

Regression

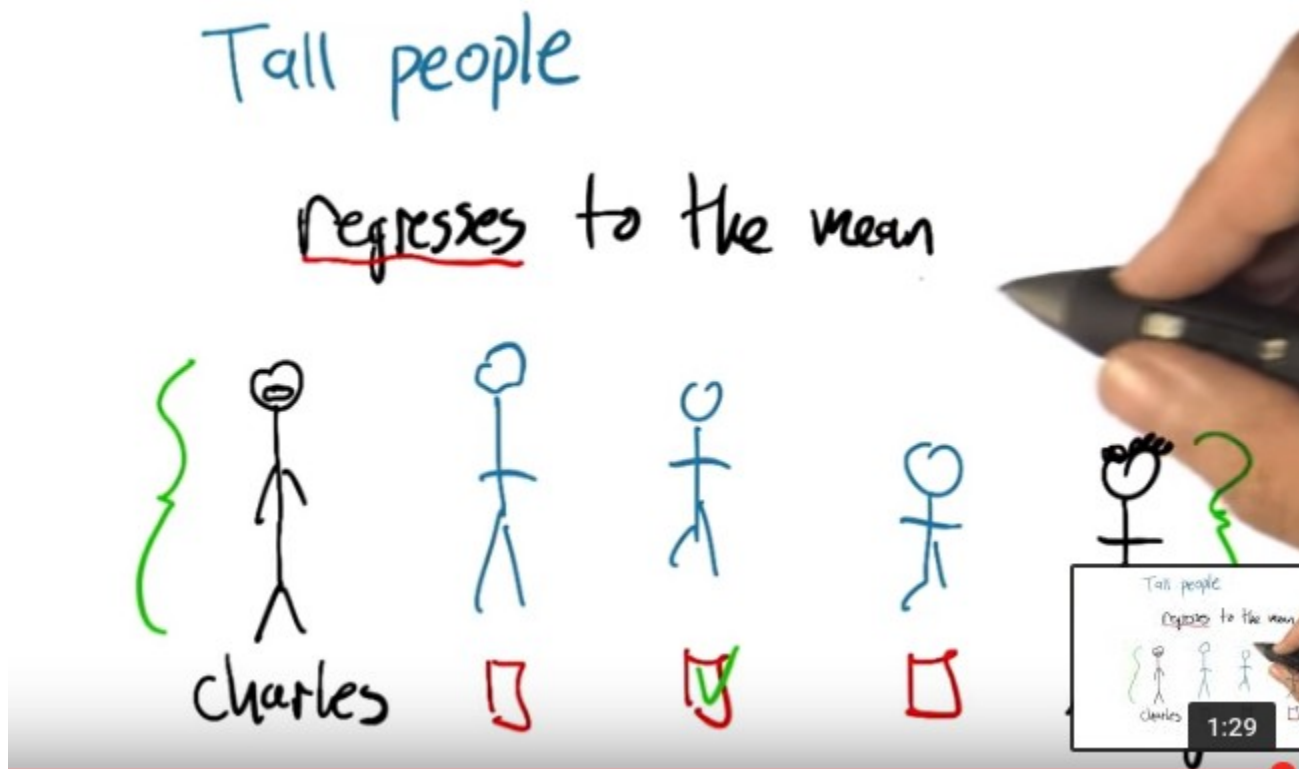
Supervised learning: Take examples of inputs and outputs. Now, given a new input, predict its output.

Mapping continuous inputs to outputs.



discrete, continuous

1. >> Hey Charles, how you doing? >> I'm doing just fine Michael, how are you doing? >> I'm doing pretty well, thanks. I'm happy to get a chance to tell you about something today. >> Excellent, and what is it you're going to tell me about? >> We're going to talk about regression. >> Like, progression? [LAUGH]. >> No, regression. So, let me tell you about regression. So we are, in this section of the class, talking about supervised learning. And Supervised learning, in supervised learning we can take examples of inputs and outputs and based on that we are going to be able to take a new input and predict the corresponding output for that input, right. So this, this covers all of this, this the things we are talking about in the context of supervised learning, right. >> Right. >> Now, what makes regression special subtopic. We are going to be talking about mapping continuous inputs to outputs. As opposed to, what was the other thing that we were mapping, what other kinds of outputs did we think about? >> Well, we had discrete outputs and continuous outputs. >> Right, and so this is going to be the focus on continuous. (從此處 直到第 3 段結束, 都在討論 regression 這個單詞在 ML 中為何不是它本來的意思. 尼瑪.) So regression seems like sort of an odd word. It doesn't really kind of fit for this. So often I think about regression as. So this is, this is me being all sad and sort of reverting back to a childhood state. And that's, you know, that's in the psychological sense, that's what regression refers to. But it turns out that, that's not what it means in this setting. But the story by which those things became linked, I think, is kind of interesting. So let me tell you about that. Okay. So, this is a picture of you Charles. >> [LAUGH] Okay. I'll accept that. >> You can tell it's you because he's really tall. And you're, you're a fairly tall man. I know you don't think of yourself that way, but you think of everyone else as being short which is really the same thing. >> Fair enough. >> Alright, so let's say that this is Charles. Let's say that this is someone of average height. Just someone at random. >> Mm-hmm. >> So now, let's pretend, Charles, that you have children. >> I do have children. >> All right. So let's, that's okay but we can just pretend, and we want to ask the question what would you expect the average height of your children to be? Would you expect it to be sort of, you know, sort of Charles' height? Or average height or may be somewhere between. So let's let's actually ask this as a quiz.

