ConvexOptimizationI-Lecture10

Instructor (Stephen Boyd):Well, today we're starting, well, a whole new section of the course, although the boundary's not that sharp. The next sort of whole section of the course, and book, for that matter, is applications. So we'll just look at a bunch of applications. Instead of looking at sort of the analysis and things like that, we'll look at applications.

After a while, it gets a little bit tedious, because you start realizing in some ways, a lot of them are the same. But we'll look at a bunch of them, and you'll do homework on a whole bunch of them, that'll give you the basic idea of how these things go. And in some sense, this is really what the class is about. It's about actually formulating and using these methods for problems.

Okay, so our first topic is going to be approximation and fitting, and we've separated this so that at first we're not going to talk about statistical methods but we'll get to statistical methods actually very soon. So our first treatment of approximation and fitting is just non-statistical. But we'll get there and we'll see that it's actually very close. It's actually very interesting, the connection.

So the first topic, if you go down to the pad, is normal approximation. So this is a problem you’ve seen, absolutely certainly in the case of – with a 2-norm here, and it simply, you minimize the norm of AX minus B. So that is the problem. You choose X. And this has lots of applications, so geometrically you could say something like this: you are computing the projection of the point B on the range of A in this norm.

Now actually, we'll see that the choice of norm is going to actually be quite interesting here. There's lots of choices here. So that's the geometric interpretation of this problem. Another one is estimation. So you have Y equals AX plus V, and V is some kind of measurement area that's unknown, at the moment it doesn’t have a statistical model. But the point is it's sort of assumed to be small. In fact, specifically, it's assumed to be small in this norm.

So the norm of AX minus Y in this case is in fact the norm of your imputed V, because if you guess an X and you’ve measured Y, then for all practical purposes you're saying that X minus AX is what the noise was. And then the norm is a measure of plausibility. Actually, it's a measure of implausibility of the noise V, right?

So for example if norm V is large, that means it's highly implausible, your guess. If the norm is small, it's more plausible. So that's the idea.

Here, the best guess is going to be X star, the one that will maximize plausibility here. In another case, it's just optimal design. So in that case, the vector X or its coefficients, these are design parameters. These could be inputs in an optimal control problem. They could be anything. They could be inputs that you're going to send to a communication system, something like that.

AX is the result of your action, so X is an action, and then B is something like a desired action. And here, the question is to find the – sorry, the desired result. To find the action or inputs so that the actual output, that's AX, is as close as possible in this norm to this desired outcome B. So that's – all of these are just applications of a minimized norm AX minus B.

You should have seen examples of this. Least squares, I mean, certainly you've seen, or regression. So least squares just says minimize this in the 2-norm, this has a solution. In fact, the optimality conditions are exactly this. It's A transposed to A, that's the normal equations, X equals A transposed B. So if X satisfies this, it minimizes norm AX minus B, and actually, that's if and only if. So that's the condition.

If – this actually always has a solution; in particular, A dagger B, where A as the pseudo-inverse will work. But if A is full rank, then A transposed to A inverse, A transposed B will work. So this – and this is just review.

[Inaudible] approximation says it will measure the norm, in fact, by the maximum of the absolute values of the – this is called the residual, X minus B. So we'll look at the maximum of the absolute value. That, of course, transforms to an LP, so that's nothing but this. It's minimize T, T is a scaler variable subject to these two M linear inequalities here. There's one here and one here.

Another possible norm at the other extreme would be the 1-norm. in the 1-norm, you minimize the sum and the absolute residuals approximation, and that can be solved again as an LP. Here, you introduce a full set of M separate new variables, and you solve this linear program.

Okay, now these are – it's actually useful to look at this from a – even just go all the way to a more general form, which is not even a norm. So you look at a penalty function. So the idea is this: you want to minimize some penalty function, a sum of penalty functions of the residuals.

So AX minus B is called the residual. It's – well, it's just how much you're off in solving AX equals B, that's the residual. And of course by choice of X you can shape this vector, and in fact your possible choices of R is an affine set, right? It's the set of all AX minus B where X ranges all of our end. That's an affine set.

So you have this affine set, and you want to choose – you want to minimize the residual – some penalty of the residuals like this. By the way, even this is a specialized case, because here we have the same penalty function for all residuals. You could obviously care much more about some residuals than others, but once you get the idea, it's easy to generalize this any way you like.

Okay. So this is a penalty function approximation problem, like this. And you can even think of it, if you like at this point, you could even think of it as a multi-criterion problem, and here we're just taking a sum of them, but anyway.

So here are some typical penalty functions: quadratic. In fact, that's kind of the most typical one. That's for several reasons, which are all connected and tied together. First of all, historical. It's probably the first penalty – the first widely used penalty function, and actually currently most widely used penalty function in many applications. It's just quadratic. Why? Well, because there's an annalistic solution for it, like this. So – and these are obviously not unconnected, right? So that's that.

Student: Could you parse the objective function? What you mean by evaluating [inaudible] all of these different values of R?

Instructor (Stephen Boyd):Well, you commit to an X. You say, here's my X. I calculate the residuals. You got M residuals, and you run up a bill. For each residual I apply my penalty function. That's your total bill.

Student: You have to use the same penalty for each?

Instructor (Stephen Boyd):You don't have to, but we are here.

Student: Oh, okay.

Instructor (Stephen Boyd):Yeah. Okay. So this is just one. I mean [inaudible] it's the most widely used for many reasons and so on. Now there's other options, and we've seen another one which would be sort of in an L-1-norm would give you this – something like that. And of course you could have things like the three halves norm and things like that, if you wanted to.

So another one – but others are more interesting. One would be like a dead zone function. That's a function that looks like this, and then for example grows linear. That's this function. And in fact, the way you should conceptualize the function five, the penalty function, is extremely simple: it's a map of irritation versus residual. So it's how irritated you are with the residual of a certain level.

I mean, I know this is silly, but it points to the square function, tells you that if a residual is small you're actually irritated very little, which is small squared. If a residual is big, though, the square is very irritating, meaning it's big squared. If you have something like the one norm, these two reverse, relatively speaking. If you have a small residual, you find it, relatively speaking, very irritating. I mean, compared to a square.

If it's large, you find it relatively less irritating. So these are stupid ideas, they're very, very elementary, but it's actually all gonna come together. It'll actually all come together with maximum likelihood estimation and statistical models, and it'll all sort of make sense.

But the first thing is just to think of it really that simply. If you have a dead zone linear model, what you're saying is the following – I mean, it's so dumb it's amazing. It basically says for example here, I don't know what this – where the threshold is, maybe

.25 or something? It basically says the following: it says that actually for residuals between plus-minus .2, I don't care at all. That's good enough. For residuals bigger than that, I start carrying, and I start carrying linearly, okay? So that's all.

Then you can go the other way. You can have – for example, here's a log barrier. This is minus A squared log, one minus U over A squared, and if you work out what this is, it's actually very, very simple – this thing will go to plus infinity, outside the open interval minus AA, and so that's [inaudible] here, that's a log barrier. By the way, the log barrier will coincide with very high accuracy with a quadratic for small A. So this looks exactly like – in fact, I mean, it's very, very close to simply U squared for U less than let's say .3A or something like that – .2A for sure. It's really, really close.

