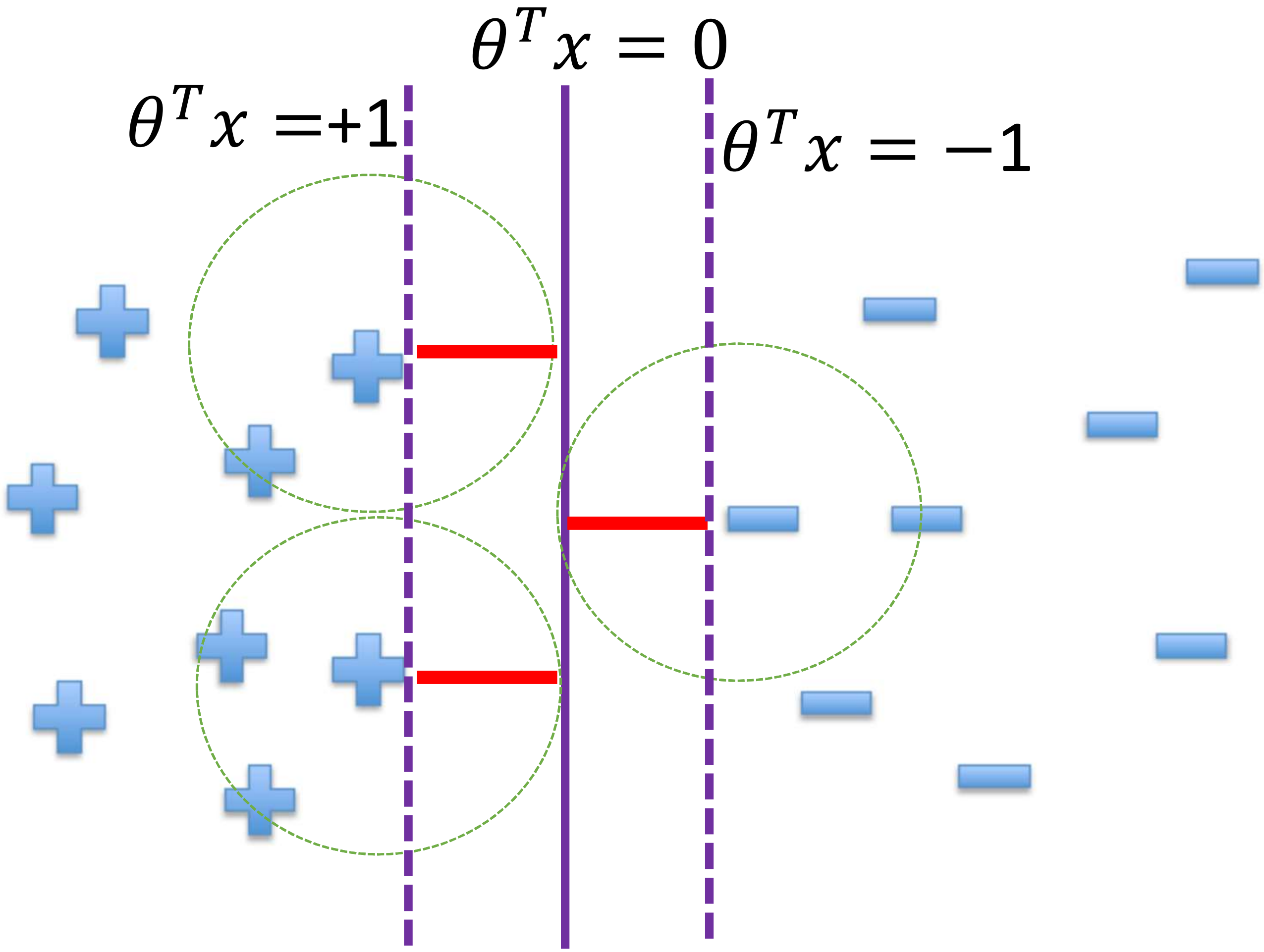


Good Separator : Maximizing the margin



Maximum Margin Hyperplane

1. To reduce the possible number of models.
2. To reduce generalization errors



Optimization Objective
Support Vector Machines

Training Set

- $\{(x_i, y_i), i = 1, \ldots, n\}, x_i \in \mathbb{R}^m, y_i \in \{+1, -1\}$

- If Linearly separable dataset

- $\theta^T x = 0 \quad ----\rightarrow \quad \textit{Separating hyperplane}$
- $\theta^T x_i > 0, \forall i \quad \textit{such that } y_i = +1$
- $\theta^T x_i < 0, \forall i \quad \textit{such that } y_i = -1$

Finite Training Set

- $\exists \epsilon > 0 \; such \; that$
- $\theta^T x_i \geq \quad \epsilon, \;\; \forall i \quad such \; that \; y_i = +1 \quad (A)$
- $\theta^T x_i \leq -\epsilon, \forall i \quad such \; that \; y_i = -1 \quad (B)$

Scaling θ

- $\theta^T x_i \geq +1, \ \forall i \quad such\ that\ y_i = +1 \quad (A)$
- $\theta^T x_i \leq -1, \forall i \quad such\ that\ y_i = -1 \quad (B)$

