PATRICK WINSTON: So where are we?

We started off with simple methods for learning stuff.

Then, we talked a little about a purchase of learning that we're vaguely inspired by.

The fact that our heads are stuffed with neurons, and that we seemed to have evolved from primates.

Then, we talked about looking at the problem and address the issue of [? phrenology ?] and how it's possible to learn concepts.

But now, we're coming full circle back to the beginning and thinking about how to divide up a space with decision boundaries.

But whereas, you do it with a neural net or a nearest neighbors or a ID tree.

Those are very simple ideas that work very often.

Today, we're going to talk about a very sophisticated idea that still has a implementation.

So this needs to be in the tool bag of every civilized person.

This is about support vector machines, an idea that was developed.

Well, I want to talk to you today about how ideas develop, actually.

Because you look at stuff like this in a book, and you think, well, Vladimir Vapnik just figured this out one Saturday afternoon when the weather was too bad to go outside.

That's not how it happens.

It happens very differently.

I want to talk to you a little about that.

The next thing about great things that were done by people who are still alive is you can ask them how they did it.

You can't do that with Fourier.

You can't say to Fourier, how did you do it?

Did you dream it up on a Saturday afternoon?

But can call Vapnik on the phone and ask him questions.

That's the stuff I'm going to talk about toward the end of the hour.

Well, it's all about decision boundaries.

And now, we have several techniques that we can use to draw some decision boundaries.

And here's the same problem.

And if we drew decision boundaries in here, we might get something that would look like maybe this.

If we were doing a nearest neighbor approach, and if we're doing ID trees, we'll just draw in a line like that.

And if we're doing neural nets, well, you can put in a lot of straight lines wherever you like with a neural net, depending on how it's trained up.

Or if you just simply go in there and design it, so you could do that if you wanted.

And you would think that after people have been working on this sort of stuff for 50 or 75 years that there wouldn't be any tricks in the bag left.

And that's when everybody got surprised, because around the early '90s Vladimir Vapnik introduced the ideas I'm about to talk to you about.

So what Vapnik says is something like this.

Here you have a space, and you have some negative examples, and you have some positive examples.

How do you divide the positive examples from the negative examples?

And what he says that we want to do is we want to draw a straight line.

But which straight line is the question.

Well, we want to draw a straight line.

Well, would this be a good straight line?

One that went up like that?

Probably not so hot.

How about one that's just right here?

Well, that might separate them, but it seems awfully close to the negative examples.

So maybe what we ought to do is we ought to draw our straight line in here, sort of like this.

And that line is drawn with a view toward putting in the widest street that separates the positive samples from the negative samples.

That's why I call it the widest street approach.

So that makes way of putting in the decision boundary-- is to put in a straight line but in contrast with the way ID tree puts in a straight line.

It tries to put the line in in such a way as the separation between the positive and negative examples.

That street is as wide as possible.

All right.

So you might think to do that in the UROP project, and then, let it go with that.

What's the big deal?

So what we've got to do is we've got to go through why it's a big deal.

So first of all, we like to think about how you would make a decision rule that would use that decision boundary.

So what I'm going to ask you to imagine is that we've got a vector of any length that you like, constrained to be perpendicular to the median, or if you like, perpendicular to the gutters.

It's perpendicular to the median line of the street.

All right, it's drawn in such a way that that's true.

We don't know anything about it's length, yet.

Then, we also have some unknown, say, right here.

And we have a vector that points to it by excel.

So now, what we're really interested in is whether or not that unknown is on the right side of the street or on the left side of the street.

So what we'd what to do is want to project that vector, u, down on to one that's perpendicular to the street.

Because then, we'll have the distance in this direction or a number that's proportional to this in this direction.

And the further out we go, the closer we'll get to being on the right side of the street, where the right side of the street is not the correct side but actually the right side of the street.

So what we can do is we can say, let's take w and dot it with u and measure whether or not that number is equal to or greater than some constant, c.

So remember that the dot product has taken the projection onto w.

And the bigger that projection is, the further out along this line the projection will lie.

And eventually it will be so big that the projection crosses the median line of the street, and we'll say it must be a positive sample.

