Practical Machine Learning

Lecture 5
Support Vector Machine (SVM) and Kernel trick

Dr. Suyong Eum

OSAKA UNIVERSITY

A question from the last class

- How do we infer p(Z|X) using q(Z)?
  - Only data X are given
  - p(X,Z) is known

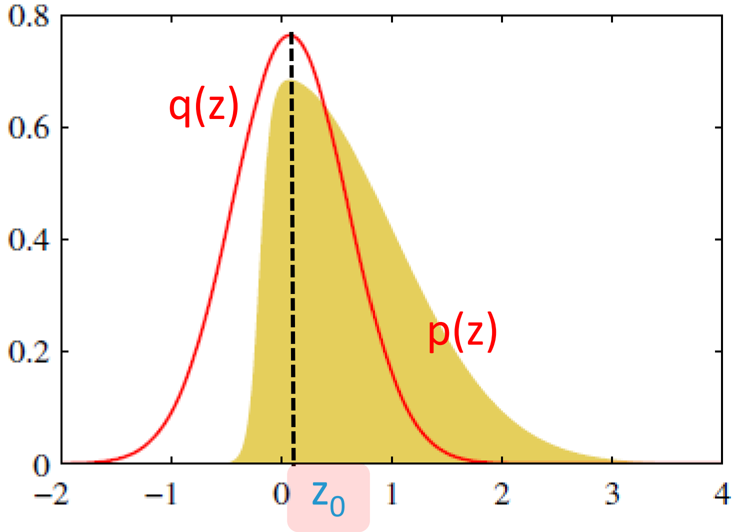
$$p(Z \mid X) = \frac{p(X,Z)}{p(X)} = \frac{p(X,Z)}{\boxed{\sum_Z p(X,Z)}}$$

- When the variable z is continuous
- When the number of variables z is many (?)

	Resource	Time	Accuracy
Laplace approach (Gaussian approximation)	Good	Good	Worse
Sampling (Numerical approach)	Worse	Worse	Good
Variational Inference (Analytical approach)	Medium	Medium	Medium

Laplace approach

- Finding the mode of the posterior distribution and then fitting a Gaussian centered at that mode.



Mode of p(z): local maxima

$$p(z) \propto \exp(-z^2/2)(1+\exp^{-20z-4})^{-1}$$

$$p(z) = \frac{1}{C} f(z) \qquad C = \int f(z)dz$$

Normalizing factor, which is unknown

$$\left.\frac{df(z)}{dz}\right|_{z=z_0} = 0$$

It becomes mean of q(z)

$$A = -\left.\frac{d^2}{dz^2} \ln f(z)\right|_{z=z_0}$$

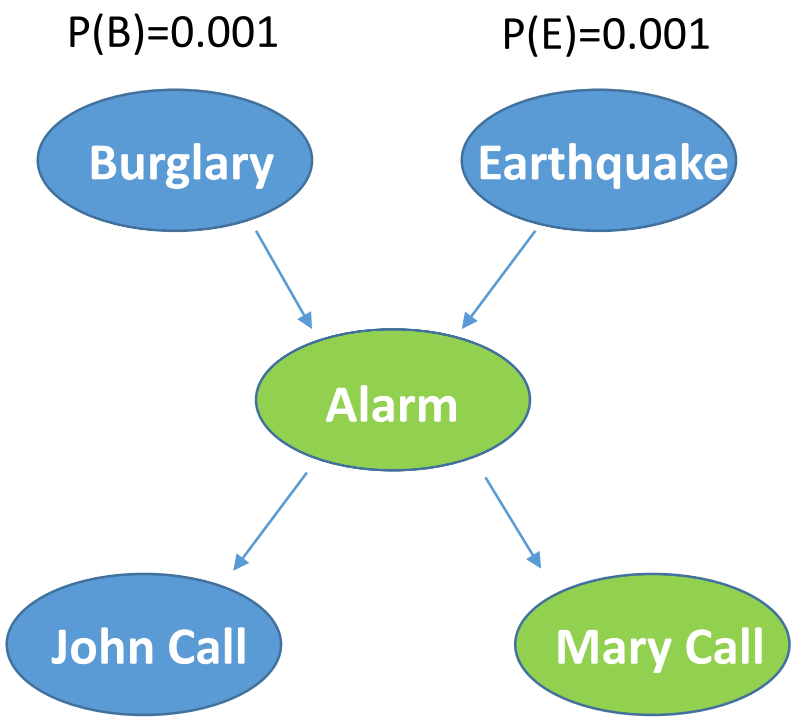
It becomes precision of q(z)

$$q(z) = \left(\frac{A}{2\pi}\right)^{1/2} \exp\left\{-\frac{A}{2}(z-z_0)^2\right\}$$

$$N(x|\mu,\sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$

$$\frac{d^2}{dx^2} \ln N(x|\mu,\sigma^2) = -\frac{1}{\sigma^2}$$

Sampling approach



A	P(J|A)
T	0.90
F	0.05

A	P(M|A)
T	0.70
F	0.01

B	E	P(A|B,E)
T	T	0.90
T	F	0.05
F	T	0.29
F	F	0.001

$$p(B,E,A,J,M) = p(B)\,p(E)\,p(A \mid B,E)\,p(J \mid A)\,p(M \mid A)$$

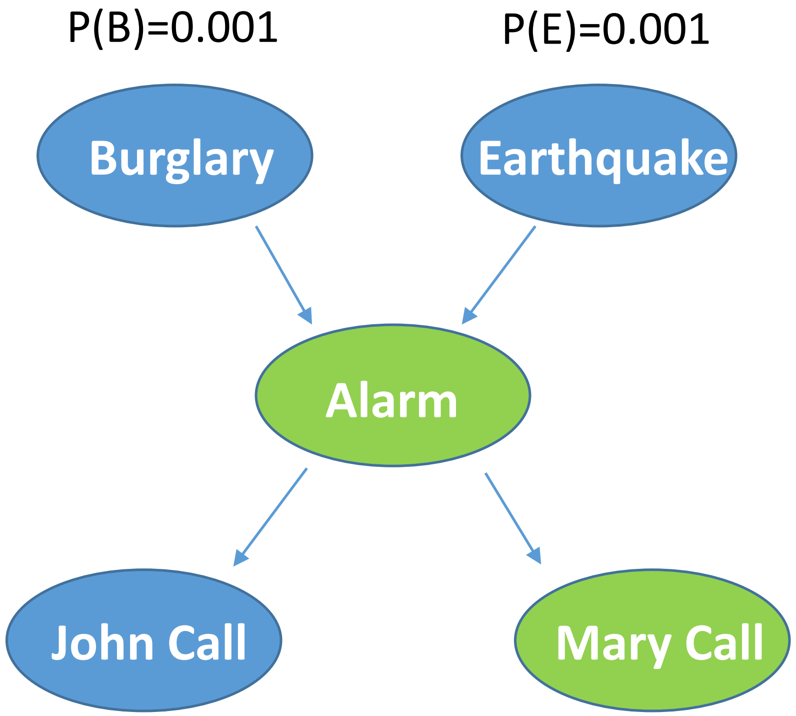
$$p(B,E,J \mid A,M)\ ?$$

$$p(E \mid B,A,J,M) = p(E \mid A,B)$$

$$p(B \mid E,A,J,M) = p(B \mid A,E)$$

$$p(J \mid B,E,A,M) = p(J \mid A)$$

Sampling approach



A	P(J|A)
T	0.90
F	0.05

A	P(M|A)
T	0.70
F	0.01

B	E	P(A|B,E)
T	T	0.90
T	F	0.05
F	T	0.29
F	F	0.001

$$p(B,E,A,J,M) = p(B)p(E)p(A|B,E)p(J\mid A)p(M\mid A)$$

$$\boxed{p(B,E,J\mid A,M)}\ ?$$

$$p(E\mid B,A,J,M) = \boxed{p(E\mid A,B)}$$

$$p(E,A,B) = p(A|B,E)p(B)p(E)$$

$$p(E,A,B) = p(E\mid A,B)p(A,B)$$

$$= p(E\mid A,B)p(A|B)p(B)$$

$$p(A|B,E)p(B)p(E) = p(E\mid A,B)p(A|B)p(B)$$

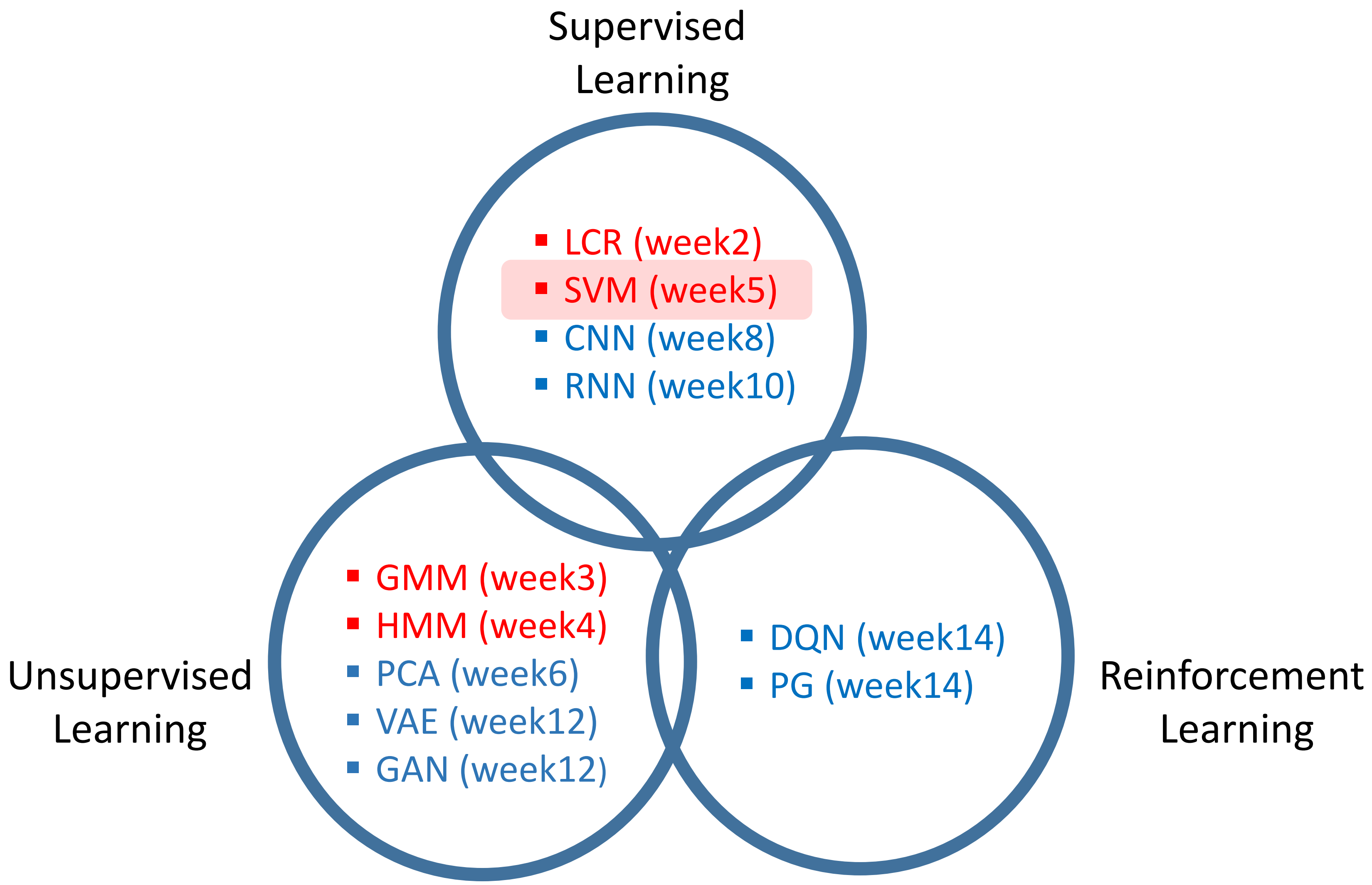
$$\boxed{p(E\mid A,B) = \frac{p(A|B,E)p(E)}{\sum_{E} p(A|B,E)}}$$

