Machine Learning

Contents

7.1. Characteristics of Support Vector Machine

- **Feed-forward Neural Network (Perceptron, MLP, RBFN..)**
  - Stochastic algorithm
  - Generalizes well but need a lot of tuning
  - Can be learned in incremental fashion
  - To learn complex functions: use hidden layers

- **SVM**
  - Deterministic algorithm
  - Nice Generalization with few parameters to tune
  - Hard to learn – quadratic programming techniques
  - Using kernel tricks to learn very complex functions

7.2. Linear Separator and Perceptron

- **Linear Separator**

$$g(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + w_0 \qquad L = \{\boldsymbol{w}^T\boldsymbol{x} + w_0 = 0\}$$

  - For any two points $x_1, x_2 \in L$

$$\boldsymbol{w}^T(\boldsymbol{x}_1 - \boldsymbol{x}_2) = 0$$

  - Define unit normal vector $\boldsymbol{w}^* = \boldsymbol{w}/||\boldsymbol{w}||$

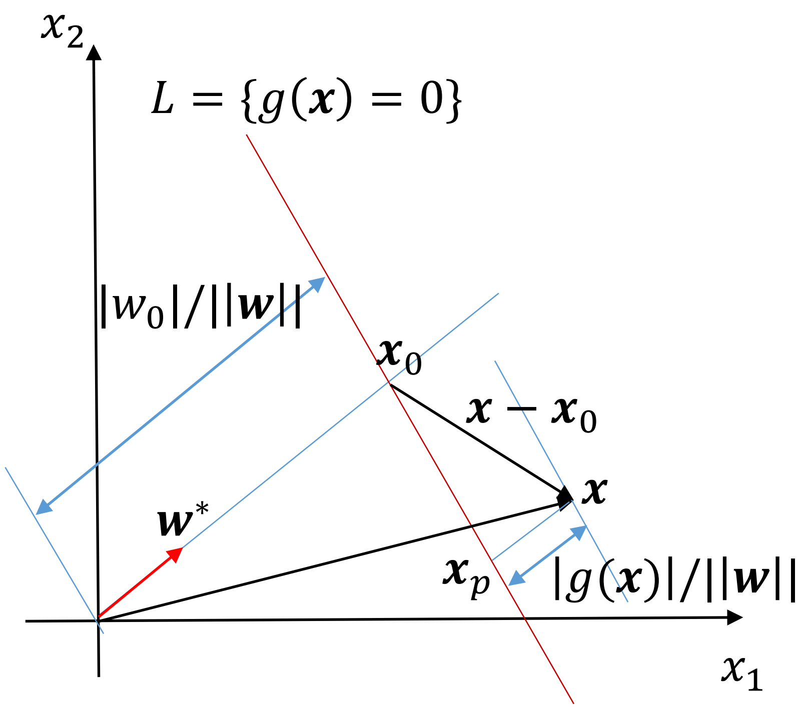
  - For any point $x_0 \in L$, $\boldsymbol{w}^T\boldsymbol{x}_0 = -w_0$

  - Distance of any x to L, $\boldsymbol{w}^{*T}(\boldsymbol{x} - \boldsymbol{x}_0) = \dfrac{\boldsymbol{w}^T\boldsymbol{x} + w_0}{||\boldsymbol{w}||}$

  - The geometric margin of example $< \boldsymbol{x}_i, y_i >$ with respect to the hyperplane

$$y_i \frac{\boldsymbol{w}^T\boldsymbol{x} + w_0}{||\boldsymbol{w}||}, \quad y_i \in \{-1, +1\}$$

  - A point is misclassified iff its margin is negative



Perceptron Learning Algorithm

- To minimize

$$D(\mathbf{w}, w_0) = -\sum_{i \in M} y_i(\mathbf{w}^T\mathbf{x}_i + w_0)$$

- Gradient

$$\frac{\partial D(\mathbf{w}, w_0)}{\partial \mathbf{w}} = -\sum_{i \in M} y_i \mathbf{x}_i \qquad \frac{\partial D(\mathbf{w}, w_0)}{\partial w_0} = -\sum_{i \in M} y_i$$

퍼셉트론 알고리즘

○입력과 목표 값의 쌍으로 구성된 학습패턴 $<\mathbf{x}_i, y_i>$를 저장한다.

① 가중치 $\mathbf{w}$와 w_0를 임의의 값으로 초기화 시킨다.

② n개의 학습패턴에 대하여 가중치를 다음과 같이 변경시킨다.

$$\text{If } y_i(\mathbf{w}^T\mathbf{x}_i + w_0) \le 0 \text{ then } \begin{cases} \mathbf{w} := \mathbf{w} + y_i\mathbf{x}_i \\ w_0 := w_0 + y_i \end{cases} \qquad (7.2.9)$$

③ 오인식된 학습패턴이 있으면 과정 ②를 다시 수행한다.

④ 새로운 입력 $\mathbf{x}$가 주어지면 $g(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0$의 부호로 예측한다.

Perceptron Algorithm: Dual Representation

- α_i : a count of the number of times that example i was misclassified
- Initial weights are all zeros
- Then, final weights are

$$\boldsymbol{w} = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i \qquad w_0 = \sum_{i=1}^{n} \alpha_i y_i$$

- The output of linear predictor is

$$h(\boldsymbol{x}) = sign(\boldsymbol{w}^{\boldsymbol{T}} \boldsymbol{x} + w_0) = sign \sum_{i=1}^{n} \alpha_i y_i (\boldsymbol{x}_i^T \boldsymbol{x} + 1)$$

퍼셉트론 알고리즘의 이중적 표현

○입력과 목표값의 쌍으로 구성된 학습패턴 $< \boldsymbol{x}_i, y_i >$를 저장한다.

① α_i는 영으로 초기화 시킨다.

