ML_3

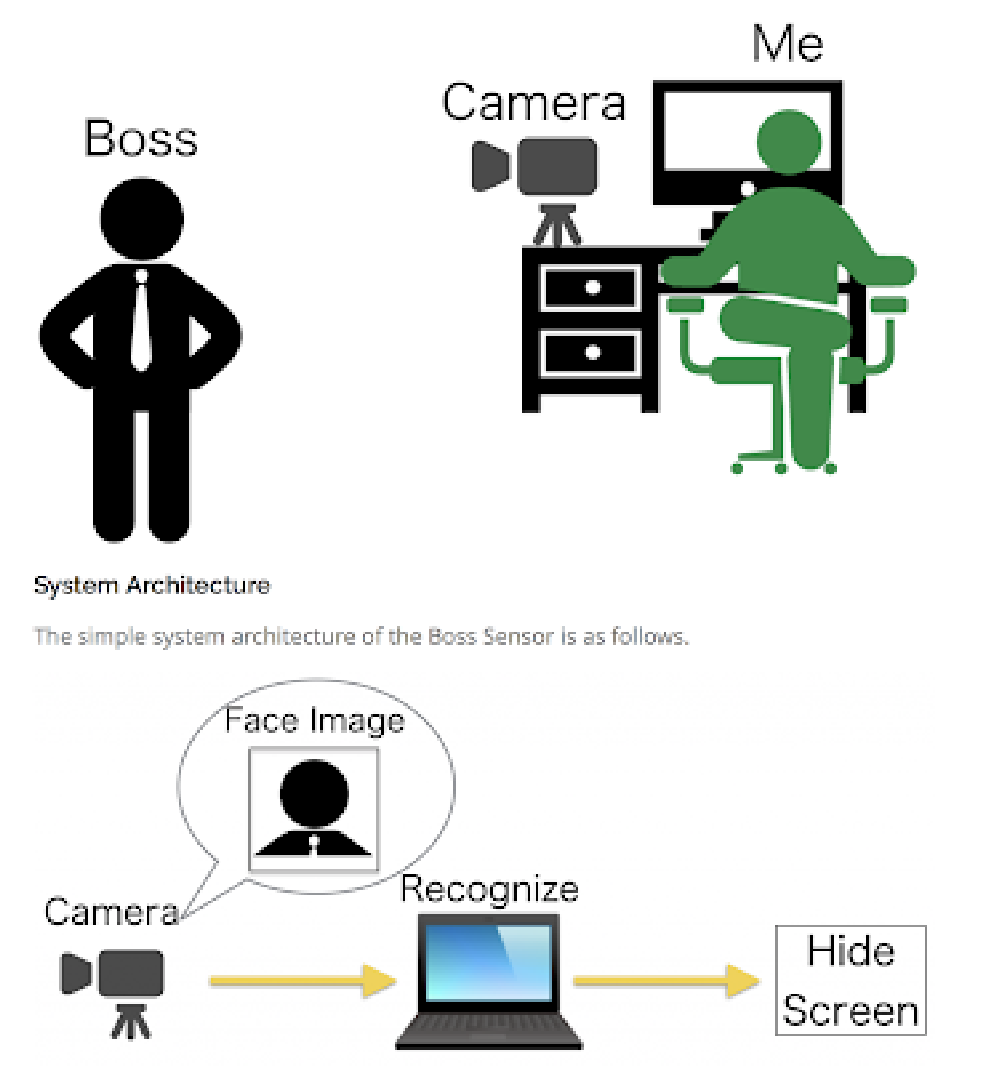
Team BMS

INDEX

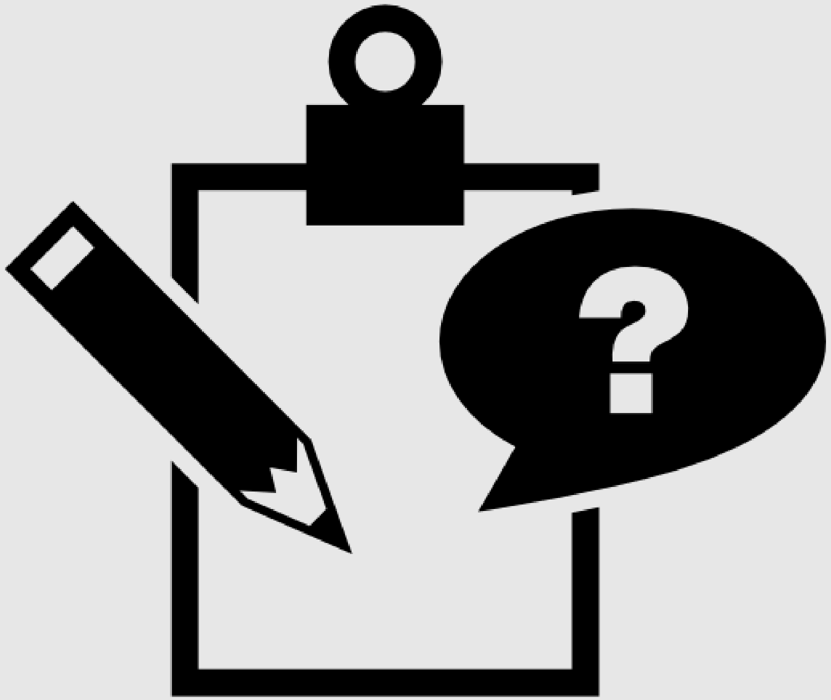
ML Study
3 Week

Ice Breaking

Ice Breaking

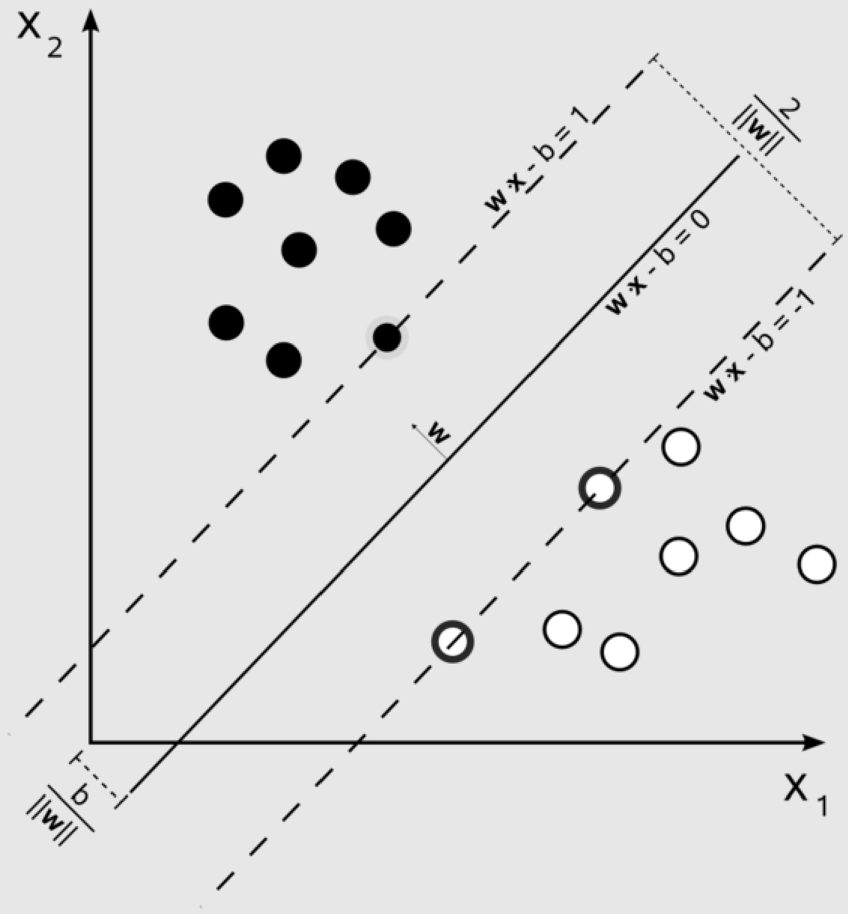


Before Start



Support Vector Machine

Support Vector Machine

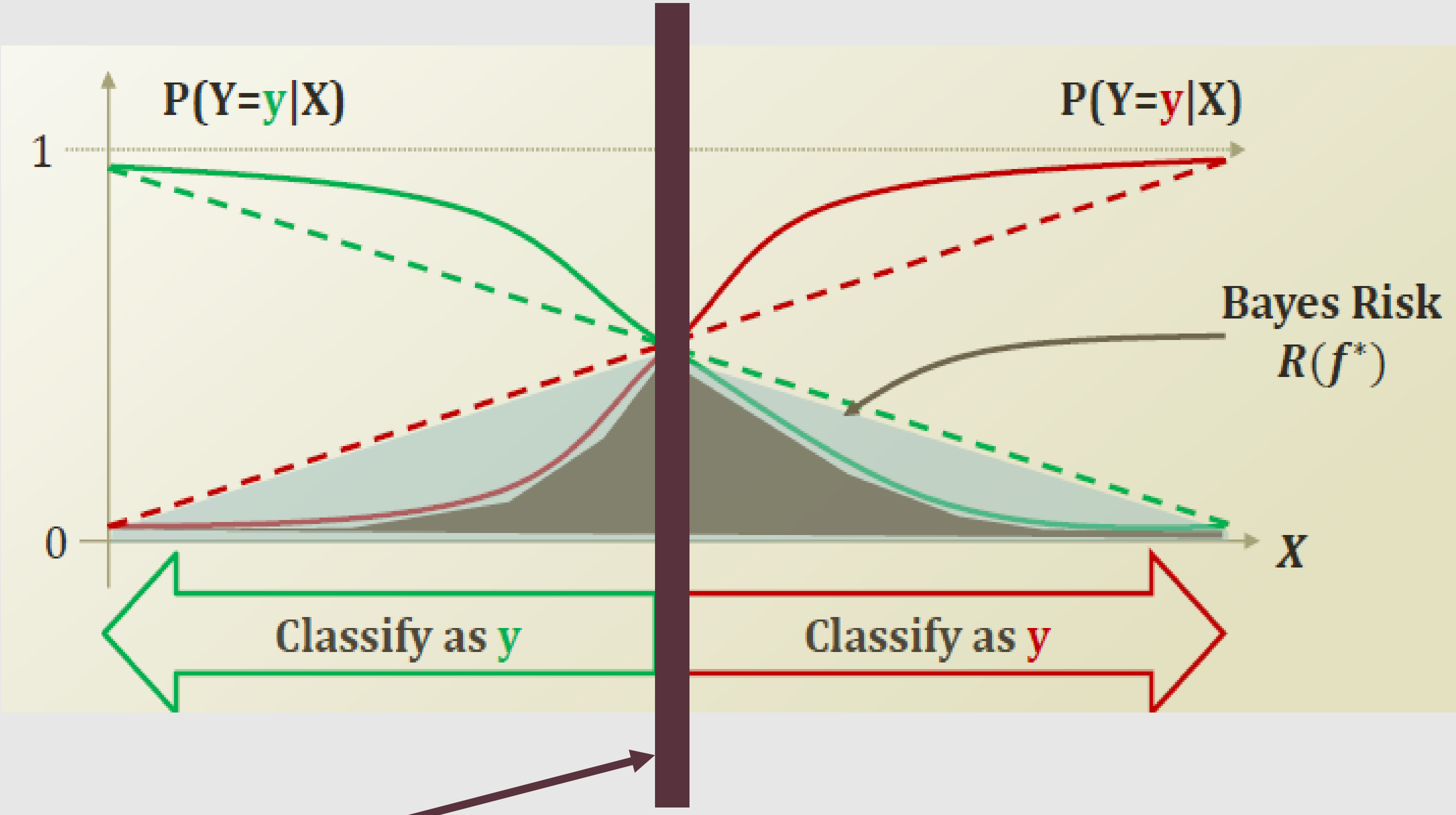


SVM?