But then this – what happens is this says that for small residuals, you care sort of quadratically. But as they start getting bigger, you care super-quadratically, and in fact you are, in this case, infinitely offended by residuals that are bigger than or equal to A in absolute value. So that's the meaning of this one.

Well obviously when you solve this problem you get different values of X and different residual distributions. So here's just a silly example just generated – everything's completely random, A is 100 by 30. So you have 30 variables, and you minimize the penalty function, penalty applied to the residuals for – this is absolute value, this is going to end up being the same as 2-norm, dead zone linear, and a log barrier.

And here's what you get – and actually, this is already really interesting. Let's go back to quadratics. So that's the quadratic residual, and you see a lot of residuals sort of packed around the value – let's see, that's a half? You know, plus-minus a half. You see a lot of residuals here. They're kind of evenly distributed.

You see a couple of residuals out here, okay? Now, these are out here because they have to be out here. In other words, you can't have all the residuals small or something like that. That's actually the point. If you look at the L-1 solution, you see something much, much more interesting, as you can look at the scale to sort of figure this out.

There are a whole bunch of residuals that are – well, a histogram would tell you near zero, but I'll tell you exactly what they are: they are zero. So there's a whole bunch of residuals; in fact, something like almost 40 – 35 are exactly zero. That's what this is, okay? Now, how can you explain that? Well, if you compare it to, say, least squares like this, it's – I mean, we're gonna explain this many, many ways, but just to get the first intuitive picture of why this happens, this is quite easy.

I'll anthropomorphize the solution of these convex problems as nothing more, but here's what happens. In L-2, once you get a residual small, you expend no effort to make it any smaller, because your irritation level is small squared at that point, and there's just no point – you don't care. So the importance is that once these residuals are pushed into sort of in this area, there's no – the diminishing marginal irritation is very small, and so you just quit, okay? That's fine. You focus on big residuals.

Now in L-1, it's actually quite interesting. I mean, this is kind of – everything I'm saying is kind of obvious, but you'd be surprised at what the utility of some obvious things are. The marginal irritation in L-1 is constant. In other words, decreasing something by .1, if it's out at 10, or if it's at .1, is actually about the same. Well, sorry, it's not about the same; it's exactly the same, right? To decrease a residual by – so your marginal irritation level, that's the derivative of the penalty function, is constant.

So what this says is you will – when you have small residuals in L-1 – and again, I'm anthropomorphizing it, but that's okay – it's actually worth its while to actually take those residuals and push them all the way to zero. That's the first – that's the reason you actually get a whole bunch of zeros here.

By the way, this is actually very – this is sort of – this has been around, people have known these things and used them for, I don't know, 20, 30 years now. But in the last couple of years, ideas related to this are very, very fashionable, let's just say. They're not just fashionable; they're actually useful. But they also happen at the moment to be very, very fashionable. And I'll talk more about these in some other context, so. The point here is the residual distribution for L-1 is you get lots of zeroes, and that is not an accident by any means.

All right, let's look at dead zone linear. Well, dead zone linear basically says – yeah, that's minus a half, so plus-minus a half here. Okay, so what this basically says is there's a free ride on residuals out to plus-minus .5. Above that, you're gonna pay linearly. Now interestingly – I mean, this is hardly anything you – you wouldn't suspect anything else, but of course if it's a free ride for a residual at the plus-minus .5, it's hardly surprising that a giant pile of them will end up right at the boundary of where you start paying.

So they'll end up right at plus .5 and minus .5, like that, okay? Those of you in communications will know that this – or will connect this actually to something called blind equalization, so – or maybe – well, I'm not sure, okay, so, if you're in communication and know about blind equalization, you would recognize this.

So this is hardly surprising, right, that you sort of push some out where it's a free ride. By the way, these are in here sort of accidentally. I mean, because it's completely – these are free to move left, right, doesn't make any difference. They're there just because they're – it didn't matter. It actually helped to have them in here, and it helped out other things.

Out here, you sort of care less about these, and you can see actually that the outliers, for example, are a little bit farther out than in least square, it's not surprising. Okay.

Here's the log barrier. The domain of the log barriers is plus-minus one. So obviously, all residuals have to be between plus and minus one, strictly. I mean, obviously, and you can see that's done. Now they go pretty much up to – almost up to one. Now the reason out here, these residuals are very irritating. They're there only because they sort of have to be there.

By the way, I didn’t show it here but if I minimized L-infinity norm, what do you think the residual distribution would look like? What do you think?

Student: All of them about equal.

Instructor (Stephen Boyd):What's that?

Student: All of them about equal.

Instructor (Stephen Boyd):Equal, or in absolute value?

Student: Yeah.

Instructor (Stephen Boyd):Yeah, exactly. If you do L-infinity norm approximation, then you take the infinity norm, and the infinity norm is simply the – you imagine something that goes like this and squeezes the residuals. And you squeeze the residuals down until you can't squeeze any more, and you're absolutely right – if you do L-infinity norm minimization – [inaudible] approximation, you'll find a whole bunch of residuals will be at the positive limit and a whole bunch at the negative limit. That's exactly right, okay? So that's the picture. And these are all very simple ideas.

Here's one that's quite interesting. It's the Huber penalty function, and it's a function that looks like this. It's quadratic out to some transition point, and then above that it's linear, okay? But we'll have a statistical interpretation – one statistical interpretation of this soon.

It's actually very interesting, what it does. Obviously, it's convex. You can do a Huber function approximation. What's extremely interesting, it sort of combines L-1 and L-2, and let me show you what it does, and you can sort of guess. So here's an example. I mean, it's a made-up example, but it gives the idea.

Here's a whole bunch of points – like, I don't know, 40 Q points, like this. Actually, this is 40 points here that kind of lies on the line, and then we threw in two, you know, serious outliers, like that. Okay? Well, the least squares fit is this dashed line, and I mean, it's kind of, you know, obvious that that's gonna happen. That's an extreme outlier, and the cost of one point being way far out is, well, it's way far squared in least squares.

So this thing will actually, even though that's a mere one, two points here, this thing will actually rotate considerably, even though it's just one point, one outlier here and one here. So I guess they're putting a strong torque on it, here. Well, by the way, that analogy is actually quite perfect if you attach to this line springs. Then – of unit, if you put springs with unit stiffness, it's in compliance. It's exactly correct, right?

So these two actually put a torque on it, because they're quite extended and they put a big force on it, and they rotate this like that, okay? That's the least squares thing. Well of course; I mean, that's what you'd expect.

If you make that – if you look at the Huber estimate of these, you get something very interesting. It's the dark line here, okay? And you can see, is it twisted? Little bit. But actually, not much. Now, you know, obviously if you have data, you know, that you can plot, you don't need any fancy methods. You should use your eyeball to fit things.

