Lecture 11: Principal Component Analysis (PCA) ©

Last time: Say A is symmetric, $A = A^T$, $D \times D$ matrix.

Then there ~~are~~ is an orthonormal basis

$\vec{v}_1, \dots, \vec{v}_D$ ("eigenvectors") and numbers

$\lambda_1, \dots, \lambda_D$ ("eigenvalues") such that

$A \equiv$ "stretch by factor λ_j in dir. $\vec{v}_j$ $\forall j$".

Saw alg. running in time $O\left(\frac{\log D}{\log\left(\frac{\lambda_{max}}{\lambda_{2nd\,max}}\right)}\right)$ to find

[approximations of] λ_{max}, v_{max}.

[assumed λ_j's all ≥ 0. If not, put $|\cdot|$ signs.]

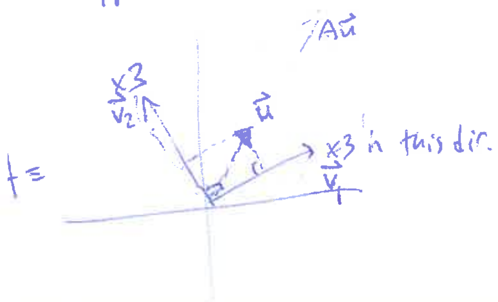
[Can iterate to find 2nd max, 3rd max, etc.]

Worry: If $\lambda_{max} = (1+\varepsilon)\lambda_{2nd}$, denom. is $\log(1+\varepsilon) = \Theta(\varepsilon)$.

If $\lambda_{max} = \lambda_{2nd}$, denom is 0!

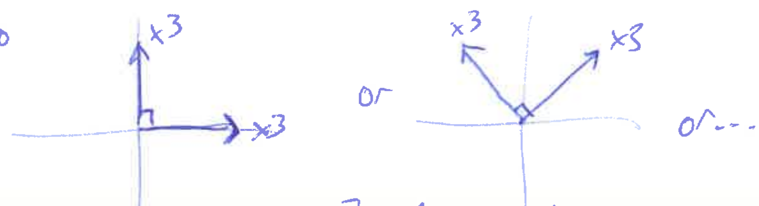
[I'm here to tell you it's not really a problem.]

Suppose $\lambda_{max} = \lambda_{2nd\,max} = 3$. Indeed, say just $D = 2$.



Then actually $A =$ "scale by 3".

Equiv. to



["eigenvectors" $\vec{v}_1, \vec{v}_2$ not ~~even~~ uniquely defined.] Any two orthonormal vecs in this 2-D space are valid "eigenvecs of stretch 3,3".

Our alg. will just find a random pair, and that's okay!

[efficiently, if $\frac{\lambda_{2nd\,max}}{\lambda_{3rd\,max}}$ large]

[On to PCA...] [Back to "big data" scenario.]

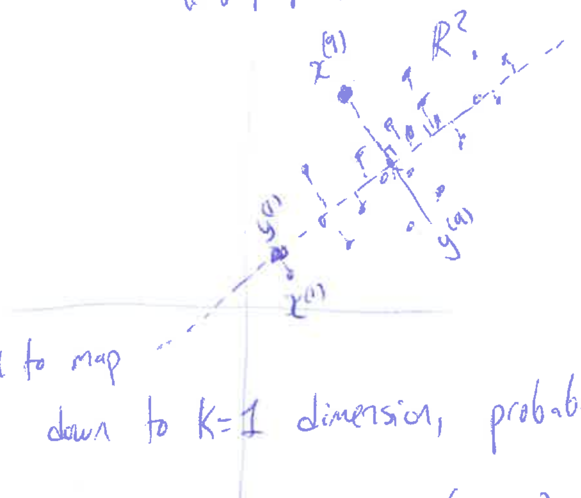
Say we have "items" $x^{(1)}, \dots, x^{(n)}$, each with D numerical features, [e.g. people]

so $x^{(j)} \in \mathbb{R}^D$

$[\; — x^{(j)} — \;]$

[PCA people like the items as row vectors, which annoys me, but oh well.]

e.g. $D=2$ [hard to draw $D=3$]



[If you wanted to map these points down to $K=1$ dimension, probably dashed line is best...]

Find the "best fitting" line ($K=1$), or 2D-plane ($K=2$), 3D-plane ($K=3$), etc.

(Focus on $K=1$ (line) for now.)

Goal: minimize $\sum_{j=1}^{n} (err^{(j)})^2$, where $err^{(j)} = \|x^{(j)} - y^{(j)}\|$

$y^{(j)}$: projection of $x^{(j)}$ onto K-dim space

[Why square the distance? Well, it makes the math nice :-.]