Variational inference

- The idea is
  - Finding p(Z|X) by minimizing Kullback divergence to q(Z)
  - Minimizing KL between p(Z|X) and q(Z) is equivalent to maximizing a function where the conditional distribution p(Z|X) is replaced with the joint distribution p(Z, X).
  - Factorizing the joint distribution on the assumption that the latent variables Z are independent.
  - Developing the derivation in terms of one latent variable on the assumption of the other latent variables are known.
  - Then, do some algebra..

- Refer the backup slides which include the derivation

Where we are


Supervised Learning
LCR (week2)
SVM (week5)
CNN (week8)
RNN (week10)
Unsupervised Learning
GMM (week3)
HMM (week4)
PCA (week6)
VAE (week12)
GAN (week12)
DQN (week14)
PG (week14)
Reinforcement Learning

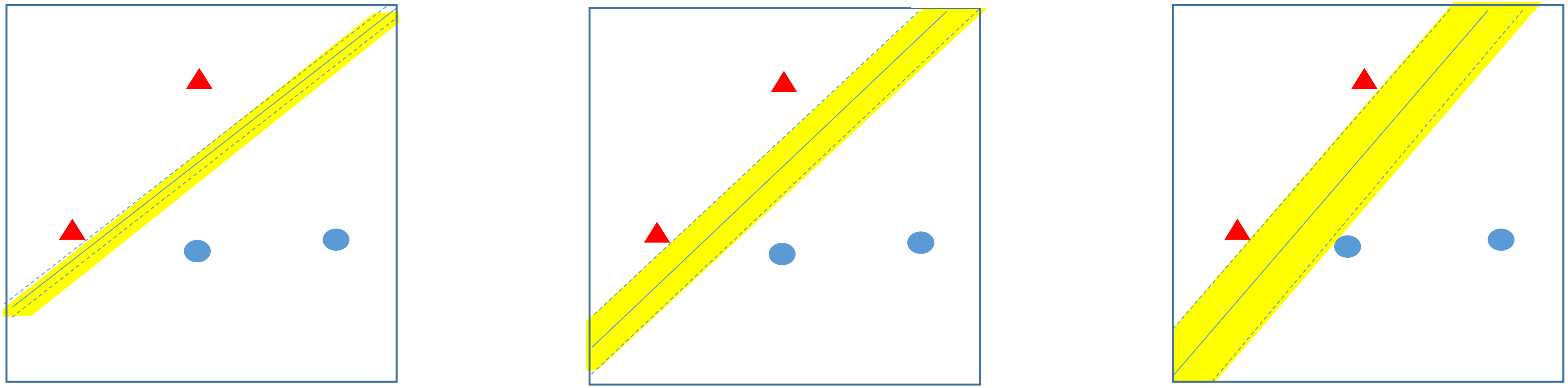
You are going to learn

- ❑ An idea of Support Vector Machine (SVM)
- ❑ Problem formulation of SVM
  - Linear classification: Hard Margin SVM
- ❑ Non-linear classification
  - Soft Margin SVM
  - Kernel trick

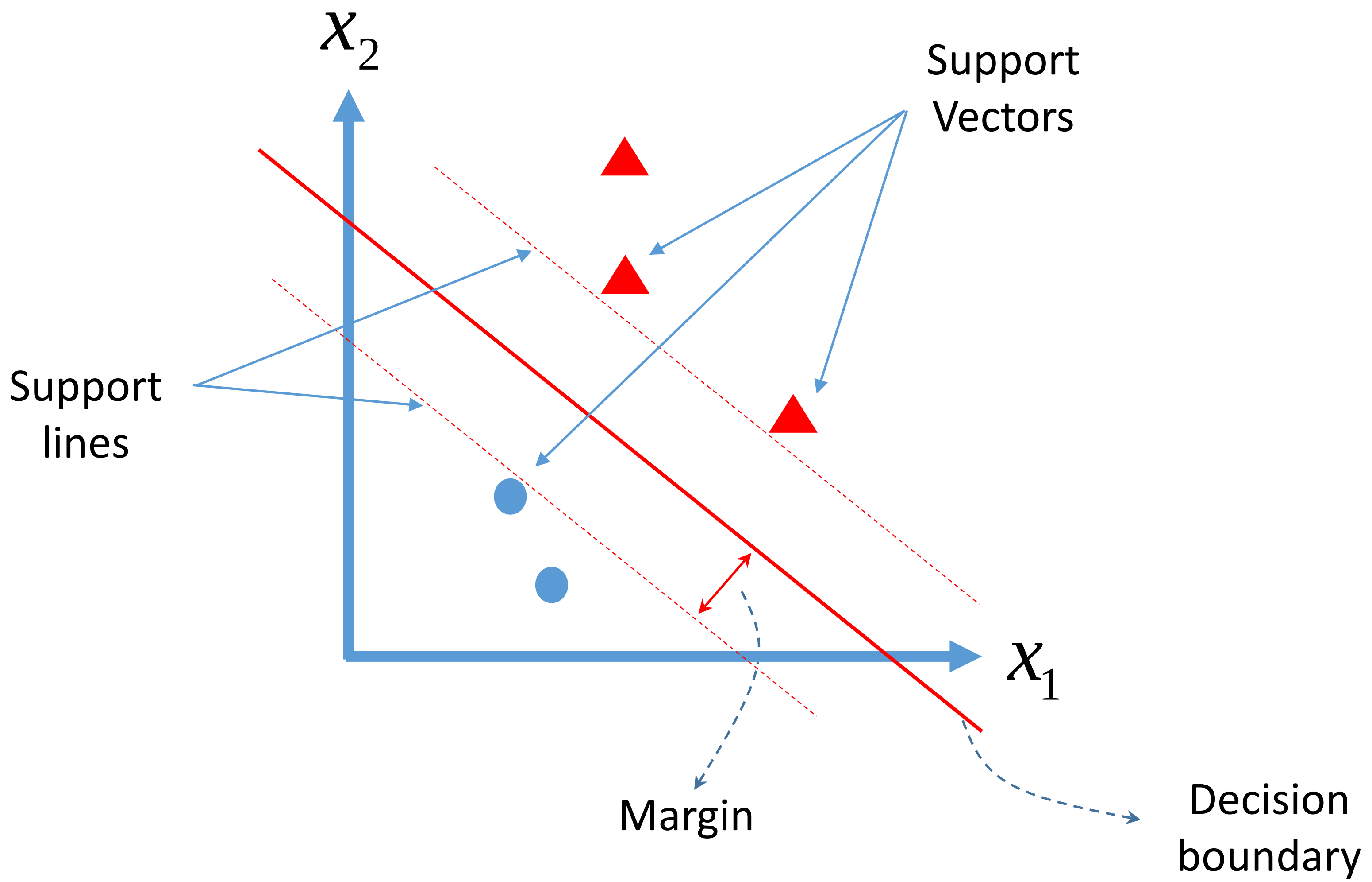
Why Support Vector Machine?

- Most widely used classification approach (practical)
  - Linearly separable data set
  - Linearly separable data set with a few violation
  - Non-linearly separable data set
- Supported by well defined mathematical theories
  - Geometry,
  - Optimization,
  - Quadratic programming,
  - Lagrange method,
  - Kernel, etc.

Which one is better for classification?



Terminology used in this lecture



Some geometry

$$y(\mathrm{x}) = w_2 x_2 + w_1 x_1 + w_0$$

$$y(\mathrm{x}^a) - y(\mathrm{x}^b) = w_2 x_2^a + w_1 x_1^a + w_0 - w_2 x_2^b - w_1 x_1^b - w_0$$

$$= w_2 (x_2^a - x_2^b) + w_1 (x_1^a - x_1^b)$$

$$= [w_1, w_2] \begin{bmatrix} x_1^a - x_1^b \\ x_2^a - x_2^b \end{bmatrix}$$

(1x2)(2x1)=(1x1)

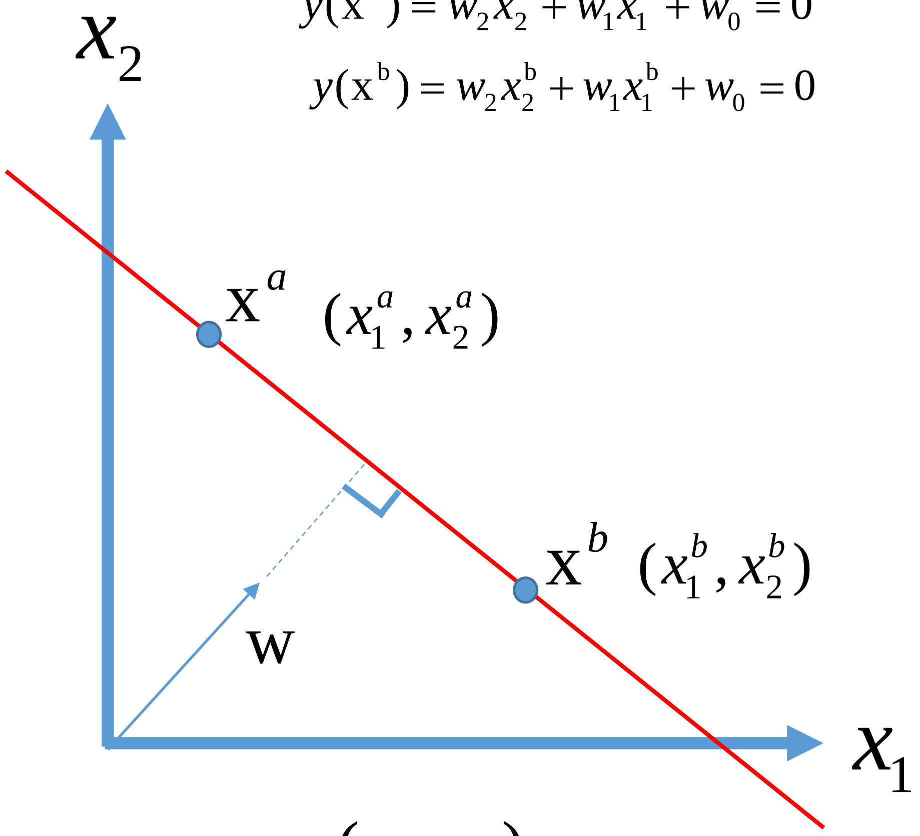
$$0 = \mathrm{w}^{\mathrm{T}} (\mathrm{x}^a - \mathrm{x}^b)$$