So obviously this example is not the point. The point is that you can now do this with, let's say, a thousand measurements in estimating 400 parameters or something. And let me tell you, your eyeball cannot possibly identify outliers in things like that. Totally out of the question. Completely out of the question.

This will work in a way that actually often is spooky, it's so good, okay? And we'll see reasons why and things like that. By the way, you can do all sorts of – you can imagine all sorts of methods for treating outliers that go beyond this, so for example, you might use a Huber estimate to start with, then simply go back and look at the residuals at that point.

In this case, it's embarrassingly simple – these two points would be obviously flagged as super-high residual, and then you'd trim the residuals and you'd refit. So you'd flag those data, those measurements as flawed and you'd refit it. And by the way, in this case, it would just work perfectly.

By the way, if I were to crank these down like this, this method in least squares would fail. It would actually start failing. You'd start identifying points up here as the outliers and things like that. The Huber one would actually work for a huge – I mean, when it actually starts getting not too obvious here.

Student:[Inaudible] will actually choose them?

Instructor (Stephen Boyd):Not – for this lecture? Yeah, I can. You do it the way you think you might do it. You wiggle M until you like what you see. Now, no one will ever tell you that. No one will admit that, but that's the truth. That's how you – that's more like asking somebody how do you set the weight functions in regularization? Truth is, they wiggle it until they like what they see.

Now later, actually, there's a very good statistical method you can actually optimize actually by convex optimization over M and X. It's called concomitant something or other. I don't – we'll get to that.

You know. No, yes?

Student:[Inaudible.]

Instructor (Stephen Boyd):You can do cross-validation, that's a perfect – so actually, there are better methods. Oh, by the way, we'll also – if you have any idea about the variance of the nominal noise, then of course that sets M. That would set M right off the bat, so.

Student: For the histograms, wouldn't you think that having a steeper barrier function move your residuals closer to zero rather than spreading them out?

Instructor (Stephen Boyd):Yeah. That's what it does.

Student: Why is it then when you steepen it to the point of infinity, it spreads it to basically uniform? I mean, in the previous slide, it showed, like, pretty much all the bars being [inaudible] uniform as you steepen that curve.

Instructor (Stephen Boyd):Right. So you're not steepening it. Quadratic is much steeper out for big numbers.

Student: Right, that's what I'm saying.

Instructor (Stephen Boyd):So you're making it soft. No, no, it's the other way around. When you go to L-1, you're much more relaxed about big. No one likes outliers – I mean, no one likes large residuals, okay?

Student: Yes.

Instructor (Stephen Boyd):But among convex penalty functions, very important, among convex penalty functions, a linear tail is the most flexible you can be about large residuals. So that's why some will spread out, and it'll allow a few to spread out if you can make up for it with the rest. And that's what this does. So that's the idea.

Also, I don't know if – I'm not sure how this – I'm not sure if the point was heard, but we heard from some of our statisticians that you could choose M of course by cross-validation, so. Which is true. We'll get to some of that. Okay.

Now the dual of these are least norm problems. So in a least norm problem, you minimize norm X subject to X equals B, and lots of interpretations of this. I mean, one is that you have a set of solutions – AX equals B – that's again an affine set, and you want to compute the minimum norm point; that's the projection of the point zero onto this affine set.

In estimation, here, the model is this. You have not enough measurements to identify a parameter perfectly, so I have B equals AX. A is fat here, so. However, the measurements are perfect. So the bad news is I don't have enough measurements. The good news is the measurements are perfect, okay? So I have perfect measurements, but not enough of them, and what that says is AX equals B is the set of parameter values that are consistent with my perfect measurements. So that's what that is.

If someone says to you please estimate X, the answer to that point is I cannot possibly; you have to combine that with prior information about X to say anything intelligent about X. Because, in fact, any X that satisfies AX equals B is consistent with your perfect measurements.

So norm X then becomes a plausibility measure – implausibility measure. So if X is – if norm X is large, that means it's sort of implausible. You assume it's more likely that X is small than large. So that's – and then this says find me the most plausible X or something like that.

In design, you can think of AX equals B as a set of M constraints, or specs, actually – specifications, requirements, whatever you want to call them. There's design choice. In other words, A is fat so there's lots of solutions of AX equals B. For example, X might be a set of forces or something that you're going to apply to something, and this might be some moment constraints or the constraints that a vehicle arrives at a given state at a given time, or something like that, that's what this is.

But there are lots of force programs, let's say, that do the trick, and among those you wanna find the one of least norm, for example. And that would be something like a most efficient design, or this could be a minimum fuel problem, for example, if this was a one norm, and if the one norm was a good measure of fuel usage. Okay.

Of course for least squares, this has an analytical solution, which I won't go into much. In the L-1 case, it's actually quite interesting, because when you minimize the 1-norm of X subject to AX equals B, you get a sparse solution. You get a sparse solution of a set of linear equations.

By the way, there are now – I don't know, every three days, I would say, maybe more, a paper appears on this subject – this topic. So, and literally this one problem right here. And you hear oh, boy, is it fashionable now, and people are talking about – they call it compressed sensing, and you hear all sorts of various names for this.

Very, very fashionable. Actually, in some cases, for some very good reasons. What this does – I can give you a quick example here. A quick example would be something like this. Suppose I want to estimate a vector X. Let's say it's an image or something like that, or it doesn’t matter. But I'm told it's sparse. So I have a vector in, I don't know, in R-5000 or something like that, but I'm told it's sparse.

Most of the entries are zero, so it's sort of an image with lots of zeroes, and there's some sparse thing in it, like that. And then I say, well, I'll give you some measurements of X, and these could be, you know, projection measurements, they could be anything. A could be a convolution operator that smears it so I can give you a smooth image. A could be a Fourier transform – this is exactly what would happen in medical imaging. In, for example, MRI, I would give you a Fourier transform of what you're imaging.

Now if B is – if A is square, then of course, I mean, there's nothing to say. You'd just take the inverse – you'd take X as A inverse B. If A is tall, you’ve taken more measurements than there are parameters to estimate, and of course you'd do something like a least squares, and you actually use the extra measurements to blend and give you a better estimate.

But now let's go down to the case when A is fat. So I want to measure a vector in R-5000, but I only give you 500 measurements. Now of course if someone asks you this, you should normally just turn around and walk away, because it's ridiculous. No one should ask you to estimate a vector in R-5000 based on 500 measurements, right? Well, unless they tell you there's more to it, okay?

So the more to it is that X is sparse, okay? Now that's actually a hard problem. For example, I might tell you X has only 300 non-zeroes. Now by the way, if someone tells you which non-zeroes they are, it's easy again. Because if they tell you which of the 5,000 entries, which 300 are non-zero, I simply pull out those columns of A, I get a reduced system, and now A is skinny again and we have – we're back in the realm of more measurements than parameters to estimate, okay? And everyone's happy.

But they don't tell you. They say, no, no, I don't know. I don't know what they are. But there's only – there's no more than 300 of them. Well it turns out this – if you solve this – this is convex, so it's quite simply, this is quite straightforward to solve – if you solve this problem, then at least there are actually now – first let me just say the practical fact.

