

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 to view additional materials from hundreds of MIT courses, visit MIT OpenCourseWare at ocw.mit.edu.

GILBERT
STRANG:

OK. Now, clustering for graphs. So this is a topic-- this is one of the important things you can try to do with a graph. So you have a large graph. Let me kind of divide it into two clusters. So you've got a giant graph. And then the job is to make some sense out of it. And one possible step is to be able to subdivide it, if, as I see here, there's a cut between two reasonably equal parts of the graph-- reasonable-- reasonably same size.

And therefore, that graph could be studied in two pieces. So the question is, how do you find such a cut by a algorithm? What's an algorithm that would find that cut? So that's a problem. Let's say we're looking for two clusters. We could look for more clusters, but let's say we want to look for two clusters. So what are we trying to do? We're trying to minimize. So this is the problem, then. So we look for-- find positions x and y , let's say. Two which will be the centers, so to speak, of the-- and really, it's just these points that-- so the data is the points and the edges, as always-- the nodes and the edges.

So the problem is to find x and y so that-- to minimize. So it's a distance of points a_i from x -- maybe should emphasize we're in high dimensions-- plus the distance of other points. So the a_i will be these points-- these nodes. And the b_i will be these nodes, plus the sum of b_i minus y squared. And you understand the rule here-- that together the a 's union the b 's give me all nodes. And I guess to be complete, the a 's intersect the b 's is empty. Just what you expect. I'm dividing the a 's and the b 's into two groups.

And I'm picking an x and a y sort of at the center of those groups, so that is a minimum. So I want to minimize. And also, I probably want to impose some condition that the number of a 's is reasonably close to the number of b 's. In other words, I don't want just that to be the a , and all the rest to be the b 's. That would be not a satisfactory clustering. I'm looking for clusters that are good sized clusters. So minimize that. OK.

So there are a lot of different algorithms to do it. Some are more directly attacking this problem. Others use matrices that we associate with the graph. So let me tell you about two or

three of those algorithms. And if you've seen-- studied-- had a course in graph theory, this-- you may already have seen this problem. First question would be, suppose I decide these are the a's, and those are the b's-- or some other decision. Yeah, probably some other decision. I don't want to solve the problem before I even start. So some a's and some b's.

What would be the best choice of the x once you've decided on the a's? And what would be the best choice of the y once you've decided on the b's? So we can answer that question if we knew the two groups. We could see where they should be centered, with the first group centered at x, the second group centered at y, and what does centering mean? So let's just say-- so I think what I'm saying here is-- let me bring that down a little.

So given the a's-- the a's-- this is a_1 up to, say, a_k . What is the best x just to make that part right? And the answer is, to do you know, geometrically, what x should be here? X is the-- so if I have a bunch of points, and I'm looking for the middle of those points-- the point x-- a good point x to say, OK, that's the middle. It'll make the sum of the distances, I think, squared-- I hope I'm right about that-- a minimum. What is x? It is the--

AUDIENCE: [INAUDIBLE].

GILBERT Centroid. Centroid is the word. X is the centroid of the a's. And what is the centroid? Let's see.
STRANG: Oh, maybe I don't know if x and y were a good choice, but let me see what-- I guess, it's the average a. It's the sum of the a's-- of these a's. Those are vectors, of course, divided by the number of a's. I think. Actually, I was just quickly reviewing this morning, so I'm not totally on top of this centroid.

What I'm going to talk-- the algorithm that I'm going to talk about is the k-- well, the k-means, it's always called. And here it will be the-- k will be 2. I just have two-- partitioning into two sets, a's and b's, so I just-- k is just 2. OK. What's the algorithm? Well, if I've chosen a partition-- the a's and b's have separated them-- then that tells me what the x and the y should be. But, now what do I do next? So is this going to be a sort of an alternating partition?

Now I take those two centroids. So step one is for given a's and b's, find the centroids x and y. And that's elementary. Then the second step is, given the centroids, x and y-- given those positions-- given x and y-- they won't be centroids when you see what happened. Given x and y, redo-- form the best partition-- best clusters.

