JEREMY KEPNER: Welcome. Happy Halloween. And for those of you watching this video at home, you'll be glad to know the whole audience has joined me and they're all dressed out in costumes. And it's a real fun day here as we do this. So making me not feel alone in my costume. A lot of moral support there, so that's great.

So, yeah. So this is Lecture 05 five on Signal Processing on Databases, just for a recap. For those of you who missed earlier classes or are going this out of order on the web, signal processing really alludes detection theory, finding things, which alludes to the underlying mathematical basis of that, which is linear algebra. And the databases really refers to working with unstructured data, strings, and other types of things. Two things that aren't really talked about together, but we're bringing them together here because we have lots of new data sets that required it.

And so this talk is probably the one that's getting most into something that we would say relates to detection theory, because we're going to be dealing with a lot with background data models. And in particular, power laws and methods of constructing power law data sets, methods on sampling and fitting, and using that as a basis for doing the kind of work that you want to do. So moving in here. So just your outline. And so we've got a lot of material to go over here today.

This is all using the data set, when we get into the data set that we talked about in the last lecture, which is this Reuters data set. So we'll be applying some of these ideas to that data set. So just going to-- introduction here, and then I'm going to get to sampling theory, and subsampling theory, various types of distributions. And then end up with the Reuter's data set.

The overall goal of this lecture is to really develop a background model for these types of data sets that is based on what I'm calling a perfect power law. And then we're going to, basically, after we can construct a perfect power law, we're going to sample that power law and look at what happens when we sample it. What are the effects of sampling it? And then we can use the power law to look at things like deviations and such.

Now you might ask, well, why are we so concerned about backgrounds and power laws? And it's because here's sort of the basis of detection theory on one slide. So in detection theory, you basically have a model which consists of noise and the signal, and you have two hypotheses, H0, which is there's only noise in your data, and H1 that there's signal plus noise. And so, essentially, when you do detection theory, what you're doing is given these models, you can then compute optimal filters for answering the question, is there a signal there or is it just noise? That's essentially what detection theory boils down to.

In this data, now when we deal with graph theory, it's obviously not so clean in terms of our dimensions here. We'll have some kind of high dimensional space and our signal will be projected into that high dimensional space. But nevertheless, the concept is still just as important that we have noise and a signal. And that's what we're trying to do here.

Detection theory works in the traditional domains that we've applied it to because we have a fairly good model for the background, which tends to be Gaussian random noise. It's kind of the fundamental distribution that we use in lots of our data sets. And if we didn't have that model, it would be very difficult for us to proceed with much of detection theory. And the Gaussian random noise model works, because in many respects, if you're collecting sensor data, you really will have Gaussian physics going on.

You also-- law of large numbers in terms of if you have lots of different distributions that are pulled together, they will end up beginning to look like a Gaussian as well. So that's what we have in many of the traditional fields that we've worked in in signal processing, but now we're in this new area where a lot of our data sets arrive from artificial processes, processes that are a result of human actions. Be it data on a network, be it data in a social network, be it other types of data that have a strong sort of artificial element to them. We find that the Gaussian model does not reveal itself in the same way that we've seen in other sets.

So we need this. We really need a background model, so we have to do something about it. So there has been a fair amount of research and literature to come up with first principles, methods for creating power law distribution in data sets. We talked a little bit about these distributions in the previous lectures. And they've met with mixed results.

It's been difficult to come up with, what is the underlying physics of the processes that result in certain vertices in graphs having enormous number of edges and other vertices only having a few? And so there has been work on that. I encourage you to look at that literature. Here we're

going to do much more of a-- sort of go from the reverse direction, which is let's begin by coming up with some way to construct a perfect power law. With no concept of well, what is the underlying physics motivating this? Essentially, a basic linear model for a perfect power law.

That we probably can do. And then we'll go from there. And linear models are something that we often use in our business. And it's a good first starting point. So along those lines, this is a way to construct a perfect power law in a matrix. This is basically a slide of definitions here, so let me spend a little time going through them.

So we're going to represent our graph or our data as a random matrix. Basically of zero where there's no connection between-- this is a set of vertices connected with another set of vertices. There are N out of these vertices and N in of these vertices. You have a dot here. The row corresponds to the vertex at the edge left, and the column corresponds to the vertex that the edge is going into.

