KAIG

딥러닝 스터디

SVM 모델 분석 및

Training & Feature 조사

이종혁

목차

모델 설명

- Hyperplane
- Support Vector
- Margin

매개변수

- Kernel
- C
- gamma

조사내용

- 프로젝트 내의 SVM
- Feature/Dimension/Attribute

Support Vector Machine

"How do we divide the space with decision boundaries?"

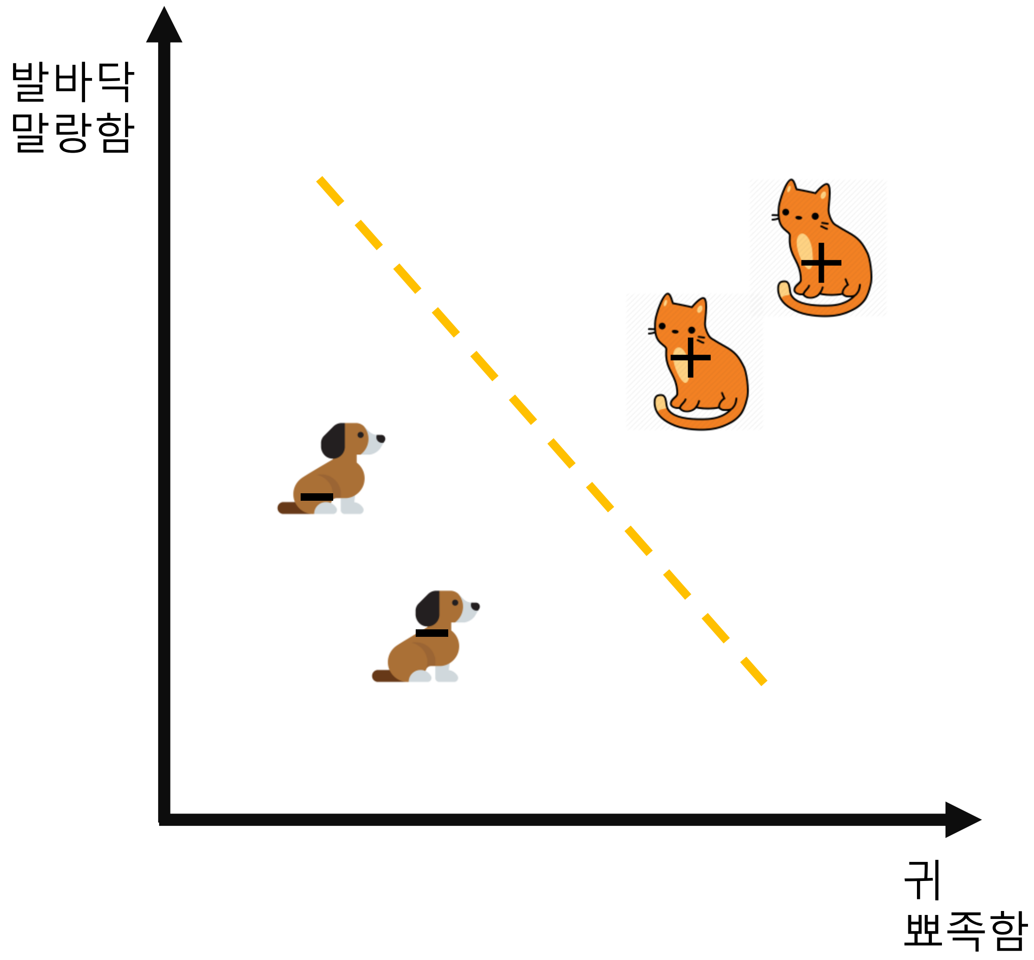
공간을 어떻게 구분할 것인가?

회귀, 분류, 이상치 감지, 패턴인식 등에 사용되는
- 지도학습(Supervised) 모델 -

특히 이진분류에 굉장히 효과적

Support
Vector
Machine

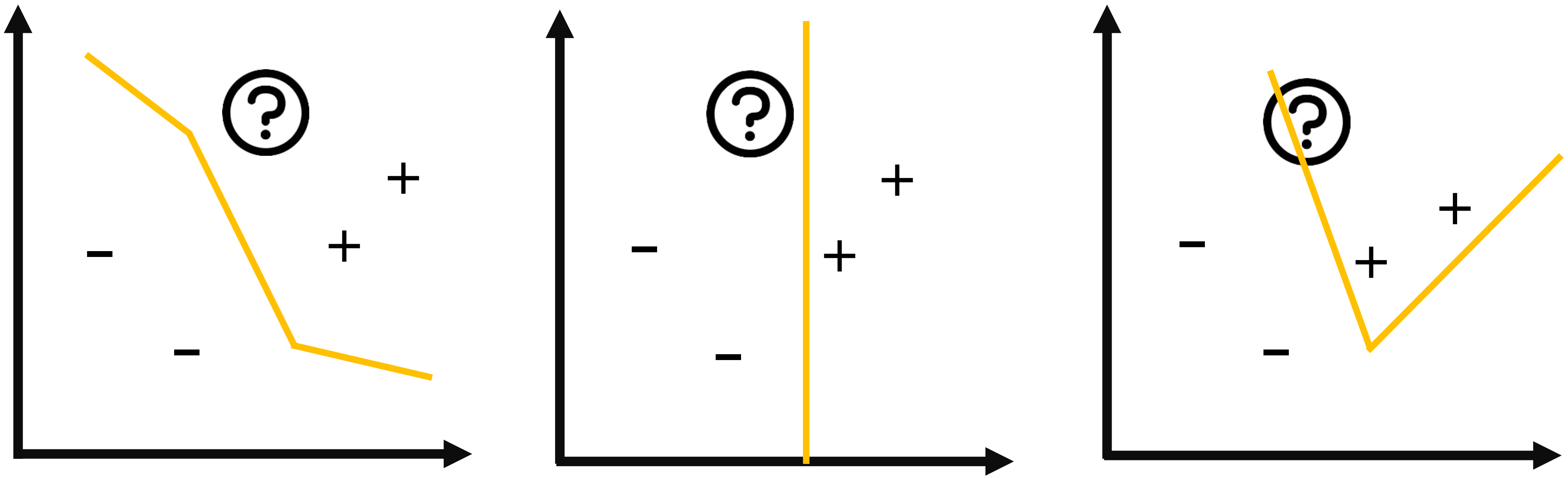
I. 모델 설명



1. 우리가'고양이'샘플과 '강아지'샘플을 구분하고 싶다면, 어떻게 나눠야 하는가?
2. 직선을 그어 나눈다면 선을 어떻게 그을 것인가?

Support
Vector
Machine

I. 모델 설명



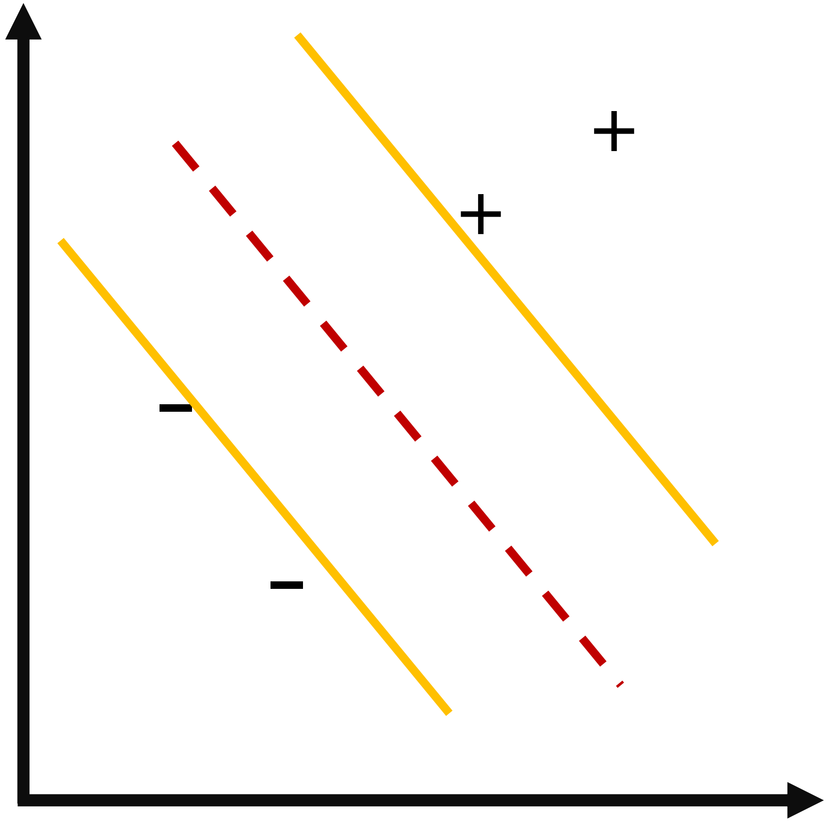
Nearest Neighbor Approach　　Decision Tree Approach　　Neural Network Approach

어떻게 그리는 게 최선일까?