$$\mathrm{w}^{\mathrm{T}} \perp \boxed{(\mathrm{x}^a - \mathrm{x}^b)}$$

Vector on the decision boundary

$$y(\mathrm{x}^{\mathrm{a}}) = w_2 x_2^{\mathrm{a}} + w_1 x_1^{\mathrm{a}} + w_0 = 0$$

$$y(\mathrm{x}^{\mathrm{b}}) = w_2 x_2^{\mathrm{b}} + w_1 x_1^{\mathrm{b}} + w_0 = 0$$



$$\mathrm{x} = (x_1, x_2)$$

$$\mathrm{w} = (w_1, w_2)$$

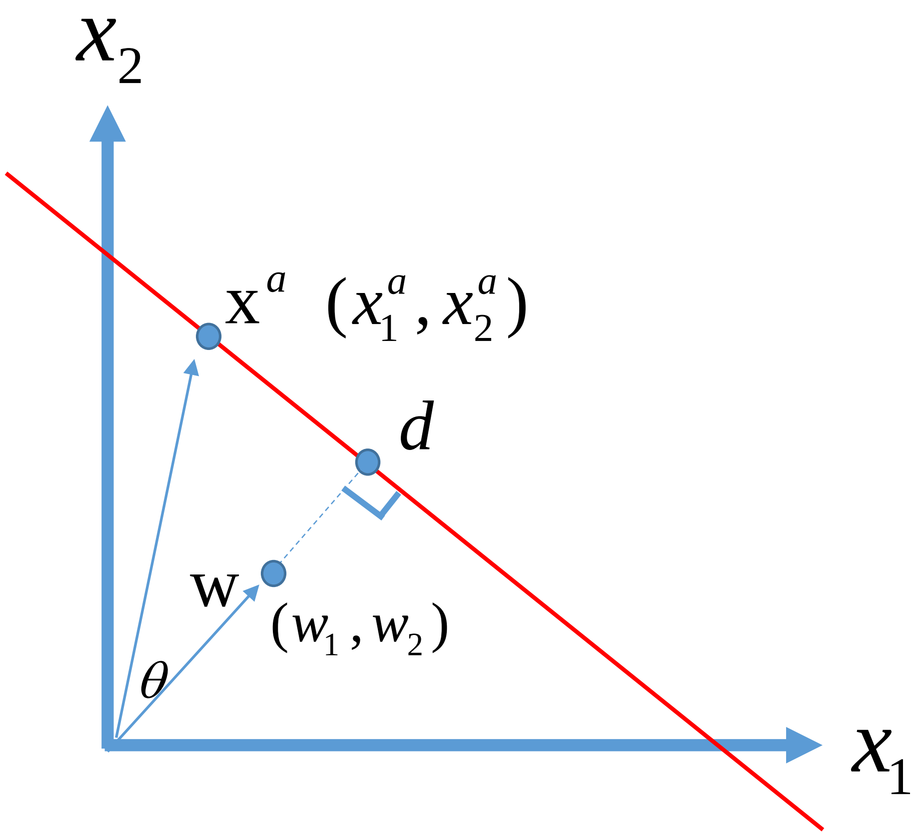
Some geometry

- Inner product

$$(x_1^a, x_2^a)\cdot(w_1, w_2) = \|(x_1^a, x_2^a)\|\|(w_1, w_2)\|\cos\theta$$

- cosθ definition

$$\cos\theta = \frac{\|d\|}{\|(x_1^a, x_2^b)\|} \Rightarrow \|d\| = \|(x_1^a, x_2^b)\|\cos\theta$$



$$\|(w_1, w_2)\|\|d\| = (x_1^a, x_2^b)\cdot(w_1, w_2)$$

$$\|d\| = \frac{w_2 x_2^a + w_1 x_1^a}{\|(w_1, w_2)\|} = \frac{-w_0}{\|(w_1, w_2)\|}$$

$$\|d\| = \frac{-w_0}{\|\mathrm{w}\|}$$

Margin distance

Size of the vector (x^b->x^c)

$$x^c = x^b + \|r\| \frac{w}{\|w\|}$$

Unit vector showing the direction only

- Let's multiply w^T and add w_0 in both sides.

$$w^T x^c + w_0 = w^T x^b + w_0 + w^T \|r\| \frac{w}{\|w\|}$$

$$y(x^c) = w^T \|r\| \frac{w}{\|w\|}$$

$$\|r\| = \frac{y(x^c)}{\|w\|}$$

$$\|r\| = \frac{1}{\|w\|}$$

Let's say

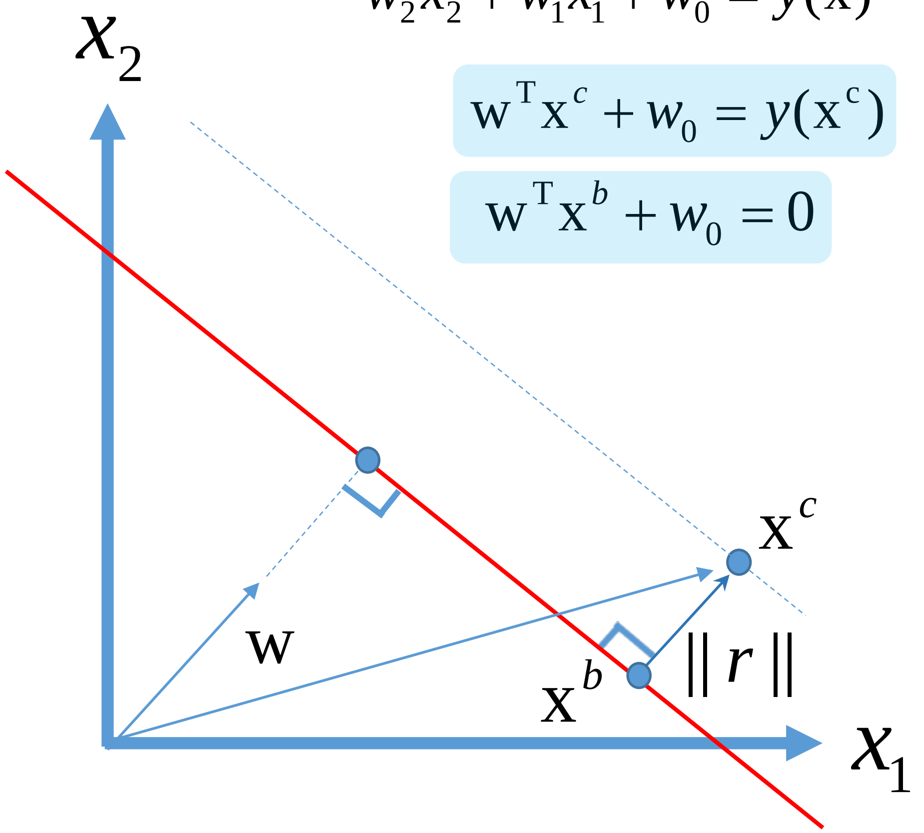
$$|y(x^c)| = 1$$

To make the margin become one… (see later)

$$w_2 x_2 + w_1 x_1 + w_0 = y(x)$$

$$w^T x^c + w_0 = y(x^c)$$

$$w^T x^b + w_0 = 0$$



Problem formulation

- Finding a decision boundary which maximizes the margin.

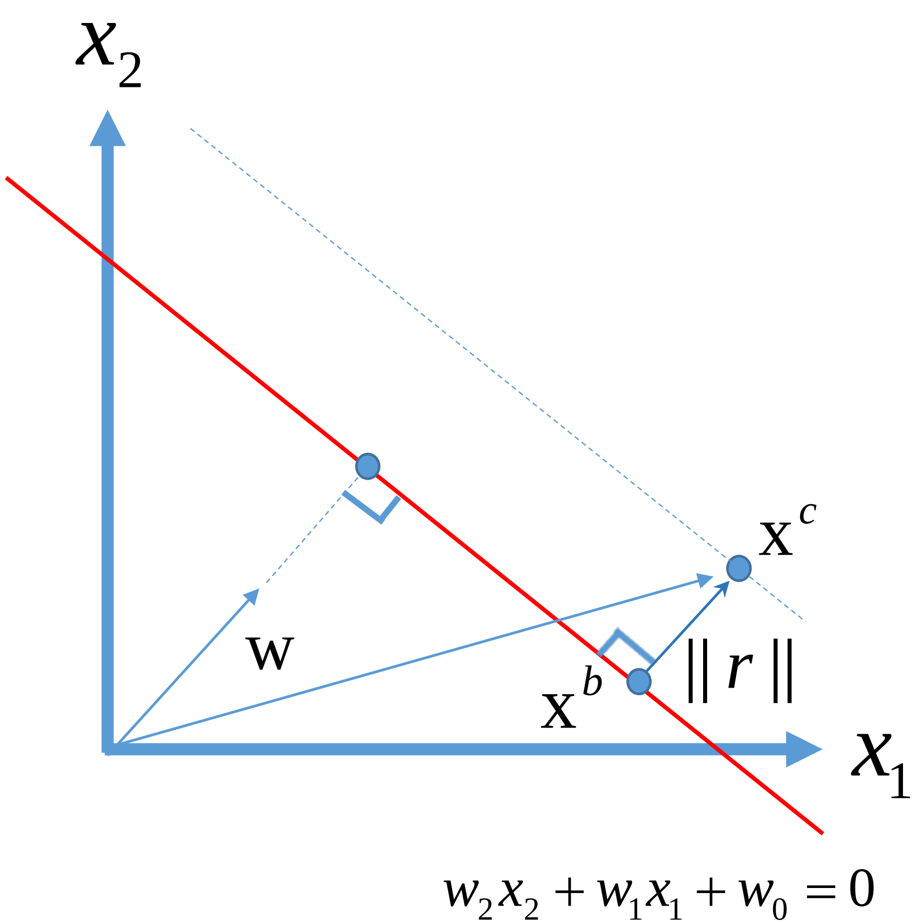
$$\max \| r \| = \frac{1}{\| \mathrm{w} \|}$$

s.t.

