The following content is provided under a Creative Commons license. Your support will help MIT OpenCourseWare continue to offer high quality educational resources for free. To make a donation or view additional materials from hundreds of MIT courses, visit MIT OpenCourseWare at ocw.mit.edu.

PROFESSOR: So let's begin. Today, I'm going to review of linear algebra. So I'm assuming that you already took some linear algebra course. And I'm going to just review the relevant content that will appear again and again throughout the course. But do interrupt me if some concepts are not clear, if you don't remember some concept from linear algebra.

I hope you do. But please let me know. I just don't know. You have very different background knowledge. So it's hard to tune to one special group. So I tailored this lecture notes so that it's a review for those who took the most basic linear algebra course. So if you already have that experience, and don't understand it, please feel free to interrupt me.

So I'm going to start by talking about matrices. A matrix, in a very simple form, is just a collection of numbers. For example 1, 2, 3; 2, 3, 4; 4, 5, 10. You can pick any number of rows, any number of columns. You just write down numbers in a square format. And that's the matrix. What's special about it?

So what kind of data can you arrange in a matrix? So I'll take an example, which looks relevant to us. So for example, we can index the rows by stocks, by companies, like Apple. Morgan Stanley should be there, and then Google. And then maybe we can index the column by dates.

I'll say July 1st, October 1st, September 1st. And the numbers, you can pick whatever data you want. But probably the sensible data will be the stock price on that day. I don't know for example 400, 500, and 5,000. That would be great. So these kind of data, that's just the matrix.

So defining a matrix is really simple. But why is it so powerful? So that's an

application point of view, just as a collection of data. But from a theoretical point of view, a matrix, an m by n matrix, is an operator. It defines a linear transformation. A, defines a linear transformation from the vector space, n dimensional vector space to the m dimensional director space. That's sounds a lot more abstract than this.

So for example, let's just take a very small example. If I use a 2 by 2 matrix, 2, 0, 0, 3. Then 2, 0, 0, 3 times let's say 1, 1 is just 2, 3. Does that makes sense? It's just matrix multiplication.

So now try to combine the point of view. What does it mean to have a linear transformation defined by a data set? And things start to get confusing. What is it? Why does a data set define a linear transformation? And does it have any sensible meaning?

So that's a good question to have in mind today. And try to remember this question. Because today I'll try to really develop a theory of eigenvalues and eigenvectors in a purely theoretical language. But it can still be applied to these data sets, and give very important properties and very important quantities. You can get some useful information out of it. Try to make sense out of why it happens. So that will be the goal today, to really treat linear algebra as a theoretical thing. But remember that there's some data sets, like really data set underlying.

This doesn't go up. That was a bad choice for my first board. Sorry.

So the most important concepts for us are the eigenvalues and eigenvectors of a matrix, which is defined as a real number, lambda, and vector V, is an eigenvalue, and eigenvector of a matrix A, if A times V equals lambda times V. We also say that V is an eigenvector corresponding to lambda.

So remember eigenvalues and eigenvectors always come in pairs. And they are defined by the property that A times V is equal to lambda time V. First question, does all matrix have eigenvalues and eigenvectors? Nope? So AV, it looks like this is a very strange equation to satisfy. But if you change it in this form, A minus lambda i equals zero. That still looks strange.

But at least you understand that-- it's an only if, this can happen only if this can happen. Happens only if A minus lambda i does not have full rank. So determinant of A minus lambda i is equal to 0, if and only if, in fact.

So now comes a very interesting observation. Determinant of A minus lambda i is a polynomial of degree n. I made a mistake. I should have said, this is only for n by n matrices. This is only for square matrices. Sorry.

It's a polynomial of degree n. That means it has a solution. It has to give n in terms of lambda. So it has a solution. It might be a complex number. I'm really sorry. I'm nervous in front of the video. I understand why you were saying that is doesn't necessarily exist.

Let me repeat. I made a few mistakes here. So let me repeat here. For n by n matrix A, a complex number lambda, and the vector V is an eigenvalue and eigenvector if it satisfies this condition. It doesn't have to be real. Sorry about that. And now if we rephrase it this way, because this is a polynomial, it always has at least one solution. That was just a side point. Very theoretical.

So we see that there always exists at least one eigenvalue and eigenvector. Now we saw its existence, what is the geometrical meaning of it?

