unsupervised learning은 data에 숨겨진 pattern이나 의미를 찾는 행위.
⇒ 무슨 task를 할지도 직접 정해야 한다.

(1) Clustering ⇒ 살짝 주관이 들어갈수도 (기준이 무엇인지에 따라서)

동일 cluster의 data들이 공통점↑ & 다른 cluster data들 공통점↓
high intra similarity   low inter similarity

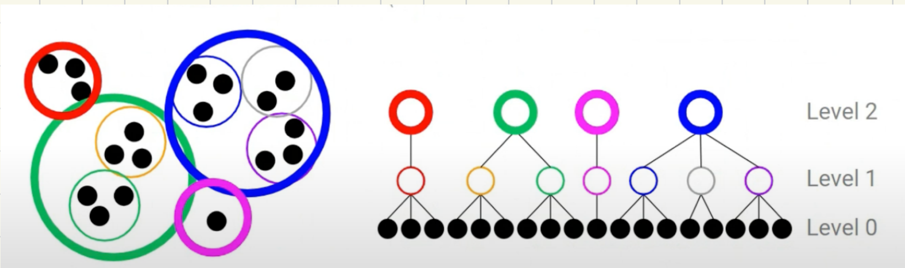
① data들 사이의 유사성/거리를 나타낼수 있는 metric을 정의해야한다.

ex) $L_n(A,B) = \sqrt[n]{A^n + B^n}$

<1 - Bottom-up 방식>
data 각각이 cluster라고 생각하는 관점. n개의 data = n개의 group
<2 - partional>
모든 data를 1개로 두고 split하는 관점.

Bottom up (Hierarchical clustering)

1) 모든 data가 각 group
2) 초기에는 threshold 엄격 ⇒ 친한애 group을 merge
3) threshold를 낮추면서 group merging 진행.



cluster들 사이의 similarity는? ⇒ 다양한 방법들 ⇒ 주로 average 사용.

KMeans

① K개 random point
② data들을 가장 가까운 K개 group화
③ group의 average를 다시 구하고 다시 가까운 data들끼리 group화.) repeat.

→ 더이상 group의 average가 변하지 않으면 terminate.

이를 수식화 해보면 아래와 같이 표현이 가능합니다.

$$J = \sum_{i=1}^{N}\sum_{j=1}^{k} \underbrace{r_{i,j}}_{\text{소속여부}} \| x_i - \mu_j \|^2$$

↳ group의 평균, "centroid"

결국 kmeans = minimize J 이다.

수도 Code

step (1) $O(n \times k)$ $c^{(i)} = \arg\min \| x^{(i)} - \mu_k \|^2$ // 가장 가까운 group mapping

step (2) $O(n)$ $\mu_k = \dfrac{\sum_{i=1}^{n} 1(c^{(i)}=j)\, x^{(i)}}{\sum_{i=1}^{m} 1(c^{(i)}=j)}$ // centroid update.

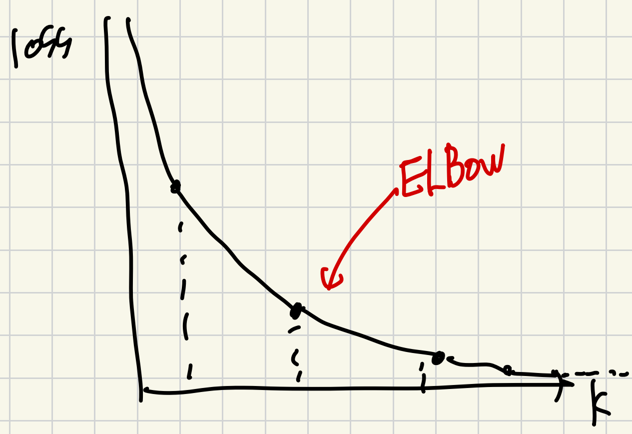
worst : $O(k^n)$ np hard problem 이지만, 거의 발생 X.

* Cost가 감소하는 경우에만 group을 change ⇒ 그래서 언제나 끝난다.

Because 모든 경우를 다 봐도 언젠간 terminate.

* 초기 k가 달라지면 결과가 달라지기 쉽다.

* k 값을 키우면 키울수록 Loss는 줄어든다. 그러면 k를 몇으로 설정해야할까?



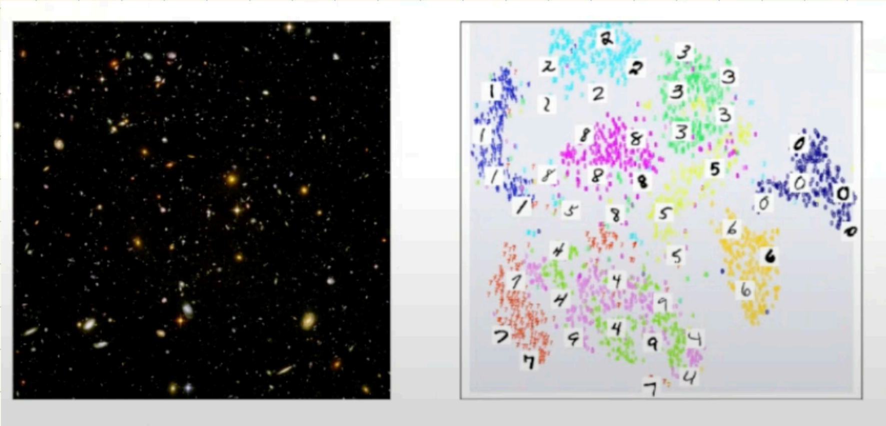
k↑ 일수록 Loss는 줄지만 점점 얻는 이득이 줄어든다.

