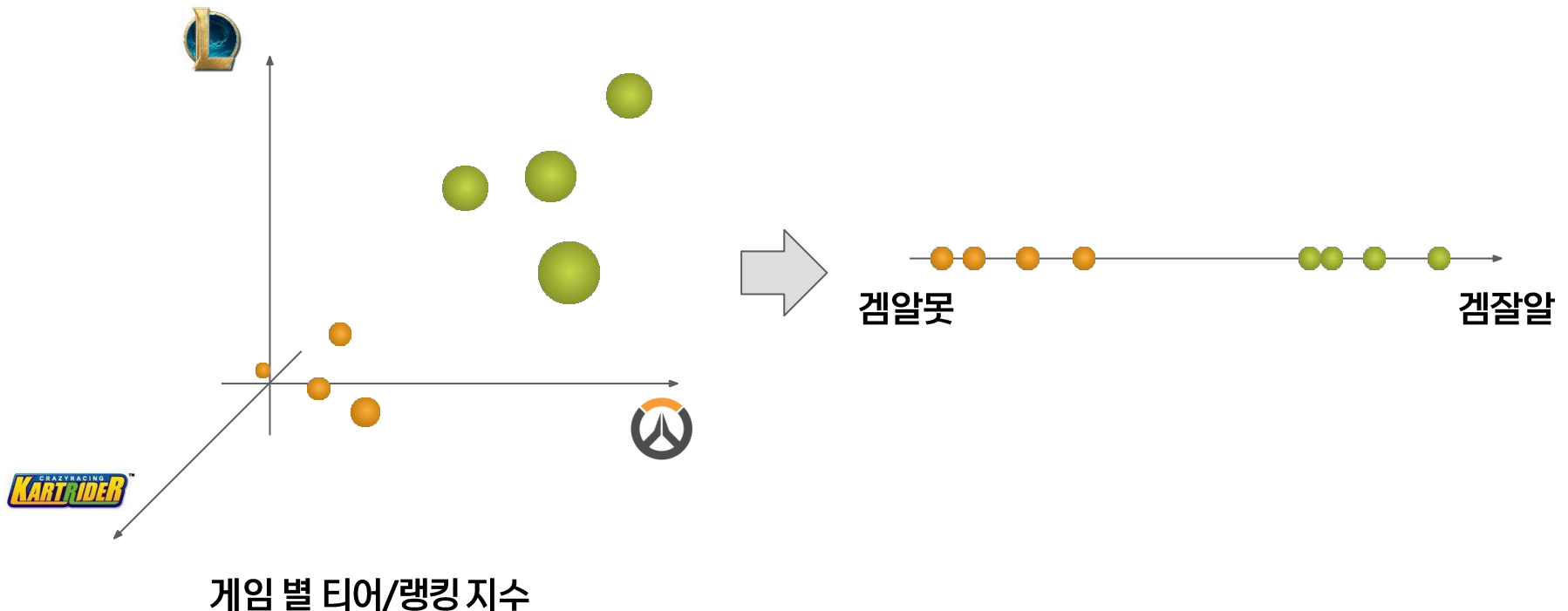


Principal Component Analysis

- 주 성분 분석
 - 데이터의 분포를 결정하는 핵심 성분 찾기
 - 예) 원래 데이터: 게임별 티어 → 주 성분: 게임DNA

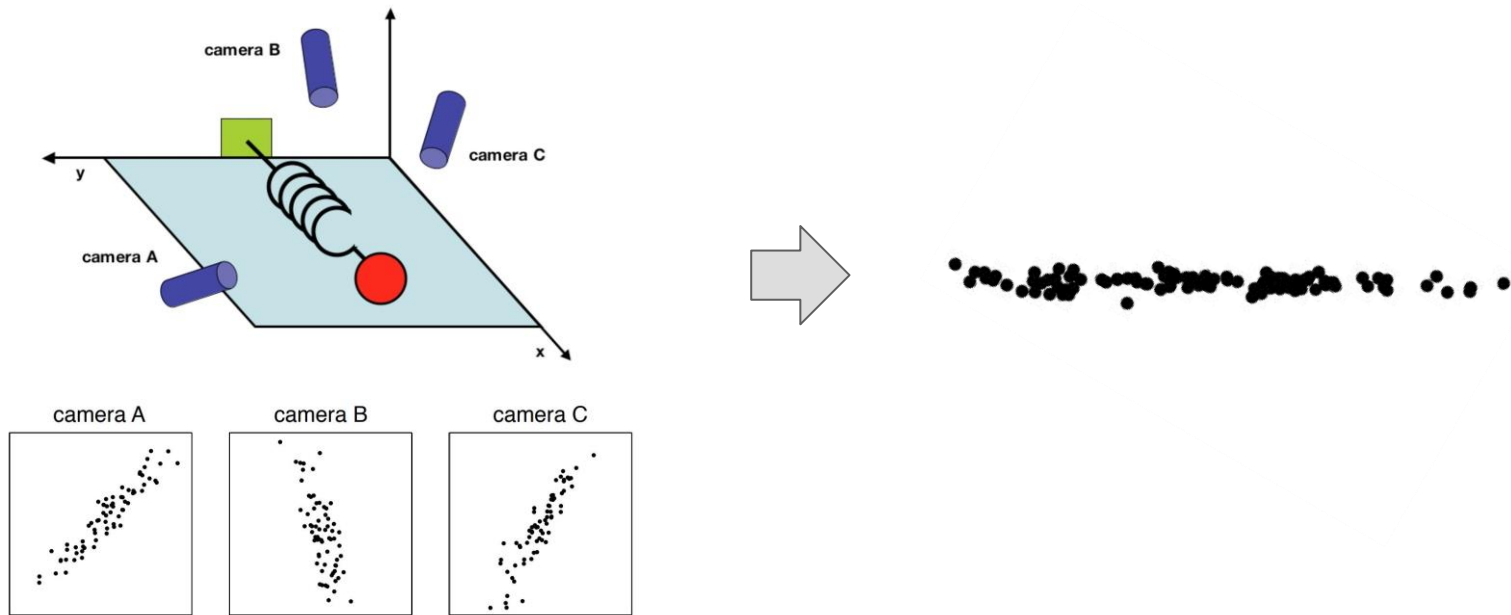


Principal Component Analysis

- 주 성분 분석

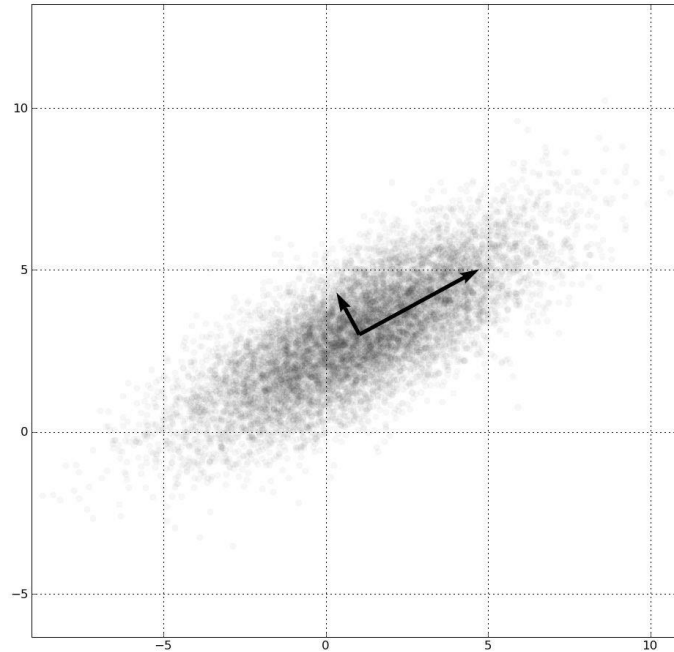
- 데이터의 분포를 결정하는 핵심 성분 찾기

- 예) 원래 데이터: 게임별 티어 → 주 성분: 게임DNA
 - 예) 원래 데이터: 카메라별 공의 위치 → 주 성분: 스프링의 힘



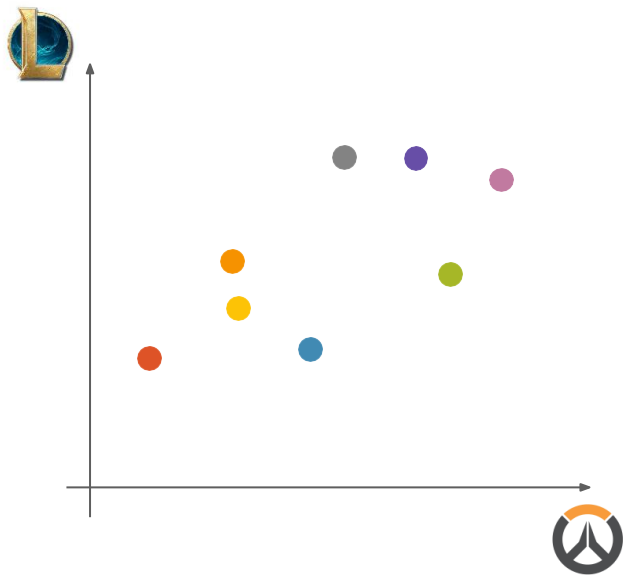
Principal Component Analysis











- 주 성분 분석
 - 분산을 최대화 하면서 서로 직교하는 새로운 축을 찾음



차원 축소

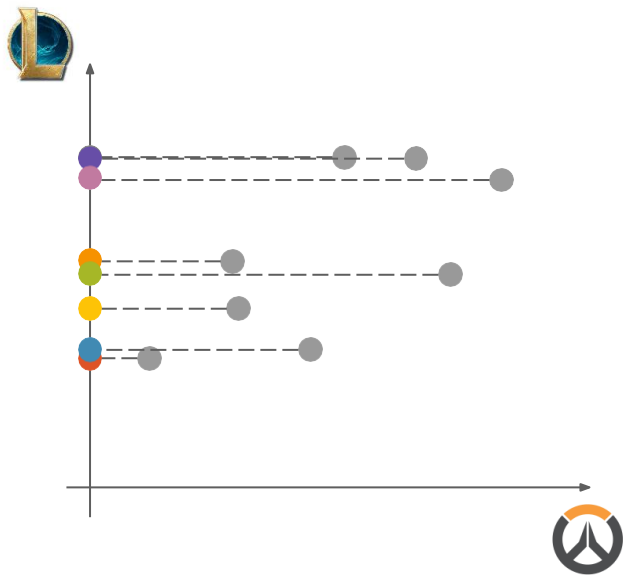
- 차원 축소 방법



								
