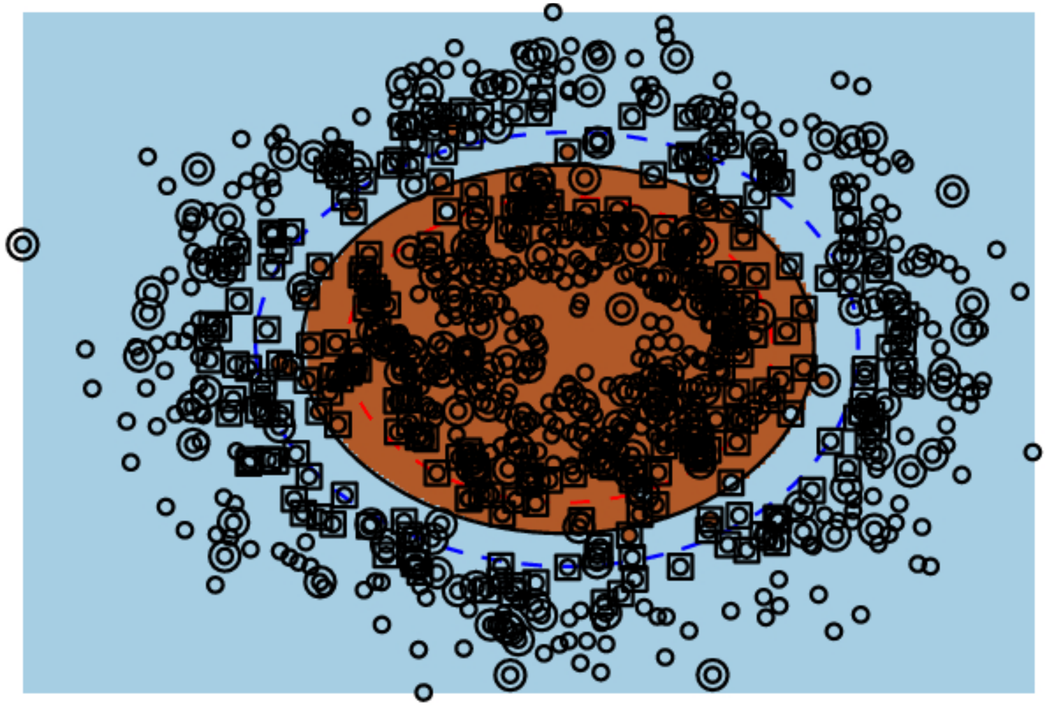
2.160 Identification, Estimation, and Learning

Part 4 Machine Learning and Nonlinear System Modeling

Lecture 22

Support Vector Machines and Kernel Methods



H. Harry Asada
Department of Mechanical Engineering
MIT

The Final Stretch of 2.160

❑ 11/18: Deep Learning: CNN and RNN — Problem 1

❑ 11/30:Support Vector Machines and Kernel Methods — Problem 2

COP-4 due

PS#6

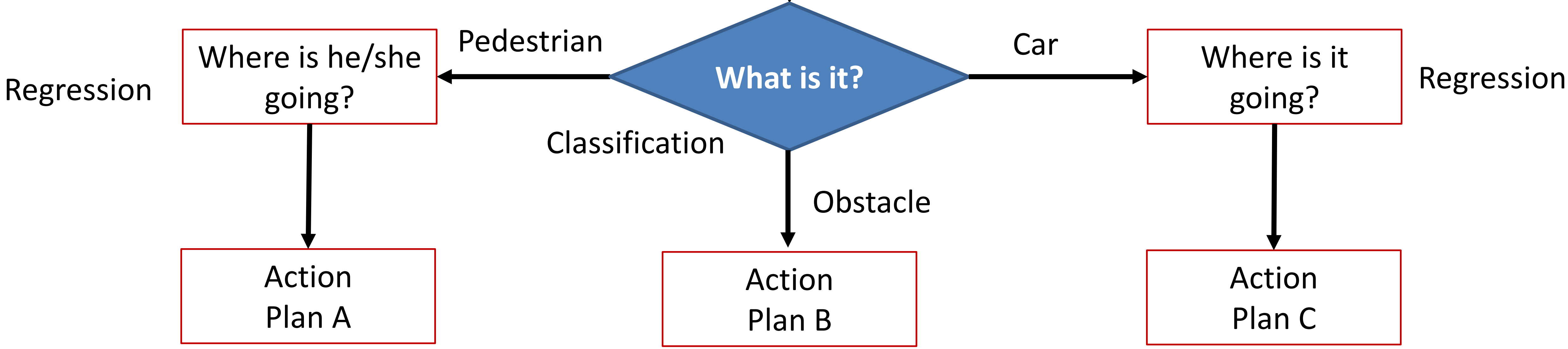
❑ 12/2: Gaussian Processes — Problem 3

12/4: Final study group meeting

❑ 12/7: Koopman Operator Theory for Exact Linearization of Nonlinear Dynamical Systems — Problem 4

❑ 12/9: Dual Faceted Linearization with Application to Model Predictive Control — PS#6 Due

Self-Driving Car



What is it?

Classification

Pedestrian

Where is he/she going?

Regression

Action Plan A

Obstacle

Action Plan B

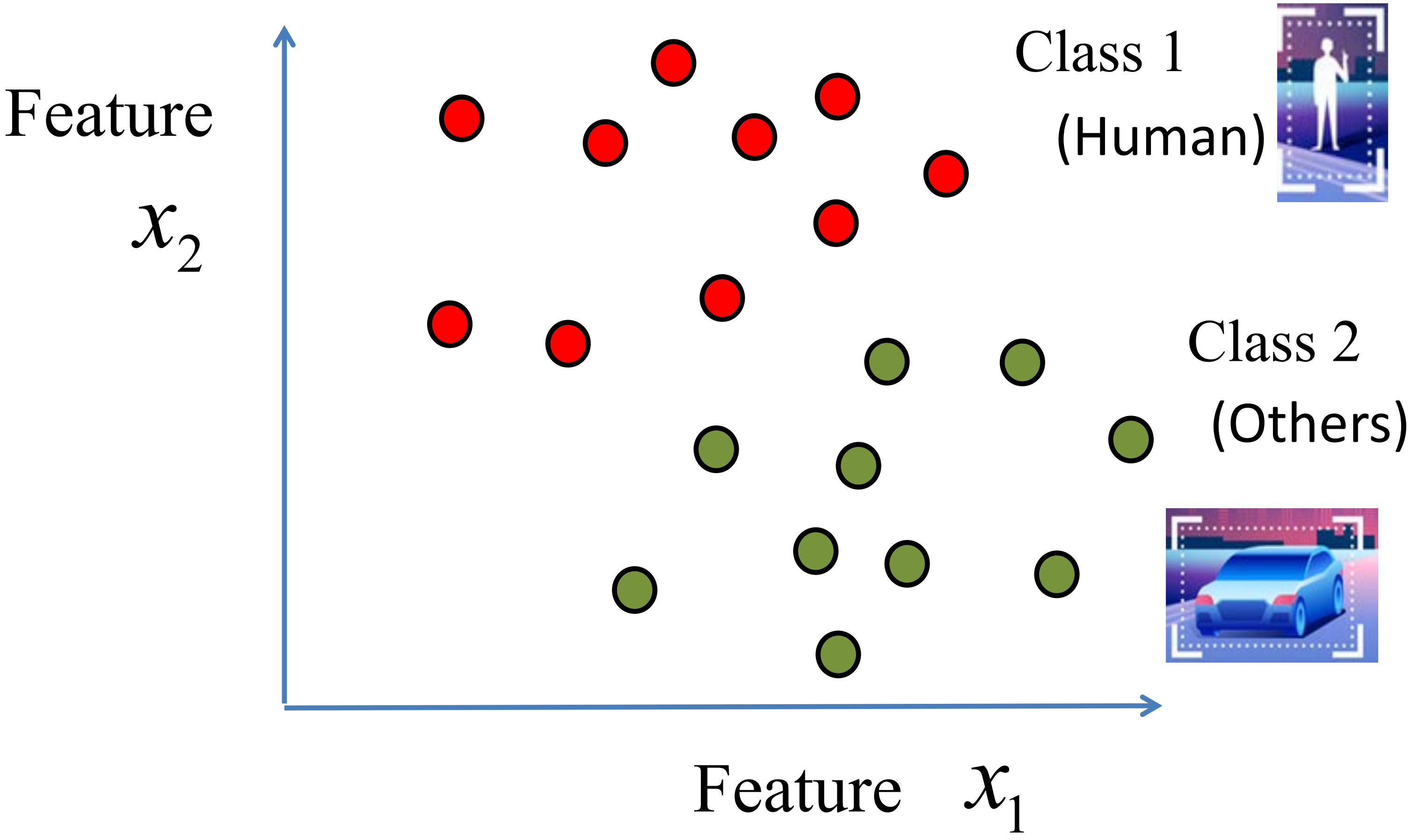
Car

Where is it going?

