

wk4 (26)

What if we want to fit a line that doesn't necessarily pass through the origin? Now we're fitting two parameters, as in $y = mx + b$. The text uses an example ^{year} vs. world record for men's mile,

1870	268.8
1880	264.5
1890	258.4

⋮ approximate

So we'd like solutions to

$$268.8 = 1870m + b$$

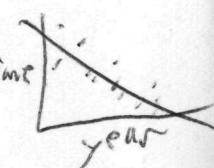
$$264.5 = 1880m + b$$

$$258.4 = 1890m + b$$

⋮

If we want to fit a line to this data, we'd have

$$[time] = m [year] + b$$



And we can write this in matrix form as

$$\begin{pmatrix} 268.8 \\ 264.5 \\ 258.4 \\ \vdots \end{pmatrix} = \begin{pmatrix} 1870 \\ 1880 \\ 1890 \\ \vdots \end{pmatrix} m + \begin{pmatrix} b \\ b \\ b \\ \vdots \end{pmatrix} = \begin{pmatrix} 1870 \\ 1880 \\ 1890 \\ \vdots \end{pmatrix} m + \begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \end{pmatrix} b = \begin{pmatrix} 1870 & 1 & (m) \\ 1880 & 1 & (b) \\ 1890 & 1 & \\ \vdots & & \end{pmatrix}$$

"A"

Where a single parameter gives us a space of models with one dimension, this has two dimensions, because m or b can vary. In terms of the actual models that can be produced, the two basis vectors are $\begin{pmatrix} 1870 \\ 1880 \\ \vdots \end{pmatrix}$ and $\begin{pmatrix} 1 \\ 1 \\ \vdots \end{pmatrix}$ which can be independently scaled.

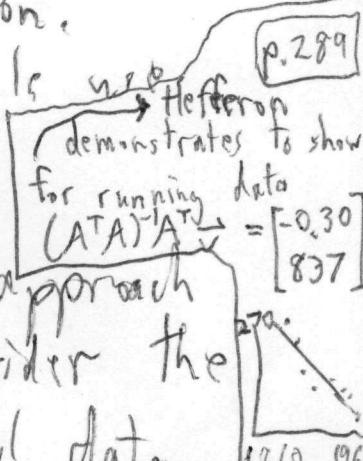
So A is the matrix of basis vectors we need for a projection,

Using the "projection onto a subspace" result, we have: data \leftarrow vector of inputs and is
 $\text{proj}_m(\vec{v}) = \vec{A}\vec{c} = \vec{A}(\vec{A}^T\vec{A})^{-1}\vec{A}^T\vec{v}$

But as with the one-dimensional case, we're more interested in the parameters of the model than the actual projection. The parameters are the vector \vec{c} , which we solved to be $(\vec{A}^T\vec{A})^{-1}\vec{A}^T\vec{v}$. This is that vector of parameters (\vec{b}) that we wanted; $\vec{A}(\vec{b})$ would give us those points on the projection.

In practice of course, most sane people use software to figure out these projections. p.289

We still haven't mentioned how this approach gives us the least squared error. But, consider the vector from the point representing the actual data to the corresponding point ^{that is} the projection. The length of this vector is $\sqrt{r_1^2 + r_2^2 + r_3^2 + \dots}$ where r_i is the residual for data point i . We determined that a projection minimizes this distance from vector to projected vector. And minimizing that minimizes its square which is the sum of squared errors. If there's a closer fit, it wasn't the projection. 



Other linear models

Last squares fits are more versatile than they first appear — for example, we can fit exponential models too as long as we're willing to be a little flexible about the error we're minimizing.