History

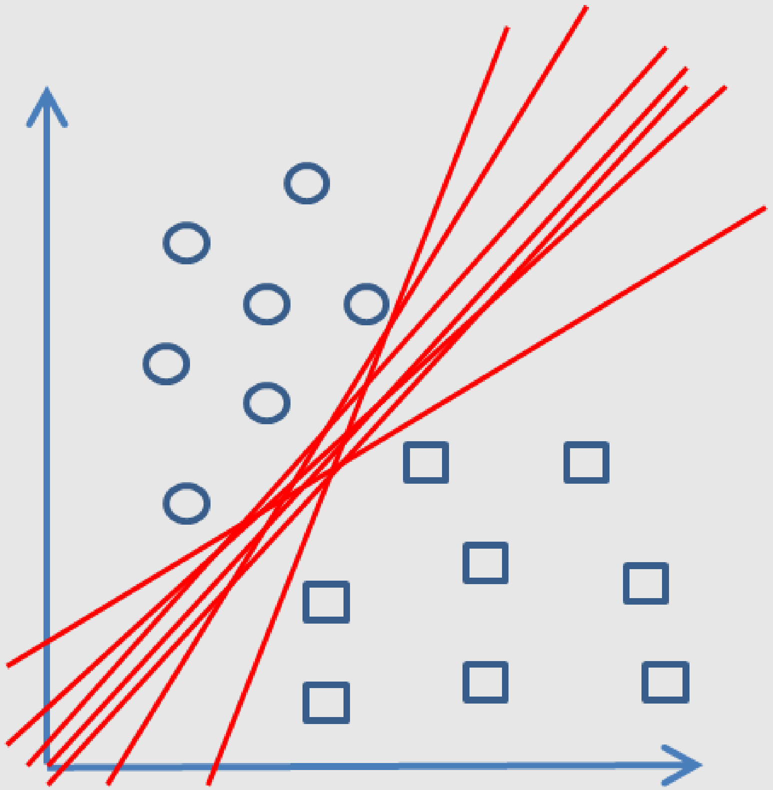
뉴럴넷의 몰락

Decision Boundary

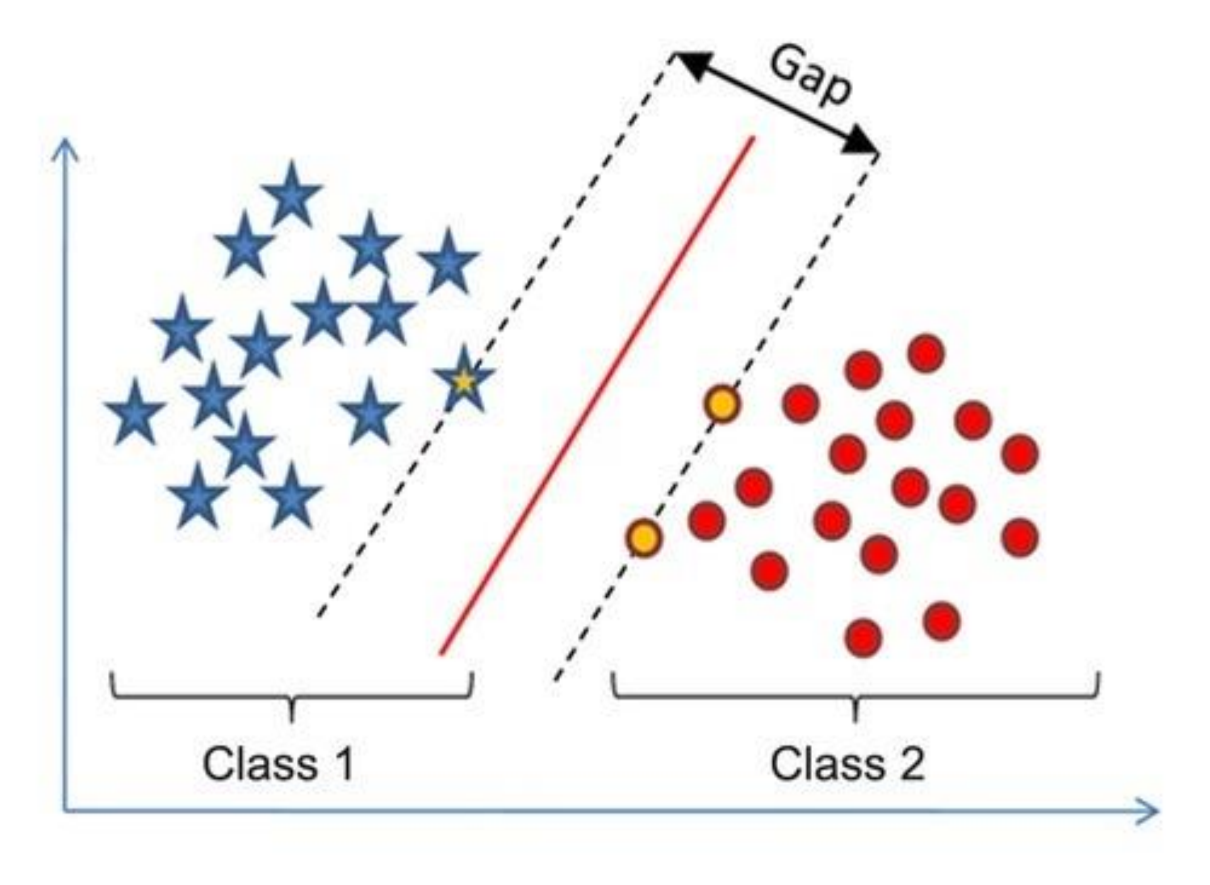


Decision Boundary

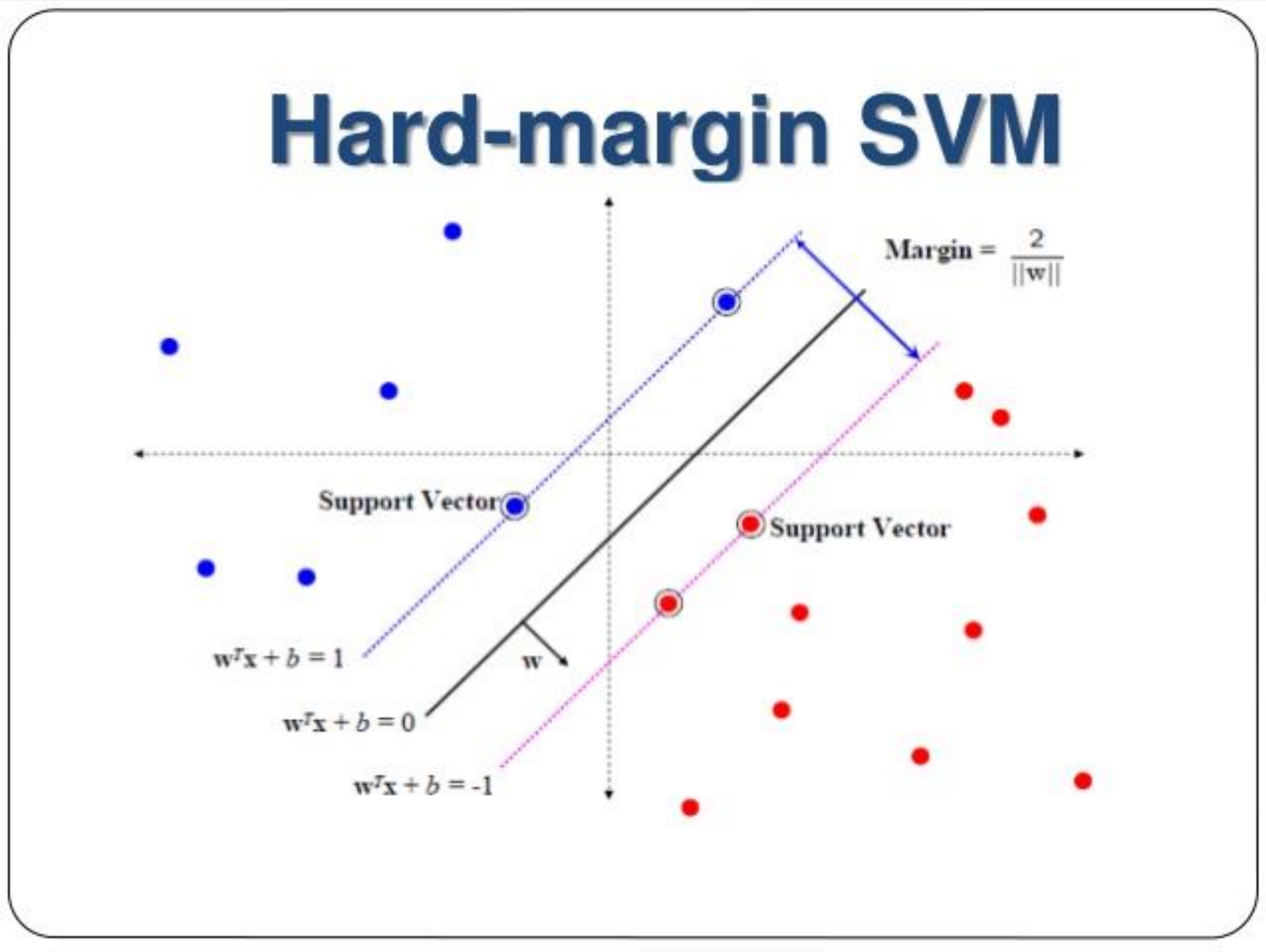
Decision Boundary



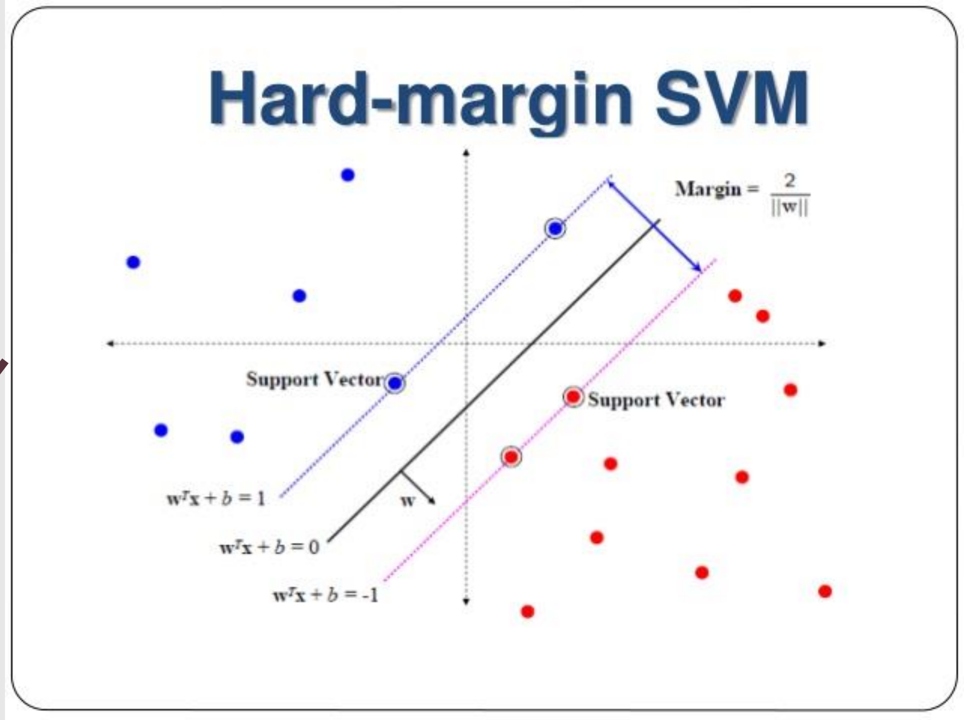
Decision Boundary



Decision Boundary



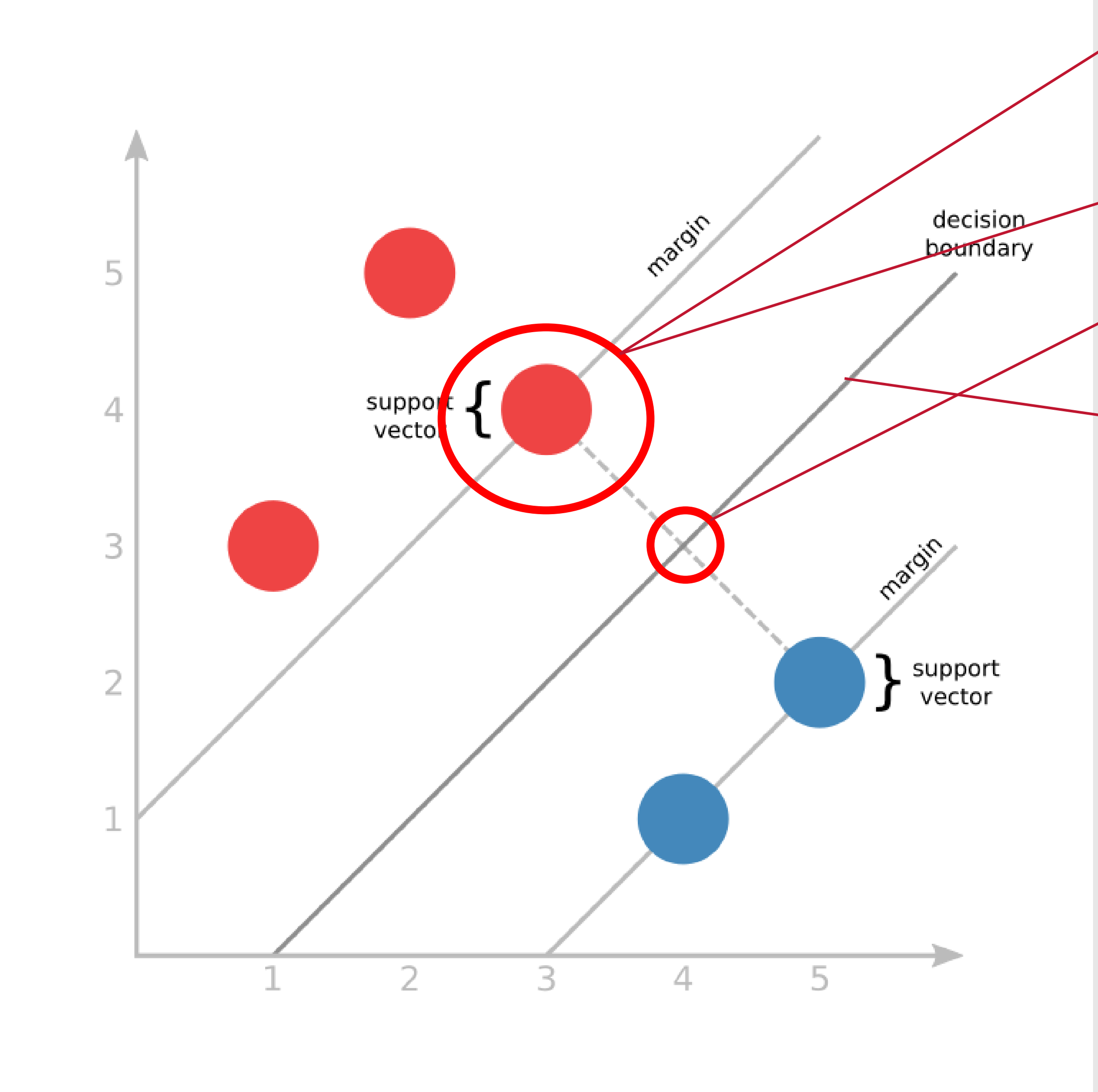
Maximizing the Margin



Let's say

- $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$
- A point $\mathbf{x}$ on the boundary has
  - $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b = 0$
- A positive point $\mathbf{x}$ has
  - $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b = a, a > 0$

Maximizing the Margin



직각이라고 할 때

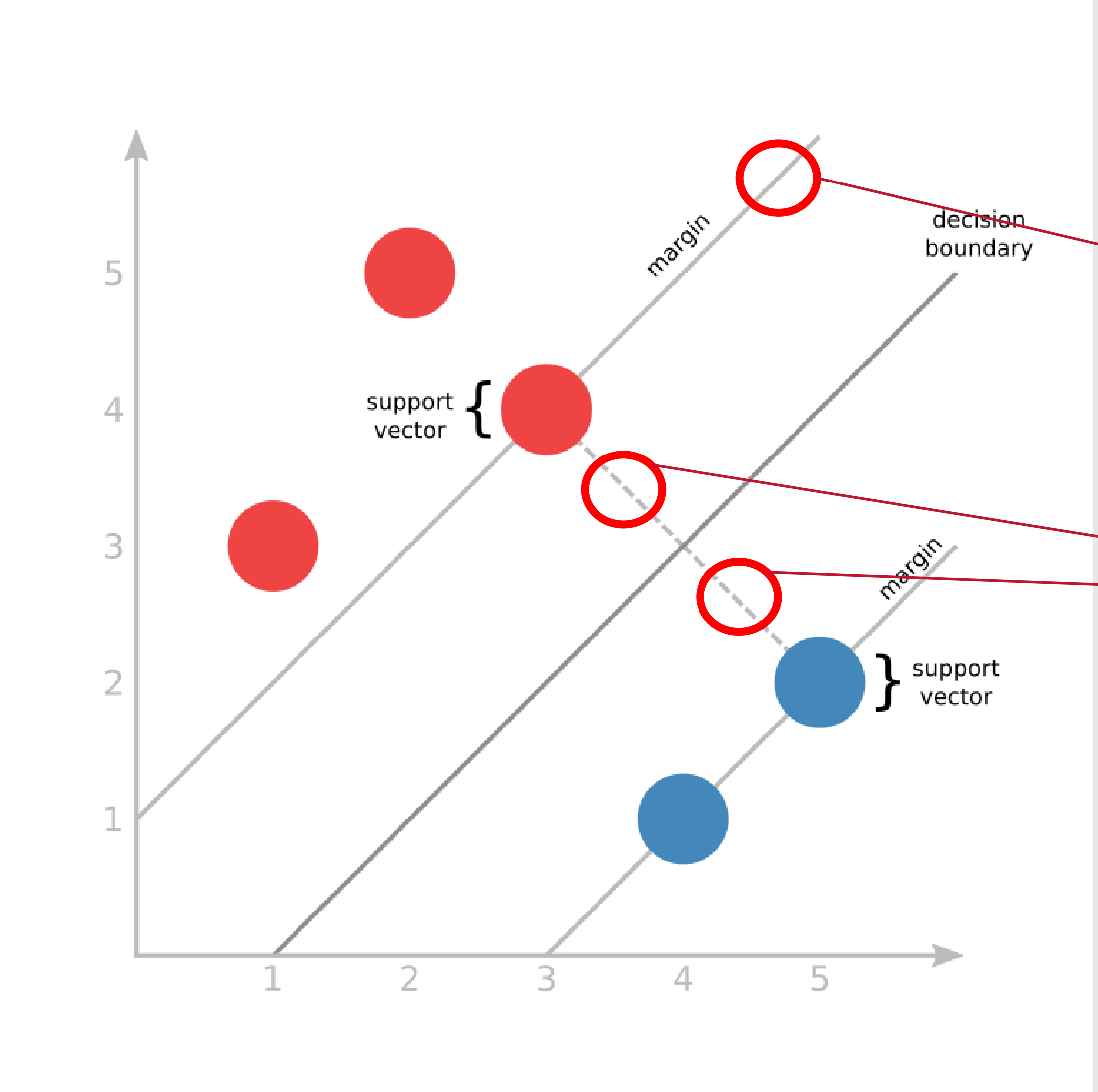
$$x = x_p + r\frac{w}{||w||}, f(x_p) = 0$$

$$f(x) = w \cdot x + b = w\left(x_p + r\frac{w}{||w||}\right) + b$$

$$= wx_p + b + r\frac{w \cdot w}{||w||} = r||w||$$

$$\Rightarrow f(x) = r||w|| \Rightarrow r = \frac{f(x)}{||w||} = \frac{a}{||w||}$$

Maximizing the Margin



최적화 = 최대 마진 결정 경계

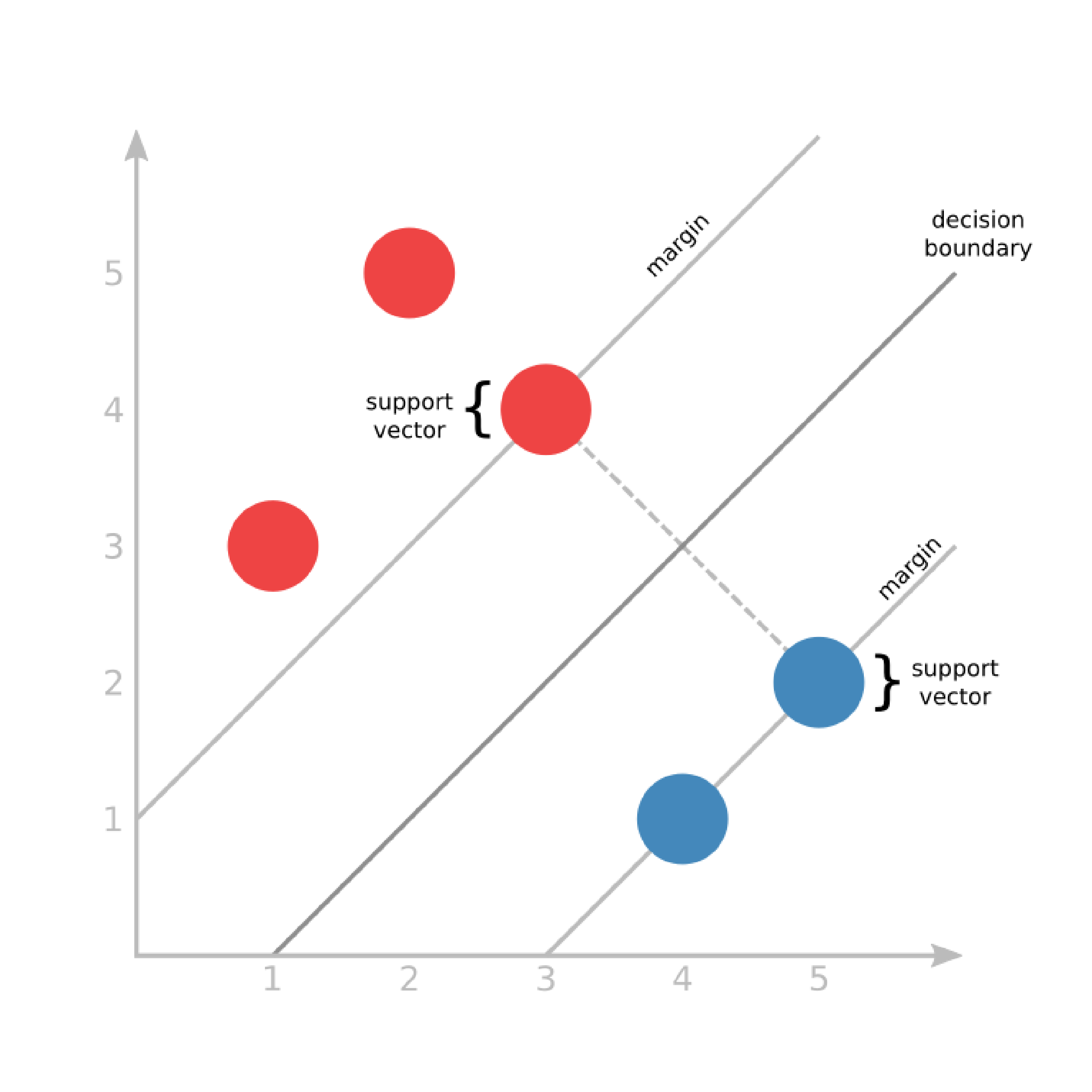
$$r = \frac{a}{||w||}$$

$$f(x) = w \cdot x + b = -a$$

$$\max_{w,b} 2r = \frac{2a}{||w||}$$

$$\Rightarrow \min_{w,b} ||w||$$

Maximizing the Margin



$$\min_{w,b} ||w||$$

$$w = \sqrt{w_1^2 + w_2^2 + \cdots}$$

w는 Quadratic form

Quadratic Programming

이차계획법 [二次計劃法 quadratic programming]

要約 :
수리계획법의 일종으로 선형등식 또는 선형부등식으로 주어진 제약 아래에서 이차함수의 최소값 또는 최대값을 구하는 방법. 약칭 QP. 회귀분석·포트폴리오분석과 같이 원래 목적함수가 이차식인 경우뿐만 아니라 이차식 이외의 비선형 목적함수를 이차식에 근사시키는 경우에도 적용된다.

詳細說明 :
수리계획법의 일종으로 선형등식 또는 선형부등식으로 주어진 제약 아래에서 이차함수의 최소값 또는 최대값을 구하는 방법. 약칭 QP. 회귀분석·포트폴리오분석과 같이 원래 목적함수가 이차식인 경우뿐만 아니라 이차식 이외의 비선형 목적함수를 이차식에 근사시키는 경우에도 적용된다.

[최적조건]
실행가능해(實行可能解)의 영역은 선형계획법의 경우와 마찬가지로 볼록다면체가 되지만, 최적해(最適解)가 꼭지점 중에만 존재한다고는 할 수 없으며 꼭지점 이외의 경계상에 존재하거나 볼록다면체 내부에 존재하는 경우도 있다. 실행가능해가 최적이기 위한 필요조건으로는 선형등식, 선형부등식 이외에 두 변수의 곱이 0(즉 두 변수 중 적어도 한 쪽이 0이다)인 조건이 첨가된다. 이것은 볼록인 이차식을 최소로 하는 경우(또는 오목인 이차식을 최대로 하는 경우)에는 충분조건이 된다. P. 울프·G.B. 단치히·E.M. 빌 등의 많은 연구자에 의해 해법이 개발되었는데 선형계획법의 심플렉스법(單體法)을 변형시킨 것이 많다.