SVM

Support
Vector
Machine

I. 모델 설명

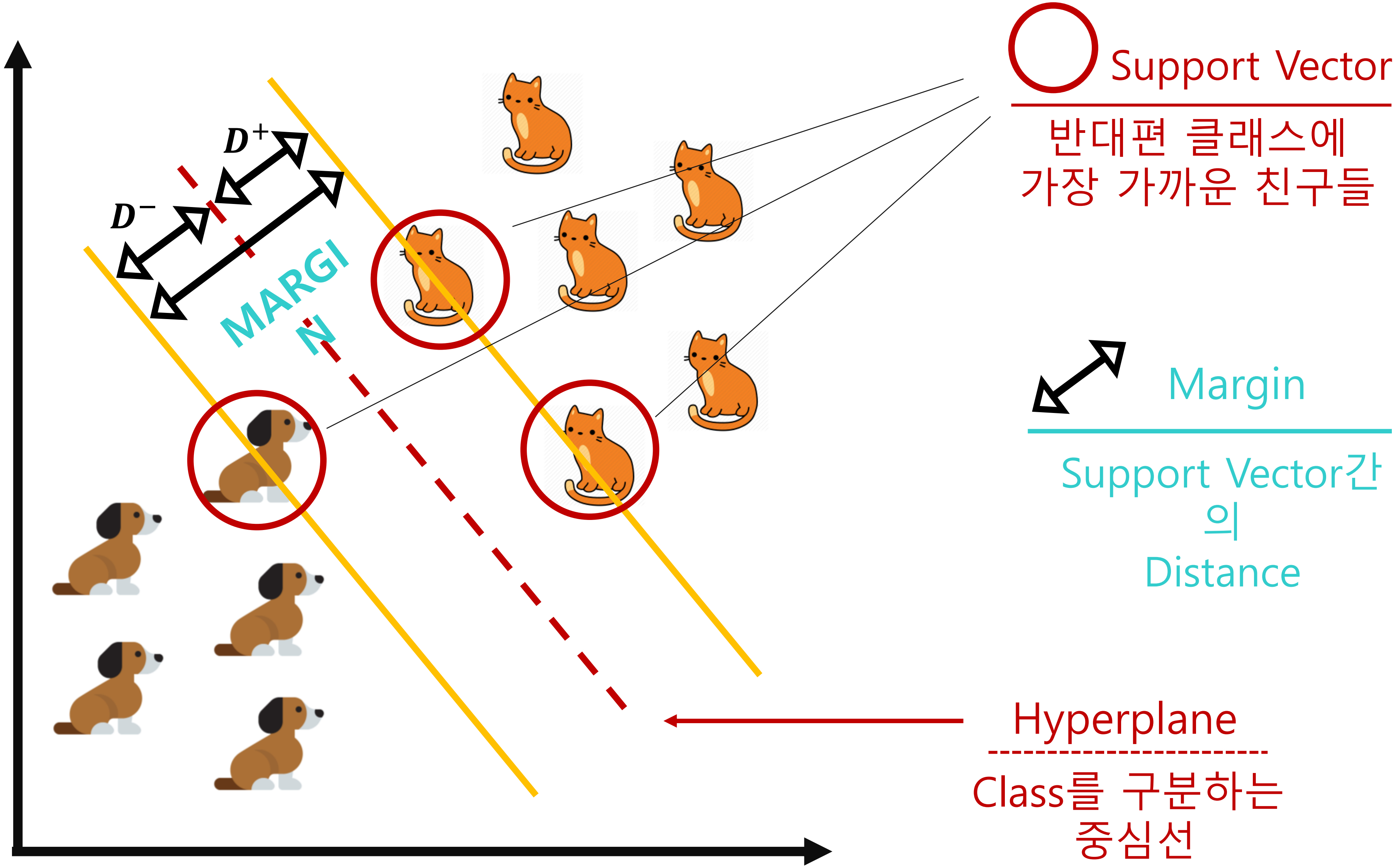


Widest Street Strategy

'+'와 '-'샘플 사이의 거리를 가장 넓게 쓰는
어떤 line(점선)을 그리면 되지 않을까?

I. 모델 설명

SVM


D^+
D^-
MARGIN
Support Vector
반대편 클래스에
가장 가까운 친구들
Margin
Support Vector간
의
Distance
Hyperplane
Class를 구분하는
중심선

Support
Vector
Machine

II. 매개변수

4-1.Learning SVM

```
In [18]: #SVM 학습
         SVM = svm.SVC(C=1.0, kernel='linear', degree=3, gamma='auto')
         SVM.fit(Train_X,Train_Y)

Out[18]: 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=False, random_state=None, shrinking=True,
           tol=0.001, verbose=False)
```

Support
Vector
Machine

II. 매개변수

Parameters:

C : *float, optional (default=1.0)*

Penalty parameter C of the error term.

kernel : *string, optional (default='rbf')*

Specifies the kernel type to be used in the algorithm. It must be one of 'linear', 'poly', 'rbf', 'sigmoid', 'precomputed' or a callable. If none is given, 'rbf' will be used. If a callable is given it is used to pre-compute the kernel matrix from data matrices; that matrix should be an array of shape `(n_samples, n_samples)`.

degree : *int, optional (default=3)*

Degree of the polynomial kernel function ('poly'). Ignored by all other kernels.

gamma : *float, optional (default='auto')*

Kernel coefficient for 'rbf', 'poly' and 'sigmoid'.

Current default is 'auto' which uses 1 / n_features, if `gamma='scale'` is passed then it uses 1 / (n_features * X.var()) as value of gamma. The current default of gamma, 'auto', will change to 'scale' in version 0.22. 'auto_deprecated', a deprecated version of 'auto' is used as a default indicating that no explicit value of gamma was passed.

coef0 : *float, optional (default=0.0)*

Independent term in kernel function. It is only significant in 'poly' and 'sigmoid'.

shrinking : *boolean, optional (default=True)*

Whether to use the shrinking heuristic.

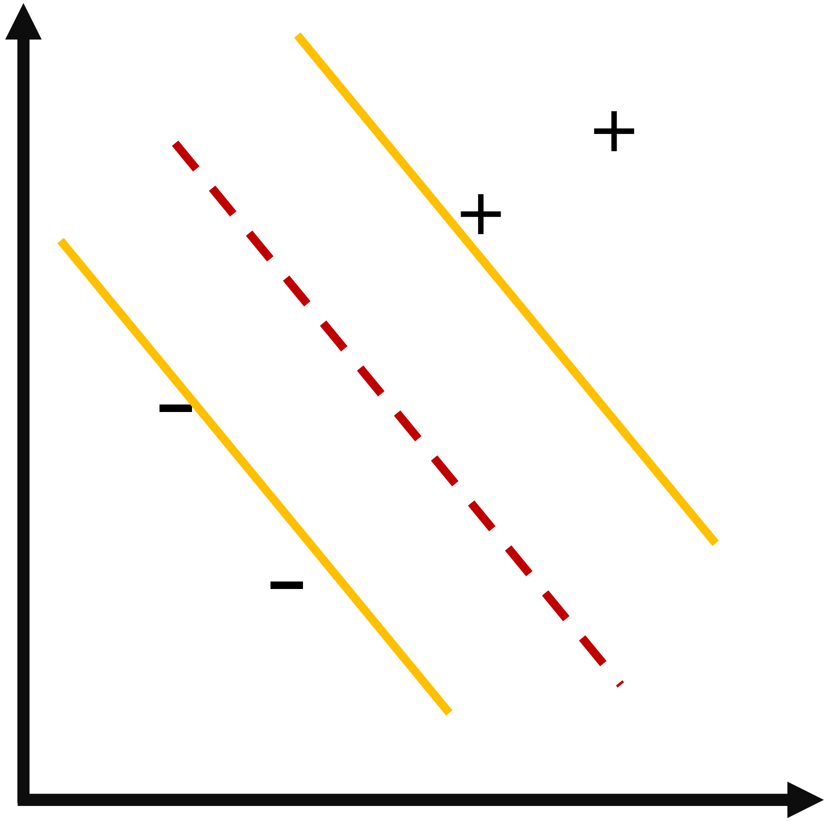
probability : *boolean, optional (default=False)*

Whether to enable probability estimates. This must be enabled prior to calling `fit`, and will slow down that method.

