14. 벡터 공간 분류

201372237 이선아

목차

1. Introduce
2. Rocchio 분류
3. kNN 분류
4. 선형 vs 비선형 분류기
5. 2-범주 분류 이상의 분류
6. 편향-분산 반비례

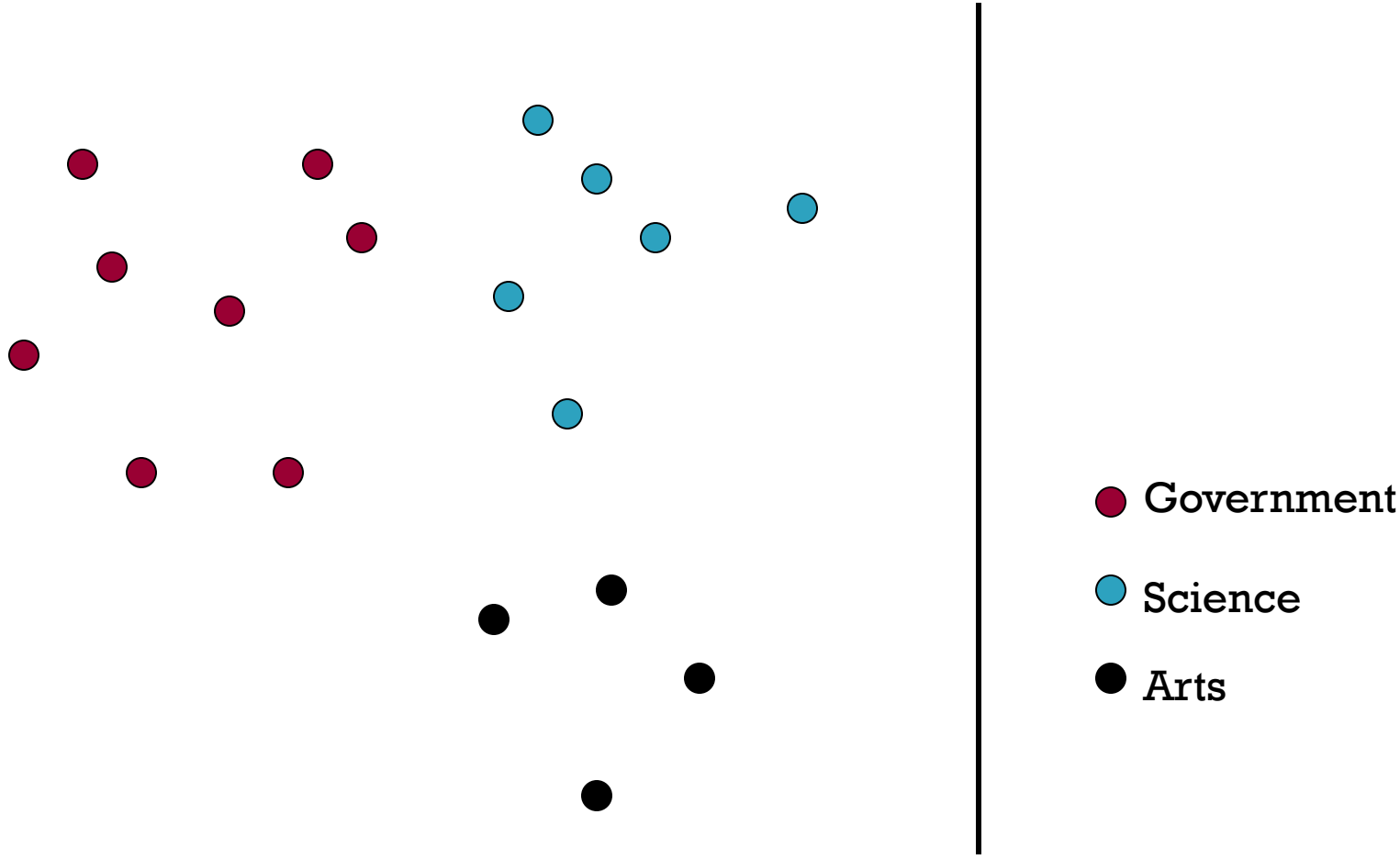
Introduce