Regression

Action Plan C

Classification, Detection, Decision



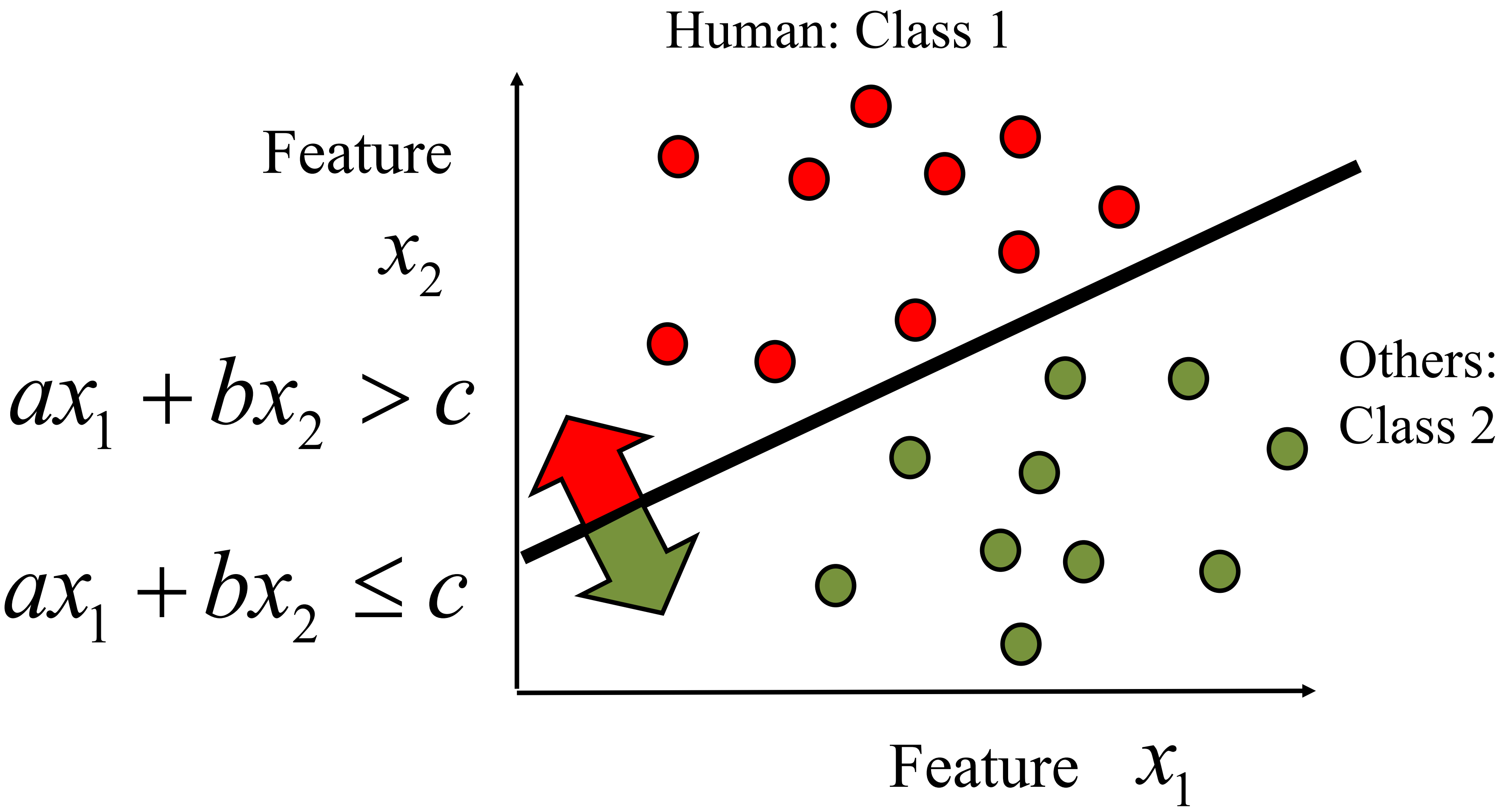
Feature x_2

Feature x_1

Class 1 (Human)

Class 2 (Others)