<https://blog.naver.com/wono77/140041709970>

주어진 어떤 방정식의 조건을 만족하면서, 선형인 목적함수를 최적화 해내는 방법

Implementation

```
from sklearn import svm

svm_a = svm.SVC(kernel = "linear", probability = True, C = 1.0)
svm_a.fit(train_x, train_y)
```

```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='auto', kernel='linear',
  max_iter=-1, probability=True, random_state=None, shrinking=True,
  tol=0.001, verbose=False)
```

```
svm_a.score(train_x, train_y)
```

```
0.9722650231124808
```

```
svm_a.score(test_x, test_y)
```

```
0.9861111111111112
```

```
svm_a.predict(test_x)
```

```
array([False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False,  True, False, False, False, False, False, False,
       False, False, False,  True,  True,  True,  True,  True,  True])
```

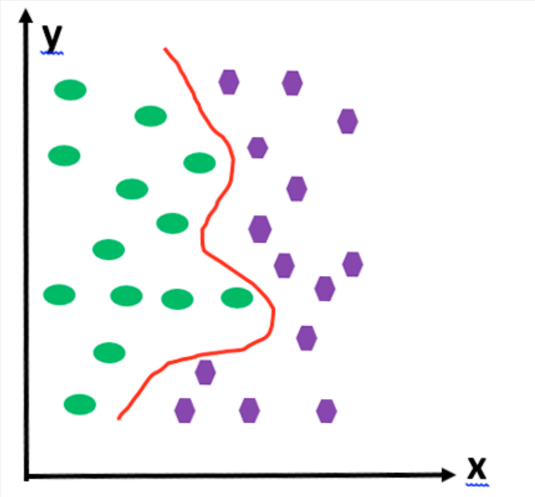
Implementation

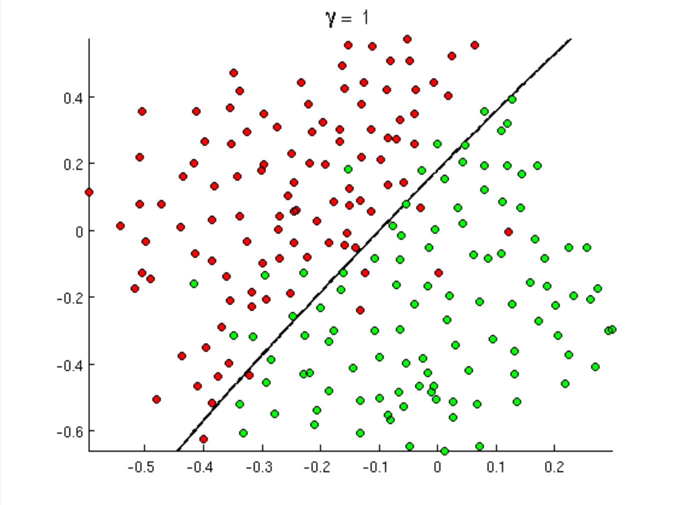
```
dat1.iloc[705]
```

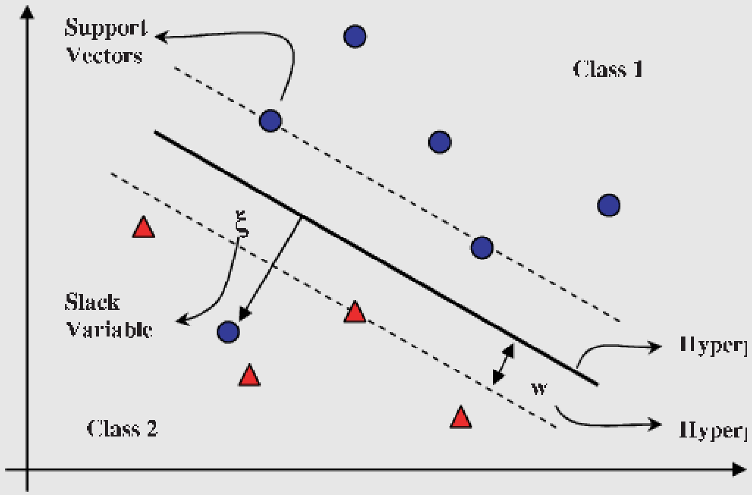
```
Number                        706
Name                       Goodra
Type_1                     Dragon
Type_2                        NaN
Total                         600
HP                             90
Attack                        100
Defense                        70
Sp_Atk                        110
Sp_Def                        150
Speed                          80
Generation                      6
isLegendary                 False
Color                      Purple
hasGender                    True
Pr_Male                       0.5
Egg_Group_1                Dragon
Egg_Group_2                   NaN
hasMegaEvolution            False
Height_m                     2.01
Weight_kg                   150.5
Catch_Rate                     45
Body_Style         bipedal_tailed
Name: 705, dtype: object
```

미끄래곤

Error Handling





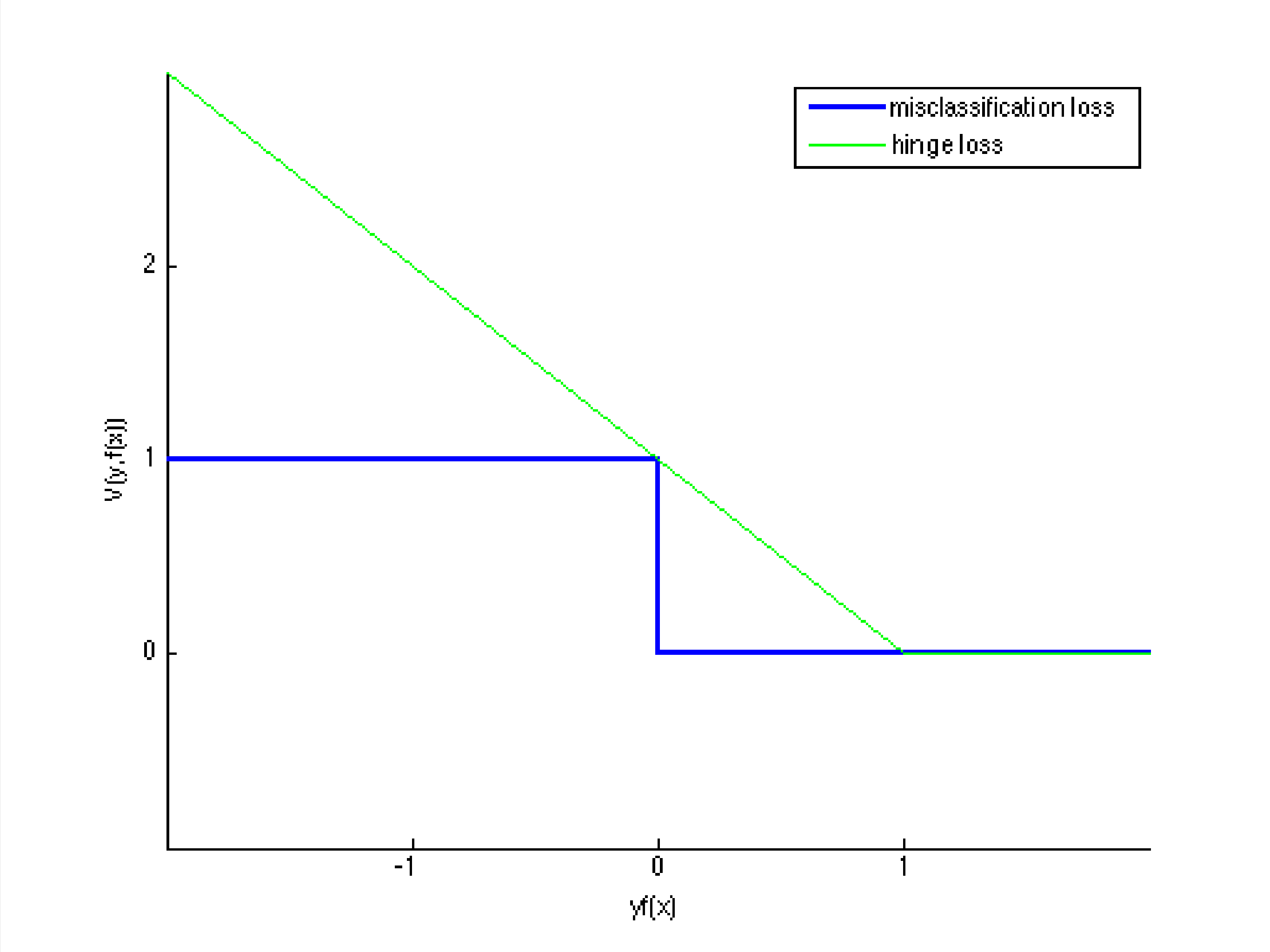


곡선

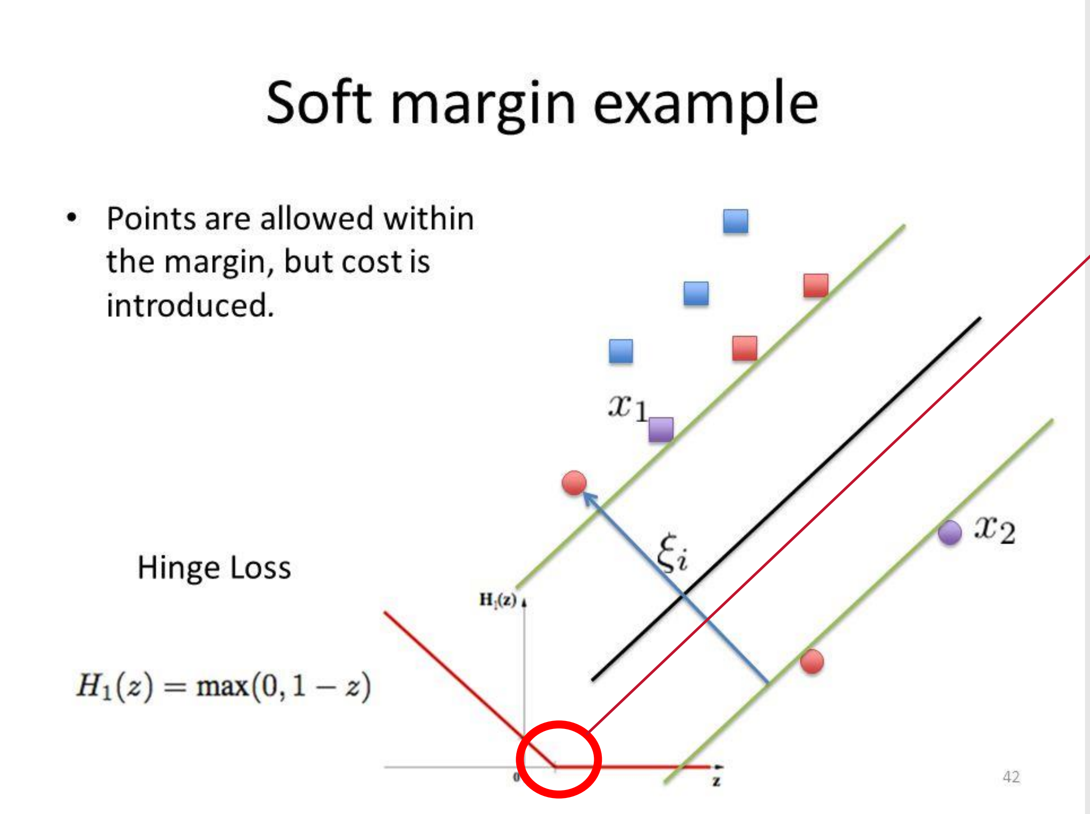
무시 — 페널티

사실상 같은 말

0-1 loss function



Hinge loss function



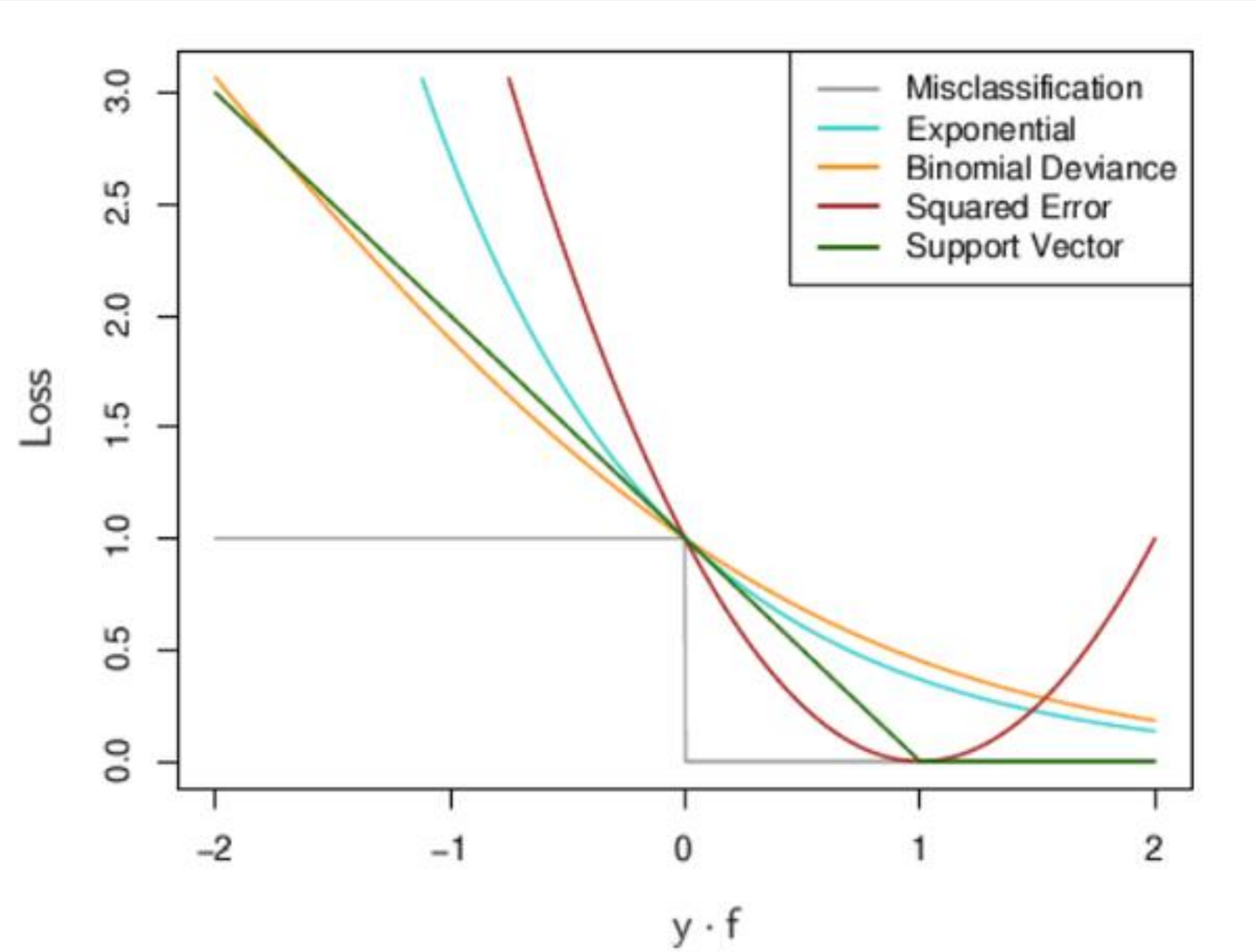
결정 경계 넘어섰을 때

Hinge loss function

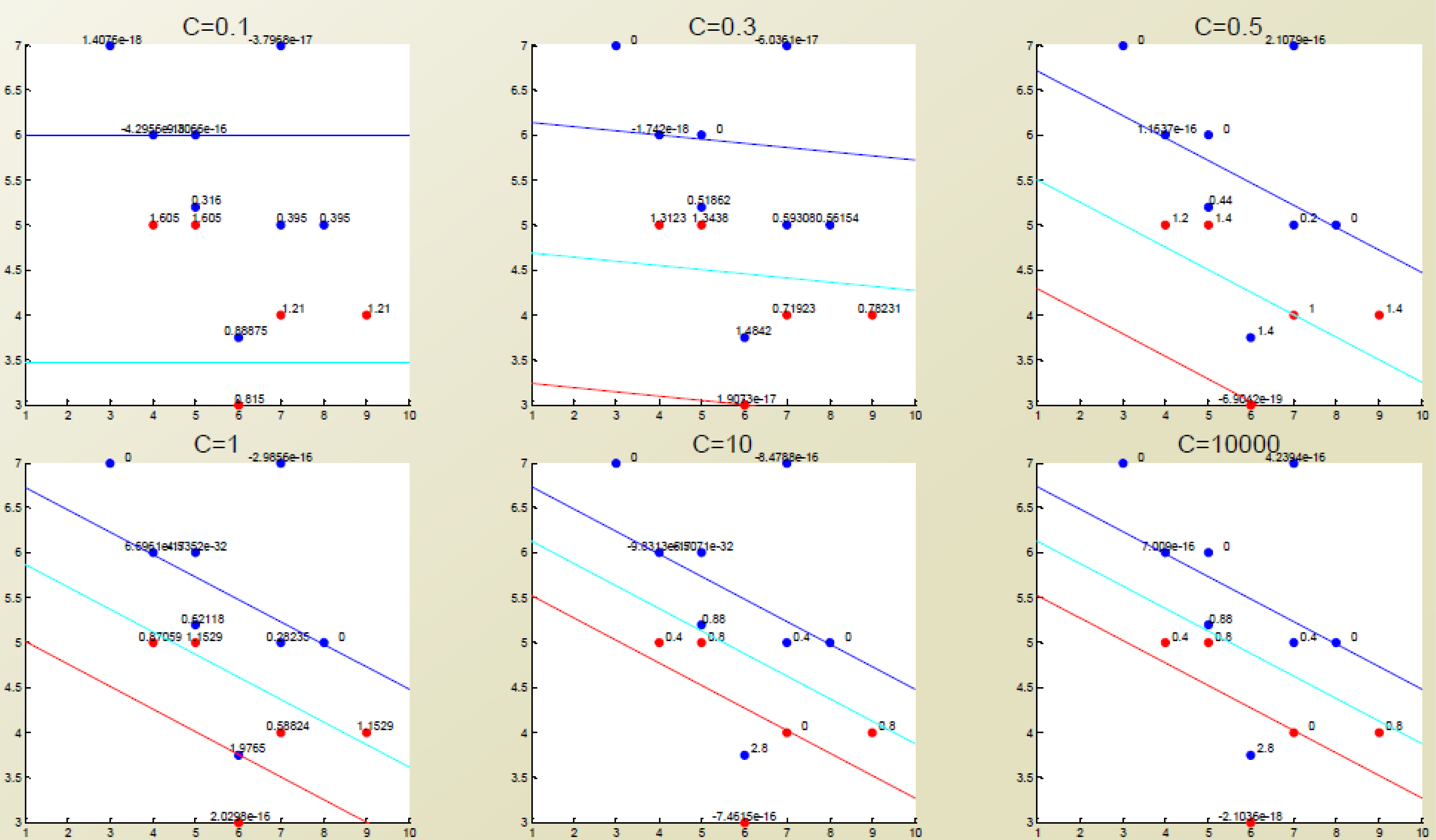
$$\min_{w,b} ||w|| + C \sum_j \xi_j$$

$$(wx_j + b)y_j \geq a - \xi_j$$

Comparison loss function



Comparison C value



Comparison C value

```
from sklearn import svm

svm_a = svm.SVC(kernel = "linear", probability = True, C = 10000.0)
svm_a.fit(train_x, train_y)
```

```
SVC(C=10000.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='auto', kernel='linear',
  max_iter=-1, probability=True, random_state=None, shrinking=True,
  tol=0.001, verbose=False)
```

```
svm_a.score(train_x, train_y)
```

```
0.9738058551617874
```

