

## ConvexOptimizationII-Lecture15

**Instructor (Stephen Boyd):** All right, I think this means we are on. There is no good way in this room to know if you are – when the lecture starts. Okay, well, we are down to a skeleton crew here, mostly because it's too hot outside. So we'll continue with  $L_1$  methods today. So last time we saw the basic idea. The most – the simplest idea is this. If you want to minimize the cardinality of  $X$ , find the sparsest vector  $X$  that's in a convex set, the simplest heuristic – and actually, today, we'll see lots of variations on it that are more sophisticated. But the simplest one, by far, is simply to minimize the one norm of  $X$  subject to  $X$  and  $Z$ .

By the way, all of the thousands of people working on  $L_1$ , this is all they know. So the things we are going to talk about today, basically most people don't know. All right. We looked at that. Last time we looked at polishing, and now I want to interpret this – I want to justify this  $L_1$  norm heuristic. So here is one. We can turn this – we can interpret this as a relaxation of – we can make this a relaxation of a Boolean convex problem. So what we do is this. I am going to rewrite this cardinality problem this way. I am going to introduce some Boolean variables  $Z$ . And these are basically indicators that tell you whether or not each component is either zero or nonzero.

And I'll enforce it this way. I'll say that the absolute value of  $X_i$  is less than  $RZ_i$ . Now  $R$  is some number that bounds, for example – like it could be just basically a bounding box for  $C$ , or it can be naturally part of the constraints. It really doesn't matter. The point is that any feasible point here has an infinity norm less than  $R$ . If we do this like this, we end up with this problem. This problem is a Boolean convex. And what that means is that it is – everything is convex, and the variables, that's  $X$  and  $Z$ , except for one minor problem, and that is that these are 0/1. Okay? So this is a Boolean convex problem. And it's absolutely equivalent to this one.

It is just as hard, of course. So we are going to do the standard relaxation is if you have a 0/1 – 0, 1 variable, we'll change it into a left bracket 0, right bracket 1 variable. And that means that it's a continuous variable. This is a relaxation. And here, we have simply – we have actually worked out, this is simply – well, it's obvious enough, but this is simply the convex hull of the Boolean points here. Now if you stare at this long enough, you realize something. You have seen this before. This is precisely the linear program that defines – this is exactly the linear program that defines the  $L_1$  norm.

So here, for example, this is at norm  $X$  is – it's an upper bound on –  $Z_i$  is an upper bound on one over  $RX_i$ . And so, in fact, this problem is absolutely the same as this one. And so now you see what you have. That Boolean problem is equivalent to this. By the way, this tells you something. It says that when you solve that  $L_1$  problem, not only do you have – is it a heuristic for solving the hard Boolean problem, it's says it's a relaxation, and you get a bound.

The bound is this: you have to put a one over  $R$  here, where  $R$  is an upper bound on the absolute value of any entry in  $C$ , or for that matter any entry that might – is a potential

solution or something like that. And this tells you that when you solve this  $L_1$  problem not only do you get – is it a heuristic for getting a sparse  $X$ , but in fact it gets you within a factor of  $R$  a lower bound on the sparsity. So that's that. By the way, it's a pretty crappy lower bound in general, but nevertheless it's a lower bound. Okay. Now we can also interpret this in terms of a convex envelope. And let me explain what that is.

If you have a function,  $F$ , on a set  $C$  – and we'll assume  $C$  is convex, so on a convex set  $C$  – in fact, I should probably change this to convex set. This function is not convex. The envelope of it, it is the largest convex function that is an under-estimator of  $F$  on  $C$ . And let me just draw a picture, and we'll – I'll show you how this works for the function we are interested in. The function that we are interested in looks like this: it's zero here, and then it's one over here. So that's our indicator function. And, if you like, we can do this on the interval, plus one minus one, okay. And so our function looks like this, basically.

That's our cardinality function. And what we want to know is this. What is the smallest – sorry, the largest convex function that fits everywhere beneath this function on that interval? By the way, well, I can leave it this way. So the simplest – any convex functions, it's got to go through this point; it's got to go through that point; and therefore, this gives you – that's it. So no one would call – no one would call this absolute value function a good approximation of this function, for sure. But it does happen to be the largest convex function that's an under-estimator of it, okay. So that's that.

And then, actually, you can go like this, if you like. Because it's on this – it's just on this one set. So that's the convex envelope. Now we can relate to all sorts of interesting things. I mean one is this that the – one way to talk about the envelope of a function in terms of sets is this. You form the epigraph of the function, and then simply take the convex hull. And it turns out that's the epigraph of the envelope. And you can see that over here, too. So the original function has an epigraph that looks like this. It's all of this stuff, and then this little one tendril that sticks out down there. So it's everything up here, and one little line segment sticks out there.

Convex hull of that fills in this part and this part. And what you end up with is the absolute value restricted to plus minus one. So that's going to be the convex envelope here. And another interesting way to say it is this, it is, in general, it's  $F^*$ . And I don't want to get into technical conditions, but – actually, the conditions aren't that big of deal, it's just that it should be closed or something like that. Actually, this function here is not closed so – but anyway, it looks like this. It's the conjugate of the function, which is always convex, and then that starred. So it's the conjugate of the conjugate.

So now if the function  $F$  were closed and convex originally, this would cover  $F$ . That's kind of obvious. Okay. So for  $X$  scalar, absolute value of  $X$  is the convex envelope of  $\text{card } X$  and minus one one. And if you have a box of size  $R$ , an  $L$  infinity box here, then  $\|X\|_1$  is the convex envelope of  $\text{card } X$ . So that gives you another interpretation. So if someone says, "What are you doing?" You say, "I am minimizing the  $L_1$  norm in place of the cardinality." They can say, "Why?" You would say, "Well, it's

a heuristic for getting something sparse.” And they go, “Well, that’s not very good. Can you actually say anything about it?”