- Combining (A) and (B) gives

$$yi(\theta^T x_i) \geq 1, \ \ \forall i$$

Separating hyperplanes

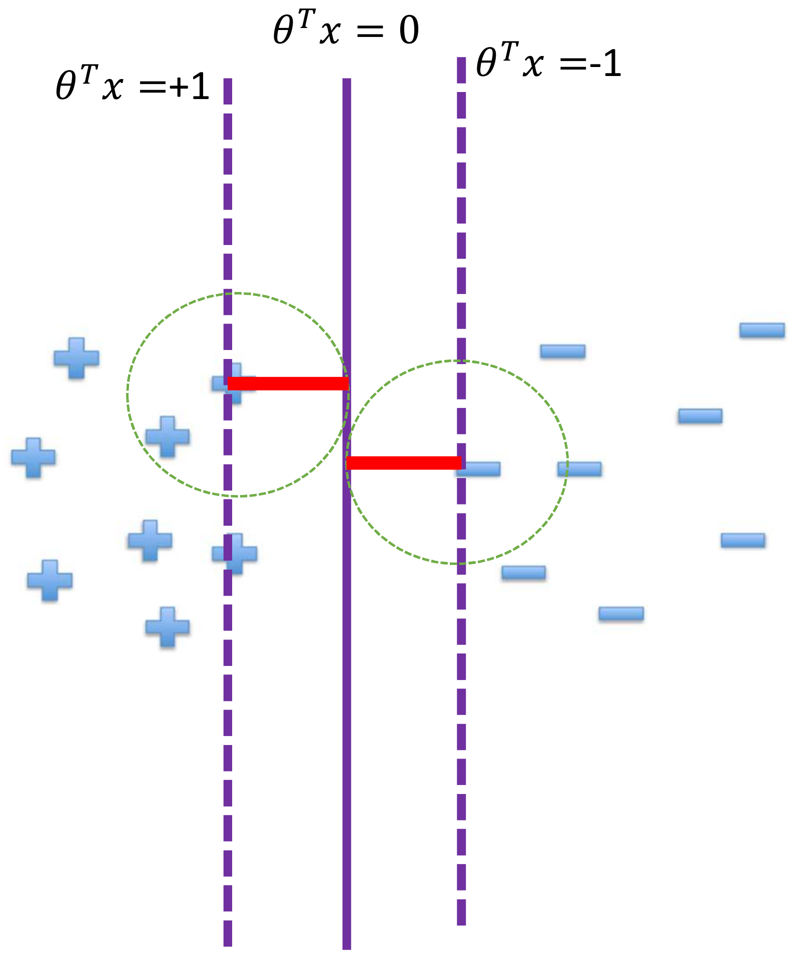
- No training samples between the two parallel hyperplanes

$$\theta^T x = +1,$$

$$and$$

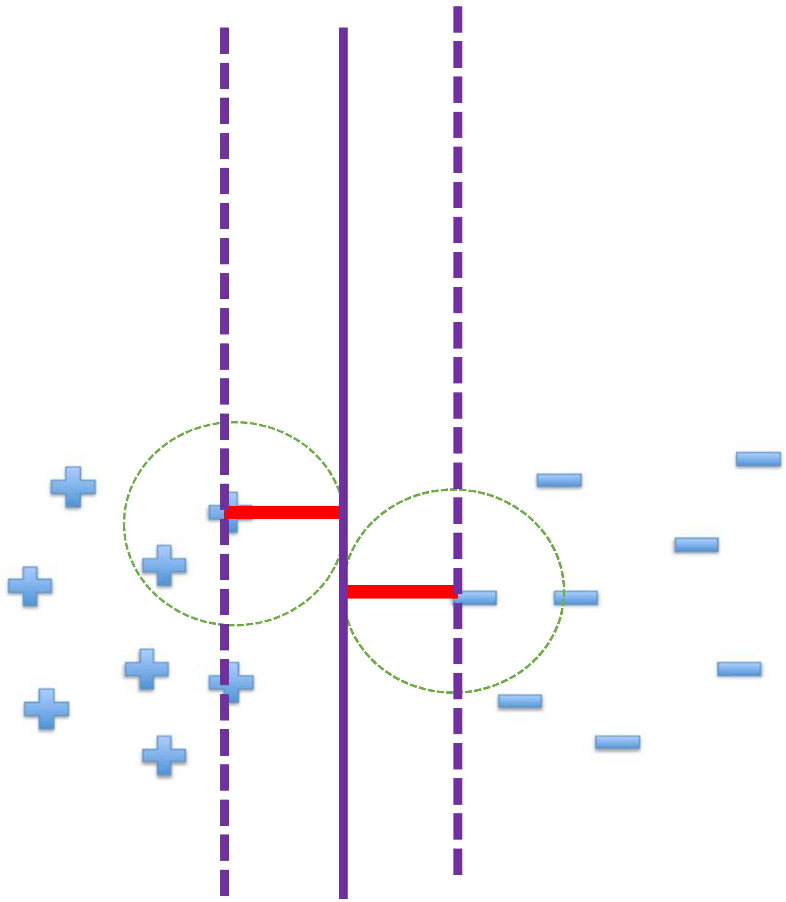
$$\theta^T x = -1,$$

Distance between two hyperplanes = $\frac{2}{\|\theta\|}$



Support Vectors

- The closest pattern/sample to the optimal hyperplane
- Distance between hyperplane and the Support vector is $\frac{1}{\|\theta\|}$



Optimal Hyperplane : Constrained Optimization Problem

- Maximize $\frac{2}{\|\theta\|}$

- Thus,

  Minimize $\frac{1}{2}\|\theta\|$

- i.e.

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

  Subject to constraint $yi(\theta^T x_i) \geq 1, \ \ \forall i$

Primal Optimization Problem

SVM Dual Problem

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

Subject to constraint $yi(\theta^T x_i) \geq 1, \quad \forall i$

Number of constraints = Number of training samples

The problem can be solved using Lagrangian dual

Transforms a difficult optimization problem into a simpler one

Key idea : Introduce slack variables α_i for each constraint

α_i indicates how important a constraint is.

Lagrangian Duality

- $L(\theta, \alpha) = \frac{1}{2}\sum_{j=1}^{d} \theta_j^2 - \sum_{i=1}^{n} \alpha_i (y_i \theta^T x - 1)$

  $s.t. \alpha_i \geq 0, \forall i$

Minimize over θ and maximize over α

SVM Dual Representation

$$Maximize \quad J(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle$$

$$s.t. \, \alpha_i \geq 0, \forall i$$
$$\sum_i \alpha_i y_i = 0$$

α_i *Values and support vectors*

- Point is a support vector if

$\alpha_i > 0$ and the constraint $yi(\theta^T x_i) = 1$

- Point is not a support vector if
- $\alpha_i = 0$

Compute Optimal θ

- $\theta^* = \sum_{x_i \in sv} \alpha_i^* y_i x_i$
- sv = set of Support vectors

Takeaways

- Quadratic Cost function and linear constraints using Duality problem
- Training data vectors appear only as inner product
- Optimization is over number of training samples irrespective over the features of the dataset

