Howdy. All right. So, we're going to get started. So, this week, we're going to wrap up our discussion on clustering. Today is supposed to be a hierarchical clustering, okay? Just a couple of quick announcements. So, I hope everyone's received assignment one grades. Y'all did pretty well. I think the three times out of the class, you all did the best on assignment one. And maybe chat GBT, I don't know, right? But you all did a great job overall, okay? The main thing that people lost points for were for not following instructions, okay? Please make sure you don't just look into the problems. We also look at the instructions up top, okay? The other thing, too, if you all know it or not, it's in the syllabus. Past the week, I will not change grades, okay? So, next Monday's class, after I end class, no more changes in grades, okay? Because a lot of times, I'll get to the last week of class and guess what people are complaining about in my office. You made a mistake on assignment one. I can't do anything about it, right? Make sure you get the change within the week, okay? The other thing, too, please, please, please do not contact the grader. That's outside their job scope. If you all want to hate anybody, please hate me, okay? So, please come to my office hours. Let me know if there are any issues. Don't contact the grader nor the TA with any kind of grade discrepancies, okay? The other thing, too, that was really big that I think a lot of people had for the two E and F with the variance question and mean or particular variance. Depending on how you set the units, you might get, like, a slightly different answer, and so we're correcting that now, and so that will be revised for any people who have issues with two E and F, okay? The other thing, too, as you all know, assignment two is due this Friday. Like I've been telling people, a lot of people come to my office hours today and were just like, Dr. Peoples, I was confused. You should have came to me before the last minute on the assignment to clear up any confusion, right? The number of times I've had office hours this semester and nobody shows up, please make sure you let me know if you have any questions, okay? We've got one sample answer. So, can we follow this one, like, this structure to submit the next assignment? Can you repeat your question? I couldn't hear it. What are you saying? So, we have basically one answer for the assignment. Excuse me. Hey, excuse me, y'all. When your class is talking, please be quiet. Go ahead. Yes, sir. This is one answer for the assignment one. So, can we assume that this is the format we need to follow, like, this is how we have to organize our answers for the next and the best place? All right, so the question was for those who didn't hear it is that, basically, with assignment one, I provide solutions to y'all, right? The question is, can you use it as a guide for future assignments? I would highly recommend you use it as a guide, right? So, again, the first assignment is just a learning lesson for a lot of people. We'll now just take it moving forward, okay? Got a question. All right, any other questions?

All right, any other questions? Yep. Can you speak loud? Regarding the lecture audios, will lecture seven and eight, like, the audio, will that ever be recovered, or are we assuming that it's gone? Yeah, great question. So, the question is, are lecture seven and eight audio be recovered? Not with this current system. So, right now, what I'm doing is I'm going back to last year's lectures and trying to see if I can get those lectures and down with them for you. I'm having some issues with media sites, so I'll have to reach out to IT to see if they can help me download them. But for right now, if we're not able to recover the audio, I'm just going to ask you about the book chapters and what's on the slide, not anything that I may have mentioned in class, okay? All right, any other questions? My hope is, according to IT, the system should be fixed. So, I'm going to test this lecture out on Zoom and through the user system, so hopefully our issues are resolved, okay? So, when you all write your teaching evaluations, don't blame me for the audio issues. Please complain about IT, not Dr.

Peoples, okay? Just joking there, okay? Any questions? And I think some people are curious, so you all should be able to see, like, what the average is for an assignment. I believe the average is, like, a 90 percent of 45, so you all did really well for the first assignment, okay? All right, so last lecture we covered

expectation-maximization algorithm in more detail, okay? So, last week, last Monday, what did we discuss primarily? What clustering method? Gaussian mixtures, right, okay? And why do we need expectation-maximization for Gaussian mixtures? It's tough to calculate the measure-maximization. I hear a lot of mumbling. You all are grasping, and you're supposed to be leaders, right? Can someone raise their hand and tell me why do we use EM? I want to hear from someone else.

I want to hear from someone else. Go ahead. It's tough to calculate the values inside the summation. Correct, right. Yeah, so when we looked at it, it's difficult because we had that summation within our log, or sorry, we had the summation within our expression, right? And so that's why we use expectation-maximization. And what expectation-maximization, what was the E step for? What did E step do, expectation? What was the expectation step?

What are you doing there? Close. It starts with A. Assignment of points, right? Expectation is you assign your points, okay? And then what's the M step? Maximization. Maximization, so now you're updating your parameters based on the assignments that you have in the previous step, right? And then you're just doing, like, this iterative process, okay? Sorry, I meant to lower the tables down before class. I can't see some of your faces, so make sure everything's clear there. Anyway, right, so we talked about Gaussian mixtures. And then last class, what else did we talk about with expectation-maximization, like another random variable? What did we discuss? What other random variable? Bernoulli, right? So we talked about flipping a coin, and we flipped it multiple times. And so we mentioned this could also be modeled with what random variable? Binomial. Binomial, right? Because the binomial is what? Binomial with multiple trials. But, yes, some of independent Bernoulli random variables, right? So those are some of the things that I'll remember from last lecture, okay? And so on the assignment, hopefully you both get practice with flipping the coins as well as let the Gaussian mixture model for expectation-maximization, okay? All right, so today's lecture is going to focus on hierarchical clustering. I highly, highly encourage you all to make sure you're reading Chapter 14 for this. Again, this is a grad class, so what I like to do is I also like to reference research papers. And so there was a really good survey paper published about two years ago that provides a really good overview about hierarchical clustering, I recommend, okay? And also the other textbook, massimining data sets, is also a good reference. So just to recap, last, the past two weeks we talked about representative base clustering. What is this clustering trying to achieve? So, grouping. I mean, all clustering is grouping, right? But what was special about representative base clustering? Yeah, each cluster has a representative. How do we represent our clusters with our two algorithms? So k-means, what was our representative? The mean, right?