Pattern Classification



Human: Class 1

Feature x_2

$ax_1 + bx_2 > c$

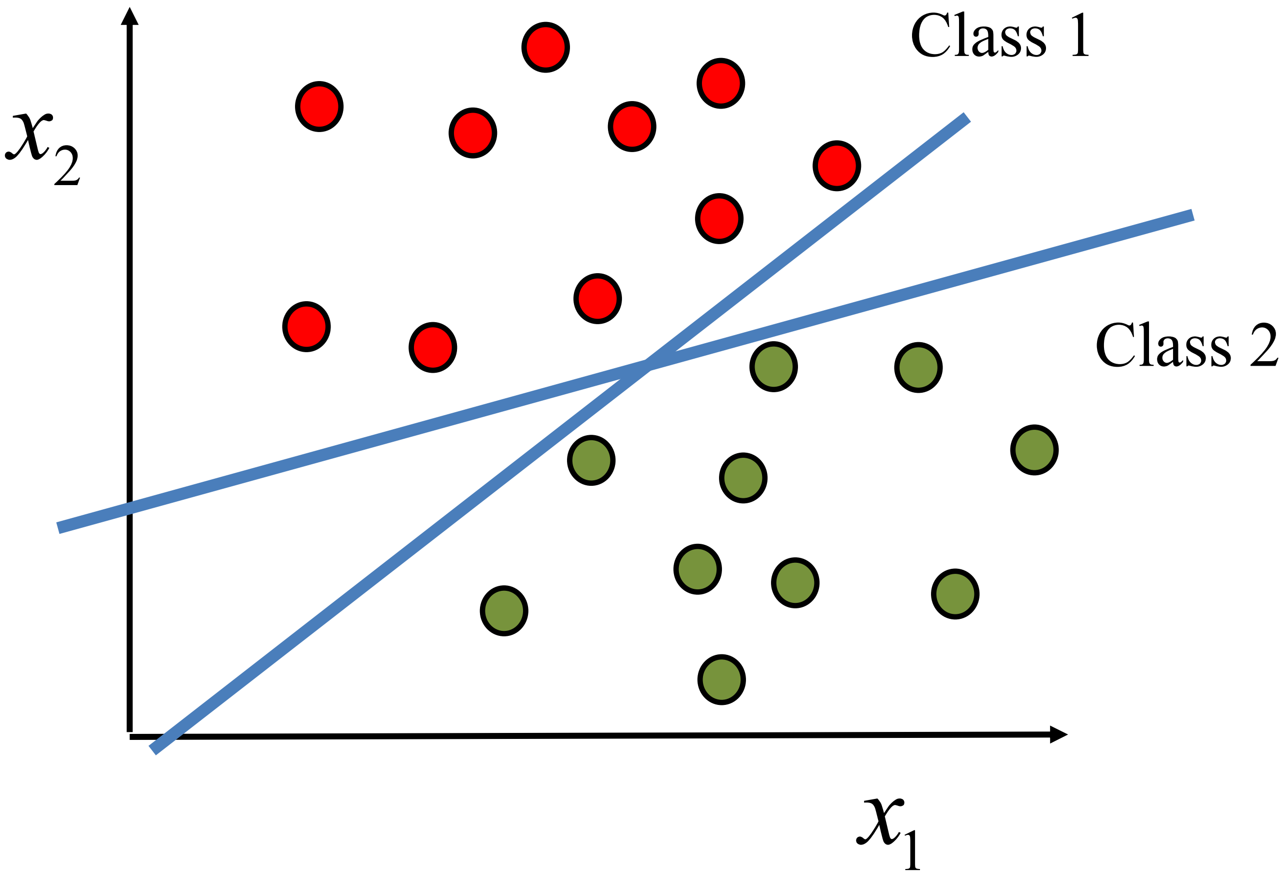
$ax_1 + bx_2 \leq c$

Others: Class 2

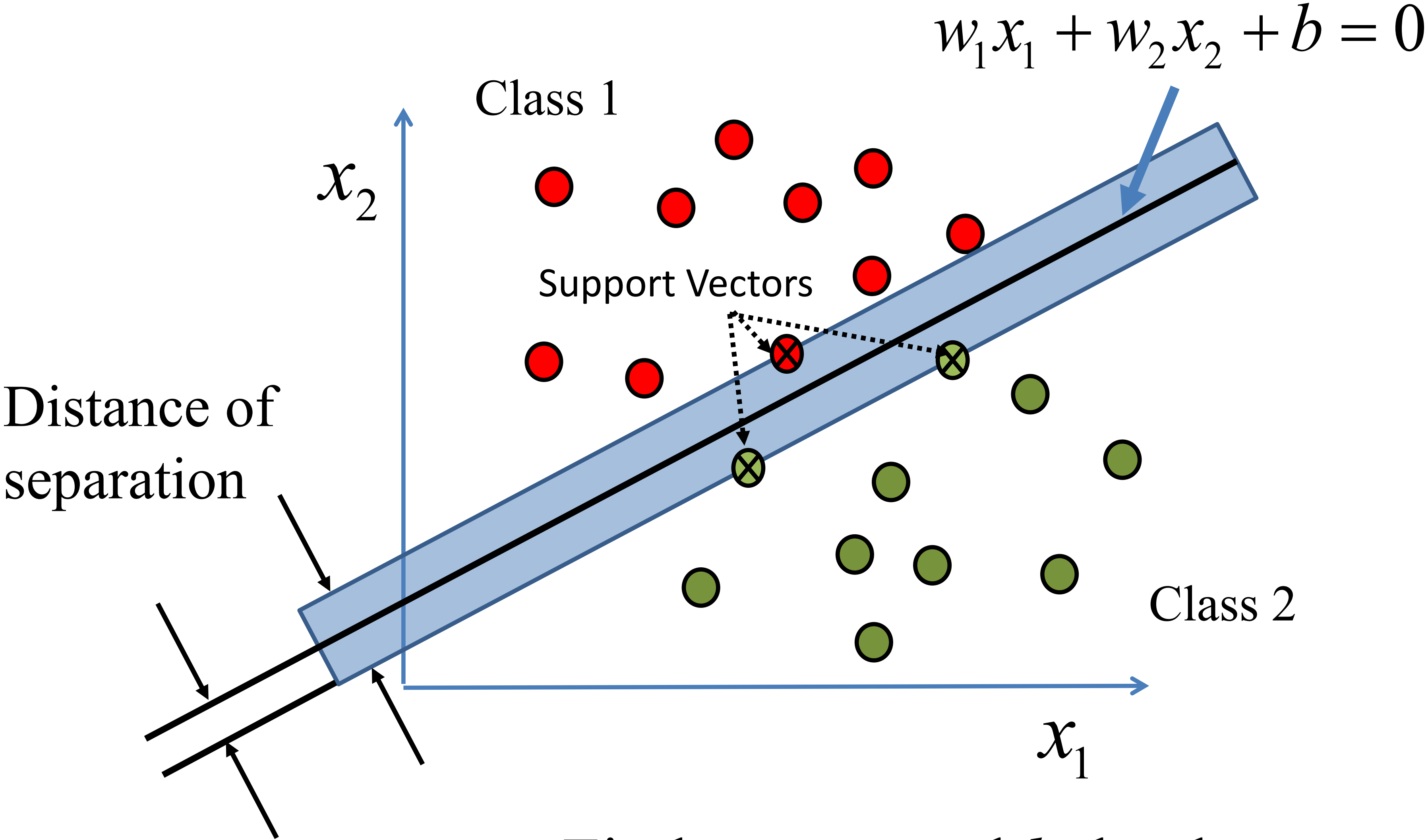
Feature x_1

Find *a*, *b*, and *c* that best separate classes 1 and 2.

How can we best separate the two classes?



How can we best separate the two classes?



$w_1 x_1 + w_2 x_2 + b = 0$

Distance of separation

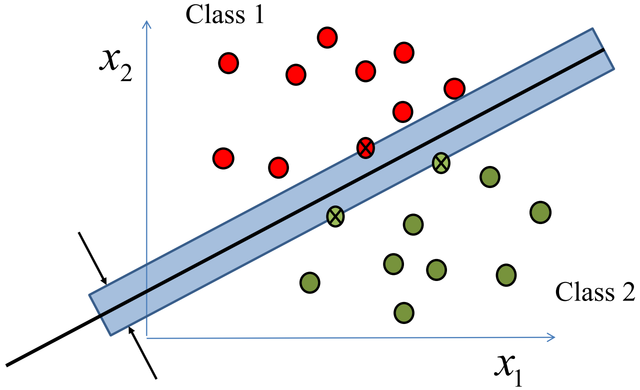
Margin

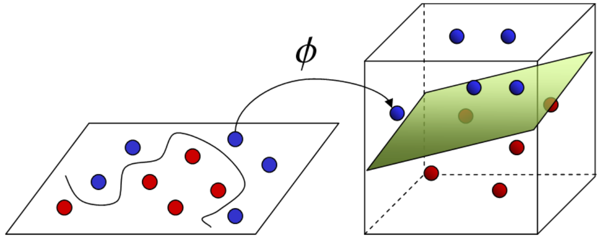
Find w_1, w_2 and b that best separate classes 1 and 2.

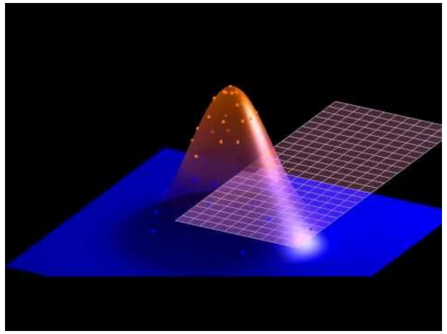
Outline

Support Vector Machines and Kernel Methods

- – SVM maximizes the margin in two-class classification
- – Mathematical derivation of the optimal classifier with the largest margin
- – Linear separability: revisite XOR
- – Augmentation of feature space with kernels
- – Kernel trick
- – Positive-definite, symmetric kernels
- – Reproducing kernel Hilbert space
- – Radial-basis functions







Support Vector Machine

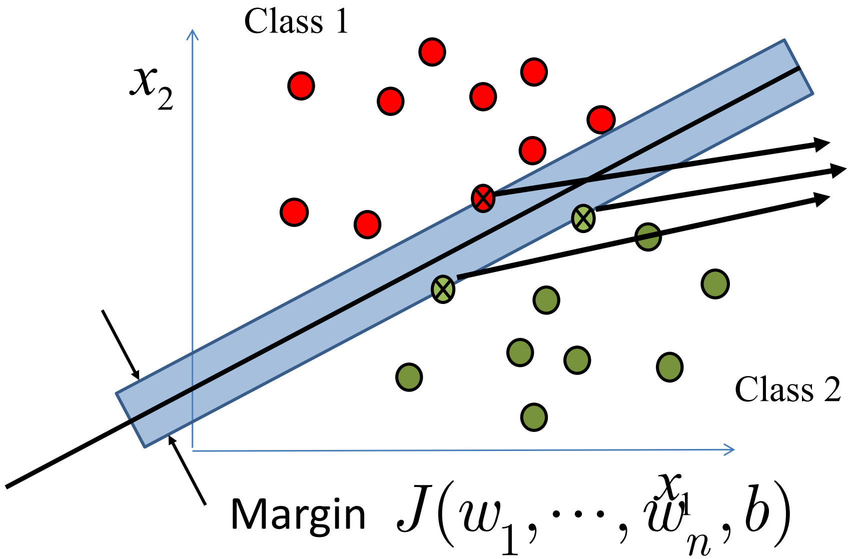
Find $w_1, \cdots, w_n$ and b that maximizes the margin $J(w_1, \cdots, w_n, b)$ for separating classes 1 and 2.

$$\max J(w_1, \cdots, w_n, b)$$

Subject to the correct classification:

All the data points in class 1 ; $w_1x_1 + w_2x_2 + \cdots + w_nx_n + b > 0$

All the data points in class 2 ; $w_1x_1 + w_2x_2 + \cdots + w_nx_n + b \leq 0$



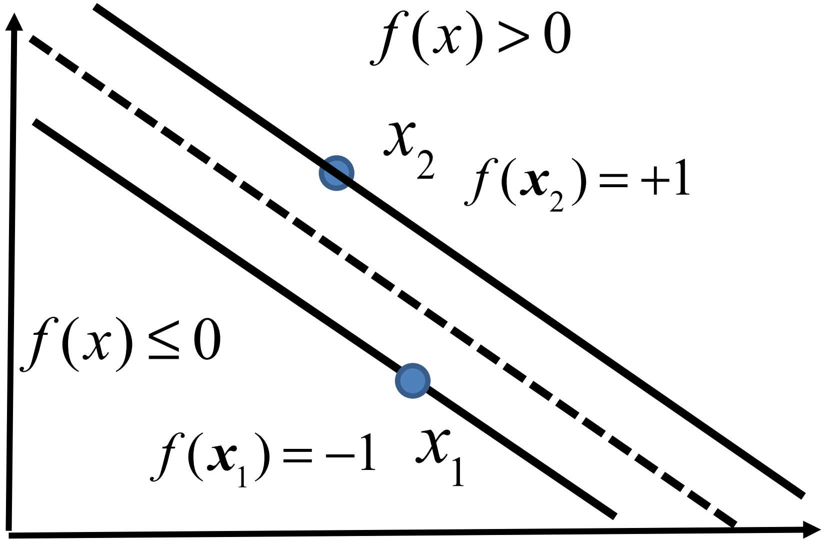
A limited number of points on the borders are called Support Vectors.