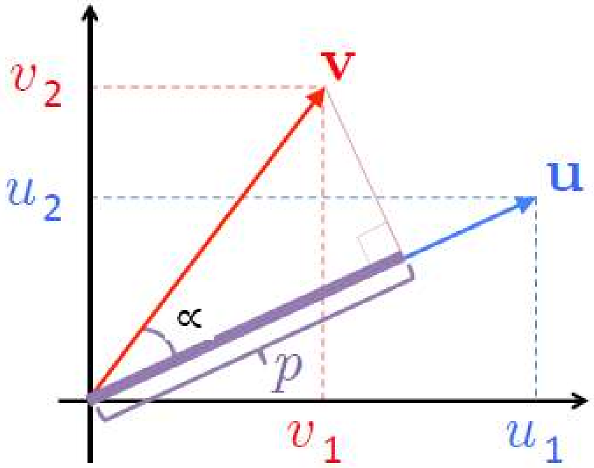
References

- https://towardsdatascience.com/the-kernel-trick-c98cdbcaeb3f#:~:text=The%20%E2%80%9Ctrick%E2%80%9D%20is%20that%20kernel,the%20data%20by%20these%20transformed
- https://nptel.ac.in/content/storage2/courses/117108048/module9/Lecture31.pdf
- http://cs229.stanford.edu/notes/cs229-notes3.pdf

Large Margin Classifier

Support Vector Machines

Vector Inner Product

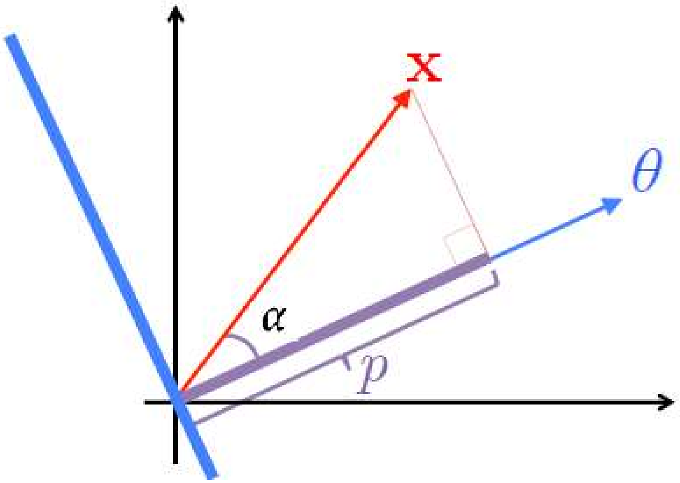


$$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} ; \vartheta = \begin{bmatrix} \vartheta_1 \\ v_2 \end{bmatrix}$$

$$\|u\|_2 = \sqrt{u_1^2 + u_2^2}$$

$$\begin{aligned} u^T v = v^T u \\ &= u_1 v_1 + u_2 v_2 \\ &= \|u\| \|v\| \cos \alpha \\ &= p \|u\|_2 \quad where\ p = ||v|| \cos \alpha \end{aligned}$$

Hyperplane



$$\theta^T x \quad = \|\theta\|\|x\| \cos\alpha$$
$$= p\|\theta\|_2$$

$If\ \ p\|\theta\|_2$ >= 1; y=1

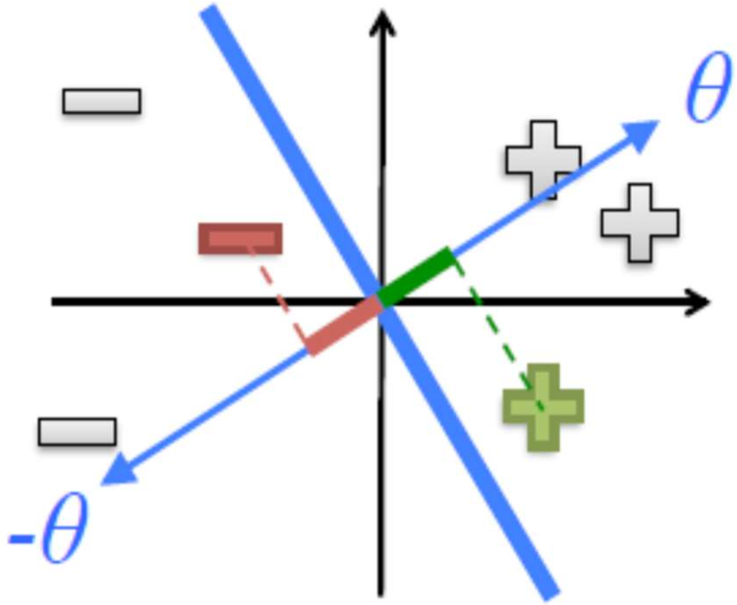
$If\ \ p\|\theta\|_2$ <= -1; y=-1

Thus, $p\|\theta\|_2$ = 1 leads to

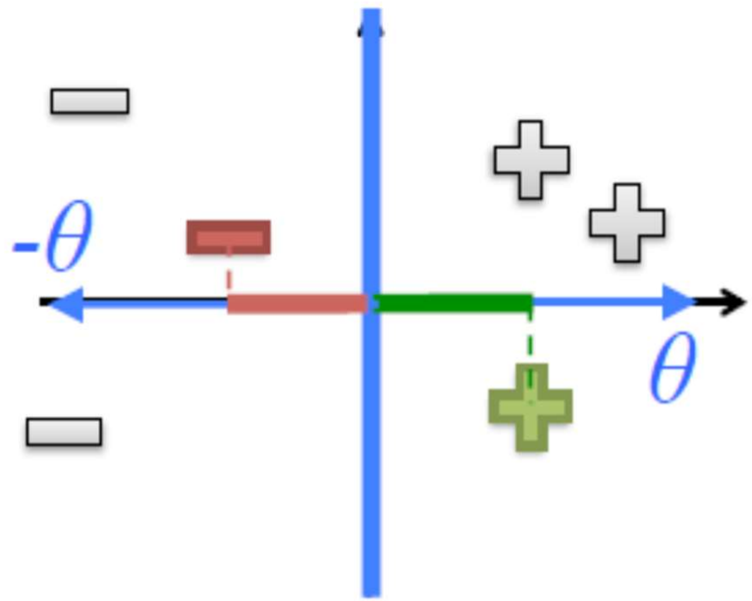
$$p = \frac{1}{\|\theta\|};$$

Maximizing the margin

Let *pi* be the projection of **x***i* onto the vector $\boldsymbol{\theta}$



Since p is small, therefore $\|\boldsymbol{\theta}\|_2$ must be large to have $p\|\boldsymbol{\theta}\|_2 \geq 1$ (or ≤ -1)