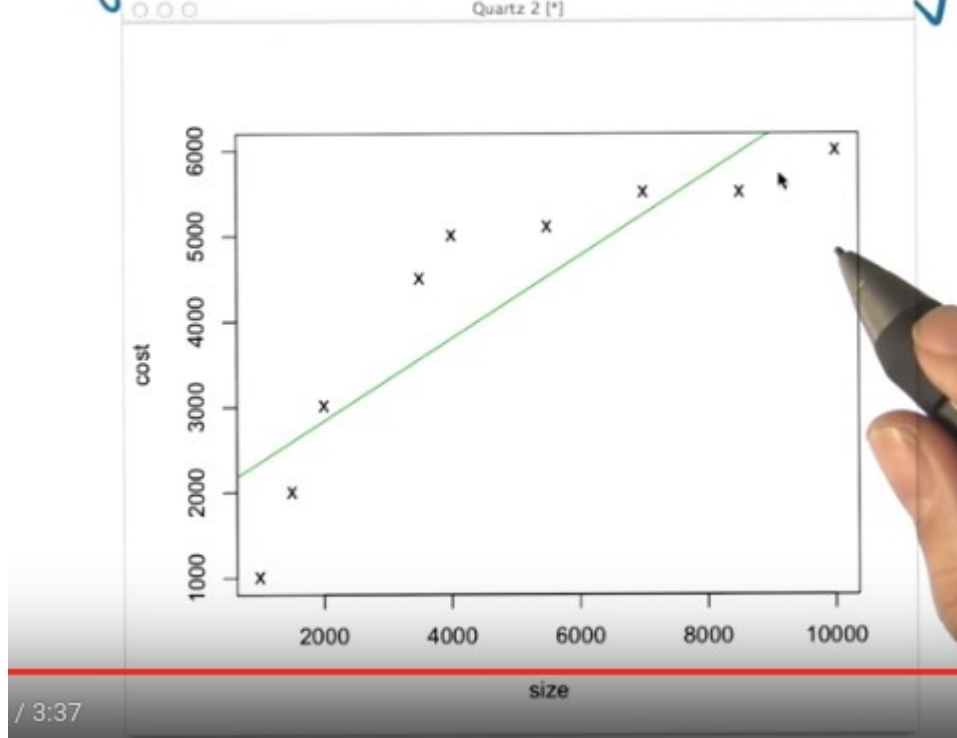


2. Okay, Charles, so what do you, what do you think about this? >> Wait, how old are my children? >> Let's say what their adult height is going to be. >> I would expect them to be a little bit smaller than me. >> A little bit smaller than you? >> Mm-hmm. >> So, so their, the average height, their average height of your children, you would you say it would be like an average height person? Or like your height or sort of in between? >> In between. >> In between. All right. So it turns out that if you actually do this, you measure people's heights and you measure the heights of their children, that that is in fact what you tend to see. That very, **very tall people**, like you **tend to have taller than average children**. But the height is between. It actually **regresses to the mean**. And here we really do mean **regresses** in the sense of going back to this kind of more primitive state that, if you think about average height people, as being like your ancestors, then, you know, you as a, as a very tall person tend to have kids that that **tend regress back toward that average value** that sort of, more older, more ancient value. So does that that make some sense to you? >> That makes some sense. But one comment and one question. Comment, that is awesome because I've always actually wondered what **regresses to the mean** actually means. The second, what prevents us from all being the same height then? >> Yes, so what, what seems to be happening is that **there's a kind of a noisy process and some people turn out to be taller, but then the, then the next generation there's a little bit of a history effect in people stay taller, but it tends to drift back towards the mean**. So it's, so it's, it's sort of like a random walk, to some extent. >> Oh, that actually kind of makes sense.