And you go, “Actually, I can. One over  $R$  times my optimal value is a lower bound on the number of nonzeros that any solution can have.” Okay, by the way, if you understand this, you already know something that most of the many thousands of people working on  $L_1$  don’t know. You are actually now much more sophisticated. And I’ll show you, and it’s going to be stupid, but it’s actually completely correct. Suppose for some reason I told you that  $X_1$ ,  $Y$  is between one and two, so it lies here. What is the convex envelope of that? Well, it’s this. It looks like that. And what you see is that the function is now no longer an absolute value.

It’s asymmetric, okay. So it’s asymmetric. You can write out what it is. Would it be a better thing to do if you wanted to minimize the cardinality? In this case, the answer is absolutely it would be better. It would work better in practice. In theory, of course, it’s better because it gives you the actual convex envelope and so on. So what that says is that when you minimize the one norm of  $X$  as a surrogate for minimizing cardinality, you are actually making an implicit assumption. The implicit assumption is that the bounding box of  $X$  – of your set – that the bounding box is kind of – it’s like a box. It’s a uniform, right.

All of the edges are about the same, and they are centered. If you ever had a problem where you are minimizing cardinality and  $X$  is not centered, like for example, some entries lie between other numbers, you would be using a weird, skewed, weighted thing like that where you have different positive and negative values. Oh, what if I told you that  $X_2$  lies between, for example, between two and five? What can you say then? Then  $X_2$  is not a problem because  $X_2$  will never be zero. So it’s just – it’s a non – then it’s easy. Okay. Well, we just talked about this.

If you had a – if you knew a bounding box like this with an  $LI$  and a  $UI$ , then – and by the way, you can find bounding box values very easily by simply maximizing and minimizing  $X_i$  subject to over this set. So you can always calculate bounding box values. Now if the upper bound is negative or the lower bound is positive, then that’s stupid because that means that  $X$  has a certain sign, and there is no issue there. It is a non-issue. If they straddle zero, that means there is the possibility that that  $X$  is zero. And in that case the correct thing to do is to minimize this. If these things are equal here, that reduces to  $L_1$  norm minimization.

So that’s what that is. This will also give you a lower bound on the cardinality. Okay. So let’s look at some examples. I think we looked at this last time briefly. I’ll go over it a little bit better this time, or we’ll go over it in a little bit more detail. This is a regressor selection problem. You want to match  $B$  with some columns of  $A$ , linear combinations of columns of  $A$ , except I am telling you, you can only use as many as  $K$  columns here. So, okay, so the heuristic would be to add, for example, a one norm here and adjust  $\lambda$  until  $K$  has fewer than  $K$  nonzeros.

And then what would happen is – you'd look at this value of that then. So here is the – here is sort of a picture of a problem. It's got 20 variables, 2 to the 20 is around a million. And therefore, you can actually calculate the global solution. You can do it by branch and bound. We are going to cover that later in the quarter, but you can also – in this case, you just work out all million. So one million least squares problems, you'd check all possible patterns, and not a million. Yeah, yeah, this is a million. You just solved a million of them and for each one. So the global optimum is given here, like this.

And this one gives you this – the one obtained by the heuristic. And you can see a couple of things here. It looks to me like you never – I am not quite sure here, but I think for most of it you are never really off by – well, no, here you are off by two. That's a substantial error. You are off by one; sometimes you are exactly on and stuff like that. But the point is that this curve is obtained by the heuristic, which was one-millionth the effort there. Okay. So now we'll look at sparse signal reconstruction. It's actually the same problem, different interpretation. You want to minimize norm  $AX$  minus  $Y$ .

$Y$  is a received signal.  $X$  is the signal you want to estimate. The fact that there is a two norm here, this might be, for example, that you are doing maximum likelihood estimation of  $X$  with a Gaussian noise on your measurement, so you have your  $Y$  equals  $X$  plus  $V$ . Then this is prior information that the  $X$  you are looking for has no more than  $K$  nonzeros. So that's the – that's the other. The other one, the heuristic would be to minimize this two norm subject to norm  $X$  in one less than beta. In statistics, this is called LASSO here. I can't pronounce it – you have to pronounce it with Trevor Hastay's charming South African accent.

I tried to learn it last time I went over this material, but I never succeeded. So I'll just call it LASSO. Okay, so that's this thing. And another form is simply to add this as a penalty and then sweep gamma here. And in this case it's called basis pursuit denoising or something like that, so. And I can explain why it's basis pursuit. If you are selecting columns of  $A$ , you think of that as sort of selecting a basis. So this, I guess – don't ask me why it's called basis pursuit, but that's another name for it. Okay. Let's do an example. It's actually – when you see these things, they are actually quite stunning.

And they are rightly making a lot of people interested in this topic now; although, as I said earlier, the ideas go way, way, way, way back. So here it is. I have a thousand long signal. And I am told that it only has 30 nonzeros, so it's a spike signal. Now just for the record, I want to point out that a thousand choose 30 is a really big number. Okay. Just so the number of possible patterns of where the spikes occur is very, very large. For all practical purposes, it's infinitely large. And here is what's going to happen. We are going to be given 200 noisy measurements. And we'll just generate  $A$  randomly.

And there will be Gaussian noise here, okay. And then you are asked to guess  $X$ . Now, by the way, I should point something out. If someone walks up to you on the street and says, "I have a thousand numbers I want you to estimate. Here are 200 measurements." You should simply turn around and walk the other way very quickly, okay. Get to a

lighted place or something like that as soon as you can, or a place with other people. The reason is it doesn't – this is totally ridiculous, right.

Everyone knows you need a thousand – if you are going to measure a thousand signals, you need a thousand measurements, okay – so at least a thousand, right, and better off is 2,000 or 3,000 or something like that to get some redundancy in your measurements, especially if there is noise in it. But the idea that someone would give you one-fifth the number of measurements as you have data to estimate and expect you to estimate it is kinda ridiculous, okay. Now, by the way, the flip side is this. If someone told you which 30 were nonzero, you move from five-to-one more parameters than measurements to the other way around.