Suppose I wanted to fit a model of the form $y = e^{mx}$. I have some points:

$$\begin{array}{ll} \text{not linear!} & y = e^{mx} \\ x=1 & y=5 \\ x=2 & y=8 \\ x=3 & y=11 \end{array}$$

To turn this into a linear model, I just need to observe

~~•~~ $\ln y = \ln(e^{mx}) = mx$ \leftarrow linear!

$$\text{Now } \ln y = \ln \begin{pmatrix} 5 \\ 8 \\ 11 \end{pmatrix} = m \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} = \begin{pmatrix} 1.6 \\ 2.1 \\ 2.4 \end{pmatrix}$$

I can now calculate a least-squares fit as before.

$$\frac{\begin{pmatrix} 1.6 \\ 2.1 \\ 2.4 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}}{\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}} = \frac{13}{14} \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \text{ and we want } m \text{ to be } \frac{13}{14}.$$

$$e^{\frac{13}{14}(1)} = 2.5 \quad e^{\frac{13}{14}(2)} = 6.4 \quad e^{\frac{13}{14}(3)} = 16.3$$

We're not minimizing the actual error anymore, though — we're minimizing the square of the log of the error, which is going to result in bigger absolute errors on the high end.

Note: A more general 2-parameter exponential model can be fit in a way similar to $y = mx + b$. ~~•~~ $y = Ce^{ax}$ becomes $\ln y = \ln C + ax$; treat $\ln y$ as b , C as m .

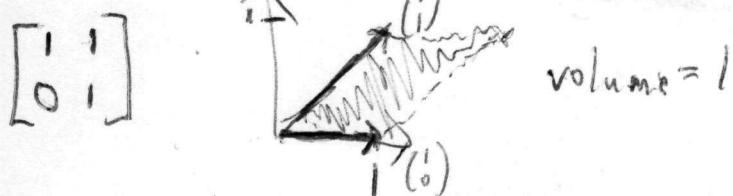
Determinants

$\det(A)$ or $\begin{vmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{vmatrix}$ ← determinant calculation

Determinants are numbers defined for square matrices that help determine whether they are singular, among other things. Before getting into how to calculate them, here are some properties of determinants:

1) They are ~~not~~ zero iff the corresponding matrix is singular. (And thus it also tells us if it's invertible.) (This fact is how the formulas are derived.)

2) If we take the columns of a matrix as vectors defining a parallelopiped (prism), the determinant is its volume, absolute value of the



Aside: 2×2 determinant formula

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad ad - bc$$

n-dimensional scales volume that is so Tossing a dimension causes flatness → no volume

3) $\det(A^T) = \det(A)$, but $\det(A^{-1}) = \frac{1}{\det(A)}$

4) The absolute value is how much the matrix scales in a linear transformation. In general, we can define determinants recursively for larger matrices by summing over smaller determinants; specifically, we can say

$$|T| = t_{i_1} \cdot T_{i_1} + t_{i_2} \cdot T_{i_2} + \dots + t_{i_n} \cdot T_{i_n}$$

where i is some row (this works for columns, too),

t_{ij} are those row entries, and T_{ij} is a determinant formed by ignoring the row and column ij in the original matrix, multiplied by $(-1)^{i+j}$.

$\det(AB) = \det(A)\det(B)$

For example, $\begin{vmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{vmatrix} = 1 \cdot \begin{vmatrix} 5 & 6 \\ 8 & 9 \end{vmatrix} - 2 \cdot \begin{vmatrix} 4 & 6 \\ 7 & 9 \end{vmatrix} + 3 \cdot \begin{vmatrix} 4 & 5 \\ 7 & 8 \end{vmatrix}$.

expand row 1

If you happen to have memorized the equation/process for 3×3 matrices, which goes

$$\begin{array}{|ccc|} \hline a & b & c \\ d & e & f \\ g & h & i \\ \hline \end{array} \rightarrow aei + bfg + cdh - geg - hfa - idb,$$

then you can see for yourself that the aforementioned definition produces the same results—but it generalizes to $n \times n$ matrices.