So this adjacency matrix A is just going to be constructed by randomly filling this matrix with entries. And the only real constraint on them is that when you sum A, we're going to allow multiple edges. So you can have more-- the values aren't just zero and one, but they can have more than that. But when you sum the matrix A, all its values up, you get a value M, which is the total number of edges in the graph. So we have essentially a graph with-- this could be a bipartite graph or not. But with essentially N out vertices, N in vertices, and M total edges.

The perfect power law, we're going to have essentially two perfect power laws here. One on the out degree. So if you sum these rows, and then you do a histogram, you're going to want to produce a histogram that looks something like this. So you have an out degree for each vertex. And this would show how many vertices have that out degree.

And so our power law says that these points should fall on a slope with a negative power law coefficient of alpha out. So that's essentially the one definition there. And then when you likewise have another going in the-- for the other degree. So the in degrees have their own power law.

So these are the definitions. This is what we're saying is a perfect power law. So we're saying a perfect power law has these properties. Now we're going to attempt to construct it. We have no physical basis for saying why the data should look this way. We're just saying this is a linear

model of the data and we're going to construct it that way. And again, these can be undirected, multi-edge, we can allow self-loops and disconnected vertices, and hyper-edges. Anything you can get by just randomly throwing down values onto a matrix. And again, the only constraint is that the sum in both directions is equal to the number of edges.

So given that, can we construct such a thing? Well, it turns out we can construct such a thing fairly simply. And so in MATLAB we can construct a perfect power law graph with this four line function here. It will construct a degree distribution that has this property. And the three number parameters to this distribution are alpha, which is the slope, dmax, which is the maximum degree vertex, and then this number Nd, which is roughly proportional to the number of bins that we're going to have here. Essentially the number of points. It's not exactly that, but roughly proportional to that.

And so when you do this little equation here, the first thing that you will see is we are going to be creating a logarithmic spacing of bins. We kind of need to do that here. But at a certain point, we get below a value where the spacing will be-- we'll have essentially one bin per integer. And so these are two very separate regimes. You're going to have one which is called the integer regime, where basically each integer has one-- is representative. And then it transitions to a logarithmic regime.

You might say this is somewhat artificial. It's actually very reflective of what's really going on. We really see here-- we really have a dmax, we really have, almost always, a count at 1. And then we have a count at 2 or 3, and then they start spreading out. And so this is just an artificial way to create this type of distribution, which is a perfect power law distribution.

So it's a very simple, very efficient code for creating one of these. It has a smooth transition from what we call the integer bins and the logarithmic bins. And it also gives a very nice what we call poor man slope estimator. So there's a lot of research out there about how do you estimate the slope of your power law. And there's all kinds of algorithms for doing this.

Well, the simplest way is just to take the two endpoints. Take the first point and the last point, and you know you're perfectly fitting two points. And you could argue you're perfectly fitting the two most important points. And you get this nice, simple value for the slope here.

In addition, you can make the argument that regardless of how you bin the data, you'll always have these two bins. You will always have a bin at dmax and you will always have a bin at 1. And all the other bins are going to be somewhat a matter of choice, or of fitting. And so again,

that's another reason to rationalize alpha.

So I would say, if you plot your data and you have to estimate an alpha, then I would just say, well, here's what you do. And it's as good an estimate of alpha as any. And it's very nicely defined. So we call that-- when we talk about estimating the slope here, this is the formula we're going to use.

So far, this code has just constructed a degree distribution, i.e., the degree, and then the number of vertices with that degree will be the outputs of this perfect power law function. We still have to assign that degree distribution to an actual set of edges. OK? And here's the code that will do that for. It will say, given a degree distribution, it will create a set of vertices that do that.

Now, the actual pairing of the vertices into edges is arbitrary. And in fact, these are all different adjacency matrices for the same degree distribution. That is, every single one of these has the same degree distribution in both the rows and the columns. So the actual which vertices are connected to which is a second order statistic. So the degree distribution in this first order [INAUDIBLE], but how you want to connect those vertices up is somewhat arbitrary. And so that's a freedom that you have here.