이때 적절한 ELBOW를 찾아서 사용.
(visually 직감적으로 고른다)

Dimension Reduction : 차원을 최대한 줄이면서 정보는 잃지 않기

data manifold ≒ Data는 특정 영역에 몰려있다.

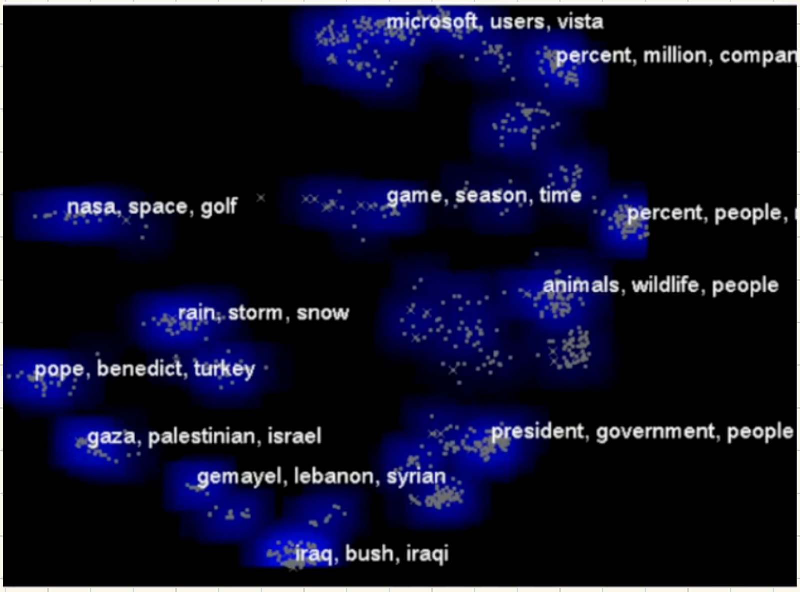
고차원 space에 특정 영역이 몰려있어 ⇒ detail만 희생하면 훨씬 낮은 차원으로 data들을 분류가능.



general한 형식으로 manifold Learning (Multidimensional Scaling : MDS) 표현

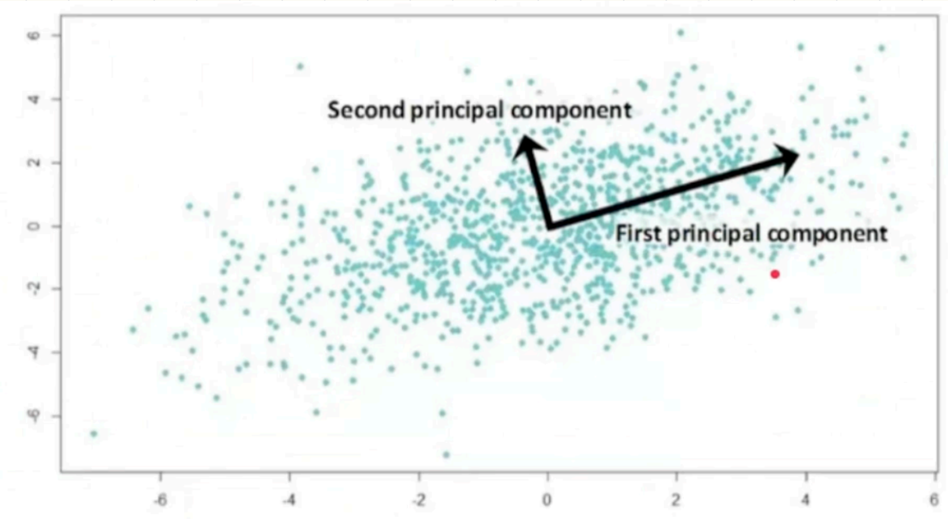
$$\sum_{i>j} R\left(\|f(x^{(i)}) - f(x^{(j)})\|, \rho(x^{(i)}, x^{(j)})\right)$$

차원축소 이후에도 원래 ρ와 비슷하게



단어들도 유사한 의미를 갖는 단어들이 비슷한 위치에 오게 된다.

PCA : MDS의 Special Case



Data를 나누는 가장 중요한 "축"을 찾는것이다.

우선 data의 Scale을 맞추기 위해 Zero-Centered & normalized 실행.

data의 error를 minimize 하니깐 covariance의 diagonal element가
eigen value가 된다 → 즉, eigen value가 높은 vector가 중요 factor가
나온다.

PCA의 dimension 또한
kmeans처럼 Elbow로 결정가능.

Regression Vs PCA