The practical fact of the matter is that you will do stunningly well at actually getting the exact X. I mean, shockingly well. So that's actually been known for a while, but that's okay. What is actually new is that there are actually some results that tell you, depending on A, depending on M, the known scarcity of X and so on, that this method will actually, with extremely high probability, produce the exact answer, okay? So this is the idea behind sort of compressed sensing.

So in compressed sensing, in MRI you do it – you run it for one-fifth the time, so you're actually, you have one-fifth the number of Fourier transform measurements than you would normally take. And then you use something like this, so. All right, that was just a little aside, and we'll – you'll see bits and pieces of this in other areas. Okay, so that's just a 1-norm.

And the extension is least penalty, and of course same ideas hold. Except here, the penalty's on X. So for example, if FI is something like a fuel use function, then this would be a minimum fuel control problem, for example. Okay.

Closely related is this idea of regularization. We'll look at a couple of examples, and then you should be able to generalize it. Oh, there are lots of things I'm not saying here because I assume that in the fifth week of the class these are totally obvious. If I go back to these approximation problems and minimum norm problems, I can add any convex constraint I like.

I could say, for example, X is positive, I can give you lower and upper bounds for X, I can give you polyhedral constraints – anything I like. But I don't need to say it, because totally obvious, okay? So that's not even worth saying. So you just see the basic one here. Okay.

Let's look at regularized approximation. Well, the basic idea in regularization – I mean the most basic one, and we'll see extensions of it in a minute – the most common use of regularization is this: you want to fit something or minimize some norm, AX minus B here, but at the same time, you want X to be small.

This comes up in all sorts of things, so it's a bicriteron problem, and in fact the right way to say it is this: it's minimized with respect to R plus two. That’s the cone. This pair of objectives. And the Pareto optimal curve, or the optimal trade-off curve between fit and size, if you wanna call it that, is something that would tell you exactly how, you know, these two would trade off. Okay.

And, you know, this has lots of application. I mean, for example, in estimation it would tell you this. It would say there's many ways you would actually – reasons you would want X small. Now remember, to make X small in general you will give up fit. So what you're saying here when you solve this, or choose some Pareto optimal point is you're going to accept a larger – a worse fit for a reduction in size of X. That's the meaning here, okay?

So you might ask why would you do that, and there's lots of things – lots of reasons people might do that. One would be that your model, Y equals AX plus V, is actually only – that's irritating. What do you think? Think I can – I'm just not gonna – I'm not gonna be able to do it. I'm not fast enough. I'll try, though. We'll see what happens. Where is it? Okay. Mm, okay, all right. I'll just ignore it. All right, all right.

So you might know that this model is only valid for X small, and in fact YA could be obtained, of course, as – there could be a nonlinear mapping from parameters to measurements – extremely common. And A could be a linearized version of it.

By the way, A could be from the Jacobean. That would be if you're still an adherent of classical calculus. But A could also be from something like a particle filter or something like that, okay? And to do that, you would of course generate a bunch of Xs that are plausible, run it through your measurement system, get the measurements, and then you'd actually fit a linear function to it, period, okay? So that would be what that is.

And then here you want it small, because if you choose an X that's big and have a very good fit, you could say wow, I have a great fit. But the point is X is so big that your model is no longer good, so your fit is only sort of on paper here. Okay.

In optimal design, you're trading off something like fit versus some kind of cost or something like this. It turns out also, and we'll see this later, that the size of X is related to the sensitivity in this problem with respect to uncertainty in A, in the matrix A. And let me just explain that very, very carefully now, or roughly now and then we'll get back to it later.

Let me ask you this. Let's suppose that A were to change, like all the entries were to change a little bit, like 5 percent. If X is zero, how much difference does it make? None whatsoever. And if X is huge, how much difference does it make? How much? A lot. So I rest my case. So roughly speaking, the size of X is at least related to the sensitivity of AX to changes in A.

So therefore, you might want X small because that's related to how sensitive AX is to changes in A. this is kind of obvious, but. Okay.

So how do you solve a regularized bicriterion problem? You scalarize. And in fact there are many methods to scalarize. This is, by the way, not how it's done, because not enough people know that it's even possible. The number of people who even know you could do this is very small. It includes you and other graduates of this class, but not many others.

So one way to do it is to just add a [inaudible] you scalarize. You take the two objectives like this. Now here, as you change gamma, of course you have a – in fact, what they call it is a regularization. In general, it's called a Pareto optimal curve; it's the optimal trade-off curve of norm-X versus norm-X minus B. It's also called, in this case, specific case, it's called the regularization path. So that's a very common name for it – regularization path, like this, as you vary gamma.

Okay. Now – by the way, the way, depending on historically what the norms are, if you square AX minus B and norm-X, then this actually can be solved by just standard least squares. So it's common to do this. And of course you get the same trade-off curve. It's a different parameterization and obviously gamma and delta don't correspond example, but they trace exactly the same curve out. So every point on this curve is obtained by this – okay, now that's irritating. Don't – don't say anything. No.

That was irritating. It's taunting me. It's taunting me. Hm. Okay, all right. All right, so this is the most classic regularization. It's taken off regularization, and why? Because it stays with least squares and everything's the same. This is just completely standard, so I'm not even going to go into it.

But we'll do an example, and I should say this is only the simplest version of regularization. In general, you tack on – you may tack on a bunch of regularization terms. That's even more common. So for example, I don't know, take an image. You might – it still there'd be no reason to believe that your image is sparse or whatever, and I mean, I don't know, unless you're taking – if you're imaging something from outer space or something like that.

There'd be no reason to believe. More likely [inaudible] it's smooth, and then you'd put something like a smoothing [inaudible]. It'd be the norm. You'd regularize – to make something smooth, you regularize with a smoothing operator. And you could do both, so.

And in fact the way these things generally work is you'd keep adding regularization operators and things like that, regularization terms and twiddling with the knobs until you like what you see.

Student:[Inaudible] is it just like a [inaudible] problem?

Instructor (Stephen Boyd):Which one, here? It is, yeah. It is exactly that. So yeah, it's exactly that, yeah. By the way, that's another way to solve this, is to do this. Is to minimize norm AX minus B subject to norm-X less than some, you know, kappa or something. And then you vary kappa. And the indeed, the Lagrangian for this is something like that. Well, except for a term minus gamma K. So yes, it's related to that.

Actually, you should have build neural links between regularization in – scalarize in multi-criterion optimization and duality, because both of them involve kind of the same picture, and the picture kind of looks like this. Like that. That's the picture. And you change, you know, either dual variables or weights, and this thing rotates and the point of contact rotates. So kind of the same pictures should work in both places. You look confused. No? Okay.

So okay. So let's look at a – this is sort of a typical example of how you'd use regularization. So we have a linear dynamical system, so we just have a convolution operator here, like this. And we have three objectives, so this tracking error, so there's some desired thing and we want the output to track this thing.

So if you only cared about this, you might call that a deconvolution problem or something like that, because that's a convolution and essentially, you wanna unconvolve the desired to get the optimal input. So it's just a straight deconvolution problem.