P.S. You can also use the determinant to find eigenvalues
(see eigenvalue/eigenvector material)

(end week 4)

Week 5

Similarity & Diagonalization

The next few tricks we'll look at will serve the following purposes:

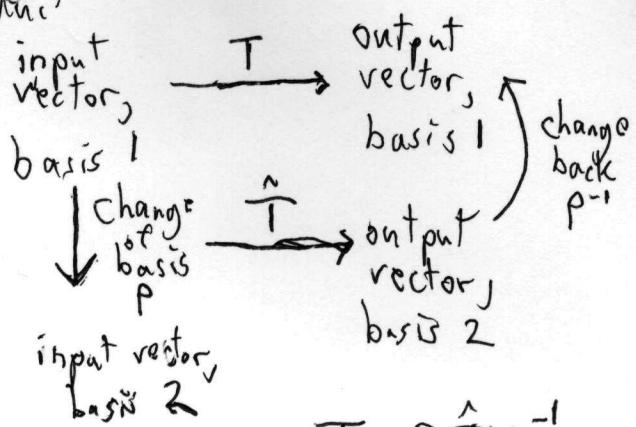
- Performing computations that make further more computationally convenient
- Determining the long-term behavior that results when we repeatedly multiply by a matrix.

We'll start with (a) and the idea of similarity.

We might sometimes want to perform computations with a change of basis, or some other transformation that makes things easier. For example, we might want to do some graphics computations rotated so that we consider a vector toward the light source to be a basis vector (for example) - maybe that will make a lot of dot products 0, or something.

~~The idea here is that, instead of computing~~

~~matrix products~~



$$T = P \hat{T} P^{-1}$$

(and, by algebra, $\hat{T} = P^{-1} T P$)

If \hat{T} exists and functions the way we want - acting the same as T but in a different basis - we can switch bases, do our work with \hat{T} instead of T , then convert back.

If $\tilde{T} = P \tilde{T} P^{-1}$, for some nonsingular matrix P , then T and \tilde{T} are called "similar." We can view them as doing basically the same work, but one is working in a transformed space. We transform with P , do the work with \tilde{T} , then transform it back.

(The book says $\tilde{T} = P \tilde{T} P^{-1}$, but this is just a change of variable names. Notice that T is similar to \tilde{T} if \tilde{T} is similar to T , and vice versa).

For example, consider the matrix $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$.

This matrix is similar to the matrix $\begin{bmatrix} 4 & 3 \\ 2 & 1 \end{bmatrix}$, which is the matrix that does the same thing but with the x and y coordinates swapped. The transformation of trading x and y coordinates is given by the matrix $P = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, which is its own inverse (trading again swaps back):

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \text{ Identity!} \quad \text{So } P^{-1} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \text{ as well.}$$

In this particular case, PTP^{-1} looks like a similar process for either matrix, since $P = P^{-1}$ in this case.

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 4 & 3 \end{bmatrix} = \begin{bmatrix} 4 & 3 \\ 2 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 4 & 3 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 3 & 4 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

These are matrices that perform actions that we can see as equivalent in their respective worlds.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 6 \\ 5 \end{bmatrix} = \begin{bmatrix} 16 \\ 38 \end{bmatrix}$$

$$\begin{bmatrix} 4 & 3 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 5 \\ 6 \end{bmatrix} = \begin{bmatrix} 38 \\ 16 \end{bmatrix}$$

Similarity is an "equivalence relation," partitioning the space of all matrices; each falls into a particular ^(similar) "equivalence class" where all the matrices are equivalent to each other. (We've argued similarity is symmetric, matrices are similar to themselves with the identity, and it's transitive because we can transform twice, then undo the transform twice.) So we might wonder if there's always some really convenient matrix to work with in the place of the original.

Diagonalization

A diagonal matrix is one with zero entries everywhere but the diagonal (where row $i \leq$ column j). The identity is an example, but so is $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 10 \end{bmatrix}$.

A matrix is diagonalizable if it is similar to a diagonal matrix — that is, for some nonsingular P , $P T P^{-1}$ is a diagonal matrix. (Notice that because similarity is transitive, all matrices that diagonalize to the same matrix are also similar to each other.)

