Lecture 9: Dimensionality Reduction

[We live in an era of "Big Data".

Not unusual for an "object" (image, person, video, ...) to have millions, billions of numerical "features", and/or to have millions, billions of items. Doing one thing a billion times is maybe okay, algorithmically. Thousands of such things? Not so. Billion × billion? Don't make me laugh.]

Say you have n data items, $x^{(1)}, \dots, x^{(n)}$ [$n = 2^{20 \dots 30}$ lots!]

Each has D numerical features. [pixel colors, song preferences...]

↳ an item $x_i \approx$ vector/point in $\mathbb{R}^D$. [Idea: nD is very large; too large.]

"Big Data" / "High-Dim. Data" ["Curse of dimensionality"]

[Any number of "geometric" algs. you might want to run on data...]

- clustering
- nearest neighbor queries
- PCA [next time]
- ....

Idea: (try to) map data to low dimension,

$$x^{(i)} \mapsto y^{(i)} \in \mathbb{R}^K, \quad K \ll D$$

so that ... relationships... preserved

↳
- distances
- lengths
- angles
- dot products

[these are all kinda the same...]

Recall inner/dot product of $u, v \in \mathbb{R}^m$: $\langle u, v \rangle = u \cdot v = \sum_{i=1}^{m} u_i v_i$

$u \cdot u = \sum_i u_i^2 = \|u\|^2$. Fact ("cosine law"): $u \cdot v = \|u\| \cdot \|v\| \cdot \cos\theta$

θ ↳ angle between u, v.

$$\text{dist}(u,v)^2 = \|u - v\|^2 = (u-v) \cdot (u-v) = u \cdot u - 2 u \cdot v + v \cdot v$$

[So: all interrelated.]

[Potential objection: seems impossible to preserve all this in lower dimensions.]

Worry: Say $\|x^{(i)}\|^2 = 1 \quad \forall i$ [all data vectors are "unit vectors"]

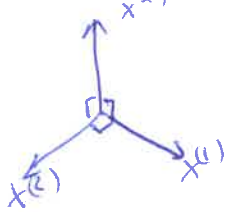
And $\|x^{(i)} - x^{(j)}\| = \sqrt{2} \quad \forall i \neq j$

$$\Rightarrow \|x^{(i)} - x^{(j)}\|^2 = 2 \Rightarrow \underbrace{x^{(i)} \cdot x^{(i)}}_{1} - 2x^{(i)} \cdot x^{(j)} + \underbrace{x^{(j)} \cdot x^{(j)}}_{1} = 2$$

$\Rightarrow x^{(i)} \cdot x^{(j)} = 0$

$\Rightarrow$ perpendicular (angle $= \frac{\pi}{2} = 90°$)

$\therefore$ [high-dim. analogue of]



Basic linear algebra $\Rightarrow$ n perpendicular vecs need to be in n dims.

$\therefore$ cannot achieve $K < n$.

But. Say you don't mind $\approx$ "10% error".

$\rightarrow$ $y^{(i)}$'s with $.9 \leq \|y^{(i)}\|^2 \leq 1.1 \quad \forall i$,

$$.9\sqrt{2} \leq \|y^{(i)} - y^{(j)}\|^2 \leq \sqrt{2} \cdot 1.1 \quad \forall i \neq j$$

$$\Longleftrightarrow -0.1 \leq y^{(i)} \cdot y^{(j)} \leq 0.1$$

Possible to have n such $y^{(i)}$'s in.... $\approx 100 \log n$ dims!

($100 = \frac{1}{(10\%)^2}$)

Exponentially fewer dims!!

[$K \approx 3000$ for $n = 1$ Billion!!]

[Processing one data item is now 300,000x faster!!]

[We'll show it's possible for any $x^{(i)}$'s but let's first see why for these special $x^{(i)}$'s. How to find exponentially many "almost orthonormal" vectors in K dims?]

[When you don't know how to find something, pick it at random!!]

Idea: Let each $y^{(i)} = \left.\begin{bmatrix} +1 \\ -1 \\ -1 \\ +1 \\ \vdots \\ +1 \end{bmatrix}\right\} K$ random ±1 "bits".

Er, then $\|y^{(i)}\|^2 = \sum^K (\pm 1)^2 = K$, supposed to be ≈ 1.

So actually, $y^{(i)} = \frac{1}{\sqrt{K}} \begin{bmatrix} +1 \\ -1 \\ -1 \\ \vdots \end{bmatrix}$, randomly, for $i = 1, 2, 3, \ldots$

Q: What is $y^{(1)} \cdot y^{(2)}$? [for any $y^{(i)} \cdot y^{(j)}$?]

It's like $\frac{+1}{\sqrt{K}} \cdot \frac{-1}{\sqrt{K}} + \frac{-1}{\sqrt{K}} \cdot \frac{+1}{\sqrt{K}} + \frac{-1}{\sqrt{K}} \cdot \frac{-1}{\sqrt{K}} + \cdots$

all these bits random

"distributed as" $\sim \frac{1}{K}\left((-1) + (-1) + (+1) + \cdots\right)$

$=$ avg. of K random ±1 bits

In expectation, 0. $\mathbb{E}[y^{(1)} \cdot y^{(2)}] = 0$ [exactly]

[But we have to worry about fluctuations. Won't be this exactly. Is it, like 80% of the time at most .1, or what?]