Support
Vector
Machine

II. 매개변수 - Kernel

Linear SVM

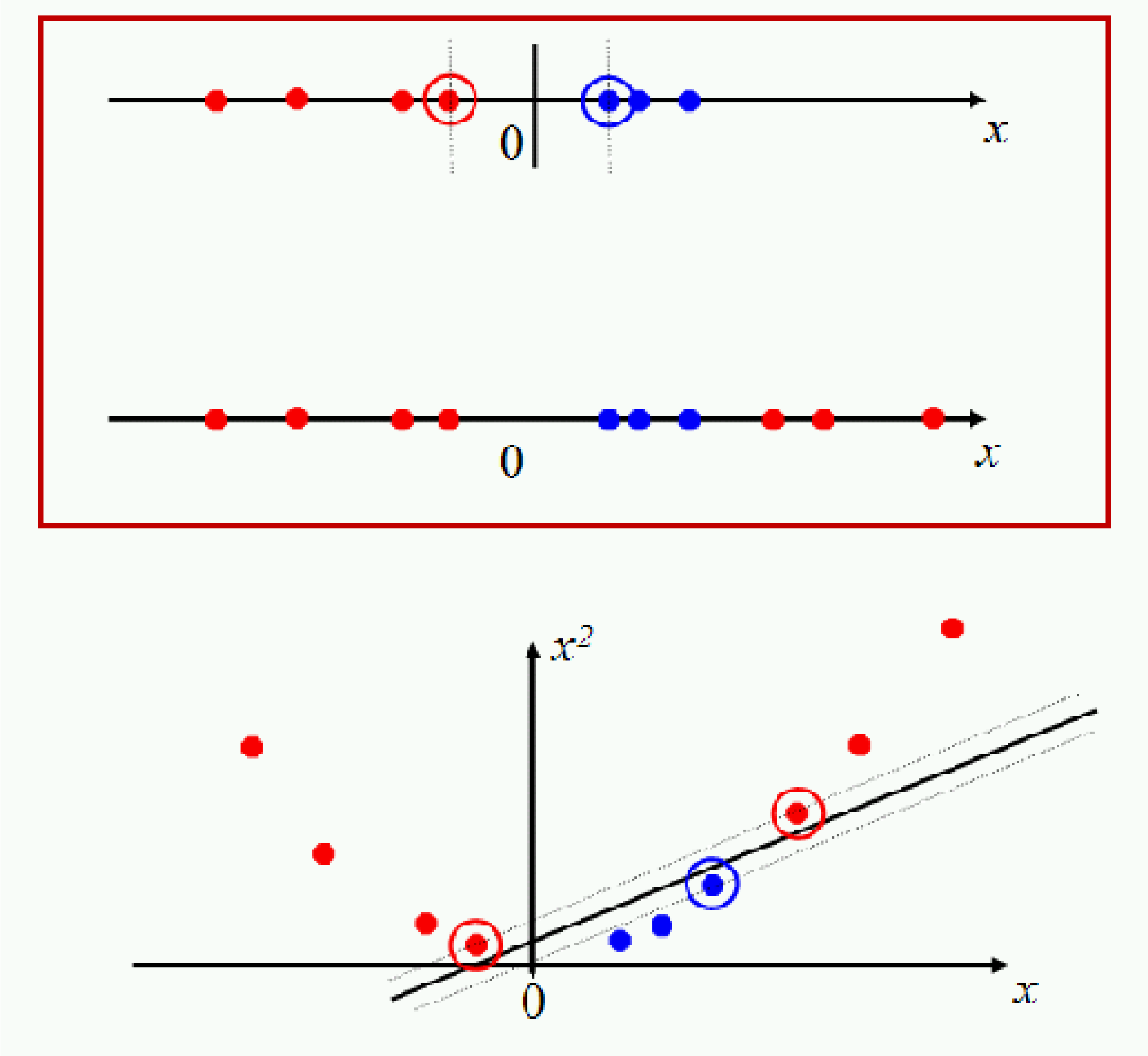


Linearly Saperatable

Support
Vector
Machine

II. 매개변수

Non Linear SVM

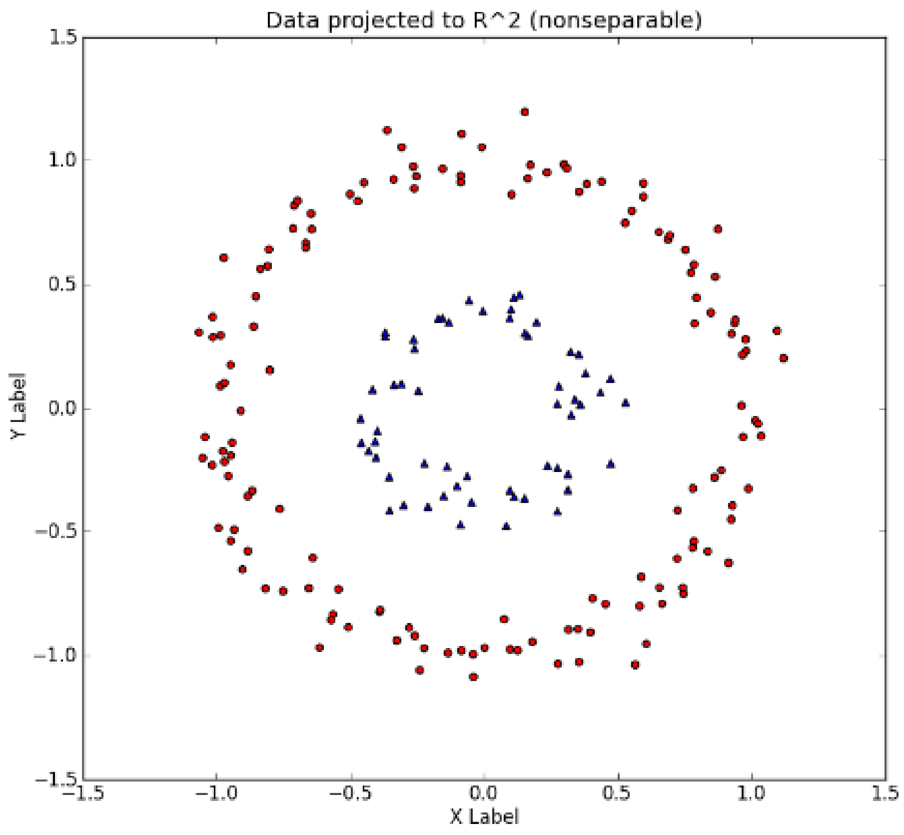


1D to 2D
(Polynomial)