3. Alright, so what does this have to do with function approximation or regression. So how does this notion of regression of falling back toward the mean have to do with this idea of approximating functions, mapping inputs to outputs, it seems kind of odd. So it turns out that the relationship is, here's the, here's the connection between them. I'm going to draw a graph and on this axis will be the parent height. And on this axis will be the average child height. So if we plot these against each other, let's let me put the mean up here. Let's say that this is mean height for the population. And now say that you know pair, we sort of imagine that parents of average height will have children of average height. But

parents that are really tall, like that hypothetical person from before, will have children that are taller than average but not as tall as themselves. And similarly people that are very let's say of smaller stature will have children that are also you know short. And, but not quite as short again closer to the mean. And it turns out that you have this, this very nice linear relationship between these quantities, and, there's an important aspect to this. Which is that the slope of this line is less than one, it's two thirds. Right. If the slope of this line was one, what would that mean Charles? >> That would mean that everybody's children had the, would, the same height of their parents. >> Right, right, and so that's exactly right, and so but if this slope is less than one, like it is, it turns out to be in, in real populations. Then what's happening is the children are little shorter than the parents. Children of taller parents are shorter than they are. And the children of short parents are taller than they are. And that's the fact that this is less than one is what makes it regression to the mean. Now this, this was worked out in I believe in the late 1800s and it was just such a beautiful way of connecting all these different quantities together. To kind of think of them as being related in this functional way. That people said, oh this is really great. I'm going to use this idea of regression. And what they started to mean actually was this not this idea of regression to the mean. But this idea of (regression 在 ML 中的意思:) finding a, a mathematical relationship based on a bunch of measurements of points. So this term ended up getting misused. But that's the term that we have now. So regression now refers to not this idea of collapsing back towards, towards the mean, but, the idea of using functional form to approximate a bunch of data points. Isn't that weird. >> That's pretty cool. >> There's another example of this sort of idea where where a reasonable word, like, like regression which we're referring to some physical thing in the, in the world due to experiments like psych experiments at this point became this mathematical concept where the name doesn't really fit anymore, like there isn't really anything regressing in what we're doing. >> Mm-hm. >> There's another, really important example that we're going to get to the later in the course. Do you, do you know what I'm thinking of Charles? >> No. >> So reinforcement learning is my field of study. And often your field of study. >> Often. >> Often. And it turns out that reinforcement learning doesn't mean what the words mean anymore. That this was a concept that the psychologist used to explain what they were observing. And then some mathematicians, well let's call them computer scientists, took the word themselves, started to use it and used it wrong, but now it stuck. [LAUGH] And regression is another example like that. They, the word is sort of being used wrong, but it stuck and that's what we're going to use. >> This explains why every time I have a conversation with a psychologist about reinforcement learning, we talk past each other. >> Yes, they get very confused. I tried it to tell them upfront that's not really what I mean, but I'm going to use the words anyway, but it still confuses them. >> Hmm.

Regression in Machine Learning



4. Alright, so, one of the things that's very helpful about regression is that in many ways it's very simple to visualize, it's very simple to think about what some of the issues are and all the various topics in machine learning that are really important to understand and sometimes are difficult concepts really do come up in a fairly easy to understand way. So what I'd like to do now is to step through an example of doing some regression and to point out what some of the pitfalls are and how they're generally handled in the machine learning context. So, this graph that I put up here, is, we just made these numbers up, but it's supposed to tell us a, a little bit about housing prices. So let's imagine that we're off to buy a house and what we notice is that there's lots of different houses on the market, and there are lots of different sizes, right. So, the square footage of the house can vary. And in this case the houses that I visited can be between, about 1,000 to 10,000 square feet. And of course, as you get bigger houses, you tend to get more, the, the prices tend to go up, too. Alright, so the price that the house cost is, tends to rise with the size of the house. So, what I've done here is I've plotted as a little x say a set of nine houses that I've observed. Start off over here with a house that's a 1,000 square feet and cost a \$1,000? I don't know what year this happened in. And we end up with a house that is 10,000 square feet and cost about \$6,000. Again, I don't. This is not true in Providence Rhode Island, I'll tell you that. >> Are you sure? >> Yeah, I'm pretty sure. >> Yeah, it's really not true in Atlanta Georgia. >> So Alright... So, so, so imagine that this is the relationship we observe. But now we want to answer a question like, Well, what happens if we find a house on the market and it's about \$5,000, what do you think a fair price for that would be? So what do you, what do you think, Charles? Looking at this, what do you think a fair price for a 5,000 square foot house would be? >> Apparently about \$5,000. >> About, \$5,000. Right. So, how did you do that? >> I looked at the graph, I went over to 5,000 square feet at the x-axis and I went up. Until I found, where one of the x's was on the y axis and I said, oh, that's about 5,000 square feet. >> Well, but there was no corresponding point for that, so you had to interpolate or something