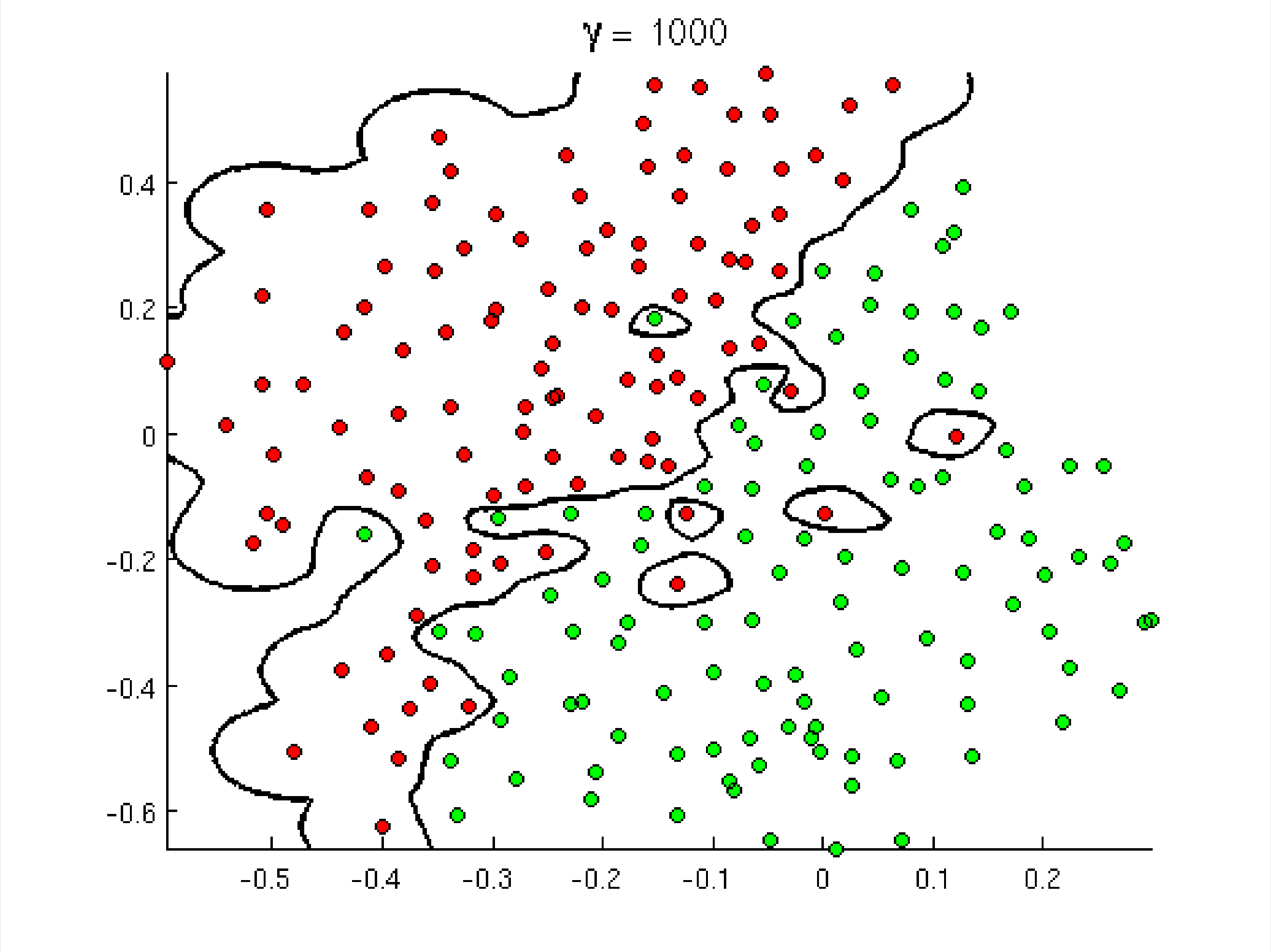
```
svm_a.score(test_x, test_y)
```

```
0.9583333333333334
```

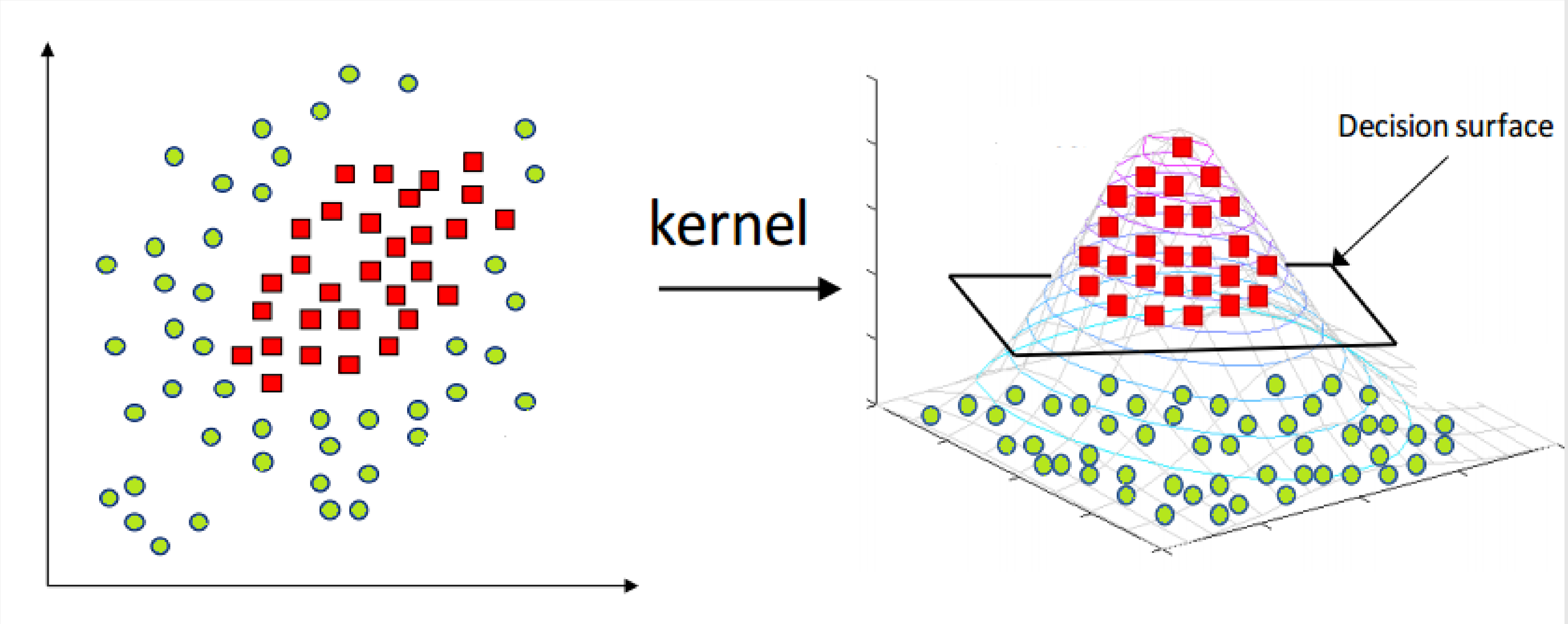
```
svm_a.predict(test_x)
```

```
array([False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False,  True, False, False, False, False, False, False,
       False, False, False,  True,  True,  True,  True, False, False])
```

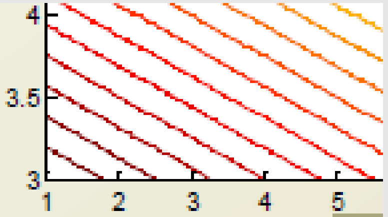
Kernel Trick



Kernel Trick



Kernel Trick



$$min_{w,b,\xi_j} ||w|| + C\sum_j \xi_j$$

$$s.t.$$

$$(w\varphi(x_j) + b)y_j \geq 1 - \xi_j, \forall j$$

$$\xi_j \geq 0, \forall j$$

$$\varphi(< x_1, x_2 >) =$$

$$< x_1, x_2, {x_1}^2, {x_2}^2, x_1x_2, {x_1}^3, {x_2}^3, {x_1}^2x_2, x_1{x_2}^2 >$$

Lagrange function

주어진 예산제약하에서 효용극대화 문제를 풀 때 주로 사용

- Lagrange Prime Function: $L(x,\alpha,\beta) = f(x) + \alpha g(x) + \beta h(x)$
- Lagrange Multiplier: $\alpha \geq 0, \beta$
- Lagrange Dual Function: $d(\alpha,\beta) = inf_{x \in X} L(x,\alpha,\beta) = min_x L(x,\alpha,\beta)$
- $max_{\alpha \geq 0, \beta} L(x,\alpha,\beta) = \begin{cases} f(x): if\ x\ is\ feasible \\ \infty: otherwise \end{cases}$
- $min_x f(x) \rightarrow min_x max_{\alpha \geq 0, \beta} L(x,\alpha,\beta)$

Lagrange function proof

증명 [편집]

1차원 오일러-라그랑주 방정식 유도는 수학에서 고전으로 꼽힌다. 이 증명의 근거는 변분법의 기본정리이다.

함수 f 가, 경계값 조건 $f(a) = c$, $f(b) = d$를 만족하고, 다음과 같이 주어지는 범함수 J를 최대 또는 최소로 만든다고 하자.

$$J = \int_a^b F(x, f(x), f'(x))\, dx.$$

여기서 F가 연속적인 편미분값을 가진다고 가정한다. (가정을 더 약하게 잡을 수도 있으나, 그러면 증명이 더 복잡해진다.)

만일 f가 상대 범함수를 최대, 최소로 한다고 하면, f에 매우 작은 변화를 가했을 때, J의 값이 늘거나(*f가 J를 최소화할때)* J의 값이 줄 수 있다.(f가 J를 최대화할때)

여기서 f에 매우 작은 변화를 준 함수 $g_\varepsilon(x) = f(x) + \varepsilon\eta(x)$ 를 도입하자. 여기서 $\eta(x)$ 는 $\eta(a) = \eta(b) = 0$ 를 만족하는 미분가능한 함수이다. 이제, f대신 g 를 넣은 J는 다음과 같은 함수가 될 것이다.

$$J(\epsilon) = \int_a^b F(x, g_\epsilon(x), g'_\epsilon(x))\, dx.$$

이제 J를 ε 에 대해 미분한 전미분 을 구하면,

$$\frac{\mathrm{d}J}{\mathrm{d}\varepsilon} = \int_a^b \frac{\mathrm{d}F}{\mathrm{d}\epsilon}(x, g_\varepsilon(x), g'_\varepsilon(x))\, dx.$$

전미분의 정의에서 다음과 같은 식이 나오며,

$$\frac{\mathrm{d}F}{\mathrm{d}\epsilon} = \frac{\partial F}{\partial x}\frac{\partial x}{\partial \varepsilon} + \frac{\partial F}{\partial g_\varepsilon}\frac{\partial g_\varepsilon}{\partial \varepsilon} + \frac{\partial F}{\partial g'_\varepsilon}\frac{\partial g'_\varepsilon}{\partial \varepsilon} = \eta(x)\frac{\partial F}{\partial g_\varepsilon} + \eta'(x)\frac{\partial F}{\partial g'_\varepsilon}.$$

그러므로

$$\frac{\mathrm{d}J}{\mathrm{d}\epsilon} = \int_a^b \left[\eta(x)\frac{\partial F}{\partial g_\varepsilon} + \eta'(x)\frac{\partial F}{\partial g'_\varepsilon}\right] dx.$$

만약 $\varepsilon = 0$ 이 되면 $g_\varepsilon = f$이고, f가 J를 극값으로 만드는 부분이므로, $J'(0) = 0$, 일 것이다. 수식으로 쓰면,

$$J'(0) = \int_a^b \left[\eta(x)\frac{\partial F}{\partial f} + \eta'(x)\frac{\partial F}{\partial f'}\right] dx = 0.$$

좀 더 정리하기 위해, 두 번째 항에 부분적분을 한다. 그러면 다음과 같은 식을 얻는다.

$$0 = \int_a^b \left[\frac{\partial F}{\partial f} - \frac{d}{dx}\frac{\partial F}{\partial f'}\right]\eta(x)\, dx + \left[\eta(x)\frac{\partial F}{\partial f'}\right]_a^b.$$

η에 대한 경계값 조건을 이용하면,

$$0 = \int_a^b \left[\frac{\partial F}{\partial f} - \frac{d}{dx}\frac{\partial F}{\partial f'}\right]\eta(x)\, dx.$$

변분법의 기본정리 를 적용하면, 다음과 같은 오일러-라그랑주 방정식을 얻는다.

$$0 = \frac{\partial F}{\partial f} - \frac{d}{dx}\frac{\partial F}{\partial f'}.$$

Lagrange Function Example

소비자 A의 효용암수는 $U(X,Y) = xy$일 때 (둘 다 최대한 많이 사야 할 때),
그가 가진 돈이 90원이고 사과(x)는 3원, 배(y)는 5원이라고 하자.

이 상황을 식으로 나타내면

$$3x + 5y = 90$$

즉, $\max U = xy \;\; s.t, 3x + 5y = 90$

이제 이 식을 어떻게 풀 것인가?

Lagrange Function Example

그냥 풀기엔 어렵기 때문에 라그랑지 함수로 나타내면

$$L = xy + \lambda(90 - 3x - 5y)$$

$$Lx = y - 3\lambda = 0 \Rightarrow y = 3\lambda \cdots (a)$$

$$Ly = x - 5\lambda = 0 \Rightarrow x = 5\lambda \cdots (b)$$

λ에 대하여 미분하면 $90 - 3x - 5y = 0$

(a), (b)식을 람다에 대하여 미분한 식에 넣어주면 $\lambda = 3$

$\lambda = 3$를 (a),(b)식에 넣으면 x=15, y=9

➔ 사과 15개와 배9개를 소비할 때 효용이 극대화된다.

Lagrange function

Primal Problem	Lagrange Dual Problem
$min_x f(x)$ $s.t.\ \ g(x) \leq 0, h(x) = 0$ ↓ $min_x max_{\alpha \geq 0, \beta} L(x, \alpha, \beta)$	$max_{\alpha > 0, \beta} d(\alpha, \beta)$ $s.t. \alpha > 0$ ↓ $max_{\alpha \geq 0, \beta} min_x L(x, \alpha, \beta)$

- Strong duality
  - $d^* = max_{\alpha \geq 0, \beta} min_x L(x, \alpha, \beta) = min_x max_{\alpha \geq 0, \beta} L(x, \alpha, \beta) = p^*$
  - When Karush-Kunh-Tucker (KKT) Conditions are satisfied

KKT Condition

- $\nabla L(x^*, \alpha^*, \beta^*) = 0$
- $\alpha^* \geq 0$
- $g(x^*) \leq 0$
- $h(x^*) = 0$
- $\alpha^* g(x^*) = 0$

x^*, u^*, v^*가 KKT 조건을 만족

x^*는 primal 문제의 optimal value
u^*, v^*는 dual 문제의 optimal value

+ strong duality

Dual Problem of SVM

$$min_{w,b}||w||$$
$$s.t.\left(wx_j+b\right)y_j \geq 1, \forall j$$

$$min_{w,b}max_{\alpha\geq 0,\beta}\frac{1}{2}w\cdot w-\sum_j \alpha_j\left[\left(wx_j+b\right)y_j-1\right]$$
$$s.t.\,\alpha_j \geq 0, for\ \forall j$$

Du

$$max_{\alpha\geq 0}min_{w,b}\frac{1}{2}w\cdot w-\sum_j \alpha_j\left[\left(wx_j+b\right)y_j-1\right]$$
$$s.t.\,\alpha_j \geq 0, for\ \forall j$$

$$\frac{\partial L(\mathbf{w},b,\boldsymbol{\alpha})}{\partial \mathbf{w}}=0\ ,\ \frac{\partial L(\mathbf{w},b,\boldsymbol{\alpha})}{\partial b}=0$$
$$\alpha_i \geq 0, \forall i$$
$$\alpha_i\left(\left(wx_j+b\right)y_j-1\right)=0, \forall i$$

Dual Problem of SVM

$$max_{\alpha\geq 0} min_{w,b} \frac{1}{2} w \cdot w - \sum_j \alpha_j [(wx_j + b)y_j - 1]$$

$$s.t. \alpha_j \geq 0, for \ \forall j$$

$$\frac{\partial L(\mathbf{w},b,\boldsymbol{\alpha})}{\partial \mathbf{w}} = 0 \ , \ \frac{\partial L(\mathbf{w},b,\boldsymbol{\alpha})}{\partial b} = 0$$

$$\alpha_i \geq 0, \forall i$$

$$\alpha_i \left((wx_j + b)y_j - 1 \right) = 0, \forall i$$

w에 대해서 미분

$$w - \sum_j \alpha_j x_j y_j = 0 \Rightarrow w = \sum_j \alpha_j x_j y_j$$

b에 대해서 미분

$$-\sum_j \alpha_j y_j = 0 \Rightarrow \sum_j \alpha_j y_j = 0$$

Dual Problem of SVM

w에 대해서 미분

$$w - \sum_j \alpha_j x_j y_j = 0 \Rightarrow w = \sum_j \alpha_j x_j y_j$$

b에 대해서 미분

$$-\sum_j \alpha_j y_j = 0 \Rightarrow \sum_j \alpha_j y_j = 0$$

- $L(w,b,\alpha) = \frac{1}{2} w \cdot w - \sum_j \alpha_j [(wx_j + b)y_j - 1]$
  - $= \frac{1}{2} ww \ - \sum_j \alpha_j y_j w x_j - b \sum_j \alpha_j y_j + \sum_j \alpha_j$
  - $= \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i x_j - \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i x_j - b \times 0 + \sum_j \alpha_j$
  - $= \sum_j \alpha_j - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i x_j$