Diagonalizing a matrix can be useful for finding powers efficiently. Suppose $D = P T P^{-1}$, where

D is a diagonal matrix. Then $T = P^{-1}DP$.

[wk 5 30]

Then $T^2 = (P^{-1}DP)(P^{-1}DP) = P^{-1}D^2P$ with the two P and P^{-1} terms in the middle canceling. And in general, $T^n = P^{-1}D^nP$. So to exponentiate, we transform once, multiply by a diagonal matrix raised to the power, then transform back again. Multiplying diagonal matrices is much easier than multiplying normal matrices, because $\begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix}^n = \begin{bmatrix} a^n & 0 & 0 \\ 0 & b^n & 0 \\ 0 & 0 & c^n \end{bmatrix}$, so this

could save a lot of operations.

We'll ^(later) go into how to find this diagonal matrix, as well as how to take advantage of this structure to look way ahead and effectively see what happens "in the limit" when we multiply a matrix by itself.

Eigenvalues & Eigen vectors

In the long term, if a linear system has any kind of stable state, we'd expect to find a vector representing that state \vec{x} such that $A\vec{x} = \vec{x}$; everything has "settled down" so that updating the state doesn't change anything. Of course, this isn't necessarily possible, but we might at least find the direction that an unstable system heads off in forever may be $A\vec{x} = \lambda\vec{x}$, so that the new vector is not the same, but is "headed" in the same direction. An eigenvector ^{of A} is a vector that remains unchanged when multiplied by A , except for its length.

If $A\vec{x} = \lambda\vec{x}$, then \vec{x} is an eigen vector of A and λ is its associated eigenvalue - a number that scales \vec{x} with every multiplication. Eigenvectors and eigenvalues can tell us about the long-term behavior of a linear system; if $\lambda > 1$, then the component in the direction of \vec{x} will be amplified forever, while if $0 < \lambda < 1$, the component will shrink to nothing.

For example, let's take the matrix

$$\begin{bmatrix} 1 & 5 \\ 2 & 4 \end{bmatrix}$$

$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ is an eigenvector of this matrix - it points the same direction after multiplication, but is scaled.

$$\begin{bmatrix} 1 & 5 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 6 \\ 6 \end{bmatrix} \text{ The eigenvalue is } 6.$$

Another eigenvector is $\begin{bmatrix} \frac{1}{2} \\ -\frac{1}{5} \end{bmatrix}$.

$$\begin{bmatrix} 1 & 5 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} \frac{1}{2} \\ -\frac{1}{5} \end{bmatrix} = \begin{bmatrix} -\frac{1}{2} \\ \frac{1}{5} \end{bmatrix} \text{ The eigenvalue is } -1.$$

We'll learn soon enough that there couldn't be more eigenvalues than this. First, let's try to solve for some eigenvalue/eigenvector pairs.

We know that the matrix doesn't change its eigenvectors besides scaling by the eigenvalue. So:

$$\begin{bmatrix} 1 & 5 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Subtract $\lambda \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ from both sides:

$$\begin{bmatrix} 1 & 5 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \lambda \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Now we'll just work the λx_1 and $-\lambda x_2$ factors into the matrix itself.

$$\begin{bmatrix} 1-\lambda & 5 \\ 2 & 4-\lambda \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Now recall that vectors sent to $\vec{0}$ are a part of the "kernel" and this is larger than $\vec{0}$ only if the matrix is singular (otherwise the matrix preserves all the dimensions of the input). So we're now trying to determine the conditions under which $\begin{bmatrix} 1-\lambda & 5 \\ 2 & 4-\lambda \end{bmatrix}$ is singular. We have a handy way of doing that: use the determinant, which is 0 iff the matrix is singular.

$$\begin{vmatrix} 1-\lambda & 5 \\ 2 & 4-\lambda \end{vmatrix} = 0 \quad (1-\lambda)(4-\lambda) - 5 \cdot 2 = 0 \\ 4 - 5\lambda + \lambda^2 - 10 = 0 \\ \lambda^2 - 5\lambda - 6 = 0$$