If I tell you that there is 30 numbers I want you to estimate, and I give you 200 measurements, now you are 8-to-1 in the right direction, or you are 7-to-1 in the right direction. In other words, you got seven times more measurements than – everyone following this? Okay. So, all right, so what happens is if you simply give this  $L_1$  thing – you just – well, you can see in this case you just recover it like – I guess it's perfect. I mean it's not completely perfect, some of these. I think the sparsity pattern is maybe perfect. It looks to me like it's perfect. Yeah, if it's not perfect, it's awfully close.

By the way, what that means is if you polish your noise will go down lower, and you'll get these very close. You won't get it exactly because you have noise. But the point is this is really quite impressive. If, in contrast, you would use an  $L_2$  reconstruction, a [inaudible], and you would solve this problem, you can adjust gamma – basically, it never looks good, the reconstruction. But this would be an example of what you might reconstruct, something like that. So this is the rough idea, okay. So these are actually pretty cool methods.

I mean I don't really know any other really sort of effective method for doing something like this, for having – for saying, look, here is 200 measurements of a thousand things. Oh, and here is a hint, prior information. The thing you are looking for is sparse; please find it. And these work. I should also mention, unlike least squares. Least squares is kind of nice. It's a good way to blend a bunch of measurements and get a very good – it can work beautifully well if you have got like 50 times more measurements than variables to estimate. You use least squares. Almost like magic, it's a 263 level, right.

All of a sudden, from all of these crazy [inaudible] with noise, out comes like a head that you are imaging or something like that. I mean it's really quite spectacular. And it kind of fails gracefully. I should add something here. These don't fail gracefully. And I bet you are not surprised at that. So if you take this – which you can, all of the source is on the web, everything – and you just start cranking up sigma. You crank it up, and up, and up. And it will work quite well up until some pretty big sigma – you'll give sigma one more crank up and, boom, what will be reconstructed will be just completely – it will go from pretty good reconstructed to just nonsense very quickly.

So I just thought I'd mention that, so. Somehow it's not surprising, right. Okay. Let me mention some of these theoretical results. Obviously, I am going to say very little about it. They are extremely interesting. In fact, just the idea that you can say anything at all about it I find fascinating. But here it is. The problem is going to be that the set  $C$  is going to be very embarrassing. It's going to be an affine set. So basically, suppose you have  $Y$  equals  $AX$ , where the cardinality of  $X$  is less than  $K$ . And you want to reconstruct  $X$  here. Now, obviously, you need – the minimum you could possibly have would be  $K$  measurement.

So in other words, if someone comes up to you and says, "Wow, that's good. You have got my 30 non – my spike signal with 30 things. What if I give you 19 signals?" That's not even – that's not enough to get 30. So on the other hand, if the number of measurements is bigger than the size of the number of parameters, then we are back in 263 land, and everything is easy to do. So that's trivial. So the interesting part is where  $M$  lies between  $K$ , that's the sparsity – known sparsity signal and the number of parameters. And the question is: when would the  $L_1$  heuristic that's minimizing norm  $X_1$  subject to  $AX$  equals  $Y$ .

I mean and notice how simple the set is,  $C$  if the set of  $X$  –  $AX$  equals  $Y$ . When would it be constructed exactly? That's the question. And actually, this – you can actually say things which are quite impressive. And it basically says this. It says that depending on the matrix  $A$  here – but there is a lot of matrices. Actually, there is a long – there is a lot of matrices that would actually work here. And there is actually all sorts of stuff known but what exactly what it is about the matrix that does the trick; it has to do with coherence or something like that.

And it says basically that if  $M$  is bigger than some factor times  $K$  – so this is sort of – if I gave you the hint, if I told you what the sparsity pattern is, you would need  $M$  bigger or equal to  $K$ . So the extent to which this number goes above one is how much more you need than the minimum if you had the secret information as to what the sparsity pattern was. And this is an absolute constant  $C$  times  $\log N$  here. Then it says if this works, then the  $L_1$  heuristic will actually reconstruct the exact  $X$  with overwhelming probability. What that means is that as you go above this, actually the probability of error goes down exponentially. Okay? That's it.

And some valid  $A$ 's would be this. If the entries in the matrix were picked randomly, that would do the trick. If  $A$  was a rows of a DFT matrix, of a discrete 48 transform, then it would work. As long as the rows are not like bunched up or something like that, then it would work. And there are lots of others. So this is it. And these are beautiful things. I would take you about four seconds to find these on the web with Google, to find references and just get the papers. So, okay, so we will go on to the second part of this lecture, and that is going to be here. There we go. Great. Okay. So we are going to look at some more advanced topics here, so – and just other applications, and variations, and things like that.

So one is total variation reconstruction. I should add that this predates the current fad. The current fad,  $L_1$  fad, it depends on the field. I mean statisticians have been doing it for a long time, like 12 years or 15 years now. People in geology, I think, have been doing it for 20. Others have been doing it probably 20 or something like that. But the recent thing, actually, was spurred by these results. And that's in the last five years, let's say. But total variation, this goes back, I believe, easily to the early '90s or something like that. So it works like this. You want to fit a corrupt – you have a corrupted signal, and you want to fit it with a piecewise constant signal with no more than  $K$  jumps.

Well, a simple way to do that is to trade off –  $\hat{X}$  is going to be your estimate of your signal. You trade off your fit here – by the way, if this were Gaussian noise that would be something like a negative log likelihood function here. You would trade off your fit with the cardinality of  $DX$  where  $D$  is the first order difference matrix. And what you do is you would vary  $\gamma$ . If you made  $\gamma$  big enough, the solution  $X$  is constant. And then, of course, it's equal to the average value. As you crank  $\gamma$  down from that – from that number, what will happen is the  $\hat{X}$  will first have one jump, so it will be piecewise constant with one jump, then two, then three, then so on and so forth. Okay.