	A	B	C	D	E	F	G	H
	3.1	3.4	4.6	3.2	7.9	7.8	4.4	7.5
	1.0	4.2	4.0	5.7	6.2	8.1	9.3	9.9

차원 축소

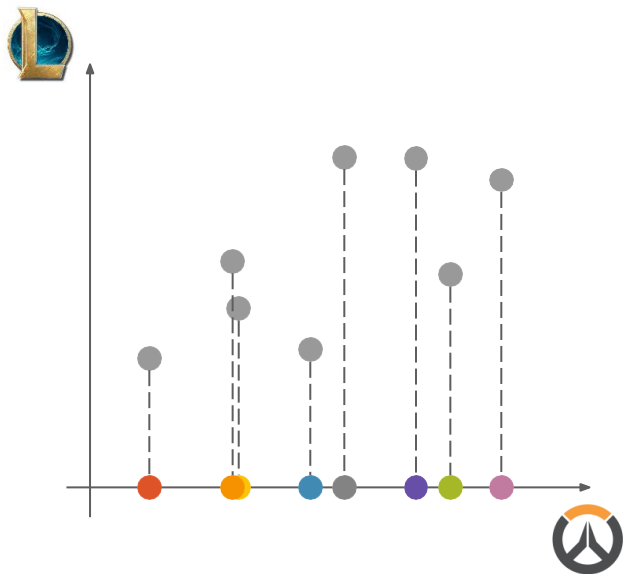
- 차원 축소 방법
 - 방법1. 아무 차원이거나 지운다.





	A	B	C	D	E	F	G	H
League of Legends	3.1	3.4	4.6	3.2	7.9	7.8	4.4	7.5
Overwatch	1.0	4.2	4.0	5.7	6.2	8.1	9.3	9.9

차원 축소

- 차원 축소 방법
 - 방법1. 아무 차원이거나 지운다.
 - 어떤 차원을 지우는 것이 더 좋은가?



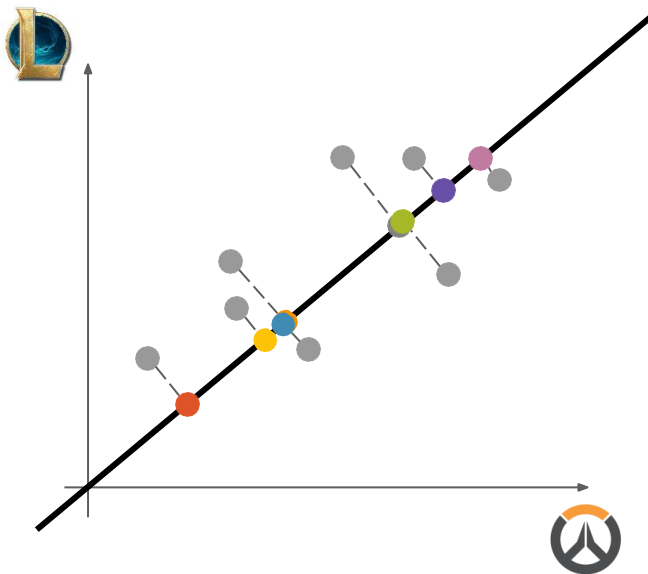
	A	B	C	D	E	F	G	H
	3.1	3.4	4.6	3.2	7.9	7.8	4.4	7.5
	1.0	4.2	4.0	5.7	6.2	8.1	9.3	9.9



차원 축소

- 차원 축소 방법

- 방법2. 새로운 축(선분)을 찾는다. = 주 성분 찾기

- 분산을 최대로...!
 - 어떻게 찾지...?