On the other hand, we don't want U to be gigantic, so we're going to limit the size of U. And we also want the input to be small – sorry, smooth. And to make it smooth, this is an example where we just take a quadratic form, this is a first-order difference, and it's a quadratic measure. Everything here is quadratic. That's it.

Okay. So we used regularized least squares. We're gonna minimize J track plus delta, J, and this is the derivative, I suppose. And eta times J times the – that's really, really irritating. Okay, sorry.

Of course, people watching this later will have absolutely no idea what's going on. It's a student who's taunting me, just for those of you watching this on tape. Yeah. If I get their student ID number, they're in deep, deep trouble, so okay. All right.

So okay, and this is just a least squares problem or whatever, and it's just to kind of give a rough idea of what happens. So in the first case, we take – let's see, so in the first case we put no penalty whatsoever on smoothness, and we put a small penalty on size. And this is the input that we're told does it, and you can actually sort of see here both things are – the two things shown here are the desired one, the desired trajectory you want to hit, and the trajectory you do hit.

And they're not perfect, by the way, and of course if I take eta and make it smaller, these will get bigger and I'll get better tracking here, okay? So that's the first one.

Now in the middle, again, no derivative penalty, but we increase eta. So you increase eta and it says basically I want a smaller signal. And you can see this thing goes between plus five and I don't know, minus eight, or something like that. You say, I'll take a smaller signal but I'll – well, you have to accept worst tracking error, okay? So, and in fact that's exactly what happens here.

You can see it smaller here – well, you can just look at the scale and see that it's smaller. And indeed, you can now actually see worst tracking error. Okay.

And in the final one – in the final one, we take the following one. We're gonna actually now increase a derivative penalty, and you can see here's a smooth input, and your tracking error is – actually, this is the kind of thing you want, by the way. This is what you – what you want regularization for is exactly success stories like this, where you get something you want while giving up very little for it. I mean, that's kind of what – that's why you use regularization.

And in fact, strangely, that's often the case that that works out this way. And here you can see I don't know what the difference in tracking error between this and this is, but it's not – whatever it is, it's not huge, and I'm using the eyeball test here. It's not huge. And yet you could say, again, depends. I mean, you'd have to be able to argue that you like this thing a lot better than that, or something like that.

And you can say look, here's something a little bit worse than that, but look, it's much smoother and it's smaller or something like that. Okay, so this is just this kind of regularization. That's what it looks like.

Signal reconstruction. So this is a very special case in fact of this, but it's widely used, and it looks like this. Here's what's happened. I want a signal. There's a signal I'm given, but it's been corrupted, and just additively, so the linear operator A is I. It's the identity. So I'm giving it corrupted signal, and I want to sort of decorrupt – I just wanna subtract off the corrupted part.

So xhat is gonna be my approximation of what the true signal is, without the corruption or noise or whatever you wanna call it. So when I guess xhat, then I'm really guessing that the corruption or noise was xhat minus X corrupted. That's the Y, that's the observed one, okay?

Now, then if I take, for example, a 2-norm measure of this, this is – often, people would take an RMS measure, that's just a multiple of that, and what that tells you, that tells you, like, how big the noise – your imputed noise is, because you're basically imputing a noise in this case. And we'll see later, statistically this would correspond to a log likelihood term for the noise.

But what this says is, you know, basically if this is small – in fact, what you'd really probably want is you'd want this thing to be on the order of what you would guess it would be if you knew something about the noise involved. So that's what you would really want this to be. If you had some idea of how big the corruption is.

Okay. Now you're gonna pay – you're actually gonna deviate from what you saw, but what you're gonna do is you're gonna at the same – you're gonna minimize, or here's what you're gonna gain, so you're gonna gain a much smoother signal. So FI here is called, like, a smoothing function or a regularization function, it's got all sorts of names, and examples would be things like this. So this is just in one dimension. In one dimension it would look something like this.

A quadratic smoothing function would be this. So it'd be the L-2 norm, squared, and some of the squares of the differences, okay? So this just penalizes the sum of the squares of the differences here. This would be completely classical.

One also very interesting is total variation norm. It's not a norm. People say norm, though, but it's not a norm. So that's – put that in the – flag that with the same flag you use when people say overcomplete basis, by the way, because – anyway, for hundreds of years in math, a basis was by definition not overcomplete. But anyway.

So this is just – it's called total variation, you will hear it called total variation norm. It's not a norm. By the way, it's not a norm because when are both of these – when do these vanish? Or what Xs would they vanish?

Student:[Inaudible] constant.

Instructor (Stephen Boyd):Yeah, constant. I mean, when X is constant. By the way, they all vanish with a vector of all ones, okay? Norms don't do that, you know, by definition. Still people call this the total variation norm.

Anyway, let me show you what this is. Actually quite interesting. If you have a signal, you know, it looks like this. I don't wanna make it too complicated or it's going to be hard on myself. So here's what – the way you calculate the total variation is actually very interesting. Here's another way to do it. It's exactly this: is you identify the peak to valley things, like this, and it's the sum of these – of alternating peak to valley heights. I don't know if that makes any sense. This make sense? So that’s what it is.

So if you see a signal like this, it's the – and so by the way, this puts a huge penalty on wiggling, right, because if you wiggle like this, you run up a very big total variation bill, okay? But by the way, that's also true of the quadratic one. The one interesting thing about this one, which now you can guess because you know a little bit of intuition about these L-1 things, what this says is when you do have to make a shift, total variation is gonna care a lot less about it than least squares.

Least squares, this measure here, is gonna charge you a lot if your signal goes like this and then goes down. It's gonna – this, you're gonna run up a big bill here. You'll run up a bill here, but the bill here will be much bigger, in the sense of bigger squared. Everybody got this?

So this is total variation. It's been used since the '80s, actually, in image processing, very successfully, by the way. Audio reconstruction. I mean, this – all the old recordings and stuff like that have been, a lot of the wax things have been, recordings have been reconstructed, I mean, almost magically by things like total variation denoising. So I mean, it's actually just amazing.

Okay. So let's just look at an example. Here's a signal. Here's the original signal, which we don't know, by the way. What we get is this. Now, you know, honestly, from a signal processing point of view, this is not a big deal. I mean, you look at this and you say, there's a frequency spread, there's a nice spectral spread between the signal, which is obviously low frequency, and the noise, which high frequency.

Everybody would know what to do. You need a low-pass filter, you need to smooth it. Okay, so – by the way, guess what taken off – guess what, if you regularize with this, guess what you get? You get a low-pass filter. Is it causal? It's a low-pass smoother. It's got an impulse response that goes both ways. It decreases both ways. So that's what it is. You can check. Check. I'm okay.

All right. So, yeah, you can check. It's a nice, smooth kernel like this, it's a convolution. It's almost a convolution, actually, I should say. There's sort of end effects, but okay.