② 학습 패턴 n개에 대하여 가중치를 다음과 같이 변경시킨다.

$$\text{If } \sum_{j=1}^{n} y_i \alpha_j y_j (\boldsymbol{x}_j^T \boldsymbol{x}_i + 1) \leq 0 \text{ then } \alpha_i := \alpha_i + 1 \tag{7.2.13}$$

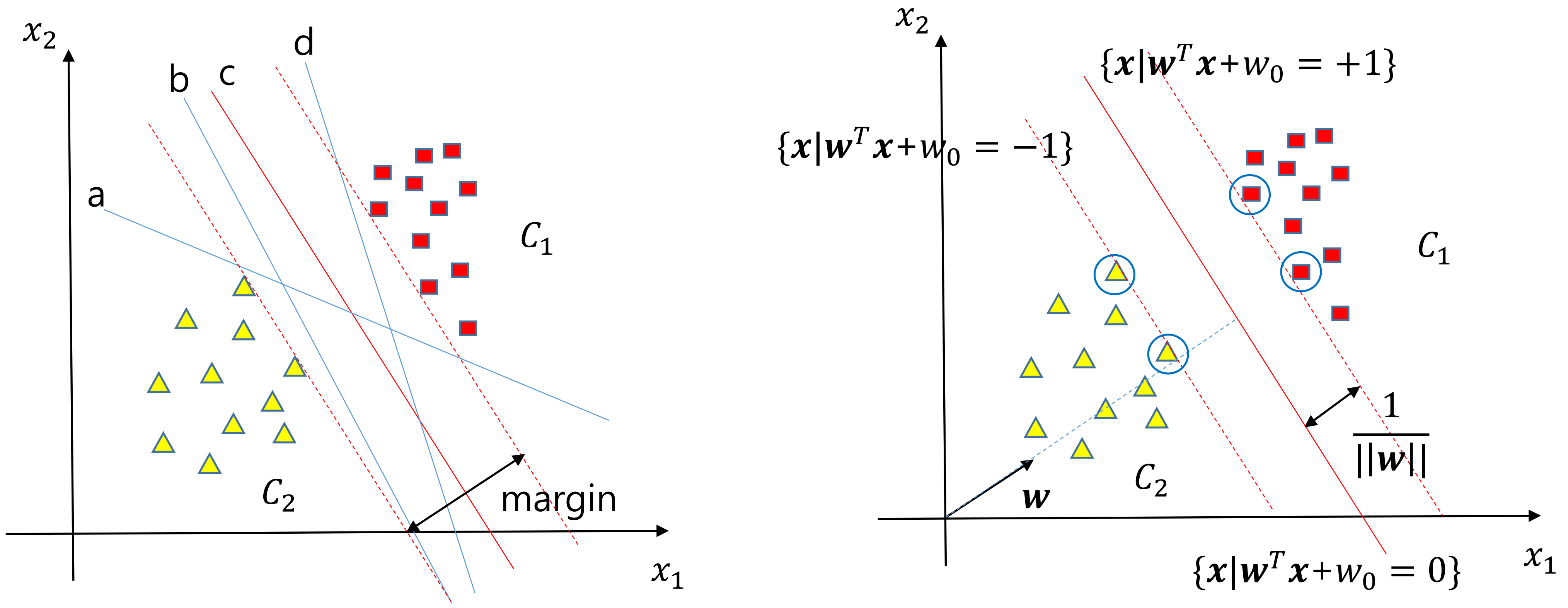
$$\boldsymbol{w} = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i \text{ and } w_0 = \sum_{i=1}^{n} \alpha_i y_i$$

③ 오인식된 학습패턴이 있으면 과정 ②를 다시 수행한다.

④ 새로운 입력 $\boldsymbol{x}$가 주어지면 $h(\boldsymbol{x})$로 예측한다.

7.3. Support Vector Machine

- Maximizing the margin
- The decision boundary: determined by a subset of the data points, known as support vectors (indicated by the circles).



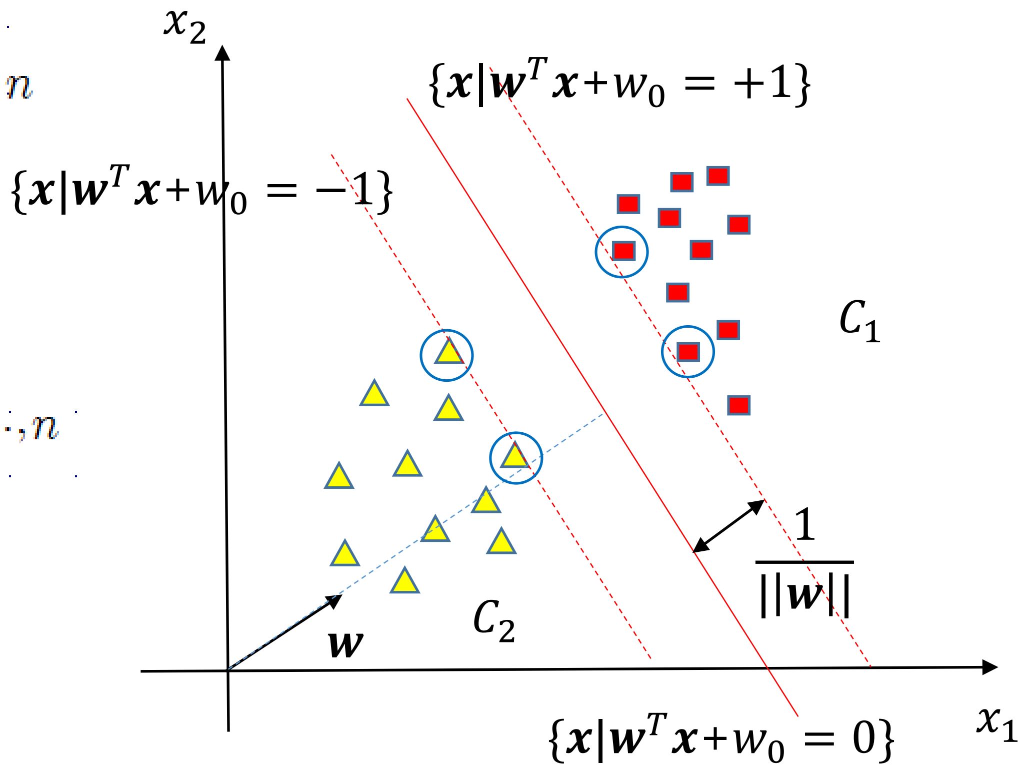
Support Vector Machine

- **Support vector machines**
  - Names a whole family of algorithms of the **maximum margin separator**. The idea is to find the separator with the maximum margin from all the data points.
- Optimization problem

$$\max_{w_0,\boldsymbol{w}} C \quad \text{subject to} \quad \frac{1}{\|\boldsymbol{w}\|} y_i(\boldsymbol{w}^T\boldsymbol{x}_i + w_0) \geq C \;\; i = 1,2,..,n$$