Now let's go back to the linear transformation point of view. So suppose A is a 3 by 3 matrix. Then A takes the vector in R3 and transforms it into another vector in R3. But if you have this relation, what's going to happen is A, when applied to V, it will just scale the vector V. If this was the original V, A of V will just be lambda times this vector. That will be our A V, which is equal to lambda V.

So eigenvectors are those special vectors which when applied this linear transformation just gets scaled by some amount, which whereas that amount is exactly lambda. So what we established so far, what we recall so far is every n by n matrix has at least one such direction. There is some vector where the linear transformation defined by A just scales that vector.

Which is quite interesting, if you ever thought about it before. There's no reason such vector should exist. Of course I'm lying a little bit. Because these might be complex vectors. But at least in the complex world it's true.

So if you think about this, this is very helpful. It gives you the vectors from these vectors' point of view, this linear transformation is really easy to understand. That's why eigenvalues and eigenvector are so good. It breaks down the linear transformation into really simple operations.

Let me formalize that a little bit more. So in an extreme case a matrix, an n by n matrix A, we call it diagonalizable, if there exists an orthonormal matrix, I'll call what it is, U, such that A is equal to U times D times U inverse for a diagonal matrix D.

Let me iterate through this a little bit. What is an orthonormal matrix? It's a matrix defined by the relation U times U transposed is equal to the identity. What is a diagonal matrix? It's a matrix whose nonzero entries are all on the diagonal. All the rest are zero.

Why is it so good to have this decomposition? What does it mean to have an orthonormal matrix like this? It means basically I'll just explain what's happening. If that happens, if a matrix is diagonalizable, if this A is diagonalizable, there will be three directions, V1, V2, V3, such that when you apply this A, V1 scales by sum lambda 1. V2 scales by sum lambda 2. And V3 scales by sum lambda 3. So we can completely understand the transformation A, just in terms of these three vectors.

So this, the stuff here will be the most important things you'll use in linear algebra throughout this course. So let me repeat it really slowly. So an eigenvalue and eigenvector is defined by this relation. We know that there are at least one eigenvalue for each matrix, and there is an eigenvector corresponding to it. And eigenvectors have this geometrical meaning where vectors and eigenvector, if the linear transformation defined by A just scales that vector.

So for our setting, the real good matrices are the matrices which can be broken down into these directions. And those directions are defined by this U. And D

defines how much it will scale. So in this case U will be our V1, V2, V3. And D will be our lambda 1, lambda 2, lambda 3, all 0.

Any questions so far? So that is abstract. Now remember the question I posed in the beginning. So remember that matrix where we had stocks and dates and stock prices in the entries? What will an eigenvector of that matrix mean? What will an eigenvalue mean? So try to think about that question. It's not like it will have some physical counterpart. But there's some really interesting things going on there.

The bad news is that not all matrices are diagonalizable. If a matrix is diagonalizable, it's really easy to understand what it does. Because it really breaks down into these three directions, if it's a 3 by 3. If it's an n by n, it breaks down into n directions.

Unfortunately, not all matrices are diagonalizable. But there is a very special class of matrices which are always diagonalizable. And fortunately we will see those matrices throughout the course. Most of the matrices, n by n matrices, we will study, fall into this category.

So an n by n matrix A, is symmetric if A is equal to A transposed. Before proceeding, please raise your hand if you're familiar with all the concepts so far. OK. Good feeling. So a matrix is symmetric if it's equal to its transpose. A transpose is obtained by taking the mirror image across the diagonal. And then it is known that all symmetric matrices are diagonalizable. Ah, I've made another mistake. Orthonormally.

So with this I missed matrices orthonormally diagonalizable. So it's diagonalizable if we drop this condition, and replace it with an invertible. So symmetric matrices are really good. And fortunately most of the n by n matrices that we will study are symmetric. Just by the nature of it, it will be symmetric.

The one I gave as an example is not symmetric. It's not symmetric. But I will address that issue in a minute. And another important thing is symmetric matrices have real eigenvalues. So really this geometrical picture just the-- for symmetric

matrices, this picture is really the picture you should have in mind.

So proof of theorem 2. Suppose lambda is an eigenvalue with eigenvector V. Then by definition we have this. Now multiply V transposed on both sides. We'll work it as lambda times the norm V.

Now take the complex conjugate of real symmetric. And then first A conjugate, we have VTATV, and then take the conjugate of it. Then we get lambda V. And this side is equal to VTATV. But because A is real symmetric, we see that A is equal to the conjugate, complex conjugate of A.