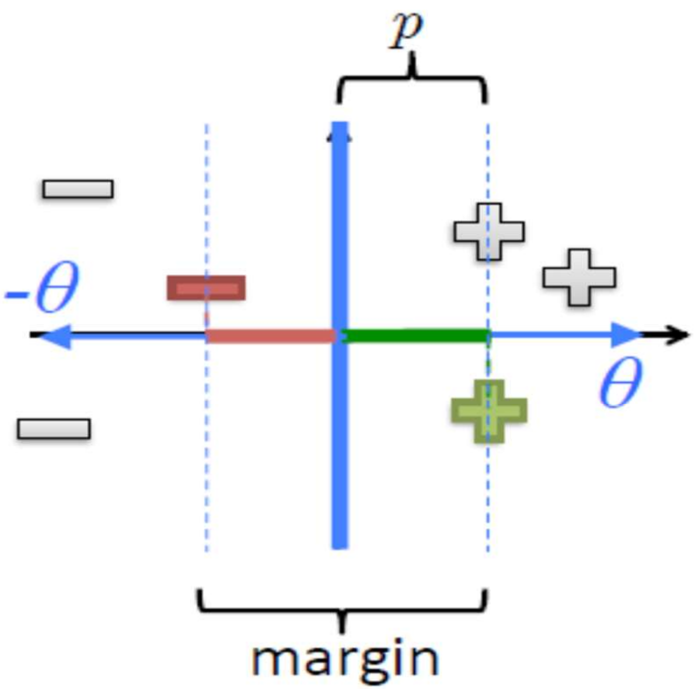


Since p is larger, $\|\boldsymbol{\theta}\|_2$ can be smaller in order to have $p\|\boldsymbol{\theta}\|_2 \geq 1$ (or ≤ -1)

Margin Size

For the support vectors, we have $p\|\boldsymbol{\theta}\|_2 = \pm 1$

- p is the length of the projection of the SVs onto $\boldsymbol{\theta}$



Therefore,

$$p = \frac{1}{\|\boldsymbol{\theta}\|_2}$$

$$\text{margin} = 2p = \frac{2}{\|\boldsymbol{\theta}\|_2}$$

Maximizing Margin : Constrained Optimization Objective

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

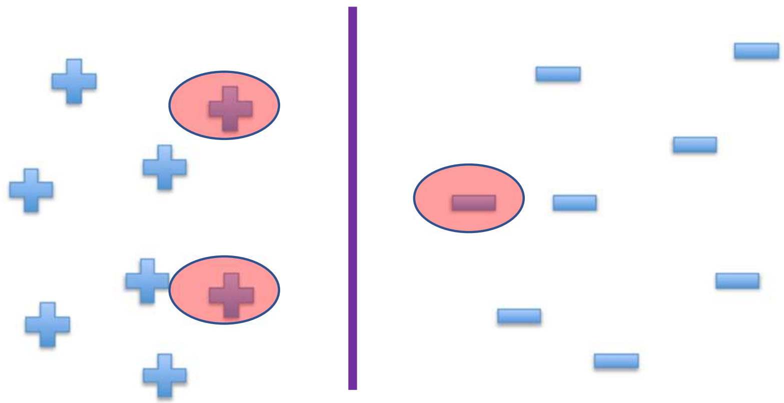
$$\text{s.t. } \boldsymbol{\theta}^\intercal \mathbf{x}_i \geq 1 \quad \text{if } y_i = 1$$
$$\boldsymbol{\theta}^\intercal \mathbf{x}_i \leq -1 \quad \text{if } y_i = -1$$

$$\text{s.t. } y_i(\boldsymbol{\theta}^\intercal \mathbf{x}_i) \geq 1 \ \ \forall i$$

Support Vectors

- Margin : Perpendicular distance from the optimal hyperplane to only the closest points from each of the classes
- These points/samples are called support vectors since they support or define the hyperplane

$$\theta^T x = 0$$



References

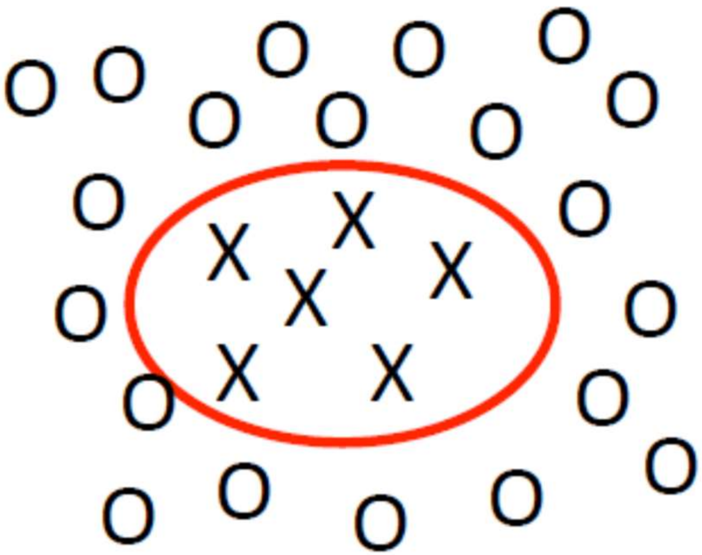
- https://towardsdatascience.com/the-kernel-trick-c98cdbcaeb3f#:~:text=The%20%E2%80%9Ctrick%E2%80%9D%20is%20that%20kernel,the%20data%20by%20these%20transformed