So  $DX_1$ , by the way, is the sum of the differences of the absolute values of a signal – this is scalar signal for now. That's got a very old name. It goes back to the early 1800's. That's called a total variation of a signal – of a function – a signal, that's fine. And this is called total variation reconstruction. And there is a lot of things you can say about TVR reconstruction, but what happens is they actually are able to remove sort of high frequency noise without smoothing it out. We'll see how that works, like an  $L_2$  regularization will just give you a low-pass filter; it will smooth everything out.

Now these, they are very famous here because they didn't – these were some of the methods do things like recover. These original from the wax recordings of Caruso or something, they actually reconstructed them. They got all sorts of jazz stuff from the '20s and reconstructed them using these methods. And they are amazing. I mean sort of the clicks and pops just go away. I mean they are just – they just – they are just removed. Okay. So here is an example. And so here we have a signal that looks like this. It's kind of slowly moving, but it's got these jumps as well. I mean this is just to make a visual point.

There is a signal, and the corrupted one looks like this. So there is a high frequency noise added here. Okay. And if we do total variation reconstruction, these are three values of  $\gamma$ . And they are actually chosen – one is supposed to be like too much; one is too little; and one is not enough. But you can see something very cool. The jumps are preserved. So – and that's not like – that's not smoothed out at all. That's jump – that's smoothed out perfectly. Okay. And this is sort of too much because I have actually sort of flattened out the curvature that you saw here.

This is maybe you haven't removed enough of the signal, but you still get this jump here. And this might be just enough. By the way, if you were listening to this – in fact, I should probably produce some like little jpeg – I mean some little audio files or something like

this so you can hear total variation denoising. It's very impressive. L\_2 denoising, in a minute we'll see that, that's just low pass filtering. You have a lot of high-frequency stuff with everything just muffled. You hit a drum, everything is muffled. The L\_1 [inaudible] will actually cut out this high-frequency noise.

But when someone hits a snare drum, it's right there. So it's pretty impressive. We should – it would be fun to do that, actually. Okay. Here is the L\_2 reconstruction, just to give you a rough idea of what happens. In L\_2 reconstruction, to really smooth out the noise, basically, you lose your big edge here. And this is sort of, maybe, the best you can do with L\_2 or the best trade-off. And this would be – if you still – by the way, you still – in this case, it's not – that has not been preserved exactly. It's actually been smoothed a little bit. You just can't see it here. And you still are not – you are not getting enough noise attenuation, so just get a picture. Yeah.

**Student:**[Inaudible] that even this sounds very, very good.

**Instructor (Stephen Boyd):**This one?

**Student:**Yes, L\_2 because that's the one we didn't do extraneous [inaudible].

**Instructor (Stephen Boyd):**Yes.

**Student:**This one sounds very good. And why should we go the extra half if we are going for L\_1? I mean [inaudible].

**Instructor (Stephen Boyd):**Oh, in cases – I mean to do things like remove clicks and pops. And then, if you started listening carefully, you would find out this did not sound good at all. I mean not at all.

**Student:**Okay.

**Instructor (Stephen Boyd):**Yeah. Because you'd either hear this noise, right, or you start muffling this. And that makes a drum sound like – then you are not tapping a drum; you are tapping like a pillow or something like that. And it's no longer a drum. I mean just – so that's the – if you listen to these things, it's quite audible. We can adjust the parameters so the 263 methods work well, which of course naturally we do in 263. Wouldn't we? So, okay. So, okay. But actually, the total reconstruction variation is really done more often for images. And I believe it's also done even for – in 3-D. And I believe it's done even for 4-D, so for 3-D movies with space-time.

But we'll look at it in 2-D, and it's quite spectacular. So here is the idea. And this is going to be very crude. And I'll make some comments about how this works. So what it is, you have X and RN. These are the values on a N by N grid. So our R grid has about a thousand points on it. I mean it's small, but that's it. And the idea is I want – here is a prior knowledge is that X has relatively few – so X is sort of piecewise constant. In other words, it's like a big region. It looks like a cartoon. It's got big regions where it's

constant and with a boundary. So everybody see what I'm saying? So that's the – it's cartoon looking.

It looks like a cartoon, or a line drawing, or whatever. All right, now this problem. You get 120 linear measurements; that's, of course, a big joke. You are whatever that is, six or seven times – I guess you are seven times under or six, whatever it is. You're some big factor under here, eight maybe that is, I don't know, eight. So you are eight times under sample. In other words, I want you to estimate 960 numbers; I'll give 120 measurements. These are exact. These are exact. So the way we'll do this is we'll say, look, among – this has, among all of the  $X$ 's that are consistent with our measurements, that's a huge set.

In fact, what's the dimension of the set of  $X$ 's that satisfy this? What do you think? What's the dimension on that? You don't have to get it exactly, just roughly. What 840 dimensions? Yeah. You got – you get 961 points. You get 120 measurements, null spaces on the order of the – is the difference, right. So, I don't know, you have eight. So this is a huge number of  $X$ 's are consistent with our measurements; 840 dimensional set of images are consistent with our linear measurements. But among those, what we'll do is to pick one we'll do this. We would like to minimize – this is the sum of the cardinalities of the differences, and let me show you what that is over here.

And I'll explain in a minute how to make this better – look better, anyway. It's this that we have our grid. And basically, we would – we are going to charge for the number of times two edges – two values are different. And that's both this way and this way. So, for example, that big objective would be zero if the entire image were constant. Otherwise, everywhere where there is sort of a boundary, you are going to get charged, okay. So that's the picture. Now we can't solve that problem, but we can solve this variation on it.