So this expression and this expression is the same. The right side should also be the same. That means lambda is equal to the conjugate of lambda. So lambda has to be a real.

So theorem 1 is a little bit more complicated, and it involves more advanced concepts like basis and linear subspace, and so on. And those concepts are not really important for this class. So I'll just skip the proof. But it's really important to remember these two theorems.

Wherever you see a symmetric matrix you should really feel like you have control on it. Because you can diagonalize it. And moreover, all eigenvalues are real, and you have really good control on symmetric matrices. That's good.

That was when everything went well. We can diagonalize it. So, so far we saw that if for a symmetric matrix, we can diagonalize it. It's really easy to understand. But what about general matrices?

In general, not all matrices are diagonalizable, first of all. But sometimes we still want to decomposition like this. So diagonalization was A equals U times D times U inversed.

But we want something similar. We want to understand. So our goal, we want to still understand the matrix, give a matrix A through simple operations, such as scaling. When the matrix was a diagonalizable matrix this was done, this was possible.

Unfortunately, it's not always diagonalizable. So we have to do something else.

So that's what I want to talk about. And luckily the good news is there is a nice tool we can use for all matrices, even those slightly weaker, in fact, a little bit more weaker than this diagonalization. But still it distills some very important information of the matrix. So it's called singular value decomposition.

So this will be our second tool of understanding matrices. It's very similar to this diagonalization, or in other words I call this eigenvalue decomposition. But it has a slightly different form. So what is its form? So a theorem. Let A be an m by n matrix. Then there always exists orthonormal matrices U and V such that A is equal to U times sigma times V transposed. For some diagonal matrix sigma.

Let me parse through the theorem a little bit more. Whenever you're given a matrix, it doesn't even have to be a square matrix anymore. It can be non-symmetric. So whenever we're given an m by n matrix, in general, there always exists two matrices, U and V, which are orthonormal, such that A can be decomposed as U times sigma times V transposed, where sigma is a diagonal matrix. But now the size of the matrix are important so U is an m by n matrix, sigma is an m by n matrix, and V is an n by n matrix. That just denotes the size, the dimensions of the matrix.

So what does it mean for an m by n matrix to be diagonal? It just means the same thing. So only the ii entries are allowed to be nonzero.

So that was just a bunch of words. So let me rephrase this. So let me compare now eigenvalue decomposition, with singular value decomposition. So this is EVD, what we just saw before. And only SVD. This only works for n by n matrices, which are diagonalizable. SVD works for all general m by n matrices.

However, this is powerful. Because it gives you one frame. So V1, with a V2, V3, for which A acts as a scaling operator. Kind of like that. That's what A does, A does, A does.

That's because these U on the both sides are equal. However, for singular value decomposition, this is called singular value decomposition. I just erased It. What you

have instead is first of all, the spaces are different. If you take a vector in R to the m, and bring it to R to the n, apply this operation A. What's going to happen here is there will be one frame in here, and one frame in here. So there will be vectors V1, V2, V3, V4 like this. And there will be vectors U1, U2, U3 like this here.

And what's going to happen is when you take V1, A will take V1 to U1 and scale it a little bit according to that diagonal. A will take V2 to U2, it will scale it. It'll take V3 to U3, scale it. Wait a minute. But for V4, we don't have U4. What's going to happen is this is just going to disappear. U4 when applied A, will disappear.

So I know it's a very vague explanation, but this geometric picture try to compare them. a diagonalization eigenvalue decomposition works within its frame, so it's very, very powerful. You just have some directions and you scale those directions. But the singular value composition it's applicable to a more general class of matrices, but it's rather more restricted.

You have two frames, one for the original space, one for the target space. And what the linear transformation does is, is just sends one vector to another vector and scales it a little bit.

So now is another good time to go back to that matrix in the very beginning. So remember that example where we had a vector of companies, and dates, and the entry was stock prices.

So if it's an n by n matrix, you can try to apply both eigenvalue decomposition, and singular value decomposition. But what will be more sensible is singular value decomposition in this case. I won't explain why, and what's happening here. Peter will probably. You will come to it later. But just try to do some imagining before listening what's really happening in real world. So try to use your own imagination, your own language to express. See what happens for this matrix, what this decomposition is doing.