So for example, if I just take the vertices out of here and I just say, all right, every single vertex in your list, I'm just going to pair it up with yourself, I will get an adjacency matrix that's all diagonals. Essentially, all self-loops. If I take that list and just randomly reorder the vertex labels themselves, then I get something that looks like this. If I just randomly reorder the edge pairs, I get something like this. And if I randomly relabel both the vertices and reconnect the vertices into different edges, I get something like this.

And for the most part, when we talk about our randomly generating our perfect power laws, we're going to talk about this. Which is probably most like what we really encounter. It's essentially something that's equivalent to randomly labeling your vertices, and then randomly taking those vertices and randomly pairing them together.

So that basically talks about how we can actually construct a graph from our perfect power law edge. Now, so this is a forward model. Given a set of these three sort of parameters, we can generate a perfect power law. But if we're dealing with data, we often want slightly different parameters. So as I said, before our three parameters were alpha, which is greater than 0, dmax, which is the highest degree in the data, which we're saying is greater than 1, and then

this parameter Nd, which roughly corresponds to number of bins.

So we can generate a power law model for any of these values here that satisfy these constraints. So that's a large number. However, what we'll typically see is that we want to use these parameters. So we'll want to have an alpha, a number of vertices, and a number of edges, is more often the parameters we want to work with. And we can compute those by inverting these formulas. That is, if we compute the degree, we can sum it to compute the number of vertices. And likewise, we can sum the distribution times the degree to get the number of edges.

So given these, an alpha and our model, we can invert these. All right? And what you see here is for a given value of alpha, the allowed values of N and M, given-- that is, the values of N and M that will be a power law. So what you see is that not all combinations of vertices, vertex count and edge count, can be constructed in a power law. There's a band here. This is a logarithmic graph. It's a wide band. But there's a band here of allowable data that will produce that.

And typically, what you see is kind of the middle of this band is like a ratio of around 10, which happens to be the magic number that we see in lots of our data sets when people say, I have power law data. Someone will ask you, well, what's your to vertex to ratio? And [INAUDIBLE] we say, it's like 8, or 10, or 20, or something like that. And again, you see it's because in order for it to be power law data, at least according to this model, it has to fall into this band here.

You'll also see this is a very nonlinear function here. And we'll get into fitting that later. And it's a nasty, nasty function to invert, because we have integer data, and data that's almost continuous. And it's a nasty, nasty bus-- we can do it, but it's kind of a nasty business. But given an alpha and an N that are consistent, we can actually then generate a dmax and an Nd that will best fit those parameters.

So let's do an example here. So I didn't just dress up in this crazy outfit for nothing. We have a whole Halloween theme to our lecture today. So this is-- when I go trick or treating with my daughter, of course, our favorite thing is to do the distribution of the candy when we're done. And so this shows last year's candy distribution. We'll see how it varies.

As you can see, Hershey's chocolate bars, not surprisingly, extremely popular. What else is popular here? Swedish fish, not so popular. Nestle's Crunch bars, not so popular. Again, I

actually found this somewhat-- this list hasn't change since when I went trick or treating. This is a tough list to break into. Getting a new candy that makes it to Halloween-worthy candy is pretty hard.

So this year shows the distribution of all the candy that we collected. And here are some basic information. So we had 77 pieces of candy, or distinct edges. We had 19 types of candy. Our edge to vertex ratio was 4. The dmax was 15. So we had 15 Hershey's Kisses. N1, we had eight types of candy that we only got one of. And then our power slope was alpha.

And then our fit parameters to this, when we actually fit, where we got 77, 21, and M/N of 3.7. And this shows you the data. So this is the candy degree. And this is the number. And this shows you what we measured. This is the poor man's slope here. This is the model. And then one thing we can do is actually-- which is very helpful-- is we can re-bin the measured data using the bins extracted from the model. And that gets you these red x's here, or plus signs here.

And we'll discover that's very important. Because the data you have is often very rarely binned in a way that's proper for seeing the proper distribution. And we can actually use this model to come up with what a better set of bins would be, and then bin the data with respect to that. So that's just an example of this in actual practice.

So now that we have a mechanism for generating perfect power law, let's see what happens when we sample it. Let's see what happens when we do the things to it that we typically do to clean up our data. I bring this up because in standard graph theory, as I've talked about in previous lectures, we often have what we call random, undirected Erdos-Renyi graphs, which are basically vertices without direction. And usually the edges are unweighted. So we just have a 0 or a 1. So very simplified graphs. I'm actually going to-- getting a little hot here in the top hat.