Now, by the way, when you do  $L_1$  this way on an image, and you just go – you charge for this way and this way, what happens is you are going to tend to get images – or you'll get things that will – they actually prefer like this direction or this direction, and you get weird things. I think we had that in a maybe a homework – no final – was it final exam problem 364? Was it? I can't remember. I think it was. We made Jacob's happy face. Midterm? Final? Midterm.

**Student:**[Inaudible].

**Instructor (Stephen Boyd):**What?

**Student:**Homework.

**Instructor (Stephen Boyd):**Homework. Okay, homework, sure. Anyway, so okay, so let's see how this works. So here is the – here is the total variation reconstruction. And the summary is it's perfect.

**Student:**[Inaudible].

**Instructor (Stephen Boyd):** Yeah. I know.

**Student:** I just [inaudible].

**Instructor (Stephen Boyd):** Great. Okay. Good. Good, okay. I forget, too. Wait until you see the [inaudible] 364c.

**Student:** [Inaudible] 364c?

**Instructor (Stephen Boyd):** Oh, yeah. Yeah. We are starting it this summer just for you. You are already enrolled. You'll like it. We're bringing the final exam back on that one, except that it's going to be every weekend, though, 24-hour. But you are learning a lot. You are going to learn a lot, though. Okay. So I – I mean this is – you get the idea. In this case, you recovered it exactly. So I mean these are kinda impressive when you see these things. Variations on this, by the way, are quite real. This is a fake toy example. You can go look at the source code yourself, which is probably like all of ten lines or something.

The plotting, needless to say, is many more lines than the actual code. These are actually quite real things. I mean there is stuff going on now where you do total variation reconstruction MRI from half of this – a third of the scan lines, and you get just as good an image. So, okay, and this is what happens if you do  $L_2$ . And I mean this is what you would imagine it to look like. That's kind of what you'd guess. And you can adjust your gamma and make the bump look higher, or less, or whatever. But it's never going to look that great. That's – this is your 263 method, so. That's essentially a least-normal problem. Yeah.

Actually, I'm sorry, there is no gamma in this problem. That's just least normal. Okay. So that finishes up some of these – oh, I should – I said I promised I was going to mention how these methods actually done. What you really want in image is you really want your estimate of – is to be approximately rotation and variant. So, in fact, you get a much better looking – visually now, if you really do this, not by just taking your differences this way, but you'll also take this difference here as well. And you will – and that thing, you'll divide by square root two or something like that.

And the other, the most sophisticated way, is to take your favorite multi-point approximation of the gradient and take the two norm of it and minimize the sum of the two norms of those gradients. That's the correct analog of total variation reconstruction in an image. And that will give you beautiful results that will be approximately rotation in variant. Okay. So let me talk about some other methods. This is also – this is just starting to become fashionable, the  $L_1$ . And there is other variations on it. And I think people call it beyond  $L_1$  or something like this. And it goes like this.

So one way is to iterate, and I'll give a – we'll give a very simple interpretation of this in a minute. So I want to minimize the cardinality of over  $X$  and  $C$ . So what you do is this is instead of minimizing the  $L_1$  norm, we'll minimize like a weighted  $L_1$  norm. Now we have already seen good reasons to do that. The weights correspond – one over the

weights correspond to your prior about the bounding box. So that's one way to do – one way to justify weights. But the idea here is this. You solve an  $L_1$  problem, and then you update the weights. And the weight update is extremely interesting.

It works like this. If you run this  $L_1$  thing, and one of these numbers comes out zero, this – then you get the biggest weight you can give, which is one over epsilon. What that means is thereafter, it's probably going to stay zero because it went to zero at first. Once you are zero, in the next iteration your weight goes way up. And then there is very strong encouragement to not become nonzero. What's cool about this is if  $X$  turns out to be small but not zero, so you are actually being charged for it cardinality-wise, what this does is it puts a big weight on it. And it basically makes that one look very attractive.

And it basically mops – cleans up small ones, just gets rid of them. On the other hand, if  $X$  came out big, you are taking that as a heuristic to mean something like, well look, if you minimize an  $L_1$  norm and something comes out – one of the entries comes out to be really big, it basically means, look, that thing is not going to be zero anyway. You are not going to drive it to zero. So therefore, relax, and basically says reduce the weight on it. So if it wants to get bigger, let it get bigger. So this is the picture. And this will typically give you modest improvement. Well, I mean it will actually give you real improvement over the basic  $L_1$  heuristic.

And it typically converts in five or fewer steps. By the way, a more sophisticated version, actually, is not symmetric here with the weights. We'll see what the more sophisticated version is, but anyway. So here is the interpretation. So we'll work with a case where  $X$  is bigger than equal to zero. And we'll do that by splitting  $X$  into positive and negative parts where those are both non-negative. And the cardinality of  $X$  then, it's the same as this thing, I mean provided one of those is always zero. And we'll use the following approximation. Instead of – let me show you this. In fact, this kind of the idea behind all of these new methods that are beyond  $L_1$ .

I mean it goes back to – I mean all of these things go back to stuff that is very stupid. By the way, these things are very stupid, and yet it doesn't stop people from writing fantastically complicated papers about it, right, and making it look not stupid. But they are stupid. So let's go to one and stop there. Okay. So here is the function we want. It jumps up and goes like that. Here is our first approximation, not an impressively good – not what you call a good approximation. And so some of the – this one says you replace it with a log one plus  $X$  over epsilon. And, you know, if you allow me to scale it and change various things, that's a function that looks like this.

I'll shift it and scale it, if you don't mind. And it's a function that looks like that. Well, let me just make it go through there. There, okay, so it looks like that. And this little curve at the bottom, that's the epsilon like that. So it looks kinda like that. Okay? I drew it with epsilon exaggerated. I really shouldn't have. Let me redraw it, and it looks like this. And then you have got a little thing like that. That's sort of how you are supposed to see it. Okay? And you are supposed to say, well, yeah, sure, okay. This function here is a way better approximation of this thing than this, okay. What? You don't think it is?