- 문헌은 하나의 벡터
- 각 용어는 벡터의 원소
- 원소 값은 tf-idf 가중치와 같은 실수형

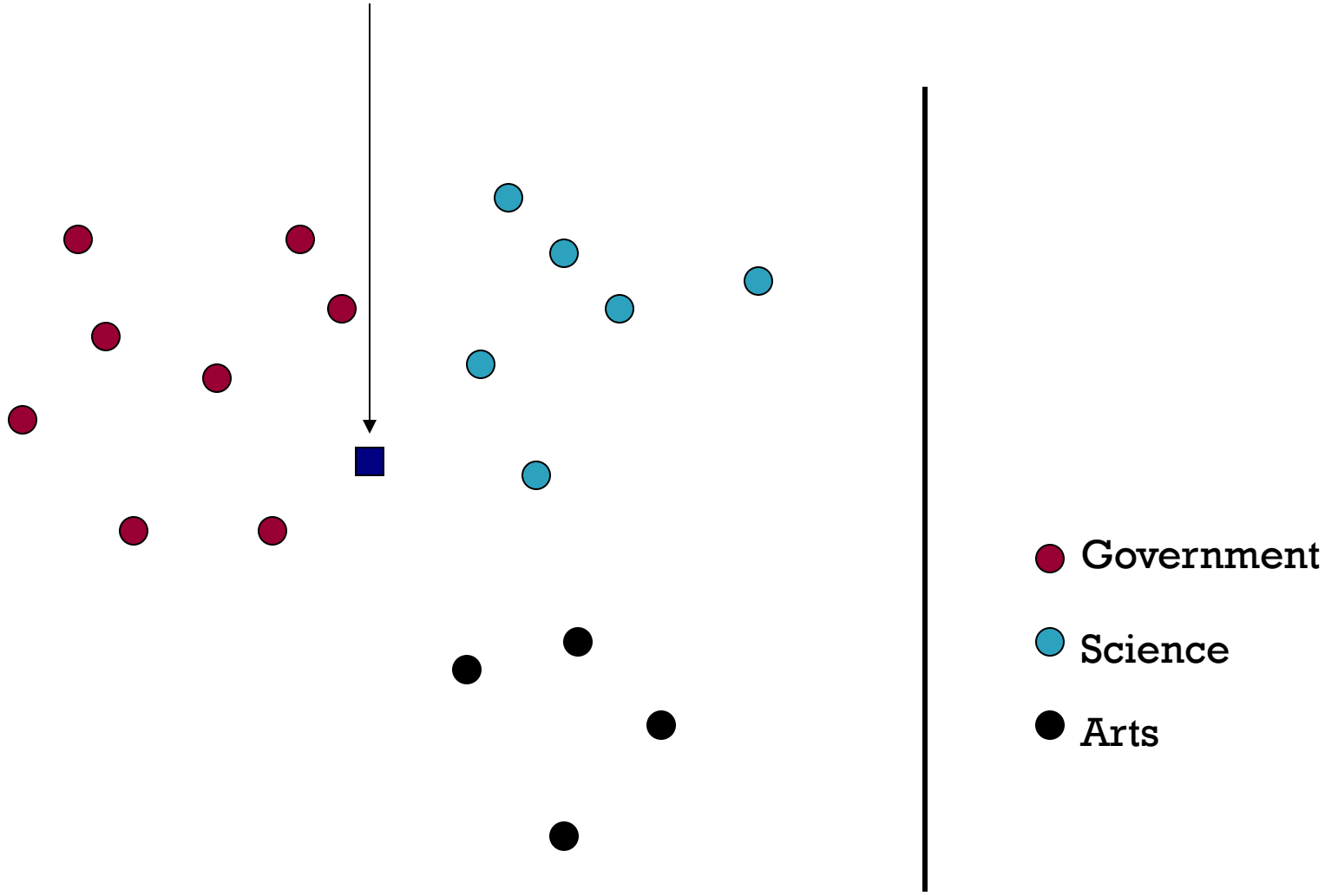
- **인접 가설**: 같은 범주에 속하는 문헌은 인접한 지역에 있고,
  다른범주에 속하는 문헌과는 서로 겹치지 않는다.

