Lecture 22
- high-dimensional spaces
- Random projections
- Pseudoinverse

Consider random pt. $\vec{p} \sim \mathcal{N}(0, I) \in \mathbb{R}^d$

What is the distribution of its length?
- You'd think each component is close to 0
  ↳ this is wrong in higher dimensions

Turns out majority of pts. are at approximately the same distance from the mean
↳ they lie on a thin shell

Squared distance: $\|\vec{p}\|_2^2 = p_1^2 + \dots + p_d^2$

each p_i sampled independently from a univariate standard normal distr.

Sum of each component comes from a chi-squared distribution:

$$p_i \sim \mathcal{N}(0,1) \quad p_i^2 \sim \chi^2(1) \quad E[p_i^2] = 1 \quad Var(p_i^2) = 2$$

$$\Rightarrow E[\|\vec{p}\|^2] = d \cdot E[p_i^2] = d$$

$$Var(\|\vec{p}\|^2) = d\, Var(p_i^2) = 2d$$

$$\sigma(\|\vec{p}\|^2) = \sqrt{2d}$$

large $d \to \|\vec{p}\|$ concentrated in thin shell w/ radius $\approx \sqrt{d}$ and a thickness $\propto (2d)^{1/4}$

Mean $E[\|p\|]$ not exactly $\sqrt{d}$, but close since std deviation $\sqrt{2d}$ much smaller than d
↳ same w/ $\sigma(\|p\|)$

What about a uniform distribution?
- Consider 2 spheres of radii r and $r-\varepsilon$
  - Volume of outer ball proportional to r^d
  - Vol. of inner proportional to $(r-\varepsilon)^d$

$$\Rightarrow \frac{V_1}{V_0} = \frac{(r-\varepsilon)^d}{r^d} = \left(1 - \frac{\varepsilon}{r}\right)^d \approx e^{\frac{-\varepsilon d}{r}}$$

ratio small for a large $d \Rightarrow$ most of volume comes from very thin shell of outer sphere
e.g. $\frac{\varepsilon}{r} = 0.1$ & $d = 100 \Rightarrow V_1$ has $0.9^{100} = 0.0027\%$ of volume

→ random points from uniform & Gaussian distributions in high dimensions almost always fall in some outer shell
⇒ in high dimensions, nearest neighbour & 1000-nearest neighbors don't differ by much
⇒ k-nearest neighbors & k-means clustering are less effective for large d

Angles Between Random Vectors

What is the angle θ between $\vec{p} \sim N(0, I)$ & an arbitrary $\vec{q} \in \mathbb{R}^d$?

W/o loss of generality, let $\vec{q} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$ * q's value doesn't matter since p is random in all directions

$$\cos\theta = \frac{\vec{p}^T\vec{q}}{\|\vec{p}\|\|\vec{q}\|} = \frac{p_1}{\|\vec{p}\|}, \quad E[\cos\theta] = 0, \quad \sigma(\cos\theta) \approx 1/\sqrt{d}$$

($\mu = 0, \sigma = 1$ for p_1; $\|\vec{p}\| \sim \sqrt{d}$)

→ large $d \Rightarrow \cos\theta \approx 0$, $\cos\theta \approx 0 \Rightarrow \theta \approx 90°$

→ Vectors close to being orthogonal

Random Projection

- alternative to PCA as preprocessing for clustering, classification, and/or regression
- approximately preserves distances between points

Procedure: project onto random subspace (instead of PCA subspace)
- Sometimes preserves distance better than PCA
- Works best when you project high-dim. space to medium-dim subspace
- Similar distance ⟹ k-means & nearest neighbor perform similarly but quicker

• Pick small ε, small δ, random subspace $S \subset \mathbb{R}^d$ of dimension k

$$k = \left\lceil \frac{2\ln(1/\delta)}{\frac{\varepsilon^2}{2} - \varepsilon^3/3} \right\rceil$$ ← doesn't depend on d

round up →

→ for any pt. $\vec{q}$, let $\hat{q}$ be orthogonal projection onto S, multiplied by $\sqrt{\frac{d}{k}}$ (helps preserve distances)

Johnson-Lindenstrauss Lemma: for any $\vec{q}, \vec{u} \in \mathbb{R}^d$ (distance between original pts.)

$$(1-\varepsilon)\|\vec{q}-\vec{u}\|^2 \le \|\hat{q}-\hat{u}\|^2 \le (1+\varepsilon)\|\vec{q}-\vec{u}\|^2 \quad \text{w/ prob.} \ge 1-2\delta$$

(since this process is random)

($\hat{q}, \hat{u}$: projected pts.)

ε typically $\in [0.02, 0.5]$ $\quad \delta \in [1/n^3, 0.05]$

W/ these values, squared distance after projection may change by 2% to 50%

experiment w/ k to find best speed-accuracy tradeoff

Small δ, $\delta < 1/n^2 \Rightarrow$ inter-sample-pt. distances small

↳ Subspace of dim. $\Theta(\log n)$

- reducing δ doesn't cost much, reducing ε costs more
- w/ random projections, now can bring 1,000,000 → 1,000 w/ 6% error

Why does this work?

random projection of $\vec{q}-\vec{u}$ is like taking a random vector & selecting k components

↳ mean of squares of the k components approximates mean for whole population

how do we get a uniformly distributed random proj. direction?

- choose each comp. from univariate gaussian & normalize

how do we get a random subspace?

1) choose k random directions
2) use GS to make them mutually orthogonal

Pseudoinverse, SVD

Back to supervised learning!

Suppose $D \in \mathbb{R}^{n \times d}$ is diagonal

then we find D^+ by transposing D & replacing every nonzero value w/ its reciprocal

→ $\mathbf{D} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{3} \\ 0 & 0 & 0 \end{bmatrix}$ $\mathbf{D}^\dagger = \begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 \end{bmatrix}$ $\mathbf{DD}^\dagger = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$ $\mathbf{D}^\dagger\mathbf{D} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

→ If D square w/ nonzero diagonal entries, then D^+ is its inverse, $DD^+ = D^+D = I$

In general, DD^+ & D^+D are always diagonal matrices w/ 1's & 0's

- $DD^+D = D$, D^+DD^+ and $D^2D^+ = D$

Now, Pseudoinverse of arbitrary $X \in \mathbb{R}^{n \times d}$ w/ SVD $X = UDV^T$, $rk(X) = rk(D)$

then $X^+ = VD^+U^T$

Properties: 1) $XX^+ = U(DD^+)U^T$ which is symm. PSD

2) $X^+X = V(D^+D)V^T$ also symm PSD

3) Same rank: $D, D^+, DD^+, D^+D, X, X^+, XX^+, X^+X$

4) if $rk(X) = n$, then $XX^+ = I_{n \times n}$, X^+ is right pseudoinverse

5) if $rk(X) = d$, $X^+X = I_{d \times d}$, X^+ is left inverse

6) $XX^+X = X$ 7) $X^+XX^+ = X^+$

* Pseudoinverse always gives good solution to LS linear regression even when X^TX is singular

Proof: Show $\vec{w} = X^+y$ is a solution to $X^TX\vec{w} = X^T\vec{y}$

- if normal equations have multiple solutions, then $\vec{w} = X^+\vec{y}$ is the min. norm sol.

X^+ helpful when X^TX is singular since n-d pts. lie on subspace of feature space

- if X has small singular value, then its reciprocal is large, large effect on $\vec{w}$
  - if $\sigma_i = 0$, that value has no effect on $\vec{w}$

if we have a very small σ, should we pretend it's 0?

- ridge regression kinda does this