Principal Comp Analysis

-- inputs

-- output

-- objective

1) Examples: Collection of N Datapoints$\mathbb{R}^d$

2) X input $\mathbb{R}^d$ => f(x) = out (y = Gx)

x = D x 1 matrix

G = K x D matrix

3) What does PCA actually optimize for? What PCA find out? **Explained variance**, each one of the columns of G are orthogonal and maximize variation.

2d Example:

Consequence of maximizing variation

X -> N datapoints in $\mathbb{R}^d$

- Draw two datapoints at random xi,xj with replacement.
- $E[||x_i - x_j||_2^2] = E[(x_i - x_j)^T(x_i - x_j)] = E(x_i^T x_i - 2x_j^T x_i + x_j^T x_j) = 2E(x_i^T x_i) - 2E(x_i)^T E(x_i) = 2Var(x)$
- Hence, maximizing variation <=> maximizing distance
- x_i, x_j and y_i, y_j
  want to know the distortion:
  $||y_i - y_j||_2^2 < ||x_i - x_j||_2^2 * \epsilon$

  $\epsilon = \frac{\sum_i^k \sigma_i^2}{\sum_i^D \sigma_i^2}$(top-K single value / all single value)
- This is not a property of just certain data. Almost all data sets have low dimensional structure.

Hashing v.s. Dim Reduction

$S_1, S_2, ..., S_N$ N strings. Space of all strings is really big. But we work with a finite set of strings.

Hash them: $Number_1, Number_2, ..., Number_N$

N is too big, relative to K in PCA.

The intuition behind the fact that almost every high dimensional dataset can be compressed.

$X \in \R^{N*d} \ \ \ \ \ O(r^d)$ is the dimensions in which your data could come from, however, you actually only have n data points.

1）Finite Metric Space: A set of N distinct data points in $\mathbb{R}^d$, with pairwise Euclidean distance. $X_{i=1....N}$

Another big result from 1980s!!

Johnson-Lindenstross lemma

if I have a Euclidean FMS. There exists a mapping such that $$(1 - \epsilon) || x_i - x_j||_2^2 \leq ||f(x_i) - f(x_j)||_2^2 \leq (1 + \epsilon) || x_i - x_j||_2^2$$

f: -> linear compression schema

- Actually "random" linear compression achieve the above result

There exists a mapping such that you can bound this distortion and you can control this distortion

---

Example

1) $x_i \in \mathbb{R}^d$ and vector $a \sim N(0, I_d)$

a) $E(a * x_i) = 0$

b) $E[(a * x_i)^2] = ||x||_2^2$

2) x_i, x_j

$||x_i - x_j||_2^2 = ||U||_2^2$

$a * u = a * x_i - a * x_j$

if we take all data points and normal random dot product. The pairwise distances are preserved in expectation.

y = G*x we can see G is a collection of a

$G = [a_1, a_2, ..., a_k]$ they all preserve pairwise distances.

Back to J-L lemma.

- How big does K have to be? $k = O(\frac{logn}{\epsilon^2})$ no d, the original complexity of space doesn't as much matter as how many data points you have.

  Very similar to the results in birthday paradox. no collisions $O(\sqrt{n}/a)$

- Distance measure of d(x1,x2) can't always project this into a lower dimensional space.