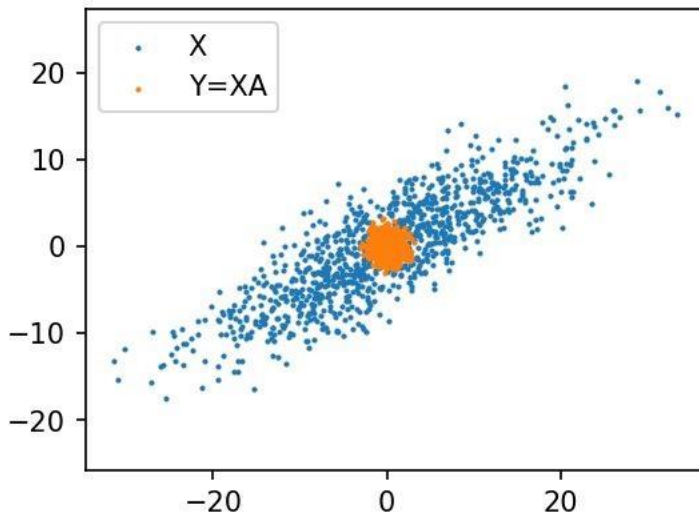


	A	B	C	D	E	F	G	H
	3.1	3.4	4.6	3.2	7.9	7.8	4.4	7.5
	1.0	4.2	4.0	5.7	6.2	8.1	9.3	9.9
?	3.3	5.4	6.1	6.5	10.0	11.2	10.3	12.4

<http://i.imgur.com/Uv2dlsH.gif>

주성분 찾기

- 표준 데이터 X
 - 각 차원의 평균 = 0, 분산 = 1, 차원간 공분산 = 0
 - $d = X$ 의 차원
- $A = d \times d$ 대칭 행렬
- $Y = XA$



$Y =$

	
3.1	1.0
3.4	4.2
4.6	4.0
3.2	5.7
7.9	6.2
7.8	8.1
4.4	9.3
7.5	9.9

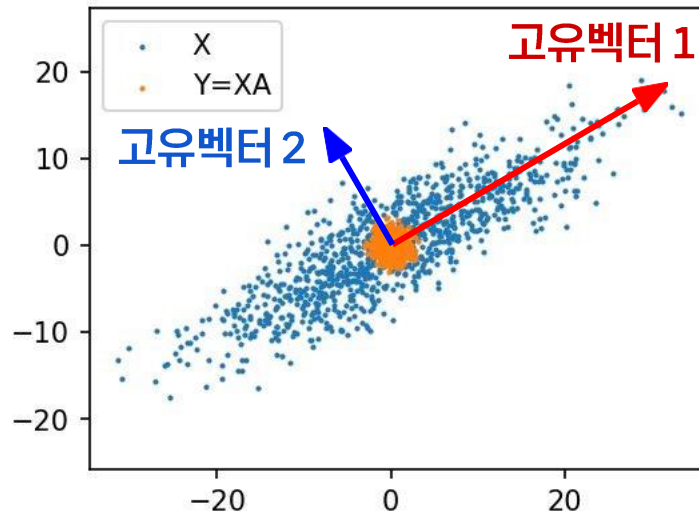
$$A = \begin{bmatrix} 10 & 4 \\ 4 & 5 \end{bmatrix}$$

주성분 찾기

- $(Y = XA)$ 의 주성분은 A 의 고유벡터이다!
 - 행렬 A 에 대해 다음 수식을 만족하는 벡터 v 를 고유벡터라 함: $Av = \lambda v$ (단, λ 는 임의의 상수=고윳값)

질문 1. Y 에 대한 행렬 A 를 어떻게 구하지?

질문 2. 행렬 A 의 고유벡터를 어떻게 구하지?



주성분 찾기

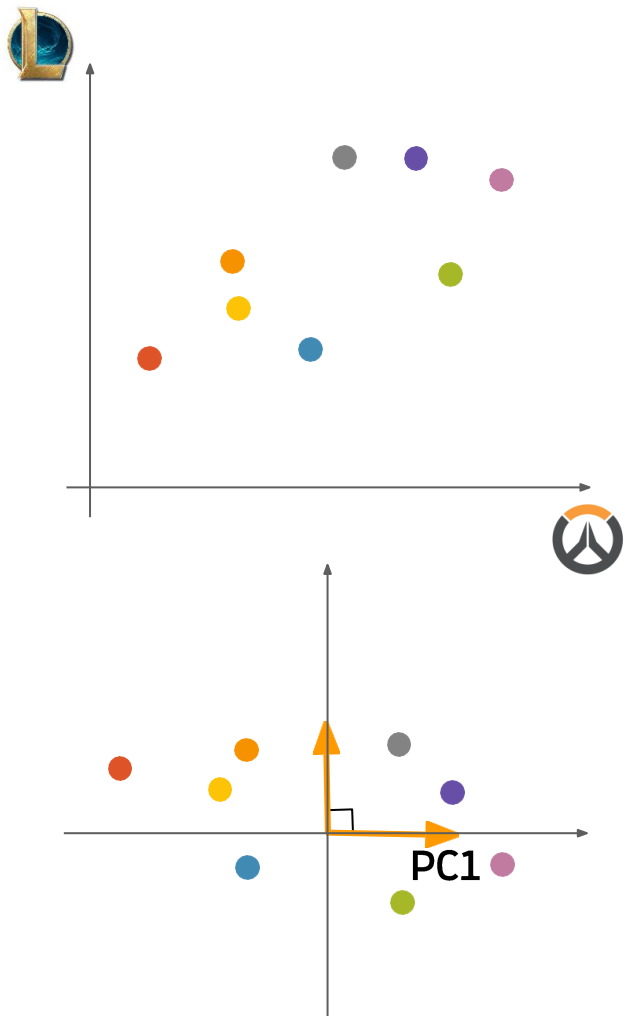
- 질문 1. Y 에 대한 행렬 A 를 어떻게 구하지?











- A 를 구하기 어렵기 때문에, Y 의 공분산 행렬(covariance matrix) Σ 를 활용! $\rightarrow \Sigma = A^T A$ (증명?)
 - Σ_{ij} = Y 의 i 번째 차원과 j 번째 차원의 공분산
 - $\Sigma = (Y^T Y) / n$ (단, $Y^T = Y$ 의 전치행렬, $n = Y$ 의 행 수)









- 질문 2. 행렬 A 의 고유벡터를 어떻게 구하지?

- A 의 고유벡터 = Σ 의 고유벡터 (증명?)
- Σ 를 고윳값 분해 (eigen decomposition)
 - Power method를 반복
 - 기타 다양한 고윳값 분해 Solver 활용