- Set $\|\boldsymbol{w}\|$ to $1/C$

$$\min_{w_0,\boldsymbol{w}} \frac{1}{2}(\|\boldsymbol{w}\|)^2 \quad \text{subject to} \quad y_i(\boldsymbol{w}^T\boldsymbol{x}_i + w_0) \geq 1 \;\; i = 1,2,..,n$$



Support Vector Machine: Formulation

- **Quadratic optimization problem**

$$\min_{w_0, \boldsymbol{w}} \frac{1}{2}(\|\boldsymbol{w}\|)^2 \quad \text{subject to} \quad y_i(\boldsymbol{w}^T\boldsymbol{x}_i + w_0) \geq 1 \;\; i = 1, 2, \cdots, n$$

- Lagrangian formulation of constrained optimization

$$\min_{w_0, \boldsymbol{w}} \max_{\alpha \geq \boldsymbol{0}} L(w_0, \boldsymbol{w}, \alpha) = \frac{1}{2}(\|\boldsymbol{w}\|)^2 - \sum_{i=1}^{n} \alpha_i \left[y_i(\boldsymbol{w}^T\boldsymbol{x}_i + w_0) - 1 \right]$$

- Kuhn-Tucker Theorem $\min_{w_0, \boldsymbol{w}} \max_{\alpha \geq \boldsymbol{0}} L(w_0, \boldsymbol{w}, \alpha) = \max_{\alpha \geq \boldsymbol{0}} \min_{w_0, \boldsymbol{w}} L(w_0, \boldsymbol{w}, \alpha)$

$$\frac{\partial L(w_0, \boldsymbol{w}, \alpha)}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i = 0 \qquad \boldsymbol{w} = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i$$

$$\frac{\partial L(w_0, \boldsymbol{w}, \alpha)}{w_0} = -\sum_{i=1}^{n} \alpha_i y_i = 0 \qquad \sum_{i=1}^{n} \alpha_i y_i = 0$$

Support Vector Machine: Formulation & Solution

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

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

$$\min_{w_0, \boldsymbol{w}} \max_{\alpha \geq \mathbf{0}} L(w_0, \boldsymbol{w}, \alpha) = \frac{1}{2}(\|\boldsymbol{w}\|)^2 - \sum_{i=1}^{n} \alpha_i \left[y_i (\boldsymbol{w}^T \boldsymbol{x}_i + w_0) - 1 \right]$$

$$L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{j=1}^{n} \sum_{k=1}^{n} \alpha_j \alpha_k y_j y_k (\boldsymbol{x}_j^T \boldsymbol{x}_k)$$

$$\text{Maximize } L(\alpha) \text{ subject to } \boldsymbol{\alpha} \geq \mathbf{0} \text{ and } \sum_i \alpha_i y_i = 0$$

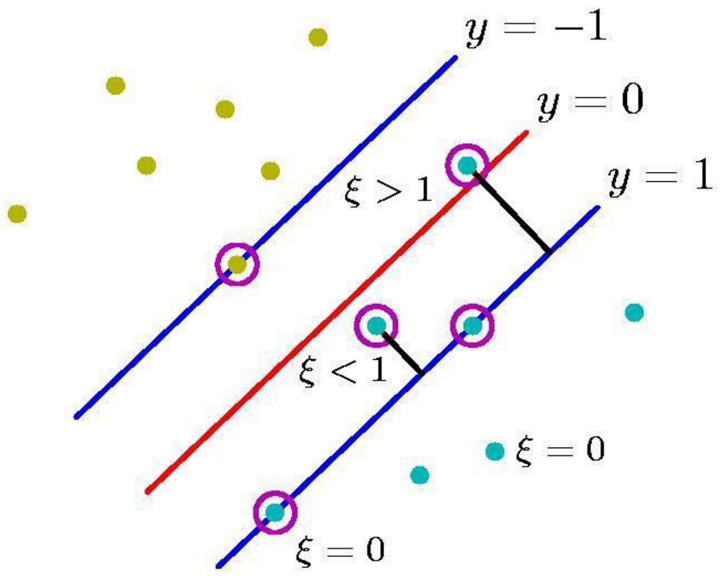
Finding optimal α_i : computationally tractable quadratic programming problem

Support Vector: points with margin=1

$$y_i (\boldsymbol{w}^T \boldsymbol{x}_i + w_0) = 1 \qquad w_0 = y_i - \boldsymbol{w}^T \boldsymbol{x}_i$$

Support Vector Machines

- What if the problem is not linearly separable?



- Introduce slack variables
  - Need to minimize:

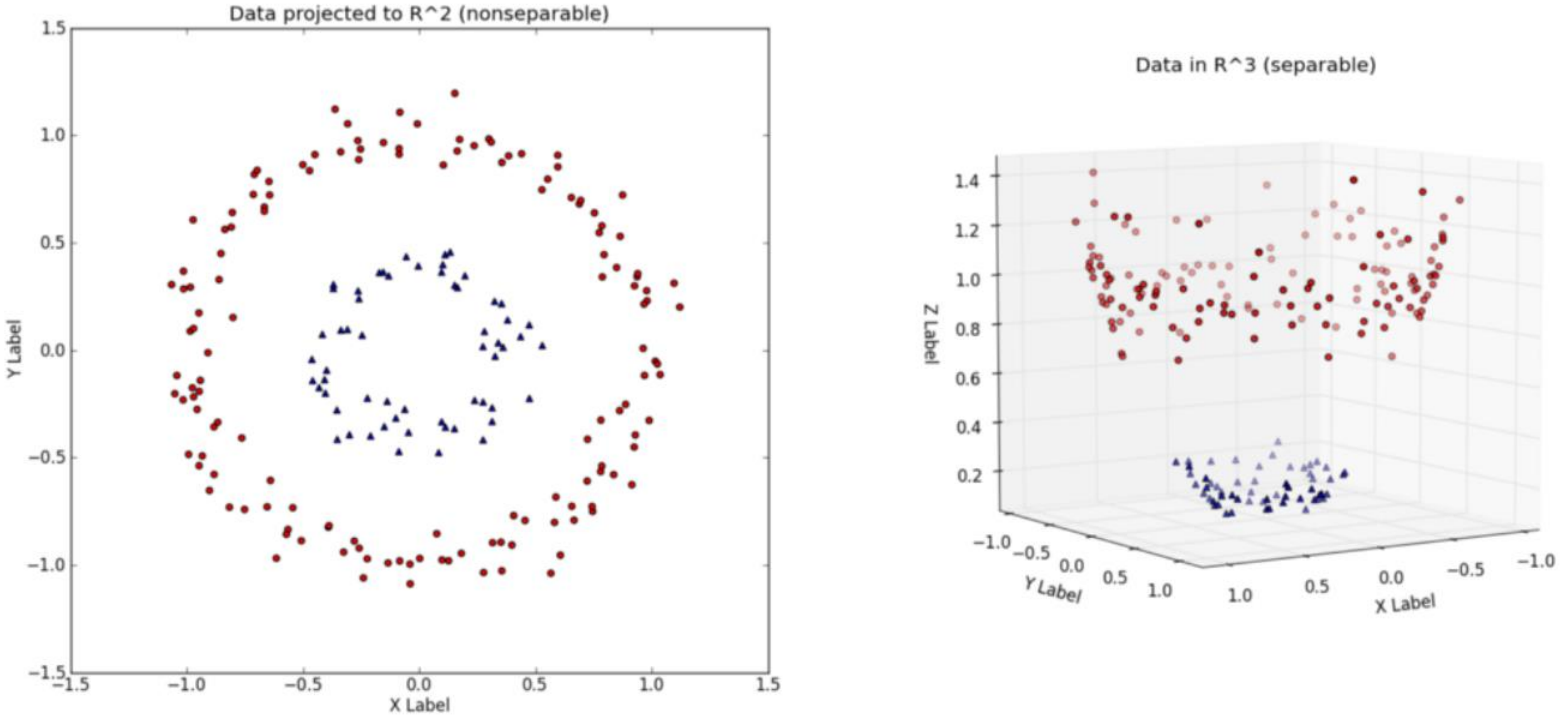
$$L(w, \xi) = \frac{\| \vec{w} \|^2}{2} + C\left(\sum_{i=1}^{m} \xi_i \right)$$

  - Subject to:

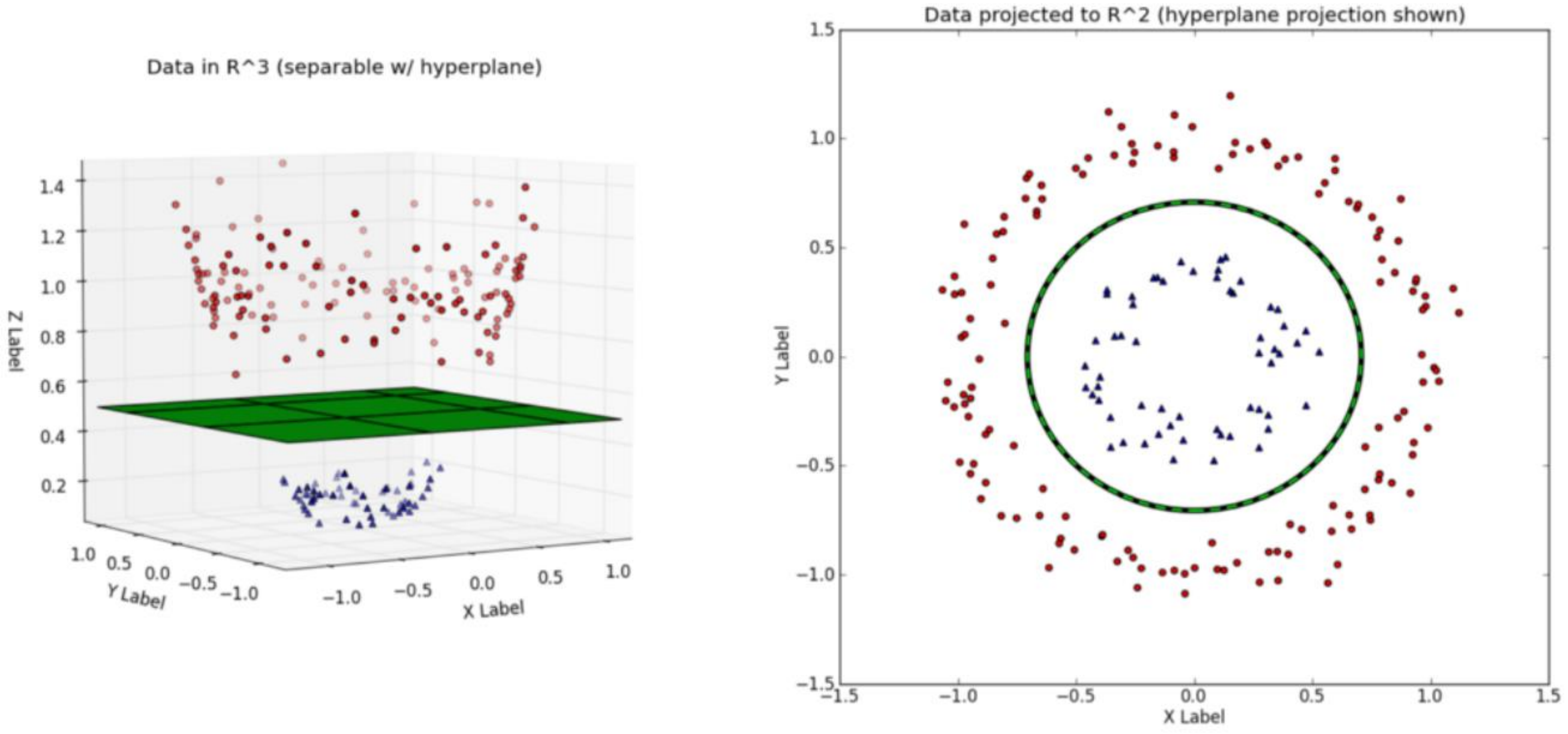
$$y_i(\vec{w} \bullet \vec{x}_i + b) \geq 1 - \xi_i, \text{for all } (\vec{x}_i, y_i) \text{ in D}$$

Support Vector Machines

- What if decision boundary is not linear?