So a lot of our graph theory is based on these types of graphs. And as we've talked about before that our data tends to not look like that. So one of the things we do so that we can apply the theory to that data is that we often make with data look like that. So we'll often make with data undirected. We'll often make the data basically unweighted and other types of things so we can apply all the theory that we've developed over the last several decades on these particular types of very well studied graphs.

So now that we have a perfect power law graph, we can see what happens if we apply those

same corrections to the data. And so here's what we see. So we generated a perfect power law graph. The alpha is 1.3. The dmax was 1,000. Our Nd was 50. This generated a data set with 18,000 vertices and 84,000 edges. And so here's a very simple way to make it.

We're going to make it undirected by basically taking the matrix and adding its transpose, and then taking the upper diagonal. This is actually the best way to make an adjacency matrix undirected, to take that upper portion, because a lot of the statistics that-- basically it saves you having to deal with a lot of annoying factors of 2. So a lot of times, we'll just do A plus A transpose, but then you get these annoying factors of 2 lying around. And so this is a way to sort of not do that.

So we're getting rid of-- we've made it undirected. We're made it undirected by doing that. We're going to make it unweighted by basically converting everything to a 0 or 1, and then back to double. So that makes it unweighted. And then we're getting rid of the diagonal, so that eliminates self-loops.

So we've done all these things. We've cleaned up our data in this way. So what happens? Well, so the triangles were the input model. Well, now we've cleaned up our data, and we see this sort of mess that we've done to our data. And in fact, I'll call this-- in keeping with our Halloween theme here, we'll call this our witch's broom distribution here. And if anybody's looked at degree redistributions on data, it will be like you'll recognize this shape instantly because you have this bendiness coming up here. And then sort of fanning out down here. Very common thing that we see in the data sets that we plot.

And in fact, there's not an insignificant amount of literature devoted to trying to understand these bumps and wiggles, and do they really mean something underlying about the physical phenomenon that's taking place. And while it's the case that those bumps and wiggles may actually be representative of some physical phenomenon, based on this, we also have to concede the fact it's also consistent with our cleaning up procedure. That is, the thing we're trying to do to make our data better is introducing nonlinear phenomenon on the data, which we may confuse with real phenomena.

So this is very much a cautionary tale. And so based on that, I certainly encourage people not to clean up their data in that way, and keep the directedness. Don't throw away the self-loops. Keep the weightedness. Do your degree distributions in this way. And live with the fact that that's what your data really is like and try and understand it that way, rather than trying

cleaning up in this way.

Sometimes you have no choice. The algorithms that you have will only work on data being that's been cleaned up in this way. But you have to recognize you are introducing a new phenomena. It's a highly non-linear process, this cleaning up, and you have to be careful about that.

However, given that we've done this, is there a way that we can recover the original power law? So we can try that. So we have here is the data that we-- the original data that we cleaned up is now these circles. OK? And we're going to take that data set and compute an alpha and an N and M from it from using pour inversion formulas. And then compute what the power law of that would be. So that's the triangles.

So here's our poor man's alpha fit. This is our model. This is what the model is. These triangles here is the model, we're saying. And then we can say, aha, let's use the bins that came from this model to re-bin these circles onto these red plus signs here. So that's our new data set. And what you see is that we've done a pretty good job of recovering the original power law.

So if we had data that we observed to look like this, we wouldn't be sure it was a power law. Like, we don't know. Say, well, what's this bend here? And what's this fanning out here? But then if you go through this process and re-bin it, you can be like, oh, no, that really looks like a power law. And so that's a way of recovering the power law that we may have lost through some filtering procedure.

Here's another example. So what we're going to do is essentially take our matrix and compute the correlation of it. We talked a lot about-- if we say I have an incidence matrix, we multiply it to do correlation. In this case, we're treating our random matrix as not an adjacency matrix, but as an incidence matrix, a randomly generated incidence matrix. And so these are, again, the parameters that we use.

We're converting it to all unweighted, all 0's and 1's. And then we are correlating it with itself to construct the adjacency matrix. Taking the upper diagonal, and then removing the diagonal. And this is the result of what we see.