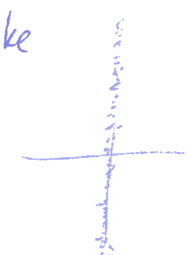
Preprocessing

[Linear algebra doesn't like lines/planes not thru origin.
So translate your data so centered at origin (which turns out to make best-fitting line thru origin too)]

- Compute avg. of all $x^{(j)}$'s (rows), ["center of mass"] then subtract it from each [no big deal...]

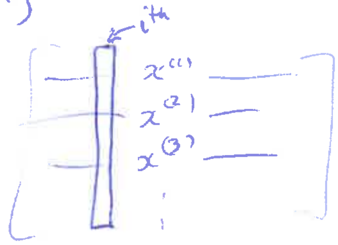
  $\rightsquigarrow$ Now we can assume $\operatorname{avg}_{j=1}^{n} (-x^{(j)}-) = [0\,0 \cdots 0]$. :)

[Imagine feature 1 (column) is a rating from −5 to 5, and feature 2 is a rating from −10000 to 10000. Data will look like



and best-fitting line will be vertical, for a dumb reason.]

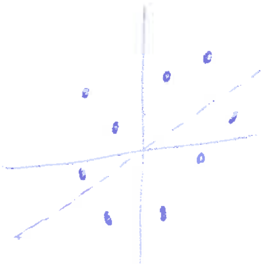
- Scale i^{th} column



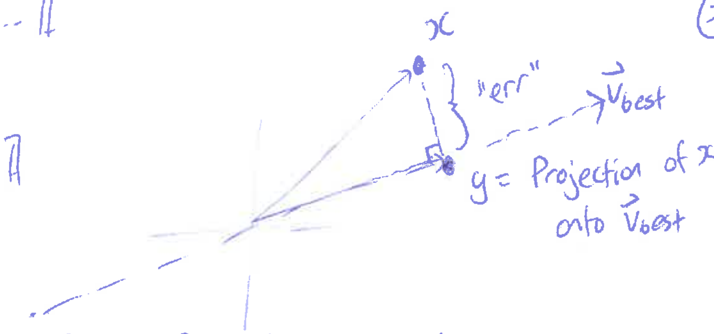
so it's a unit-length vector.

[Kinda just changing the "units" feature i is measured in.]

[[Okay, enough warmup, let's get to it...]]



[[Zoom in on one point...]]

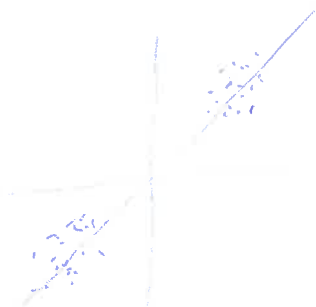


note: For any x, $err^2 = \|x\|^2 - \|y\|^2$ (Pythagoras)

$\therefore$ making err^2 small $\equiv$ making $\|y\|^2$ big [[$\|x\|^2$ is fixed]]

$\therefore$ "best-fitting line" is equivalently the one in direction $\vec{V}_{best}$

which maximizes $\sum_{j=1}^{n} \| Proj_{\vec{v}_{best}}(x^{(j)}) \|^2$,

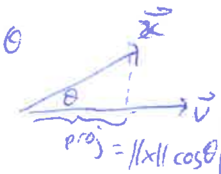


[[also an interesting interpretation: "maximizing variance"]]

$$\|Proj_{\vec{v}}(\vec{x})\| = \underbrace{|\vec{v}\cdot\vec{x}|}$$

unit.

[[$\vec{v}\cdot\vec{x} = \underset{1}{\|\vec{v}\|}\cdot\|x\|\cdot\cos\theta$



]]

$$\therefore \|Proj_{\vec{v}}(\vec{x})\|^2 = (\vec{v}\cdot\vec{x})^2$$

$$\vec{v}\cdot\vec{x} = \begin{bmatrix} - & \vec{v} & - \end{bmatrix}\begin{bmatrix} | \\ \vec{x}^T \\ | \end{bmatrix} = \begin{bmatrix} - & \vec{x} & - \end{bmatrix}\begin{bmatrix} | \\ \vec{v}^T \\ | \end{bmatrix}$$

$$\therefore (\vec{v}\cdot\vec{x})^2 = \begin{bmatrix} - & \vec{v} & - \end{bmatrix}\begin{bmatrix} | \\ \vec{x}^T \\ | \end{bmatrix}\begin{bmatrix} - & \vec{x} & - \end{bmatrix}\begin{bmatrix} | \\ \vec{v}^T \\ | \end{bmatrix}$$