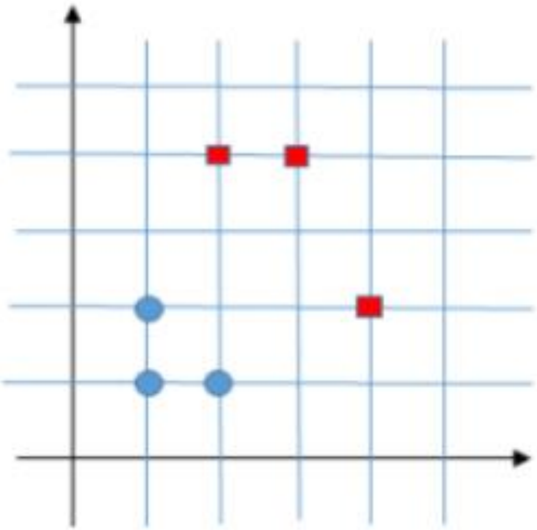
A nonseparable dataset in a two-dimensional space R^2, and the same dataset mapped onto threedimensions with the third dimension being x^2+y^2 (source: http://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html)



The decision boundary is shown in green, first in the three-dimensional space (left), then back in the two-dimensional space (right). Same source as previous image.

예제 7.3-1

그림과 같이 2차원 공간상에 두 개의 클래스에 해당하는 점집합이 주어졌다. 사각형 클래스는 $y_i = 1$, 원형 클래스는 $y_i = -1$이라고 두고서 퍼셉트론의 이중적 표현에 의한 학습을 두 epoch 동안 반복하여 보아라. 그리고, Support Vector의 지점을 구하고, 이를 근거로 SVM에 의한 구분자를 구하여라.



내적	$\boldsymbol{x}_1$	$\boldsymbol{x}_2$	$\boldsymbol{x}_3$	$\boldsymbol{x}_4$	$\boldsymbol{x}_5$	$\boldsymbol{x}_6$
$\boldsymbol{x}_1$	20	6	22	10	16	8
$\boldsymbol{x}_2$	6	2	7	3	6	3
$\boldsymbol{x}_3$	22	7	25	11	20	10
$\boldsymbol{x}_4$	10	3	11	5	81	4
$\boldsymbol{x}_5$	16	6	20	8	20	10
$\boldsymbol{x}_6$	8	3	10	4	10	5

풀이

먼저, 6개의 점들의 순서를 각 클래스가 섞이도록 $\boldsymbol{x}_1(2,4)$, $\boldsymbol{x}_2(1,1)$, $\boldsymbol{x}_3(3,4)$, $\boldsymbol{x}_4(1,2)$, $\boldsymbol{x}_5(4,2)$, $\boldsymbol{x}_6(2,1)$와 같이 정하였다. 이 점들 사이의 내적을 표로 만들면 위와 같다. 이 표를 활용하여 수식 $\alpha_i(i=1,2,..,6)$를 변경시킨다. 여기서, $\boldsymbol{x}_1$, $\boldsymbol{x}_3$, $\boldsymbol{x}_5$는 $y_i=1$이고 나머지 점들은 $y_i=-1$이다.

우선, 첫 단계로 모든 $\alpha_i(i=1,2,..,6)$를 영으로 설정한다.

첫 epoch에서 $\boldsymbol{x}_i(i=1,2,..,6)$을 순차적으로 입력하면

$$\boldsymbol{x}_i(i=1)\text{입력}: \text{모든 } \alpha_i\text{는 영이므로 } y_1\sum_{j=1}^{n}\alpha_j y_j(\boldsymbol{x}_j^T\boldsymbol{x}_1+1)=0,\ \alpha_1:=\alpha_1+1=1$$

$$\boldsymbol{x}_i(i=2)\text{입력}: y_2\sum_{j=1}^{n}\alpha_j y_j(\boldsymbol{x}_j^T\boldsymbol{x}_2+1)=-1[1\times(6+1)]=-7,\ \alpha_2:=\alpha_2+1=1$$

$$\boldsymbol{x}_3\text{입력}: y_3\sum_{j=1}^{n}\alpha_j y_j(\boldsymbol{x}_j^T\boldsymbol{x}_3+1)=1[1\times(22+1)-1\times(7+1)]=15,\ \alpha_3:=\alpha_3=0$$

$$\boldsymbol{x}_4\text{입력}: y_4\sum_{j=1}^{n}\alpha_j y_j(\boldsymbol{x}_j^T\boldsymbol{x}_4+1)=-7,\ \alpha_4:=\alpha_4+1=1$$

$\boldsymbol{x}_5$ 입력 : $y_5 \sum_{j=1}^{n} \alpha_j y_j (\boldsymbol{x}_j^T \boldsymbol{x}_5 + 1) = 1, \ \alpha_5 := \alpha_5 = 0$

$\boldsymbol{x}_6$ 입력 : $y_6 \sum_{j=1}^{n} \alpha_j y_j (\boldsymbol{x}_j^T \boldsymbol{x}_6 + 1) = 0, \ \alpha_6 := \alpha_6 + 1 = 1$

두 번째 epoch에서 다시 $\boldsymbol{x}_i (i = 1,2,..,6)$을 순차적으로 입력하면

$\boldsymbol{x}_1$ 입력 : $y_1 \sum_{j=1}^{n} \alpha_j y_j (\boldsymbol{x}_j^T \boldsymbol{x}_1 + 1) = -6, \ \alpha_1 := \alpha_1 + 1 = 2$