Notice that the solutions at $(\lambda-6)(\lambda+1)=0$ right match the eigenvalues we saw $\lambda=6$ or -1 before. To find eigenvectors we can plug these values back into our multiplication equation (note any vector in the right direction is an eigenvector):

$$\begin{bmatrix} 1-6 & 5 \\ 2 & 4-6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \begin{aligned} -5x_1 + 5x_2 &= 0 \\ 2x_1 - 2x_2 &= 0 \end{aligned} \quad \vec{x} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \text{ (or any multiple)}$$

$$\begin{bmatrix} 1 - (-1) & 5 \\ 2 & 4 - (-1) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$2x_1 + 5x_2 = 0$$

$$2x_1 + 5x_2 = 0$$

$$\vec{x} = \begin{bmatrix} \frac{1}{2} \\ -\frac{1}{5} \end{bmatrix}$$

In general, we can find eigenvectors and eigenvalues by:

- Finding the determinant of $A - \lambda I$
- Solving for λ if the determinant is 0
- Substituting back into $A - \lambda I = \vec{0}$ to find the eigenvector for each λ .

Notice that it's also relatively easy to check your work here: try the eigenvector, see if it comes out scaled by the eigenvalue.

Application to Markov Chains & PageRank

In CS1800, we cover Markov Chains - random processes where the probability of each place to jump to in the next state depends on the current state. The text's example describes a chain where the state is a restaurant (Bertucci's, Margarita's, Sato's) and the chance of being in each state in the next time step is given by the equations:

$$\begin{aligned} \text{pr}(b_{i+1}) &= 0.7 \text{pr}(b_i) + 0.3 \text{pr}(m_i) + 0.3 \text{pr}(s_i) \\ \text{pr}(m_{i+1}) &= 0.2 \text{pr}(b_i) + 0.6 \text{pr}(m_i) + 0.2 \text{pr}(s_i) \\ \text{pr}(s_{i+1}) &= 0.1 \text{pr}(b_i) + 0.1 \text{pr}(m_i) + 0.5 \text{pr}(s_i) \end{aligned}$$

That is the chance of being in a state at the next time is the sum of [transition prob]*[state] over all the states.

We now recognize this as a linear system of equations, multiplying by a probability vector gives the probabilities for the next time step.

$$\begin{bmatrix} 0.7 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.5 \end{bmatrix} \begin{bmatrix} \text{pr}(b) \\ \text{pr}(m) \\ \text{pr}(s) \end{bmatrix}$$

Markov chains have stationary distributions, the probabilities of each state in the long term after the chain has had time to "mix." The thing we know about the stationary distribution is that it's the vector of probabilities that doesn't change once we reach it. In other words, it's an eigenvector.

$$\begin{bmatrix} 0.7-\lambda & 0.3 & 0.3 \\ 0.2 & 0.6-\lambda & 0.2 \\ 0.1 & 0.1 & 0.5-\lambda \end{bmatrix} = (0.7-\lambda)(0.6-\lambda)(0.5-\lambda)$$

... actually, we must want an eigenvalue of 1 if the vector stays the same. So then we want to solve

$$\begin{bmatrix} -0.3 & 0.3 & 0.3 & 0 \\ 0.2 & -0.4 & 0.2 & 0 \\ 0.1 & 0.1 & -0.5 & 0 \end{bmatrix}$$