The mean, right? The mean vector, okay? And then when we went to Gaussian mixtures, what was the representatives there? So we were looking at Gaussian distributions, right? And you all mentioned that we needed what? Right. Yeah, for multiple dimensions, mean vector, covariance matrix, and what else did we need? Mixture weights, right? How much weight we're putting on each Gaussian, okay? Yeah, probability, right? And we mentioned that, and we looked at that through the base of what theorem? Or the lens of what theorem? I already told the answer. Base theorem, right? And so now we'll go to a different clustering paradigm, which is hierarchical clustering. How many people are familiar with hierarchical clustering? I say a couple of people. So the goal of hierarchical clustering essentially is like we can look at it through two lens. We can do either divisive or agglomerative clustering, okay? Your book is going to more so focus on agglomerative clustering, but they want to mention in this lecture divisive clustering as well, okay? And so we're going to go through agglomerative first throughout this lecture, okay? So when you hear agglomerative clustering, you want to think about like a bottom-up approach is what we refer to as, okay? And so what you did is you start off with all of your data points as a single cluster, and then you essentially just form

groups combining data points based on similarity or dissimilar, okay? And so with this, we're going to talk about something called a dendrogram, which simply shows the organization of how you're clustering your data, okay? And so does everyone see why it's called a bottom-up approach? We're working our way from where each sample is a cluster all the way up to everything's within one cluster, so like a bottom-up aspect of it, okay? And so we're going to spend a lot of time talking about how do we actually merge these clusters, so it's really important with this, okay? So one way that we represent this is that we represent something called a dendrogram, which is this diagram on the right, okay? So we have our different points.

So we have our different points. How many data points do we have here? Five, right? And so there are different ways that we can merge our data points to get different clusters, okay? And so most of the time, people usually use a dendrogram, but you also might see, like, a table like this as well, okay? So C1, what do you all think that represents? So it's like the starting point, right? So a lot of times people do C0, C1. Sometimes you'll see, like, C1 in this case, right? Like, so the first step is all data samples are one cluster, okay? So, and then we got to the next cluster, and now we're grouping A and B together based off some kind of criterion that we'll talk about throughout the lecture, okay? So I always get this question from students. So do you all think within each merge step, you can only merge, like, one data point at a time, for example? No, you can merge multiple, right?

But in this example, they're just showing, oh, A and B are similar, but you can also have a step where A and B are similar, and then C and D are also grouped together, right? It just depends on how your algorithm is set up, okay? So as you can imagine, there are a lot of different solutions that you can have for agglomerative clustering. And so if someone were to ask you, based on the number of data points you have, how many unique dendrogram or cluster diagrams that you can have, you're going to use this formulation here, okay? So what does this symbol represent? N is the number of samples? Yeah, product, right?

N is the number of samples, okay? And does everyone know what the double factorial means? Now, you know what a factorial is, but what's a double factorial? What is it? Yeah, you're in close to it, right? So if you see a double factorial in Mav, essentially, depending on what the parity of N is, a parity meaning is an odd or even, it'll be a factorial of odd or even numbers, depending on what the number of samples are. So I'll give an example here, okay? So if, you know, say we have five samples, okay? Let me see if I can, sorry, I didn't say it was M. So if we have, say, five samples, and I want to do this double factorial, so it's five and even or an odd number? Odd, right?

Okay. So how do you all think this would be? Five times what? Three times one. And so if we change this to six, what would six be? 642. 642, right. So that's all the double factorial means. Does that make sense, everyone? And so that's the number of unique dendrograms that you can have for a higher-cool cluster, for a glomber cluster, in this case, okay? And so as you can imagine, this algorithm, the design choice that you choose are really important, especially something that we'll talk about, which is called a linkage. All right?