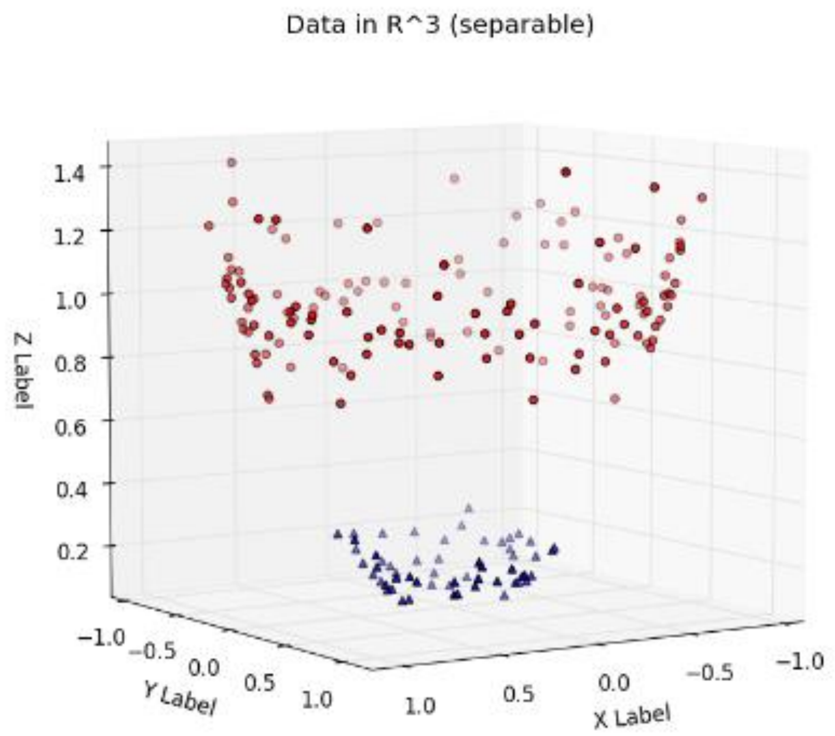
Support
Vector
Machine

II. 매개변수

Non Linear SVM



R^2

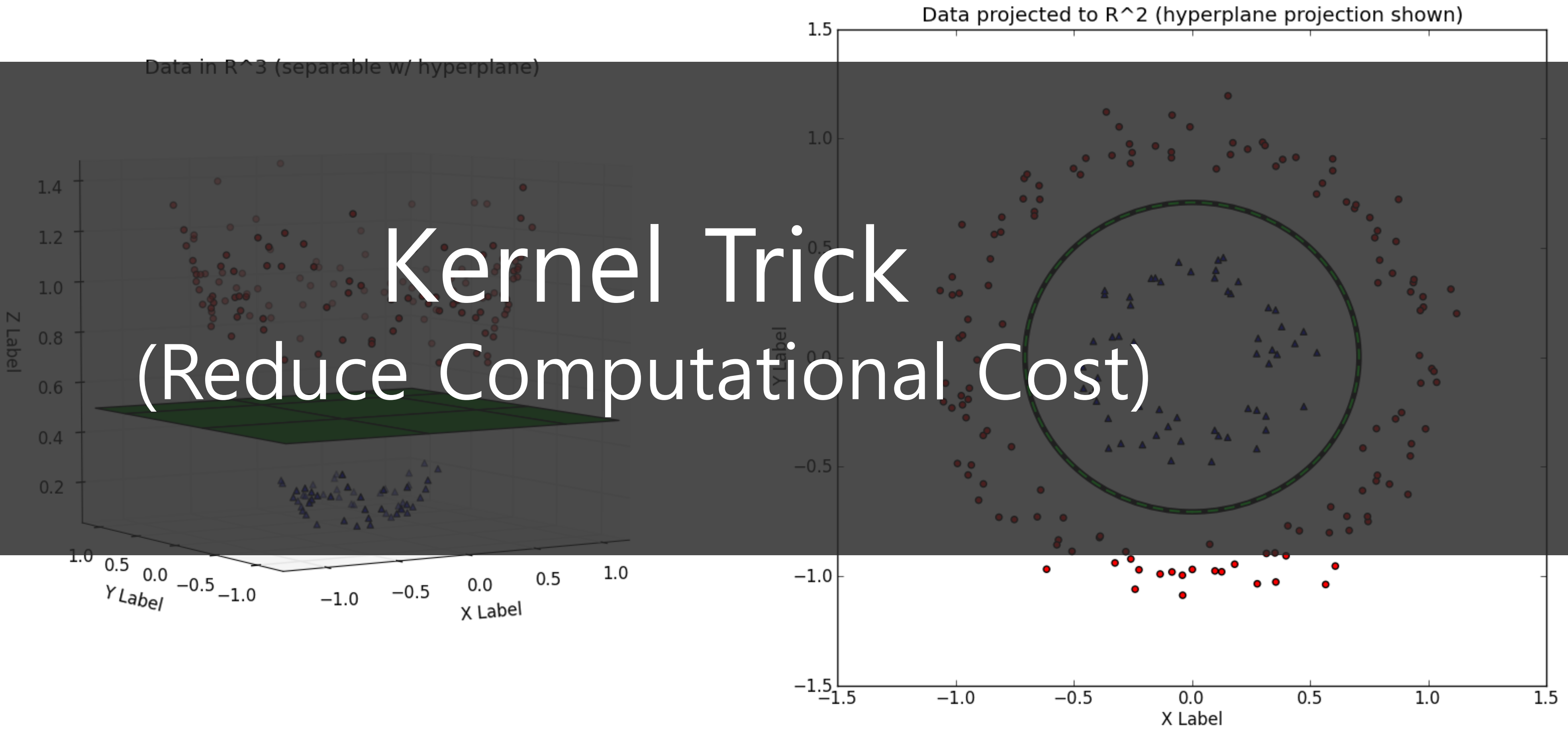


R^3

2D to 3D

II. 매개변수

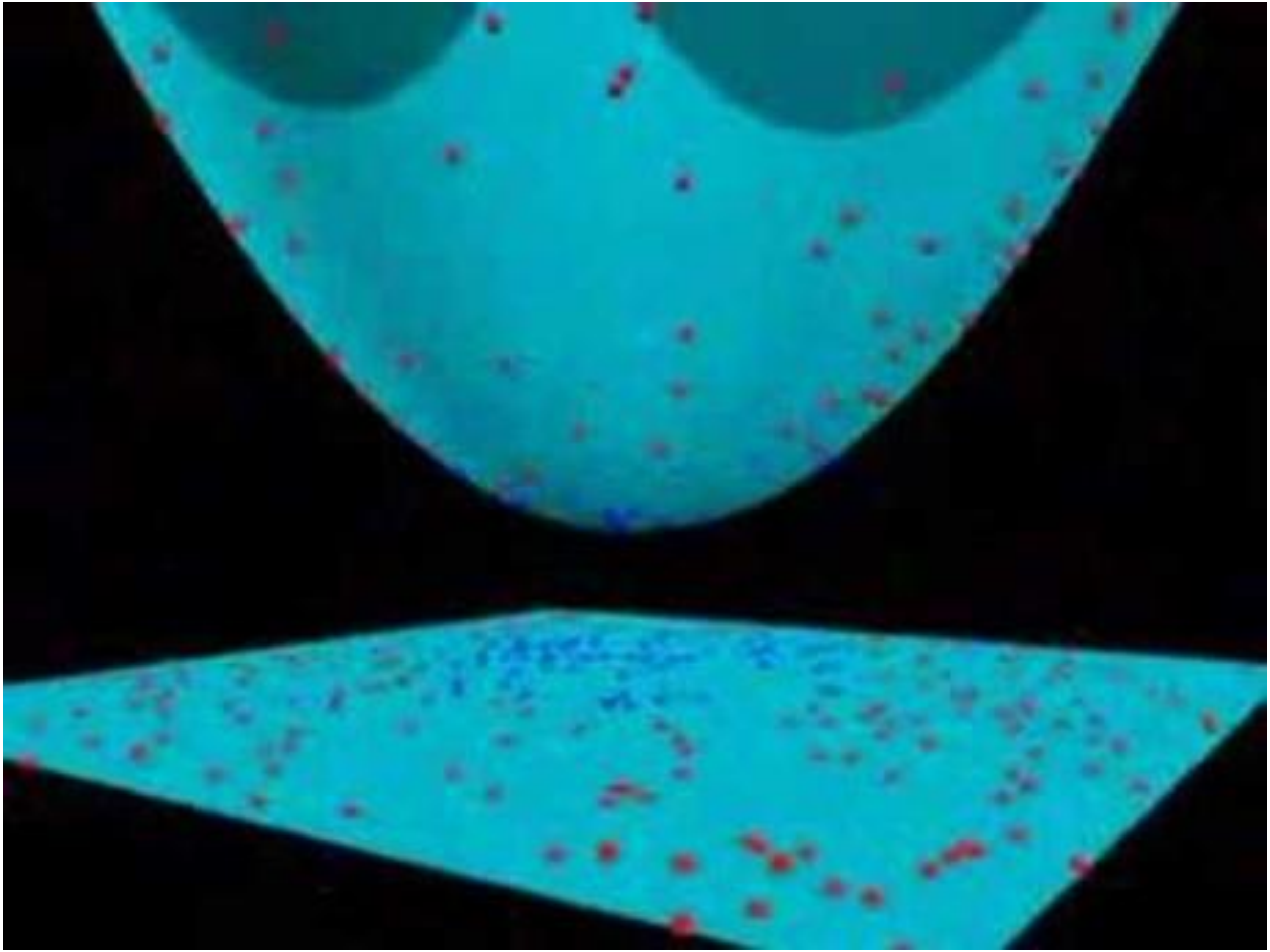
Non Linear SVM



Kernel Trick
(Reduce Computational Cost)

2D to 3D

Support Vector Machine



Support
Vector
Machine

II. 매개변수

많이 사용되는 커널

>> LINEAR KERNEL

텍스트마이닝에서
주로 쓰이는 커널

>> POLYNOMIAL KERNEL

>> RADICAL BASIS FUNCTIONAL KERNEL(RBF)
혹은 GAUSSIAN KERNEL

>> SIGMOID KERNEL

성능이 가장 좋아
자주 쓰이는 커널

Support
Vector
Machine

II. 매개변수

Text Categorization with Support Vector Machines: Learning with Many Relevant Features

Thorsten Joachims

Universität Dortmund
Informatik LS8, Baroper Str. 301
44221 Dortmund, Germany

Most text categorization problems are linearly separable: All Ohsumed categories are linearly separable and so are many of the Reuters (see section 5) tasks. The idea of SVMs is to find such linear (or polynomial, RBF, etc.) separators.