SVM Theoretical Derivation

Discriminant Function $f(x) = w^T x + b$

(Classification function)

$\boldsymbol{x} \in \Re^n \quad w \in \Re^{n\times 1} \quad b \in \Re$



- ❑ We care only the sign of the discriminant function.
- ❑ We can scale the function so that it takes the value of 1 at a support vector on positive side, and -1 at a support vector on the negative side.

Correct classification: $f(x) > 0 \rightarrow y = +1 \quad f(x) \leq 0 \rightarrow y = -1$

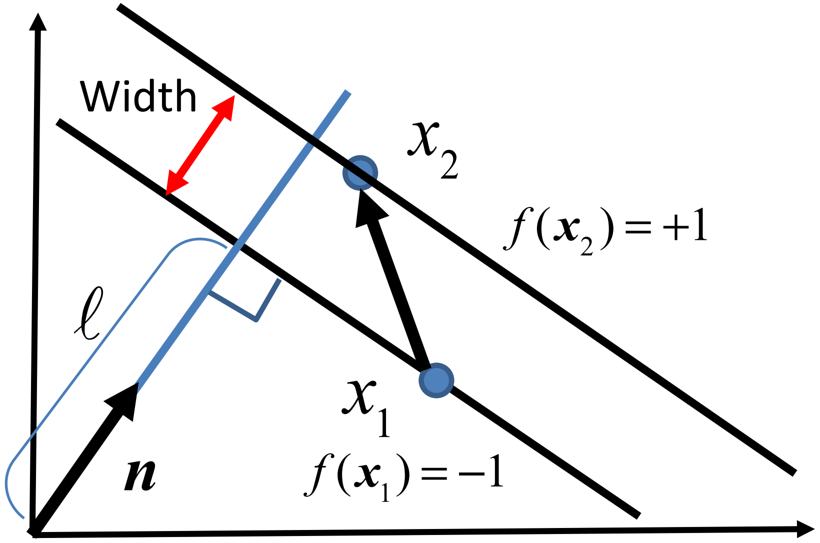
These two can be combines and replaced by

$$y_i(w^T x_i + b) \geq 1, \quad i = 1, \cdots, N$$

This applies to all the sample points. $D = \left\{(x_i, y_i) \mid i = 1, \cdots, N\right\}$

SVM Theoretical Derivation

Discriminant Function $f(x) = w^T x + b$ $\quad w \in \Re^{n\times 1} \quad b \in \Re$



The width of the separation, or the Margin of Classifier, is given by using a unit vector ***n***:

$$Width = \boldsymbol{n}^T (x_2 - x_1)$$

Since x_1 satisfies $w^T x_1 + b = -1$

and x_2 satisfies $w^T x_2 + b = 1$

$$2 = w^T (x_2 - x_1)$$

Dividing both sides by |*w*| yields: →

$$\frac{2}{|w|} = \frac{w^T}{|w|}(x_2 - x_1) = \boldsymbol{n}^T (x_2 - x_1) = Width$$

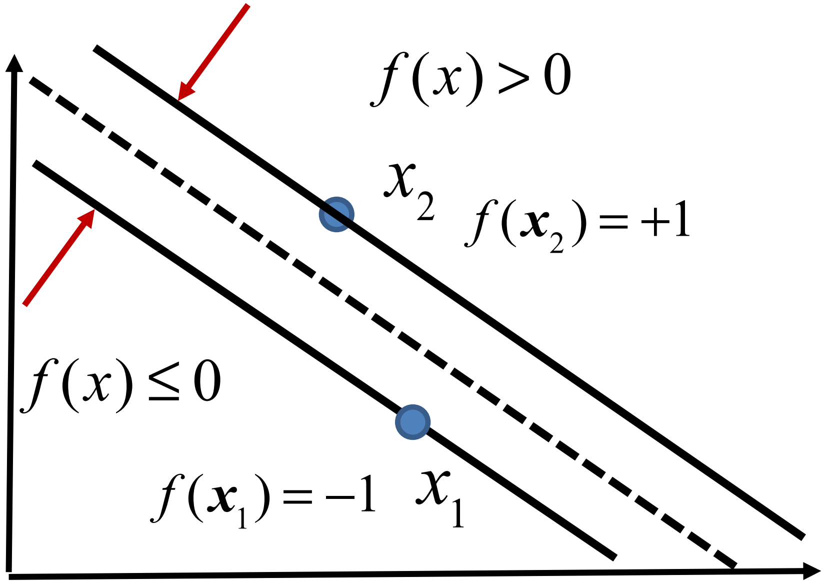
Unit vector $\boldsymbol{n} = \frac{\boldsymbol{w}}{|\boldsymbol{w}|}$

$$\Rightarrow \quad Width = \frac{2}{|w|}$$

SVM Theoretical Derivation

Maximize $Width = \frac{2}{\left| w \right|}$

Maximize $Width = \frac{2}{\left| w \right|^2}$



Maximizing this is equivalent to minimizing its reciprocal:

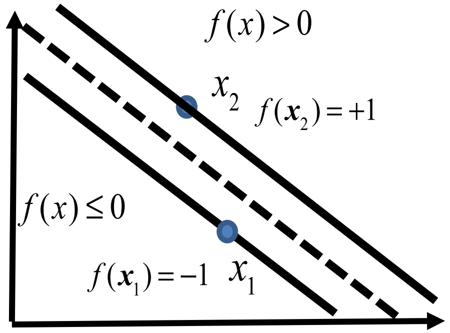
$$\min_{w,b} \frac{1}{2} w^T w$$

Subject to $y_i(w^T x_i + b) \ge 1, \quad i = 1, \cdots, N$

- To eliminate any discriminant function that is unable to correctly classify all the sample points, consider the following penalty function:

$$\max_{0\le\alpha_i\le M} \alpha_i\left[1-y_i(w^Tx_i+b)\right] \qquad M \gg 1$$

- If $y_i(w^Tx_i+b)=1$, then α_i is arbitrary;
- If $y_i(w^Tx_i+b)>1$, then $\alpha_i = 0$; and
- If $y_i(w^Tx_i+b)<1$, then $\alpha_i = M \gg 1$.



- Therefore, the above penalty function becomes 0 if all the sample points are correctly classified, and it takes a large value only when some sample points are incorrectly classified.

- Consider the minimization of the following functional

$$L(w,b,\alpha_1\cdots\alpha_N)=\frac{1}{2}w^Tw+\sum_{i=1}^{N}\max_{0\le\alpha_i\le M}\alpha_i\left(1-y_i(w^Tx_i+b)\right)$$

- Note that any discriminant function that cannot correctly classify some sample points is eliminated in the minimization of the functional.

Minimizing: $$L(w,b,\alpha_1\cdots\alpha_N)=\frac{1}{2}w^Tw+\sum_{i=1}^{N}\max_{0\le\alpha_i\le M}\alpha_i\left(1-y_i(w^Tx_i+b)\right)$$

Swapping min and max yields,

$$\min_{w,b,\alpha_1\cdots\alpha_N} L=\min_{w,b}\left[\frac{1}{2}w^Tw+\sum_{i=1}^{N}\max_{0\le\alpha_i\le M}\alpha_i\left(1-y_i(w^Tx_i+b)\right)\right]$$

$$=\max_{0\le\alpha_i\le M}\left\{\min_{w,b}\underbrace{\left[\frac{1}{2}w^Tw+\sum_{i=1}^{N}\alpha_i\left(1-y_i(w^Tx_i+b)\right)\right]}_{L^*}\Bigg]\right\}$$

The necessary conditions for L* to be min.

$$\frac{\partial L^*}{\partial w}=0:\quad w=\sum_{i=1}^{N}\alpha_iy_ix_i \qquad\qquad \frac{\partial L^*}{\partial b}=0:\quad \sum_{i=1}^{N}\alpha_iy_i=0$$

Therefore,

$$\min L=\max_{0\le\alpha_1\cdots\alpha_N\le M}\left\{\frac{1}{2}\left(\sum_{j=1}^{N}\alpha_jy_jx_j^T\right)\left(\sum_{k=1}^{N}\alpha_ky_kx_k\right)+\sum_{i=1}^{N}\alpha_i\left[1-y_i\left(\left(\sum_{j=1}^{N}\alpha_jy_jx_j^T\right)x_i+b\right)\right]\right\}$$