So we want unit vector $\vec{v}_{best}$ maximizing

$$\sum_{j=1}^{n} \vec{v}\, x^{(j)T} x^{(j)} \vec{v}^T = \vec{v}\Big(\underbrace{\sum_{j=1}^{n} x^{(j)T} x^{(j)}}_{\begin{bmatrix} | \\ x \\ | \end{bmatrix}\begin{bmatrix} - & x & - \end{bmatrix}\ \text{D×D matrix}}\Big)\vec{v}^T$$

$= \vec{v} A \vec{v}^T$, where

$$A := \sum_{j=1}^{n} x^{(j)T} x^{(j)}$$

Exercise: If $X = \begin{array}{c} \begin{bmatrix} — & x^{(1)} & — \\ — & x^{(2)} & — \\ & \vdots & \end{bmatrix} \\ \longleftarrow D \longrightarrow \end{array} \updownarrow n$, then $A = X^T X$.

$$X^T X = \begin{bmatrix} | & | & \\ x^{(1)} & x^{(2)} & \cdots \\ | & | & \end{bmatrix} \begin{bmatrix} — & x^{(1)} & — \\ — & x^{(2)} & — \\ & \vdots & \end{bmatrix}$$

[It's literally just by the def^n of how matrix mult. works.]

Ta-da, A is a symmetric matrix ☺. $A^T = (X^T X)^T = X^T X^{TT} = X^T X = A$.

[We're so happy, we understand symmetric matrices!]

~~Which vector v makes $[—v—] A \begin{bmatrix} | \\ v \\ | \end{bmatrix}$ biggest?~~

We know $A \equiv$ stretch by $\lambda_1, \ldots, \lambda_D$ in dir. $v_1, \ldots, v_D$ (unit basis)

$\Rightarrow A v_j = \lambda_i v_i \Rightarrow v_i \cdot (A v_i) = v_i^T A v_i = v_i \cdot (\lambda_i v_i) = \lambda_i$, since $v_i \cdot v_i = 1$

$v_i^T X^T X v_i = (X v_i)^T (X v_i) = (X v_i) \cdot (X v_i) = \|X v_i\|^2$

$\therefore \lambda_i = \|X v_i\|^2 \geq 0$

[all eigvals ≥ 0, extra ☺]

And now what $\begin{bmatrix} | \\ \vec{v}_{best} \\ | \end{bmatrix}$ makes

$$\underbrace{[— \vec{v}_{best} —] \underbrace{A \begin{bmatrix} | \\ \vec{v}_{best} \\ | \end{bmatrix}}_{\text{rotates in dir. of } v_{max}}}_{\text{dot product...}}$$ largest?

$\therefore$ easy to see it's just v_{max}.

Conclusion: $\vec{v}_{best}$ = max eigenvector of $A := X^T X$

[which we learned how to find last time]

Exercise/extension: For original data $X = \begin{bmatrix} — x^{(1)} — \\ — x^{(2)} — \\ \vdots \end{bmatrix}$,

best-fitting K-dimensional subspace [we just did $K=1$],

= "most important K dirs"

is given by top K eigenvectors of largest eigenvalues of $A = X^T X$.

[which we also discussed finding.]

↳ Computing these is called "PCA: Principal component analysis."

Applications:

① Data visualization: pick $K = 2$ or 3, project data onto K "principal components", $y^{(i)} = \begin{bmatrix} \vec{v}_{best1} \cdot x^{(i)} \\ \vec{v}_{best2} \cdot x^{(i)} \\ \vec{v}_{best3} \cdot x^{(i)} \end{bmatrix}$,

plot.

② Inferring latent features of data

$$\text{people} \updownarrow \begin{bmatrix} \xleftarrow{\quad \text{shows} \quad} \\ \text{ranking} \\ \\ \end{bmatrix}$$

$\vec{v}_{best} = [\quad]$ is how much each show satisfies "Mystery Feature" F

$$\begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{bmatrix} = \begin{bmatrix} x^{(1)} \cdot \vec{v}_{best} \\ x^{(2)} \cdot \vec{v}_{best} \\ \vdots \end{bmatrix}$$

gives how important "F" is to each person

2nd ~~best~~ best direction gives 2nd "mystery feature" F_2, orthog to 1st., etc.

[Maybe F's are "quality" or "funniness" or "length" or ....]

③ Clustering data:

[now look for nearest neighbors]