- **문헌의 경계를 결정 & 벡터 공간 분류법 소개**
  - Rocchio 분류
  - kNN 분류

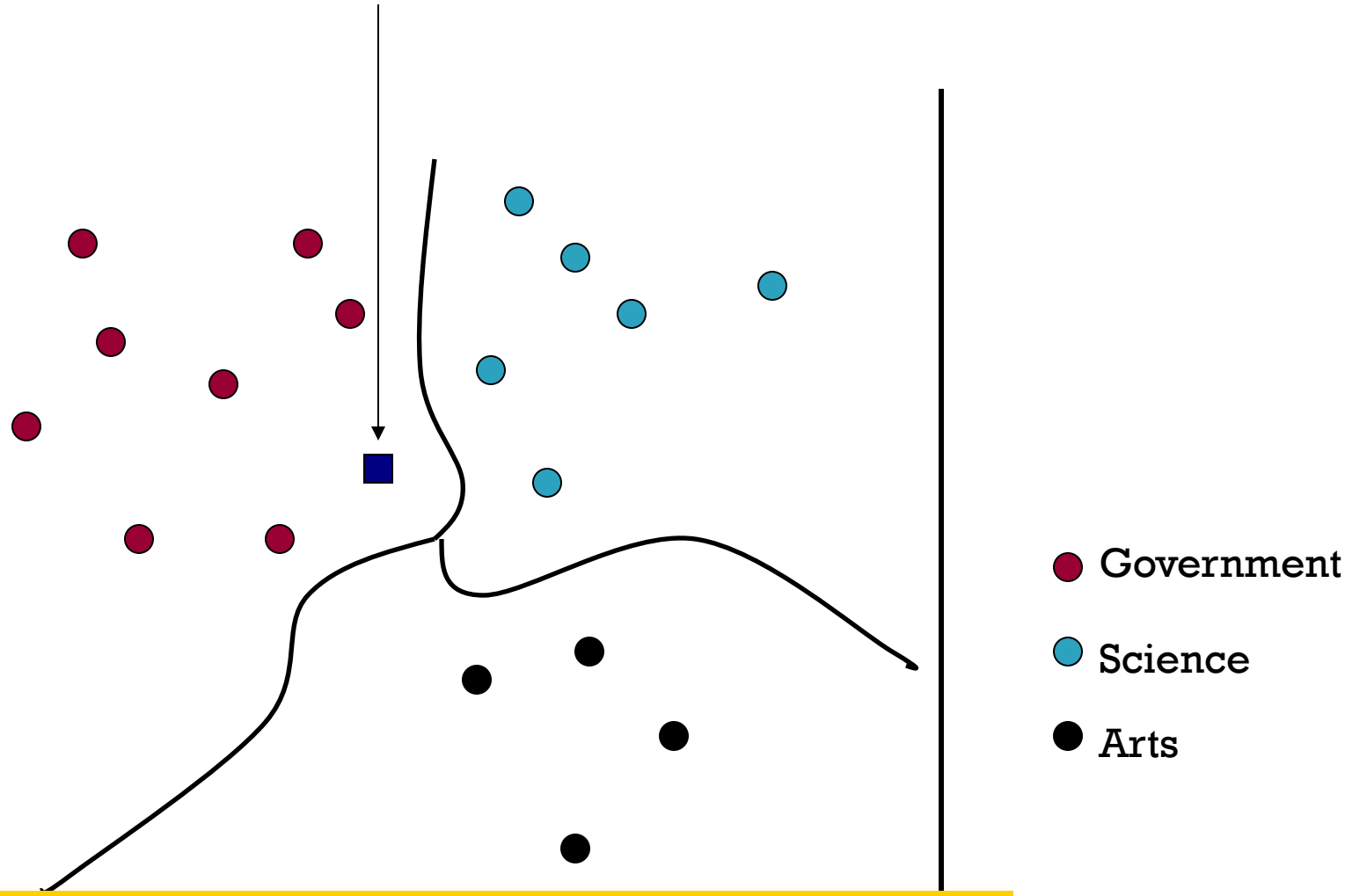
Documents in a Vector Space



Test Document of what class?