It just looks like totally nonsense. Why does this have even a geometry? Why does it define a linear transformation and so on? It's just a beautiful theory, which just gives

many useful information. I can't really emphasize more. Because emphasize enough because really this is just universal, being used in all science these. I think the eigenvalue decomposition, and the singular value decomposition. Not just for this course, but pretty much it's safe to say in every engineering, you'll encounter one of the forms.

So let me talk about the proof of the singular value decomposition. And I will show you an example of what singular value decomposition does for some example matrix, the matrix that I chose. Proof of singular value decomposition, which is interesting. It relies on eigenvalue decomposition.

So given a matrix A, consider the eigenvalues of A times A transposed. Oh, A transpose A. First observation, that's a symmetric matrix.

So if you remember, it will have real eigenvalues, and it's diagonalizable. So A T of A has eigenvalues lambda 1, lambda 2, up to it's an n by n matrix, so lambda n. And corresponding eigenvectors V1, V2, up to Vn.

And so for convenience, I will cut it at lambda r, and assume all rest is 0. So there might be none which are 0. In that case we use all the eigenvalues. But I only am interested in nonzero eigenvalues. So I'll say up to lambda r, they're nonzero. Afterwards it's 0. It's just a notational choice.

And now I'm just going to make a claim that they're all positive. This part is kind of just believe me. Then if that's the case, we can rewrite the eigenvalues. Rewrite eigenvalues as sigma 1 square, sigma 2 square, sigma r square, and 0.

That was my first step. My second step, that was step one, step 2 is to define U of 1 as A times V1 over sigma 1, U of 2 as A times V2 over sigma 2. And U of r, as A times V of r over sigma r. And then U times r plus 1 as up to U times m as complete the above into a basis.

So for those who don't understand, just think of we pick U1 up to Ur first, and then arbitrarily pick the rest. And you'll see why I only care about the nonzero eigenvalues. Because I have to divide by sigmas, the sigma values. And if it's zero, I

can't do the division. So that's why I identified those which are not zero. And then we're done.

So it doesn't look at all like we're done. But I'm going to let my U be this, U1, U2, up to Un. Sorry, it has to be n. My V I will pick as V1, V2, up to Vr. And then the r plus one up to Vn. So this again just complete into a basis. Now let's see what happens.

So A times U transposed times V. Oh, ah. That's why it's a problem. You have to do U times A times V transposed. So I would write V is n, and this n.

Ah yes, so U times A times V transposed here. That will be U1, U2, Um A. V transposed will be V1 transposed, V2 transposed, to Vn transposed. I messed up something. Sorry.

Oh, that's the form I want, right? Yeah. So I have to transpose U and V there. OK, sorry. Thank you. Thank you for the correction. I know this looks different from that. But I mean if you flip the definition it will be the same. So I'll just not-- stop making mistakes. Do you have a question? So, yeah. Thank you. Yeah. That will make more sense. Thank you very much.

And then you're going to have U1 transposed up to Un transposed. A times V, because of the definition of V, will be lambda 1 of V1. A times V2 will be lambda 2 of V2. Up to lambda r of Vr, and the rest will be zero. These all define the columns.

Now let's do a few computations. So U1T times lambda 1 V1. U1 T, and lambda 1 V1. When you take the dot product, what you're going to get is V1TA transposed of V1 lambda 1. I'm missing something. Ah, sorry about that. This is not right. These are As. I defined the eigenvalues and I went to the A transpose A.

Then that's U1 transposed times sigma 1 times U1. That will be sigma 1. And then if you look at the second entry, U1 transposed times AV2, you get U1 transposed times sigma 2 of U2. But I claim that this is equal to 0. So why is that the case?

U1 transposed is equal to V1 transposed A transposed over sigma 1. And we have sigma 2. U2 is equal to A times V2 over sigma 2. So those two cancel. And we have

V1TATAV2 over sigma 1. But V1 and V2 are two different eigenvectors of this matrix.

At the beginning we can have an orthonormal decomposition of A transposed A. That means V1T times V2 times that has to be equal to zero. Because that's an eigenvalue. We have V1T times lambda 2 V2 over sigma 1. So we have lambda 2 over sigma 1 times V1 transposed V2. These two are solved and also get 0.

So if you do the computation, what you're going to have is sigma 1, sigma 2 on the diagonal, up to sigma r, and then 0, 0 rest. And 0 the rest. Sorry for the confusion.