Or we could say, without loss of generality that the dot product plus some constant, b, is equal to or greater than 0.

If that's true, then it's a positive sample.

So that's our decision rule.

And this is the first in several elements that we're going to have to line up to understand this idea called support vector machines.

So that's the decision rule.

And the trouble is we don't know what constant to use, and we don't know which w to use either.

We know that w has to be perpendicular to the median line of the street.

But there's lot of w's that are perpendicular to the median line of the street, because it could be of any length.

So we don't have enough constraint here to fix a particular b or a particular w.

Are you with me so far?

All right.

And this, by the way, we get just by saying that c equals minus b.

What we're going to do next is we're going to lay on some additional constraints whether you're toward putting enough constraint on the situation that we can actually calculate a b and a w.

So what we're going to say is this, that if we look at this quantity that we're checking out to be greater than or less than 0 to make our decision, then, what we're going to do is we're going to say that if we take that vector w, and we take the dot product of that with some x plus, some positive sample, now.

This is not an unknown.

This is a positive sample.

If we take the dot product of those two vectors, and we had b just like in our decision rule, we're going to want that to be equal to or greater than 1.

So in other words, you can be an unknown anywhere in this street and be just a little bit greater or just a little bit less than 0.

But if you're a positive sample, we're going to insist that this decision function gives the value of one or greater.

Likewise, if w thought it was some negative sample is provided to us, then we're going to say that has to be equal to or less than minus 1.

All right.

So if you're a minus sample, like one of these two guys or any minus sample that may lie down here, this function that gives us the decision rule must return minus 1 or less.

So there's a separation of distance here.

Minus 1 to plus 1 for all of the samples.

So that's cool.

But we're not quite done, because carrying around two equations like this, it's a pain.

So what we're going to do is we're going to introduce another variable to make like a little easier.

Like many things that we do, and when we develop this kind of stuff, introducing this variable is not something that God says has to be done.

What is it?

We introduced this additional stuff to do what?

To make the mathematics more convenient, so mathematical convenience.

So what we're going to do is we're going to introduce a variable, y sub i, such that y sub i is equal to plus 1 for plus samples and minus 1 for negative samples.

All right.

So for each sample, we're going to have a value for this new quantity we've introduced, y.

And the value of y is going to be determined by whether it's a positive sample or negative sample.

If it's a positive sample it's got to be plus 1 for this situation up here, and it's going to be minus 1 for this situation down here.

So what we're going to do with this first equation is we're going to multiply it by y sub i, and that is now x of i, plus b is equal to or greater than 1.

And then, you know what we're going to do?

We're going to multiply the left side of this equation by y sub i, as well.

So the second equation becomes y sub i times x sub i plus b.

And now, what does that do over here?

We multiplied this guy times minus 1.

So it used to be the case that that was less than minus 1.

So if we multiply it by minus 1, then it has to be greater than plus 1.

The two equations are the same, because that introduces this little mathematical convenience.

So now, we can say that y sub i times x sub i plus b.

Well, what we're going to do-- Brett?

STUDENT: What happened to the w?

PATRICK WINSTON: Oh, did I leave out a w?

I'm sorry.

Thank you.

Yeah, I wouldn't have gotten very far with that.

So that's dot it with w, dot it with w.

Thank you, Brett.

Those are all vectors.

I'll pretty soon forget to put the little vector marks on there, but you know what I mean.

So that's w plus b.

And now, let me bring that 1 over to the left side, and that's equal to or greater than 0.

All right.

With Brett's correction, I think everything's OK.

But we're going to take one more step, and we're going to say that y sub i times x sub i times w plus b minus 1.

It's always got to be equal to or greater than 0.

But what I'm going to say is if we're for x sub i in a gutter.

So there's always going to be greater than 0, but we're going to add the additional constraint that it's going to be exactly 0 for all the samples that end up in the gutters here of the street.

So the value of that expression is going to be exactly 0 for that sample, 0 for this sample and this sample, not 0 for that sample.

It's got to be greater than 1.

All right?

So that's step number two.