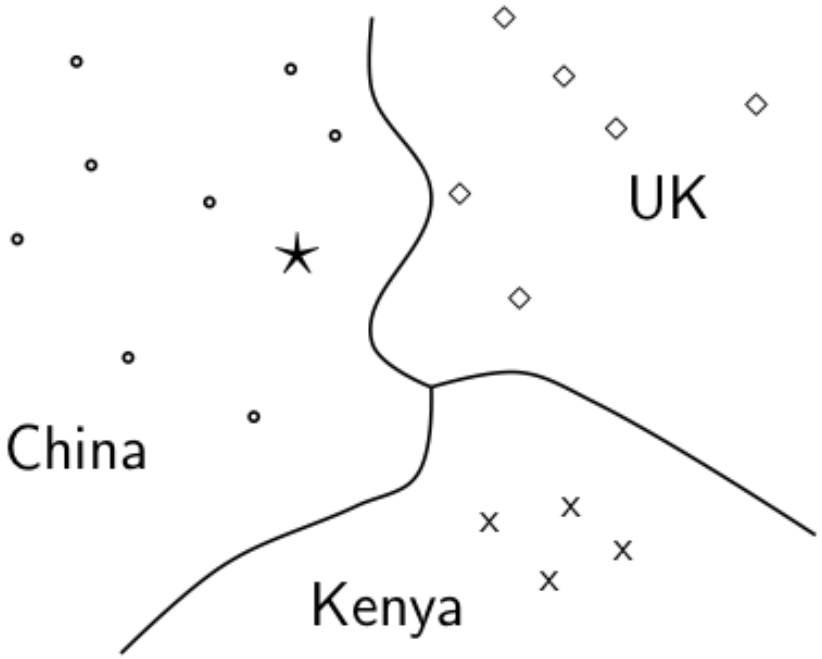


Test Document = Government



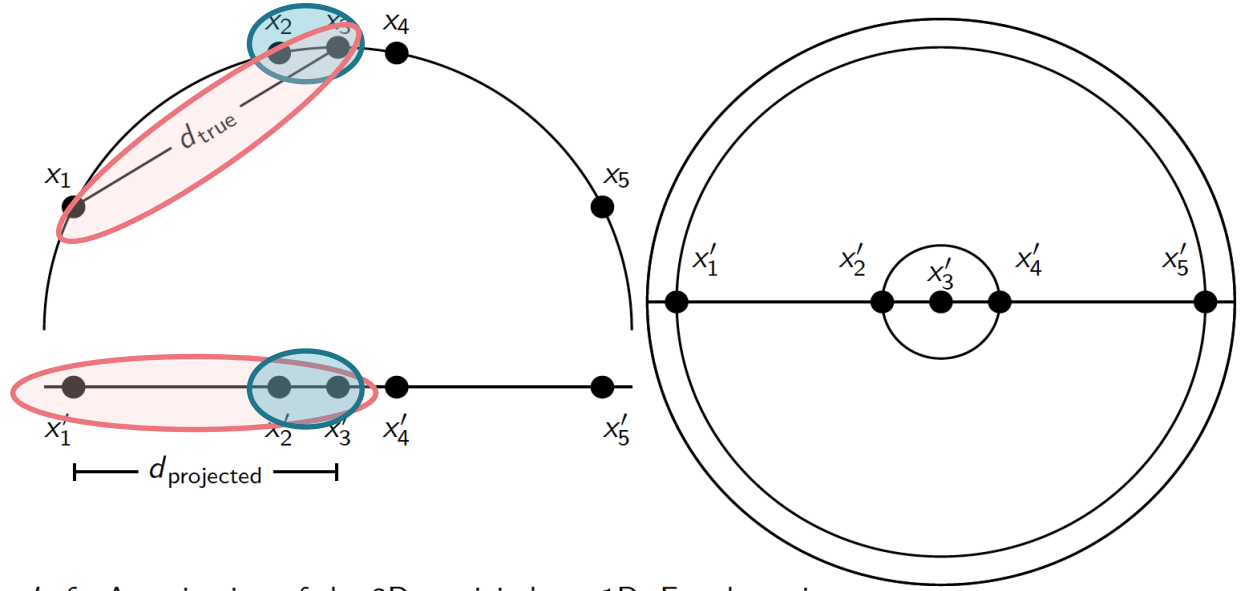
Our main topic today is how to find good separators

벡터 공간에서 문헌 표현과 관련성 척도



- 문헌 벡터들을 평면 위의 점들로 표현
- 문헌 벡터들은 길이-정규화된 단위 벡터로 표현
- 초구면을 평면으로 투영하여 표현

벡터 공간에서 문헌 표현과 관련성 척도



Left: A projection of the 2D semicircle to 1D. For the points x_1, x_2, x_3, x_4, x_5 at x coordinates $-0.9, -0.2, 0, 0.2, 0.9$ the distance $|x_2x_3| \approx 0.201$ only differs by 0.5% from $|x'_2x'_3| = 0.2$; but $|x_1x_3|/|x'_1x'_3| = d_{\text{true}}/d_{\text{projected}} \approx 1.06/0.9 \approx 1.18$ is an example of a large distortion (18%) when projecting a large area. *Right:* The corresponding projection of the 3D hemisphere to 2D.

- 벡터공간분류기는 거리를 기반으로 범주를 결정
- 기본적인 거리는 Euclidean거리 사용
- 벡터공간분류에서 벡터의 중심 혹은 평균이 매우 중요