So here's our input model. Again, the triangles. And then this is the measured-- what we get out from there. And if you saw this, you might be like, wow, that's a really good power law. In

fact, I've certainly seen data-- I mean, most the time I would see, yep, that is a power law distribution. We absolutely have a power law distribution.

However, we then apply our procedure. So again, we have our measured data. OK, we're going to do our parameters here. Get our poor man's alpha parameter. And then fit, the triangles are the new fit. OK. And then we use those bins to re-bin. And we see here that when we actually re-bin the data, we get something that looks very much not like a power law distribution.

So there's an example of the reverse. Before we had data that didn't look like a power law, but when we re-binned it, we recovered the power law. Here we have data that may-- just sort of in this random binning may look like a power law. We actually see it has this bump. And then continuing with our Halloween theme, we can call this the witch's nose distribution, because it comes along here as this giant bump and then goes back to a power law. And this actually, there's meaning for this. And we will see this later in the actual data. But this is not just that when you do these correlation matrices, certain types of them, particularly self-correlations, very likely will produce this type of distribution.

But again, even though we have this bump, you would still argue that our linear power law is still a very good first order fit. So we still captured most of the dynamic range of the distribution. And this is now a delta from that. And so we're very comfortable with that, right? We start with our linear models. That models most of the data. And then we have a second order. If we wanted to, we could go in and come up with some kind of second order.

Subtract the linear model from here and you would see some kind of hump distribution here. And you could then model your data as a linear model, plus some kind of correction. Again, very classic signal processing way to deal with our data, and certainly seems as relevant here as anywhere else.

Let's see here. And again, so the power law can be preserved as we talked about there. So moving on, another phenomenon that's often documented in the literature is called the densification. In fact, there's many papers written on what is called densification. This is the observation that if you construct a graph, and you compute the ratio of the edges to vertices over time, that ratio will go up.

And there's a lot of research talking about the physical phenomenon that might produce that type of effect. And so while that physical phenomenon might be there, it's also a byproduct of

just sampling the data. So for instance here, what we're going to do is we created our perfect power law graph and we're going to sample it. We're basically going to take subsamples of that data.

And we're going to do it in little chunks, about 10% of the data at a time. And the triangles and the circles show when we look at each set of data independently. And then we have these lines that show what happens when we do it cumulatively. We basically take 10% of the data, then 20% of the data, then 30% and move on here.

And we have two different ways of sampling our data here. Random is, I'm just taking that whole matrix and I'm randomly picking edges out of it. And what you see is that each sample has a relatively low edge to vertex distribution. But as you add more and more and more up, it gets denser. And this is just simply the fact that given a finite number of vertices, you eventually, if you kept on adding edges and edges and edges, eventually it would go-- this would become infinite.

If you add an infinite number of edges to a finite [INAUDIBLE] vertices, then it will get denser and denser and denser and denser. And this is sort of a byproduct of treating these as 0 and 1's. And not recognizing. So this just naturally occurs through sampling.

The linear sampling here is where, basically, I'm taking whole rows at a time. Or I could have taken whole columns at a time. So I'm taking each row and eventually, all right, I'm dropping them down. And there you say it's constant. So I'm basically for each-- I'm essentially taking a whole vertex and adding it at a time. And here the sampling is somewhat independent of it. The densification-- if you sample whole rows, the density of that will be the same as if you did it. So this is just good to know. And good to know about sampling and these phenomena can take place, and then how sampling can play an important role in the data that we observe.

Another phenomenon that's been studied extensively is what happens to the slope of the degree distribution as you add data. And again, we do this exact same type of sampling. And you see here that when we just-- this data had a slope, I believe, of 1.3. And you can see if we sample it randomly, just take random vertices, that the slope starts out very, very high. And each sample stays high. But when we start accumulating them, they start converging on the true value-- converging from above.

And likewise, when you do linear sampling, you have a direction in the opposite way. And they both end up converging onto the true value here of 1.3. And so again, this just shows that the

slope of your degree distribution is also very much a function of the sampling. It could also be a function of the underlying phenomenon. But again, a cautionary tale that one needs to be very aware of how one's sampling. And again, these perfect power law data sets are a very useful tool for doing that.