And this is step number one.

OK.

So now, we've just got some expressions to talk about, some constraints.

Now, what are we trying to do here?

I forgot.

Oh, I remember now.

We're trying to figure out how to arrange for the line to be such at the street separating the pluses from the minuses as wide as possible.

So maybe we better figure out how we can express the distance between the two gutters.

Let's just repeat our drawing.

We've got some minuses here, got pluses out here, and we've got gutters that are going down here.

And now, we've got a vector here to a minus, and we've got a vector here to a plus.

So we'll call that x plus and this x minus.

So what's the width of the street?

I don't know, yet.

But what we can do is we can take the difference of those two vectors, and that will be a vector that looks like this, right?

So that's x plus minus x minus.

So now, if I only had a unit normal that's normal to the median line of the street, if it's a unit normal, then I could just take the dot product or that unit normal and this difference vector, and that would be the width of the street, right?

So in other words, if I had a unit vector in that direction, then I could just dot the two together, and that would be

the width of the street.

So let me write that down before I forget.

So the width is equal to x plus minus x minus.

OK.

That's the difference vector.

And now, I've got to multiple it by unit vector.

But wait a minute.

I said that that w is a normal, right?

The w is a normal.

So what I can do is I can multiply this times w, and then, we'll divide by the magnitude of w, and that will make it a unit vector.

So that dot product, not a product, that dot product is, in fact, a scalar, and it's the width of the street.

It doesn't do as much good, because it doesn't look like we get much out of it.

Oh, but I don't know.

Let's see, what can we get out of it?

Oh gee, we've got this equation over here, this equation that constrains the samples that lie in the gutter.

So if we have a positive sample, for example, then this is plus 1, and we have this equation.

So it says that x plus times w is equal to, oh, 1 minus b.

See, I'm just taking this part here, this vector here, and I'm dotting it with x plus.

So that's this piece right here.

y is 1 for this kind of sample.

So I'll just take the 1 and the b back over to the other side, and I've got 1 minus b.

OK?

Well, we can do the same trick with x minus.

If we've got a negative sample, then y sub i is negative.

That gives us our negative w times dot over x sub i.

But now, we take this stuff back over to the right side, and we get 1 plus b.

So that all licenses to rewrite this thing as 2 over the magnitude of w.

How did I get there?

Well, I decided I was going to enforce this constraint.

I noted that the width of the street has got to be this difference vector times a unit vector.

Then, I used the constraint to plug back some values here.

And I discovered to my delight and amazement that the width of the street is 2 over the magnitude of w.

Yes, Brett?

STUDENT: So your first x plus is minus b, and x minus is 1 plus b.

PATRICK WINSTON: Yeah.

STUDENT: So you're subtracting it?

PATRICK WINSTON: Let's see.

If I've got a minus here, then that makes that minus, and then, the b is minus, and when I take the b over to the other side it becomes plus.

STUDENT: Yeah, so if you subtract the left with the right [INAUDIBLE].

PATRICK WINSTON: No.

No, sorry.

This expression here is 1 plus b.

Trust me it works.

I haven't got my legs all tangled up like last Friday, well, not yet, anyway.

It's possible.

There's going to be a lot of algebra here eventually.

So this quantity here, this is miracle number three.

This quantity here is the width of the street.

And what we're trying to do is we're trying to maximize that, right?

So we want to maximize 2 over the magnitude of w if we're to get the widest street under the constraints that we've decided that we're going to work with.

All right.

So that means that it's OK to maximize 1 over w, instead.

We just drop the constant.

And that means that it's OK to minimize the magnitude of w, right?

And that means that it's OK to minimize 1/2 times the magnitude of w squared.

Right, Brett?

Why did I do that?

Why did I multiply by 1/2 and square it?

STUDENT: Because it's mathematically convenient.

PATRICK WINSTON: It's mathematically convenient.

Thank you.

So this is point number three in the development.

So where do we go?

We decided that was going to be our decision rule.

We're going to see which side of the line we're on.

We decided to constrain the situation, so the value of the decision rule is plus 1 in the gutters for the positive samples and minus 1 in the gutters for the negative samples.