Is there one with me so far? So I feel like double factorial is that special symbol that these get used to. Does it mean, like, factorial or factorial? I don't understand your question. What do you mean, factorial or factorial? It doesn't mean, like, you're taking the factorial of five factorial. Oh, right, yes, right, yes. Yeah, this is a special operation, just meaning, depending what the parity, odd or even, of your samples are, you would change up how the factorial is formulated, right? Yeah, this might be a really good formula to have on your cheat sheet, potentially. All right, so with hierarchical clustering, you don't necessarily need to specify a number of clusters beforehand. You can usually use, like, some kind of, like, distance threshold, okay? And so we talked about distance

before. What's, like, a way that we measure distances that we talked about previously? Yeah, Mahalo, that was Euclidean. Like, whatever distance measure you're looking at, essentially, this is us, you know, comparing. Sorry, let me get my pointer here. You know, essentially, we're saying, like, you know, what is the distance between, say, like, different samples or clusters, like, as we go up in our diagram, okay? And so on the y-axis, this is going to represent, like, your cluster distance. So say this is sample one, sample two. This is the distance between those, right? And then say we merge one and two together. Now we're comparing it with three and four. Now we have, like, a shorter distance between, like, our new merge clusters here, okay? And so people use this to say, okay, given some kind of distance threshold, I want to cut off my hierarchical clustering, and this is where you would, say, select, like, the number of clusters you choose, okay? So we'll talk about this a little bit later, but you want to pay attention to how many lines intersect your diagram to say how many clusters you have, okay? So in this example, say, like, our distance threshold was y3, okay? How many lines do you all see that intersect with y3? Four, right? So we have a total of how many clusters? Four. So that's how you read it, okay? Any questions? All right, so now let's do an example. All right, so for this class, what are all the deliverables for the class? What are all the deliverables? Projects, what else? Assignments and exams, okay? So for the project, say, like, I want to... So we have, like, assignment one grades out now, right? Say, like, I wanted to break you all up into project groups. To spoil it for you all, you won't get to choose who your groups are, okay? So... I won't use this criterion just so you all know. But anyway, essentially, say, like, I was looking at trying to group different students together for, like, a project, okay? And so I have... So I have my students here, right? And then the marks would be, say, like, what they scored on, like, assignment one. Hopefully, nobody's getting 10 points on assignment one, right? But when we look at this, what's another way that you all would say we represent this here? What we talked about before, like, lecture two? A data matrix, right? So each row is what? Sample and then column is a... Right, so you can think about the marks as being, like, a feature, right? A very simple case, so just one D feature, okay? So my next step is to say, let me look at the students, and now let me create a proximity matrix, okay? Some people also call this a similarity matrix, okay? Proximity is usually with distances, similarity would be with some kind of similarity score. So if I'm looking to compare student one and student two, how do y'all think I got this value here? Just subtraction. Are you 100% just subtraction? Yeah, absolutely, I'm looking for a certain phrase. Oh, I see, absolutely different. You all are close. L1, L1 distance, right? Yeah, so just taking the L1 difference or L1 between the different features, okay? And so as y'all can see, this is going to be a symmetric matrix as expected, okay? And so do I have to use L1? No, I can use whatever I want, right? I can use Euclidean, I can use Mahalanobis, whatever I want to do, right? You need to specify some kind of distance matrix for your proximity, okay? So we're going to go throughout the lecture. I want y'all to pay attention to all the different choices that are selected by the user. So the first thing is defining what your distance metric will be, okay? So after that, now we want to assign each data point to a given cluster, okay? So again, we're doing like a bottom-up approach. So we have five data points, we have five clusters, okay? So now the next step is I want to find the smallest distance in the proximity matrix to start merging the clusters together, okay? So hopefully it makes sense, sir. One y to diagonal zero, y is up. You're comparing the points itself, right? So this should be zero, okay? So if we look at this here, which one has like the smallest value, or the two students? One and two, okay? It's also getting away on the slide, right? So that's the easy question, okay? So now we want to like say merge those points, okay? So now we have five clusters of four, now we have a new cluster that has which students? One and two, okay? So now the next design question is how do you merge students or merge data points? And we'll talk about that throughout the lecture, okay? So when you hear me say merging clusters, the phrase for hierarchical clustering is something called leakage, or linkage, okay? How are you linking data points, how are you linking clusters

together, okay? So you can do, so here we use what the maximum value, right? So if we go back to the previous slide here, student one was ten, okay? And so I just said, oh, between those two students take the highest value, okay? And that represents my new cluster here. So now after we get our new cluster, the next step is now we need to compute our what matrix? Different proximity matrix, okay? And just repeat the process until all the data points are merged together. All right, any questions here? So that's pretty much a hello, right?

Pretty easy so far, okay? So where are the two design choices that you all have noticed with this method so far? Two design choices. You can't get like one rule by the time. You can't, but that's not exactly what we touched on, right? So the point was, like, we can either merge certain points at a time or merge multiple good. Right, so we talked about the distance metric for the proximity matrix and also our leakage, which here we use maximum, okay? So those are two design choices so far, okay? So we'll talk about this a little bit later. A lot of this method is really, like, heuristic, so you're, you know, trying different combinations of things to see how well it clusters your data, okay? Yeah? The heuristic means the prior knowledge exit. The heuristic means the expert knowledge. Sorry, yeah, I mean, yeah, you could use heuristics, but also, I guess, like, sorry, empirically is what I mean, yeah. Okay, and so you just repeat this process until, like, the clusters are merged, okay? And now the next step is now we need to use a dendrogram to decide, like, how many clusters we have, okay? So now, going back to, like, our example that we showed before, okay? So we have students one and two, okay? And then what is the distance between students one and two? Three. Three, okay? So, yeah, you just plot that value here, right? And then you just continue to do this as you develop, like, your diagram here. And so the next question that we get is, where should your distance threshold be?