Actually the process is quite simple. I was just lost in the computation in the middle. So process is first look at A transpose A. Find the eigenvalues and eigenvectors. And using those, they define the matrix V. And you can define the matrix U by applying A times V over sigma. Each of those will define the entries of U.

The reason I wanted to go through this proof is because this gives you a process of finding a singular value decomposition. It was a little bit painful for me. But if you have a matrix there's just these simple steps you can follow to find the singular value decomposition. So look at this matrix, find its eigenvalues and eigenvectors. Just arrange it in the right way. Of course, the right way needs some practice to be done correctly. But once you do that, you just obtain a singular value composition.

And really I can't explain how powerful it is. You will only later see it in the course how powerful this decomposition will be. And only then you'll more appreciate how good it is to have this decomposition, and be able to compute it so simply.

So let's try to do it by hand. Yes?

STUDENT: So when you compute the [INAUDIBLE].

PROFESSOR: Yes.

STUDENT: [INAUDIBLE]

PROFESSOR: It would have to be orthonormal, yeah. It's should be orthonormal. These should be

orthonormal. These also. And that's a good point, because that can be annoying when you want to do it by hand. Actually this decomposition. You have to do some Gram-Schmidt process or something like that.

What I mean by hand I don't really mean by hand, other than when you're doing homework. Because you can use the computer to do it. And in fact, if you use computer there are much better algorithms than this that are known, which can do this a lot more quickly and more efficiently.

So let's try to do it by hand. So let A be this matrix 3, 2 2; 2, 3, negative 2. And we want to make the eigenvalue decomposition of this. A transpose A, we have to compute that, is 3, 2, 2; 2, 3, negative 2. And you will get 13, 12, 2; 12, 13, negative 2; 2, negative 2, 8.

And let me just say that the eigenvalues are 0, 9, and 25. So in this algorithm, sigma 1 squared will be 25. sigma 2 squared will be 9. And sigma 3 squared will be 0. So we can take sigma 1 to be 5, sigma 2 to be 3, sigma 3 to be 0.

Now we have to find the corresponding eigenvectors, eigenvectors to find the singular value decomposition. And I'll just do one just to remind you how to find an eigenvector. So A transpose A, minus 25i is equal to, if you subtract 25 from these entries, you're going to get negative 12, 12, 2; 12, negative 12, negative 2; 2, negative 2, negative 13.

And then you have to find the vector which annilihates this matrix. And that will be, I can take one of those vectors to be just 1 over square root of 2, 1 over square root of two, 0, after normalizing. And then just do it for other vectors.

You find V2 to be 1 over square root 18, negative 1 over square root 18, 4 over square root 18.

Now then find V3 to be the one that annilihates this. But I'll just say it's x, y, z. This will not be important. I'll explain why it's not that important.

Then our V as written above, actually there it was transposed. So I will transpose it.

That will be 1 over square root of 2, 1 over square root of 2, 0. V2 is that. So we can write 1 over square root 18, negative 1 over square root 18, 4 over square root 18. And here just write x, y, z.

And U will be defined as U1 and U2, where U1 is A times V1 over sigma 1. U2 is A times V2 over sigma 2. So multiply A by this vector, divide by sigma 1 to get U. I already did the computation for you. It's going to be-- and this is going to be-- yes?

STUDENT: How did you get V1?

PROFESSOR: V1? So if you did the computation right in the beginning to get the eigenvalues, then ATA minus 25i, this has to be-- has to not have full rank. So there has to be a vector V, which when multiplied by this gives 0, 0, 0 vector. And then you say a, b, c and set it equal to 0, 0, 0. And just solve the system of linear equations. There will be several of them. For example, we can take 1, 1, 0 as well. But I just normalized it to have [INAUDIBLE].

So there's a lot of work involved if you want to do it by hand, even though you can do it. You have to find eigenvalues, find eigenvectors. In this case, you have to find three of them. And then you have to do more work, and more work. But it can be done. And we are done now.

So now this decomposes A into U sigma V transformation. So U is given as 1 over square root 2, 1 over square root 2; 1 over square root 2, minus 1 over square root 2. Sigma was 5, 3, 0. And V is this. So V transposed is just transposed of that. I'll just write it like that, where V is that.

So we have this decomposition. And so let me actually write it, because I want to show you why x, y, z is not important. 1 over square root 2, 1 over square root 2, 0. 1 over square root 18, minus 1 over square root 18, 4 over square root 18, x, y, z.