And then, we discovered that maximizing the width of the street led us to an expression like that, which we wish to maximize.

Should we take a break?

Should we get coffee?

Too bad, we can't do that in this kind of situation.

But we would if we could.

And I'm sure when Vapnik got to this point, he went out for coffee.

So now, we back up, and we say, well, let's let these expressions start developing into a song.

Not like that, that's vapid, speaking of Vapnik.

What song is it going to sing?

We've got an expression here that we'd like to find the minimum of, the extremum of.

And we've got some constraints here that we would like to honor.

What are we going to do?

Let me put what we're going to do to you in the form of a puzzle.

Is it got something to do with Legendre?

Has it got something to do with Laplace?

Or does it have something to do with Lagrange?

She says Lagrange.

Actually, all three were said to be on Fourier's Doctoral Defense Committee-- must have been quite an example.

But we want to talk about Lagrange, because we've got a situation here.

Is this 1801?

1802?

1802.

We learned in 1802 that if we going to find the extremum of a function with constraints, then we're going to have to use Lagrange multipliers.

That would give us a new expression, which we can maximize or minimize without thinking about the constraints anymore.

That's how Lagrange multipliers work.

So this brings us to miracle number four, developmental piece number four.

And it works like this.

We're going to say that L-- the thing we're going to try to maximize in order to maximize the width of the street-- is equal to 1/2 times the magnitude of that vector, w, squared minus.

And now, we've got to have a summation over all the constraints.

And each or those constraints is going to have a multiplier, alpha sub i.

And then, we write down the constraint.

And when we write down a constraint, there it is up there.

And I've got to be hyper careful here, because, otherwise, I'll get lost in the algebra.

So the constraint is y sub i times vector, w, dotted with vector x sub i plus b, and now, I've got a closing parenthesis, a minus 1.

That's the end of my constraint, like so.

I sure hope I've got that right, because I'll be in deep trouble if that's wrong.

Anybody see any bugs in that?

That looks right. doesn't it?

We've got the original thing we're trying to work with.

Now, we've got Lagrange multipliers all multiplied.

It's back to that constraint up there, where each constraint is constrained to be 0.

Well, there's a little bit of mathematical slight of hand here, because in the end, the ones that are going to be 0, the Lagrange multipliers here.

The ones that are going to be non 0 are going to be the ones connected with vectors that lie in the gutter.

The rest are going to be 0.

But in any event, we can pretend that this is what we're doing.

I don't care whether it's a maximum or minimum.

I've lost track.

But what we're going to do is we're going to try to find an extremum of that.

So what do we do?

What does 1801 teach us about?

Finding the maximum-- well, we've got to find the derivatives and set them to 0.

And then, after we've done that, a little bit of that manipulation, we're going to see a wonderful song start to emerge.

So let's see if we can do it.

Let's take the partial of L, the Lagrangian, with respect to the vector, w.

Oh my God, how do you differentiate with respect to a vector?

It turns out that it has a form that looks exactly like differentiating with respect to a scalar.

And the way you prove that to yourself is you just expand everything in terms of all of the vector's components.

You differentiate those with respect to what you're differentiating with respect to, and everything turns out the same.

So what you get when you differentiate this with respect to the vector, w, is 2 comes down, and we have just magnitude of w.

Was it the magnitude of w?

Yeah, like so.

Was it the magnitude of w?

Oh, it's not the magnitude of w.

It's just w, like so, no magnitude involved.

Then, we've got a w over here, so we've got to differentiate this part with respect to w, as well.

But that part's a lot easier, because all we have there is a w.

There's no magnitude.

It's not raised to any power.

So what's w multiplied by?

Well, it's multiplied by x and y sub i and alpha sub i.

All right.

So that means that this expression, this derivative of the Lagrangian, with respect to w is going to be equal to w minus the sum of alpha sub i, y sub i, x sub i, and that's got to be set to 0.

And that implies that w is equal to the sum of some alpha i, some scalars, times this minus 1 or plus 1 variable times x sub i over i.

And now, the math is beginning to sing.

