Lecture 10: Vector Spaces, Eigenvectors

Not as bad, but still a bit annoying.

Rotates vector by \(20\) degrees.

To find symmetric:

\[
A = \begin{bmatrix}
34 & 0 \\
0 & 34
\end{bmatrix}
\]

Check: \(A = AV_A\)

Along \(x\)-axis, factor 5.

\[A \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 5x \\ 0 \end{bmatrix} \]

\[A \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 5y \end{bmatrix} \]

Symmetric: \(A = A^T\)

When is a matrix?

Why is a matrix?

I can't do form like this.

Not the hardest thing.

Factors: symmetric (or rectangular). Non-symmetric (or rectangular).

directed vs. undirected.

dimension. points and planes.

First, I need algebra. First, some (linear algebra)?

So I don't want to miss out on the algebra.

Today, we'll (eventually) talk about "principal component analysis (PCA),"

1
Linear Algebra Fact: Let $A$ be a $D \times D$ symmetric matrix. Then there is some orthogonal basis of unit vectors $\mathbf{v}_1, \ldots, \mathbf{v}_D$ such that $A = \text{"stretch by } \lambda_j \text{ in direction } \mathbf{v}_j \text{"}$.

"eigenvectors"

So symm. matrices are really simple: just "stretchers", in some basis!

Q: How to find $\mathbf{v}_j$'s and $\lambda_j$'s?

A1: Type "eigs (A)" in Matlab ;-) [But how does it do it?]

A2: Some Gaussian-elimination-type alg. you learn in linear algebra class. Time is $\Theta(D^3)$, yuck.

A3: Assume, for simplicity, $\lambda_j$'s all $\geq 0$. ["Positive semi-definite matrix.

Not easy to tell, given, $A$, but also not hard to handle.

But in future application to PCA, it will be true.]

(Fuzzy) Alg.

1. Pick a random vector $\mathbf{u}_0$, make it unit-length by replacing with $\mathbf{u}_0 / \lVert \mathbf{u}_0 \rVert$

2. For $i = 1, 2, 3, \ldots$ [people often use Gaussians so it's rotationally symm., but doesn't matter really; $\pm$ works ok.]

   $\mathbf{u}_i := A \mathbf{u}_{i-1}$, then make it unit

   If $\mathbf{u}_i \approx \mathbf{u}_{i-1}$, up to some precision break.

3. $\mathbf{v}_{\text{max}} := \mathbf{u}_i$ // find "first" eigenvec.

   $A \mathbf{v}_{\text{max}} = \lambda_{\text{max}} \mathbf{v}_{\text{max}}$ for some scalar $\lambda_{\text{max}}$; calculate it.

This (heuristically) finds the largest scale factor (eigenval.) $\lambda_{\text{max}}$ and $\mathbf{v}_{\text{max}}$. ["Repeat?" We'll come back to that.]
After eig. v's, coord. #3 of v₁ is 10, x₂ = 7, all others A₁, 1 > 9.

Imaginary, e.g., A₂ = 10, all others A₂, 1 > 0.

So v₀ = x₁, vₙ = x₁^ₙ

For analysis, don't need to keep unifying...

W.O.C.: vₙ = 0. So A = 1... D. So A

... may as well assume if the "usual" axis is the x's, "usual" axis is the y's...

Analysis: Though the eig. doesn't have eigen-values vₙ = yₙ, we have eig. vectors yₙ = xₙ; yₙ = xₙ

Keeps finding toward axes of most stretch.

Matrix: Dₙ = xₙ, E.g., Aₙ, Dₙ = xₙ

Matrix, Dₙ = xₙ, time ago, since input Aₙ is a

Kinds of a linear-time alg, since input Aₙ is a list

Usually small, so we'll see it

Time: O(1)
We find the direction of greatest stretch.

In this case, after unitizing \( u \), (cooks like)  

In this case where \( \frac{b}{a} \gg \frac{1}{\log} \) "cooks" in the special gap.

The ratio \( \theta \) called the

All cooks other than \( \delta \) and \( \theta \) in \( \theta (816) \)

...
This worked well because we assumed $\lambda_{\text{max}} > \lambda_{\text{2nd max}}$ by a decent factor.

What if $\lambda_{\text{max}} = (1 + \epsilon) \lambda_{\text{2nd max}}$?

# iters needed: $t \gg \log D \log(\frac{\lambda_{\text{max}}}{\lambda_{\text{2nd max}}}) \log(1 + \epsilon)$

How much is $\log(1 + \epsilon)$? ($1 + \epsilon \approx e^\epsilon$)

$\ln(1 + \epsilon) \approx \epsilon$]

\[ If it's even possible \epsilon = 0 \ldots \]

But actually, it's sorta not that bad.

Imagine $\lambda_{\text{max}} = \lambda_{\text{2nd}}$. E.g. $A = \begin{pmatrix} \text{stretch}_x \text{ stretch}_y \end{pmatrix}$

This is same as $\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ or any basis. So "correct" axes not even defined.

Alg. will just return some vector in the subspace spanned by all those axes $\vec{v}$ with $\lambda$'s tied for $\lambda_{\text{max}}$.

Repeat? This just finds the direction of max stretch.

Want to find whole basis, all stretch factors
Or maybe, at least, top few largest stretches.

E.g., $\vec{v}_{\text{max}}$ we just figured this out.

Now want to figure this out.
Idea: Repeat algo, but **stay perpendicular to axis (axes)** found already. **Stay in subspace of mystery axes**

→ If you start **with** \( \hat{u}_0 \) perpendicular to \( \hat{v}_{\text{max}} \) ["0" in the direction \( \hat{v}_{\text{max}} \)] all \( \hat{u}_k \)'s will be too, and you'll find \( \hat{v}_{\text{and max}}, \hat{v}_2 \text{ max} \).

How to achieve that? I(?

Init: Pick \( \hat{u}_0 \) at random.

→ Replace with 
\[
\hat{u}_0 \rightarrow \hat{u}_0 - \text{Proj}_{\hat{v}_{\text{max}}} (\hat{u}_0)
\]

(\text{len. of } \hat{u}_0 \text{ projected onto } \hat{v}_{\text{max}}) \cdot \hat{v}_{\text{max}}

\[
\hat{u}_0 \cdot \hat{v}_{\text{max}} = ||\hat{u}_0|| \cdot ||\hat{v}_{\text{max}}|| \cdot \cos \theta
\]

\( \text{[Can repeat, projecting away all the eigenvectors you've found so far. Finds "top k eigenvectors" in time } O(kD \log D) \text{]} \)

Why? When do you ever have this situation??

Let \( x^{(1)}, \ldots, x^{(n)} \in \mathbb{R}^D \) be "data items"

Let 
\[
X = \begin{bmatrix}
- x^{(1)} \\
- x^{(2)} \\
\vdots \\
- x^{(n)}
\end{bmatrix}
\]

\( \text{[Like last time. I. Let } A = X^T X = [ I ] \)

\( A^T = (X^T X)^T = X^T X = A \)

\( \Rightarrow A \text{ symmetric & square } \)

Also, all \( A_{ij} \) > 0 \( \text{[easy to show] } \)

\( V \)'s \& \( \lambda \)'s for \( A \) give crucial info about \( x^{(i)} \)'s . . . 