These arguments give theoretical evidence that SVMs should perform well for text categorization.

Support
Vector
Machine

II. 매개변수

Type of Kernel	Inner product kernel $K(\vec{x},\vec{x}_i),\ i=1,2,\ldots,N$	Comments
Polynomial Kernel	$K(\vec{x},\vec{x}_i)=\left(\vec{x}^T\vec{x}_i+\theta\right)^d$	Power p and threshold θ is specified a priori by the user
Gaussian Kernel	$K(\vec{x},\vec{x}_i)=e^{-\frac{1}{2\sigma^2}\lVert\vec{x}-\vec{x}_i\rVert^2}$	Width σ^2 is specified a priori by the user
Sigmoid Kernel	$K(\vec{x},\vec{x}_i)=tanh(\eta\,\vec{x}\,\vec{x}_i+\theta)$	Mercer's Theorem is satisfied only for some values of η and θ
Kernels for Sets	$K(\chi,\chi')=\sum_{i=1}^{N_\chi}\sum_{j=1}^{N_{\chi'}}k(x_i,x'_j)$	Where $k(x_i,x'_j)$ is a kernel on elements in the sets χ, χ'
Spectrum Kernel for strings	count number of substrings in common	It is a kernel, since it is a dot product between vectors of indicators of all the substrings.

Support
Vector
Machine

II. 매개변수

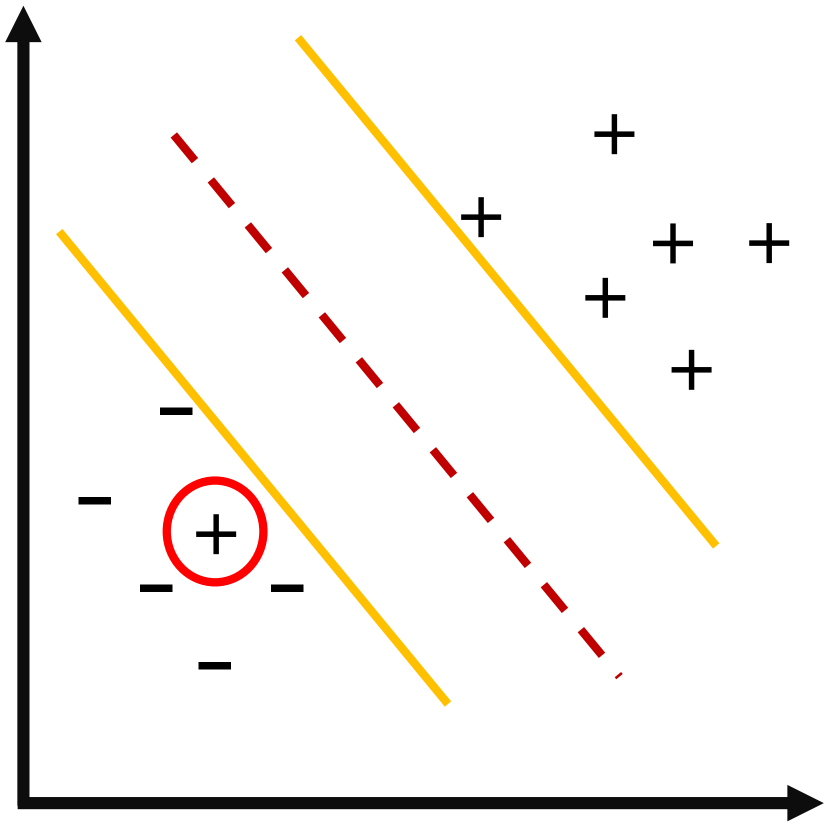
4-1.Learning SVM

```
In [18]: #SVM 학습
SVM = svm.SVC(C=1.0, kernel='linear', degree=3, gamma='auto')
SVM.fit(Train_X,Train_Y)

Out[18]: 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=False, random_state=None, shrinking=True,
  tol=0.001, verbose=False)
```

Support
Vector
Machine

II. 매개변수

C(Cost) & gamma(margin의 영향력)