So step one, we had a guess at what the best clusters were. And we found they're centroids.

Now, we start with the centroids, and we form new clusters again. And if these clusters are the same as the ones we started with, then the algorithm is converged. But probably they won't be-- these clusters. So you'll have to tell me what I mean by the best clusters. If I've got the two points, x and y , I want the points-- I want to separate all the points that cluster around x to the ones that cluster around y .

And then, they're probably different from my original start. So now I've got new-- now I repeat step one. But let's just think, how do I form the best clusters? Well, I take a point and I have to decide, does it go with x , or does it go within the x cluster, or does it go in the cluster around y ? So how do I decide that? Just whichever one it's closer to. So each point goes with each node. You should-- I could say, each node goes with the closer of x and y .

So points that should have been-- that are closer to x -- now we're going to put them in the cluster around x . And does that solve the problem? No, because-- well, it might, but it might not. We'd have to come back to step one. We've now changed the clusters. They'll have different centroids. So we repeat step one-- find the centroids for the two new clusters. Then we come to step two.

Find the ones that should go with the two centroids, and back and forth. I don't know. I don't think there's a nice theory of convergence, or rate of convergence-- all the questions that this course is always asking. But it's a very popular algorithm, k -means. k would be to have k clusters. OK. So that's a-- I'm not going to discuss the-- I'd rather discuss some other ways to do this, to solve this problem. But that's one sort of hack that works quite well. OK.

So second approach is what is coming next. Second solution method-- and it's called the spectral clustering. That's the name of the method. And before I write down what you do, what does the word spectral mean? You see spectral graph theory, spectral clustering. And in other parts of mathematics, you see that-- you see spectral theorem. I gave you the most-- and I described it as the most important-- perhaps-- theorem in linear algebra-- at least one of the top three. So I'll write it over here, because it's not-- it doesn't-- this is-- it's the same word, spectral.

Well, let me ask that question again? What's that word spectral about? What does that mean? That means that if I have a matrix, and I want to talk about its spectrum, what is the spectrum of the matrix? It is the eigenvalues. So spectral theory, spectral clustering is using the eigenvalues of some matrix. That's what that spectral is telling me. Yeah. So the spectral

theorem, of course, is that for a symmetric matrix S , the eigenvalues are real, and the eigenvectors are orthogonal.

And don't forget what the real, full statement is here, because there could be repeated real eigenvalues. And what does the spectral theorem tell me for symmetric matrices, if $\lambda = 5$ is repeated four times-- if it's a four times solution of the equation that gives eigenvalues, then what's the conclusion? Then there are four independent, orthogonal eigenvectors to go with it. We can't say that about matrices-- about all matrices. But we can say it about all symmetric matrices.

And in fact, those eigenvectors are orthogonal. So we're even saying more. We can find four orthogonal eigenvectors that go with a multiplicity for eigenvalues. OK. That's spectral theorem. Spectral clustering starts with the graph Laplacian matrix. May I remember what that matrix is? Because that's the key connection of linear algebra to graph theory, is the properties of this graph, Laplacian matrix. OK. So let me say L , for Laplacian.

So that matrix-- one way to describe it is as $A^T A$, where A is the incidence matrix of the graph. Or another way we'll see is the D -- the degree matrix. That's diagonal. And I'll do an example just to remind you. Minus the-- well, I don't know what I'd call this one. Shall I call it B for the moment. And what matrix is B ? That's the adjacency matrix. Really, you should know these four matrices. They're the key four matrices associated with any graph.

The incidence matrix, that's m by n -- edges and nodes-- edges and nodes. So it's rectangular, but I'm forming $A^T A$ here. So I'm forming a symmetric, positive, semi-definite matrix. So this Laplacian is symmetric, positive, semi-definite. Yeah. Let me let me just recall what all these matrices are for a simple graph. OK. So I'll just draw a graph. All right. OK. So the incidence matrix-- there are 1, 2, 3, 4, 5 edges-- so five rows. There are four nodes-- 1, 2, 3, and 4. So four columns. And a typical row would be edge 1 going from node 1 to node 2, so it would have a minus 1 and a 1 there.