Because it tells us that the vector w is a linear sum of the samples, all the samples or some of the sample.

It didn't have to be that way.

It could have been raised to a power.

It could have been a logarithm.

All sorts of horrible things could have happened when we did this.

But when we did this, we discovered that w is going to be equal to a linear some of these vectors here.

Some of the vectors in the sample set, and I say some, because for some alpha will be 0.

All right.

So this is something that we want to take note of as something important.

Now, of course, we've got to differentiate L with respect to anything else it might vary, so we've got to differentiate L with respect to b, as well.

So what's that going to be equal to?

Well, there's no b in here, so that makes no contribution.

This part here doesn't have a b in it, so that makes no contribution.

There's no b over here, so that makes no contribution.

So we've got alpha i times y sub i times b.

That has a contribution.

So that's going to be the sum of alpha i times y sub i.

And then, we're differentiating with respect to b, so that disappears.

There's a minus sign here, and that's equal to 0, or that implies that the sum of the alpha i times y sub i is equal to 0.

Hm, that looks like that might be helpful somewhere.

And now, it's time for more coffee.

By the way, these coffee periods take months.

You stare at it.

You work on something else.

You've got to worry about your finals.

And you think about it some more.

And eventually, you come back from coffee and do the next thing.

Oh, what is the next thing?

Well, we've still got this expression that we're trying to find the minimum for.

And you say to yourself, this is really a job for the numerical analysts.

Those guys know about this sort of stuff.

Because of that little power in there, that square.

This is a so-called quadratic optimization problem.

So at this point, you would be inclined to hand this problem over to a numerical analysts.

They'll come back in a few weeks with an algorithm.

You implement the algorithm.

And maybe things work.

Maybe they don't converge.

But any case, you don't worry about it.

But we're not going to do that, because we want to do a little bit more math, because we're interested in stuff like this.

We're interested in the fact that the decision vector is a linear sum of the samples.

So we're going to work a little harder on this stuff.

And in particular, now that we've got an expression for w, this one right here, we're going to plug it back in there, and we're going to plug it back in here and see what happens to that thing we're trying to find the extremum of.

Is everybody relaxed, taking deep breath?

Actually, this is the easiest part.

This is just doing a little bit of the algebra.

So the think we're trying to maximize or minimize is equal to 1/2.

And now, we've got to have this vector here in there twice.

Right?

Because we're multiplying the two together.

So let's see.

We've got from that expression up there, one of those w's will just be the sum of the alpha i times y sub i times the vector x sub i.

And then, we've got the other one, too.

So that's just going to be the sum of alpha.

Now, I'm going to, actually, eventually, squish those two sums together into a double summation, so I have to keep the indexes straight.

So I'm just going to write that as alpha sub j, y sub j, x sub j.

So those are my two vectors and I'm going to take the dot product of those.

That's the first piece, right?

Boy, this is hard.

So minus, and now, the next term looks like alpha i, y sub i, x sub i times w.

So you've got a whole bunch of these.

We've got a sum of alpha i times y sub i times x sub i, and then, that gets multiplied times w.

So we'll put this like this, the sum of alpha j, y sub j, x sub j in there like that.

And then, that's the dot product like that.

That wasn't as bad as I thought.

Now, I've got to deal with the next term, the alpha i times y sub i times b.

So that's minus sub of alpha i times y sub i times b.

And then, to finish it off, we have plus the sum of alpha sub i minus 1 up there, minus 1 in front of the summation, such as the sum of the alphas.

Are you with me so far?

Just a little algebra.

It looks good.

I think I haven't mucked it, yet.

Let's see.

alpha i times y sub i times b. b is a constant.

So pull that out there, and then, I just got the sum of alpha sub i times y sub i.

Oh, that's good.

That's 0.

Now, so for every one of these terms, we dot it with this whole expression.

So that's just like taking this thing here and dotting those two things together, right?

Oh, but that's just the same thing we've got here.

So now, what we can do is we can say that we can rewrite this Lagrangian as-- we've got that sum of alpha i.

That's the positive element.

And then, we've got one of these and half of these.

So that's minus 1/2.