단순히 풀어 씀

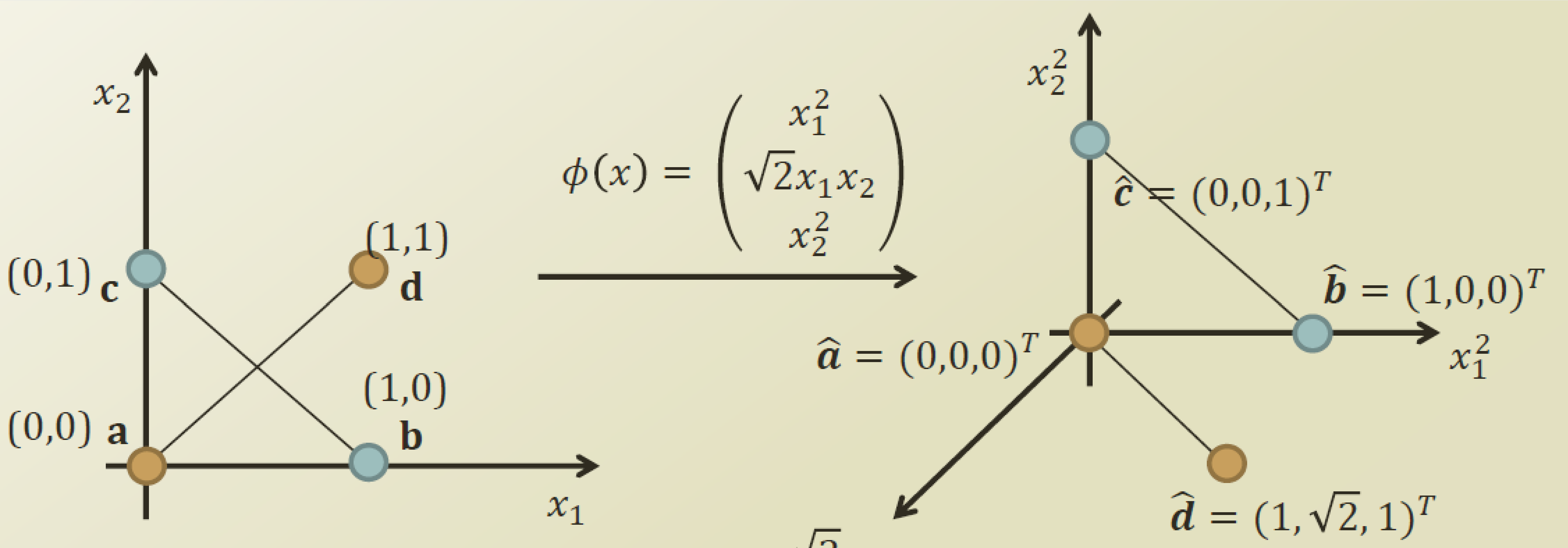
KKT조건식 대입

다시 Quadratic form

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

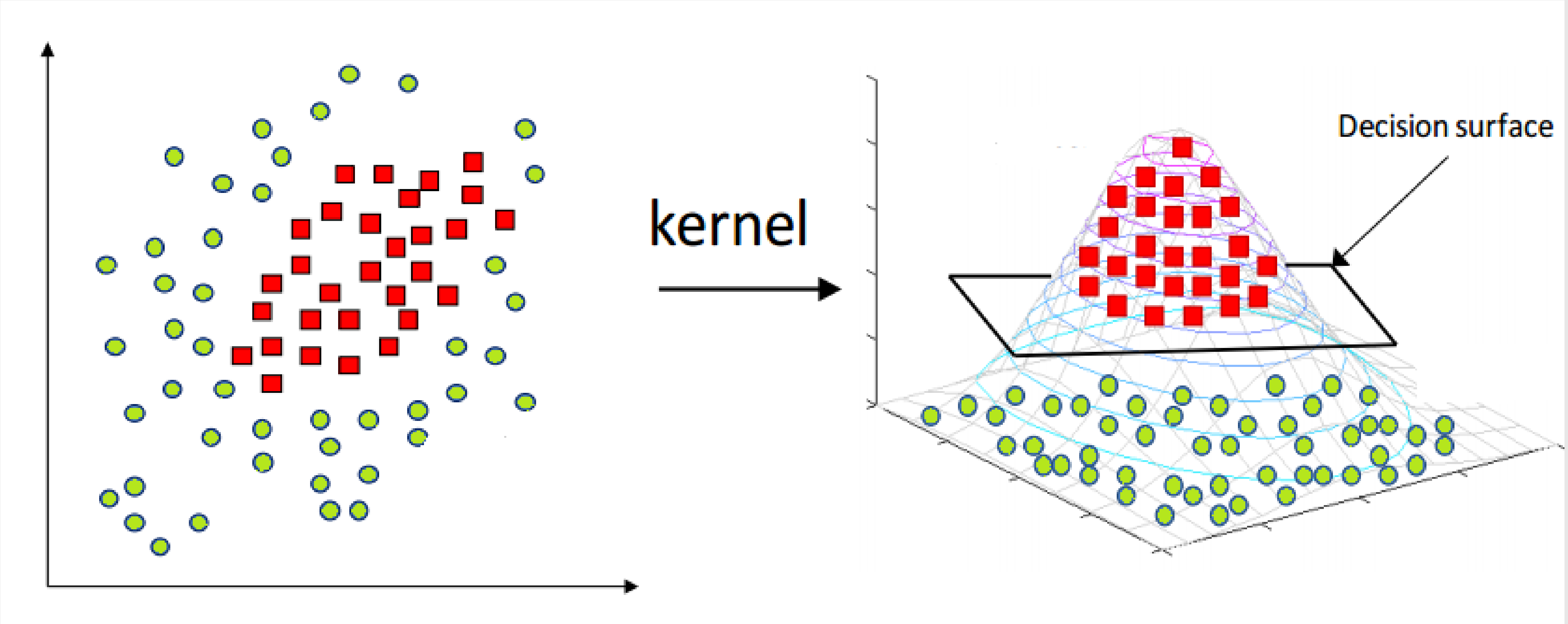
$$\alpha_i \left((wx_j + b) y_j - 1 \right) = 0$$

Kernel for XOR



(a) Original space, L (b) Mapping space, H

Kernel Example



Mercer's Theorem

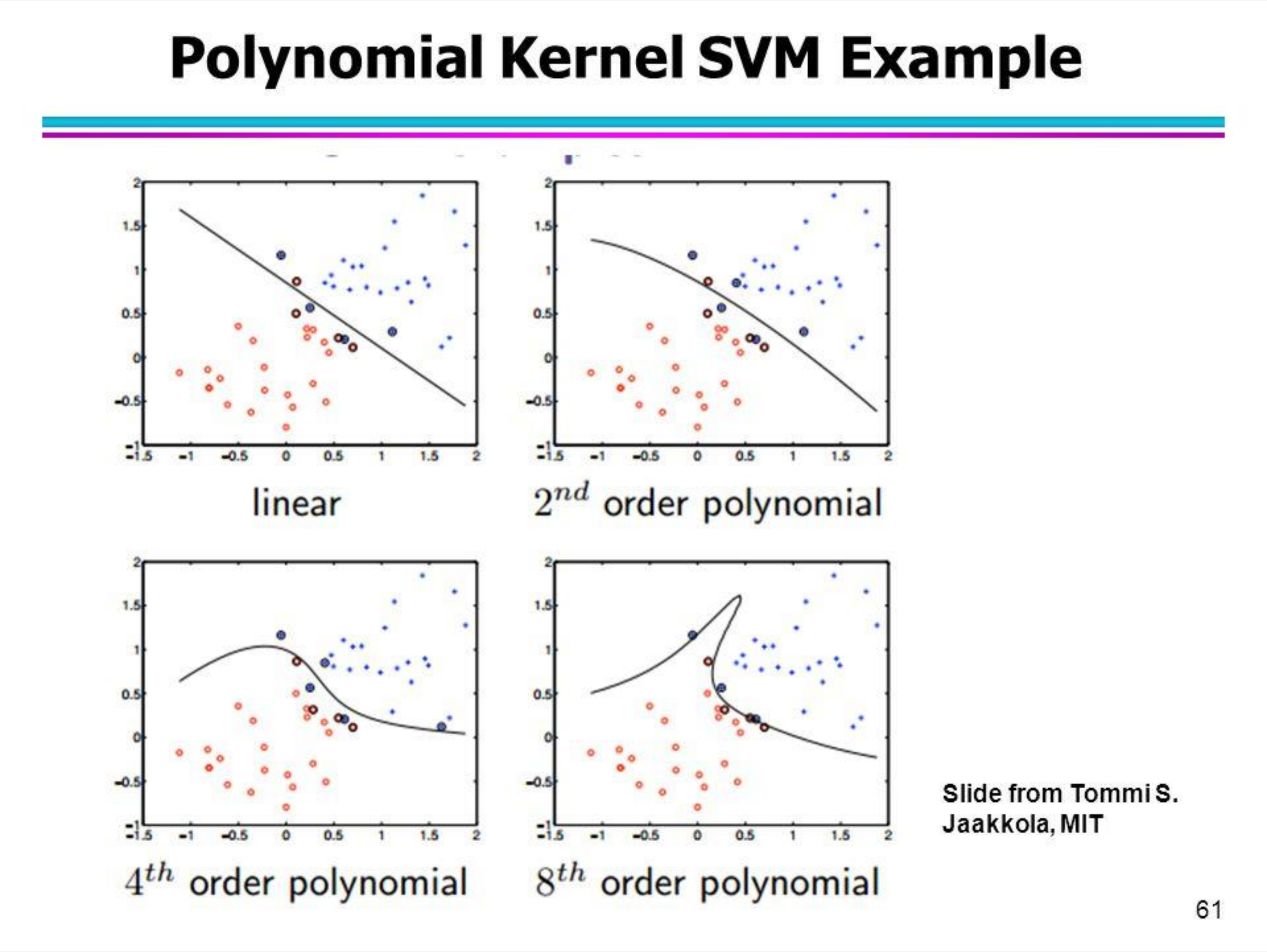
In mathematics, specifically functional analysis, Mercer's theorem is a representation of a symmetric positive-definite function on a square as a sum of a convergent sequence of product functions. This theorem, presented in (Mercer 1909), is one of the most notable results of the work of James Mercer. It is an important theoretical tool in the theory of integral equations; it is used in the Hilbert space theory of stochastic processes, for example the Karhunen–Loève theorem; and it is also used to characterize a symmetric positive semi-definite kernel.

➔확률론에서의 공간이론과 커널 특성화에 주로 사용되는 정리

$K(a,b)$가 머서의 조건(K가 매개변수에 대해 연속, 등)을 따르면 a와 b를 다른 공간에 매핑하는 $K(a,b) = \emptyset(a)^T \cdot \emptyset(b)$와 같은 함수 ∅가 존재한다.

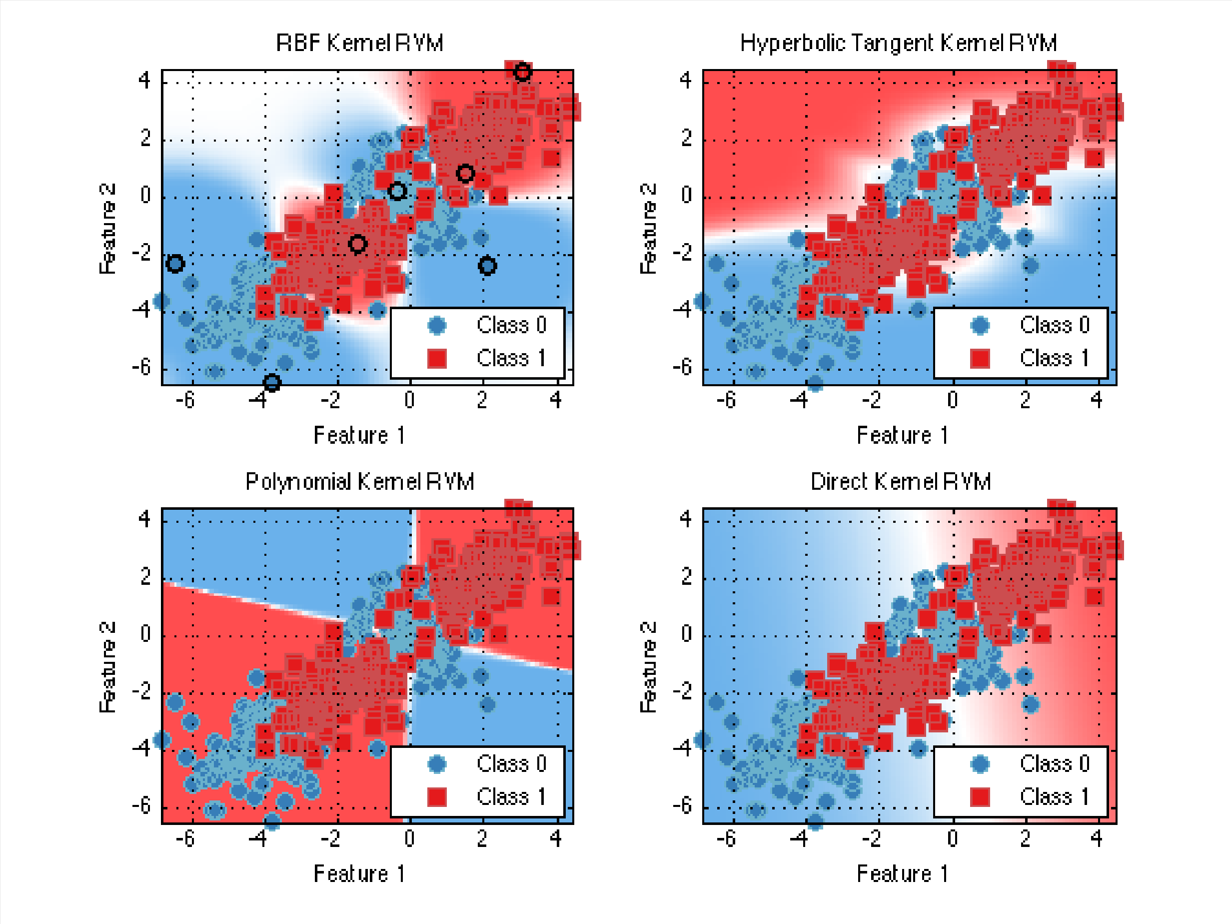
- 선형: $K(a,b) = a^T \cdot b$
- 다항식: $K(a,b) = (\gamma a^T \cdot b + r)^d$
- 가우시안RBF: $K(a,b) = \exp\left(-\gamma ||a-b||^2\right)$
- 시그모이드: $K(a,b) = \tanh(\gamma a^T \cdot b + r)$