$$t_n y(\mathrm{x}_n) > 0$$

Every data points are classified correctly.

$$\begin{cases} t_n = +1, & y(\mathrm{x}_n) > 0 \\ t_n = -1, & y(\mathrm{x}_n) < 0 \end{cases}$$



Problem formulation

- Let's make it a quadratic programming problem.

$$\max \frac{1}{\|\mathrm{w}\|}$$

$$s.t. \quad t_n y(\mathrm{x}_n) > 0, \quad \forall n$$

- Do you remember?

$$\max \frac{1}{\|\mathrm{w}\|}$$

$$s.t. \quad t_n \mathrm{y}(\mathrm{x}_n) \geq 1, \quad \forall n$$

Let's say

$$|\,\mathrm{y}(\mathrm{x}^{\mathrm{c}})\,| = 1$$

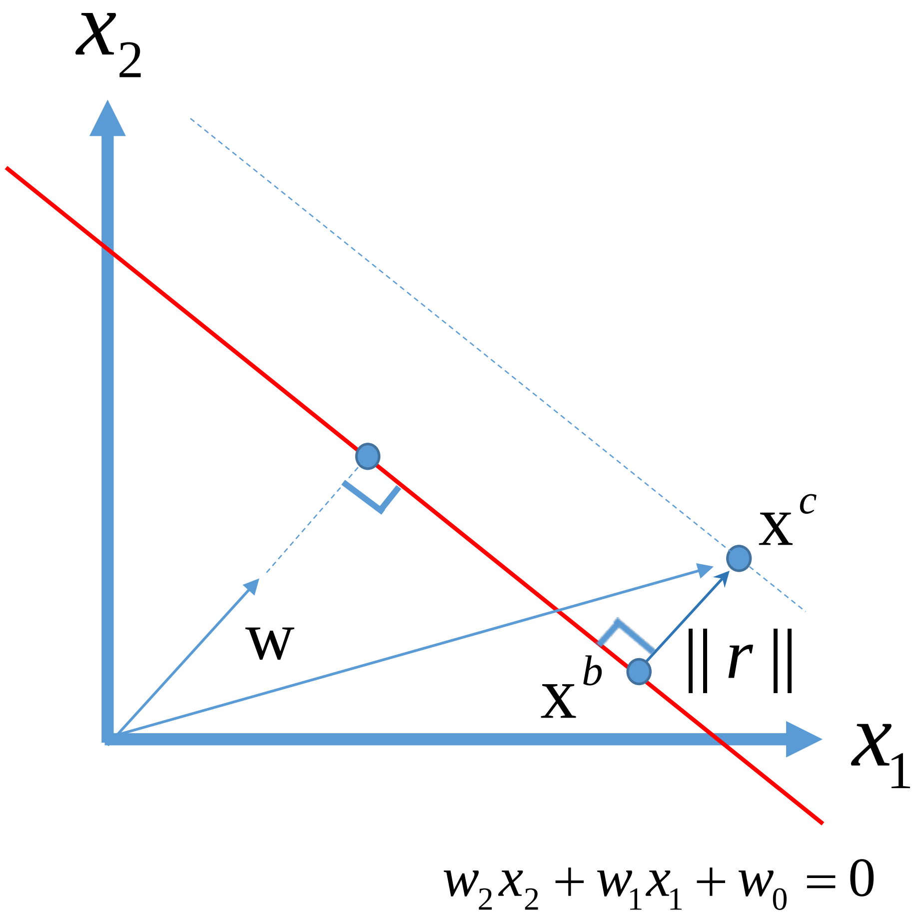
meaning that any data point is away from the decision boundary at least 1

- Finally

$$\min \frac{1}{2}\|\mathrm{w}\|^2$$

$$s.t. \quad t_n(\mathrm{w}^{\mathrm{T}}\mathrm{x}_n + w_0) \geq 1, \quad \forall n$$

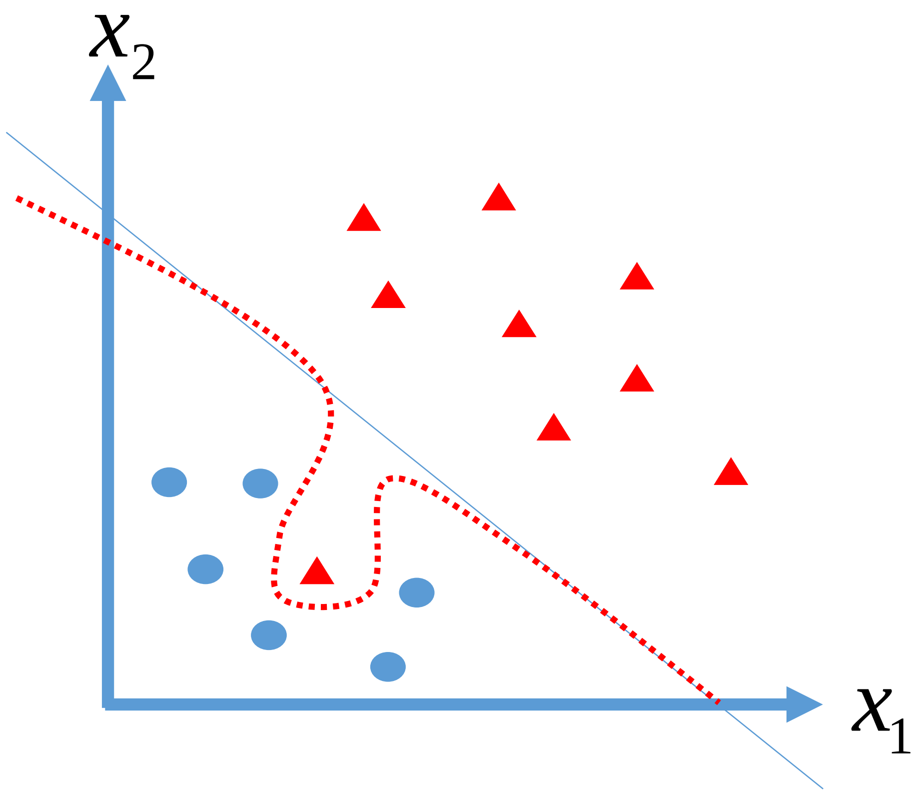
Quadratic programming

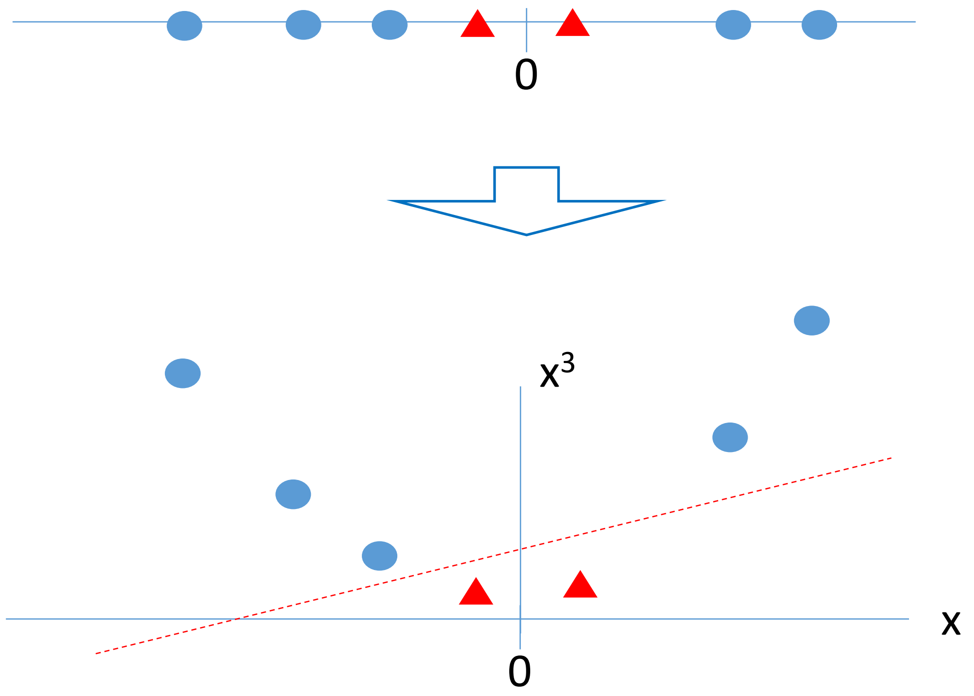


How about non-linearly separable case?

$$\min \frac{1}{2}\|\mathrm{w}\|^2$$
$$s.t. \quad t_n(\mathrm{w}^T\mathrm{x}_n + w_0) \ge 1, \quad \forall n$$

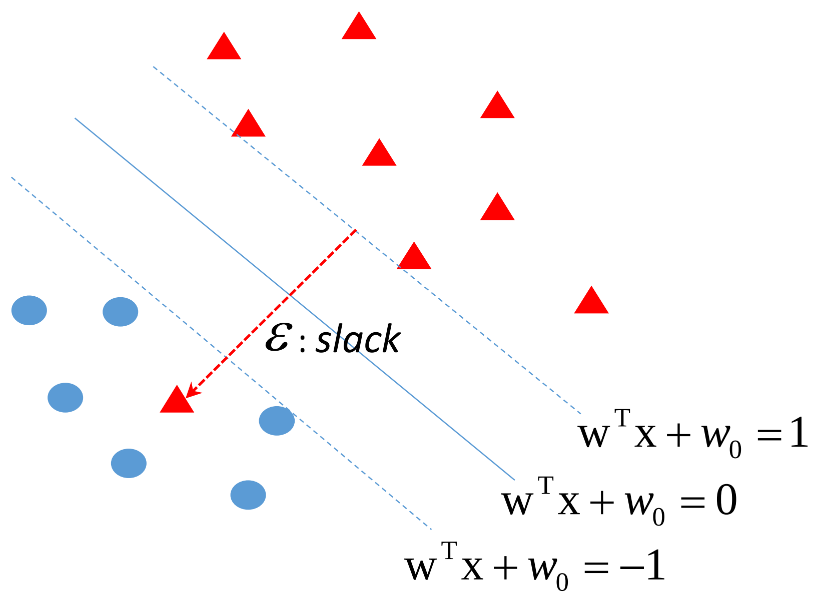
	Approaches
Option 1	Soft margin SVM
Option 2	Kernel trick

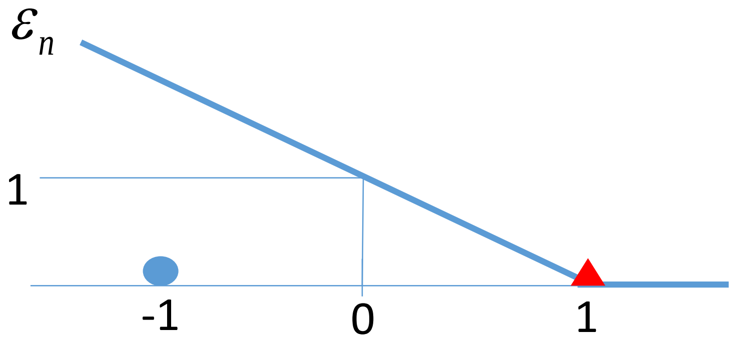




Soft margin SVM

Option 1: soft margin SVM





- Remember the constraint below?