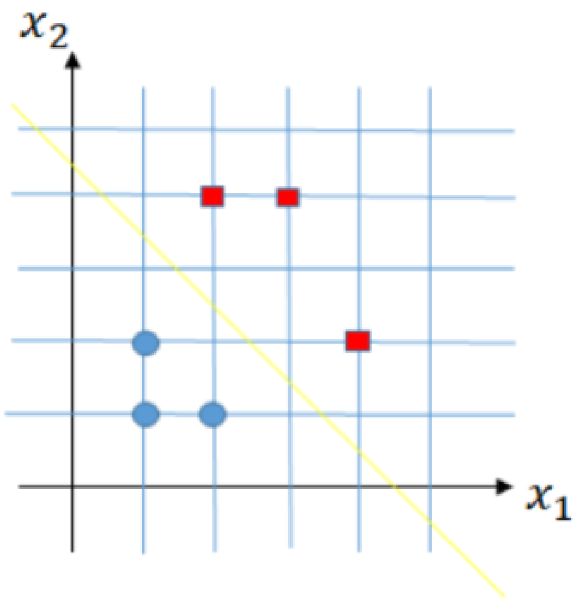
$\boldsymbol{x}_2$ 입력 : $y_2 \sum_{j=1}^{n} \alpha_j y_j (\boldsymbol{x}_j^T \boldsymbol{x}_2 + 1) = -3, \ \alpha_2 := \alpha_2 + 1 = 2$

$\boldsymbol{x}_3$ 입력 : $y_3 \sum_{j=1}^{n} \alpha_j y_j (\boldsymbol{x}_j^T \boldsymbol{x}_3 + 1) = 7, \ \alpha_3 := \alpha_3 = 0$

$\boldsymbol{x}_4$ 입력 : $y_4 \sum_{j=1}^{n} \alpha_j y_j (\boldsymbol{x}_j^T \boldsymbol{x}_4 + 1) = -3, \ \alpha_4 := \alpha_4 + 1 = 2$

$\boldsymbol{x}_5$ 입력 : $y_5 \sum_{j=1}^{n} \alpha_j y_j (\boldsymbol{x}_j^T \boldsymbol{x}_5 + 1) = -11, \ \alpha_5 := \alpha_5 + 1 = 1$

그림에서 Support Vector는 사각 클래스의 $\boldsymbol{x}_1(2,4), \boldsymbol{x}_5(4,2)$와 원 클래스의 $\boldsymbol{x}_4(1,2)$, $\boldsymbol{x}_6(2,1)$이다. 이 4개의 점들을 기준으로 $\boldsymbol{x}_1, \boldsymbol{x}_5$에서는 +1의 값을 가지고 $\boldsymbol{x}_4, \boldsymbol{x}_6$에서는 -1의 값을 가지는 선형 구분자는 $g(\boldsymbol{x}) = \frac{2}{3}(x_1 + x_2) - 3$이다. 이를 그림으로 그리면 아래와 같다.



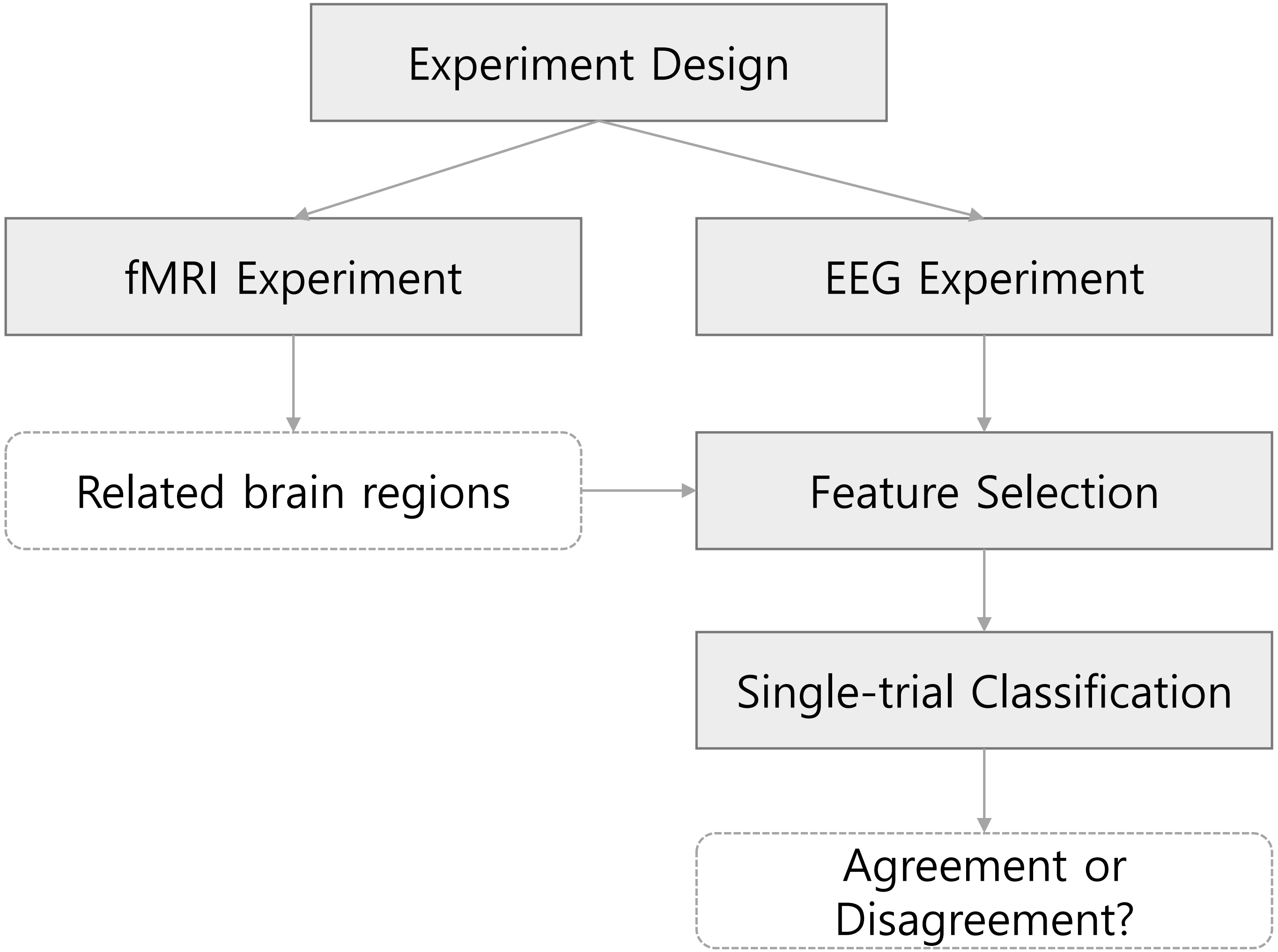
7.4. Application of SVM[10] [Suh-Yeon Dong, et al. 2016]