What do you all think? Right, so we talked about intracluster, what? Intercluster compactness and intercluster separability, right? So usually a good rule of thumb is what people do is you want to choose, like, say, choose a distance threshold that has the largest intercluster distance, like, amongst all your samples, okay? So you want to make sure there's, like, a lot of separability amongst your groups. So that's the idea, okay? Like, you only see that if you're, like, if you're playing with distance, you're actually affecting both ways of trade-off, and then increasing something, increasing something. Can you clarify what the trade-off is? Potentially, right?

Potentially, right? Yeah, so, yeah, there's always this inherent trade-off between, like, a compactness and stuff, but ideally you're able to achieve both, but not necessarily, right? It really, it really depends on what you're trying to achieve. Typically, people are focusing, like, on cluster separability with this agglomerative clustering, but if a compactness is important to you, then you might want to choose a threshold that achieves that, right? Yeah, okay, yeah. I did.

It's kind of like a rule of thumb. It's not, like, a hard and fast rule, yeah. Yeah, that's correct, right, yeah, right. Yeah, so the comment here is that if, say, like, you're trying to make sure intercluster distances are large, you might want to try and achieve, like, the least amount of intersections, so, like, the fewest number of clusters that we're grouping here, right? Oh, like, doesn't that mean, like, the maximum is going to be able to have one cluster, and you don't have that. Well, so if all your data points are within one cluster, what is your separability? Yeah, I mean... You say it's infinity, is that true? Yeah. You don't have a single cluster. All right, so you have a single cluster, so what's the difference between the cluster and itself? Oh, okay, zero. Zero, right. Yeah, no need to apologize. Any questions here on how to read a dendrogram? All right, so that's the intuition.

Now let's actually get into the algorithm and also, like, a little bit of math,

okay? So, like I said, it's a pretty simple algorithm for ergometer clustering, okay? So like we talked about before, each data point is going to start in its own separate cluster, okay? Next step is to compute this distance matrix, which we also called, what, proximity matrix, okay? So once we do that, now we need to figure out how we're going to, like, merge our clusters, which is this part here, which we're going to talk about some. And now, once we start merging clusters, now we need to update our proximity matrix and continue to iterate, okay? And then, sorry, I mentioned this point here.

And then, sorry, I mentioned this point here. What is this line saying here? This K is the distance. K is the distance. So what does, like, the absolute value sign mean here?

We'll see. First, what is C? Cardinality, right?

You remember, I remember from the transactions, right? So even though we mentioned that we're using, like, a distance threshold, some people also say, oh, you want to iterate until you have a certain number of clusters, K, okay? So, like I said, usually people use a dendrogram, but also in the pseudocode they're mentioning you want to keep on merging until you reach a certain number of clusters as they are here, okay? But now I want to spend some time talking about the distance matrix, okay? So the idea here is that we need something to define how do we link clusters or data points together, okay? So you all are familiar with Euclidean, right?

How many people heard of Manhattan distance? What's another phrase for this? TicyCab, right?

Also, city block. See, all those mean the same thing, okay? And so depending on what you're trying to achieve, right, different distance metrics can give you, like, different insight, okay? And so the other thing, too, that we talked about before is that for a representative cluster, remember, we have, like, centroids here, okay? But in this case, now we're relying on actual data points to form, like, the cluster that we're looking at, so that's, like, the small difference here. So with this, some common metrics, so, again, this is a big area, say, that you can look at more, but we're just going to focus on five different ways to assess, like, distances between clusters, okay? The first one is going to be single link, okay? So if you all look at this here, it's saying if I have two samples, okay? So what is this norm here? L2 norm, right?

L2 norm, right? Generally speaking, if you don't see a number here, it's going to be the L2, okay? Usually people put, like, one or anything else for, like, another norm, okay? But what is it asking for us to find? Right.

Right. So we want to find two points where the middle distance, so say we have, like, cluster one, cluster two. We're looking for samples that will have, like, the smallest distance between them, okay? So it's, like, single linkage, okay? So this method has an advantage that can handle some, like, non-electrical clusters, right? What do you all think would be, like, a downfall of this method? What would be a downfall of this? Yeah, it's going to be very sensitive to outliers and noise, okay? So if you have any kind of small perturbation within your data, this is going to be highly, highly sensitive to some kind of noise that you may encounter outliers, okay? But the thing is, like I said, it's pretty, like, quick to compute, right? But now with single link, now we transition to complete link, okay? So we talked about the min, now we have the max. What do you all think about this compared to single link?