Rocchio 분류

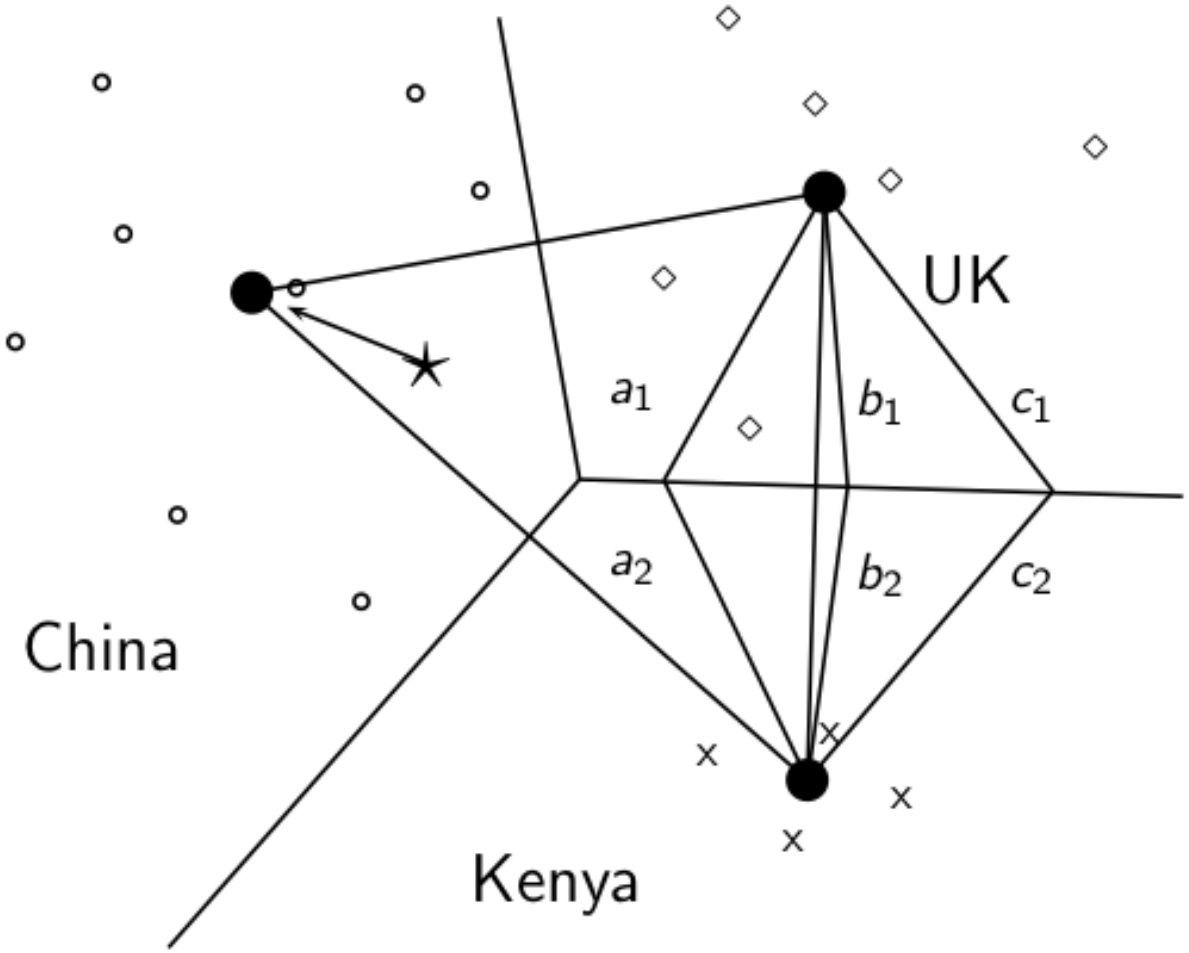
- 벡터 공간 분류에서 유사한 범주들을 Prototype **형태로** 또는 **중심을 정의하여** 범주들 사이의 **경계**를 정하는 분류 방법
- 벡터공간을 여러 영역으로 나눔 (영역 = 범주)
- 중심 : 영역에 속하는 모든 문헌의 무게 중심 or 모든 문헌의 평균

$$\vec{\mu}(c) = \frac{1}{|D_c|} \sum_{d \in D_c} \vec{v}(d)$$

- D_c : 범주 c에 속하는 문헌 집합
- $\vec{v}(d)$: 문헌 d의 정규화된 벡터

- 범주들 사이의 **경계** :
  두 범주의 중심으로부터 동일한 거리에 있는 점들의 집합(점들의 집합은 항상 직선)
- 초평면 : Rocchio분류에서 범주영역의 경계 (점들의 집합)

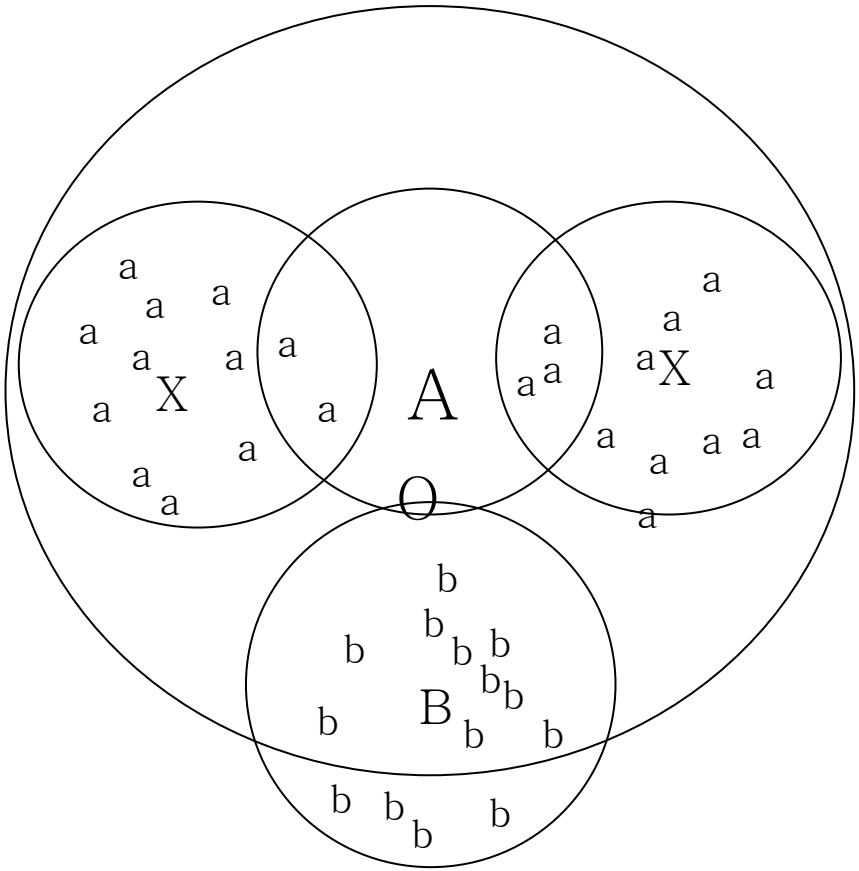
Rocchio 분류



내가 원하는 점에
가장 가까이 있는
$\vec{\mu}(c)$ 중심을 찾고,
c 를 그 점의 범주로 정
함

Rocchio 분류

- Eucliden 거리를 이용한 방법과 코사인 유사도를 이용한 방법은 서로 다른 결과를 갖을 수 있다.