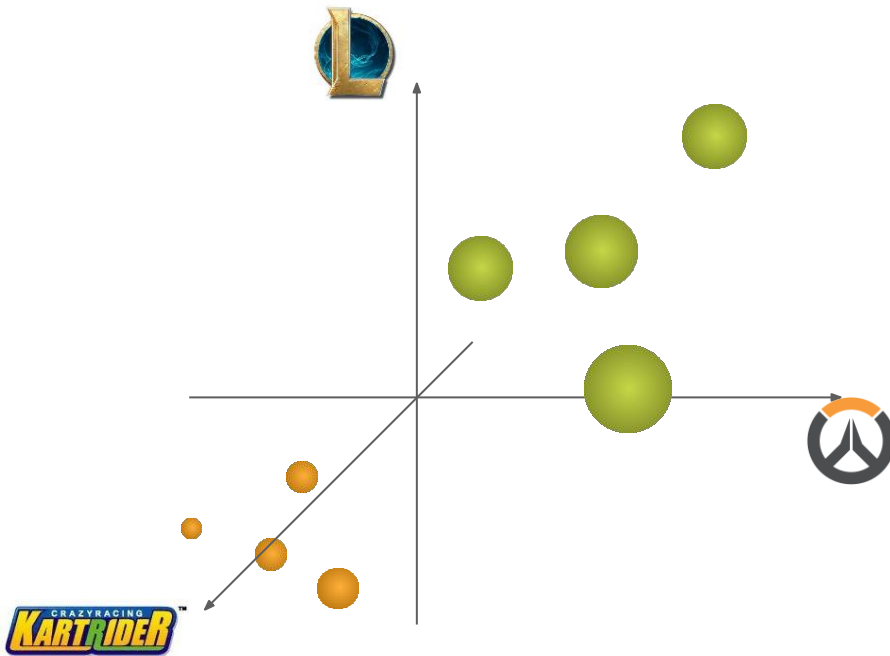
주성분으로 데이터 표현



								
	A	B	C	D	E	F	G	H
	3.1	3.4	4.6	3.2	7.9	7.8	4.4	7.5
	1.0	4.2	4.0	5.7	6.2	8.1	9.3	9.9

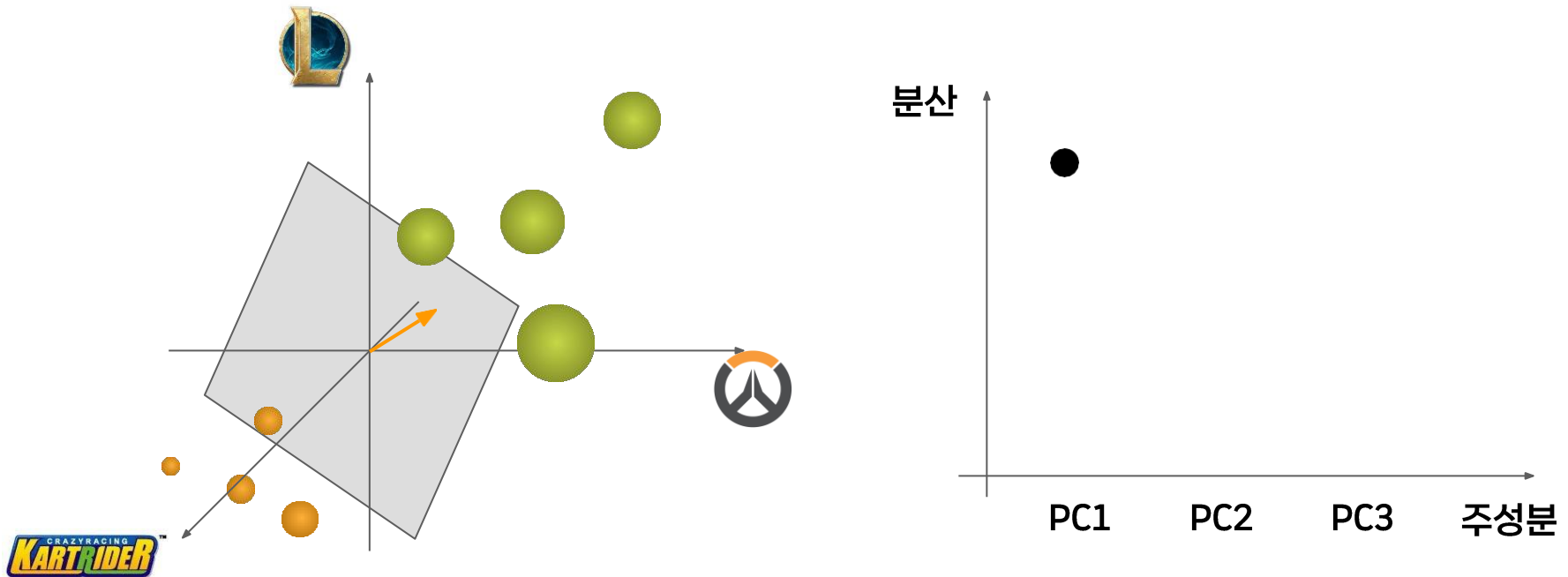
								
	A	B	C	D	E	F	G	H
PC1	-4.9	-2.8	-2.1	-1.6	1.9	3.1	2.1	4.3
PC2	1.1	0.7	1.5	-0.3	1.7	0.3	-0.9	-0.3

3차원 예시



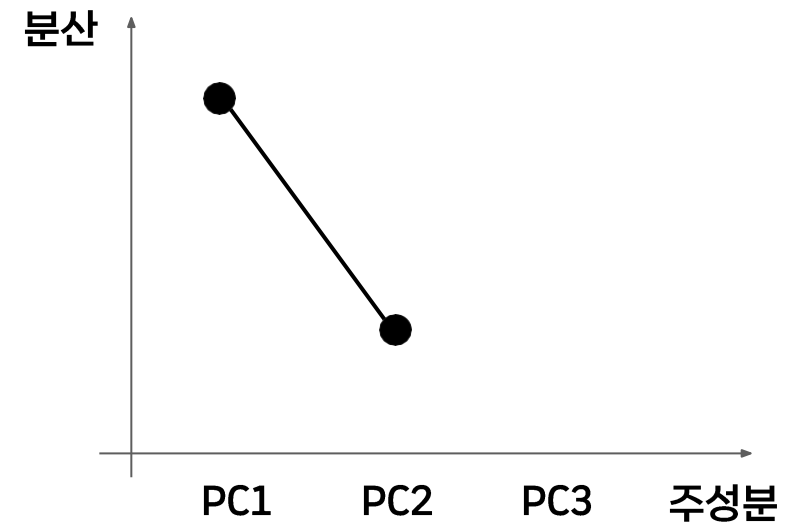
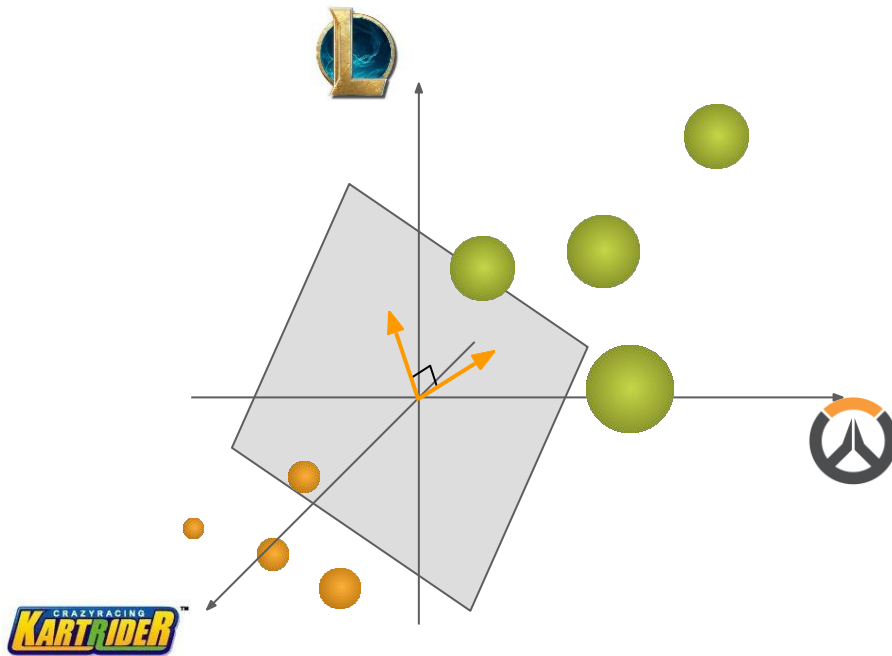
3차원 예시