Would it be more robust outliers, you think? You say no, not exactly. Let's see. What's it saying? The max and min are both sensitive to outliers, right? But think about what we're trying to achieve here. We're trying to group right here and

increase separability, right? So if you think about this, complete link gives you, like, a bigger picture of, like, your data, right? Like, a better, like, global picture. So that's why people say it's more robust to outliers, okay? Because ideally, like you mentioned before, we're trying to achieve, like, large distances between our clusters, and so this would be a bit more robust to, like, some of those things we're looking at the data, okay? Yep. I would say that this might affect, like, might give, like, a, might give us a hold inside. Was it? I would say, like, maybe it's gonna, like, give us a hold inside of the data because it's giving you, like, if it says it's outliers. So you're gonna say, okay, like, this is a cluster, even it should be closer. It should be closer than you. Right, yeah, essentially giving you, like, a global picture, as opposed to, like, the single and giving you, like, smaller view. I would say that maybe most of the, like, most of the points of the, either of the clusters are closer, but maybe two of the outliers are, you know, distance. So then it is not saying anything, right? It's giving a false kind of image, I mean, false message. Can you explain the false message a bit more? It's not clear. I'm saying that maybe two points are very, very far from each other, but most of the, rest of the points are close to each other. So then it is, like, we are only taking the max of this distance. So, like, by calculation we'll get a large distance, but actually most of the points are, of the, both of the clusters are, you know, close. So then it is kind of giving a false perception. Right, so I want to make sure it's clear. I didn't mention any of these are perfect methods, right? Like, each one has their own, like, downfall, right? So there's all some kind of corner case that it can't account for, but, yeah, that's a good point. Alright, any questions here? So single link uses minimum and then complete uses max, okay? So if you're stuck on the exam, just think about single versus complete. Complete being, like, a larger picture of the data, okay? So next, people can also do, like, group averaging. Okay? So what did you all notice before about single and complete link? It's just, like, one. Really looking at, like, essentially one data point, right? So now I'm saying, now let me look at several different samples that belong to a cluster. Let me try and merge my data that way, okay? So would you all say this method is more robust to, like, noise and outliers? Right. And so for those who are taking, like, an image processing class, think about, like, an average filter, right? Or, like, average pooling, for example. What does the average do? It takes the average of the x. Right, yeah, it takes the average, but what does it do to your data? It smooths it, right? Smooth, right? So think about this as, like, a way to smooth any kind of, like, noise within your clustering groups, okay? And so here, what do you all think n i and n j represent? It's the number of samples. Right, number of samples. Yeah, so n i would be the number of samples within cluster i, right? And then n j would be the number of data points within cluster j, okay? And then you all know how to compete, like, the norm, right, between, like, all data points, okay? All right, so the next method to discuss... Excuse me. Yep, the mean, okay? And so the previous three methods relied on either, like, a single data point or multiple data points, right? This one, now we're looking at centroids, so this sounds very familiar to what? K-means. K-means, right? So with this, if you want to... So this, as you could think about, is, like, a combination of representative-based clustering as well as hierarchical clustering, right? So it's kind of like a best of both worlds that people can use to, like, merge data points together, okay? So, and again, the main takeaway here is that, you know, if I gave you all five of these methods, which one would you say is, like, the least robust method? Single link, right? And so the thing about as we're going through this, we're just essentially strengthening, like, our algorithm for each case, okay? All right, so last one, which is usually the most common method, is going to be a minimum variance, okay? So we have, like, our... Say, like, we have, like, our different clusters, right? And so we want to compute, like, compare, like, the means, okay? So very similar to the mean distance. And so the idea here is that we want to minimize the total within cluster variance, okay? So when you hear me say minimize the total within cluster variance, what does that mean? Compactness, right? Yeah. So ideally with this... That's what's called minimum variance, because you want to minimize the variance, like, amongst the data

points that you're merging. So that's why people call it minimum variance for Ward's method, okay? Yeah, with any... Because since you want your clusters to be tight, right? And then remember what we talked about before, with the Dendrogram, what are we focusing on? So this mentions cluster compactness, and then Dendrogram's looking at what? Separability, right? So ideally, if you're working with Ward's method in tandem with your Dendrogram, you're going to have compact and well-separated clusters, ideally. Any questions here? Like, how the level one and level two works? What is it?