Let $B_1, \ldots, B_K$ be random ±1. Let $A = \text{avg}(B_1, \ldots, B_K)$

stddev$[A]$ = ? [key quantity to understand]

$= \sqrt{\text{Var}[A]} = \sqrt{\text{Var}\left[\frac{1}{K}(B_1 + \cdots + B_K)\right]}$.

$= \frac{1}{K^2}\left(\text{Var}[B_1] + \cdots + \text{Var}[B_K]\right)$

fact:
$\text{Var}[cX] = c^2\,\text{Var}[X]$
$\text{Var}[X+Y] = \text{Var}[X] + \text{Var}[Y]$ if independent [as B_i's are]

$\text{Var}[B_i] = \mathbb{E}[B_i^2] - \mathbb{E}[B_i]^2$
$= 1 - 0^2 = 1$.

$\therefore \text{Var}[A] = \frac{K}{K^2} = \frac{1}{K}$. $\therefore$ stddev$[A] = \frac{1}{\sqrt{K}}$.

$\therefore$ intuitively/heuristically we expect A is "mean ± a few stddevs"
= "0 ± a few $\frac{1}{\sqrt{K}}$'s".

A distributed like dot prod. of $y^{(1)}, y^{(2)}$, want it ≤ 0.1 "ε"

So want $\frac{1}{\sqrt{K}} \approx \varepsilon \iff K \approx 1/\varepsilon^2$. [that was 100, for $\varepsilon = 10\%$]

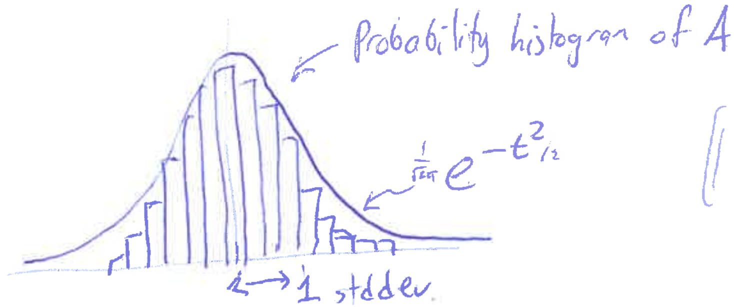
[OK, but that's just one dot product $y^{(1)} \cdot y^{(2)}$. Want it $\approx \frac{1}{\sqrt{K}}$ for all $\binom{n}{2}$ ...]

Q: What is $\Pr[A \geq 20 \cdot \frac{1}{\sqrt{K}}]$? [[I just made up "20".]]

Heuristic fact: linear combo of random bits, [[indeed, any "nice" r.v.'s]]

$$c_1 B_1 + \dots + c_K B_K,$$

acts like a Gaussian random var. [[of same mean, stddev.]]



[["Bell curve"]]

$$\Rightarrow \Pr\left[|A| \geq t \cdot \text{stddev}\right] \leq e^{-\Theta(t^2)}$$

$\therefore$ for $t = \text{Const} \cdot \sqrt{\log n}$, can make expon. like $-5\ln(n)$.

$$\Rightarrow \Pr\left(|A| \geq C \cdot \sqrt{\log n} \cdot \tfrac{1}{\sqrt{K}}\right) \leq e^{-3\ln(n)} = \frac{1}{n^5}.$$

$\therefore$ any particular pair $y^{(i)}, y^{(j)}$ has Prob $\leq \frac{1}{n^5}$ of having

$$|\text{dot prod}| \geq C \cdot \sqrt{\log n} \cdot \tfrac{1}{\sqrt{K}}$$

There are $\binom{n}{2} \leq n^2$ pairs $\therefore$ by Union Bound,

$$\Pr\left[\text{any pair has } |\text{dot prod}| \geq C \cdot \sqrt{\log n} \cdot \tfrac{1}{\sqrt{K}}\right] \leq n^2 \cdot \frac{1}{n^5} \leq \frac{1}{n^3}$$ [[like 1 in a Billion]]

$\therefore$ "w.h.p." [[with high probability]], all pairs have $|\text{dot prod}| \leq O\left(\sqrt{\log n} \cdot \frac{1}{\sqrt{K}}\right)$

$= \varepsilon$ if $K = O\left(\frac{\log n}{\varepsilon^2}\right)$ [[solving]]

[OK, but that was for a specific set of data items $x^{(1)} \dots x^{(n)}$, an orthonormal basis.]

In general, say $x^{(1)}, \dots, x^{(n)} \in \mathbb{R}^D$, let $\varepsilon > 0$ be given.