Substituting $\sum_{i=1}^{N}\alpha_iy_i=0$ yields $$\min L=\max_{0\le\alpha_1\cdots\alpha_N\le M}\left[\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_jy_iy_jx_j^Tx_i\right]$$

$$\min L = \max_{0 \le \alpha_1 \cdots \alpha_N \le M} \left[\sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_j^T x_i \right] \quad \text{Subject to} \quad \sum_{i=1}^{N} \alpha_i y_i = 0$$

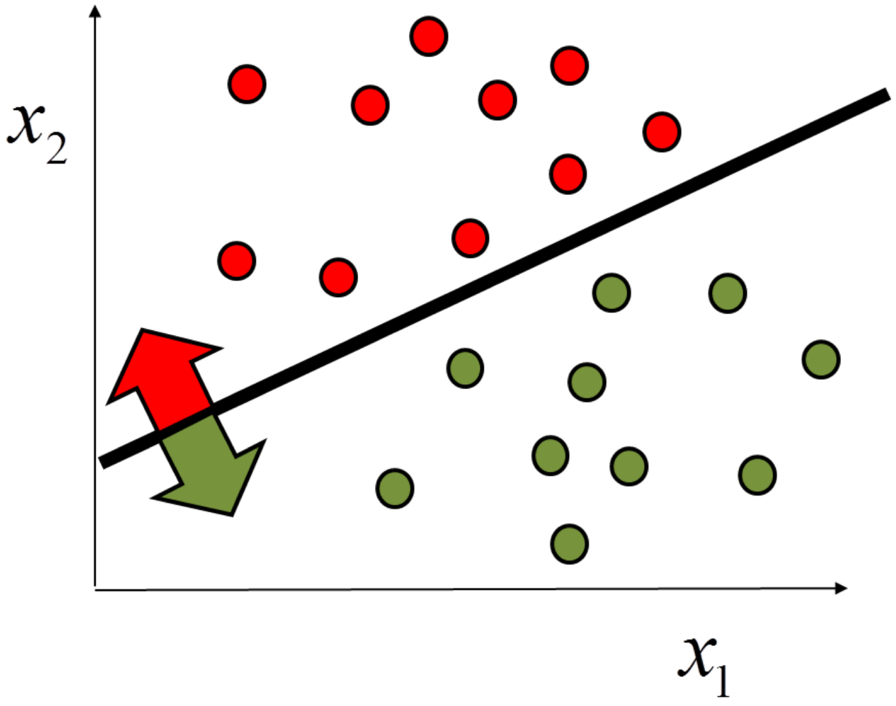
- This cost function is quadratic in terms of α_i , and the constraint is a linear equation.
- Therefore, this is a Quadratic Programming problem for which effective solution methods exist.
- Substituting the optimal weights $w = \sum_{i=1}^{N} \alpha_i y_i x_i$ into the discriminant function:

$$f(x) = w^T x + b \xrightarrow{\quad\quad\quad\quad} f(x) = \sum_{i=1}^{N} \alpha_i y_i (x_i^T x) + b$$

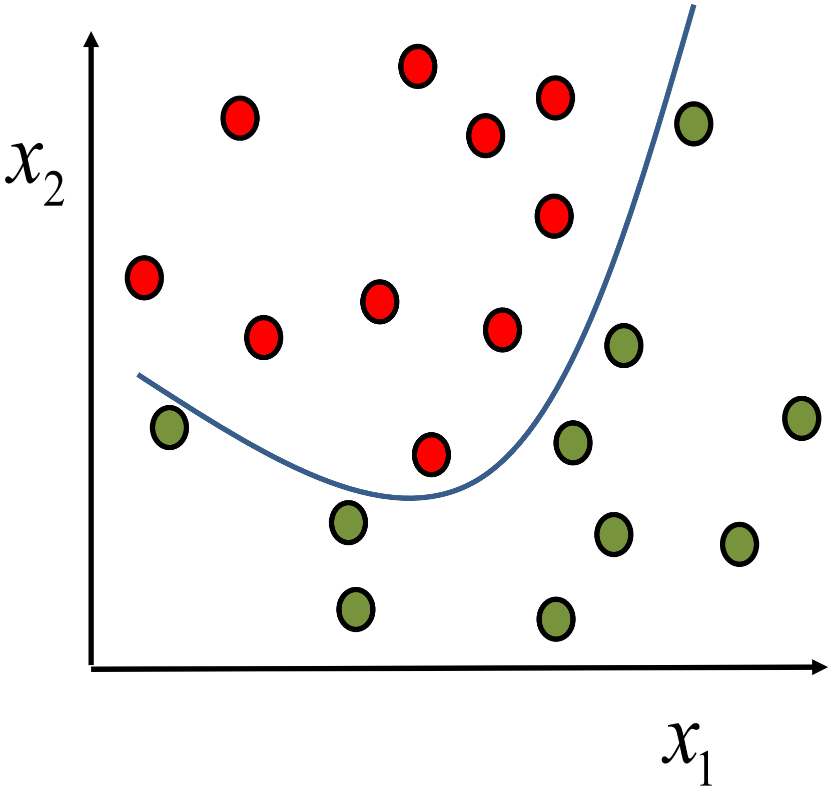
- Since support vectors lie on the marginal hyperplanes, for any support vector x_i, $w^T x_i + b = y_i$ and thus b can be obtained as:

$$b = y_i - \sum_{j=1}^{N} \alpha_j y_j (x_j{}^T x_i)$$

Linear Separable Case



Linearly separable.

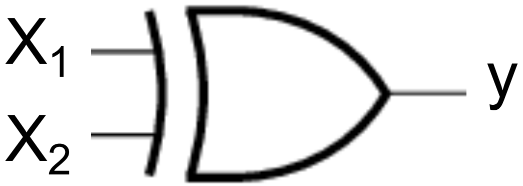


Not linearly separable.

Revisiting the XOR problem: An example of not linearly separable problem

The Exclusive OR Problem

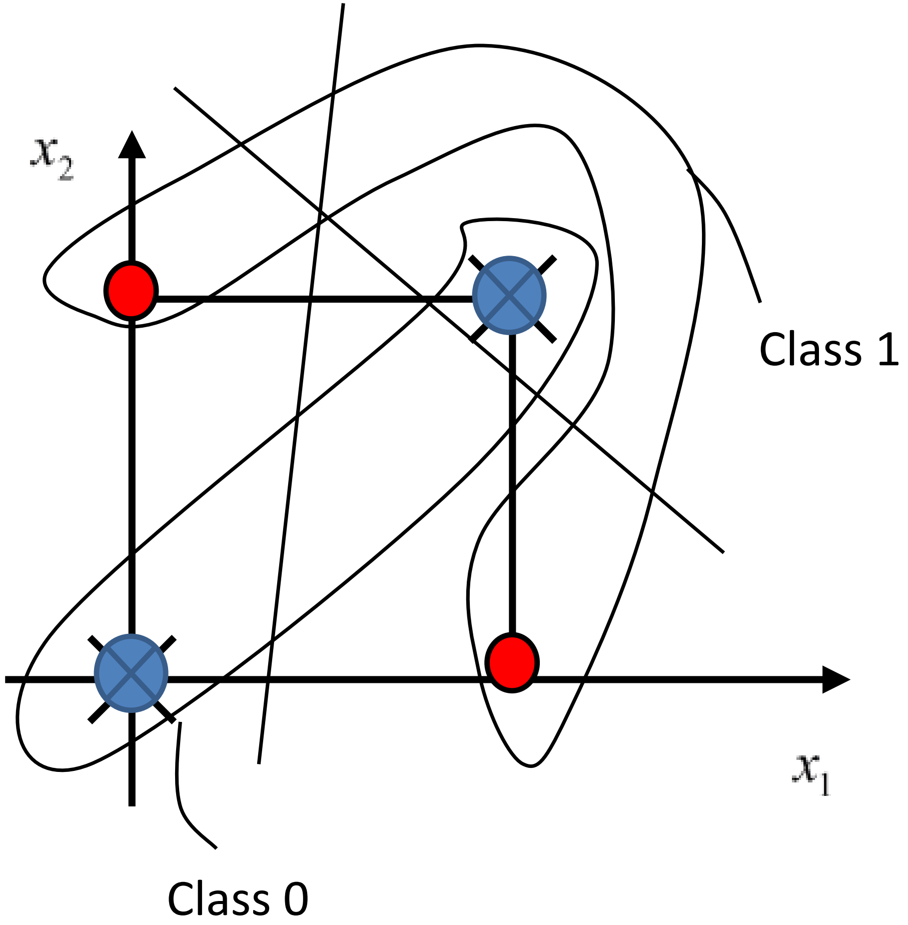
Input		Output
0	0	0
0	1	1
1	0	1
1	1	0
X_1	X_2	y



An example of not linearly separable problem

Discriminant function:

$$f(x_1, x_2) = w_1 x_1 + w_2 x_2 + b$$



The Exclusive OR Problem

Input		Output
0	0	0
0	1	1
1	0	1
1	1	0
X_1	X_2	y

No matter how you pick w_1, w_2 and b, these points cannot be separated into the two classes.