What is it? I can... How the level one and level two are working? Right. So look here, right? So what do you see here? This... Right, how many clusters are here? Two, right? Cluster one, cluster two is level one. Now we want to merge those data points in such a way that they're going to be close together. So that's what, like, level one and level two means here, okay? This is before we merge, after we merge, okay? Does that make sense? Also, everything I'm covering to you all, like, I really like this blog post at the end of the slides, or at the bottom of the slides, so I highly recommend they get through all, like, these visuals here. So the book gives you the math, all these visuals I'm pulling from this data resource here, so I highly recommend reviewing that, okay? Yep. Yep. Ideally, right. Yes, now remember we're going to have, like, multiple clusters, right? So ideally, we're trying to minimize the total within cluster variance across, like, all of our data, right? This is just showing an example of, like, merging for one cluster. Like, this is after you merge. Correct, right. Yep. All right, so that was all about, you know, looking at our distance matrix, computing, like, essentially, like, a similar or dissimilar, okay? So now let's get into, like, how do we actually, like, update our clusters, okay? Or update our distance matrix, excuse me. So the main formulation or the equation that we use here is going to be something called Lance Williams, all right? And so these are some people that worked on this generalization of, like, how do you update the distance matrix, okay? And so if you all see here, we have a total of four terms, okay? So each term is capturing some unique information, okay? So CIJ is going to be a cluster that we, say, merged points I and J before, and now we have, like, essentially, like, a candidate cluster that we're looking to try and update, like, our distance matrix, okay? And so if you look here, each term is capturing some unique information about, like, our merging, okay? So if you all were to look at this first term, what is this capturing here? So remember, what are we trying to do? We're trying to update our distance matrix based off our new cluster IJ compared to a separate cluster CR, okay? So what are we comparing here? All right, distance between I and R, right? And then what's the next term looking at? C and J, right? And now this is looking at what? I and J, right? So this is saying CI and CJ were separate. Let me compare it to this separate cluster CR, and now what is this term looking at? The relationship between the clusters that just joined together, okay? All right? And now similarly, this is saying, like, how does our essential word, like, discrepancies between... So I and J just merged, right? So cluster I and cluster J merged. Now we're saying, like, what is the difference between these, like, two merges that we're looking at, okay? And so this gives you a holistic view of, like, how do you update your distance matrix, okay? How is the last one different? You tell me. Look at the math. Okay, so think about it concretely, right? So how many clusters did we have before? Three. CI, CJ, and what else? CR and R. Right? So what do you think these first three terms are doing? Suffering... Like, how much...? Since, like, pairs of clusters, right? Okay? And now what do you think this last term is doing? The whole picture, right? With those three clusters. Does that make sense? So how does each one compare individually? And then what's the relationship between, like, all of them, the indicator, right? So everything that we just covered before about single-length rewards measure can pretty much be derived from White-Lance-Williams formula, okay? Again, I want to ask you all to derive this on the exam, but this would be something good to know in practice is that given the White-Lance-Williams formula, you should be able to solve for any of these, like, linkages that we talked about before. Okay? So... So what do you all notice about

the beta term here for single-length? It cancels the distance between the CI and CJ for the first three measures. It cancels... Right. It cancels this term here, right? Why? The previous question doesn't matter. Let's think about single-length, right? Does single-length look essentially like the cluster representative? Just looks at the individual data points, right? So you can kind of think about just work through this formula and figure out why is a certain term zero, right? And you can try and see, does it make sense based on what we showed in the previous slide, okay? So I'd encourage you all to say, okay, let me look at single-length, let me plug these in, and does it make sense intuitively for what it's trying to capture? Any questions so far? All right, so good news for you all. Don't have to implement any of this yourself. All this is available in a second learn, okay? So I believe the default is usually wars, but you're free to use whatever you want. And depending on which linkage you use, you're going to get very different results, which makes sense, right? We talked about this is a way that we form our proximity matrix. And so depending on our data, our results are going to change pretty drastically, okay? All right, so with clustering, we focus on data points, but do you all believe me that we can also cluster features, okay? So you can also use agglomerative clustering for dimension-eye reduction, okay? So what's another dimension-eye reduction technique we talked about before? PCA, right? So a simple example, so I put it at the end of the slides for those who are interested. Do you all remember the IRES data set? Right, I remember in class, again, an example using PCA for the IRES data set. At the end of the slides, there's also another example that uses the IRES data set, but uses, like, feature agglomeration to do, like, the reduction. So you can look at that and compare PCA versus this method, okay? But essentially, just to give you all a visual, how do you take, like, a computer vision or image processing class? Okay. But they did a very simple example here, but say, like, we have, I'm assuming it's, like, a 35x35 image, right? And so you can think about, like, as each pixel is, like, a feature that we're looking at, okay? So each pixel has, like, a certain value. And so what they did was they took agglomerative clustering and said, are there any, like, redundant pixels within the image? And now we can group those features together to get, like, a reduced computation. So that's all they're showing here. Original pixel is each, or each pixel is a given feature. We can group them together to save computation, okay? So, again, I would encourage for those that are interested, I have an example using IRES data set in the slides, okay? All right, so that was all agglomerative. Now we're going to get to the device of any question about agglomerative clustering, right? So where are the two main design choices we discussed? Right, proximity matrix, right, and then we also talked about, like, our related to that, which is the linkage, right? So there's, like, a lot of different choices with this clustering, okay? All right, so agglomerative we referred to as what kind of approach? What kind of approach was agglomerative? Bottom up, right? So now what do you think the device of this? Top down, okay? So, agglomerative we started off with all our data points are a cluster. Now we're going to flip it around, and now we're going to start with all of our data points are within one cluster, and then we're going to go down until, like, all data points are within their own given cluster, okay? So pretty much the same thing, but just reversing, okay? So with this, again, for this section your book doesn't go into detail with this, so the main thing would really be from the slides. But one popular method for device of clustering is something called bisecting K-means. Have anybody heard of this before? Yeah, so essentially, before we talked about K-means, it was what type of clustering method? What type of method was K-means? Representative, right? And now we're saying can we combine both representative and hierarchical clustering, like, within one algorithm, okay? So I think some of you all came to me after class and mentioned, if you're interested in research, a lot of times people will say, oh, I have this algorithm specialized in this, another algorithm specialized in this, let me see if I can combine them together to create a new algorithm. Essentially the same idea here, okay? So you're going to start off with all your data points within one cluster, and then bisecting, you're going to break it up into two clusters, okay? And then you just continue this process until you reach a certain criterion that