The reason I'm saying this is not important is because I can just drop-- oh what did I do here? It has to be 2 by 3. I can just drop this column, and drop this column together. It has to be that form. Drop this and drop this altogether.

So the message here is that the eigenvectors corresponding to eigenvalue zero are not important. The only relevant ones are nonzero eigenvalues. So drop this, and drop this. That will save you some computation.

So let me state a different form of singular value decomposition. So this works in general. There's a corollary. We get a simplified form of SVD. Where A becomes equal to U times sigma times V transposed. And A was an m by n matrix. U is still an m by n matrix. But now sigma is also m by n matrix. This only works when m is less or equal to n. And V is an m by n matrix.

So the proof is exactly the same. And the last step is just to drop the irrelevant information. So I will not write down why it works. But you can see if you go through it, you'll see that dropping this part just corresponds to exactly that information. So that's the reduced form.

So let's see. In the beginning we had A. I erased A. A was the 2 by 3 matrix in the beginning. And we obtained the decomposition into 2 by 2, 2 by 2, and 2 by 3 matrix. If we didn't delete the fifth column and fifth row, we would have obtained a 2 by 2, times 2 by 3, times 3 by 3 matrix. But now we can simplify it by removing those.

And it might not look that much different on this board. Because I just erased one row. But many matrices that you'll see in real application have a lot lower rank than the number of columns and rows. So if r is a lot more smaller than both m and n, then this part really-- it's not obvious here. But if m and n has a big gap here, really the number of columns that you're saving, it can be enormous.

So to illustrate an example, look at this. Now look at the stock prices, where you have companies and dates. Previously I just gave an example of a 3 by 3 matrix. But it's more sensible to have dates, a lot more dates than companies. So let's say you recorded 365 days of a year, even though the market is not open all days, and just like five companies.

If you did a decomposition this this, you'll have a 5 by 5, 5 by 365, 365 by 365 here.

But now in the reduced form, you're saving a lot of space. So if you just look at the board, it doesn't look like it's so powerful. But in fact it is. So that's the reduced form. And that will be the form that you'll see most of the time, this reduced form.

So I made lot of mistakes today. I have one more topic, but a totally relevant topic. So any questions before I move on to the next topic? Yes?

STUDENT: [INAUDIBLE]

PROFESSOR: Can you press the button?

STUDENT: [INAUDIBLE]

PROFESSOR: Oh, so in this data what it means. You're asking what the eigenvectors will mean over this data? It will give you some stocks. It will give you like the correlation. So each eigenvector will give you a group of companies that are correlated somehow. It measures their correlation with each other. So I don't have a very good explanation what its physical meaning is. Maybe you can give just a little bit more.

GUEST SPEAKER: Possibly. We will get into this in later lectures. But in the singular value decomposition, what you want to think of is these orthonormal matrices are really defining a new basis, sort of an orthogonal basis. So you're taking the original coordinate system, then you're rotating it. And without changing or stretching or squeezing the data. You're just rotating the axes. So an orthonormal matrix gives you the cosines of the new coordinate system with respect to the old one.

And so the singular value decomposition then is simply sort of rotating the data into a different orientation. And the orthonormal basis that you're transforming to, is essentially the coordinates of the original data in the transformed system. So as Choongbum was commenting, you're essentially looking at a representation of the original data points in a linearly transformed space, and the correlations between different stocks say is represented by how those points are oriented in the new, in the transformed space.

PROFESSOR: So you'll have to see real data to really make sense out of it. But another way to

think of it is where it comes from. So all this singular value decomposition, if you remember the proof, it comes from eigenvectors and eigenvalues of A transposed A.

Now if you look at A transposed A, or I'll just say it's A time A transposed. It's pretty much the same. If you look at A times A transposed, you're going to get an m by n matrix. And it'll be indexed both by these companies. And the numbers here will represent how much the companies are related to each other, how much correlation they have between each other.

So by looking at the eigenvectors of this matrix, you're looking at the correlation between these stock prices, let's say, these company stock prices. And that information is represented inside the singular value decomposition. But again, it's a lot better to understand if you have real numbers and real data, which you will have later. So please be excited and wait. You're going to see some cool stuff.

So that was all for eigenvalue decomposition and singular value decomposition. And the last thing I want to mention today is something called Perron-Frobenius theorem. This one even looks a lot more theoretical than the ones I showed you. But surprisingly a few years ago, Steve Ross, he's a faculty in the business school here, found a very interesting result called Steve Ross recovery theorem that makes use of this theorem, makes use of Perron-Frobenius theorem that I will tell you today.