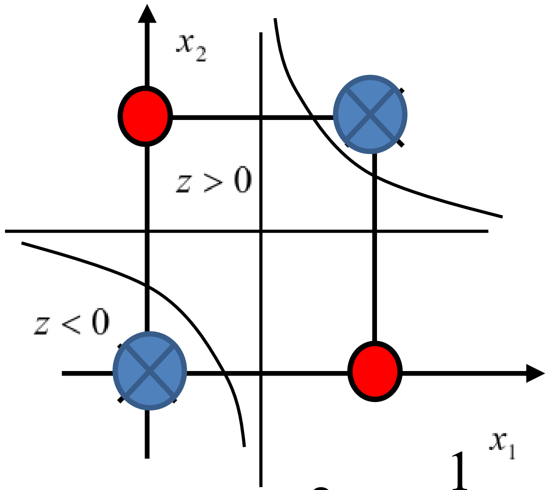
How to make it separable?

Add other variables.

Consider a third variable: $x_3 \triangleq x_1 x_2$ $\quad z = f(x_1, x_2) = x_1 + x_2 - 2x_1 x_2 - \frac{1}{3}$

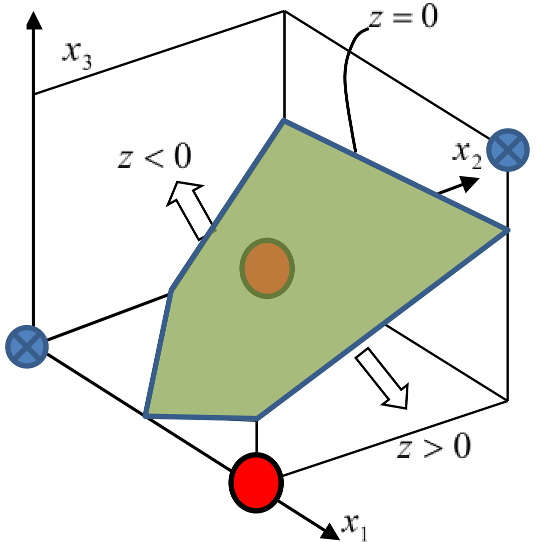
$$\left.\begin{array}{l} f(0,0) = -\frac{1}{3} \\ \\ f(1,1) = -\frac{1}{3} \end{array}\right\} \longrightarrow \text{Class 0}$$

$$f(1,0) = f(0,1) = \frac{2}{3} > 0 \longrightarrow \text{Class 1}$$



replace $x_1\ x_2$ by a new variable x_3 $\qquad z = x_1 + x_2 - 2x_3 - \frac{1}{3}$

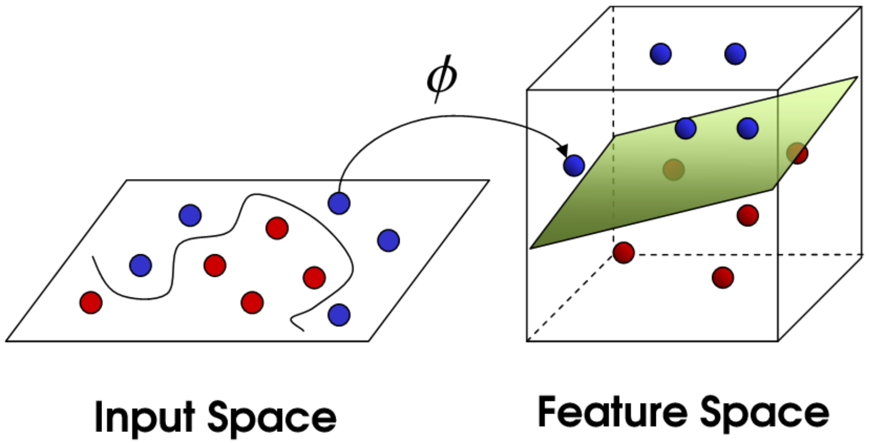
This is apparently a linear function: Linearly Separable.

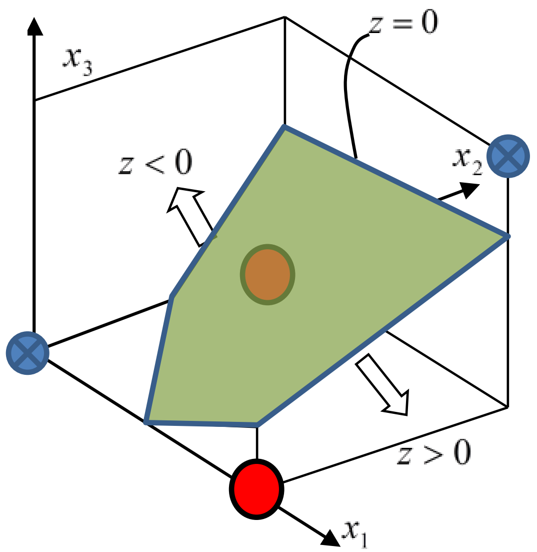


Recasting input data into a high-dimensional space

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \Longrightarrow \varphi(\boldsymbol{x}) = \begin{pmatrix} x_1 \\ x_2 \\ x_1 x_2 \end{pmatrix}$$

The algorithm of Support Vector Machine can be extended to the transformed feature space.





How can we find such a new variable in a systematic manner?

Re-formulating the SVM optimization problem

Discriminant function: $f(\varphi(\boldsymbol{x})) = \boldsymbol{w}^T\varphi(\boldsymbol{x}) + b$

where $\varphi(\boldsymbol{x}) \in \Re^{m\times 1}, \boldsymbol{w} \in \Re^{m\times 1}$

Classification:
$$\begin{aligned} f(\varphi(\boldsymbol{x})) > 0 \rightarrow y = +1 \\ f(\varphi(\boldsymbol{x})) < 0 \rightarrow y = -1 \end{aligned} \Longrightarrow y_i \cdot f(\varphi(\boldsymbol{x}_i)) \geq 1$$

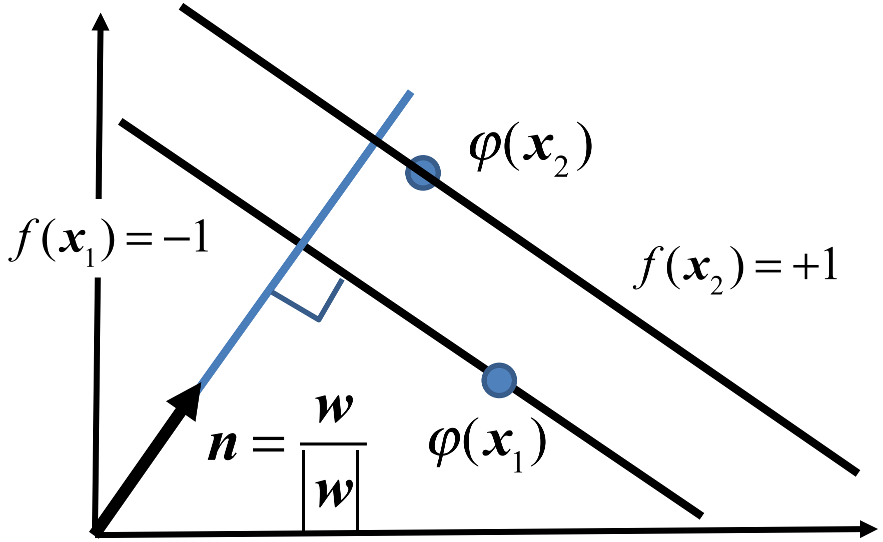
Training Data: $D = \{(\varphi(\boldsymbol{x}_i), y_i) \mid i = 1, \cdots, N\}$

$$Width = \frac{w^{\mathsf{T}}}{|w|}(\varphi(x_2) - \varphi(x_1))$$

$$\max \ Width = \frac{2}{|w|}$$

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

Subject to $y_i \cdot f(\varphi(\boldsymbol{x}_i)) \geq 1$


$\varphi(\boldsymbol{x}_2)$
$f(\boldsymbol{x}_1) = -1$
$f(\boldsymbol{x}_2) = +1$
$\boldsymbol{n} = \frac{\boldsymbol{w}}{|\boldsymbol{w}|}$
$\varphi(\boldsymbol{x}_1)$

If the data in the feature space are linearly separable, then the optimal discriminant function that maximizes the separation margin is given by

$$w = \sum_{i=1}^{N} \alpha_i y_i \varphi(\boldsymbol{x}_i)$$

where α_i are the variables that maximizes the following quadratic function:

$$\min L = \max_{0 \le \alpha_1 \cdots \alpha_N \le M} \left[\sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \varphi(x_j)^T \varphi(x_i) \right]$$

Subject to

$$\sum_{i=1}^{N} \alpha_i y_i = 0$$

This is the same simple quadratic optimization problem.