Objective: Discriminate agreement and disagreement to the given self-relevant sentence in the single-trial level.

- **Stimuli:** 74 Korean sentences from the Minnesota Multiphasic Personality inventory-II (MMPI-II). Sentence contents are related to personal experience.
- **Presentation:** Considering the Subject-object-verb (SOV) typology of the Korean language, each sentence was separated into two parts: the verb (sentence ending) and the remainder of the sentence (contents).

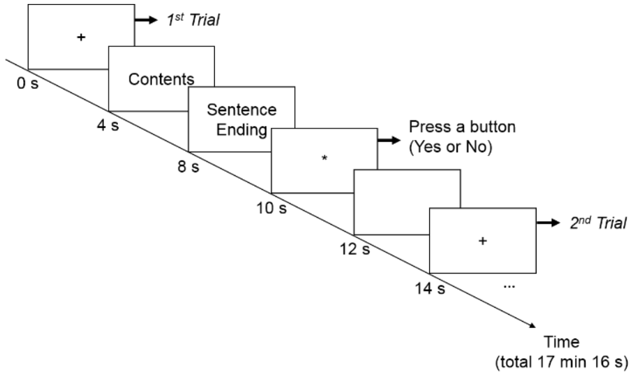
(a) Positive ending	**Contents**	**Sentence ending**
Stimulus sentence (Korean)	돈에 대해 걱정한 적이	있다
English translations in SOV form	The experience of worrying over mon ey	Does exist
Original English MMPI-2 sentence	I worry a great deal over money.	
(b) Negative ending	**Contents**	**Sentence ending**
Stimulus sentence (Korean)	기절한 적이	없다
English translations in SOV form	The experience of having a fainting sp ell	Does not exist
Original English MMPI-2 sentence	I have never had a fainting spell.	

Experiment Procedure



Experiment Procedure

- **fMRI Experiment (19 subjects)**

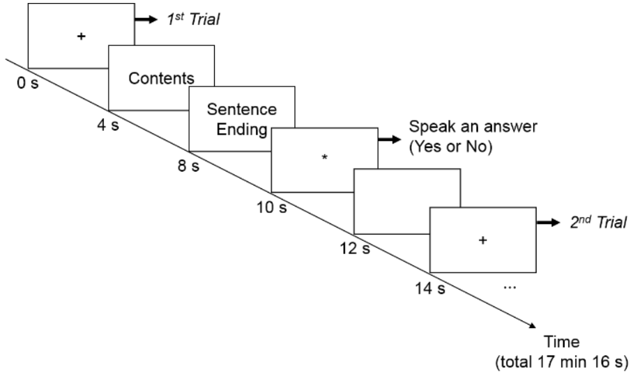


- **Image acquisition**
- 3T MR scanner (Siemens Magnetom Vero, Germany)
- MR-compatible goggle (NordicNeuroLab Visual systmes, Norway)
- Gradient-echo echo-planar imaging (EPI) sequence (36 slices; thickness = 4 mm; no gap between slices; FOV = 220 × 220 mm; matrix = 64 × 64; TE = 28 ms; TR = 2.0 s; flip angle = 90 °; voxel size 3.4 mm × 3.4 mm × 4 mm)

- **Preprocessing**
- (SPM8) Realign, coregister, segmentation, normalize, and[1] smooth

---

- **EEG Experiment (9 subjects)**



- **Data acquisition**
- BrainAmp system (Brain Products GmbH, Germany)
- 32-channel EEG cap (BrainCap)
- Eyetracker x120 (Tobii Technology, Sweden)

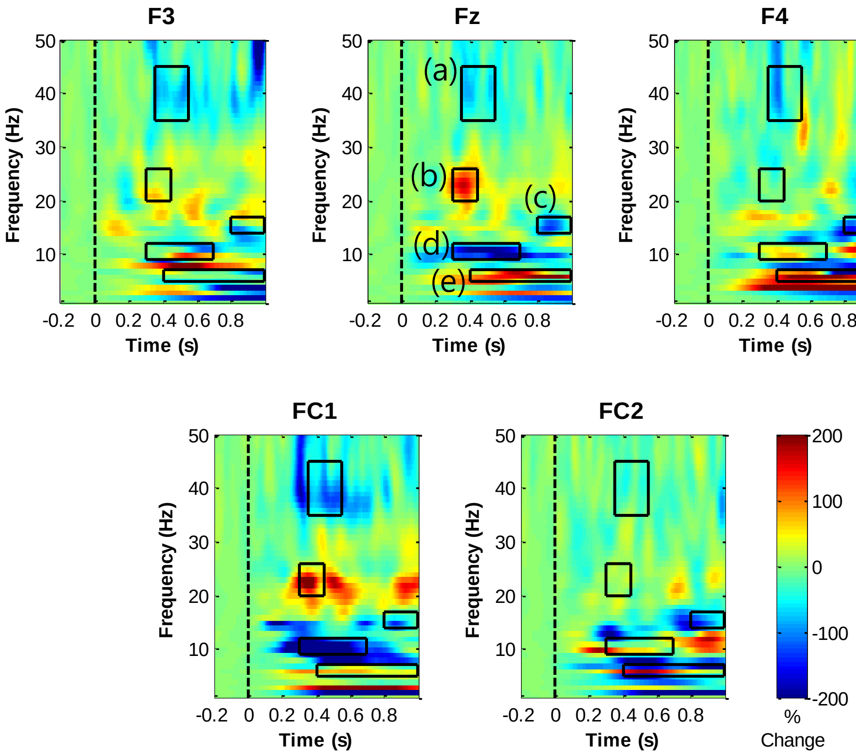
- **Preprocessing**
- 60Hz notch filtering and 1Hz high-pass filtering
- Offline re-referencing to average (except EOG and ECG)[2]
- Artifact Removal: EOG and ECG-related independent components
- Trial rejection: Reject trials whose absolute amplitude is over 70 μV