you're looking for, and we'll talk about what that means in the coming slides. But essentially, start with your data, bisect it, so buy two, break it into two clusters, take the individual clusters, and then bisect it again until you stop. Yeah, essentially like a binary tree, right? So generally speaking, why people like this is, one, it's going to be a bit more robust to outliers than the typical K-means, as what we mentioned here. Also, you can handle some more complex data structures as opposed to the typical K-means. Why do you think that's the case? So what do we talk about with the shortcoming of K-means, particularly like, say like batch K-means? It can be intense, right, because like we have a lot of samples, right? And so compared to like bisecting mean, we're going to like break up the data, and so we're able to achieve like, identify like more complex data structures as opposed to looking at everything like all at once, essentially as saying here, okay? And again, the advantage, like we mentioned before, is now we have this hierarchical
structure that can be like easy to follow, like our denture grant that we're looking at, okay? Improved efficiency, depending on your data, it can potentially converge faster. Also, people recommend this if you have a lot of data samples and you have a lot of, a high number of clusters that you need. So that's what it means by improved efficiency here, okay? I don't remember asking you this. I think it's like after our K-means lecture, but someone asked me, could you decide your clusters based off like the number of data points within a given cluster? So this method you can, okay? So two things that people look at to decide like if your, your batch section is within psychic learning, you have something called the biggest inertia. And so this is going to split your clusters based off the largest SSE. Why does that make sense to you? Why would I care about the largest SSE? It makes sense because we use N-K-means. It makes sense because we use N-K-means, not exactly. Right.

Oh, okay. Actually because we're trying to make it more compact. Right.

Does that make sense to everyone? So we have a cluster that has a very large SSE. What does that mean? The data points are very spread out, right? So it makes sense that we want to separate that cluster because those data points are probably very different. Does that make sense to everyone? So it's based off SSE, and now hopefully this is intuitive to you all as well. We can also do the largest cluster. That just means, oh, if I have, say, in one group I have 100 students, in the other group I have 20, I probably want to split the bigger group up, right, based on the number of data points, okay? So what these methods, which one do you think is least expensive to compute? Was it? So one, we're just keeping up with the number of data points. With this one, we're having to compute, like, a loss function each time, right? So, right, large question, right? Just expensive, right? Yep.

Yep. Obviously, a large cluster is, like, least expensive. Can you just simply just count in the number of data points, right? There's not much computation there, okay? Okay, like, do you have any kind of actual, like, key, specific key that we can initialize to? Do you have any, you said any K that you can initialize to? Yeah, you could, yeah, so, you can, so, well, the thing is, what do we mention about hierarchical clustering? We don't need to specify the number of clusters beforehand, right? So with this, what would your threshold be for? Right, essentially, like, your SSE, like, what is your desired SSE, right? So you could have, like, a dendrogram where you plot, like, remember, we plot the distance, but you could also plot, like, your loss function, right? And then the same thing for what's the number of data points, okay? So is there some number of data points that you want the largest cluster to be? Does that make sense, everyone? I mean, like, does it give you, at the end, say, I mean, my understanding is, at the end, you're going to get a couple of, say, different clustering, and it's going to show you, like, this, that, this level, this, that, this level, is it going to show you, like, this way, or do you have to pick before? I do, yeah, so that's a good thing about hierarchical, well, good and bad thing, we'll talk about in a couple slides,