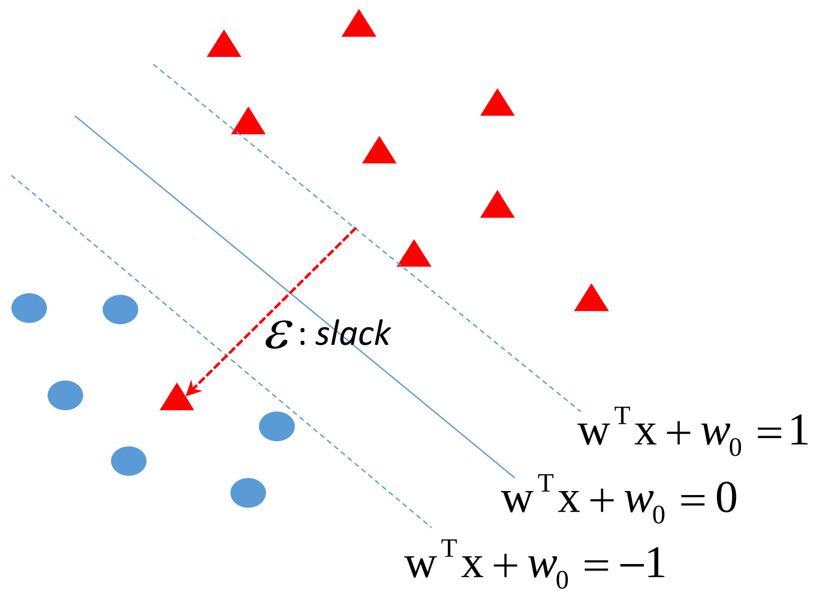
$$t_n(\mathrm{w}^{\mathrm{T}}\mathrm{x}_n + w_0) \geq 1, \quad \forall n$$

- For the data points which are non-separable, we relax the constraint:

$$t_n(\mathrm{w}^{\mathrm{T}}\mathrm{x}_n + w_0) \geq 1 \boxed{-\varepsilon_n}, \quad \forall n \quad \varepsilon_n \geq 0$$

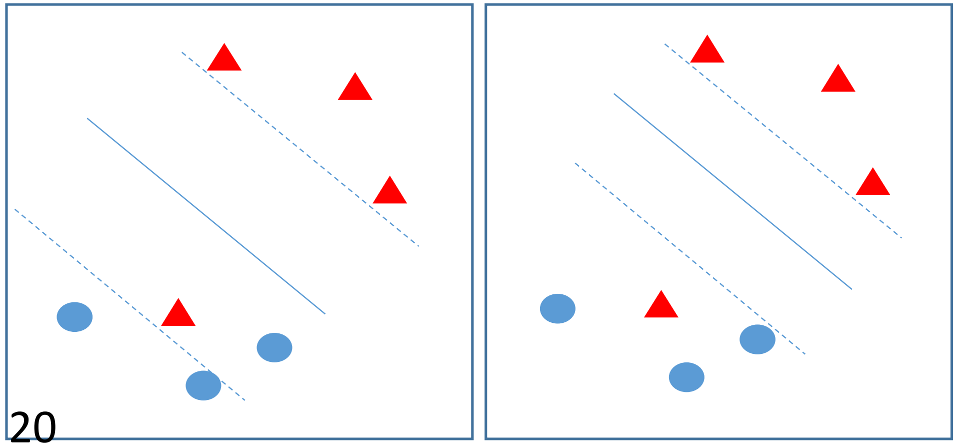
- It says that the distance between a data point and the decision boundary is allowed to be less than 1.
- ε_n is called slack variables.
- Question. Where is a data point when $\varepsilon_n = 1$?

Option 1: soft margin SVM



"C" is small

"C" is large



- So we have the constraint below. How about the objective function?

$$t_n(\mathrm{w}^T x_n + w_0) \geq 1 \boxed{-\varepsilon_n}, \quad \forall n \quad \varepsilon_n \geq 0$$

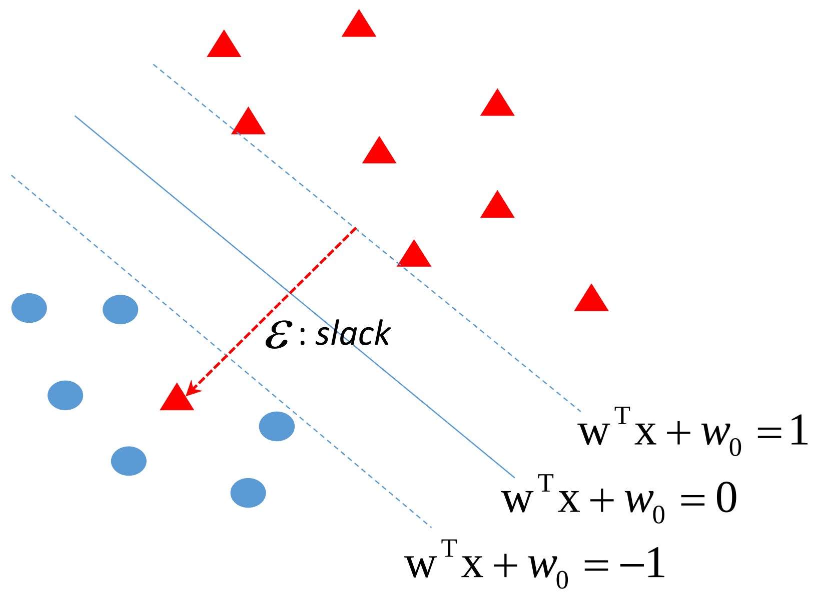
- We want to minimize the slack.

$$\min \frac{1}{2} \| \mathrm{w} \|^2 + C \sum_n \varepsilon_n$$

- If "C" is small, the slack contributes more
  1) Prefer large margin
  2) May cause large # of misclassified data points.

- If "C" is large, the slack contributes less
  1) Prefer less # of misclassified data points.
  2) May cause small margin.

Option 1: soft margin SVM



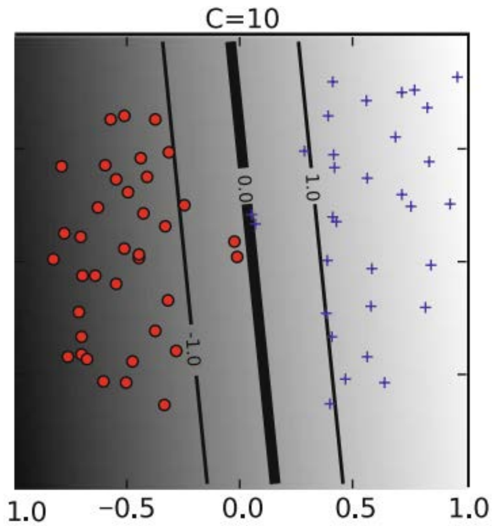
- The formulation finally becomes

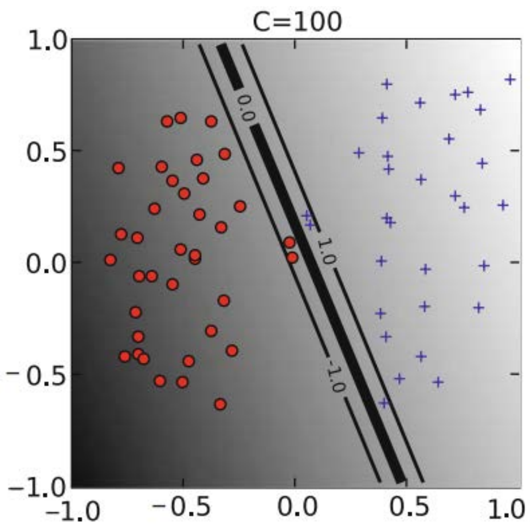
$$\min \frac{1}{2}\|w\|^2 + C\sum_n \varepsilon_n$$

$$s.t.$$

$$t_n(w^T x_n + w_0) \geq 1 - \varepsilon_n, \forall n$$

$$\varepsilon_n \geq 0$$

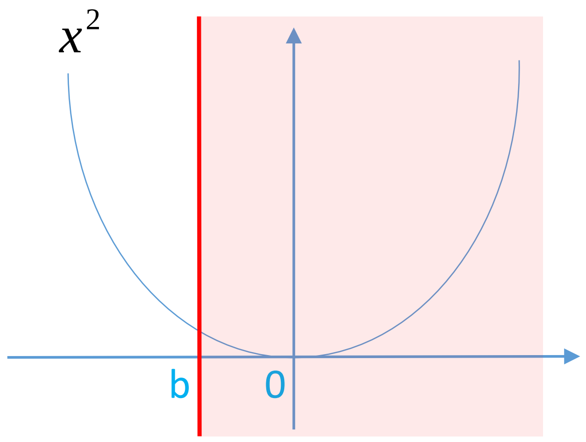




Kernel trick

Lagrange method for an optimization problem with inequality constraints

$$\min x^2 \quad s.t. \quad x \ge b$$



$$\min_x \max_\lambda x^2 - \lambda(x-b) \quad s.t. \quad \lambda \ge 0$$

- Minima is zero when $b \le 0$
- Minima is "b^2" when $b > 0$
- It means at optima: $\lambda(x\text{-}b) = 0$ (complementary slackness)
- Maximizing λ results in minimizing the objective value
    - $\lambda \ge 0$ (it should be because x-b ≥0)

Convert the quadratic problem in SVM to Lagrange optimization problem

Primal problem

$$\min_{\mathrm{w}} \max_{\lambda} \frac{1}{2}\mathrm{w}^T\mathrm{w} - \sum_{n=1}^{n} \lambda_n (t_n(\mathrm{w}^{\mathrm{T}} x_n + w_0) - 1)$$
$$s.t. \quad \lambda_n \geq 0$$

KKT conditions

1) Stationarity condition

$$\frac{\partial}{\partial \mathrm{w}}\frac{1}{2}\mathrm{w}^T\mathrm{w} - \frac{\partial}{\partial \mathrm{w}}\sum_{n=1}^{n} \lambda_n (t_n(\mathrm{w}^{\mathrm{T}} x_n + w_0) - 1) = 0$$

2) Complementary slackness condition

$$\lambda_n (t_n(\mathrm{w}^{\mathrm{T}} x_n + w_0) - 1) = 0$$

3) Duality feasibility condition

$$\lambda_n \geq 0$$

- We would like to convert again the optimization problem above into another form, which provides same results.
  - Because we want to solve the optimization problem in term of "lagrange multiplier (λ_n)".