- PC1 찾기: 사영했을 때 분산이 가장 커지는 벡터



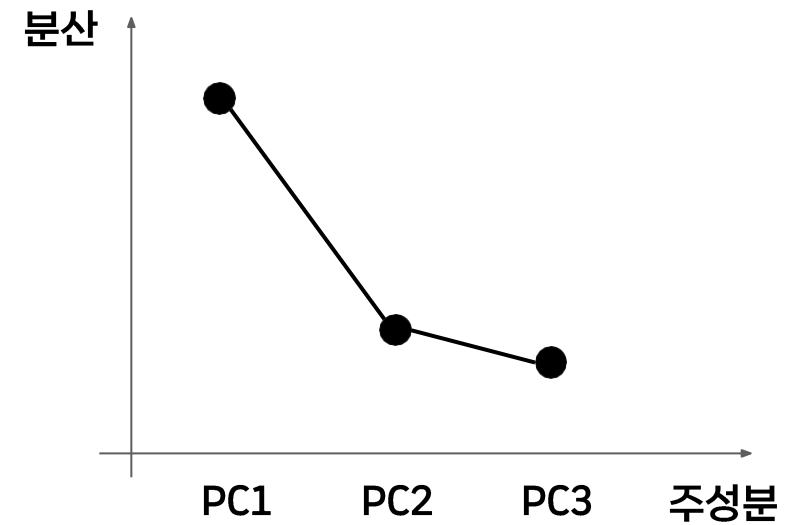
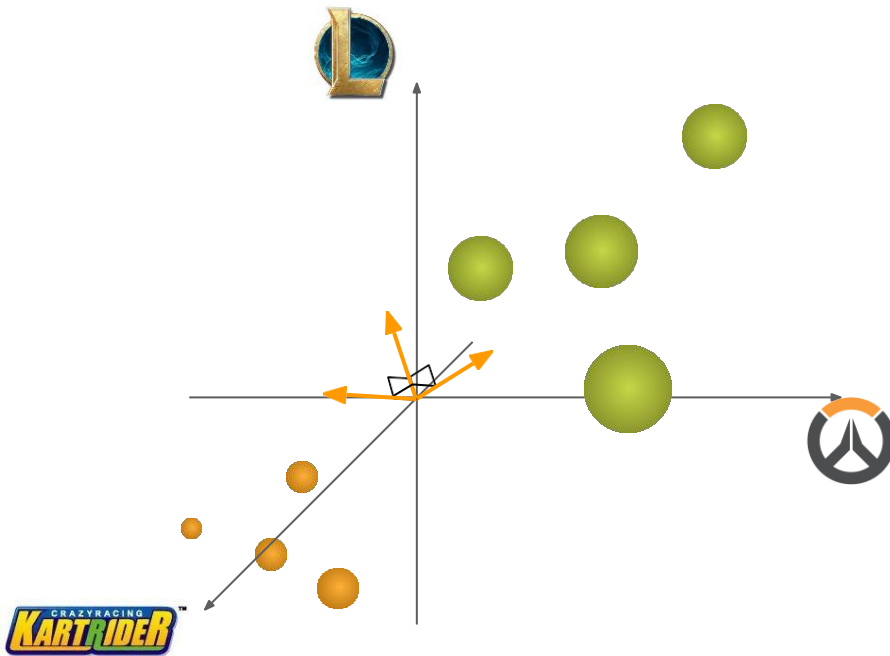
3차원 예시

- PC1의 직교평면에서 PC2 찾기



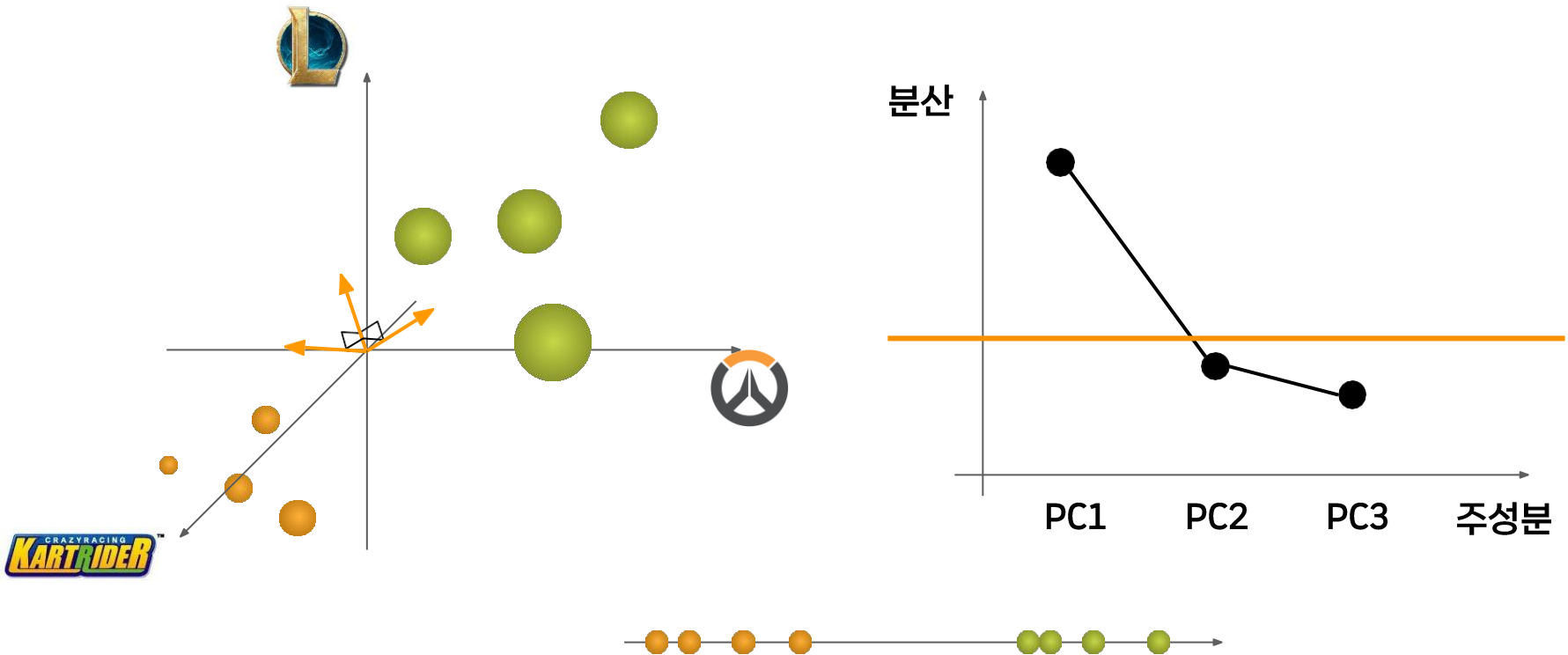
3차원 예시

- PC1과 PC2에 모두 직교하는 벡터 = PC3



3차원 예시

- PC1과 PC2에 모두 직교하는 벡터 = PC3



Questions?

지도학습 Supervised Learning

훈련 데이터(Training Data)로부터 하나의 함수를 유추해내기 위한 기계 학습(Machine Learning)의 한 방법

Training Data

[1.2,	3.8,	-1.4,	...,	4.1]	→	1.1
[3.2,	-1.2,	-0.2,	...,	2.1]	→	2.7
[2.8,	-1.4,	-0.3,	...,	2.3]	→	2.8
[1.2,	3.4,	-1.5,	...,	4.2]	→	0.9
[4.2,	2.1,	2.8,	...,	-0.5]	→	-0.1
...						
[3.2,	2.2,	2.2,	...,	-0.4]	→	-0.2

Test

[1.3,	3.2,	-1.5,	...,	4.1]	→	?
-------	------	-------	------	------	---	---

비지도학습 Unsupervised Learning

“데이터 패턴 학습”