And let me take edge 2, going from 1 to node 3, so it would have a minus 1 and a 1 there, and so on. So that's the incidence matrix A . OK. What's the degree matrix? That's simple. The degree matrix-- well, $A^T A$. This is m by n . This is n by m . So it's n by n . OK. In this case, 4 by 4. So the degree matrix will be 4 by 4, n by n . And it will tell us the degree of that, which means-- which we just count the edges. So three edges going in, node 2, three edges going in, node 3 has just two edges. And node 4 has just two edges.

So that's the degree matrix. And then the adjacency matrix that I've called B is also 4 by 4. And what is it? It tells us which node is connected to which node. So I don't allow nodes-- edges that connect a node to itself, so 0's on the diagonal. How many-- so which nodes are connected to node 1? Well, all of 2 and 4 and 3 are connected to 1. So I have 1's there. Node 2-- all three nodes are connected to node 2. So I'll have-- the second column and row will have all three 1's.

How about node 3? OK. Only edges-- only two edges are connected. Only two nodes are connected to 3, 1 and 2, but not 4. So 1 and 2 I have, but not 4. OK. So that's the adjacency matrix. Is that right? Think so. This is the degree matrix. This is the incidence matrix. And that formula gives me the Laplacian. OK. Let's just write down the Laplacian. So if I use the degree minus B-- that's easy. The degrees were 3, 3, 2, and 2. Now I have these minuses. And those were 0. OK. So that's a positive, semi-definite matrix.

Is it a positive definite matrix? So let me ask, is it singular or is it not singular? Is there a vector in its null space, or is there not a vector in its null space? Can you solve Dx equals all 0's? And of course, you can. Everybody sees that vector of all 1's will be a solution to L -- sorry. I should be saying L here. Lx equals 0.

Lx equals 0 as for a whole line of one dimensional null space of L has dimension 1. It's got 1 basis vector, 1, 1, 1, 1. And that will always happen with the graph set up that I've created. OK. So that's a first fact, that this positive, semi-definite matrix, L , has λ_1 equals 0. And the eigenvector is constant-- C, C, C, C-- the one dimensional eigenspace. Or 1, 1, 1, 1 is the typical eigenvector. OK.

Now back to graph clustering. The idea of graph clustering is to look at the Fiedler eigenvector. This is called the x_2 -- is the next eigenvector-- is the eigenvector for the smallest positive eigenvalue for a λ_{\min} excluding 0-- so the smallest eigenvalue of L -- the smallest eigenvalue and its eigenvector-- this is called the Fiedler vector, named after the Czech mathematician. A great man in linear algebra, and he studied this factor-- this situation. So everybody who knows about the graph Laplacian is aware that its first eigenvalue is 0, and that the next eigenvalue is important. Yeah.

AUDIENCE: Is the graph Laplacian named the Laplacian because it has connections to--

GILBERT To Laplace's equation, yes. Yeah, that's a good question. So why the word-- the name,

STRANG:

Laplacian? So yeah, that's a good question. So the familiar thing-- so it connects to Laplace's finite difference equation, because we're talking about matrices here, and not derivatives-- not functions. So why the word Laplacian? Well, so if I have a regular-- if my graph is composed of-- so there is a graph with 25 nodes, and 4 times 5-- 20, about 40. This probably has about 40 edges and 25 nodes.

And of course, I can construct its-- graph all those four matrices for it-- its incidence matrix, its degree matrix. So the degree will be four at all these inside points. The degree will be three at these boundary points. The degree will be two at these corner points. But the-- what will the matrix L look like? So what is L ? And that will tell you why it has this name Laplacian. So the matrix L will have-- the degree 4 right will be on the diagonal. That's coming from D . The other-- the minus 1's that come from B , the adjacency matrix, will be associated with those nodes, and otherwise, all 0's.

