

GILBERT

OK. So this video is about using eigenvectors and eigenvalues to take powers of a matrix, and I'll show you why we want to take powers of a matrix. And then the next video would be using eigenvalues and eigenvectors to solve differential equations. The two big applications.

STRANG:

So here's the first application. Let me remember the main facts. That if A -- if. This is an important point. Not every matrix has n independent eigenvectors that would go into matrix V . You remember V is the eigenvector matrix, and I need n independent eigenvectors in order to have a V inverse, to make that formula correct. So that's the key formula for using eigenvalues and eigenvectors.

And the case where we might run short of eigenvectors is when maybe one eigenvalue is repeated. It's a double eigenvalue, and maybe there's only one eigenvector to go with it. Every eigenvalue's got at least one line of eigenvectors. But we might not have two when the eigenvalue is repeated or we might. So there are cases when this formula doesn't apply. Because I must be able to take V inverse, I need n independent columns there. OK.

But when it works, it really works. So the n -th power, just remembering, is $V \lambda V^{-1}$, $V \lambda V^{-1}$, n times. But every time I have a V^{-1} and a V , that's the identity. So I move V out at the beginning. I have $\lambda, \lambda, \lambda, n$ of those, and a V^{-1} at the very end. So that's the nice result for the n -th power of a matrix.

Now I have to show you how to use that formula, how to use eigenvalues and eigenvectors.

OK. So we know we can take powers of a matrix. So first of all, what kind of equation? There's an equation. That's called a difference equation. It goes from step k to step $k+1$ to step $k+2$. It steps one at a time and every time multiplies by A . So after k steps, I've multiplied by A^k times from the original u_0 .

So instead of a differential equation, it's a step difference equation with u_0 given. And there's the solution. That's the quickest form of the solution. A to the k -th power, that's what we want. But just writing A to the k , if we had a big matrix, to take its hundredth power would be ridiculous. But with eigenvalues and eigenvectors, we have that formula.

OK. But now I want to think. Let me try to turn that formula into something that you just naturally see. And we know what happens. If u_0 is an eigenvector, if u_0 is an eigenvector, that probably won't happen because there are just n eigenvector directions. But if it happened to

be an eigenvector, then every step we'd multiply by λ , and we'd have the answer, λ^k times.

But what do we do for all the initial vectors u_0 which are maybe not an eigenvector? How do I proceed? How do I use eigenvectors when my original starting vector is not an eigenvector? And the answer is, it will be a combination of eigenvectors.

So making this formula real starts with this. So I write u_0 as a combination of the eigenvectors. And I can do it because if I have n independent eigenvectors, that will be a basis. Every vector can be written in the basis. So I'm looking there at a combination of eigenvectors.

And now the point is that as I take these steps to u_1 -- what will u_1 be? u_1 will be Au_0 . So I'm multiplying by A . So when I multiply this by A , what happens? That's the whole point. c_1 , A times x_1 is λ_1 times x_1 . It's an eigenvector. c_2 tells me how much of the second eigenvector I have. When I multiply by A , that multiplies by λ_2 , and so on, $c_n \lambda_n^n$. And that's the thing. Each eigenvector goes its own way, and I just add them together.

OK. And what about A to the k -th power? Now, that will give me u_k . And what happens if I do this k times? You've seen what I got after doing it one time. If I do it k times, that λ_1 that multiplies its eigenvector will happen k times.

So I'll have λ_1 to the k -th power. Do you see that? Every step brings another factor λ_1 . Every step brings another factor λ_2 . Every step brings-- that's the answer. That is-- well, that answer must be the same as this answer. And I'll do an example in a minute.

Right now, I'm just getting the formulas straight. So I have the quickest possible formula, but it doesn't help me much. I have the using the eigenvectors and eigenvalue formula. And here I have it that, really, it's the same thing written out as a combination of eigenvectors. And then this is my answer. That's my answer to the-- that's my solution u_k . That's it. So that must be the same as that.

Do you want to just think for one minute why this answer is the same as that answer? Well, we need to know what are the c 's? Well, the c 's came from u_0 . And if I write that equation for the c 's-- do you see what I have as an equation for the c 's? u_0 is this combination of eigenvectors.

That's a matrix multiplication. That's the eigenvector matrix multiplied by the vector c of

coefficients, right? That's how a matrix multiplies a vector. The columns, which are the x 's, multiply the numbers c_1, c_2, \dots, c_n . There it is. That's the same as that. So u_0 is Vc . So c is $V^{-1}u_0$. Oh, that's nice. That's telling us what are the coefficients, what are the numbers, what amount of each eigenvector is present in u_0 . This is the equation.

But look, you see there that $V^{-1}u_0$, that's the first part there of the formula. I'm trying to match this formula with that one. And I'm taking one step to recognize that this part of the formula is exactly c . You might want to think about that. Run this video once more just to see that step.