Polynomial Kernel

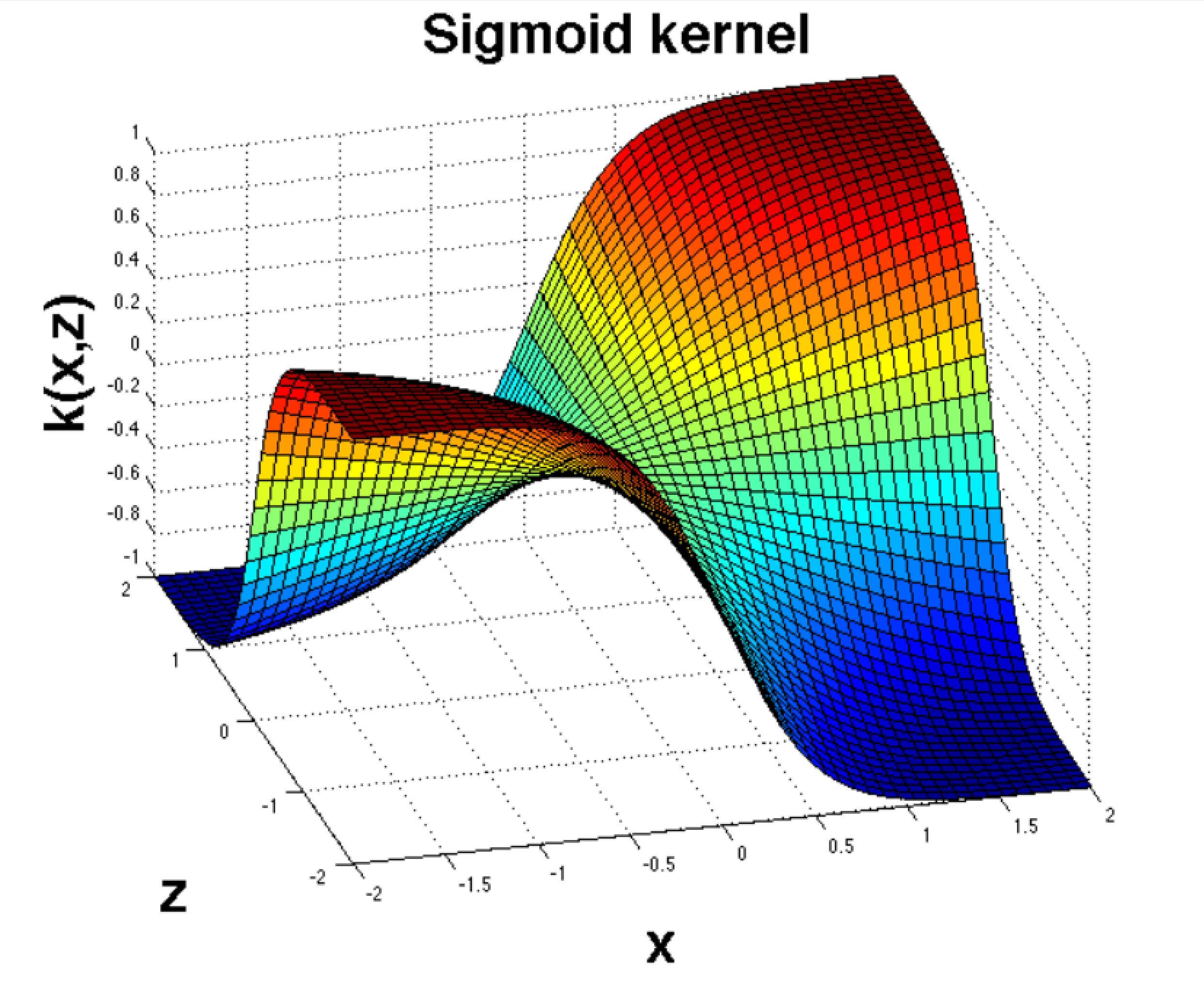


Various Kernel

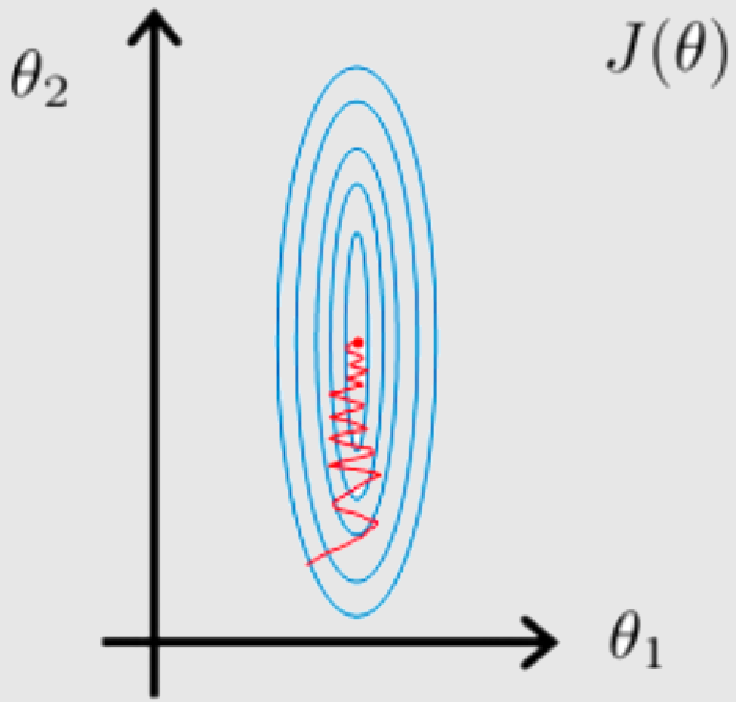
- $max_{\alpha\geq 0}\ \Sigma_j\, \alpha_j - \frac{1}{2}\Sigma_i\, \Sigma_j\ \alpha_i \alpha_j y_i y_j \varphi(x_i)\varphi(x_j)$
- $max_{\alpha\geq 0}\ \Sigma_j\, \alpha_j - \frac{1}{2}\Sigma_i\, \Sigma_j\ \alpha_i \alpha_j y_i y_j K(x_i, x_j)$



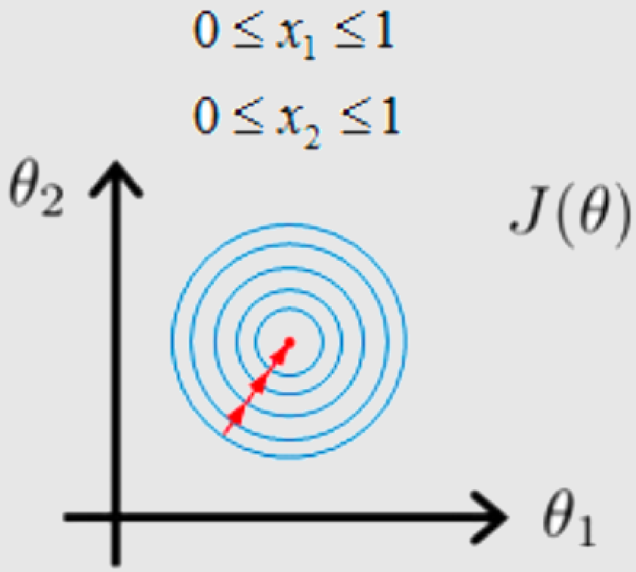
Various Kernel



Scaling

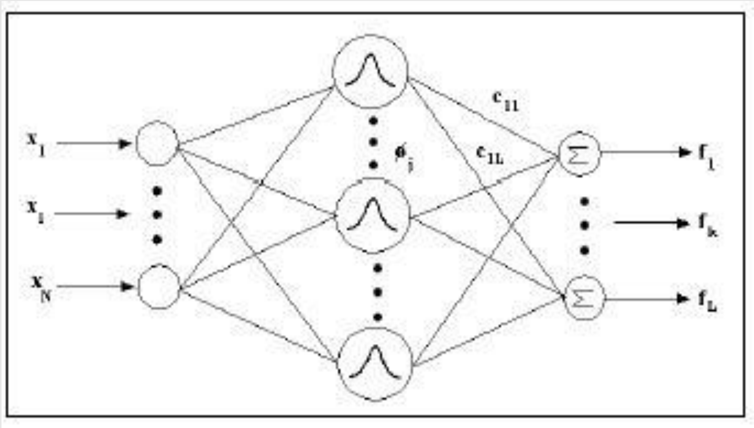


θ_2
$J(\theta)$
θ_1



$0 \leq x_1 \leq 1$
$0 \leq x_2 \leq 1$
θ_2
$J(\theta)$
θ_1

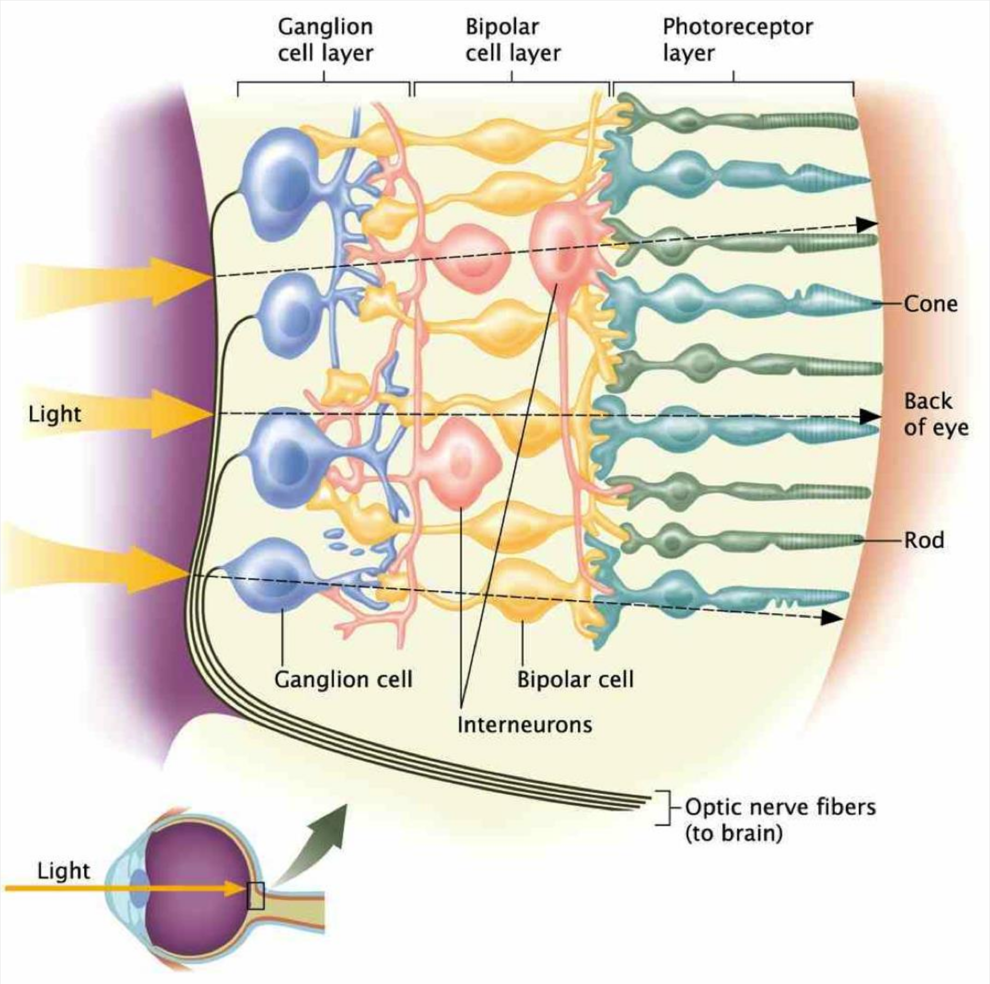
Radial Basis Function



방사형 구조를 기본으로 하는 네트워크로 1개의 은닉층에는 확률 가우시안이 적용되어 있음

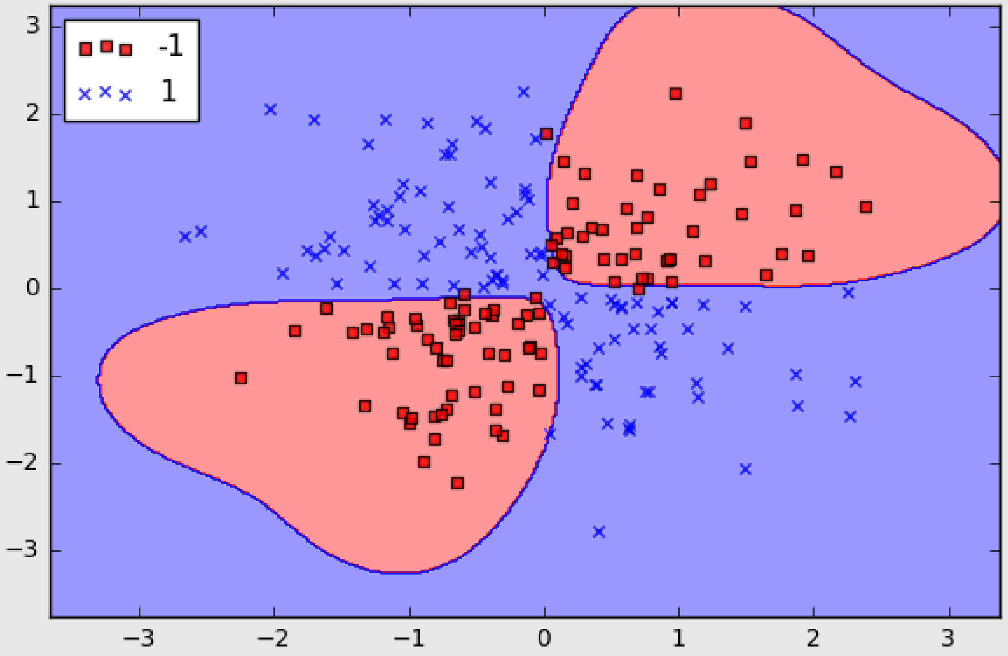
1. 은닉층이 1개
2. Euclidean (L2 Norm)
3. Back Propagation (역전파 알고리즘)
4. 안전성 판별 가능

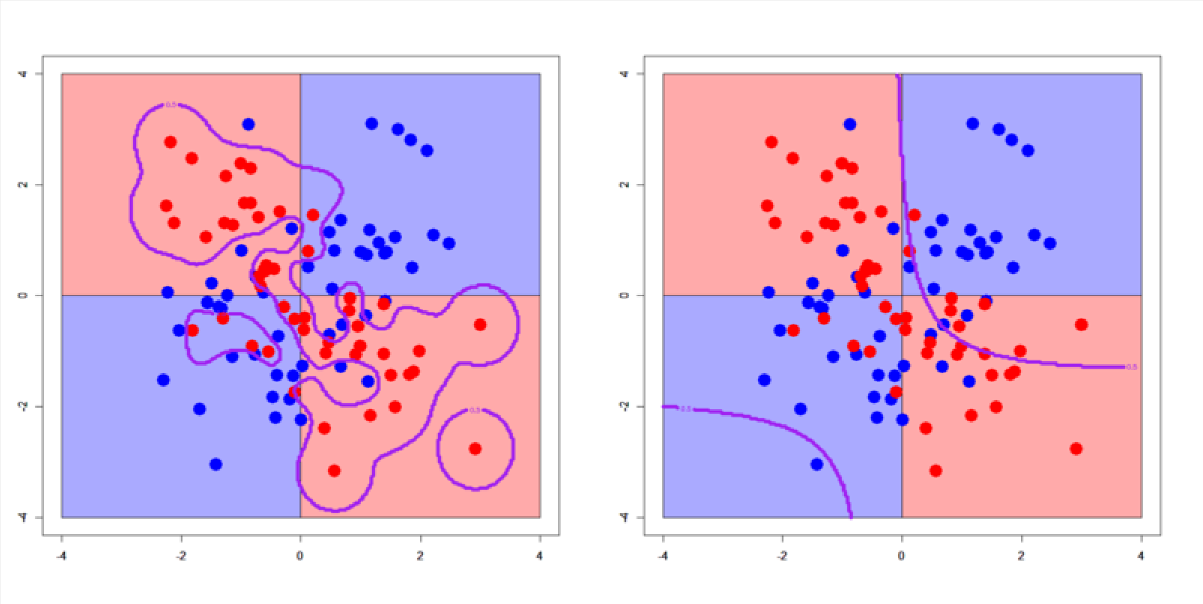
Radial Basis Function



⟨http://www.seehint.com/word.asp?no=13450⟩

Radial Basis Function Example

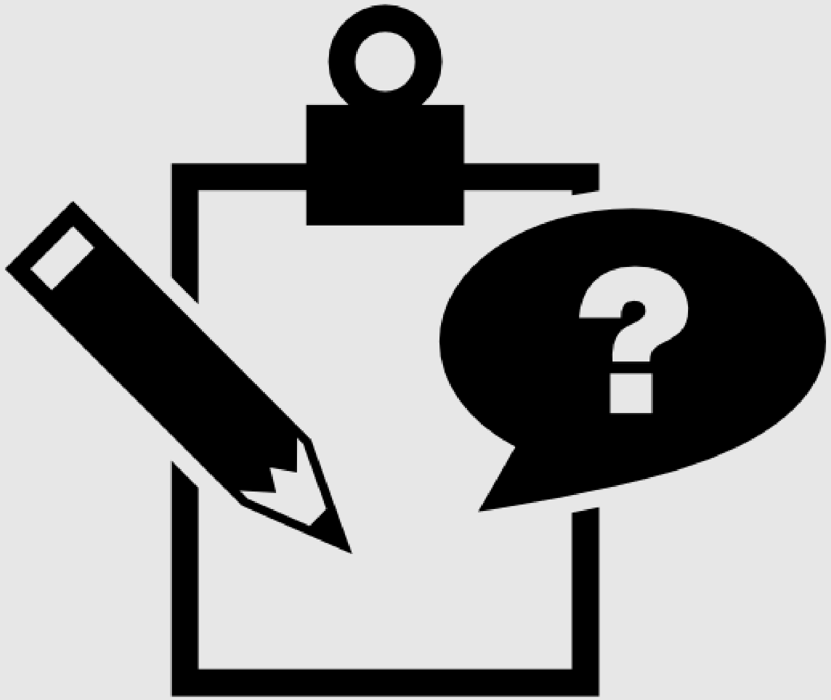




MLP vs RBF

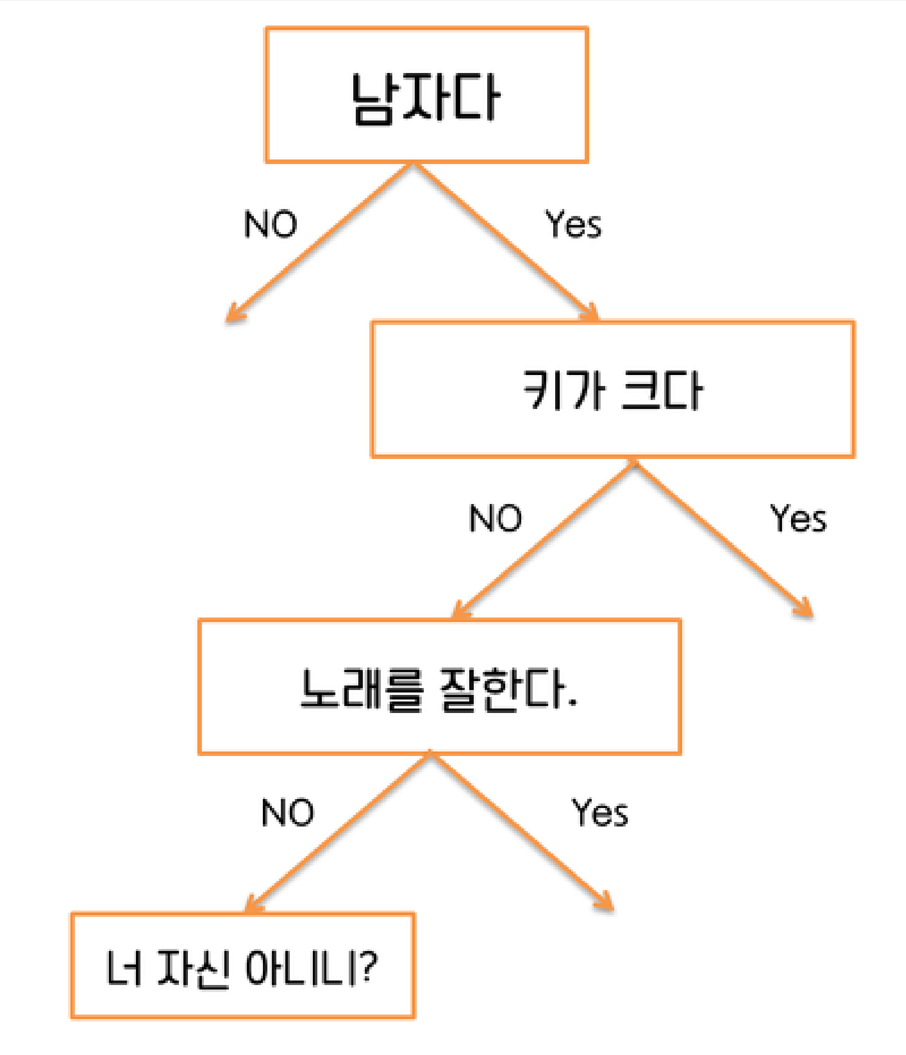
	Multi-Layer Perceptron (MLP)	Radial Basis Function (RBF)
Number of Hidden-layer	1 or more	1
Activation Function	Sigmoid or ReLU or …	Gaussian
Mathematical Analysis	Bad	Good
Output	Linear or Sigmoid	Linear
Learning Time	Slow	Fast
Data Comparison	Vector Product	Euclidean Distance