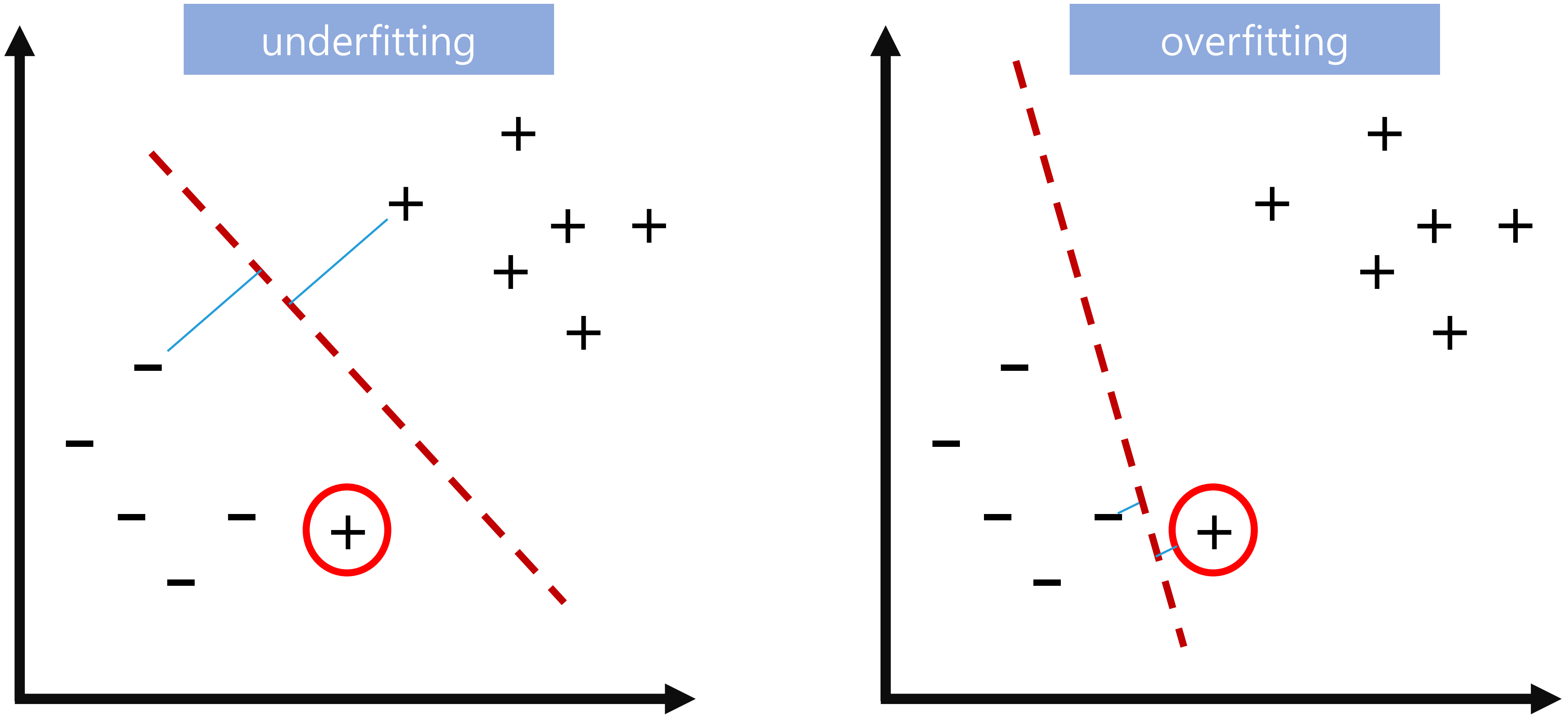
Outlier가 존재하는 경우

Support
Vector
Machine

II. 매개변수

C(Cost) - 이상치에 대한 허용 정도

얼마나 많은 샘플이 다른 클래스에 놓이는 것을 허용할 것인가



C가 작을수록 많이 허용

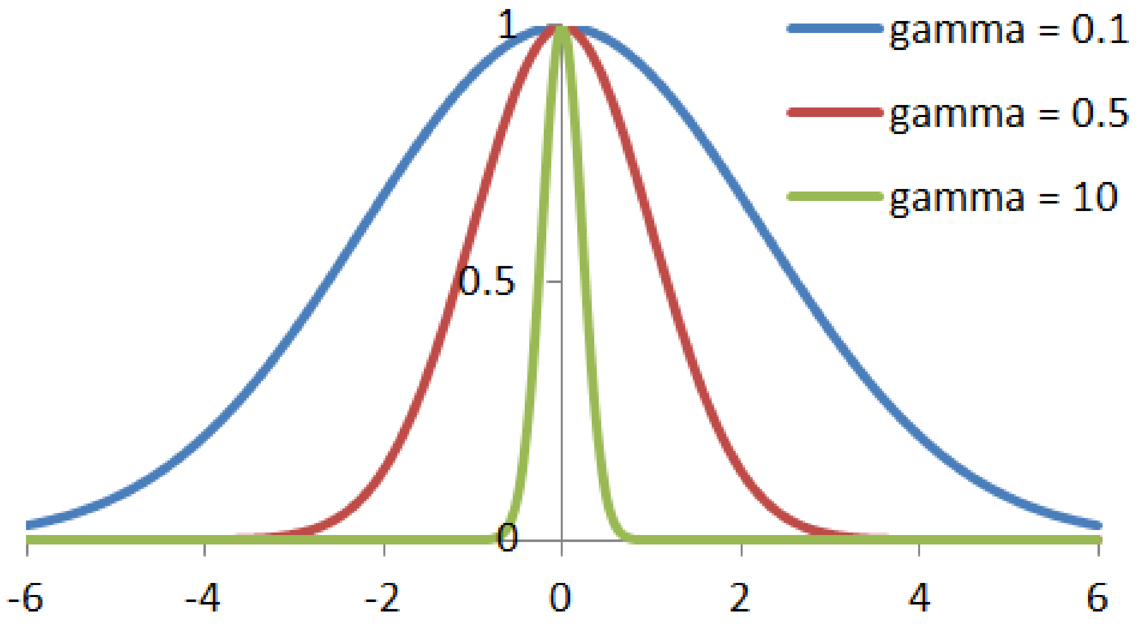
C가 높을수록 적게 허용

Support
Vector
Machine

II. 매개변수

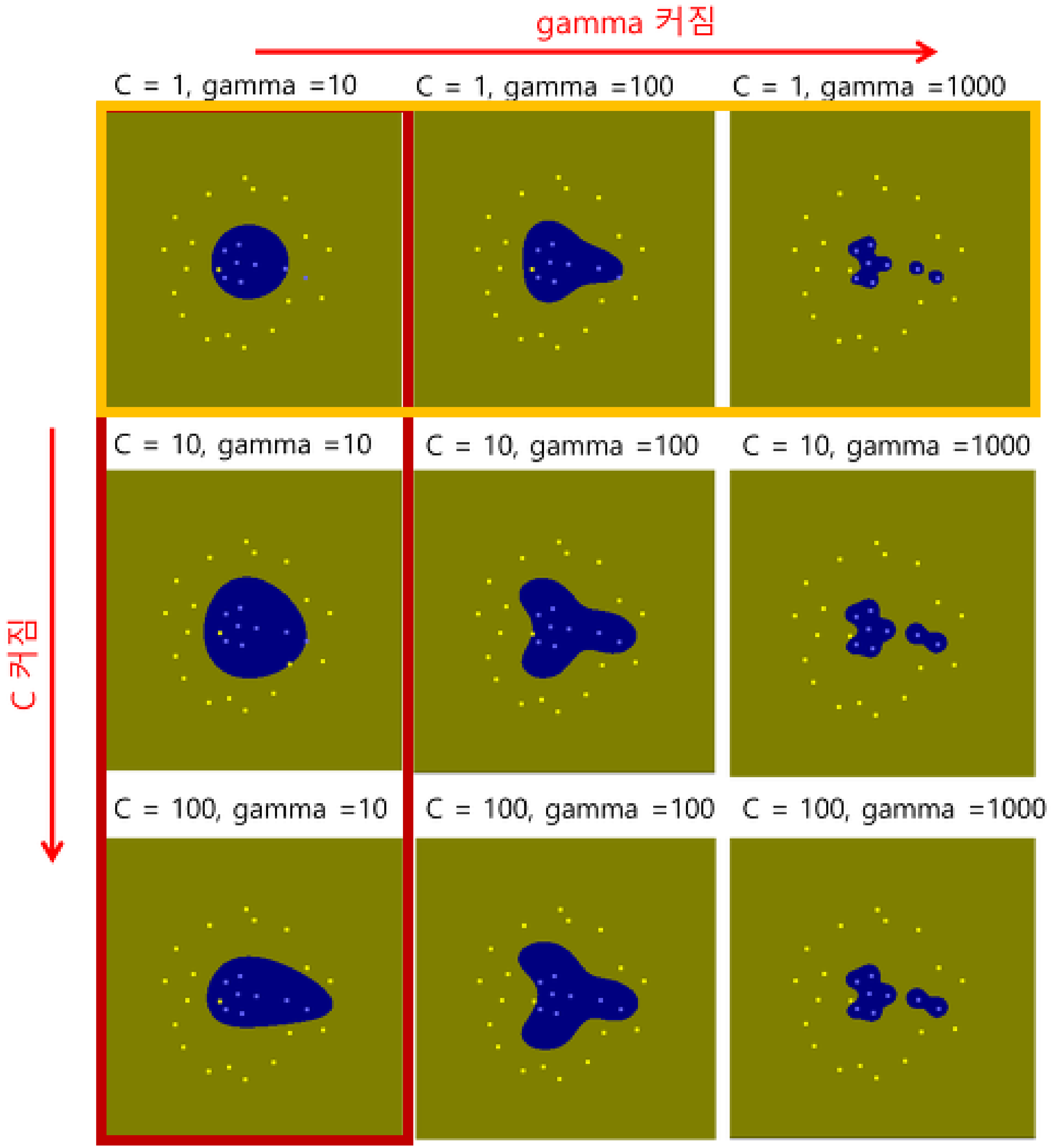
gamma - 결정 경계의 곡률

데이터 포인터들의 영향력을 행사하는 거리



Gamma 크기에 따른 가우시안 함수의 모형
<클수록 거리 감소 / 작을수록 거리 증가>

Support Vector Machine



C
- 커질수록 적게 허용
- 작을수록 많이 허용

gamma
- 커질수록 거리 감소
- 작을수록 거리 증가

Support
Vector
Machine

II. 매개변수

4-1.Learning SVM

```
In [18]:
#SVM 학습
SVM = svm.SVC(C=1.0, kernel='linear', degree=3, gamma='auto')
SVM.fit(Train_X,Train_Y)
```

```
Out[18]: 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=False, random_state=None, shrinking=True,
  tol=0.001, verbose=False)
```

입력 X

파라미터가 너무 많으면 최적의 조합을 찾기 어려워진다.

➔ 때문에 가장 중요한 kernel, C, gamma (+degree) 정도만 입력해준다.

Support
Vector
Machine

III. 조사내용 – Feature & Attribute & Dimension

Attribute(속성)

One particular Type of Data

Ex) 사람을 예시로
- 키
- 몸무게
- 나이
- 성별
- etc…

Dimension(차원)

대부분의 경우,
차원 = 속성(Attribute)의 수

- "데이터 벡터의 2차원은 사람의 연령" -

의 형태로도 사용될 수 있지만,
굉장히 드물게 나타나는 편이다.

Support
Vector
Machine

III. 조사내용 – Feature & Attribute & Dimension

Feature - 문맥에 따라 여러 가지 의미를 갖는다.

1) Attribute

2) 여러 가지 학습 모델로부터 생성된 데이터의 내부 표현
- 예를 들어, 인공신경망은 Attributes의 조합이나, 다른 feature들의 조합을 통해 feature를 추출한다.

3) 커널 메서드에 의해 유도된 데이터의 이론적 표현 (hypothethical representation)
_ in PCA, K-means, SVM 등…

Support
Vector
Machine

III. 조사내용 – Feature & Attribute & Dimension

Feature - 문맥에 따라 여러 가지 의미를 갖는다.

예를 들어)

1. 어떤 물건 X를 여러 가지의 속성(attributes)으로 표현한다고 하자.
   - Feature Extraction의 첫 번째 스텝 ➔ 어떻게 표현할 것인가?
   - 위 속성들 자체를 Feature 라고 부르기도 한다.