**Student:**It's not [inaudible].

**Instructor (Stephen Boyd):**It's not convex. You are very well trained. Right. I can tell you a story. A student of Abass's went into – they were just talking to Abass. And Abass said, “Yeah, but then you can do this problem and maximize the energy lifetime of this thing and blah, blah, blah, like that.” And the student stepped back. And he said, “Are you crazy?” And Abass said, “No. What's wrong with that?” And the student looked at him and said, “That's not convex.” Then he came and complained to me. He said, “What are you – what are you doing?” So you are right. That's not convex. Okay. But it's okay. Now you are okay. You can handle it.

So in fact, this method is – in fact, not only is it not convex, it's concave. Now if you have to minimize a concave function over a convex set, when we did sequential convex programming, you saw that there is a very good way to do that. And it's really dumb. I mean it's – what you do is you take a certain  $X$ , you linearize this thing at that point, and you optimize. No trust region, nothing. And you just keep going. If you linearize this, basically you get this thing here. This is a constant, and it's totally irrelevant when you linearize. And you actually get this. And in fact, part of that is a constant, too, like this part. Okay.

And in fact, it's the same as minimizing  $XI$  over this. And guess what sequential convex programming applied to this non-convex problem, which is supposed to fit the cardinality function better, yields that exactly. Okay. So this is really an interactive heuristic. Sorry, it is. This is a convex-concave procedure for minimizing a non-convex function versus a smooth function, which is supposed to approximate the – what do you call it. It's supposed to approximate this card function, okay. By the way, there is other – lots of other methods. And I can say what they are. Here is a very popular one. All of these work.

That's my summary of them; lots of papers coming up on all of them. Here is another one. I don't know, here is one; you don't like – let's just do it on the positive. You don't like  $X$ , how about  $X$  to the  $P$  for  $P$  less than one. So these functions start looking like that. And the small – if you make  $P$  really small, they look just like that card function. And by the way, this leads some people to refer to the cardinality as the  $L$  zero norm. Now let's just back up a little bit there. And two of you have already said that, so you can say it. I cannot bring myself to say that because there is no such thing as an  $LP$  norm with  $P$  less than one because it's not convex.

The unit balls look like this. And then I can't even say it. You see? That's what happens if you learn math when you are young. That is not the unit ball of anything. And you should not say that. You shouldn't even – you shouldn't say that. And yet, you will hear people talk about  $LP$  norm with  $P$  less than one. And it's not convex. It's not a norm, blah, blah, blah. So the methods work like this. In fact, you tell me. Let's invent a method right now. How would you – how would you minimize this approximately, and heuristically, and so on? By the way, if you minimize that, you would get a very nice sparse solution, very nice.

How would you do it? You just linearize this thing. And what would you – and what, in fact, would you be doing at each step, and if you did the convex-concave procedure on this guy? You would be solving an iteratively re-weighted  $L_1$  problem. Okay. And the only thing that would change is your weight update would be slightly different from this one. But your weight update would be reasonable. And I always do the same thing. What happens in a weight update is this. Entries that are big, you just say, ah screw it. That's probably not going to be zero, and you reduce the weight. Entries that are small, you crank the weight up.

If that thing is already zero, that's a strong inducement to pin it at zero. If it's small, thought, that's a – that makes that thing a very attractive target for being zeroed out. And that's what drives the cardinality down. Everybody got this? So that's the idea. Okay. It's a very typical example is you want to minimize the cardinality of  $X$  over some polyhedron. And the cardinality drops from 50 to 44, not that impressive. And if you run this heuristic, I guess six steps it converges, actually, after a couple, it will stop out at – no, sorry, let's see. Here we go.  $L_1$  gives you 44. And the iteratively re-weighted  $L_1$  heuristic gets you 36.

The global solution, probably, found for this problem in long time, later in the class, is 32. So just emphasize, again, we are not – we are not actually solving these problems. These are heuristics. But they are fast, and they are good, and so on and so forth. By the way, the fact that you are not solving the problem if the problem is – for example, is rising in a statistical context, I think means it doesn't matter at all. Right? Because it – you don't get a prize for getting the exact maximum likelihood estimate. Maximum likelihood estimate is just – is itself, in some way, it's just a procedure for making a really good guess as to what the parameter is.

And it's backed up by a hundred years of statistics. Okay. If you miss that – and by the way, even if you do perfect maximum likelihood, as any statistician or anybody who knows anything about it will tell you, you are not going to be getting the exact answer anyway. That's just some that – that's just some which asymptotically will do as well as any method could or something like that. That's its only – now by the way, for engineering design, that's a different story, right. You find a placement of some modules on a chip that takes 1.3 millimeters as opposed to – whereas, opposed to 1.6, that's real.

That's unlike the statistical interpretation. But still, okay. So let's look at an example of that. It's a fun example. It's just detecting changes in a time series model. So let's see how that works. We have a two term ARMA model, or I guess people call this – I'm sorry, it's not ARMA – it's AR – I think, actually, people call this AR(2). So it looks like this.  $Y$  of  $T$  plus two is equal to a coefficient times  $Y$  of  $T$  plus one, plus another coefficient times  $Y$  of  $T$  plus a noise, which is Gaussian. Now the assumption is this: these coefficients here are mostly constant. And then every now and then there is a change in the dynamics of the system.

And one or both of those numbers changes, okay. You'll be observing merely  $Y$ . And your job is to actually estimate  $A$  and  $B$ , and in particular, to find where the changes are.

So let's see how that works. By the way, well, it doesn't matter. I mean I was going to make up an application of it. And it basically says the changes could be – could tell you something about a failure in a system or something like that, or a shock in a financial system or economic system, something like that. Okay. So here is what we'll do is we will – given  $Y$ , this is a negative log likelihood term here with some constants.