So if you have a real data set and you're sampling in some way, and you want to know what is maybe real phenomenon versus what is sampling effects, if you go and generate a perfect power law that's an approximation of this data set, you can then very quickly see which phenomena are just a result of sampling a perfect power law, and which phenomena are maybe indicative of some deeper underlying correlations between the data. So again, a very useful tool here.

Moving on, we're going to talk about subsampling. And one of the problems that we have is very large data sets. And often we can't compute the degree distribution on the entire data set. Or we can't-- and this has sort of been a bread and butter of signal processing for years, where if we want to compute a background model, that we don't just simply sum up all the data. That we randomly select data from the data set, and we use that as a model of our background. And that's a much more efficient way, from a computational and data handling perspective, than simply computing the mean or the variance based on the entire data set.

So again, we need good background estimation in order to do our anomaly detection. And again, it's prohibitive to traverse the data. So the question is, can we add accurately estimate the background from a sample? So let's see what happens. We have a perfect power law. We can look at what happens when we sample that.

So we've generated a power law. OK. And this is not-- I've changed-- this may look like the degree distribution, but it's actually a different plot. So this is showing every single vertex in the data set. All right. And this shows the N degree of that vertex. And we've sorted them. So the highest degree vertex is over here and the lowest degree vertex is over here. OK. So this is all the vertices.

And so this is the true data. And this is what happens when we take a 1/40 sample. I just say, I'm only going to take 1/40 of the edges. What does it look like? Some relatively simple math here, which I won't go over. But it's there for you. We can actually come up a correction that allows us to sample that data based on the median distributions. I apologize for the slides. We will correct them for the web when we go out there.

All right. So moving on. So one of the things we talk about sampling, and we talk mainly about single distributions. We can also talk about joint distributions. So we can actually use the degree as a way of labeling the vertices and look at them. So it's a way of compressing-- if we just say, we're going to label each vertex by its degree, this is a way of compressing many vertices into a smaller dimensional space way to look at that.

And we can then count the correlations. We can look at the distribution of how many edges are there from vertices of this degree to vertices of that degree? And so it's a tool for projecting our data and understanding what's going on. And we can also then re-bin that data with a power law, which will make it more easily understood.

So if we look here, we see that we had the degree distribution. So this shows us for a perfect power law data, for vertices with this degree in and this degree out, how many edges there were between them. And as you see here, obviously, there was a lot of vertices here between low degree edges, and not so much here. But this is somewhat a misnomer because of the way the data comes out. We're not really binning it properly.

However, if we go and fit a perfect power law to this data, and then pick a new set of bins based on those, so re-bin the data, we can see here that we get a much smoother uniform distribution. So while here we may have thought that this was an artificially low dense-- not dense region, this was artificially high-- what you see here is when you re-bin this that there's a very smooth distribution from what we expect for our perfect power law here. That this is a fairly uniform distribution with respect to that. And so basically, it essentially puts more bins where we actually have data as opposed to wasting bins over here where we don't really have any data.

Using our perfect power law model, we can actually compute analytically what this should be. So here's an example of what that looks like. And again, very similar to what we see. And given the data, and a model for the data, we can then compute the ratio of the observed to the model to get a sense of what data is unusual versus what we expect from a perfect power-- or linear fit. And we see that very clearly here.

So this is the data just dividing the data by the model here. And you see again, you see all this data appears like this is the ratio, the log of the ratio. And so it's a-- zero means that it's essentially the ratio is 1, so all of this is expected. And then we see all these fluctuations here. Things that are higher than we expected and things that are lower than we expected.

And so this is the classic-- the time where you see this most is whenever they show you a map of the United States by county of some-- maybe like it's-- a classic is like a cancer cluster, or heart disease, or any phenomena. And what you see is that certain counties in the western part of the United States are extremely healthy, and certain counties are just deadly. And it's just a fact that they're very sparsely populated. And so they are dealing with small numbers effect here.

So basically, this is just showing you oscillations between 0 and 1 here. So what we call Poisson sampling. And so it makes it very difficult to know what those are. However, if we re-bin the data and then divide by the model, we see that the vast majority of our data, as expected, is in this normal regime.

Another thing we can actually do is we can look at like the most unexpected data set. So we can look at the most typical data set. Oh, these got moved here, didn't they. I don't know what's going on with PowerPoint today. But this shows us the surpluses, the deficits, and the most typical.