- https://nptel.ac.in/content/storage2/courses/117108048/module9/Lecture31.pdf

- http://cs229.stanford.edu/notes/cs229-notes3.pdf

Kernel Trick
Support Vector Machines

Non-linearly Separable dataset



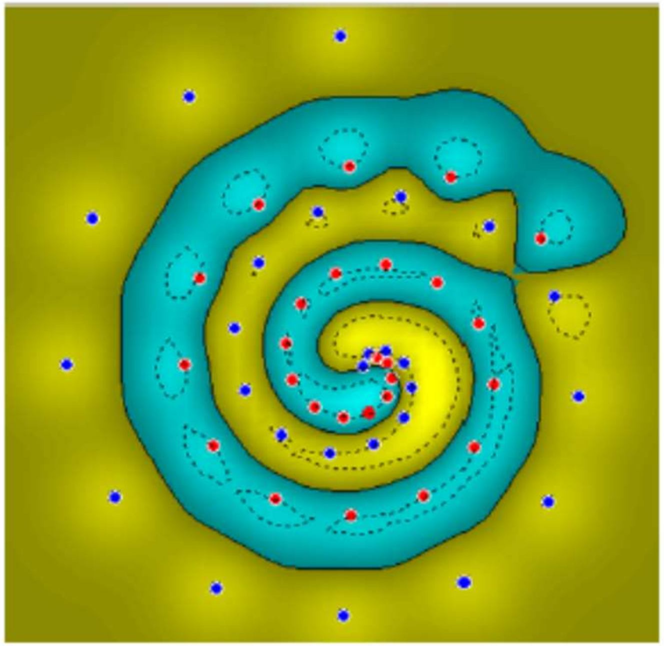
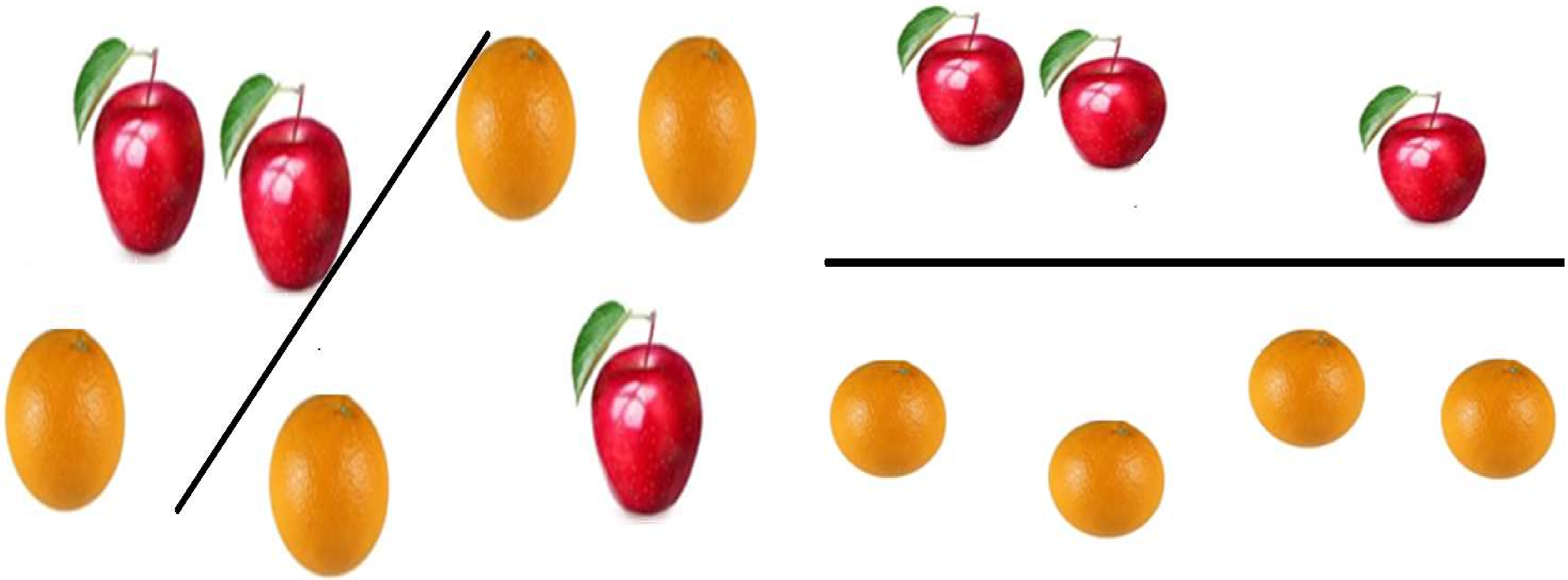


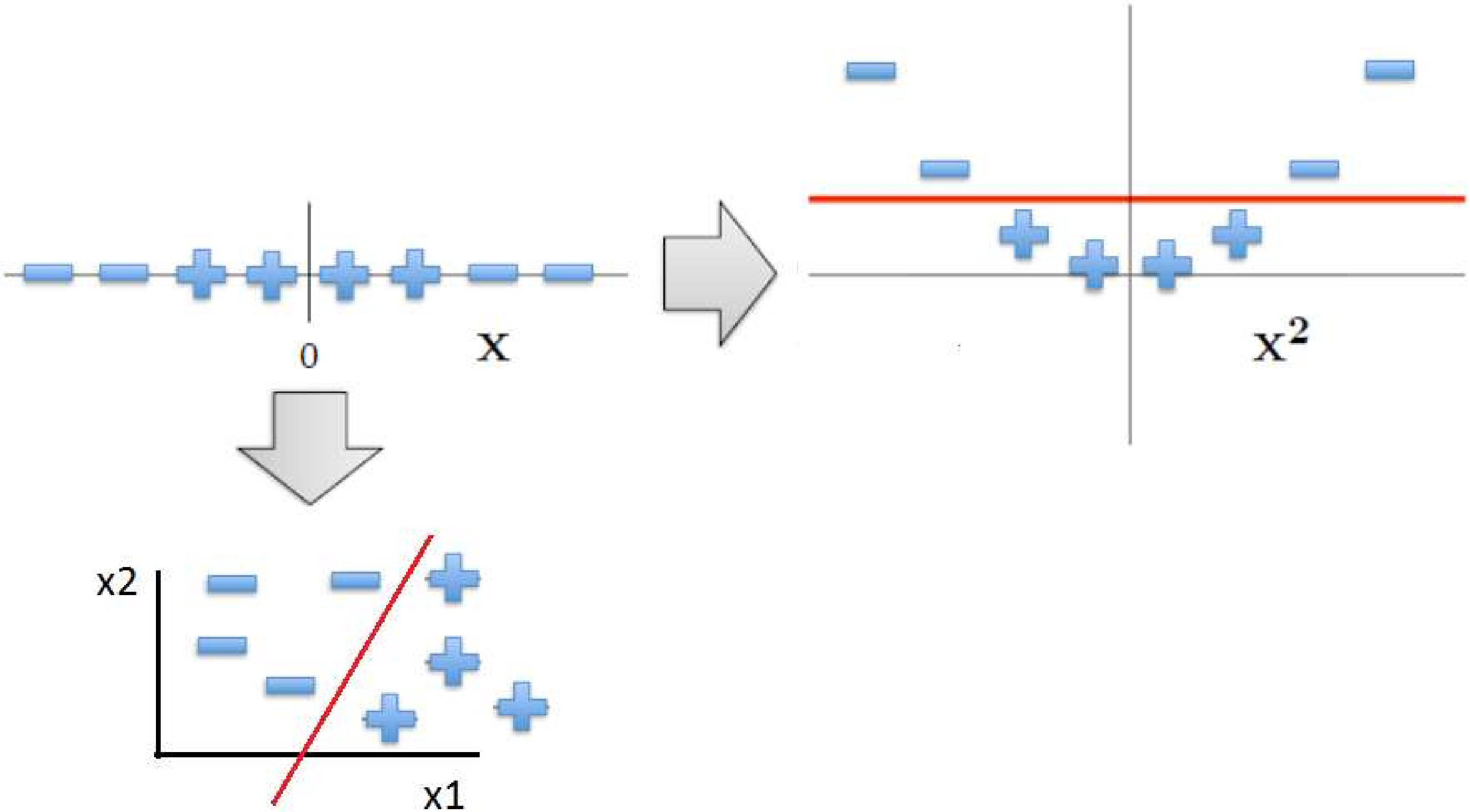
Image from http://www.atrandomresearch.com/iclass/