기계 학습의 일종으로, 데이터가 어떻게 구성되었는지를 알아내는 문제의 범주에 속함

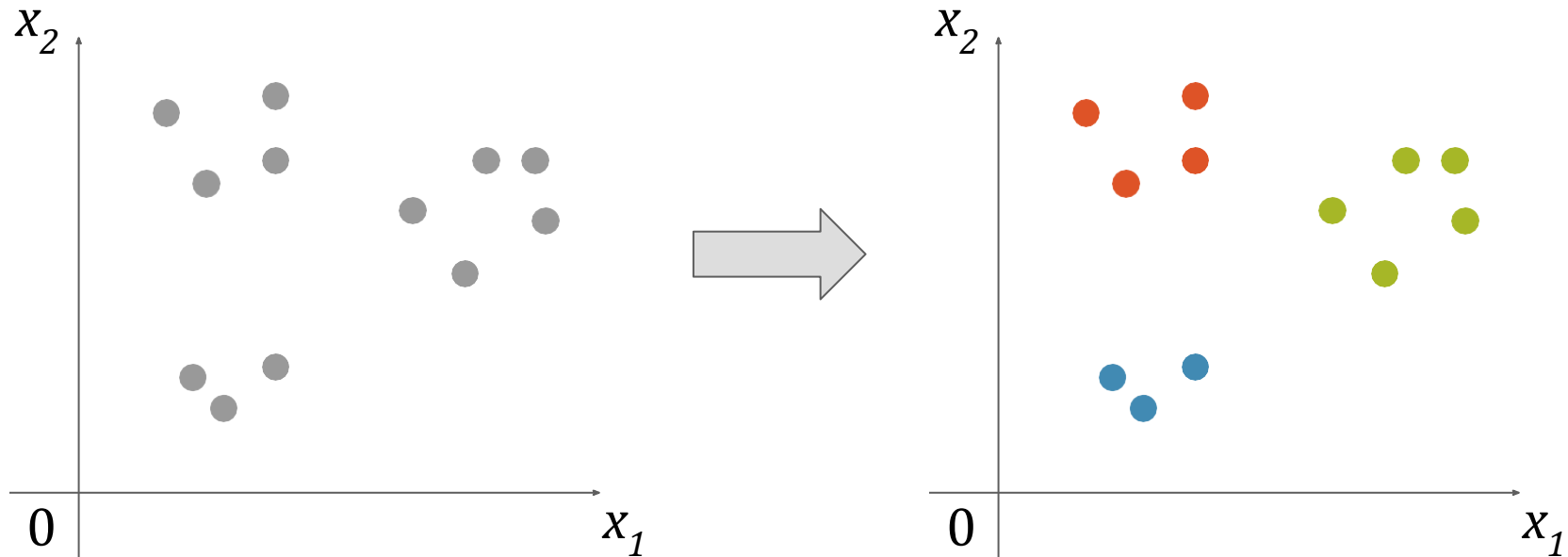
- Clustering
- Dimensionality Reduction
- Association Analysis

Data

[1.2, 3.8, -1.4, ..., 4.1]	→	?
[3.2, -1.2, -0.2, ..., 2.1]	→	?
[2.8, -1.4, -0.3, ..., 2.3]	→	?
[1.2, 3.4, -1.5, ..., 4.2]	→	?
[4.2, 2.1, 2.8, ..., -0.5]	→	?
...		
[3.2, 2.2, 2.2, ..., -0.4]	→	?

클러스터 분석 Clustering

다차원 공간에서 여러개의 점들이 존재할 때,
서로 가까이 있는 점들을 서로 연관시키는 문제



클러스터 분석 활용 예

- 인물 사진 분류
 - 인물사진들 중에 닮은 사진 모으기



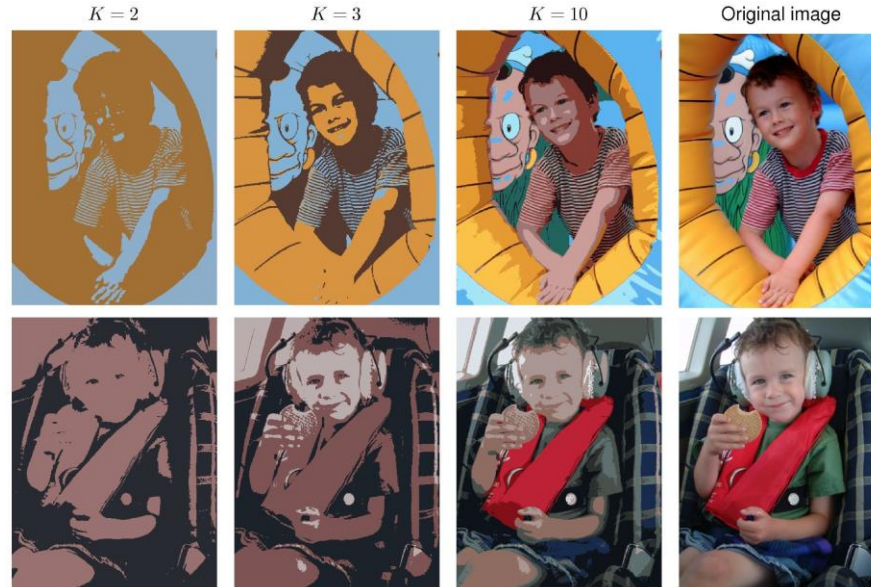
같은 사람인가요, 다른 사람인가요?



출처: <http://www.gearbax.com/19361>

클러스터 분석 활용 예

- 비슷한 뉴스 모으기
- 스팸메일 분류
- 비슷한 성향의 사용자/영화 모으기
- 사진 압축



출처: <http://norman3.github.io/prml/docs/chapter09/1.html>

K-Means Clustering

**반복적인 연산을 통해 데이터를 k 개의 클러스터로
분할하는 알고리즘**

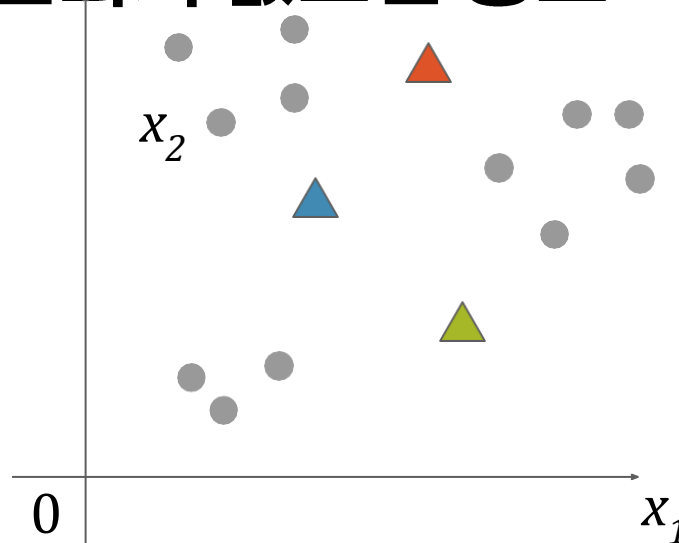
- 클러스터 분석 알고리즘
- 분할법 (partitioning)
- 클러스터 개수 (k) 지정 필요
- 반복연산 (iterative process)

K-Means Clustering

1. 임의로 k 개의 중심점(centroid)을 생성
2. 각각의 점을 가장 가까운 중심점의 클러스터에 포함시킴
3. 각 클러스터에 포함된 점들을 평균내어 새로운 중심점을 계산
4. 2-3 을 반복하다가 클러스터에 변화가 없으면 종료