So this is a typical row of L . This is typical row of L centered at that node. So maybe that's node number 5, 10, 13. That's 13 out of 25 that would show you this. And the-- sorry. Those are minus 1's. Minus 1's. So a 4 on the diagonal, and four minus 1's. That's the model problem for when the graph is a grid-- square grid. And do you associate that with Laplace's equation? So this is the reason that Laplace-- why Laplace gets in it. Because Laplace's equation-- the differential equation-- is second derivative with respect to x squared, and the second derivative with respect to y squared is 0.

And what we have here is $Lu = 0$. It's the discrete Laplacian, the vector Laplacian, the graph Laplacian-- where the second x derivative is replaced by $-1, 2, -1$. And the second y derivative is replaced by $-1, 2, -1$. Second differences in the x and the y directions. So that's-- yeah. So that's the explanation for Laplace. It's the discrete Laplace-- discrete, or the finite difference Laplace. OK.

Now to just finish, I have to tell you what the-- what clusters-- how do you decide the clusters from L ? How does L propose two clusters, the a 's and b 's? And here's the answer. They come from this eigenvector-- the Fiedler eigenvector. You look at that eigenvector. It's got some positive and some negative components. The components with positive numbers of this eigenvector-- so the positive components of x -- of-- this eigenvector.

And there are negative components of this eigenvector. And those are the two clusters. So it's-- the cluster is-- the two clusters are decided by the eigenvector-- by the signs-- plus or minus

signs of the components. The plus signs go in one and the minus signs go in another. And you have to experiment to see that that would succeed. I don't know what it would do on this, actually, because that's hardly split up into two. I suppose maybe the split is along a line like that or something, to get-- I don't know what clustering.

This is not a graph that is naturally clustered, but you could still do k-means on it. You could still do spectral clustering. And you would find this eigenvector. Now what's the point about this eigenvector? I'll finish in one moment. What do we know about that eigenvector as compared to that one? So here was an eigenvector all 1's. Let me just make it all 1's, 1, 1, 1, 1. In that picture, it's 25 1's. Here's the next eigenvector up. And what's the relation between those two eigenvectors of L ? They are--

AUDIENCE: Orthogonal.

GILBERT
STRANG: Orthogonal. These are eigenvectors of a symmetric matrix. So they're orthogonal. So that means-- to be orthogonal to this guy means that your components add to 0, right? A Vector. Is orthogonal to all 1's. That dot product is just, add up the components. So we have a bunch of positive components and a bunch of negative components. They have the same sum, because the dot product with that is 0.

And those two components-- those two sets of components are your-- to tell you the two clusters in spectral clustering. So it's a pretty nifty algorithm. It does ask you to compute an eigenvector. And that, of course, takes time. And then there's a third, more direct algorithm to do this optimization problem. Well, actually, there are many. This is an important problem, so there are many proposed algorithms. Good. OK. I'm closing up. Final question. Yeah?

AUDIENCE: Is it possible to do more than two clusters?

GILBERT
STRANG: Well, certainly for k-means. Now, if I had to do three clusters with Fiedler, I would look at the first three eigenvectors. And, well, the first one would be nothing. And I would look at the next two. And that would be pretty successful. If I wanted six clusters, it probably would fall off in the quality of the clustering. Yeah. But that certainly-- I would look at the lowest six eigenvectors, and get somewhere. Yeah. Right. So OK. So that's a topic-- an important topic-- a sort of standard topic in applied graph theory. OK.

So see you Wednesday. I'm hoping, on Wednesday-- so Professor Edelman has told me a new and optimal way to look at the problem of backpropagation. Do you remember

backpropagation? If you remember the lecture on it-- you don't want to remember the lecture on it. It's a tricky, messy thing to explain. But he says, if I explain it using Julia in linear algebra, it's clear. So we'll give him a chance on Wednesday to show that revolutionary approach to the explanation of backpropagation.

And I hope for-- I told him he could have half an hour, and projects would take some time. I hope-- now we've had two with wild applause. So I hope we get a couple more in our last class. OK. See you Wednesday. And if you bring-- well, if you have projects ready, send them to me online, and maybe a print out as well. That would be terrific. If you don't have them ready by the hour, they can go-- the envelope outside my office would receive them. Good. So I'll see you Wednesday for the final class.