Now what do we do? We've got the lambdas. So I'm taking care of the c 's, you could say. Now I need the lambda to the k -th power-- lambda 1 to the k -th, lambda 2 to the k -th, lambda n to the k -th. That's exactly what goes in here. So that factor is producing the lambdas to the k -th power.

And finally, this factor has-- everybody's remembering here. V is the eigenvector matrix x_1, x_2, \dots, x_n . And when I multiply by V , it's a matrix times a vector. This is a matrix. This is a vector. And I get the combination-- I'm adding up. I'm reconstructing the solution.

So first I break up u_0 into the x 's. I multiply them by the lambdas, and then I put them all together. I reconstruct u_k . I hope you like that. This formula, which it's like common sense formula, is exactly what that algebra formula, matrix formula, says.

OK. I have to do an example. Let me finish with an example. OK. Here's a matrix example. A equals-- this'll be a special matrix. I'm going to make the first column add up to 1, and I'm going to make the second column add up to 1. And I'm using positive numbers. They're adding to 1. And that's called a Markov matrix. So it's nice to know that name-- Markov matrix.

One of the beauties of linear algebra is the variety of matrices-- orthogonal matrices, symmetric matrices. We'll see more and more kinds of matrices. And sometimes they're named after somebody who understood that they were important and found their special properties. So a Markov matrix is a matrix with the columns adding up to 1 and no negative numbers involved, no negative numbers. OK. That's just by the way.

But it tells us something about the eigenvalues here. Well, we could find those two eigenvalues. We could do the determinant. You remember how to find eigenvalues. The determinant of $\lambda I - A$ will be something. Could easily figure it out. There's always a

lambda squared, because it's two by two, minus the trace. $0.8 + 0.7 = 1.5$ lambda, plus the determinant. $0.56 - 0.06 = 0.50$, 0.5 . And you set that to 0.

And you get a result that one of the eigenvalues is-- this factors into lambda minus 1, lambda minus $1/2$. And the cool fact about Markov matrices is lambda equal 1 is always an eigenvalue. So lambda equal 1 is an eigenvalue. Let's call that lambda 1. And lambda 2 is an eigenvalue, and that depends on the numbers, and it's $1/2$, 0.5 , 0.5 . Those are the eigenvalues. $1 + 1/2 = 1.5$. The trace is $0.8 + 0.7 = 1.5$. Are we good for those two eigenvalues? Yes.

And then we find the eigenvectors that go with them. I think that this eigenvector turns out to be $0.6, 0.4$. I could check. If I multiply, I get $0.48 + 0.12 = 0.60$, and that's the same as 0.6 . And that goes with eigenvalue 1. And I think that this eigenvector is $1, -1$. Maybe that's always for a two-by-two Markov matrix. Maybe that's always the second eigenvector. I think that's probably good. Right. OK. Yeah.

All right. What now? What now? I want to use the eigenvalues and eigenvectors, and I'm going to write out now u_k . So if I apply A^k times to u_0 , I get u_k . And that's $c_1 \lambda_1^k$ times its eigenvector $0.6, 0.4$ plus $c_2 \lambda_2^k$ times its eigenvector, the second eigenvector, $1, -1$.

That is a formula. $c_1 \lambda_1^k x_1 + c_2 \lambda_2^k x_2$. And c_1 and c_2 would be determined by u_0 , which I haven't picked a u_0 . I could. But I can make the point, because the point I want to make is true for every u_0 , every example. And here's the point. What happens as k gets large? What happens if Markov multiplies his matrix over and over again, which is what happens in a Markov process, a Markov process?

This is like-- actually, the whole Google algorithm for page rank is based on a Markov matrix. So that's like a multi-billion-dollar company that's based on the properties of a Markov matrix. And you repeat it and repeat it. That just means that Google is looping through the web, and if it sees a website more often, the ranking goes up.

And if it never sees my website, then for that, when it was googling some special subject, it never came to your website and mine, we didn't get ranked. OK. So this goes to 0. $1/2^k$ to the-- it goes fast to 0, quickly to 0. So that goes to 0. And of course, that stays exactly where it is. So there's a steady state.

What happens if page rank had only two websites to rank, if Google was just ranking two websites? Then its initial ranking, they don't know what it is. But by repeating the Markov matrix and this part going to 0, right, goes to 0 because of $1/2$ to the k -th power, there is the ranking, 0.6, 0.4. That's where Google-- so this first website would be ranked above the second one.

OK. There's an example of a process that's repeated and repeated, and so a Markov matrix comes in. This business of adding up to 1 means that nothing is lost. Nothing is created. You're just moving. At every step, you take a Markov step. And the question is, where do you end up? Well, you keep moving, but this vector tells you how much of the time you're spending in the two possible locations. And this one goes to 0.

OK. Powers of a matrix, powers of a Markov matrix. Thank you.