- Eculiden 거리를 이용한 방법 : O는 A로 분류
- But, O = B범주

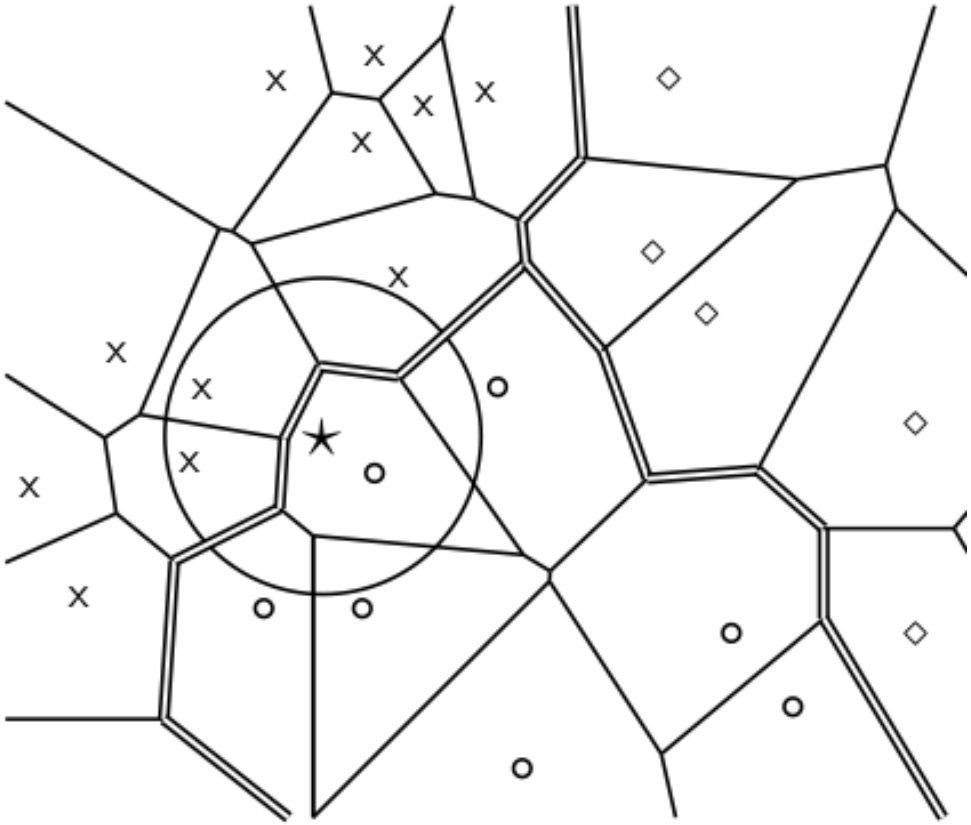
Rocchio분류법은 범주에 속하는 문헌들의 분포를 무시하고, 오로지 범주의 중심에서 떨어진 거리만 이용한다.
(다봉범주<multimodal class>는 잘 분류 하지 못한다.)

kNN 분류

- k- nearest neighbor :

  실험 문헌에 가장 근접한 K개의 문헌을 선택하여, 이들 범주중 가장 많은 범주를 선택하는 분류법.
- Rocchio와 달리 결정경계가 지역적으로 결정 됨.
- 특별한 학습이 필요 없이 바로 적용이 가능하다.
- 학습이 매우 단순하다.
  - 문헌 전처리 과정
  - k 정하기
- kNN 분류법은 Rocchio or Naïve Bayes 방법 보다 정확하다.
- 1NN (k=1) : 가장 가까이에 있는 문헌의 범주가 실험문헌의 범주가 된다.
- kNN (k>1) : 가장 가까운 여러 개의 문헌들 중 가장 많은 문헌을 범주로 정한다.
- k : 가장 가까운 여러 개의 문헌들의 수.

kNN 분류



- **k를 구하는 방법**
  - **경험적 결정**, 보통 3 or 5 (홀수로 결정)
  - **가중치 투표방법**
    실험 문헌과 학습문헌 사이의 코사인 유도를 가중치로 계산하여 구한다.

kNN 분류

- 학습 집합(문헌)이 클 경우, 복잡한 문제도 잘 처리 할 수 있다.
- But, 학습 문헌이 많을수록 효율성 (특히 속도)이 떨어질 수 있다.
- kNN은 사례기반학습.