Okay, so here are the smoothed signals. These are smoothed with different – these are three different points on the optimal trade-off curve between smoothness and misfit, okay? Now by the way, if someone were kind enough to tell me what the RMS level of my noise was, this would be easy, because I'd wiggle the parameter until the imputed noise – that's xhat minus X corrupted – in norm or RMS value has about what the noise level is. So if someone were kind enough to tell me the noise level, what you pick on the regularization path would be very, very easy to do.

So okay, so here, you know, and these are kind of obvious that I guess you might say this is too smoothed. This is maybe a smoothed right amount, and that's not enough smoothing here, this one, so these are just three values here.

Okay. Let's see what happens on an example with total variation reconstruction, and it's not – it'll give an idea of it. There's a question?

Student: Yes, on the previous slide, what is X going on the upper left for? I mean, you wouldn’t know that, would you?

Instructor (Stephen Boyd):No, you don't know that. If you know this, then that's the – you want to make a good estimate of this, given that. That's all you wanna do.

Student: Okay, so you don't really know that the top upper one on the right is too smooth, because you don't know if it –

Instructor (Stephen Boyd):That's correct.

Student: Okay.

Instructor (Stephen Boyd):Right. You got it. You wouldn’t know. There has to be – I mean, which one of these you pick has to be more. But of course it's true in any multi-criterion problem, right? So you can hardly defend on an optimal risk return curve. You can't say this is better than that, or what risk is better than – you know, what return is worth what risk. That depends on other things. It's the same here.

Okay, let's look at total variation. Total variation works like this. Here's the original signal. By the way, the signal is chosen so the traditional linear signal processing is gonna fail on it, okay? So here's traditional linear – so here, the signal is something that's kind of smoothly varying, and then has, in addition, these jumps, okay?

Here's the one where it's corrupted, and if you do traditional linear signal processing, which is what the taken-off style regularization would be, you can see your choice is something like that. If you wanna get rid of all the wiggles, you're gonna smooth out this sharp transition, something that was basically one sample transition is gonna turn into something that is – I don't know the scale here, but 50 samples.

Why? Because of that square. And actually, a signal, if X moves rapidly, you're gonna get charged a lot for it. You're gonna get charged squared for it. So this is that. Okay, so by the way, in signal processing it would be something like this. If this was an audio signal and you smoothed it – if you low-pass filter too much, then things like the attack on a snare drum or kick drum are muddled now. And I mean, it's total – it's obvious, right? That that's exactly – you can hear what – well, if you know, if you have this background, you can look at that and you know what it's gonna sound like, okay?

And these would just be muddled. So that's no longer a snare drum, that's something more like what I wanted to do to that flag that was flying around.

So, okay, let's look at total variation denoising. Well, total variation denoising does this. Here it is. I mean, these are just different levels. I have a weird feeling these two are switched, don't you? Yeah, for sure these are switched, right? Because that's got more wiggles in it, I think. So I think these two are actually – I should check that. Wanna check it and see if I switched them?

Student:[Inaudible.]

Student: The bottom line is [inaudible].

Student: Because the whole thing is [inaudible].

Instructor (Stephen Boyd):The whole thing is switched. You're right, thank you. Thank you. The whole – they're all – it's upside-down for no good reason. All of my other ones went from too smooth, just about right, under-smoothed. Okay, so these are just – sorry, switch them. So here's the too-smooth version. Actually, now you notice something very interesting here, and this is predicted from L-1. You're actually – one of the objectives, the second objective is the L-1 norm of the first order difference.

If you just apply this intuition that any time you – if you throw in an L-1 norm or something like that and minimize it, with other stuff around, the argument of the L-1 norm is typically gonna be sparse.

If the – what can you say about a signal whose first order difference is sparse? There's a name for a signal like that. No, not constant.

Student: Piecewise.

Instructor (Stephen Boyd):It's a piecewise constant. Constant, that's the name for a signal whose first order difference is zero, okay? So, but the name for a signal whose first order difference is sparse is called a piecewise constant signal. And indeed, as you crank up the parameter on the total variation, you start getting piecewise constant things here. Okay?

Now interestingly, the exact – the sharp transitions are actually preserved. But you'd expect that. That's exactly what you'd expect, because the point is that in L-1, once – you know, if you have to have a big residual, no problem. In L-2, that's not the case. A big residual is a big residual squared. And what's interesting here – I guess this is the middle one – is that something like that is actually pretty good.

I mean, I don't know where the right thing is in here, but this'll work pretty well. So total variation denoising is used in image reconstruction, audio reconstruction, and it works, like, amazingly well. I mean, just – and it actually has been, like, it's actually used in a lot of commercial applications too, so it's used. Actually, in images, I should say something about this just for fun.

In images, what you're really doing is you're looking at something like this, is you would have the total variation – let me just write it – is if you have a function in let's say two variables, X and Y. You'd actually look at something like this. The equivalent of the least squares thing would be something like this. You might have – doesn’t matter, here, zero one squared, okay?

So that would be the equivalent of taking off regularization, and by the way, this would be sometimes called – this would be a Laplacian regularization. So that's what people would call this, okay?

Total variation is this. Okay, by the way, that's still a 2-norm. That's a 2-norm because [inaudible] is like this. So, or if you like that's integral – zero one squared – integral squared plus, partial F, part Y squared. Okay? So it looks like that, okay? So that would be – this is the analogue in two dimensions. Works really well.

You might even find out that it works well sometime in the future, if we finish that problem. So that's – anyway, that's total variation in two dimensions, and so on. Okay.

All right, let's move on to another topic. I'm sub-sampling out of the book, you should actually read everything in the book, of course, and we'll sort of assume – we'll assume that you have, so. Although there's only really one topic so far I think that was important that we just skipped completely, which was theorems of the alternative, and duality for feasibility problems, but that's okay. We'll figure out a way to have you do some of that.

Okay, the next topic is robust approximation, so here you want to minimize norm-X minus B, but the problem is that you don't know A. And there's lots of models for this. One is stochastic, so you could assume A has a probability distribution here. Oh, by the way, same for B. Actually, in a sense we've already assumed B has a probability distribution; that's what this norm is dealing with.

But here, we could assume A has a probability distribution, and for example you might minimize the expected value of the norm, or for example, expected value norm squared, which the second being occasionally tractable, so that would be that.

Another model is worst case. Worst case would say something like this: you'll be given a set of possible values of A, and you would minimize, for example, the maximum of the worst case residual over that set. Okay? That's another version of it. And in fact, you'll actually hear people – amazing to hear actually grown people have empty philosophical arguments about which of these is better, or something like that. It makes no sense.

The answer to which is better is, of course, it depends entirely on the context and application, period. So people who advocate worst case would say things like oh, no one ever really knows the probability distribution, or something like that.

Anyway, the good news actually is that a lot of the methods coincide, and to first order, they're actually often the same. So, okay. All right.

Now by the way, these are both convex problems. Why? Because if for each realization of A – I mean, that's obviously a convex function and an expectation over a random variable, which with probability one is a convex function, it's convex. So these are all convex problems.

However, there may be no particularly good way to write this – you know, no closed form or anything you can actually compute with. There's Monte Carlo methods; they would actually work quite well. And there are methods to deal with both of these.