That's a negative log likely – this is an implausibility term. Because, for example, if you run up a giant bill here, it's asking you to believe that the  $V$  did some very, very unlikely things. That's what this term is. And then here, we add in a – actually, it's a total variation cost. It basically says it penalizes jumps in  $A$  and  $B$  in the coefficients. Now, by the way, if I make  $\gamma$  big enough,  $A$  and  $B$  will be constant. Okay. If I make them zero, then I will make this thing zero because I can adjust my  $A$  and  $B$ , in fact many ways, to get absolutely perfect fit here. Okay. So here is an example.

Here are how  $A$  and  $B$  changed, so  $A$  is this, and then, I guess, I don't know, I can't see now. But let's say if  $T$  equals a hundred, it changes to a new value.  $B$  is here, and a 200 pops up here. So there is three changes. And you can sort of, visually, if you squint your eyes, you can change in the dynamics of the system here that, if you look left of there, you get one kind of dynamics. You can see a little – some – with a little squinting, you can see that the dynamics on the left in between a hundred and 200 looks different. And again, between 200 and 300, well, it helps that have I told you what happened.

But none of the – it's certainly consistent. Now the interesting thing, though, is imagine I hadn't told you this. I don't know that it's that obvious. I mean certainly this doesn't look very much like that. But would you really know that something happened here? I don't know. You could – I could have made the coefficients change in such a way that you would – your eyeball – you couldn't do it. So if you run this total variation heuristic, on the left this is the estimate of the parameters. And I want to point out this thing is already like very good. It's estimated the parameters to be here. It jumps down, and it does some weird thing here.

I can explain that a little bit here. It's some little false positives. That's a false positive where this thing jumps up. Every time this thing jumps up, there is a bunch of false positives in here, and some false positive jumps in here, and so on. But actually, you know what, this is not bad at all. Not bad at all. This is kinda – it's kind of saying that there are weird changes in here and here. If you do the iterated heuristic, you can actually see visually exactly what happens. By the way, this guy is pulled down here because it's charged a lot. It only makes this big shift for which it pays a lot in this objective to make the – to try to make this overall objective small.

But what happens is actually really, really cool. What happens is this. If you iterate it, this difference is really big. And on the next step, it's going to get a less weight. So it basically says, oh, you really want to jump here? I am going to charge you less for the jump at this time step. Here, these little guys – by the way, where it's flat, it says, okay, you don't want to jump at all. I am going to make – I am going to charge you the maximum amount. That's the one over epsilon in the weight. And these little ones, that's

what these  $L_1$  – iterated  $L_1$  heuristics do. They go up, and they clean up things. They just get totally nailed.

And that's the final estimate there. And you can see that it's much, much better. I mean you are actually tracking the parameters very nicely. You might ask why the error here? Why the error here? Why did it miss the time point here? And the answer would be because there was noise. That's why, but it's still – it's awfully good to do this. Okay. It works very, very well. Okay. And our last topic is going to be the extension of these ideas to make matrices and rank. So if you have – if you have cardinality of a vector – that's a number of nonzeros, there is a very natural analog for matrices, and that's the rank.

And by the way, both of these things come up as measures of complexity of something. So in other words, sort of the complexity of a set of coefficients is something like – I mean this is very rough number of zeroes or something. And the complexity of a matrix also comes up a lot, and that's the rank. Now a convex rank problem, that's a convex problem except you have a rank constraint or rank objective, these come up all the time. And they are actually related. If you have a diagonal matrix and the rank of it is the cardinality of the diagonal, that's kind of obvious. But the interesting part is what's the analog of the  $L_1$  heuristic?

And it turns out it's the nuclear norm, which is the dual of the spectral norm or maximum singular value. And it's the sum of the singular values of a matrix. So that's it. It's not simple, but that's what it is. It is the sum of the singular values. And that's the dual of a spectral norm, which you probably didn't know but it kinda make sense. Because somewhere in your mind you should have this map there that associates – well, it should associate LP and LQ, where one over P equals one over Q is one. But particular pairings should be burned into neurons directly. That's two and two, so dual of  $L_2$  is  $L_2$  type thing.

And dual of  $L_1$  is L-infinity; dual L-infinity is  $L_1$ . These you should just know. So it shouldn't be surprising that if it's the maximum singular value, that's like and L-infinity on the singular value, roughly; that the dual norm should be the sum of the singular values. That's it. Now if a matrix is positive semi-definite, then symmetric – positive semi-definite, then the eigenvalues are the singular values, and the sum of the singular values are therefore some of the eigenvalues. That's a trace, okay; whereas, for a vector, if I have a non-negative vector, and I think the one norm it's the same as the sum.

So, oh, and by the way, that's why a lot of things would – I would still call them sort of  $L_1$  heuristics, but you might sort of grip through the paper and never see  $L_1$  mentioned because if a vector is non-negative, it's just a sum. But  $L_1$  sounds fancier than to say L – you know, than the sum. The sum heuristic seems kinda dumb. Okay, so this is the – this is the nuclear norm. And we'll do an example. Actually, it's a very interesting example. It goes like this. You are given a positive semi-definite matrix. And what you would like to do is you want to find – you want to put this in a factor model.

Now a factor model looks like this. It's an outer product, a low rank outer – a low rank part plus a diagonal. What this – I mean it would come up this way. It basically says that if covariance of a matrix, it basically said that that's – that random variable is explained by a small number of factors. In fact, it's the  $R - R$  is the dimension of  $F$ , the number of columns. It's a small number of factors. And then  $D$  is sort of an extra variation. So this would be the factor model. Now by the way, there are some very easy ways to check factor models. If  $D$  is zero, how would you approximate – how do you approximate a positive semi-definite matrix as just low rank? How do you do that?

I give you a covariance matrix like the covariance matrix of 500 returns. And I want you to tell me – approximate it as a matrix of rank five, how do you do it?