  (학습집합에 있는 모든 예제를 기억하고 실험문헌과 학습집합에 있는 문헌들 사이의 거리를 비교하기 때문에)
- kNN은 문서 분류에서 가장 정확한 학습방법과 거의 같다. (15장)

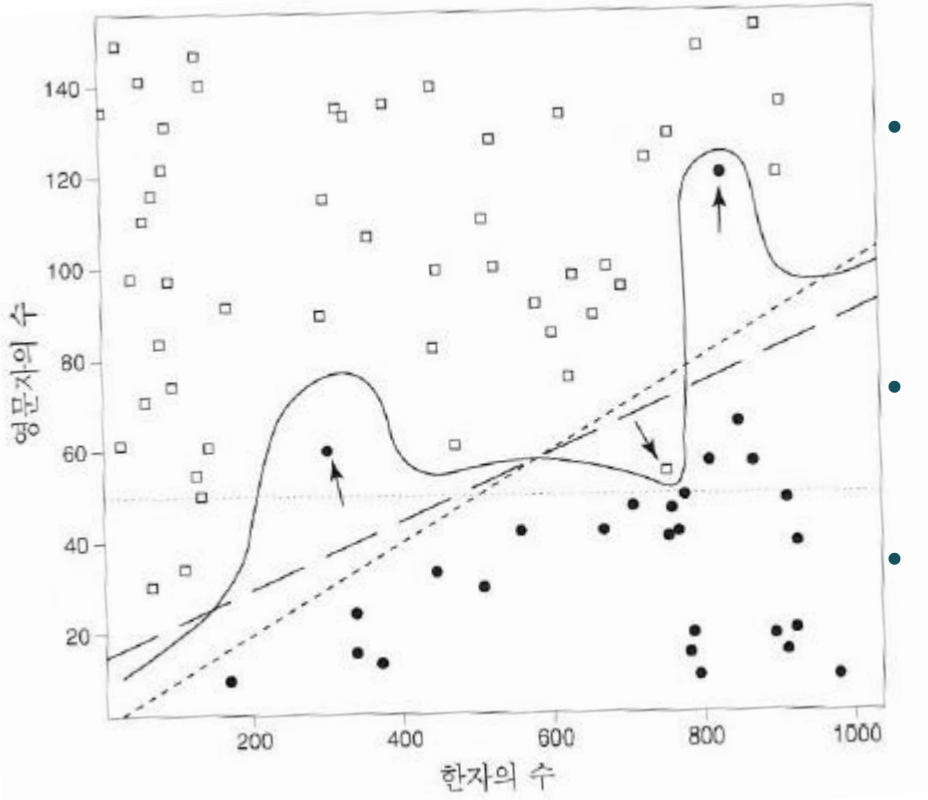
선형 vs 비선형 분류기

- 선형 분류기 : Rocchio , Naïve Bayes , **2-범주 분류기**
- 비선형 분류기 : kNN
- 2차원 공간에서 선형 분류기 = 직선
- 직선의 방정식 : $w_1x_1 + w_2x_2 = b$

  $w_1x_1 + w_2x_2 > b$: c (그렇지 않으면 c')

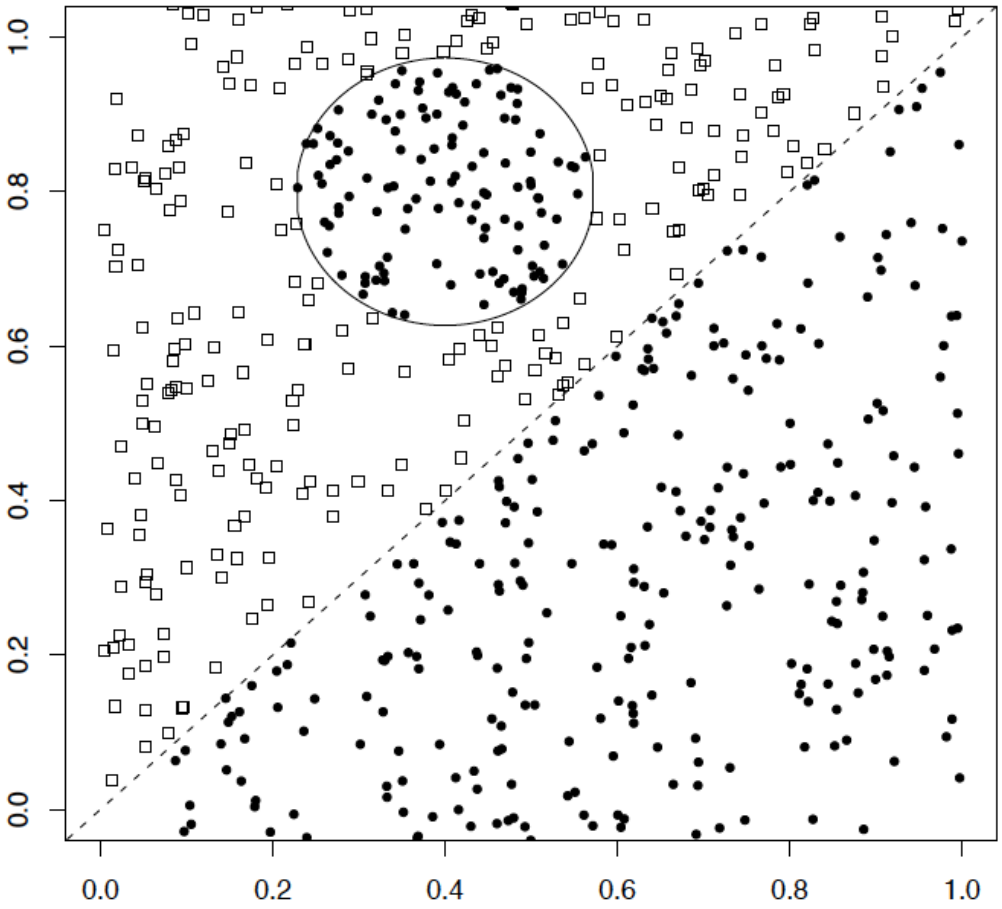
  w : 결정 경계 정의 인수 / x : 문헌의 2차원 벡터
- 결정 초평면 : 선형 분류기에서 사용되는 초평면