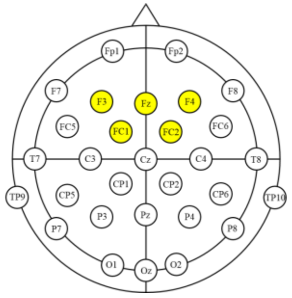
Feature Selection

- Referring to the fMRI results, responses at frontal channels are considered.

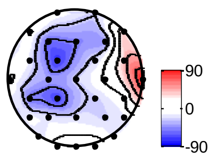
- **Time-frequency Representations (TFRs)**

Average TFR difference: Agree - Disagree

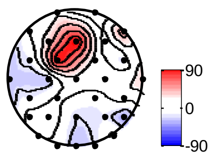




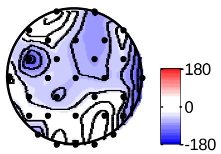
(a) Gamma 35-45Hz 350-550ms



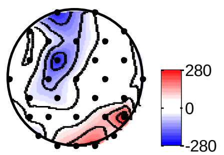
(b) Beta2 20-26Hz 300-450



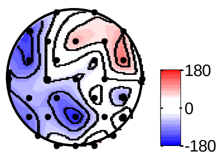
(c) Beta1 14-17Hz 800-1,000ms



(d) Alpha 9-12Hz 300-700n



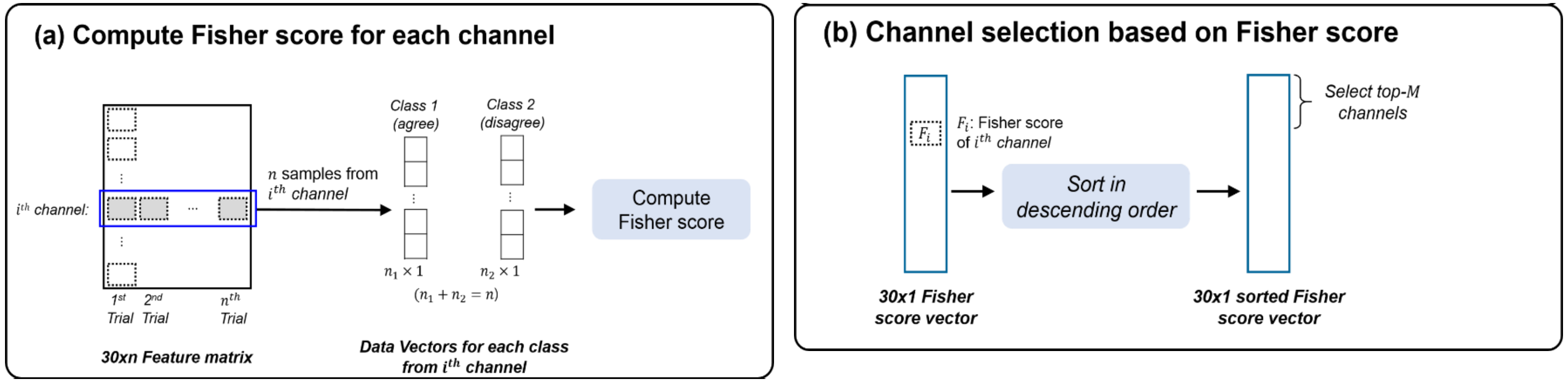
(e) Theta 5-7Hz 400-1,000ms



Select 5 feature candidates
(a) gamma 35-45Hz 350-550ms
(b) beta2 20-26Hz 300-450ms
(c) beta1 14-17Hz 800-1,000ms
(d) alpha 9-12Hz 300-700ms
(e) theta 5-7Hz 400-1,000ms

Channel Selection

- Channel selection using the Fisher score



The Fisher score for the i^{th} channel:[19]

$$F_i = \frac{\sum_{k=1}^{c} n_k \left(\mu_k^i - \mu^i\right)^2}{\sum_{k=1}^{c} n_k \left(\sigma_k^i\right)^2}$$

n_k: sample size of k^{th} class
μ_k^i: mean of k^{th} class in the i^{th} channel
σ_k^i: std of k^{th} class in the i^{th} channel
μ^i: mean of entire data in the i^{th} channel
c: Total number of classes (here, c = 2)

Rank	Theta		Alpha		Beta1		Beta2		Gamma	
	Channel	Fisher score	Channel	Fisher score	Channel	Fisher score	Channel	Fisher score	Channel	Fisher score
1	C3	0.028	C3	0.028	P7	0.034	C3	0.030	F3	0.040
2	CP5	0.027	Fz	0.027	T8	0.026	CP5	0.029	T8	0.030
3	CP2	0.025	CP1	0.026	F4	0.022	FC1	0.026	FC5	0.027
4	P7	0.025	FC1	0.025	FC1	0.022	Fp2	0.025	FC2	0.024
5	P3	0.023	F4	0.025	F3	0.020	Fp1	0.025	CP5	0.023

Classification

- Subject-dependent classification with increasing the number of selected channels
- Average accuracy using 5-fold cross validation
- SVM classifier with linear and RBF kernels (LIBSVM)

Component	Classifier	
	Linear SVM	RBF SVM
Theta	67.03% (30)	70.89% (2)
Alpha	66.39% (30)	73.86% (4)
Beta1	62.88% (30)	71.30% (4)
Beta2	65.07% (30)	73.49% (3)
Gamma	67.01% (20)	**75.54% (5)**