**Student:**It's the [inaudible].

**Student:**It's [inaudible] symmetric.

**Instructor (Stephen Boyd):**So you should say – I can [inaudible] decomposition, but it's the same as the SVD. So you take the eigenvalue decomposition, and you take the top five ones. So we know how to do factor modeling without this thing. But if I want to – let's do one more. How about factor modeling plus – suppose all of the – suppose instead of  $D$  this was  $\sigma^2 I$ . Can you do factor modeling for that? How?

**Student:**[Inaudible] eigenvalues almost [inaudible].

**Instructor (Stephen Boyd):**Exactly. So the way you know it is you look at the eigenvalues of something, and you see – you would see five large ones and a whole bunch of small ones all clustered. And that would be your hint. Okay, so that would do that. That would be one way to do it. So you can do this with this. But when these actually numbers are all different, no one can solve that problem. Actually, it's a hard problem in general. Well, in any case, this is the – this is the factor – simplest factor modeling problem. So  $C$  is a set of acceptable approximations to  $\sigma$ . And they can be – I mean it could be simple, like some normal, or it can be very complicated and statistically motivated.

For example, it could be a Kullback-Leibler divergence, which would be this for a Gaussian random variable. Okay. And that's just a very sophisticated way of saying that the two matrices are close. The two variances are – two co-variances are close. This is the one that would give you the statistical stamp of approval. The statistics stamp of approval would come from using something like that. Then a trace heuristic for minimizing rank is pretty simple. It goes like this:  $X$  plus  $D$  is your original matrix, and so your variables here are going to be  $X$  and – oh, oh, this should either be – well, I should either write – capital  $D$  is the diag of little  $D$ .

So it's – but anyway, it doesn't say that here which is weird. But that's it. So this would be the problem. And that's a convex problem. If you put rank here, it's a convex rank problem. So we'll look at an example. So here is an example where, in fact, I have a

bunch of data, which are – well, we know it. It's actually generated by three factors, all right. So what happens is you get snapshots of 20 numbers. They are all varying. They are random. You look at these 20 – you look at a bunch of these things, they – you get a full covariance matrix, full rank. But it turns out that three factors describe it plus the diagonal elements.

By the way, in a – if these were asset returns, the diagonal elements would be called the firm specific. They would be – that's the firm specific variation. Basically, each – it said that you have a bunch of factors. I think one is the overall market, typically. And then you get some other things. Some very obvious things if you look at factor models in sort of finance you get these things. And then the D's are the firm-specific volatilities. Okay. We'll just use a simple norm – a norm approximation. And you get a trace – you get a trace heuristic that looks like it's a convex problem. And what we'll do is we'll generate 3,000 samples from here.

Now, by the way, we are estimating, I guess, it's 20 by 20 covariance matrix. You have got about 200 numbers. So you are maybe about 15 times as many numbers or samples are there numbers you are supposed to get. I guess each sample is 20. So, okay, it's a couple of hundred times in terms of the estimate, but you asked me in covariance, so okay. And this is sort of what happens. It's rank three. And what happens is, is you crank up beta. You start with a rank – by the way, the top rank is 20. But you immediately get a 15-rank model. Then, as you increase beta, that's the – that multiplies the trace thing or something like that.

What happens is the rank of that X goes down, and down, and down. You get a very steep drop down at three. And by the way, that's – this is the hint that rank three is going to be – give you a nice fit. If you keep going, if you increase beta enough, it goes from – it goes down there to two and then stays there. Actually, I guess this would go down to zero at some point. We didn't show beta [inaudible] large. What's interesting is as we scan beta, this shows the eigenvalues. And so up here you have 15 nonzero. And you can see at different values of beta, eigenvalues basically being extinguished. They go to zero.

And so what happens is, right here, you end up with three from here on. And in fact, these are the right ones. So if we take beta as some number in here like this, we actually do get a rank three model. You can do polishing in a case like this, too, obviously. Well, you can figure out what polishing is in this case. But in this case, if we take 0.1357, that's the tradeoff curve, you find that the angle between the subspace, which is the range of X and the range transposed is 6.8 degrees. And that we nailed the diagonal entries; we actually got the firm, specific volatilities within about 7 percent. So this is just an example of this kind of thing.

Okay, so this actually pretty much covers up this – the whole topic of sort of  $L_1$  and cardinality. And the idea is instead of just thinking it as sort of a basic method where you just reflexively throw in an  $L_1$  norm, actually these extensions show that, first of all, it comes up in other areas like rank, minimization. These internet methods, actually very

few people know about them. They are not even used. Most people aren't even using polishing. Most people aren't even using the asymmetric  $L_1$  stuff. So if you are interested in getting sparse solutions, there is actually better things than  $L_1$  available.

And certainly, these things like these LP – I can't say it, LP norms – LP measures, I don't know. I don't know what word to say there that is okay. I'll just say it; LP, quote, norms, unquote, for  $P$  less than one. Those were these log approximations and these iterations. All of these things work. And I should also say – so I guess Emanuel Candiz and I had a long conversation like a year ago. And he said that  $L_1$  is the least squares of the 21st Century. And, okay, it's a good – that's a good – I mean that's good, actually. It's not bad. I think it's a little bit of an exaggeration, but it's not too far off, right.

It basically says that they are going to be the same way everybody needed to know about least squares and throughout the 20th Century, eventually everybody did. And by the way, by least square I mean fancy methods like Kalman Filtering, and quadratic control, and things like that. If you call that least squares, then you know a lot of signal processing, and image processing, and all sorts of other stuff ended up being least squares, period. And I think a lot is going to end up being  $L_1$  as people move forward. So, okay, so we'll quit here unless there is some questions about this material.

And then the next topic is actually going to be – we are going to jump to model predictive control. So we'll – that's going to be our next topic. Good. Good. We'll quit here.

[End of Audio]

Duration: 63 minutes