Same here. So one of the – here, too, it is convex – there's no doubt about it, that the worst case residual is a convex function of X – very complicated. It's just a soup over a family of convex functions. Now the problem is in evaluating the soup here. So if there's a case where you can't evaluate that soup, then this is convex but it really doesn’t do you any good, because you can't even evaluate the function.

And there are lots of cases where you can. So let's just look at an example – I mean, a really simple example. So here is – what we're gonna do is this. I mean, it's this dumb, that's what we're gonna do. We're actually gonna have a line segment of As. So you’re a matrix, which maps X into Y, is actually unknown and it lies on a line segment parameterized by U here. So it's a line segment. U equals zero maybe is the nominal value, something like that. So if I take – if I just ignore it and I just said A-0, A zero is what happens if U equals zero. That's the nominal value. And I just minimize norm AX minus B. I just do least squares here. That gives me X nominal.

I could then – and this, we can easily work out what this is. It's actually a regularization problem. But X stock minimizes the expected value of the norm squared here. Then – and that's with a uniform distribution on minus one one. And then you can actually work out the worst case one – actually, that's extremely straightforward. In fact, we can even do it right now.

Actually, someone tell me how this was done. How would you minimize that? How do you minimize the – it looks hard, right? Because I'm asking you to minimize the maximum over an infinitely – over a line segment of least squares residuals. Any – somebody gonna help me on that? How would I do it, do you think?

Let me ask you this: for fixed X, what kind of function is that in U? For fixed X. It's what?

Student:[Inaudible.]

Instructor (Stephen Boyd):Oh, it depends where my fingers are – there. Now it's affine. Okay? And now what is it?

Student:[Inaudible.]

Instructor (Stephen Boyd):It's quadratic. It's a quadratic function, okay? Now quadratic functions over intervals, convex quadratic functions, let's talk about that. Where do they achieve their maximum? Boundaries, period. No exceptions. Therefore, this soup is actually the max. it's actually equal to this, right? It's equal to the max of two things: norm – A-0 minus A-1, X minus B squared, and A-0 plus A-1 X minus B squared. Everyone agree with me? That's it.

Now we're back in business. Everybody knows how to solve this. If you're using a system that does compositions for you, like CVX, you'd type this in, literally, okay? If you are not using such a system or you want to know what, in fact, CVX did, that's completely trivial. You introduce a new variable T, and two constraints, that's less than T and that's less than T, and then you go – now you have a quadratically constrained quadratic program. Everybody got this?

I'm not even saying these things, because they're sort of obvious, but I just thought I'd go over this one. Okay.

Let's look at the results. Well, they're hardly surprising, and this is for some random A-1, A-0, and so on. Here's what happens. If you look at the nominal choice of X, it minimized the residual when U equals zero. So it had to be better than the other two. It had to be, by definition, it minimizes the residual, okay?

So in the other two you gave up, this is the one – this is the stochastic one, and it minimizes the average – if you had a uniform distribution on here. So for example the integral of this function over this thing, that's this one, should be least for this middle curve. And I believe that's gonna be the case. Actually, you can see it's the case. Obviously, it beats the nominal one, because for the nominal one, when A moves away from the nominal, you start paying badly, okay?

And then the worst case one is this one. And in fact the worst case, by the way, is equal on the two sides. I mean, it had to be when you minimized this. They're both – it doesn’t have to be, but it would typically be these would both be equal, and you can see it's nice and flat, okay?

And so just roughly speaking – and this is a stupid problem with one parameter unknown, but the point is here some people would call this a robustness performance trade-off – something like that. So norm AX minus B is sort of at the nominal value – that's your performance. And you might prefer this worst-case one, because in fact it's giving you a small residual even when A changes, okay?

And so I mean, actually just ideas like this is the second reason why you would regularize things, basically. So, I mean, regularization is a very simple version. This turns, for example – where is it? This one. This one turns into a regularization problem. I won't do it; you could do it easily, okay? I mean, it's as easy as this – even easier, it's an integral, okay?

So that's actually a regularized one. And so the second interpretation of regularization is that you're actually solving a stochastic robust problem, okay? So, yeah?

Student: Is there a reason you don't take an expected value of A or [inaudible] A before [inaudible]?

Instructor (Stephen Boyd):Yeah, we could do that. You could – you tell me. What happens if you look at this? What if you solve that problem? And actually, I'd like you to compare that to this. If you remember. Or you could even do this. It's a sharper one. You know the difference?

Oh, by the way, you see this one? That is what the nominal one is, right? In this case. That's the nominal one. So minimizing this, if U has a uniform distribution here on minus one one? This is exactly simply throwing in the nominal and doing it. So what's the inequality here?

Student:[Inaudible.]

Instructor (Stephen Boyd):[Inaudible] it has the name, now which direction does it go? I never remember this, by the way. What is it?

Student:[Inaudible.]

Instructor (Stephen Boyd):It's like that?

Student: Yeah.

Instructor (Stephen Boyd):Yeah, okay, good, that's it. My rule of thumb is basically wiggling makes things worse. So this is when A wiggles, so this is worse, okay? So yeah, you can think of this – by the way, people would have all sorts of fancy names for this. Some people would call this, this is the "certainty equivalent problem." I mean, that's one name for it. Very fancy name for a very stupid idea, in general, which is just plugging in the – although it's the first thing you would do, right?

I mean, of course you'd start by solving this. You'd know that you're getting an – that the actual performance when things wiggle is gonna be worse by our friend Jensen, okay?

So, okay, let's look at a couple of these and see how these work. The stochastic robust least squares, I mean, this is actually kind of easy. You just take – let's take A is A bar plus U, where U is random, at zero mean, because if it didn't have a zero mean I could plug that into here. So in particular, expected value of A is A bar, but it doesn't matter.

And we'll have an expected value of U transpose U as P, okay? So that's actually all we're gonna need. If we wanna minimize the norm squared of this, if you just – you just work out what it is. It's just algebra. Which I won't even do it, but if you expand this, I mean, that's certain and you just multiply it out, you get this.

There's a cross-term, which is expected value – two times expected value of UX transpose times this, but expected value U is zero, so this cross-term goes away and you get this. The expectation goes inside, and you get that. And now you see a beautiful thing: that if you minimize expected norm squared, where the expectation is over the matrix A here, then it turns into nothing but taken-off regularization. It's quadratic regularization.

So you can actually go back and look at taken-off regularization. If someone asks you what are you doing, you can say well, I'm regularizing. And they'd say why? And you'd say, well, I don't want X big, and so I'm doing a – I've scalarized a bicriterion trade-off here, and delta is a parameter. And if I tune it, if I make delta bigger, my X will be smaller but I'll get a worse fit.

And if I make delta smaller I'll get better fit and bigger X. And if they're not satisfied yet, then you say all right, okay, here's the deal. I actually wanna minimize expected this thing here, and I'm taking into account statistical variation in A. And then that's what this is. And then delta has a very specific meaning, so here, delta is literally – what this – if expected value U transpose U is delta – in this case it's delta I, then in fact we know exactly what delta means.