Map Input samples to some higher space



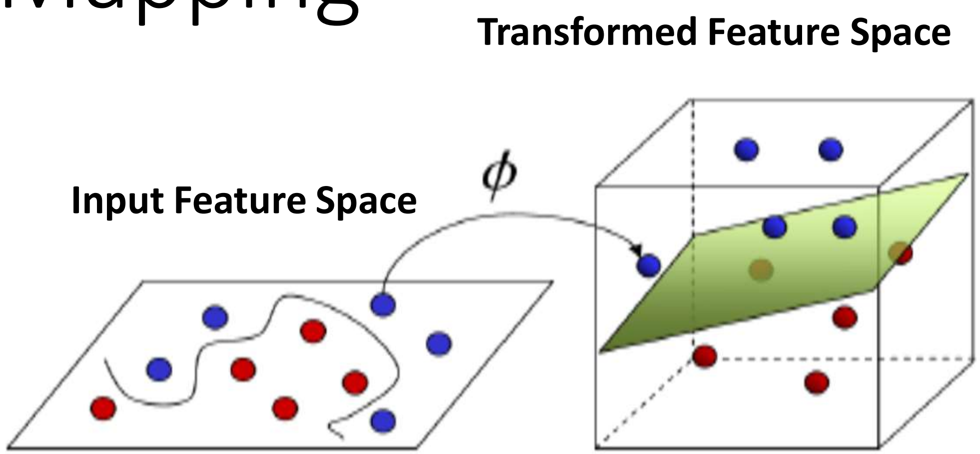
Representational Bias

- Having right features/ or more features for samples is crucial



Low to High-dimensional Mapping

$Let\ X = [x1\ , x2]$



$Let\ \phi = \mathbb{R}^2 \rightarrow \mathbb{R}^5\ given\ by$
$Z = \phi(x) = [1, x_1, x_2, x_1^2, x_2^2, x_1 x_2]$

Quadratic function in $\mathbb{R}^2$

$g(x) = a_0 + a_1 x_1 + a_2 x_2 + a_3 x_1^2 + a_4 x_2^2 + a_5 x_1 x_2$

$z_1 = x_1; z_2 = x_2; z_3 = x_1^2\ ; z_4 = x_2^2; z_5 = x_1 x_2$

Linear function in $\mathbb{R}^5$

$g(x) = a_0 + a_1 z_1 + a_2 z_2 + a_3 z_3 + a_4 z_4 + a_5 z_5$

Major Issue : Big $\left(\phi(x_i)\right)$

- P^{th} degree polynomial discriminant function in the original feature space ($\mathbb{R}^m$), leads to transformed feature vector, Z, with dimension $\mathrm{O}(m^p)$
- Learn $\mathrm{O}(m^p)$ parameters rather than m parameters
- Need huge number of training samples for achieving proper generalization.
- Extremely high and non-feasible computational costs.

Kernel

- A function that acts as a modified dot product
- Performs operations on the original space and returns the result in the transformed space.
- $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$

- How to choose a mapping function so that linear SVM can be directly applied.

Kernel Trick!!!!

Kernel trick

- Explicity mapping not required.

- $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$

- Use $K(x_i, x_j)$ in SVM

Kernel Example

Let $\mathbf{x}_i = [x_{i1}, x_{i2}]$ and $\mathbf{x}_j = [x_{j1}, x_{j2}]$

Consider the following function:

$$\begin{aligned} K(\mathbf{x}_i, \mathbf{x}_j) &= \langle \mathbf{x}_i, \mathbf{x}_j \rangle^2 \\ &= (x_{i1}x_{j1} + x_{i2}x_{j2})^2 \\ &= (x_{i1}^2 x_{j1}^2 + x_{i2}^2 x_{j2}^2 + 2x_{i1}x_{i2}x_{j1}x_{j2}) \\ &= \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle \end{aligned}$$