So here's like the most extreme bin, the most underrepresented bin, and sort of the most average bin. And these are the areas that they correspond to in the real data set. So you can use this to find extremes based on a statistical test. And again, you see that here again. We have these different-- that shows what the measured over expected is versus measured. And you can see, this is the original data set and this is the re-bin data set. And you see that the re-binning removes a lot of these very sparse points over here and gives you very narrow distribution around what you expected.

You can actually go and find selected edges if you want. So this just shows the different types of edges. The maximum. If you actually wanted to go and look at those edges, these would be maximum, these would be the minimum, and these are the other types. So it's a useful thing. You can, again, go, say all right, we found that if we have an artificially high correlation between vertices with this degree and this degree, we can then backtrack and find out which specific vertices there are, and see if that's anything that's interesting.

We can also use this plot to look at these questions of edge order. And so, hopefully, this will work today. So here's is a-- if I randomly select vertices and I compute their degree versus-- so basically, what we did before. We subsample and then compute over what is expected. And

we play this. And you can just see here we get this sort of-- when we randomly select the vertices, we basically get kind of a typical-- each sample looks very much the same.

Again, up there in these high degrees, we have this Poisson sampling effect where we have some-- this is we have no vertices, so we get a 0, which is lower than expected. And if we have like two vertices, we get higher than expected. And so again, we still have that Poisson effect.

Interestingly, if we do linear sampling, which we take whole rows at the time, we get a very different type of phenomenon. So you see you get these-- whenever you do run into a high degree row, by definition, it is unusual. Which again, means you have to be very careful. You're going to run into this high degree row eventually by sampling, but you have to be careful. It's like, oh, my goodness. This is a very, very unusual type of thing. So again, cautionary tale about sampling.

All right. So we've talked a lot about the theory here. Let's get into some real data. So this is our Reuter's data again. I showed it to you the other day. The various document distributions we had. In this case, 800,000 documents, 47,000 extracted entities, for a total of 6,000,000, essentially, edges. So it's a bipartite graph [INAUDIBLE] between documents. And four different types. And we can now look at the different degree distributions of the different classes and see what we have.

So the first one we want to look at are the locations. And so we look at the distribution of the documents and the entities. So basically, imagine we took-- so we're very clear here. So what I'm doing is just taking this part of the matrix, and the distribution of the documents is summing this way. And the distribution locations is summing up and down along the columns.

So for each one of these types, we can do those different types. We have, essentially, two different degree distributions. One associated with the documents and one associated with the entities. So this shows us our document distribution. So we have the measured data. We have our fit. We have our model, and then our re-binning. And you know, you could say that this is approximately a power law, and that when you re-bin it, that this sort of S-shaped effect is still there, which probably means it's really there. There's something going on in the data that really is making this bowing effect. And so that's really there.

Likewise here, we have our model. We have the measured data. We have our model, alpha, which is the blue line. We have the model fit. And then our re-binning. And again, you could

say the power law model's a pretty good fit, but again, we have something going on here. Some kind of phenomenon going on there.

And we can then do this for each type. We can look at the organizations. And again, we don't have as many organizations. Again, similar types of phenomenon here. This is so sparse, it's difficult to really say what's going on here.

Of course, we do people, which is, of course, people always are the first thing that you talk about when you talk about power law distributions. And again, very nicely, we have our measured data and our fit. And again, we see this sort of bent broom distribution here, bending, and then with this fan out effect here. But then when we model it and re-bin it, we get something that looks much more like a true power law. And you can see that very nicely here with the actual person data. A very good power law. So this tells us that regardless of what this underlying data looks like, this probably really is a power law distribution.

And then we have the times. And again, a similar type of thing here. Different types of things. And we actually have a little spike here. The Reuter's data has a certain collection of times associated with the actual filing of events. There's only 35 of them, so we do get a little spike here, which is actually what we expect. Although you wouldn't see that really clearly in the true data, but when you re-bin it, this bump comes out fairly clearly.

So again, proper binning extremely important. We can look at our correlations as well. So let's just look at the person-person correlations. And again, what we saw here is sort of-- we see this is the raw data. So something that looks very much like a power law. But when we go through our re-binning process, we see kind of this witch's nose effect here really in the data. So this would tell you to first order it's a power law, but you actually have this correlation taking place here, right here. So that's something that really seems to be taking place.