It tells you about a – it says that the components of A each have variance delta exactly. Did I get that right, or is that the columns? Did delta divide by N or something? No, not [inaudible] because it's delta here. I mean, if I'd thought about it, I would have written it delta squared, but I didn’t think about it, and I think I may even have – is this right? Because U transposed – I think that might even be right.

No, maybe it's just – I think it means this. It means expected value of UIJ squared equals delta. Yeah, I was gonna write the square, but I didn’t. I think this is right – that's what it means. So the regularization parameter therefore should be the variance in the entries of A. So that's a partial answer to – or you could work it either way.

It says whatever taken-off regularization parameter you're using, you can back-interpret it, you can give a statistical interpretation as you are actually protecting yourself against statistical variation in A on this order of magnitude with variance delta. That's the other way to do it.

Okay, now you can also do worst case. By the way, this has been known for a long, long time – I don't know. Although for some reason, the references I've seen, I haven’t seen them go back, like, before the '50s. but this has been obviously known for, like, a hundred years or something. Maybe not quite in this form, but something like that. I'm sure it's known.

Now we get to some stuff that's new, and new means last 10 years, meaning that nobody 15 years ago knew it. So let's do worst case robust least squares. So what we're gonna have is this is an ellipsoid of matrices A. But there's lots of these variations. We have one queued up on the homework for you, so. Lots of variations on this.

We're gonna do robust least squares, except the way it's gonna work – game actually is the correct thing, I was gonna say it – is this. You're judged by the following. In ordinary least squares, you say the [inaudible] norm AX minus B squared. In robustly squared, actually, A lies in an ellipsoid, a known ellipsoid. That's an ellipsoid.

It's the image of the unit ball under a linear affine mapping from some dimension – Euclidean space – into the space of matrices. So that's an ellipsoid. Possibly degenerating – could be flat or something – but it's an ellipsoid. And so I give you an ellipsoid, and you'll be judged, if you – you'll commit to an X and then we'll work out the absolute worst possible A matrix for your X. we'll work out the worst thing.

And we wanna minimize that. Anyone can look at that and say that that's a convex problem because it's a supreme [inaudible] bunch of convex quadratics. But the [inaudible] bunch of convex quadratics is obviously not quadratic, okay?

This can be solved – in fact, there's a whole lot of these problems that can be solved exactly. This is one of them, and the details we won't be able to cover here, but I'll just give you the flavor of how it works.

If you want to – this function, you simply write it – well, by definition it's the soup over U of this thing here, like that. And it's a strange problem, right, because you're maximizing over U and then you wanna minimize over X. So it's a bit complicated, but in fact if you fix X and then you right P of X is just P, if you wanna maximize, PU plus Q norm squared, subject to norm squared less than one – oh, let me just ask you a quick question – is that a convex problem? Where the variable's U? Careful. Is it a convex problem? Careful.

Student: Fixed X?

Instructor (Stephen Boyd):The problem – here, right here – is that convex? The variable is U. Forget X; X is gone. Someone's committed to X already. In fact, you know what this is? We're doing the worst case analysis, that's all. So someone has basically – up here we were trying to do worst case design, forget it. Somebody says here's X, and we wanna find the worst possible residual over an ellipsoid of A matrices. So that's called worst case analysis, and I just wanna solve that problem. Is that a convex problem?

Student: No.

Instructor (Stephen Boyd):No. It looks like one. You're maximizing the norm – norm squared. That's maximizing convex function. Okay. By the way, you might guess you could solve this, because watch this – I need another thing. Actually, if that fly could come back and I could train it to land right – in fact, I'd like it to land right on the Q. Block out the Q, okay? Can you solve that problem? Why?

Student: If [inaudible].

Instructor (Stephen Boyd):Exactly, right? So although – so here's an interesting fact. If I were to get rid of that, that problem, though non-convex – oh, here he comes. Okay, that problem, though non-convex, we can solve. That's trivial. This is solved by – it's the, yeah, find the maximum singular value of input direction and – well, it doesn’t matter, I'm not even gonna go into it, right?

You know how to solve this, this is very easy, even though it's a non-convex problem. So you might not be surprised to know that if you put a Q here, it's still solvable. So it turns out the Lagrange dual of this problem is this – it's an SDP. Okay? This is one of the three or four very famous problems which is a non-convex problem with zero known, zero duality gap. So by solving this, you actually get the same value of that.

Now wait a minute – this is a minimization problem. And now let's reintroduce the fact that P varies with X. We're done. It's just – it's very simple, we just end up with this. Okay? And it's an SDP. So, I mean, I don't expect anyone to get this right now. You can look in the book to sort of – you know, to look at these things.

There's actually – we have an appendix with the three or four most common non-convex problems with zero duality gap. There's actually at least one or two of them is worth knowing. Something called the X procedure, which goes back into the '40s and earlier, and you should actually read that. It's in one of the appendices.

It comes up in computational – it comes up all the time. It's actually rediscovered every 10 years, so. In different fields. Not in each field every 10 years. In different fields, it's rediscovered. So there's huge numbers of papers on it. It's got different names, and I don't even believe it's a connected set right now, in the sense that there's some fields I don't even know about where they have a very powerful lemme or something due to, you know, let's say Johnson or something.

This is probably computer science, because they think everything was invented in the last four years. So surely it's known in computer science, someone figured it out three years ago, and they imagine no one else has ever heard of this before. But in fact it's hit all fields.

Bottom line is this – is you actually end up with an SDP that solves the ellipsoidal worst case least squares problem. And I'll just show what some of the results are. I mean, this is a really stupid one, this is just for a disc of A matrices, okay? And all we did was just took a uniform distribution in Us – and by the way, the problem was to minimize the worst case, not an expected value over a uniform distribution.

But just to give you a flavor of the robustness, and this is sort of what you get. So if you ignore robustness completely, you get something like this. Note that usually, it's horrible. There's a name for this – what's the name for these points over here? What's the name?

Student:I don't know.

Instructor (Stephen Boyd):Any name. Doesn't matter – I can think of many names. Lots of names, what would you call that? Like, how would you describe it to somebody if sort of you did this and this is what happened? What?

Student:[Inaudible.]

Instructor (Stephen Boyd):The tail?

Student:[Inaudible.]

Instructor (Stephen Boyd):No, no, no, no, it's not – but it's because that's the tail, too. No, no, no, no, no. This is called luck, right? Because you completely ignored the variation, and then the variation in A actually drove your residual down, right? So that's what this – this is called luck. Possibly stupid luck, I think would be the right attribute to add in front of it.

If you wiggled – if you played with ticking off regularization, you get a tighter distribution, and we wiggled with it for it to get the tightest one we could, and it was that. Look at that – it's taunting me. Okay. The robustly squares gives you this, right? So – and all this is to show is that actually all of this stuff in solving this SDP, you’ve actually done something that in fact 15 years ago no one had a clue was possible to do. So that's actually the only point here.

Okay, so we'll quit here.

[End of Audio]

Duration: 78 minutes