Kernel Trick

- ❑ Using the optimal weights, $w = \sum_{i=1}^{N} \alpha_i y_i \varphi(\boldsymbol{x}_i)$, the optimal discriminant function, or the Support Vector Classifier, is given by

$$f(\varphi(\boldsymbol{x})) = \boldsymbol{w}^T \varphi(\boldsymbol{x}) + b = \left[\sum_{i=1}^{N} \alpha_i y_i \varphi(\boldsymbol{x}_i)\right]^T \varphi(\boldsymbol{x}) + b = \sum_{i=1}^{N} \alpha_i y_i \underbrace{[(\varphi(\boldsymbol{x}_i))^T \varphi(\boldsymbol{x})]}_{k(\boldsymbol{x}_i,\boldsymbol{x})} + b$$

- ❑ $k(\boldsymbol{x}_i,\boldsymbol{x}) = (\varphi(\boldsymbol{x}_i))^T \varphi(\boldsymbol{x})$ is a scalar quantity, called a ***kernel*** function:

$$k : X \times X \mapsto \Re, \quad \boldsymbol{x} \in X, \boldsymbol{x}_i \in X$$

- ❑ It is the inner product of two feature vectors in a high dimensional space $\varphi(\boldsymbol{x})$.
- ❑ This can be computed without obtaining the high-dimensional feature vector.
- ❑ In the quadratic optimization, too, the high dimensional feature vector $\varphi(\boldsymbol{x})$ shows up as an inner product. Therefore, we deal with only its kernel $k(\boldsymbol{x}_i,\boldsymbol{x}_j) = (\varphi(\boldsymbol{x}_i))^T \varphi(\boldsymbol{x}_j)$

Fast Computation

$$\min L = \max_{0 \le \alpha_1 \cdots \alpha_N \le M} \left[\sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \varphi(x_j)^T \varphi(x_i)\right]$$

Example of Kernel Trick

Polynomial Kernels

- ❑ Suppose that the original feature space is two-dimensional: $\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$

- ❑ Consider the following kernel function: a function of the inner product of vector ***x*** and ***x***’

$$K(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x}^T \boldsymbol{x}' + c)^2$$

- ❑ This can be extended to

$$K(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x}^T \boldsymbol{x}' + c)^2 = x_1^2 x'^2_1 + x_2^2 x'^2_2 + c^2 + 2x_1 x'_1 x_2 x'_2 + 2cx_1 x'_1 + 2cx_2 x'_2$$

$$= \begin{pmatrix} x_1^2 & x_2^2 & \sqrt{2}x_1 x_2 & \sqrt{2c}x_1 & \sqrt{2c}x_2 & c \end{pmatrix} \begin{pmatrix} x'^2_1 \\ x'^2_2 \\ \sqrt{2}x'_1 x'_2 \\ \sqrt{2c}x'_1 \\ \sqrt{2c}x'_2 \\ c \end{pmatrix} = (\varphi(\boldsymbol{x}))^T \varphi(\boldsymbol{x}')$$

Example of Kernel Trick

Polynomial Kernels (Continued)

- □ This implies that the feature vector space has been expanded from the original 2-dimensional space to a new 6-dimensional space:

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \Rightarrow \varphi(\boldsymbol{x}) = \begin{pmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}x_1x_2 \\ \sqrt{2c}x_1 \\ \sqrt{2c}x_2 \\ c \end{pmatrix}$$

$$K(\boldsymbol{x},\boldsymbol{x}') = (\boldsymbol{x}^T\boldsymbol{x}'+c)^2 = \begin{pmatrix} x_1^2 & x_2^2 & \sqrt{2}x_1x_2 & \sqrt{2c}x_1 & \sqrt{2c}x_2 & c \end{pmatrix} \begin{pmatrix} x'^2_1 \\ x'^2_2 \\ \sqrt{2}x'_1x'_2 \\ \sqrt{2c}x'_1 \\ \sqrt{2c}x'_2 \\ c \end{pmatrix} = (\varphi(\boldsymbol{x}))^T\varphi(\boldsymbol{x}')$$

Example of Kernel Trick

Polynomial Kernels (Continued)

This implies that the feature vector space has been expanded from the original 2-dimensional space to a new 6-dimensional space:

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \Longrightarrow \varphi(\boldsymbol{x}) = \begin{pmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}x_1x_2 \\ \sqrt{2c}x_1 \\ \sqrt{2c}x_2 \\ c \end{pmatrix}$$

❑ The discriminant function (Support Vector Classifier) becomes:

$$f(\varphi(\boldsymbol{x})) = \boldsymbol{w}^T\varphi(\boldsymbol{x}) + b = \sum_{i=1}^{N} \alpha_i y_i \underbrace{[(\varphi(\boldsymbol{x}_i))^T \varphi(\boldsymbol{x})]}_{K(\boldsymbol{x}_i, \boldsymbol{x})} + b = \sum_{i=1}^{N} \alpha_i y_i (\boldsymbol{x}_i^T \boldsymbol{x} + c)^2 + b$$

❑ Note: the function can be evaluated directly using the low-dimensional input vectors without actually transforming the data to the higher dimensional space.

This is called the ***Kernel Trick***.

❑ The Kernel Trick allows us to

- Significantly reduce the computational load, and
- Find the way of reducing a not-linearly separable problem to a linearly separable problem in a higher-dimensional space without actually transforming it to the new space.

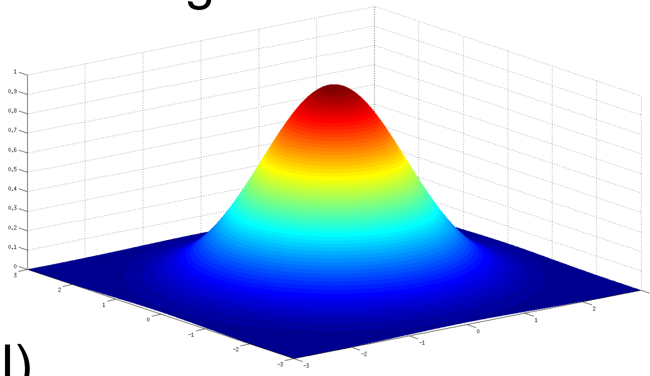
Kernel Functions

- ❑ In supervised learning, quantifying similarity between two data points is critically important. A kernel function quantifies this similarity or nearness of data, encoding features in its functional structure.
- ❑ There are numerous kernel functions being used for machine learning:
  - ▪ Dot-product (polynomial) kernel

$$k(\boldsymbol{x},\boldsymbol{x}') = (\boldsymbol{x}\cdot\boldsymbol{x}'+1)^p$$

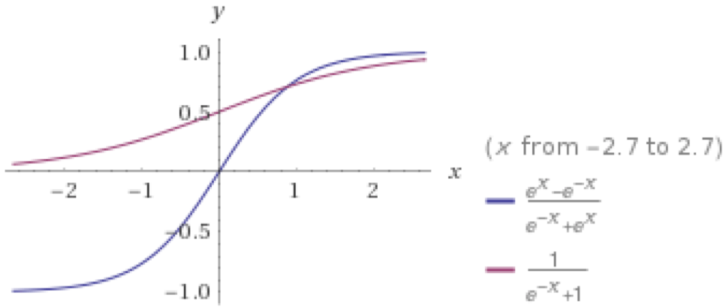
  - ▪ Radial-Basis Function (Squared exponential kernel, Gaussian kernel)

$$k(\boldsymbol{x},\boldsymbol{x}') = \exp\left(-\frac{|\boldsymbol{x}-\boldsymbol{x}'|^2}{2\ell^2}\right)$$



  - ▪ Sigmoid kernel

$$k(\boldsymbol{x},\boldsymbol{x}') = \tanh(\kappa\boldsymbol{x}\cdot\boldsymbol{x}'-\delta)$$



- ❑ Questions:
  - ▪ How are these kernel functions related to high-dimensional feature spaces?
  - ▪ Does a nonlinear feature function $\varphi(\boldsymbol{x})$ exists, which generates a kernel function in the form of inner product between $\varphi(\boldsymbol{x})$ and $\varphi(\boldsymbol{x}')$?
  - ▪ Under which conditions does it exist?

Positive-Definite Symmetric Kernels

❑ Let us examine this special property of kernel functions.