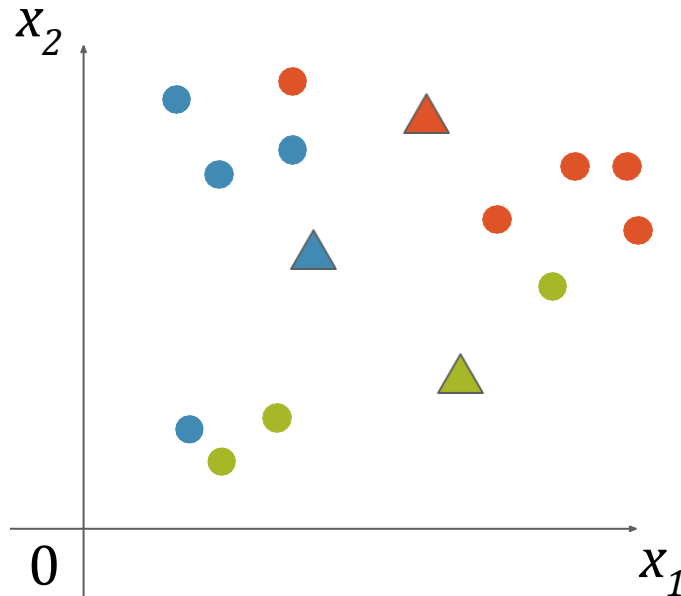
K-Means Clustering

1. 임의로 k 개의 중심점(centroid)을 생성
2. 각각의 점을 가장 가까운 중심점의 클러스터에 포함시킴
3. 각 클러스터에 포함된 점들을 평균내어 새로운 중심점을 계산
4. 클러스터에 변화가 없으면 종료



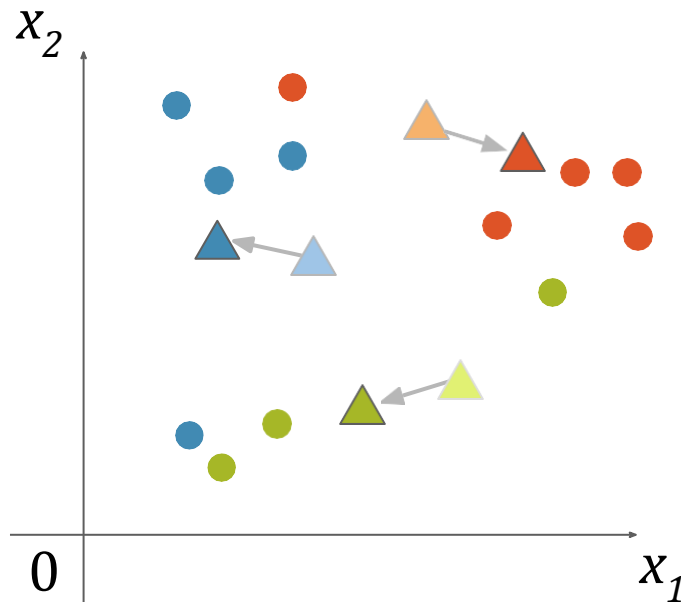
K-Means Clustering

1. 임의로 k 개의 중심점(centroid)을 생성
2. **각각의 점을 가장 가까운 중심점의 클러스터에 포함시킴**
3. 각 클러스터에 포함된 점들을 평균내어 새로운 중심점을 계산
4. 클러스터에 변화가 없으면 종료



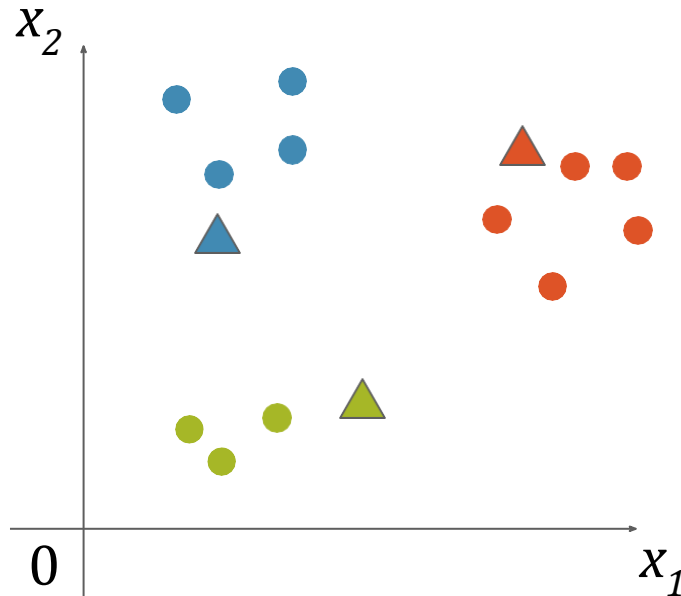
K-Means Clustering

1. 임의로 k 개의 중심점(centroid)을 생성
2. 각각의 점을 가장 가까운 중심점의 클러스터에 포함시킴
3. **각 클러스터에 포함된 점들을 평균내어 새로운 중심점을 계산**
4. 클러스터에 변화가 없으면 종료



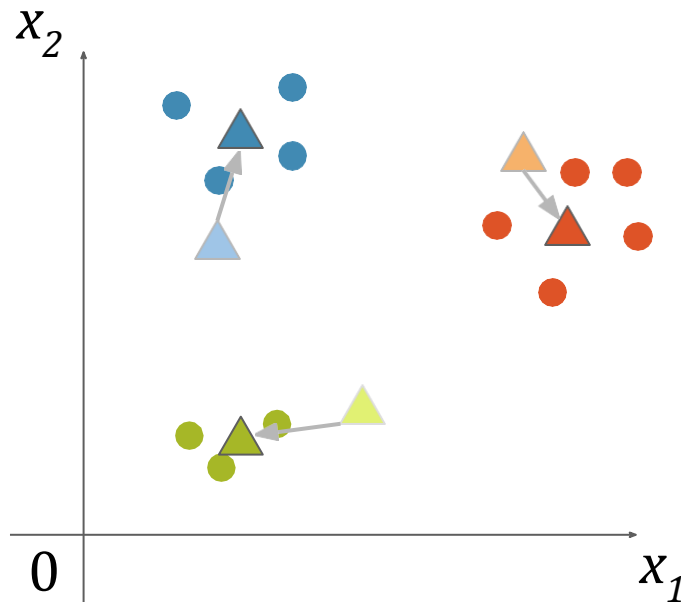
K-Means Clustering

1. 임의로 k 개의 중심점(centroid)을 생성
2. **각각의 점을 가장 가까운 중심점의 클러스터에 포함시킴**
3. 각 클러스터에 포함된 점들을 평균내어 새로운 중심점을 계산
4. 클러스터에 변화가 없으면 종료



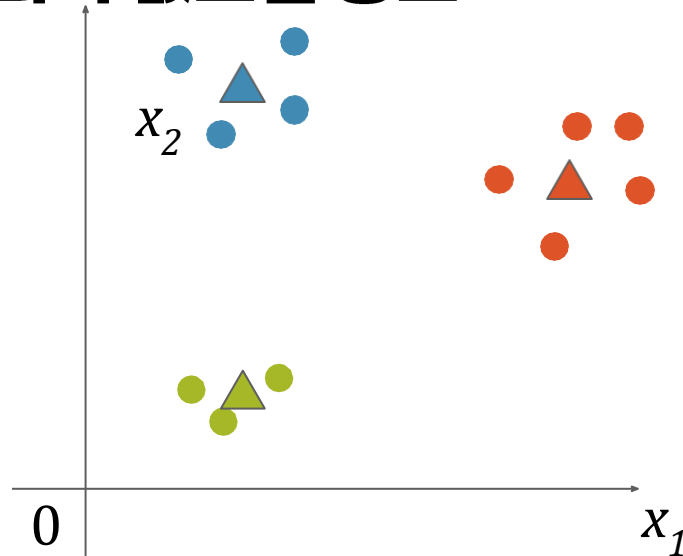
K-Means Clustering

1. 임의로 k 개의 중심점(centroid)을 생성
2. 각각의 점을 가장 가까운 중심점의 클러스터에 포함시킴
3. **각 클러스터에 포함된 점들을 평균내어 새로운 중심점을 계산**
4. 클러스터에 변화가 없으면 종료