Dual problem

$$\max_{\lambda} \min_{\mathrm{w}} \frac{1}{2}\mathrm{w}^T\mathrm{w} - \sum_{n=1}^{n} \lambda_n (t_n(\mathrm{w}^{\mathrm{T}} x_n + w_0) - 1)$$
$$s.t. \quad \lambda_n \geq 0$$

Dual problem of the quadratic problem: applying stationarity condition

$$\max_{\lambda} \min_{w,w_0} L(\mathrm{w}, \mathrm{w}_0, \lambda) = \frac{1}{2}\mathrm{w}^{\mathrm{T}}\mathrm{w} - \sum_{n=1}^{N} \lambda_n \left(t_n(\mathrm{w}^{\mathrm{T}} x_n + w_0) - 1\right)$$

$$\boxed{\mathrm{w} = \sum_{n=1}^{N} \lambda_n t_n x_n} \qquad \boxed{\sum_{n=1}^{N} \lambda_n t_n = 0}$$

$$L(\lambda) = \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} t_n t_m \lambda_n \lambda_m \mathrm{x}_n^T \mathrm{x}_m - \sum_{n=1}^{N}\sum_{m=1}^{N} t_n t_m \lambda_n \lambda_m \mathrm{x}_n^T \mathrm{x}_m - \sum_{n=1}^{N} \lambda_n t_n w_0 + \sum_{n=1}^{N} \lambda_n$$

$$\max_{\lambda} L(\lambda) = \sum_{n=1}^{N} \lambda_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} t_n t_m \lambda_n \lambda_m \mathrm{x}_n^T \mathrm{x}_m$$

$$\lambda_n \geq 0 \qquad \boxed{\sum_{n=1}^{N} \lambda_n t_n = 0} \qquad \boxed{\cancel{\mathrm{w} = \sum_{n=1}^{N} \lambda_n t_n x_n}}$$

"w" does not appear in the equation, and so we do not use this constraint anymore

Finally…

$$\max_{\lambda} L(\lambda) = \sum_{n=1}^{N} \lambda_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} t_n t_m \lambda_n \lambda_m \mathrm{x}_n^T \mathrm{x}_m$$

$$s.t. \quad \lambda_n \geq 0, \quad \sum_{n=1}^{N} \lambda_n t_n = 0$$

$$\min_{\lambda} L(\lambda) = \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} t_n t_m \lambda_n \lambda_m \mathrm{x}_n^T \mathrm{x}_m - \sum_{n=1}^{N} \lambda_n$$

$$s.t. \quad \lambda_n \geq 0, \quad \sum_{n=1}^{N} \lambda_n t_n = 0$$

- Again, the optimization problem becomes a quadratic programming problem.

Let’s summarize

$$\min_{\lambda} \quad L(\lambda) = \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} t_n t_m \lambda_n \lambda_m \mathrm{x}_n^T \mathrm{x}_m - \sum_{n=1}^{N} \lambda_n$$

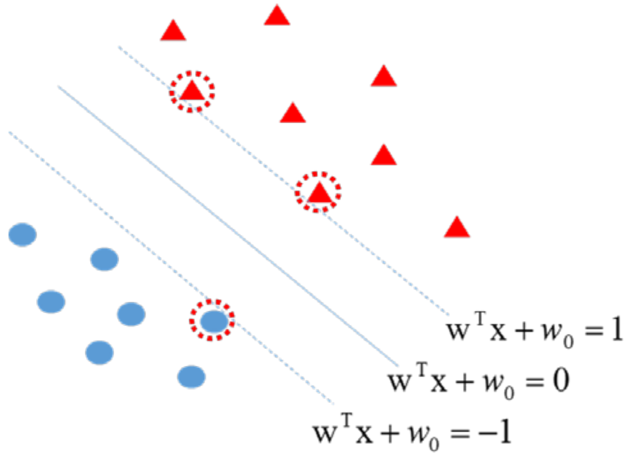
$$s.t. \quad \lambda \geq 0, \qquad t^T \lambda = 0$$

- The solution from the quadratic programming is “lagrange multipliers”(λ_n)
- Many of the solutions (lagrange multipliers) are zero
- Complementary slackness (one of KKT conditions) should be satisfied.

$$\lambda_n (t_n (\mathrm{w}^T x_n + w_0) - 1) = 0$$

- In other words, if λ_n are not zero, $(t_n(w_t x_n + w_0) - 1)$ should be zero where corresponding data points should be support vectors.
- With the non-zero λ_n, w and w_0 can be calculated using $t_n(w_t x_n + w_0) = 1$

$$\mathrm{w} = \sum_{n=1}^{N} \lambda_n t_n x_n \qquad w_0 = t_n - \sum_{n=1}^{N} \lambda_n t_n x_n x_n$$



Kernel trick

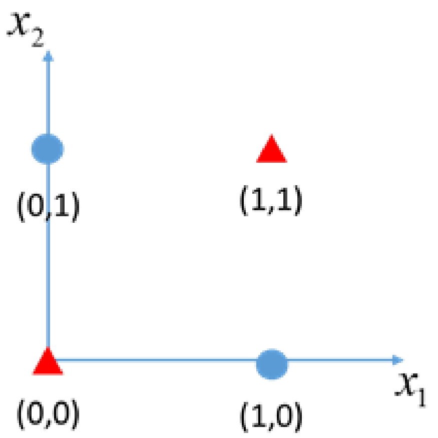
$$\min \frac{1}{2}\mathbf{w}^T\mathbf{w}$$

$$s.t. \quad t_n(\mathbf{w}^T x_n + w_0) \geq 1$$

$$\min_{\lambda} \quad L(\lambda) = \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} t_n t_m \lambda_n \lambda_m \mathbf{x}_n^T \mathbf{x}_m - \sum_{n=1}^{N} \lambda_n$$

$$s.t. \quad \lambda \geq 0, \qquad t^T \lambda = 0$$

- ❑ If data x_n are not linearly separable, what should we do?

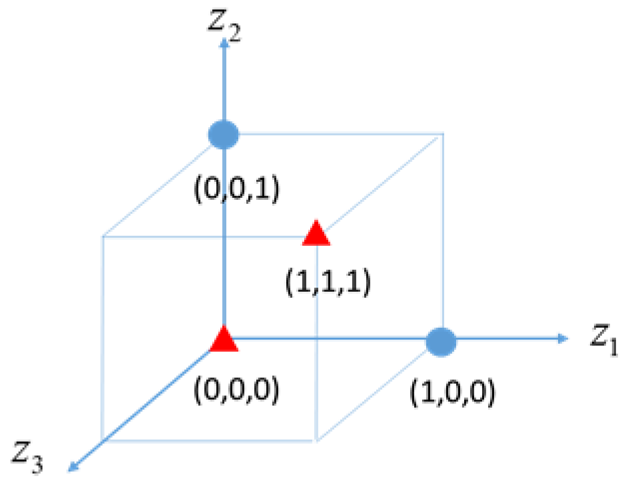


Space X

$$\phi(\mathbf{x}) = \begin{pmatrix} x_1^2 \\ x_1 x_2 \\ x_2^2 \end{pmatrix}$$

$$\phi(\mathbf{x})$$

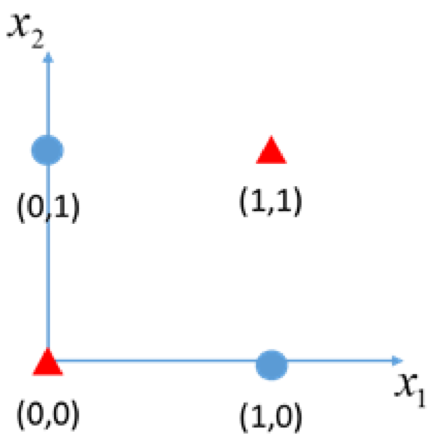
$$X \rightarrow Z$$



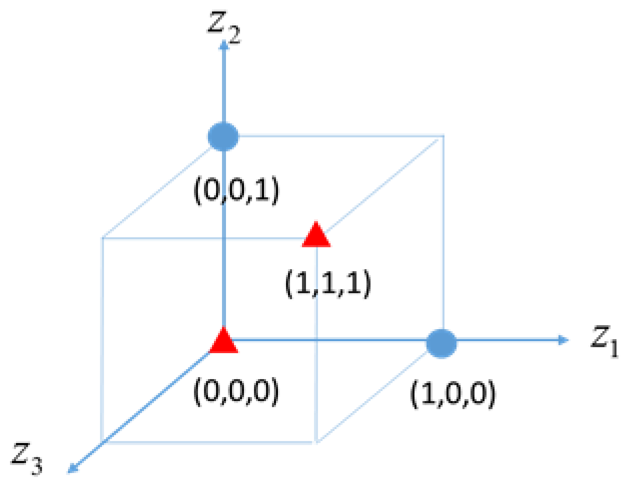
Space Z

Kernel trick

- The idea of Kernel trick begins from here: to find the scalar values (the inner product of two vectors: z_n and z_m) and so we can formulate the quadratic problem which can be linearly separable.



Space X



Space Z

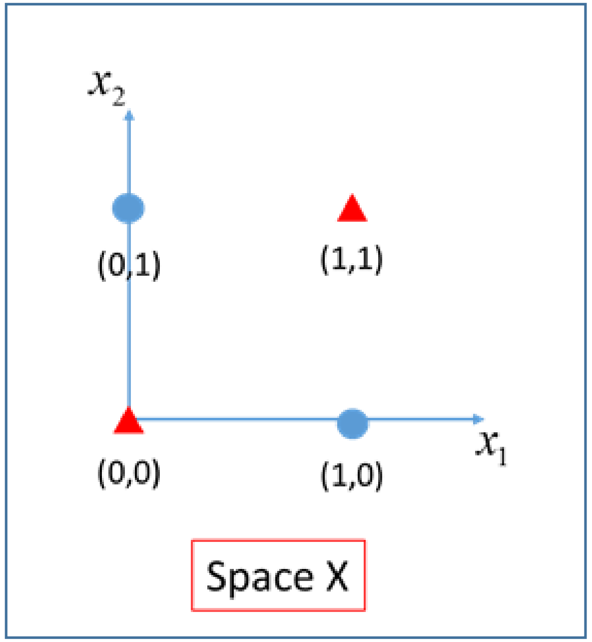
$$\min_{\lambda} \; L(\lambda) = \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} t_n t_m \lambda_n \lambda_m \mathbf{x}_n^T \mathbf{x}_m - \sum_{n=1}^{N} \lambda_n$$
$$s.t. \quad \lambda \geq 0, \qquad t^T \lambda = 0$$

$$\min_{\lambda} \; L(\lambda) = \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} t_n t_m \lambda_n \lambda_m \mathbf{z}_n^T \mathbf{z}_m - \sum_{n=1}^{N} \lambda_n$$
$$s.t. \quad \lambda \geq 0, \qquad t^T \lambda = 0$$