where

$$\Phi(\mathbf{x}_i) = [x_{i1}^2, x_{i2}^2, \sqrt{2}x_{i1}x_{i2}]$$

$$\Phi(\mathbf{x}_j) = [x_{j1}^2, x_{j2}^2, \sqrt{2}x_{j1}x_{j2}]$$

Incorporating Kernels into SVM

$$Maximize \quad J(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle$$

$$Maximize \quad J(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

$$s.t. \alpha_i \geq 0, \forall i$$

$$\sum_i \alpha_i y_i = 0$$

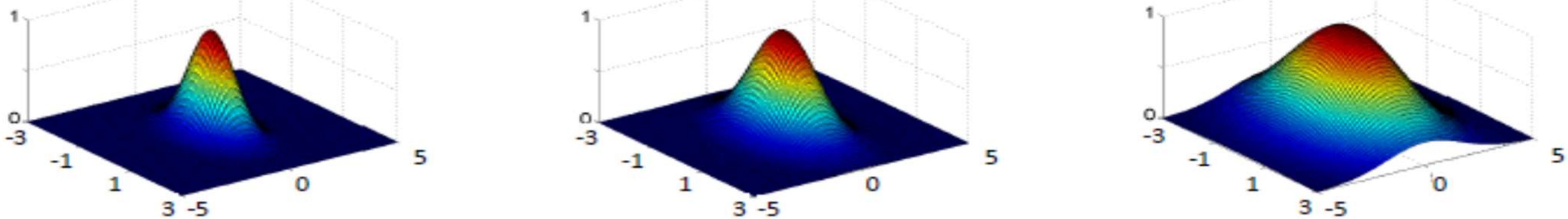
The Gaussian Kernel

- Also called Radial Basis Function (RBF) kernel

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2}\right)$$

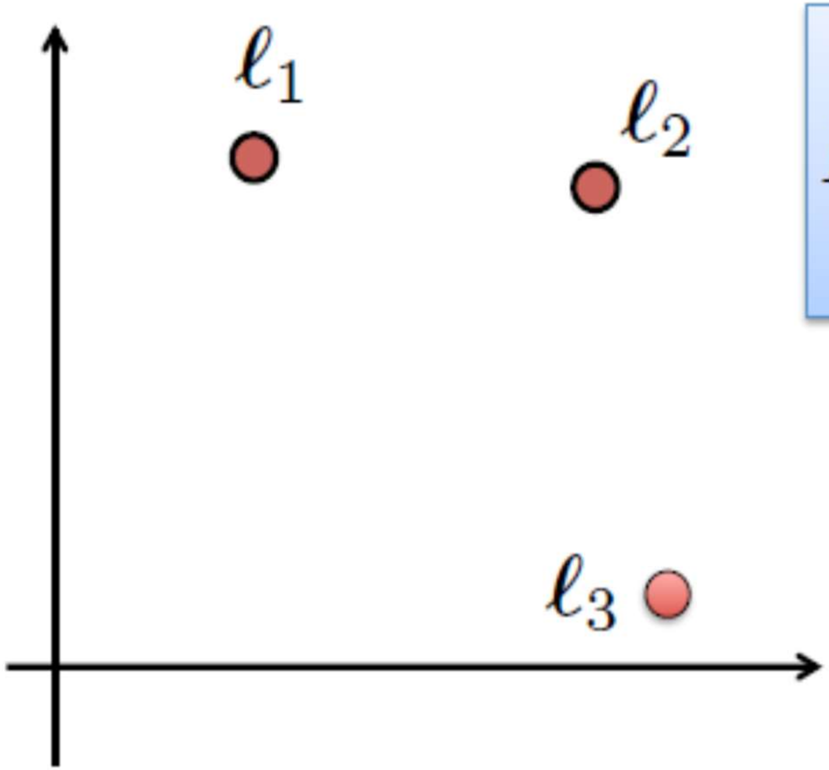
  - Has value 1 when $\mathbf{x}_i = \mathbf{x}_j$
  - Value falls off to 0 with increasing distance
  - Note: Need to do feature scaling <u>before</u> using Gaussian Kernel

$\sigma^2 = 0.5$ $\sigma^2 = 1$ $\sigma^2 = 3$



lower bias, higher variance ⟷ higher bias, lower variance

Gaussian Kernel Example



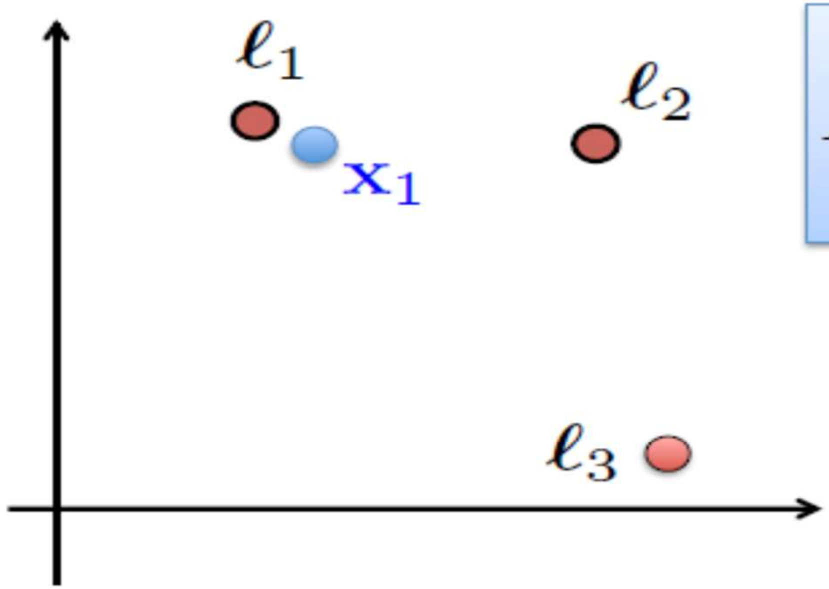
$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2}\right)$$

Imagine we've learned that:

$$\boldsymbol{\theta} = [-0.5, 1, 1, 0]$$

Predict +1 if $\theta_0 + \theta_1 K(\mathbf{x}, \boldsymbol{\ell}_1) + \theta_2 K(\mathbf{x}, \boldsymbol{\ell}_2) + \theta_3 K(\mathbf{x}, \boldsymbol{\ell}_3) \geq 0$