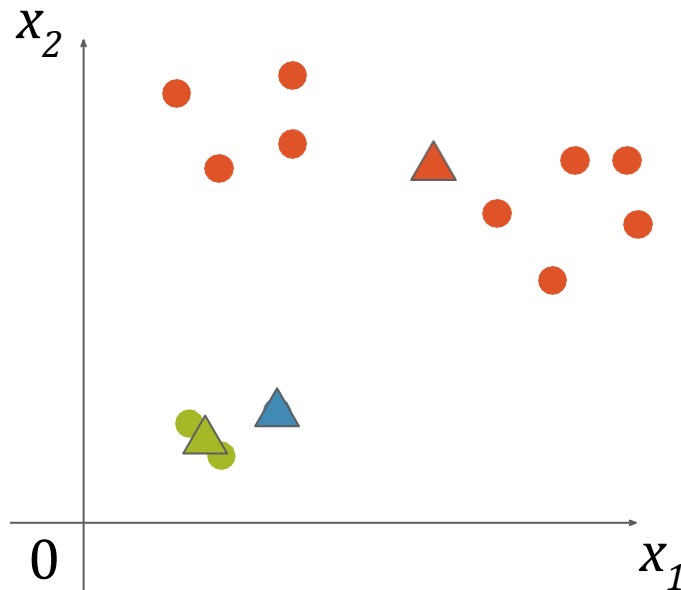
K-Means Clustering

1. 임의로 k 개의 중심점 (centroid) 을 생성
2. 각 데이터 포인트를 가장 가까운 중심점의 클러스터에 할당
3. 각 클러스터에 포함된 점들을 평균내어 새로운 중심점을 계산
4. 클러스터에 변화가 없으면 종료



이상한 경우...

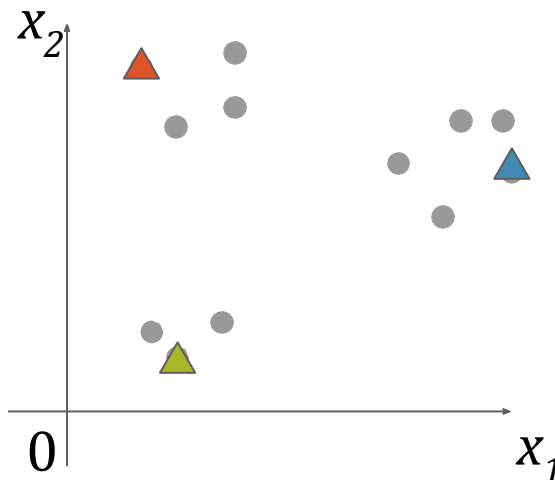
K-Means 알고리즘은 Local Optimum에 빠질 수 있다



중심점 초기화

중심점 초기화 방법에 따라 결과가 달라질 수 있다

- 임의의 벡터로 중심점 초기화
 - 여러번 반복하여 가장 좋은(?) 결과 선택
- Forgy: 데이터 점 들 중 임의로 선택
- 직접 중심점 지정하기 → 데이터를 얼추 알고 있을 때
- K-Means++: 멀리 떨어진 점들을 초기 중심점으로 사용



k 값 선택하기

- 좋은 클러스터? → 분산이 낮다 → 비용이 낮다
- 목표함수, 비용:

$$\arg \min_S \sum_{i=1}^k \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2$$

S : 데이터 집합

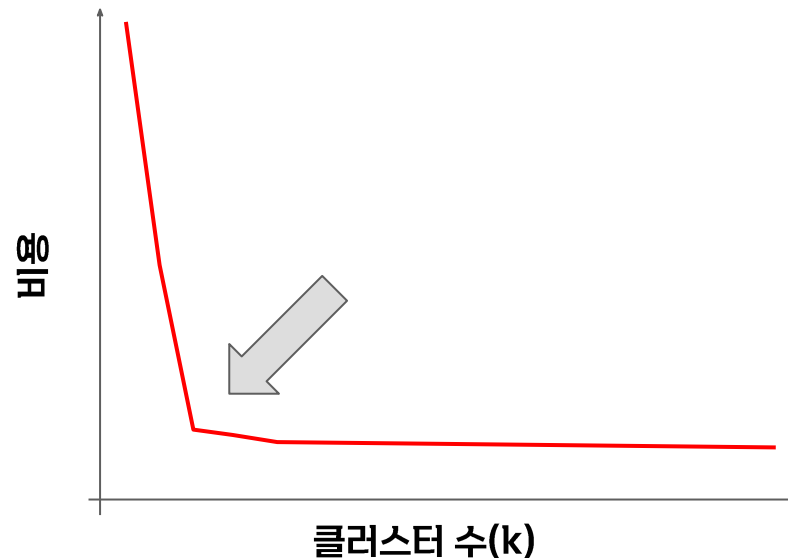
k : 클러스터 수

S_i : i 번째 클러스터에 속한 데이터 집합

μ_i : i 번째 클러스터의 centroid (데이터 평균)

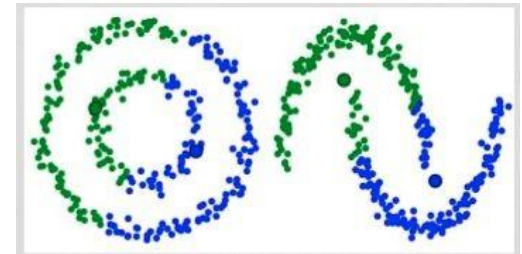
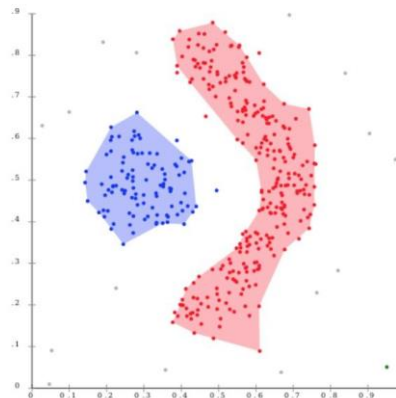
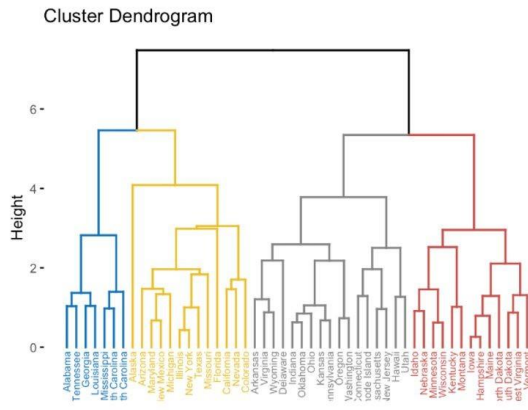
k 값 선택하기

- k값을 1부터 증가시켜가며 비용을 분석
- 비용의 감소가 급격히 줄어드는 지점 선택



다른 클러스터링 방법

- k-medoids: k-means 알고리즘이 이상치에 민감한 문제를 보완
- 계층적 클러스터링
- DBSCAN: 밀도기반 클러스터링
- 등등...



Questions?