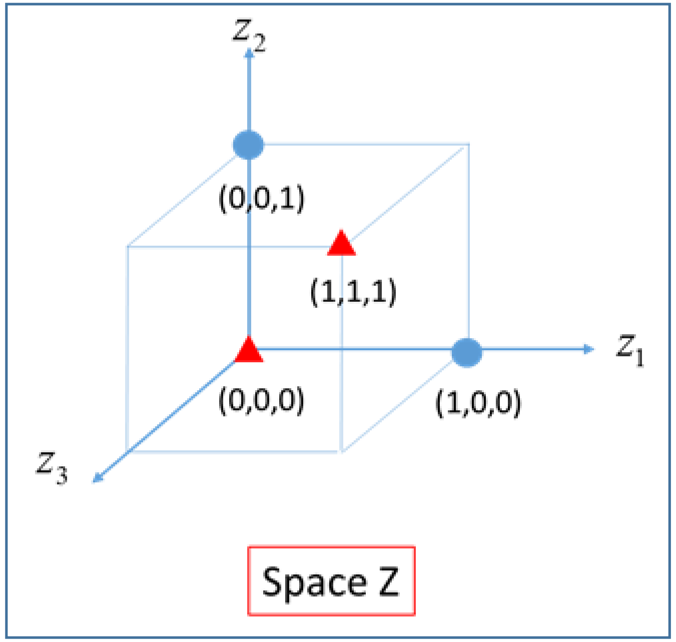
Kernel trick

- Kernel function K() is a function which returns the scalar values (the inner product of two vectors: z_n and z_m in Z space) when the data points (x_n and x_m in X space) are given.

$$K(\mathrm{x}_n^T, \mathrm{x}_m) = \phi(\mathrm{x}_n^T)\phi(\mathrm{x}_m) = \mathrm{z}_n^T \mathrm{z}_m$$



$$\mathrm{z}_n^T = \phi(\mathrm{x}_n^T)$$
$$\mathrm{z}_m = \phi(\mathrm{x}_m)$$



Finally finally…

- With the Kernel function defined previously, we want to change the quadratic problem as follows:
  - Because the Kernel function is a function of data points (x_n and x_m) which we already have.

$$\min_{\lambda} \; L(\lambda) = \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} t_n t_m \lambda_n \lambda_m z_n^T z_m - \sum_{n=1}^{N}\lambda_n$$
$$s.t. \quad \lambda \geq 0, \qquad t^T\lambda = 0$$

$\Rightarrow$

$$\min_{\lambda} \; L(\lambda) = \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} t_n t_m \lambda_n \lambda_m \mathrm{K}(x_n^T x_m) - \sum_{n=1}^{N}\lambda_n$$
$$s.t. \quad \lambda \geq 0, \qquad t^T\lambda = 0$$

$$\min_{\lambda} \; L(\lambda) = \frac{1}{2}\lambda^T \begin{bmatrix} t_1 t_1 K(x_1, x_1) & t_1 t_2 K(x_1^T, x_2) & \cdots & t_1 t_N K(x_1^T, x_N) \\ t_2 t_1 K(x_2, x_1) & t_2 t_2 K(x_2^T, x_2) & \cdots & t_2 t_N K(x_2^T, x_N) \\ \cdots & \cdots & \cdots & \cdots \\ t_N t_1 K(x_N x_1) & t_N t_2 K(x_N^T, x_2) & \cdots & t_N t_N K(x_N^T, x_N) \end{bmatrix} \lambda + (-1^T)\lambda$$

Finally finally…

$$\min_{\lambda} \; L(\lambda) = \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} t_n t_m \lambda_n \lambda_m \mathrm{K}(\mathrm{x}_n^T \mathrm{x}_m) - \sum_{n=1}^{N} \lambda_n$$

$$s.t. \quad \lambda \geq 0, \qquad t^T \lambda = 0$$

$$\mathrm{w} = \sum_{z_n \in SV} \lambda_n t_n \mathrm{z}_n \qquad w_0 = t_n - \sum_{z_n \in SV} \lambda_n t_n z_n z_n = t_n - \sum_{z_n \in SV} \lambda_n t_n K(\mathrm{x}_n, \mathrm{x}_n)$$

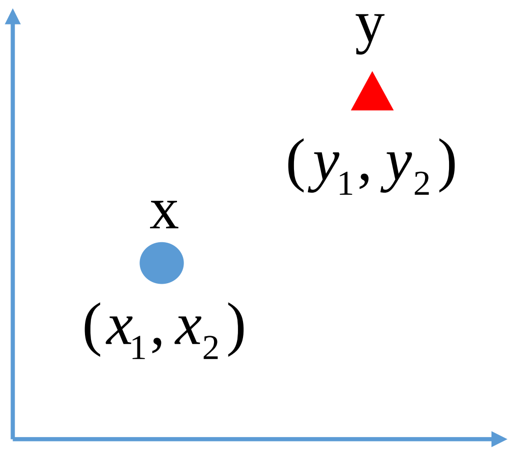
$$\mathrm{sign}(\mathrm{w}^{\mathrm{T}} \mathrm{z} + w_0)$$

$$\mathrm{sign}\left(\sum \lambda_n t_n \mathrm{z}_n \mathrm{z} + t_n - \sum_{z_n \in SV} \lambda_n t_n K(\mathrm{x}_n, \mathrm{x}_n) \right)$$

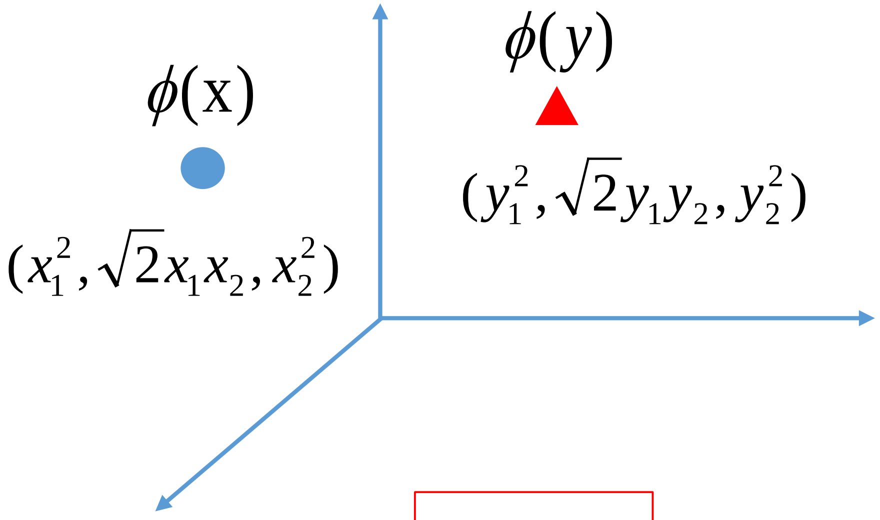
$$\mathrm{sign}\left(\sum \lambda_n t_n K(x_n, x) + t_n - \sum \lambda_n t_n K(x_n, x_n) \right)$$

- Now you have a function, which classifies a data point in z space without mapping the data point to z space at all.
- Do you see why it is called a trick?

Polynomial kernel of degree 2



Space X



Space Z

$$
\begin{aligned}
K(\mathrm{x}, y) &= (\mathrm{xy})^2 \\
&= \left((x_1, x_2)\cdot(y_1, y_2)\right)^2 \\
&= \left(x_1y_1 + x_2y_2\right)^2 \\
&= x_1^2y_1^2 + 2x_1x_2y_1y_2 + x_2^2y_2^2
\end{aligned}
$$

$$
\begin{aligned}
\phi(\mathrm{x})\phi(y) &= (x_1^2, \sqrt{2}x_1x_2, x_2^2)\cdot(y_1^2, \sqrt{2}y_1y_2, y_2^2) \\
&= x_1^2y_1^2 + 2x_1x_2y_1y_2 + x_2^2y_2^2
\end{aligned}
$$

Mapping to 3-dimension

Gaussian Kernel: derivation (inner product in the infinite z space)

Taylor series expansion of an exponential function

$$\exp(x) = \frac{x^0}{0!} + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$$

$$K(\mathrm{x}_n, \mathrm{x}_m) = \exp\left(-\alpha \, \| \mathrm{x}_\mathrm{n} - \mathrm{x}_\mathrm{m} \|^2\right)$$

$$= \exp\left(-\alpha \mathrm{x}_n^2\right) \exp\left(-\alpha \mathrm{x}_m^2\right) \exp\left(2\alpha \mathrm{x}_n \mathrm{x}_m\right)$$

$$= \exp\left(-\alpha \mathrm{x}_n^2\right) \exp\left(-\alpha \mathrm{x}_m^2\right) \sum_{k=0}^{\infty} \frac{(2\alpha)^k (\mathrm{x}_\mathrm{n})^k (\mathrm{x}_\mathrm{m})^k}{\mathrm{k}!}$$

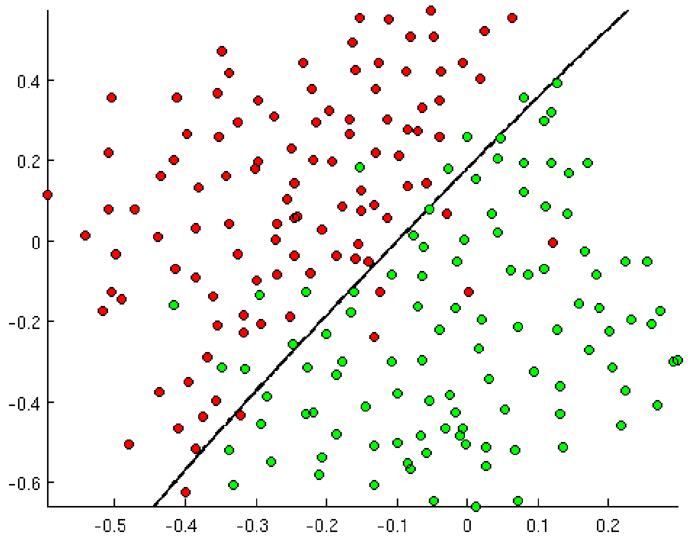
$$= \sum_{k=0}^{\infty} \sqrt{\frac{(2\alpha)^k}{k!}} \exp\left(-\alpha \mathrm{x}_n^2\right) (\mathrm{x}_\mathrm{n})^k \sqrt{\frac{(2\alpha)^k}{k!}} \exp\left(-\alpha \mathrm{x}_m^2\right) (\mathrm{x}_\mathrm{m})^k$$

$$= \phi(\mathrm{x}_\mathrm{n}) \phi(\mathrm{x}_\mathrm{m})$$

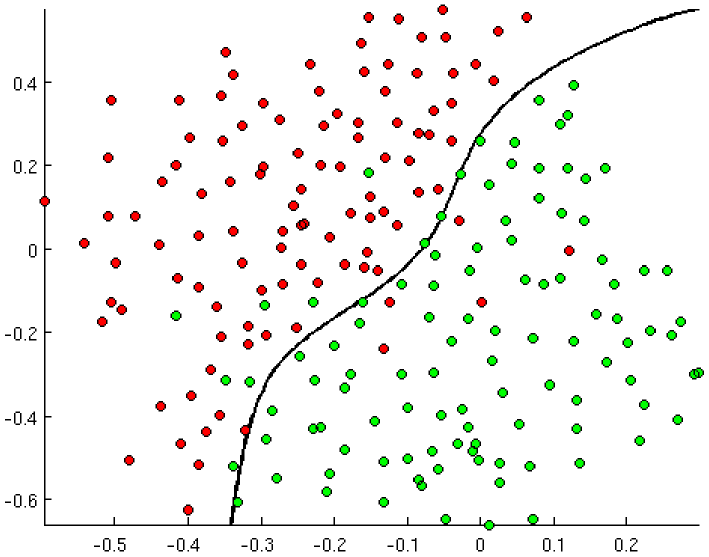
Mapping to infinite-dimension !