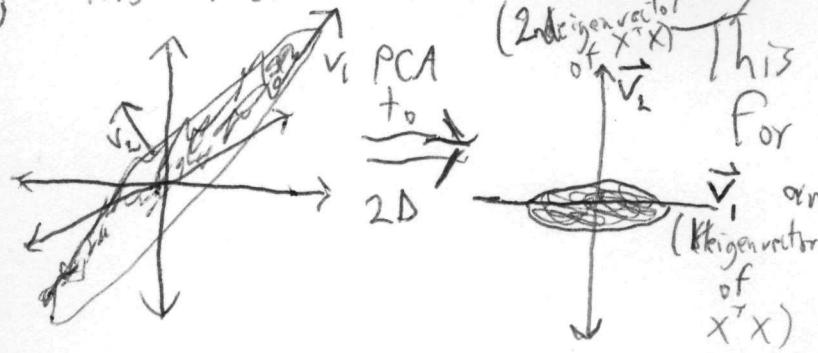
which has solutions $c \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix}$. Adding the constraint $x+y+z=1$ gives the stationary dist $\begin{bmatrix} \frac{1}{6} \\ \frac{2}{6} \\ \frac{3}{6} \end{bmatrix}$.

Application Sketch: Dimensionality Reduction

~~Part 3~~

Sometimes we want to visualize data that is high-dimensional, but have to deal with the fact that we can only see a few dimensions (2 or 3). Eigenvalues and eigenvectors can help us visualize the 2D or 3D slice that will be the most interesting to look at ~ the dimensions where the data has the most variance, or variability.

We don't have the probability or calculus to justify this yet, but if you have a matrix X of data points and want to find the best 2 or 3 dimensions to collapse things down to, then find the eigenvectors of $X^T X$. (This is the "covariance matrix," to come later in stats.) The eigenvector with the biggest eigenvalue will point in the direction of highest variance. The vector with the next biggest will point in the direction that handles the most variance after the first direction is taken into account. And so on — if you want the most important m dimensions out of n , this method will tell you what they are.



(2nd eigenvector of $X^T X$)
 \vec{v}_2
This method is called PCA
for "principal component analysis."
(k th eigenvector of $X^T X$)
 \vec{v}_k

General behavior of $A^n \vec{v}$ over time

vk 5/32 ext!

If the eigenvalues of A are unique, then we can begin to get a very good sense of how $A^n \vec{v}$ will behave over time - and we can calculate the values readily (for $A^n \vec{v}$).

If the eigenvalues are unique, then the corresponding eigenvectors must be linearly independent. (The proof by induction is not that interesting.) If we have N linearly independent vectors for an n -dimensional ~~space~~, they form a basis for that space.

If we represent a vector \vec{v} in that basis, the action of the matrix A on \vec{v} is just the sum of what it does to each component. And what it does to each eigenvector is scale it by the eigenvalue associated with that vector. So we have

$$\vec{v} = c_1 \vec{e}_1 + c_2 \vec{e}_2 + \dots + c_n \vec{e}_n$$

(linear
combo
of
eigenvectors)

$$A\vec{v} = \lambda_1 c_1 \vec{e}_1 + \lambda_2 c_2 \vec{e}_2 + \dots + \lambda_n c_n \vec{e}_n$$

$$A^2\vec{v} = \lambda_1^2 c_1 \vec{e}_1 + \lambda_2^2 c_2 \vec{e}_2 + \dots + \lambda_n^2 c_n \vec{e}_n$$

$$A^N\vec{v} = \lambda_1^N c_1 \vec{e}_1 + \lambda_2^N c_2 \vec{e}_2 + \dots + \lambda_n^N c_n \vec{e}_n$$

A few things to notice - one, this is an example of how the right basis can make exponentiation simple sometimes, (see "diagonalization")

But also, if we consider the relative influence of the biggest eigenvalue λ_1 , versus the second biggest λ_2 , the ~~biggest~~ term corresponding to λ_1 , is going to dominate if it's at all bigger, because the ratio $(\frac{\lambda_2}{\lambda_1})^n$ goes to 0 as N gets large. Vectors that contain any component in the direction of \vec{e}_1 will eventually have that vector magnified to be huge, relative to the other components. Of course, if that component is 0, then the second biggest λ 's eigenvector will have the most influence.

We can also see that if the ratio is a big difference, λ_1 will dominate faster; if it's small, it may take a while.

What will also happen in the long term is that, as $A^n \vec{v}$ becomes more and more like the first eigenvector, the degree of scaling that a multiplication by A performs will tend toward λ_1 .