〈http://happycontrol.tistory.com/entry/RBFRadial-Basis-Function-%EC%8B%A0%EA%B2%BD%EB%A7%9D-1〉

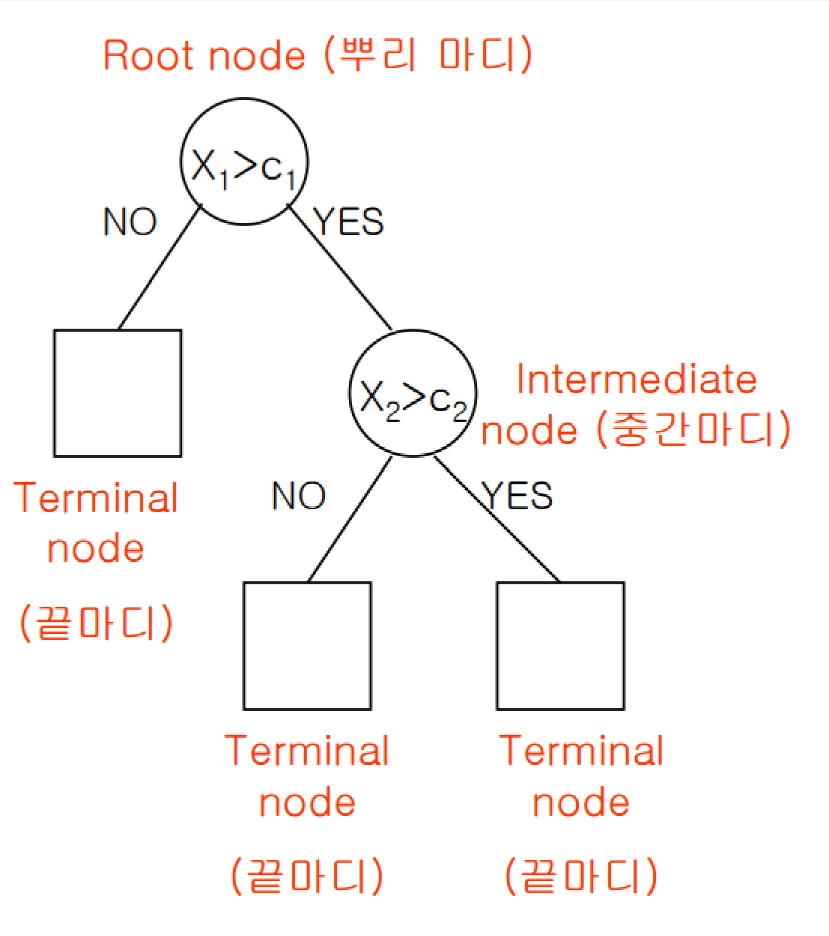


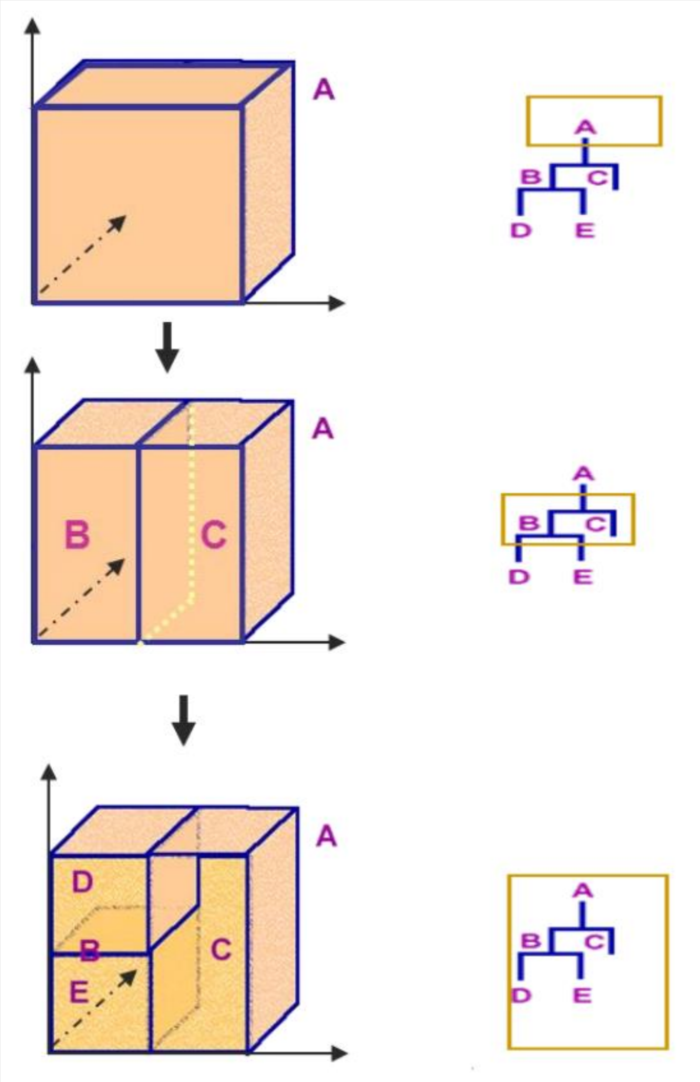
Decision Tree

Decision Tree Guide



Decision Tree Guide





Method

Information Gain (Entropy)

$$I_E(f) = -\sum_{i=1}^{m} f_i \log_2 f_i$$

Gini Index

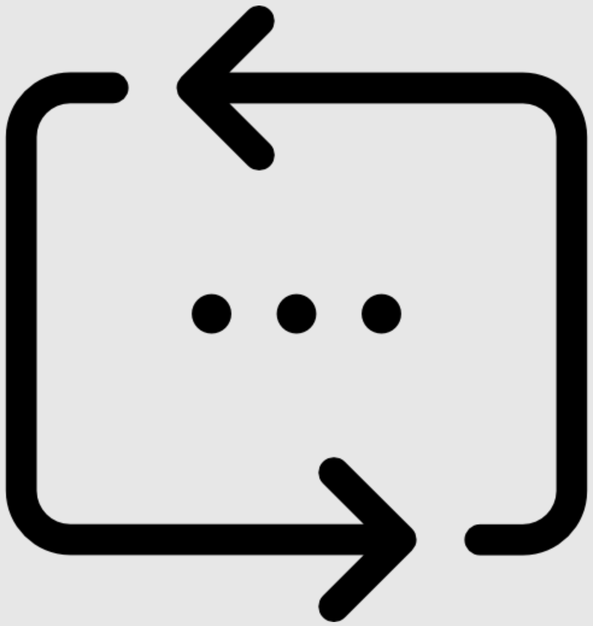
$$I_G(f) = \sum_{i=1}^{m} f_i(1 - f_i) = 1 - \sum_{i=1}^{m} f_i^2$$

Variance Reduction

$$I_V(N) = \frac{1}{|S|}\sum_{i\in S}\sum_{j\in S}\frac{1}{2}\left(x_i - x_j\right)^2 - \left(\frac{1}{|S_t|}\sum_{i\in S_t}\sum_{j\in S_t}\frac{1}{2}\left(x_i - x_j\right)^2\right.$$

Method

재귀적 분기
(Recursive Partitioning)



가지치기
(Pruning)

Recursive Partitioning

61.5	20.8	Owner
93.0	20.8	Owner
52.8	20.8	Non-owner
64.8	21.6	Owner
51.0	22.0	Owner
82.8	22.4	Owner
87.0	23.6	Owner

$$\text{분기 전 엔트로피} = -\frac{1}{2}\log_2\left(\frac{1}{2}\right) - \frac{1}{2}\log_2\left(\frac{1}{2}\right) = 1$$

$$\text{분기 후 엔트로피} = \frac{1}{24}(-\log_2 1) + \frac{23}{24}\left(-\frac{12}{23}\log_2\left(\frac{12}{23}\right) - \frac{11}{23}\log_2\left(\frac{11}{23}\right)\right) \approx 0.96$$

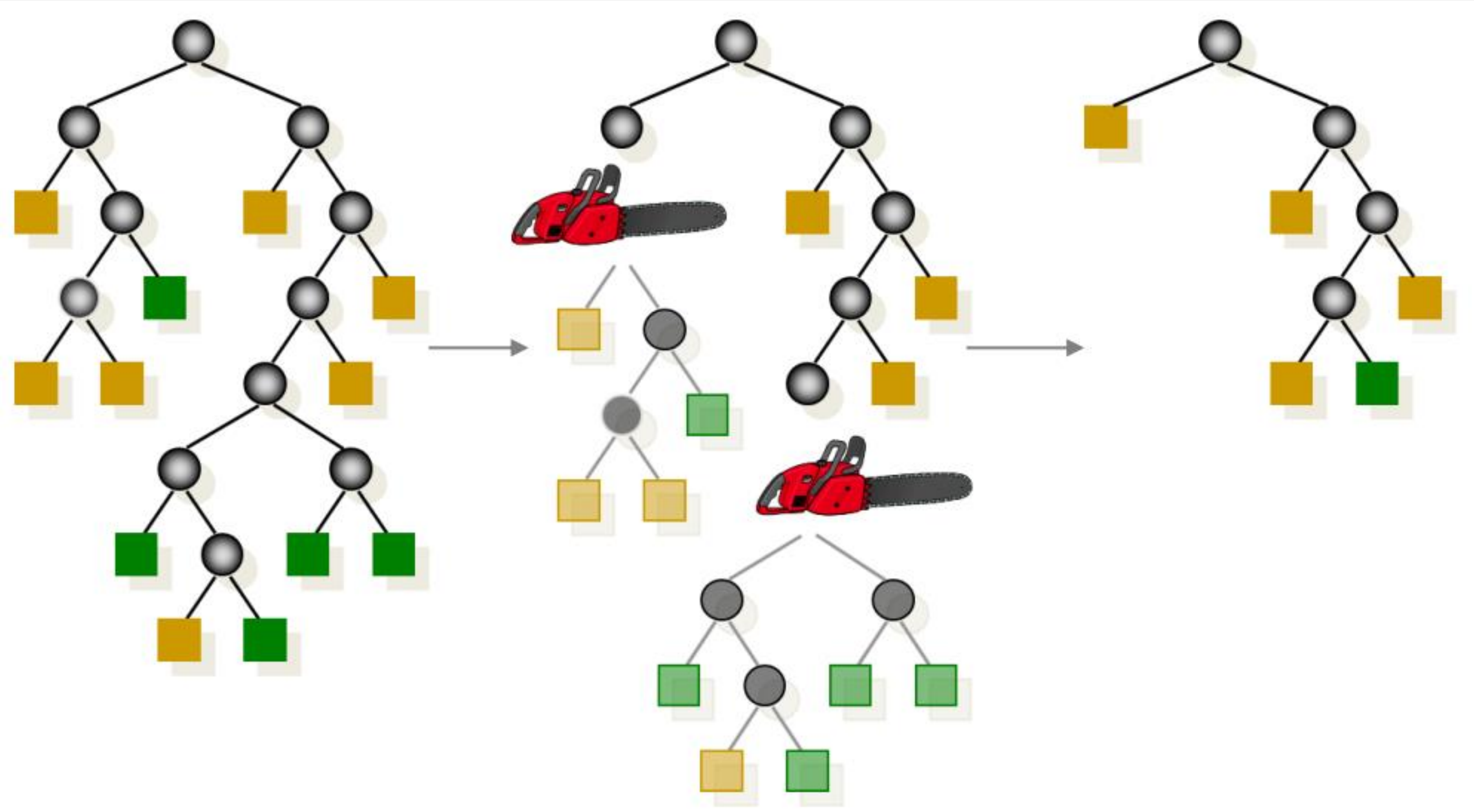
$$\text{정보획득} = 1 - 0.96 = 0.04$$

이후 분기 지점을 두번째 레코드로 두고 처음 두 개 레코드와 나머지 22개 레코드 간의 엔트로피를 계산한 뒤 정보획득을 알아봅니다. 이렇게 순차적으로 계산한 뒤, 이번엔 다른 변수인 소득을 기준으로 정렬하고 다시 같은 작업을 반복합니다. 모든 경우의 수 가운데 정보획득이 가장 큰 변수와 그 지점을 택해 첫번째 분기를 하게 됩니다. 이후 또 같은 작업을 반복해 두번째, 세번째... 이렇게 분기를 계속 해 나가는 과정이 바로 의사결정나무의 학습입니다.

그렇다면 1회 분기를 위해 계산해야 하는 경우의 수는 총 몇 번일까요? 개체가 n개, 변수가 d개라고 할 때 경우의 수는 $d(n-1)$개가 됩니다. 분기를 하지 않는 경우를 제외하고 모든 개체와 변수를 고려해 보는 것입니다.

<https://ratsgo.github.io/machine%20learning/2017/03/26/tree/>

Pruning



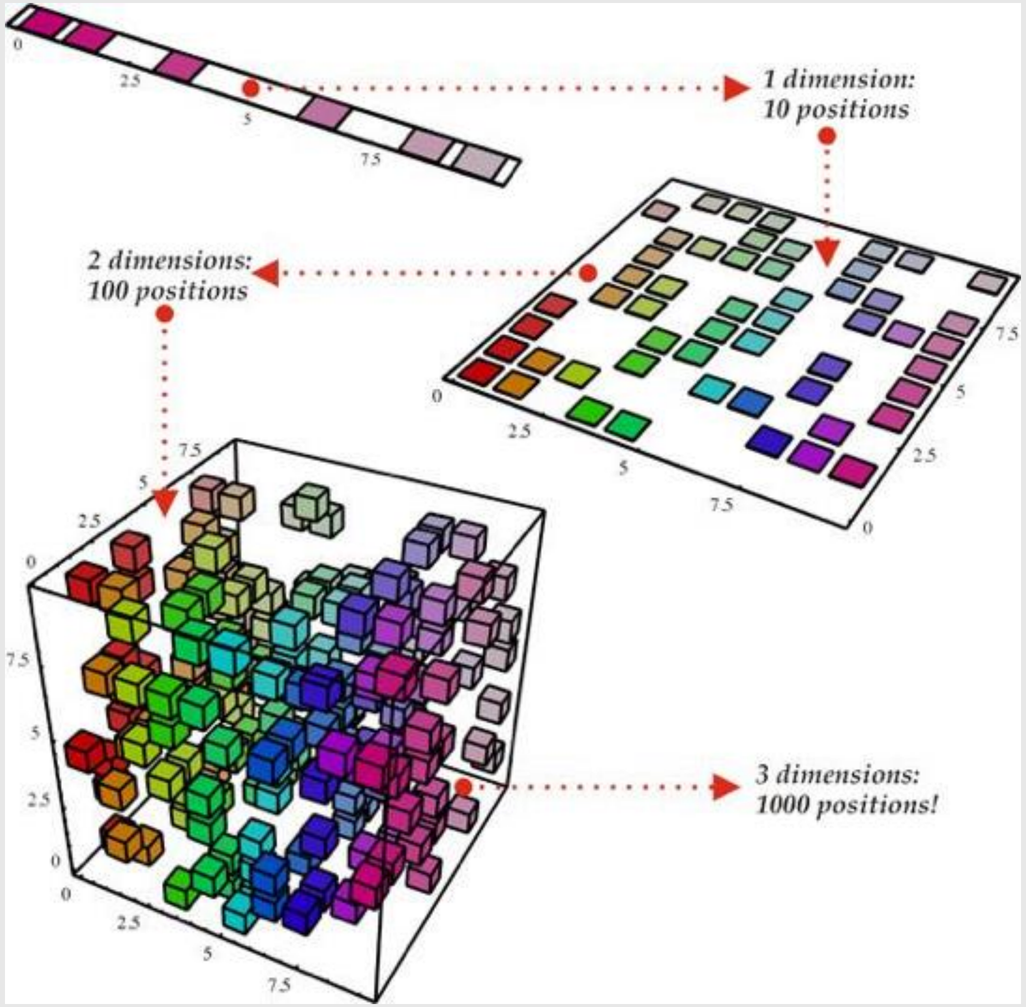
〈https://ratsgo.github.io/machine%20learning/2017/03/26/tree/〉

Curse of Dimension

<http://thesciencelife.com/archives/1001>

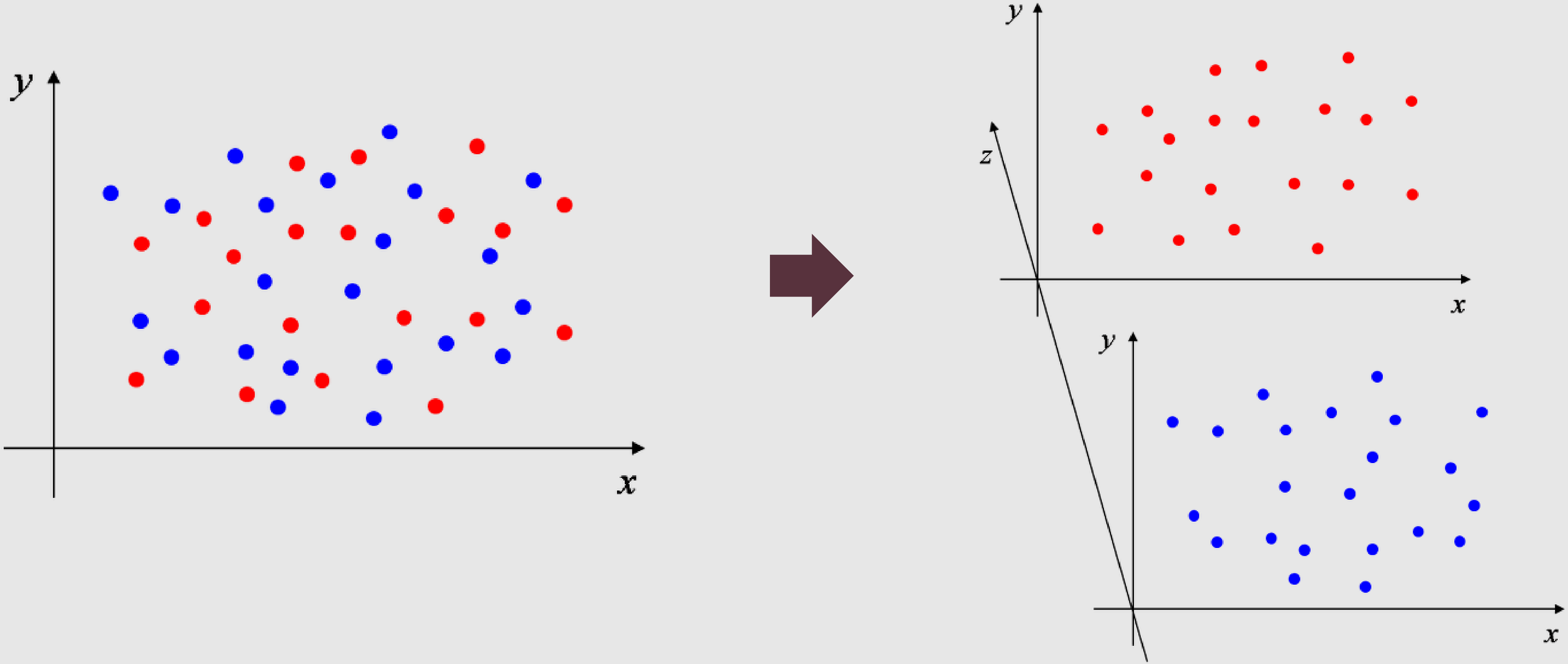
모델을 학습할 때 독립 샘플이 많을수록 학습이 잘 되는 것과 달리
차원이 커질수록 학습이 더 어려워진다.