Gaussian Kernel

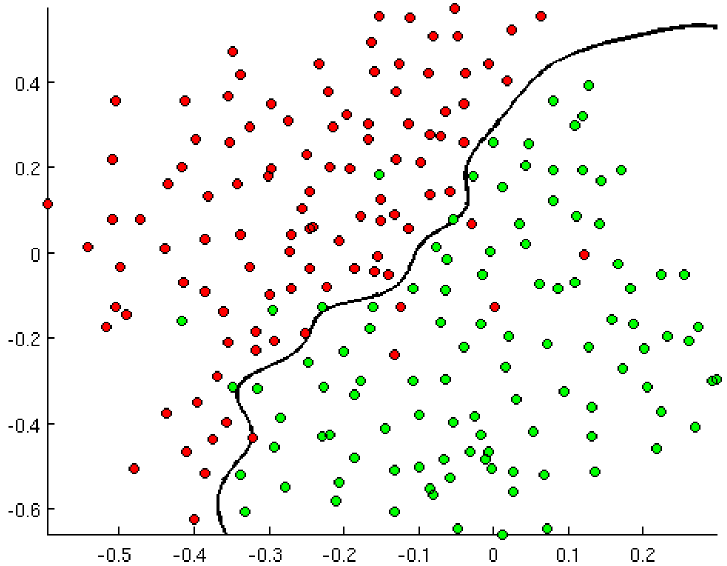
$\alpha = 1$



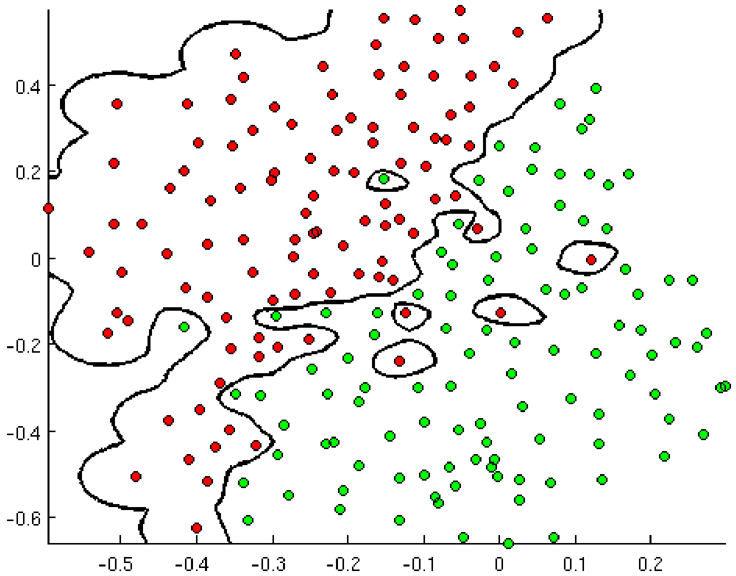
$\alpha = 10$



$\alpha = 100$



$\alpha = 1000$



http://openclassroom.stanford.edu/MainFolder/DocumentPage.php?course=MachineLearning&doc=exercises/ex8/ex8.html

Backup Slides

Variational inference: derivation

$$KL(q(Z)\,\|\,p(Z\mid X)) = -\sum_{Z} q(Z)\log\frac{p(Z\mid X)}{q(Z)}$$

→ Finding q(Z) which minimizes the Kullback divergence

$$= -\sum_{Z} q(Z)\log\frac{p(X,Z)}{q(Z)\,p(X)}$$

$$KL = -\sum_{Z} q(Z)\log\frac{p(X,Z)}{q(Z)} + \sum_{Z} q(Z)\log p(X)$$

$$KL + \sum_{Z} q(Z)\log\frac{p(X,Z)}{q(Z)} = \sum_{Z} q(Z)\log p(X)$$

Variational inference: derivation

$$KL + \sum_{Z} q(Z) \log \frac{p(X,Z)}{q(Z)} = \sum_{Z} q(Z) \log p(X)$$

- Always positive

- Always Negative
- Lower bound (***L***)

- Always negative
- It is a fixed value

- 4-8=-1
- 1-2=-1

- ❑ KL divergence and lower bound are a function of "q(Z)"
- ❑ Minimizing KL divergence is equivalent to maximizing the lower bound (*L*).

$$KL = -\sum_{Z} q(Z) \log \frac{p(Z \mid X)}{q(Z)}$$

- We do not have this conditional distribution

$$L = \sum_{Z} q(Z) \log \frac{p(X,Z)}{q(Z)}$$

- We do have the joint distribution

Variational inference: derivation

$$\max L = \sum_Z q(Z)\log\frac{p(X,Z)}{q(Z)}$$

$$= \sum_{z_1}\sum_{z_2} q(z_1)q(z_2)\log\frac{p(x_1,x_2,z_1,z_2)}{q(z_1)q(z_2)}$$

Assuming that z_1 and z_2 are independent

$$= \sum_{z_1}\sum_{z_2} q(z_1)q(z_2)\left[\log p(x_1,x_2,z_1,z_2) - \log q(z_1)q(z_2)\right]$$

$$= \sum_{z_1}\sum_{z_2} q(z_1)q(z_2)\left[\log p(x_1,x_2,z_1,z_2) - \log q(z_1) - \log q(z_2)\right]$$

$$= \sum_{z_1}\sum_{z_2} q(z_1)q(z_2)\log p(x_1,x_2,z_1,z_2) - \sum_{z_1}\sum_{z_2} q(z_1)q(z_2)\log q(z_1) - \sum_{z_1}\sum_{z_2} q(z_1)q(z_2)\log q(z_2)$$

Assuming that $q(z_2)$ is known, and so we just look for $q(z_1)$

Variational inference: derivation

$$L = \sum_{z_1}\sum_{z_2} q(z_1)q(z_2)\log p(x_1,x_2,z_1,z_2) - \sum_{z_1}\sum_{z_2} q(z_1)q(z_2)\log q(z_1) - \sum_{z_1}\sum_{z_2} q(z_1)q(z_2)\log q(z_2)$$

Assuming that $q(z_2)$ is known, and so we just look for $q(z_1)$

$$= \sum_{z_1}\sum_{z_2} q(z_1)q(z_2)\log p(x_1,x_2,z_1,z_2) - \sum_{z_1} q(z_1)\log q(z_1)\sum_{z_2} q(z_2) - \sum_{z_1} q(z_1)\sum_{z_2} q(z_2)\log q(z_2)$$

$$= \sum_{z_1} q(z_1)\sum_{z_2} q(z_2)\log p(x_1,x_2,z_1,z_2) - \sum_{z_1} q(z_1)\log q(z_1)\sum_{z_2} q(z_2) - \sum_{z_1} q(z_1)\sum_{z_2} q(z_2)\log q(z_2)$$

$$= \sum_{z_1} q(z_1)E_{z_2}\left[\log p(x_1,x_2,z_1,z_2)\right] - \sum_{z_1} q(z_1)\log q(z_1) - K\boxed{\sum_{z_1} q(z_1)}$$

It is one but we keep it for a while

$$= \sum_{z_1} q(z_1)\left[E_{z_2}\left[\log p(x_1,x_2,z_1,z_2)\right] - K\right] - \sum_{z_1} q(z_1)\log q(z_1)$$

Variational inference: derivation

$$L = \sum_{z_1} q(z_1)\left[E_{z_2}[\log p(X,Z)] - K_1 - K_2\right] - \sum_{z_1} q(z_1)\log q(z_1)$$

$$\log f(X,Z) = E_{z_2}[\log p(X,Z)] - K_1$$

$$f(X,Z) = e^{E_{z_2}[\log p(X,Z)] - K_1} = e^{-K_1} e^{E_{z_2}[\log p(X,Z)]} = Ce^{E_{z_2}[\log p(X,Z)]}$$

If we choose "C" carefully, f(X,Z) can be a probability distribution. $\int\int Ce^{E_{z_2}[\log p(X,Z)]} dXdZ = 1$

$$L = \sum_{z_1} q(z_1)[\log f(X,Z) - K_2] - \sum_{z_1} q(z_1)\log q(z_1)$$

$$= \sum_{z_1} q(z_1)\log f(X,Z) - \sum_{z_1} q(z_1)K_2 - \sum_{z_1} q(z_1)\log q(z_1)$$

$$= \sum_{z_1} q(z_1)\frac{\log f(X,Z)}{\log q(z_1)} - \sum_{z_1} q(z_1)K_2 = \sum_{z_1} q(z_1)\frac{\log f(X,Z)}{\log q(z_1)} + C'$$

Variational inference: derivation

$$L = \sum_{z_1} q(z_1)\frac{\log f(X,Z)}{\log q(z_1)} - \sum_{z_1} q(z_1) K_2 = \sum_{z_1} q(z_1)\frac{\log f(X,Z)}{\log q(z_1)} + C'$$

$$\log f(X,Z) = E_{z_2}\left[\log p(X,Z)\right] - K_1$$ We defined it previously

$$f(X,Z) = e^{E_{z2}[\log p(X,Z)]-K_1} = e^{-K_1} e^{E_{z2}[\log p(X,Z)]} = Ce^{E_{z2}[\log p(X,Z)]}$$

Lower bound (L) is maximized when log q(z1) and log p(X,Z) are equal because it is a negative KL.
Thus, …

$$\log q(z_1) = \log f(X,Z)$$

$$q(z_1) = f(X,Z) = C_1 e^{E_{z2}[\log p(X,Z)]} = C_1 e^{\sum_{z2} q(z_2)\log p(X,Z)}$$

$$q(z_2) = f(X,Z) = C_2 e^{E_{z1}[\log p(X,Z)]} = C_2 e^{\sum_{z1} q(z_1)\log p(X,Z)}$$