Stable Populations Example (from book)

Suppose we have a ~~park~~ with a ^{wildlife preserve} park population p_t and world population w_t . We know that there's some running back and forth between park & world every year, so next year's populations are given by

$$p_{t+1} = 0.9p_t + 0.01w_t$$

$$w_{t+1} = 0.1p_t + 0.99w_t$$

This is a linear transformation. So we can analyze the eigenvalues & eigenvectors to find sustainable (1) rates of growth for both and ② what ratio will achieve that (park vs world pops).

This won't go that well though - Turns out the only stable states are no growth and a slight decline.

$$\begin{bmatrix} 0.9 & 0.01 \\ 0.1 & 0.99 \end{bmatrix} \Rightarrow \begin{bmatrix} 0.9-\lambda & 0.01 \\ 0.1 & 0.99-\lambda \end{bmatrix} \rightarrow \lambda^2 - 1.89\lambda + 0.89 = (\lambda-1)(\lambda-0.89)$$

Eigenvalues are $\lambda=1$ and $\lambda=0.89$. We probably care more about the stable ($\lambda=1$) ratio. We

can get it by solving

$$\begin{bmatrix} -0.1 & 0.01 & | & 0 \\ 0.1 & -0.01 & | & 0 \end{bmatrix}$$

A park population % to the world population is stable, eg $\begin{bmatrix} 1000 \\ 10000 \end{bmatrix} - 100 + 100 = 0$

~~Annealing~~

Since the other solution $\lambda=0.89$ gives opposite signs to the populations, we probably don't need to worry about it - the ratio of animals will converge naturally to the stable state.

Relationship Between Eigenvectors/Eigenvalues and Diagonalization

We never got around to how to diagonalize a matrix earlier. But the idea is how we broke down the matrix A 's action \vec{v} into its action on the eigenvectors. The diagonal matrix, if it's possible, is composed of eigenvalues of A along the diagonal. The matrix P in $A = PDP^{-1}$ consists of column vectors that are the eigenvectors ~~(with the eigenvalues)~~ in the corresponding columns of D . So to find the required P , we need N distinct linearly independent eigenvectors. If that's impossible, the matrix can't be diagonalized; if it's possible, then it can be.

For example:

$$\begin{bmatrix} 6 & -1 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 5 & 0 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}$$

$A \qquad P \qquad D \qquad P^{-1}$

You can check that $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ is an eigenvector of A with $\lambda = 5$ and $\begin{bmatrix} 1 \\ 2 \end{bmatrix}$ is an eigenvector with $\lambda = 4$.

One more application: Recurrences

Wk 5/32 ext 3

Suppose we have a running time recurrence of $T(N) = T(N-1) + T(N-2)$, $T(1) = 1$, $T(0) = 0$ which you can get with, e.g., a naive recursive calculation of the Fibonacci numbers. What's the growth rate ($\Theta(\dots)$)? We can make a ~~vector~~ that means $\begin{bmatrix} T_i \\ T_{i-1} \end{bmatrix}$ and start it out at $\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} T_1 \\ T_0 \end{bmatrix}$. The recurrence

can be phrased as a matrix $A = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$, which sets T_{i-1} to T_i and sums to get the next value. Multiplying by this matrix gives $\begin{bmatrix} T_2 \\ T_1 \end{bmatrix}$, then $\begin{bmatrix} T_3 \\ T_2 \end{bmatrix}$ etc.

The growth rate should be how much the dominant eigenvector scales the vector raised to the N :

λ^N Solving for eigenvalues of $\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$ we have

$$|1-\lambda| = 0 \quad \text{or} \quad \lambda^2 - \lambda - 1 = 0. \quad \text{The roots are}$$

$$\lambda = \frac{1 \pm \sqrt{5}}{2} \quad \text{and the bigger one,}$$

known as the "golden ratio"

So Fibonacci recurrences grow as $\Theta\left(\left(\frac{1+\sqrt{5}}{2}\right)^N\right)$.