is that you can create your dendrogram, right, which is a good visual, but what's the downside of a dendrogram, you all think? It can be, right, if you have, like, a lot of data points and a lot of clusters, it's going to be really hard to visualize that dendrogram, right? So that's, like, a disadvantage, okay? So typically people aren't using, like, these hierarchical clustering methods, generally speaking, for, like, large amounts of data, right? It can be pretty expensive, and also it can't be, like, that clear to see, okay? And thank you for your transition to my next slide, right? So, one thing that we talked about before in class, no free lunch, right? So this method has advantages and disadvantages, okay? The good thing is that we don't make any assumptions about the cluster shapes, okay? What did we assume for k-means? Spherical clusters, right? Like, hyper-sphere and higher dimensions, okay? We don't need to set the number of clusters beforehand, but what do we have to choose in turn? What k is the number of clusters? What did you mention before? Distance threshold, right? So we mentioned here that we don't need to set the number of clusters, but we have to indirectly by saying what is, like, our distance threshold, right? And also, it's interpretable because of our dendrogram, okay? So we can see, like, how the points merge, okay? Begin the main thing, a big takeaway from this, from you all, is it's going to be very costly for large data sets. Why? You have to compute the distance between the points. Right, proximity matrix, right? So as we add more points, since we're, you know, yeah, squared, right? We're having a squared penalty or quadratic penalty, okay? And also, as you add more samples, add more clusters, it's going to be really hard to visualize, okay? And also, I hope this is another big takeaway that you all get from this lecture, is that the results are going to heavily depend on what linkage function, what distance you're using as well. So this is really important. Also, we didn't talk about this, right? But it's also not that straightforward to apply it to new data, okay? Because with the algorithm, remember, we're computing that distance matrix, and we're having to update it based off our merge. How are you going to throw new samples into that? Right, you have to recompute the distance matrix, right? Also, like, how do you, like, restart your merging, right? So if you already started your merging, how do you introduce, like, new samples into there? So it's not as straightforward to do that, okay? So, again, this is also an area of study, too, right? For those who are interested, there are different ways that you can look at that, and this could also be, like, an opportunity to develop, like, a new algorithm, okay? All right, yep. Yeah, so, like I said, by the time you all get through my class, you're going to get tired of seeing this diagram, okay? But I love this diagram, okay? Yeah, go ahead. Yeah, great question. Yeah, thank you for that. I just wanted to clarify this. So one of your questions that's a good question is, what do you all think this arrow through the mapper means? What do you think that means? So what did you do? I wasn't. I mean, the correction you get from your... Yeah, the correction. Yeah, updating your model. So, yeah, so going through it just means, oh, based off my cost-function learning algorithm, now let me update my weights or my mapper I'm looking at. So just to update. Yeah, why don't you go through it and not integrate it? Oh, yeah, so it's not confusing, right? Because if I pointed it into it, then you would say, oh, is there something going into the mapper, right? So, could you have data going in, data going out? We draw it over the diagram, so it means that you're updating it. Does that make sense? Yeah, I feel like it's affecting the model itself. It's not like... Right, yeah, we're not, like, appending it to, like, our data, right? You have a great question. All right, so you all should be pros to this by now. What happens to data D? You don't have data D? Right, data D is labels. It's unsupervised, we don't have it, okay? All right, this has been a trick question for you all, all semester. What's coming in and what's coming out? It's events, I mean, for the hierarchical one? Well, what are we... So, yeah, hierarchical, what are we trying to do? What's our goal? Clustering, right, so what's going to be output? Groups, right? Okay. Between representative clustering, what's the only thing that changed? The mapper, right? Data N, output's going to be groups, okay? This is a little tricky, okay? What could you use as, like, kind of like your cost function, so to speak? Yeah, it's got to be... I heard someone say it. Clustered, yeah, it relates to the proximity

matrix. How do we define, like, our proximity matrix? They're using the other linkage. Linkage, right. So, linkage, because we want to encourage either similarity or dissimilarity, that's our, like, objective or, like, our cost, right? We want to either maximize similarity or minimize dissimilarity or both, okay? And so what do you think the learning algorithm will be here? Oh, it depends on the linkage. Depends on the linkage, right?

I agree. So what do we talk about with the linkage? Oh, actually, it was that function that you can get... Yeah, you're close. You guys are close. Yeah, what's the name of that method we talked about? So we talked about single, linkage, right?

So we talked about single, linkage, right? We talked about complete. What was that formula we talked about? You all know it. You all know it. This is the formula. Yeah, Lance Williams, yeah. Why is it Lance Williams? Why is it Lance Williams? Because you can... We can get any linkage that we talked about before, right? You remember that slide I showed? We can get complete linkage. We can get single linkage. Everything is just simple.

This is Lance Williams' formula. What's changing? What's changing within the Lance Williams' formula? It's practice. The weights. Because how many weights do we have? Right.

So essentially, depending on how we change alpha, alpha i, alpha j, beta, and gamma, we can get different linkages. And so now we have a picture of our algorithm of hierarchical clustering. So now, pull that together.

Same question I asked you all for. There's no human in the loop, right? We got our algorithm clustering, our linkage that we defined, and now we have our Lance Williams to update our proximity matrix. And that's hierarchical clustering. All right, so I want you all to go a little early today, but just one thing here. So besides technical things I tell you on the class, what I tell you about grad school is really important. Make what? I heard someone say it. What I encourage you to do in grad school? Make friends. Make friends, yeah. So this project came about of me being friends with somebody in computer science, okay? And so what they specialized in was human-centered interaction or human-centered computing, okay? So one thing that we looked at was saying, so again, they look at human-centered computing, so they look at like a lot of diversity issues typically, right? And so one thing we asked the research question is, does having more diverse representative data hurt performance in the long run? That's like the big research question. And so what we did was before we trained a model, we did a pre-processing step using a divisive clustering technique, which the name of the algorithm is called applications quest, but it's pretty much just like a fancy bisecting K-means, okay? And by using bisecting K-means, we were able to improve the diversity of data without hurting performance, okay? And so this is the application where I have used a glomerative clustering practice, okay? So all that, you're like, I know it's like a boring professor story. The biggest takeaway from here is please make sure you're networking in grad school, okay? With that, if no one has any questions, so I'll let you all go a little bit earlier today. On Wednesday, we'll start talking about the last bit of clustering, okay? But I'll see you all on Wednesday. Thank you.