Curse of Dimension

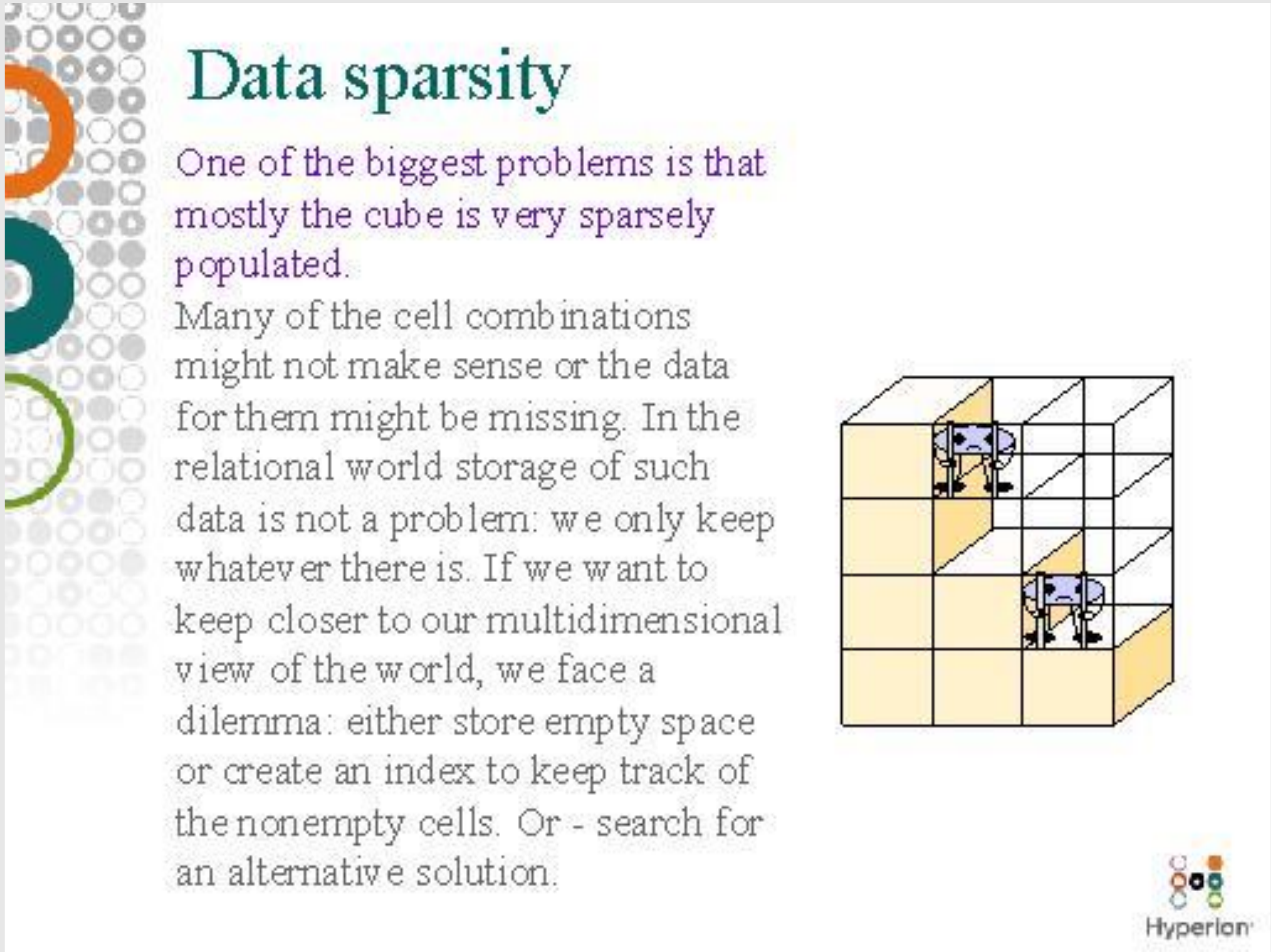


<https://brunch.co.kr/@kakao-it/161>

Curse of Dimension

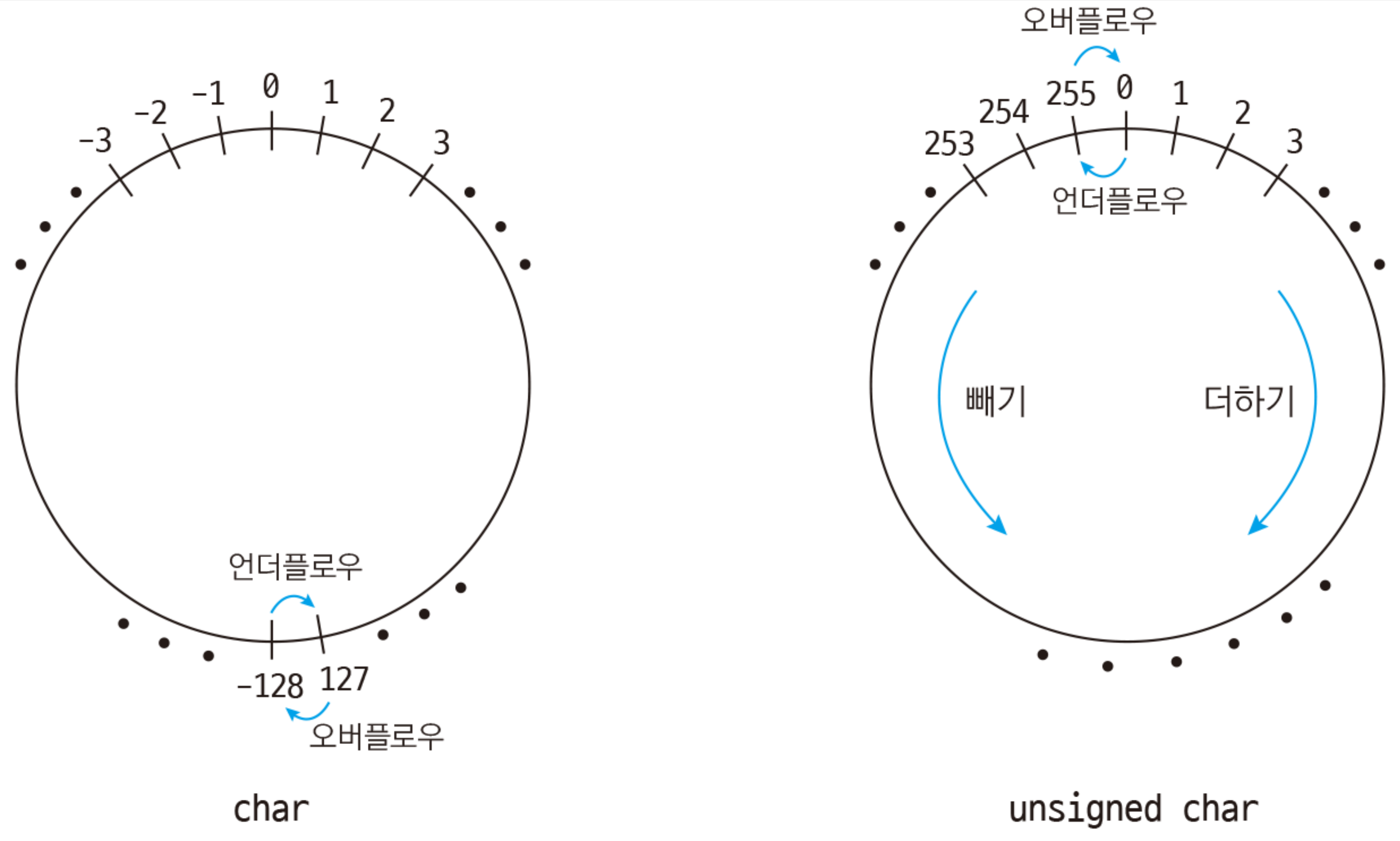


Curse of Dimension



<http://i.stanford.edu/infoseminar/archive/FallY99/russakovskii-slides/sld047.htm>

Underflow & Overflow



⟨https://dojang.io/mod/page/view.php?id=32⟩

Underflow Examle

<https://fr.ubergizmo.com/2013/07/17/paypal-verse-92-000-000-000-000-000.html>

Overflow Examle

<http://www.inven.co.kr/mobile/board/powerbbs.php?come_idx=4626&category=%EC%9D%BC%EB%B0%98&l=2699898>

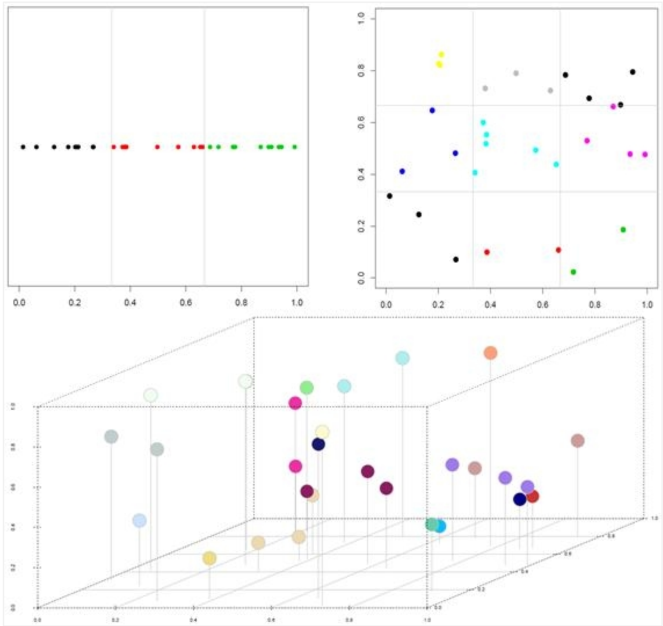
Nonparametric Model

0개 좋아요

통계학의 난제 '차원의 저주' 풀렸다

박병욱 교수, 구조화 비모수모형의 추정 통해

육심무 기자(=대전)



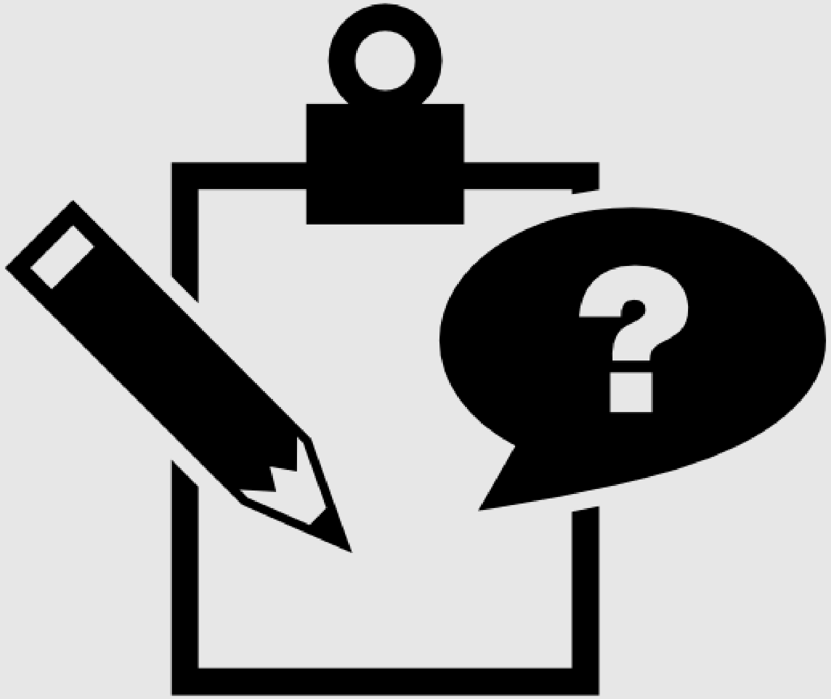
데이터의 성김 현상 (Data Sparsity) = 데이터 또는 변수의 차원이 증가함에 따라 국소영역에서 관측되는 데이터의 관측빈도가 기하적으로 감소하는 현상을 보여주며 이 현상이 차원의 저주를 발생시킨다. ⓒ 한국연구재단

〈http://www.pressian.com/news/article.html?no=206571#09T0〉

Fixed Design Additive Models

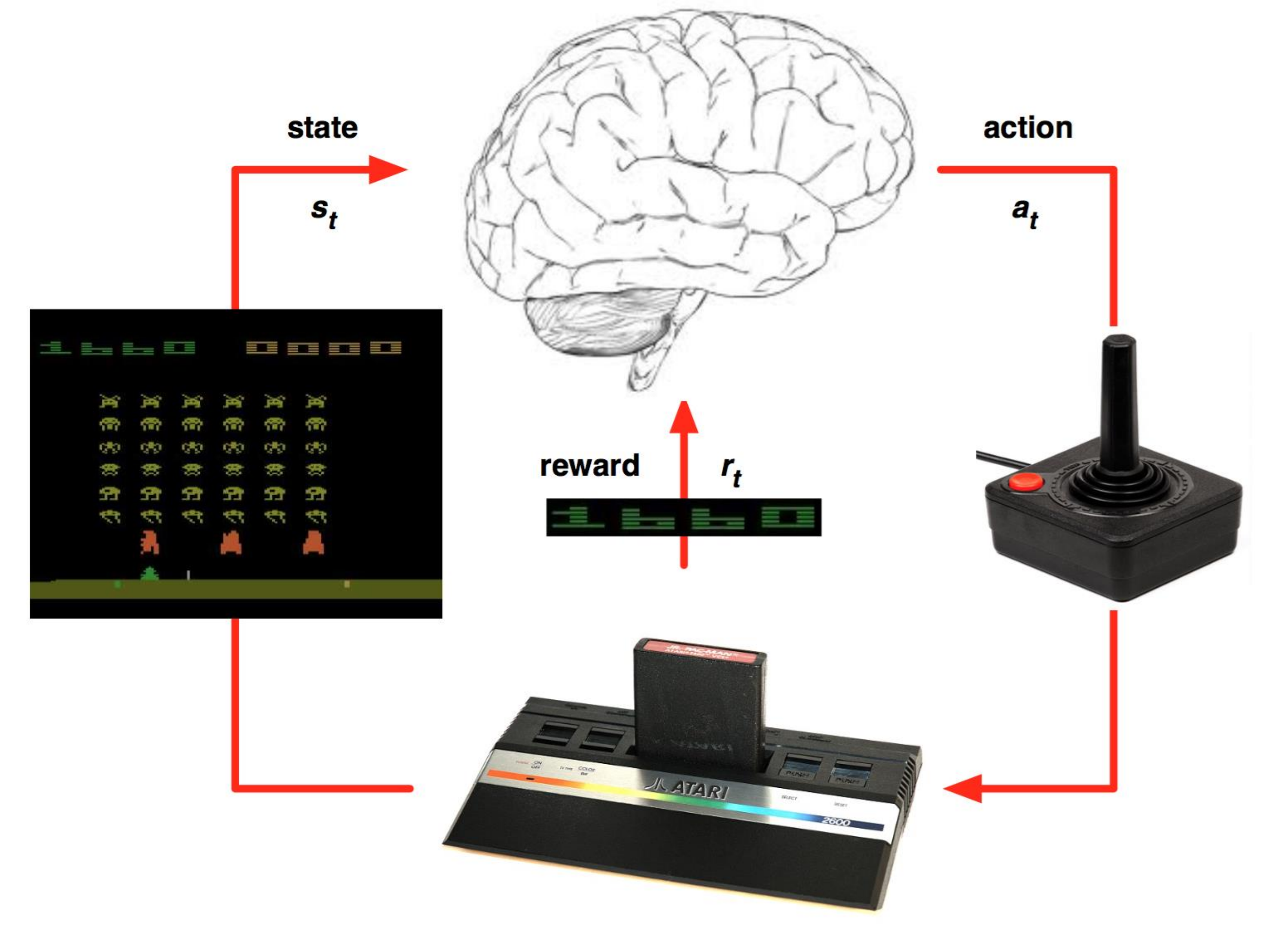
Consider a regression model of Y on X=(X1,X2,⋯,Xp),Y=m(X1,⋯,Xp)+ε, where m is the smooth function to be estimated and ε is a arandom error. When we are interested in the estimation of the regression surfacem(X1,⋯,Xp)nonparametrically, the p-dimensional multivariate smoother can be used.The multivariate local averaging procedure stil gives asymptotically consistent estimatorsto the regression surface. However, there are two major problems with this approach.First, there is the curse of dimensionality which means that the convergence rate isvery slow in high dimension. Second, estimates are difficult to interpret for p〉3.One way of avoiding these problems is to impose an additive structure on theregression function. More precisely, the regression function takes the form m(X1,⋯,Xp)=(관계식 원문참조) in the additive model, where fjs are smooth functions to beestimated.The objective of this research was to develop kernel estimaton procedures foradditive regression models with fixed design. Since the major attraction of the additivemodel is the achievability of the univariate optimal rate of convergence, the optimal rateof convergence of the estimators is important. We proposed a fixed design calledpermutation fixed design where the kernel estimator of the additive mean functionattains the univariate optimal rate of convergence. We used the Gasser-Mullerestimator for the regression model. The estimator of the additive mean function Σpj=1fj(Xj) was defined as a sum of the estimators fj and fj was constructed by smoothingresponse observations. The kernel estimator in the permutation fixed design caseattains the univariate optimal rate of convergence for any p≥1.

〈http://www.ndsl.kr/ndsl/search/detail/report/reportSearchResultDetail.do?cn=TRKO200200017168〉



Before
Finish Class

Reinforce Learning



04 IV. Before Finish Class

Reinforce Learning

〈https://www.youtube.com/watch?v=V1eYniJ0Rnk〉

Reinforce Learning

Reinforce Learning

10월 19일: 중간고사
10월 26일: (선택)
11월 02일: (선택)


Ideas worth spreading
- TED Talks

고생하셨습니다.