Gaussian Kernel Example



$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2}\right)$$

Imagine we've learned that:

$$\boldsymbol{\theta} = [-0.5, 1, 1, 0]$$

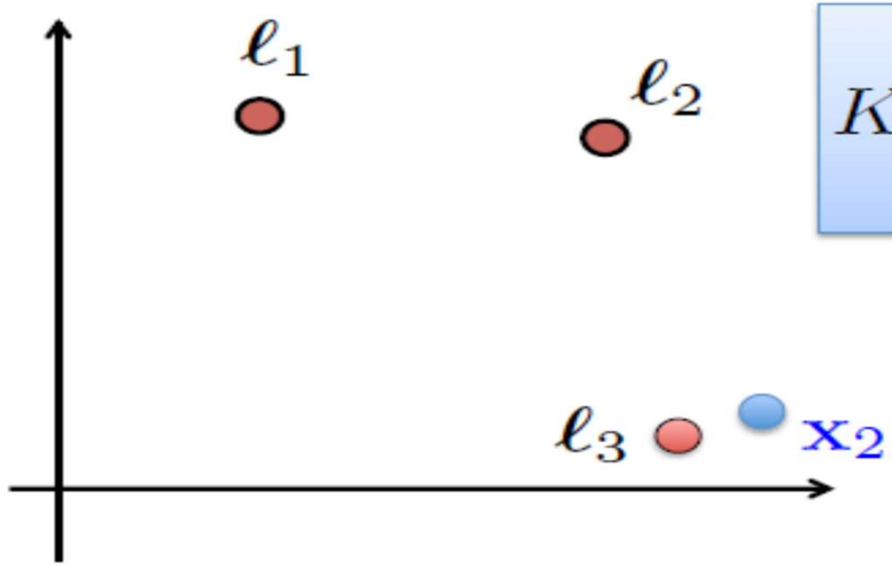
Predict +1 if $\theta_0 + \theta_1 K(\mathbf{x}, \boldsymbol{\ell}_1) + \theta_2 K(\mathbf{x}, \boldsymbol{\ell}_2) + \theta_3 K(\mathbf{x}, \boldsymbol{\ell}_3) \geq 0$

- For $\mathbf{x}_1$, we have $K(\mathbf{x}_1, \ell_1) \approx 1$, other similarities ≈ 0

$$\begin{aligned} &\theta_0 + \theta_1(1) + \theta_2(0) + \theta_3(0) \\ &\quad = -0.5 + 1(1) + 1(0) + 0(0) \\ &\quad = 0.5 \geq 0 \end{aligned}$$

, so predict +1

Gaussian Kernel Example



$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2}\right)$$

Imagine we've learned that:

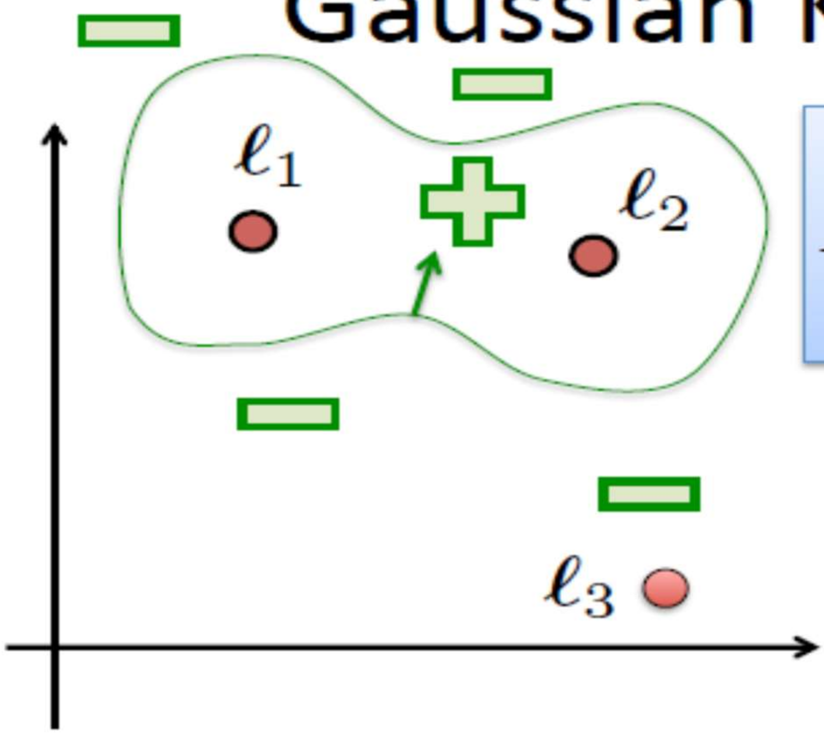
$$\boldsymbol{\theta} = [-0.5, 1, 1, 0]$$

Predict +1 if $\theta_0 + \theta_1 K(\mathbf{x}, \boldsymbol{\ell}_1) + \theta_2 K(\mathbf{x}, \boldsymbol{\ell}_2) + \theta_3 K(\mathbf{x}, \boldsymbol{\ell}_3) \geq 0$

- For $\mathbf{x}_2$, we have $K(\mathbf{x}_2, \ell_3) \approx 1$, other similarities ≈ 0

$$\theta_0 + \theta_1(0) + \theta_2(0) + \theta_3(1)$$
$$= -0.5 + 1(0) + 1(0) + 0(1)$$
$= -0.5 < 0$, so predict -1

Gaussian Kernel Example



$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2}\right)$$

Imagine we've learned that:

$$\boldsymbol{\theta} = [-0.5, 1, 1, 0]$$

Predict +1 if $\theta_0 + \theta_1 K(\mathbf{x}, \boldsymbol{\ell}_1) + \theta_2 K(\mathbf{x}, \boldsymbol{\ell}_2) + \theta_3 K(\mathbf{x}, \boldsymbol{\ell}_3) \geq 0$

Rough sketch of decision surface

A Few Good Kernels...

- Linear Kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$
- Polynomial kernel $K(\mathbf{x}_i, \mathbf{x}_j) = (\langle \mathbf{x}_i, \mathbf{x}_j \rangle + c)^d$
  - $c \geq 0$ trades off influence of lower order terms
- Gaussian kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2}\right)$
- Sigmoid kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \tanh\left(\alpha \mathbf{x}_i^\intercal \mathbf{x}_j + c\right)$

Takeaways

- Low to high dimensional space
- Kernel functions
- Kernel trick
- SVM applies kernel trick to generate a hyperplane for non-linearly separable dataset
- Some examples of Kernel functions

References

- https://towardsdatascience.com/the-kernel-trick-c98cdbcaeb3f#:~:text=The%20%E2%80%9Ctrick%E2%80%9D%20is%20that%20kernel,the%20data%20by%20these%20transformed

- http://cs229.stanford.edu/notes/cs229-notes3.pdf