❑ Definition of Kernel Matrix

- Given a set of vectors, $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m \in X \subset \Re^n$ and a kernel over X:

$$k : X \times X \mapsto \Re$$

- The following m-by-m matrix can be formed, which is called a Kernel Matrix.

$$K = \begin{pmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}_1) & \cdots & k(\boldsymbol{x}_1, \boldsymbol{x}_m) \\ \vdots & \ddots & \vdots \\ k(\boldsymbol{x}_m, \boldsymbol{x}_1) & \cdots & k(\boldsymbol{x}_m, \boldsymbol{x}_m) \end{pmatrix} \in \Re^{m \times m}$$

: a Gram matrix

❑ **Definition of Positive-Definite Symmetric Kernel**

- A kernel over X, $k : X \times X \mapsto \Re$, is said to be positive-definite and symmetric, if for an arbitrary sample $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m\}$ from X, the associated Kernel Matrix is positive semi-definite and symmetric.

$$c^T K c \geq 0, \quad \forall c \in \Re^m$$

Theorem: Existence of Feature Map $\varphi(x)$ - Reproducing Kernel Hilbert Space (RKHS) -

❑ Let $k : X \times X \mapsto \Re$ be a positive-definite and symmetric kernel. Then, there exists a Hilbert space H and a mapping $\varphi : X \mapsto H$ such that

$$k(\boldsymbol{x},\boldsymbol{x}') = <\varphi(\boldsymbol{x}),\varphi(\boldsymbol{x}')> \qquad \forall \boldsymbol{x} \in X, \forall \boldsymbol{x}' \in X$$

- Hilbert Space: Don’t get scared by this math term. In 2.160, we mean by Hilbert space an infinite dimensional vector space where inner product is defined, and any Cauchy sequence converges within its own space: completeness.
- A function $f(x)$ can be treated as an infinite dimensional vector. Therefore, a function may be an element in a Hilbert space.

❑ For a positive-definite, symmetric kernel $k : X \times X \mapsto \Re$, the associated Hilbert space H has the following property, called the reproducing property:

$$h(\boldsymbol{x}) = < h, k(\boldsymbol{x},\cdot) >, \quad \forall h \in H, \forall \boldsymbol{x} \in X$$

- Note that h is a function and an element in H. The inner product is an integral:

$$< h, k(\boldsymbol{x},\cdot) > = \int_X h(z) k(\boldsymbol{x},z) dz$$

Example: Radial-Basis Function / Gaussian Kernel / Squared Exponential Kernel

$$k(\boldsymbol{x},\boldsymbol{x}')=\exp\left(-\frac{|\boldsymbol{x}-\boldsymbol{x}'|^2}{2\ell^2}\right) \quad \rightarrow \quad k(\boldsymbol{x},\boldsymbol{x}')=<\varphi(\boldsymbol{x}),\varphi(\boldsymbol{x}')>$$

$$\exp\left(-\frac{|\boldsymbol{x}-\boldsymbol{x}'|^2}{2\ell^2}\right)=\exp\left(-\frac{|\boldsymbol{x}|^2}{2\ell^2}-\frac{|\boldsymbol{x}'|^2}{2\ell^2}+\frac{\boldsymbol{x}^T\boldsymbol{x}'}{\ell^2}\right)=\exp\left(-\frac{|\boldsymbol{x}|^2}{2\ell^2}\right)\exp\left(-\frac{|\boldsymbol{x}'|^2}{2\ell^2}\right)\exp\left(\frac{\boldsymbol{x}^T\boldsymbol{x}'}{\ell^2}\right)$$

$$\exp\left(\frac{\boldsymbol{x}^T\boldsymbol{x}'}{\ell^2}\right)=1+\frac{\boldsymbol{x}^T\boldsymbol{x}'}{\ell^2}+\frac{1}{2!}\frac{(\boldsymbol{x}^T\boldsymbol{x}')^2}{\ell^4}+\frac{1}{3!}\frac{(\boldsymbol{x}^T\boldsymbol{x}')^3}{\ell^6}+\cdots$$

Taylor series Expansion

In case $\boldsymbol{x}$ is a scalar x,

$$\varphi(x)=\exp\left(-\frac{x^2}{\ell^2}\right)\cdot\begin{pmatrix}1\\ x/\ell\\ x^2/\sqrt{2}\ell^2\\ x^3/\sqrt{6}\ell^3\\ \vdots\end{pmatrix}$$

We can confirm:

$$\exp\left(-\frac{x^2}{\ell^2}\right)\cdot\begin{pmatrix}1\\ x/\ell\\ x^2/\sqrt{2}\ell^2\\ x^3/\sqrt{6}\ell^3\\ \vdots\end{pmatrix}^T \exp\left(-\frac{x'^2}{\ell^2}\right)\cdot\begin{pmatrix}1\\ x'/\ell\\ x'^2/\sqrt{2}\ell^2\\ x'^3/\sqrt{6}\ell^3\\ \vdots\end{pmatrix}=k(x,x')$$

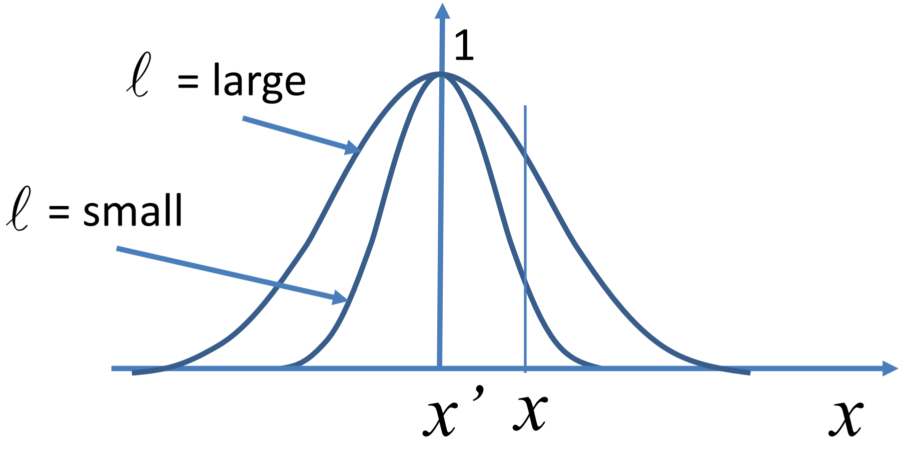
The associated feature space is ***infinite*** dimensional.

Radial-Basis Function / Gaussian Kernel / Squared Exponential Kernel

- ❑ Radial-Basis Function is the most widely used kernel function in machine learning, particularly for SVM and Gaussian Processes (Lecture 23).

$$k(\boldsymbol{x},\boldsymbol{x}')=\exp\left(-\frac{|\boldsymbol{x}-\boldsymbol{x}'|^2}{2\ell^2}\right)$$

Characteristic Length Scale



- ❑ Like other kernel functions, a Radial-Basis Function represents the similarity or nearness of two data points, $\boldsymbol{x}$ and $\boldsymbol{x}'$, with *characteristic length scale* l.
- ❑ For SVM, kernel functions are directly used in Support Vector Classifier:

$$f(\varphi(\boldsymbol{x}))=\sum_{i=1}^{N}\alpha_i y_i[(\varphi(\boldsymbol{x}_i))^T\varphi(\boldsymbol{x})]+b \rightarrow F(\boldsymbol{x})\triangleq\sum_{i=1}^{N}\alpha_i y_i k(\boldsymbol{x}_i,\boldsymbol{x})+b$$

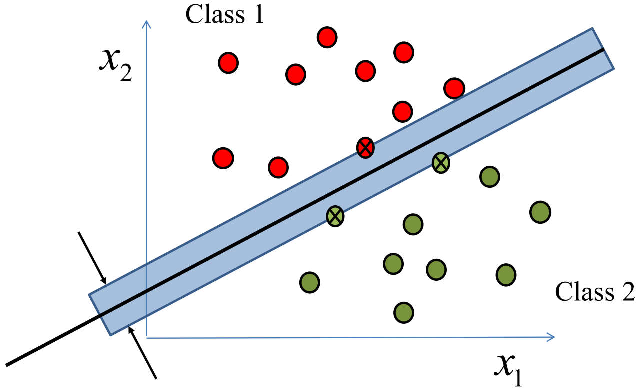
- ❑ Parameters are determined by solving the following quadratic optimization problem.

$$\alpha_1,\cdots,\alpha_N=\arg\max_{0\le\alpha_1\cdots\alpha_N\le M}\left[\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j k(x_j,x_i)\right]$$

Reflection

Support Vector Machine and Kernel Methods

- – SVM maximizes the margin in two-class classification
- – Mathematical derivation of the optimal classifier with the largest margin
- – Linear separability: revisited XOR
- – Augmentation of feature space with kernels
- – Kernel trick
- – Positive-definite, symmetric kernels
- – Reproducing kernel Hilbert space
- – Radial-basis functions


Class 1
x_2
Class 2
x_1