You see the same thing when we do time. We can look at document. Let's now look at sampling. So again, this is the same sampling that we did. We're going to now basically look at the document densification. So this is selecting whole rows. We select a whole document, so we're getting a whole row. And this just shows you the four different types here. And you see that they behave exactly as expected. Each individual sample is reflective of the overall densification, because you're taking whole rows. OK?

Now let's take entities. So now we're cutting across these. And now you see something that looks much more like random sampling. When you randomly select an entity, that's sort of a

random set of documents. And again, the individual samples, but when you start summing them up, you see that they get denser and denser and denser. So individual document has a sort of a good sample of the overall distribution. When you pick a person, as you take a higher fraction of the data, you're going to get more and more and more with that.

So again, all consistent with what we saw before. A little noisier to see here, but trust me, the power law exponent also behaves exactly as we expected. Likewise, this is essentially a linear sampling, and here's the random sampling behaving exactly as we expected.

We can look at the joint distributions. So here's the actual-- this is the location cross-correlation. So they're showing you basically the document versus entity degree distributions. So this is the measured. And it's been re-binned. This is measured divided by the expected, so the expected re-bins. This is measured divided by the model. And here's the model, and expected divided by the model. And you can basically compare all these different types to create different statistical tests.

And actually, we can find here the measured re-binned divided by the expected re-binned. We can get our surpluses and deficits and other types of things in the actual data. And so here you see something that maybe looks like an artificially high grouping here, some artificially low, and some expected. And you can then use these as ways to actually go and find anomalous documents. Or documents are like-- you say, look, this is the most typical type of thing. I mean, people are like, well, why would you try and find the most normal?

Well, a lot of times, you want to be able to, in terms of summarization, be like here's a very representative set of documents. They have the statistical properties. This is very consistent with everything else. Because people will [INAUDIBLE] like what that mean to be that? So again, a very useful way to look at the data.

Do the same thing with organization. Again, basically get the measured re-binned, the measured divided by the expected, the expected re-binned, the model, refit the model, and the various different ratios of all of them, which you can do to find various outliers and such. Again, a very useful. Interesting that it picked the most representative one way up here. So that's an interesting-- so here's a very unusual representative sample.

Persons, you can do the same thing. And I guess one of the things I'm also trying to do is that you see that these all don't look the same. Right? Hopefully, from looking at these, it's like this

one, the location distribution is clearly a little similar to the organization distribution, but the person distribution looks very different. And the time distribution looks very different. So you see that you have to be careful about-- sometimes we'll just take all these different categories and lump them together in one big distribution.

And here's a situation, like these are really pretty different things. And you really want to treat them as four distinct classes of types of things. So as an example of selected edges, this just shows the most typical. This just shows the various entities that were selected, and very representative document. And this is a very low degree, various entities. So this might show you entities-- here's the kind of surplus example. And we could go in and actually finding those edges.

Same with the person. Very generic people, higher degree. Here's a very low. So this person, Jeremy Smith, is very unusual in terms of what they connected with. Surplus ones here. Can't really read that over there, but just to give you examples of things that you can actually use this to go in and actually find things.

All right. So that brings again to the lecture part. So just, again, developing this background model's very important for the graphs. Based on the perfect power law, gives us a very simple heuristic for creating a linear model. We can then really quantify the effects of sampling, which is very important.

Again, traditional sampling approaches can easily create nonlinear phenomenon that we have to be careful of and be aware of. It allows us to develop techniques for comparing real data with power law fits that we can then use as statistical tests for finding unusual bits of data. It's a very classic detection theory here.

Come up with a model for the background. Create a linear fit for that background. Then use that model to then quantify the data to see which things are unusual. Again, just using very classic detection theory, the techniques. Again, having a background model being the linchpin of doing that.

And I'm not saying that this-- this is very recent work. This parallel model-- something we did in the last year or so. We just published it this summer. I can't guarantee that in three or four years from now that people will still be using this model, because it is very new. But I think this is representative of the kinds of things that will be in three or four or five years, that people-- it may not be this, but something like this to characterize our data. And so I think it's very useful

there.

All right. So with that, we will take a short break and then we will show the example code and talk about the assignment. So very good. Thank you very much.