Set $K = C \cdot \frac{\log n}{\varepsilon^2}$ [[a mediumly large constant]]

5

Define:

$S = \frac{1}{\sqrt{K}} \cdot$ ($K \times D$ matrix of random ± 1's)

"random projection"

$$\underbrace{S}_{} = \frac{1}{\sqrt{K}} \cdot \overbrace{\begin{bmatrix} +1 & -1 & -1 & +1 & \cdots \\ & & \vdots & & \\ & \cdots & & & +1 \end{bmatrix}}^{D}_{K \text{ rows}} \begin{bmatrix} \\ x^{(i)} \\ \\ \end{bmatrix} =: \begin{bmatrix} y^{(i)} \end{bmatrix}$$

Theorem ("J.L. Lemma"): With high prob, (failure prob. $\ll \frac{1}{n}$),

Johnson–Lindenstrauss

$(1-\varepsilon)\|x^{(i)}\| \le \|y^{(i)}\| \le (1+\varepsilon)\|x^{(i)}\|$, ① ("lengths preserved")

$\forall i, j.$

$(1-\varepsilon)\|x^{(i)} - x^{(j)}\| \le \|y^{(i)} - y^{(j)}\| \le (1+\varepsilon)\|x^{(i)} - x^{(j)}\|$ ② ("distances preserved")

__Proof sketch:__ Suffices to do ①. Why? Say we know ① true.

~~"lengths preserved"~~ Given x's, define $z^{(ij)} := x^{(i)} - x^{(j)}$.

Like $\binom{n}{2}$ new data points

① $\Rightarrow$ $\|Sz^{(ij)}\|_2 \approx \|z^{(ij)}\|$

$\Rightarrow$ $\|S(x^{(i)} - x^{(j)})\| \approx \|x^{(i)} - x^{(j)}\|$

$= \|Sx^{(i)} - Sx^{(j)}\|$ $\Leftrightarrow$ ②

(ok, "n" is now "$\binom{n}{2}$", but $\log \binom{n}{2} = O(\log n)$)

① $\Leftrightarrow$ $(1-\varepsilon)^2 \|x^{(i)}\|^2 \le \|Sx^{(i)}\|^2 \le (1+\varepsilon)^2 \|x^{(i)}\|^2$.

$(1 \pm \varepsilon)^2 \approx 1 \pm 2\varepsilon$, "$(e^{\pm\varepsilon})^2 \approx e^{\pm 2\varepsilon}$" [again $\sim O(\cdot)$]

Drop (i) for notational simplicity.

$\Leftrightarrow$ $(1 - 2\varepsilon)\sum x_j^2 \le Sx \cdot Sx \le (1+2\varepsilon)\sum x_j^2$.

$$S = \frac{1}{\sqrt{K}} \begin{bmatrix} B_{11} & \cdots & B_{1D} \\ \vdots & & \\ B_{K1} & \cdots & B_{KD} \end{bmatrix}$$

$$Sx = \frac{1}{\sqrt{K}} \begin{bmatrix} B_{11}x_1 + B_{12}x_2 + \cdots + B_{1D}x_D \\ B_{21}x_1 + \cdots \cdots B_{2D}x_D \\ \vdots \\ B_{K1}x_1 + \cdots + B_{KD}x_D \end{bmatrix}$$

Dot with itself...

$$Sx \cdot Sx = \frac{1}{K}\underbrace{(B_{11}x_1 + \dots + B_{1D}x_D)^2}_{\text{looks like}} + \frac{1}{K}(B_{21}x_1 + \dots + B_{2D}x_D)^2 + \dots$$

(K similar, independent terms)

~~expand~~

$$(+x_1 - x_2 - x_3 + x_4 - \dots x_D)^2$$

$$= x_1^2 + x_2^2 + \dots + x_D^2 + \text{cross-terms}$$

$\pm x_1x_2 \pm x_1x_3 \pm \dots$ ← random $\pm$ bits

$$= \|x\|^2 + \text{cross terms}$$

$$\therefore\ Sx \cdot Sx = \cancel{\tfrac{D}{K}}\ \|x\|^2 + \frac{1}{K}(D \text{ sets of indep. cross-terms})$$

$$\therefore\ \mathbb{E}[Sx \cdot Sx] = \|x\|^2 + 0 \quad ☺.$$

exercise: stddev $[\text{all cross terms}] \leq \frac{3}{\sqrt{K}} \cdot \|x\|^2$.

$\therefore$ heuristically, $Sx \cdot Sx \sim \|x\|^2 \pm \frac{\text{const.}}{\sqrt{K}} \cdot \|x\|^2$

↑ ε if $K = O(1/\varepsilon^2)$

painful math [but similar bell curve heuristic]

$$\Pr[\,Sx \cdot Sx \text{ not in } (1 \pm \varepsilon)\|x\|^2\,] \leq \frac{1}{n^5}\ ☺$$

Small Enough for union bound. $\square$

Remarks: Storing S, computing Sx is somewhat costly.

Known: · S can be $\approx$ sparse: only ε frac. of entries ± 1, rest 0

· OK if S not random, just comes from "random enough" hash function

· Alternate method [Ailon–Chazelle] using Fast Fourier Transform:

Can compute Sx in $\approx O(D \log D) + O(K \log^2 D)$ time instead of $O(KD)$.