And now, I'll just convert that whole works into a double sum over both i and j of alpha i times alpha j times y sub i

times y sub j times x sub i dotted with x of j.

We sure went through a lot of trouble to get there, but now, we've got it.

And we know that what we're trying to do is we're trying to find a maximum of that expression.

And that's the one we're going to had off to the numerical analysts.

So if we're going to had this off to the numerical analysts anyway, why did I go to all this trouble?

Good question.

Do you have any idea why I went to all this trouble?

Because I wanted to find out the dependence of this expression.

Wanda is telling me.

I'm translating as I go.

She's telling me in Romanian.

I want to find what this maximization depends on with respect these vectors, the x, the sample vectors.

And what I've discovered is that the optimization depends only on the dot product of pairs of samples.

And that's something we want to keep in mind.

That's why I put it in royal purple.

Now, up here, so let's see.

What do we call that one up there?

That's two.

I guess, we'll call this piece here three.

This piece here is four.

And now, there's one more piece.

Because I want to take that w, and not only stick it back into that Lagrangian, I want to stick it back into the

decision rule.

So now, my decision rule with this expression for w is going to be w plugged into that thing.

So the decision rule is going to look like the sum of alpha i times y sub i times x sub i dotted with the unknown vector, like so.

And we're going to, I guess, add b.

And we're going to say, if that's greater than or equal to 0, then plus.

So you see why the math is beginning to sing to us now.

Because now, we discover that the decision rule, also, depends only on the dot product of those sample vectors and the unknown.

So the total of dependence of all of the math on the dot products.

All right.

And now, I hear a whisper.

Someone is saying, I don't believe that mathematicians can do it.

I don't think those numerical analysts can find the optimization.

I want to be sure of it.

Give me ocular proof.

So I'd like to run a demonstration of it.

OK.

There's our sample problem.

The one I started the hour out with.

Now, if the optimization algorithm doesn't get stuck in a local maximum or something, it should find a nice, straight line separating those two guys to finding the widest street between the minuses and the pluses.

So in just a couple of steps, you can see down there in step 11.

It's decided that it's done as much as it can on the optimization.

And it's got three alphas.

And you can see that the two negative samples both figure into the solution, the weights on the Lagrangian multipliers are given by those little yellow bars.

So the two negatives participate in the solution as one of the positives, but the other positive doesn't.

So it has a 0 weight.

So everything worked out well.

Now, I said, as long as it doesn't get stuck on a local maximum, guess what, those mathematical friends of ours can tell us and prove to us that this thing is a convex space.

That means it can never get stuck in a local maximum.

So in contrast with things like neural nets, where you have a plague of local maxima, this guy never gets stuck in a local maxima.

Let's try some other examples.

Here's two vertical points-- no surprises there, right?

Well, you say, well, maybe it can't deal with diagonal points.

Sure it can.

How about this thing here?

Yeah, it only needed two of the points since any two, a plus or minus, will define the street.

Let's try this guy.

Oh.

What do you think?

What happened here?

Well, we're screwed, right?

Because it's linearly inseparable-- bad news.

So in situations where it's linearly inseparable, the mechanism struggles, and eventually, it will just slow down and you truncate it, because it's not making any progress.

And you see the red dots there are ones that it got wrong.

So you say, well, too bad for our side-- doesn't look like it's all that good anyway.

But then, a powerful idea comes to the rescue, when stuck switch to another perspective.

So if we don't like the space that we're in, because it gives examples that are not linearly separable, then we can say, oh, shoot.

Here's our space.

Here are two points.

Here are two other points.

We can't separate them.

But if we could somehow get them into another space, maybe we can separate them, because they look like this in the other space, and they're easy to separate.

So what we need, then, is a transformation that will take us from the space we're in into a space where things are more convenient, so we're going to call that transformation phi with a vector, x.

That's the transformation.

And now, here's the reason for all the magic.

I said, that the maximization only depends on dot products.

So all I need to do the maximization is the transformation of one vector dotted with the transformation of another vector, like so.

That's what I need to maximize, or to find the maximum on.