2. 주어진 속성들은 차원으로 표현된다.
   - Number of Attributes, Extracted features

3. 모델 학습을 위해 추상화된 표현(Cosine_sim / Tfidf_vect)을 사용
   - (extracted features from Features.)

Support
Vector
Machine

III. 조사내용 – Sparse Dictionary Learning (Sparse Coding)

```
In [325]: Train_X_Tfidf[0:1][0] ## Atom

Out[325]: <1x100000 sparse matrix of type '<class 'numpy.float64'>'
                  with 5 stored elements in Compressed Sparse Row format>
```

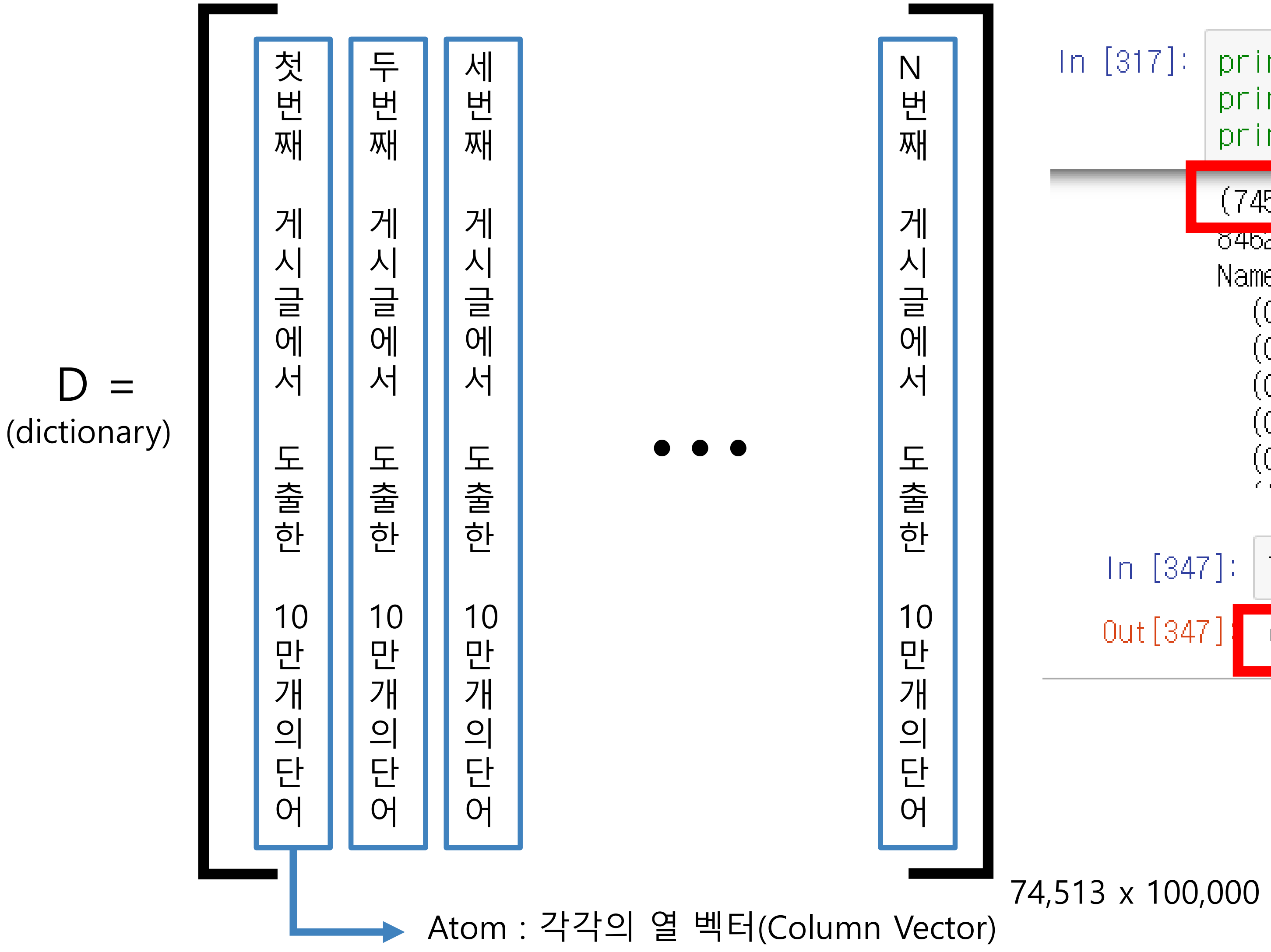
```
print(type(Train_X_Tfidf))

<class 'scipy.sparse.csr.csr_matrix'>
```