Unfortunately you will only see a lecture on Steve Ross recovery theorem towards the end of the semester. So I will try to recall what it is later. But since we're talking about linear algebra today, let me introduce the theorem. This is called Perron-Frobenius. And you really won't believe that it has any applications in finance because it just looks so theoretical.

I'm just stating a really weak form. Weak form. Let A be an n by n symmetric matrix, whose entries are all positive, with positive entries. Then there are a few properties that they have.

First there exists an eigenvalue, there exists a largest eigenvalue, lambda 0, such that lambda is less than lambda 0. Well that's true for all other lambda. So this statement is really easy for symmetric matrix. So forget about-- you can drop symmetric, but I'm just stated it, because I'm going to prove only for this weak case.

Just think about the statement when it's not symmetric. So if you have an n by n matrix whose entries are all positive, then there exists an eigenvalue, lambda 0, a real eigenvalue such that the absolute value of all of other eigenvalues are strictly smaller than this eigenvalue.

So remember that if it's not a symmetric matrix, they can be complex values. This is saying that there's a unique eigenvalue which has largest absolute value, and moreover, it's a real number.

Second part, there exists an eigenvector, a positive eigenvector with positive entries, corresponding to lambda 0. So the eigenvector corresponding to this lambda 0 has positive entries.

And the third part is lambda 0 is an eigenvalue of multiplicity 1, for those who know what it is. So this really is a unique eigenvalue with a unique eigenvector, which has positive entries. And it's larger, really larger than other eigenvalues.

So from the mathematician point of view, this has many applications. It's probability theory. My main research area is combinatorics discrete mathematics. It's also used in there. So from the theoretical point of view, this has been used in many contexts. It's not a standard theorem taught in linear algebra. So I don't think probably most of you haven't seen it before. But it's a well known result, with many uses, theoretical uses. But you also see one use in later, as I mentioned, in finance, which is quite surprising.

So let me just give you some feeling of why it happens. I won't give you the full detail of the proof, but just a very brief description. Sketch when A is symmetric, just a simple case, A is symmetric. In this case, this statement, if you look at it.

First of all A has real eigenvalues. I'll say it's lambda 1, lambda 2, up to lambda n.

And at some point, I'll say up to lambda i, it's greater than zero as to where this small arrow is zero. There are some positive eigenvalues. There are some negative eigenvalues. So that's observation one.

Things are more easy to control, because they are all real. The first statement says that-- maybe I should have indexed it as lambda 0. I'll just call this lambda 0 instead. This lambda 0 is in fact larger in absolute value than lambda n. That's the content of the first bullet.

So if they all have all positive entries, then the positive, largest positive eigenvalue dominates the smallest negative eigenvalue, which yeah. So why is that the case?

First of all, to see that you have to go through different steps. So we go into observation to 2. Lambda 1, so look at lambda 1. Lambda 1 has an eigenvector with positive entries. Why is that the case? That's because if you look at A times V equals lambda times V. If V-- let me state it this way. Lambda 0 is the maximum of all lambda, Lambda 0. That's not entirely correct. Lambda 1. Sorry about that.

So If you look at this, if V has non-positive entries, if it has a negative entry, if V has a negative entry, then flip it. Flip the sign, and in this way obtain new vector V prime. Since A has positive entries, A has positive entries. What we conclude is that A times A prime will be larger than A times V. You have to look. Think about, because it has positive entries, if it had some negative part somewhere, the magnitude will decrease. So if you flip the sign it should increase the magnitude.

And this cannot happen. This shouldn't happen. This should not happen. That's where the positive entries part is used. If you have positive entries, then it should have, the eigenvector should have positive entries as well.

So I will not work through the details of the rest. I will post it on the lecture notes. But really this theorem, in fact, can be stated in a lot more generality than this. I'm stating only a very weak form. It doesn't have to have all positive entries. It has to only be something called irreducible, which is a concept from probability theory from Markov chains.

But here we will only use it in this setting. So I will review it later, before it's really being used. But just remember that how these positive entries kick into this kind of statement, where there is an eigenvalue, largest eigenvalue, why there has to be a vector which is all positive entries. Those will all come into play later. So I think that's it for today. If you have any last minute questions? If not, I will see you on Thursday.