Then, in order to recognize-- where did it go?

Underneath the chalkboard.

Oh, yes.

Here it is.

To recognize, all I need is dot products, too.

So for that one I need phi of x dotted with phi of u.

And just to make this a little bit more consistent, the notation, I'll call that x j and this x sub i.

And that's x sub i.

Those are the quantities I need in order to do it.

So that means that if I have a function, let's call it k of x sub i and x sub j, that's equal to phi of x sub i dotted with phi of x sub j.

Then, I'm done.

This is what I need.

I don't actually need this.

All I need is that function, k, which happens to be called a kernel function, which provides me with the dot product of those two vectors in another space.

I don't have to know the transformation into the other space.

And that's the reason that this stuff is a miracle.

So what are some of the kernels that are popular?

One is the linear kernel that says that u dotted with v plus 1 to the n-th is such a kernel, because it's got u in it and v in it, the two vectors.

And this is what the dot product is in the other space.

So that's one choice.

Another choice is a kernel that looks like this, e to the minus.

Let's take the dot product of the difference of those two guys.

Let's take the magnitude of that and divide it by some sigma.

That's a second kind of kernel that we can use.

So let's go back and see if we can solve this problem by transforming it into another space where we have another perspective.

So that's it.

That's another kernel.

And so sure, we can.

And that's the answer when transformed back into the original space.

We can also try doing that with a so-called radial basis kernel.

That's the one with the exponential in it.

We can learn on that one.

Boom.

No problem.

So we've got a general method that's convex and guaranteed to produce a global solution.

We've got a mechanism that easily allows us to transform this into another space.

So it works like a charm.

Of course, it doesn't remove all possible problems.

Look at that exponential thing here.

If we choose a sigma that is small enough, then those sigmas are essentially shrunk right around the sample points, and we could get overfitting.

So it doesn't immunize us against overfitting, but it does immunize us against local maxima and does provide us with a general mechanism for doing a transformation into another space with a better perspective.

Now, the history lesson, all this stuff feels fairly new.

It feels like it's younger than you are.

Here's the history of it.

Vapnik immigrated from the Soviet Union to the United States in about 1991.

Nobody ever heard of this stuff before he immigrated.

He actually had done this work on the basic support vector idea in his Ph.D. thesis at Moscow University in the early '60s.

But it wasn't possible for him to do anything with it, because they didn't have any computers they could try anything out with.

So he spent the next 25 years at some oncology institute in the Soviet Union doing applications.

Somebody from Bell Labs discovers him, invites him over to the United States where, subsequently, he decides to immigrate.

In 1992, or thereabouts, Vapnik submits three papers to NIPS, the Neural Information Processing Systems journal.

All of them were rejected.

He's still sore about it, but it's motivating.

So around 1992, 1993, Bell Labs was interested in hand-written character recognition and in neural nets.

Vapnik thinks that neural nets-- what would be a good word to use?

I can think of the vernacular, but he thinks that they're not very good.

So he bets a colleague a good dinner that support vector machines will eventually do better at handwriting recognition then neural nets.

And it's a dinner bet, right?

It's not that big of deal.

But as Napoleon said, it's amazing what a soldier will do for a bit of ribbon.

So that makes colleague, who's working on this problem with handwritten recognition, decides to try a support vector machine with a kernel, in which n equals 2, just slightly nonlinear, works like a charm.

Was this the first time anybody tried a kernel?

Vapnik actually had the idea in his thesis but never though it was very important.

As soon as it was shown to work in the early '90s on the problem handwriting recognition, Vapnik resuscitated the idea of the kernel, began to develop it, and became an essential part of the whole approach of using support vector machines.

So the main point about this is that it was 30 years in between the concept and anybody ever hearing about it.

It was 30 years between Vapnik's understanding of kernels and his appreciation of their importance.

And that's the way things often go, great ideas followed by long periods of nothing happening, followed by an epiphanous moment when the original idea seemed to have great power with just a little bit of a twist.

And then, the world never looks back.

And Vapnik, who nobody ever heard of until the early '90s, becomes famous for something that everybody knows about today who does machine learning.