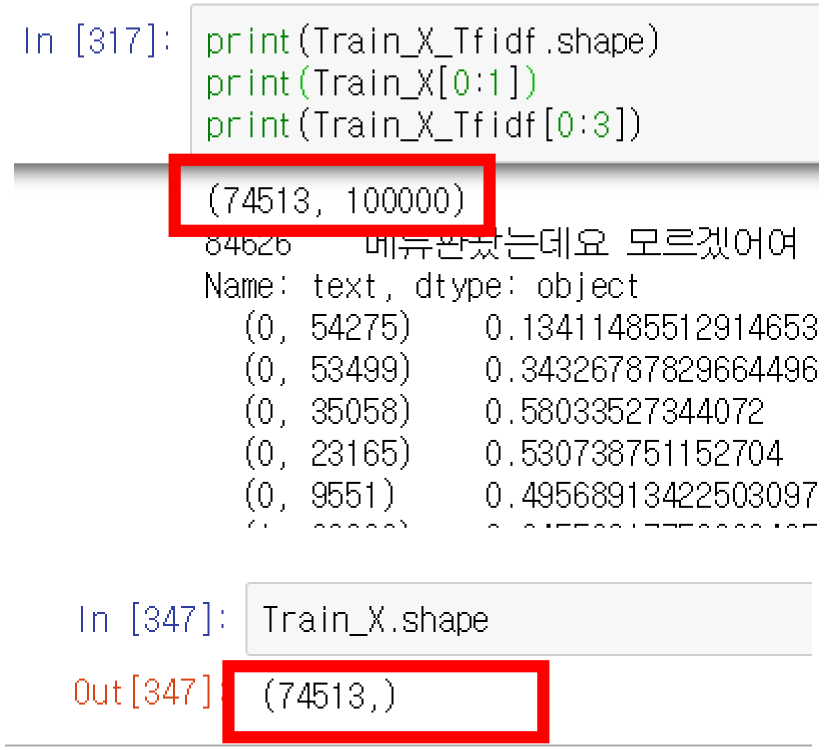
희소행렬: 대부분의 원소가 0 or 1

Sparse - 희소한, 드문
Dictionary - 사전
Learning – 학습

Support
Vector
Machine

III. 조사내용 – Sparse Dictionary Learning (Sparse Coding)



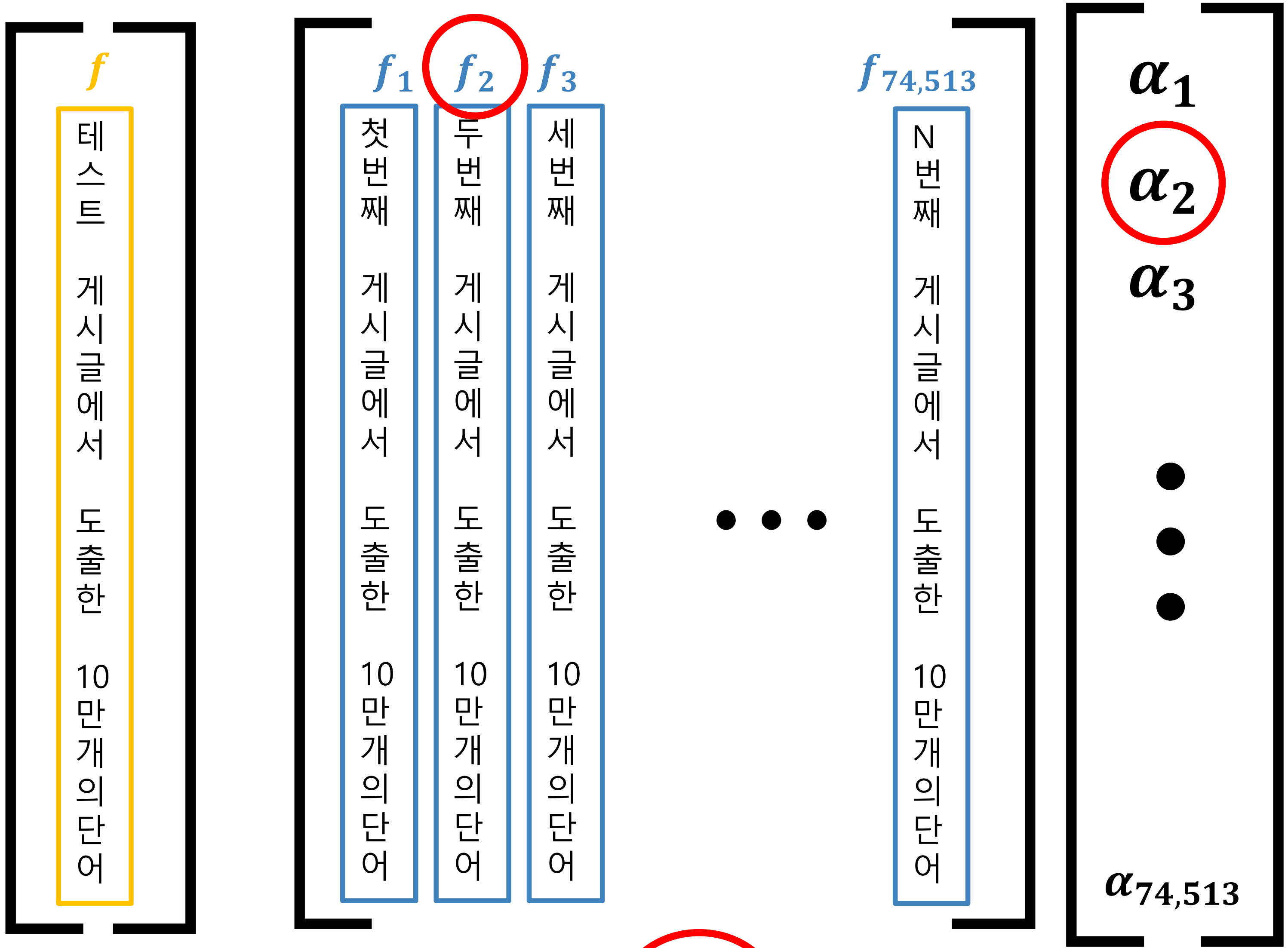


```
In [317]: print(Train_X_Tfidf.shape)
          print(Train_X[0:1])
          print(Train_X_Tfidf[0:3])

(74513, 100000)
84626     메뉴판찼는데요 모르겠어여
Name: text, dtype: object
  (0, 54275)    0.13411485512914653
  (0, 53499)    0.34326787829664496
  (0, 35058)    0.58033527344072
  (0, 23165)    0.530738751152704
  (0, 9551)     0.49568913422503097
```

```
In [347]: Train_X.shape
Out[347]: (74513,)
```

Support
Vector
Machine

III. 조사내용 – Sparse Dictionary Learning (Sparse Coding)



$$f = \alpha_1 f_1 + \alpha_2 f_2 + \cdots + \alpha_n f_n$$

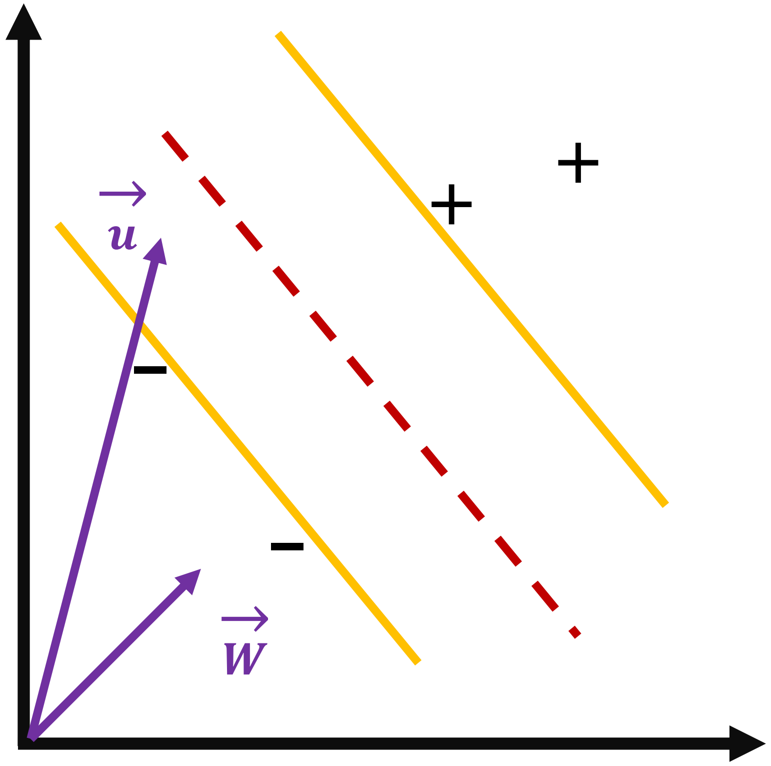
Support
Vector
Machine

III. 조사내용

III. 조사내용 – Text Categorization with SVM

Support
Vector
Machine

I. 모델 설명



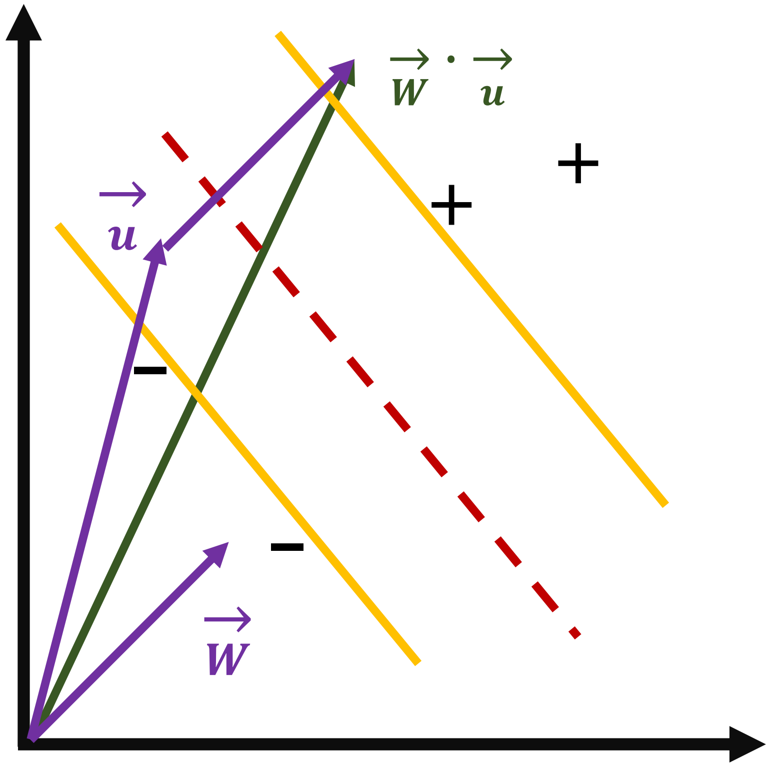
$\vec{w}$: 중심선에 직교하는 벡터 (길이는 아직 모름)

$\vec{u}$: 모르는 샘플

Support
Vector
Machine

I. 모델 설명

Decision Rule



(1) 내적을 이용한 거리계산

$$\vec{w} \cdot \vec{u} + b \geq 0 \quad then +$$

I. 모델 설명

Decision Rule

(1) 내적을 이용한 거리계산

Sample

$$\overrightarrow{\boldsymbol{W}} \cdot \overrightarrow{\boldsymbol{x+}} + \boldsymbol{b} \geq \mathbf{1}$$

$$\overrightarrow{\boldsymbol{W}} \cdot \overrightarrow{\boldsymbol{x-}} + \boldsymbol{b} \leq -\mathbf{1}$$

Support
Vector
Machine

참고 자료

Lec. 16 Learning: Support Vector Machines, Patrick Winston - MIT OCW 6.034 Fall 2010

Support Vector Machine (SVM) - Fun and Easy Machine Learning – on Youtube

Machine Learning Series Day5 (Support-Vector Machine)

Everything You Wanted to Know about the Kernel Trick, Eric Kim – 12/20/2017

초짜 대학원생의 입장에서 이해하는 Support Vector Machine

위키피디아 – 벡터 공간

위키피디아 – Kernel Method

위키피디아 – Support-Vector Machine

https://ratsgo.github.io/machine%20learning/2017/05/30/SVM3/

https://datascienceschool.net/view-notebook/69278a5de79449019ad1f51a614ef87c/