선형 vs 비선형 분류기



- 문헌을 분리할 수 있는 직선 : **분류 경계** (학습에 의해 추정된 경계와는 차이가 있다.)
- **잡음 문헌** : 학습 집합에 포함되면서 분류 오류를 증가시키는 문헌
- 두 범주를 완전하게 나눌 수 있는 초명면이 존재 하면 그 두 범주는 **"선형적으로 분리 가능하다"**

선형 vs 비선형 분류기

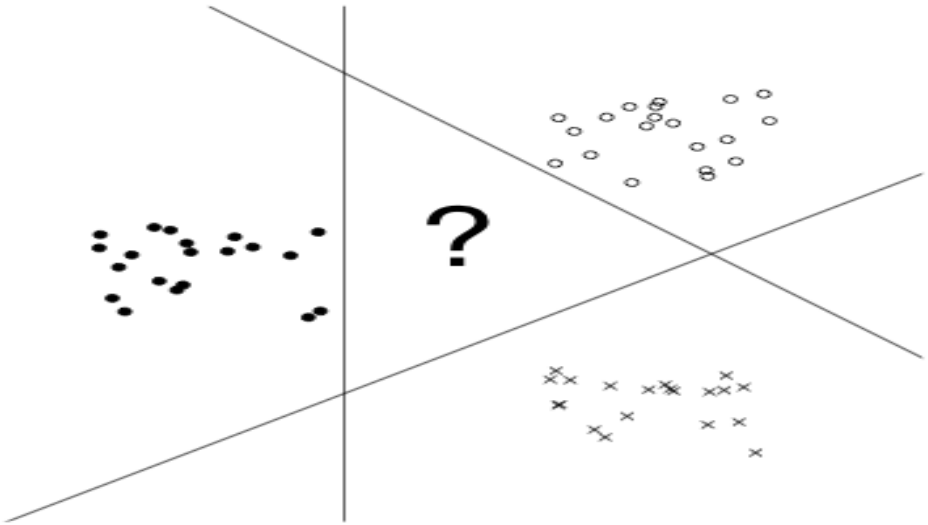


- 비선형 분류기가 필요한 형태
- kNN의 결정 경계는 지역적으로는 선형적인 조각들로 구성되나 일반적으로는 다른 복잡한 모양을 갖는다.
- 선형분리기 : 분류가 힘들다.
- 비선형 분류기 : 학습집합이 많이 있다면 정확하게 분류 가능.

- 간단한 선형적 문제-> 선형분류기를 사용
- 범주경계가 선형 초평면으로 접근이 어렵거나 비선형적 문제 -> 비선형 분류기 사용

2-범주 분류 이상의 분류

- 다범주 분류 : 서로 배타적이지 않은 범주의 분류
  - 한 문헌이 여러 범주로 분류 or 어떤 범주에도 속하지 않는 경우
- 단범주 분류(단일범주 분류) : 한 문헌이 여러 분류들 중 정확히 하나의 범주로만 분류 (kNN은 비선형 단범주 분류)



편향 –분산 반비례

- 모든 곳에 적용할 수 있는 최적 학습 방법이 존재하지 않는 이유를 설명하는 것.
- 좋은 분류기는 분산 ↓ 편향 ↓
- 편향 ↓ : 매우 다른 결정 경계를 가지는 분류 문제를 잘 학습할 수 있는 것.
- 분산 ↓ : 모든 학습 집합에서 좋은 분류기를 안정적으로 생성하는 것.

- 결국, 편향과 분산에 적절한 가중치를 주어 선택할 수 있도록 해야 한다.

- kNN 분산 ↑ 편향 ↓
- Naïve Bayes 분산 ↓ 편향↑