,uh, based on the points that were there you had to kind of imagine what might, might be happening at the 5,000 square foot mark, right? >> That's true, although this one was a little easy because at 4,000 and 6,000 square feet, they were almost exactly the same. >> Mm, and so that, to you, made it feel like there was probably ,um, that's probably the level where things in this range would be. >> Yeah. >> Okay. Alright, that seems kind of reasonable. So sure, though what we're going to do in this case is actually try to find a, a function that fits this. >> Mm-hm. >> Alright ,so what we can do is actually say, well what if there is a linear relationship. What would be the best linear function that captures the relationship between the size and the cost. So ,what I have here is, it turns out of all the possible linear functions, this is the one that minimizes the squared error, the squared deviation, between these x points and the corresponding position on green line. So it finds a way of balancing all those different errors against each other and that's the best line we've got. Now in this particular case, it's interesting right, because if you put your idea of 5,000 square feet. Look what this line predicts. It's something more like \$4,000, right. Do you see that? >> I do. That is doesn't seem right to me. >> It doesn't, yeah, it doesn't really look like a very good fit. But it does at least capture the fact that there is increasing cost with, with increase in size. >> That's true.

Finding the Best Line

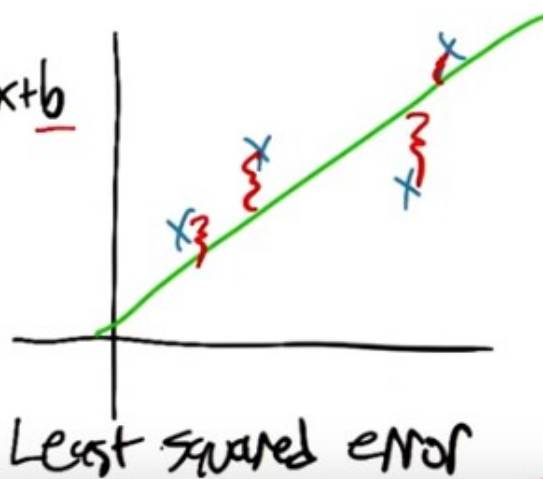
☐ Hill climbing

☒ Calculus

☐ Random Search

☐ Ask a physicist

$$y = \underline{m}x + \underline{b}$$



5. Alright. So it's worth asking, how do we find this best line? So again there's an infinite number of lines. How do we find one that fits these points the best. And again we're defining best fit. As the one that has the least squared error, where the error is going to be some of the distances between these x points and the green line that we, that we fit. I'm not even sure that this really is the best fit in this case. I just kind of hand drew it. So, okay. So is this something that we, that we would want to solve by hill climbing which is to say, we kind of pick the. The slope and the intercept of the line, and we just kind of try different values of this until it gets better and better and then we can't get any better, and we stop. Can we do this using calculus? Can we use random search, where we just like, pick a random M, pick a random B, and see if we're happy with it? Or is this the sort of thing where we probably would just go and ask a physicist because it involves, like, continuous quantities, and we're discrete people?

Finding the best constant function

X	y
x_1	y_1
x_2	y_2
\vdots	\vdots
x_n	y_n

$$f(x) = c$$

$$E(c) = \sum_{i=1}^n (y_i - c)^2$$

$$\frac{dE(c)}{dc} = \sum_{i=1}^n 2(y_i - c) \cdot (-1)$$

LOSS
ERROR

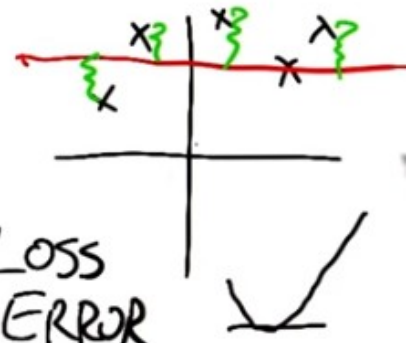
$$+ \sum_{i=1}^n 2(y_i - c) = 0$$

$$\sum_{i=1}^n y_i - \sum_{i=1}^n c = 0$$

$$n \cdot c = \sum_{i=1}^n y_i$$

$$c = \frac{\sum_{i=1}^n y_i}{n}$$

MEAN



6. And that's the correct answer. All right. So let's actually go through that exercise and derive how we do that. because it's not so bad in two dimensions and it generalizes to higher dimensions as well. >> Okay. >> So it turns out that we can use calculus to do this, I am not going to step through the two-variable example for reasons that I am embarrassed to say. But I am going to show you a different example. So imagine that what we're trying to do is that we've got a bunch of data points, and we're trying to find the best constant function, right? So the best function that has the form, the value of the function for any given X is always the same constant, C. So if our data looks like this, we got a bunch of X's and a bunch of Y's, then what we're going to do, we're going to say for any given value of C, any given constant, we can have an error. What's the error going to be? The error is going to be the sum over all of the data points. Speaker 1: The square difference between that constant we chose and what the actual y_i value is. So that's why. >> Michael. >> These differences here. Yes, Charles. >> Can I ask you a question? >> Sure. >> Why are we doing sum of squares? >> There is many different error functions and sometimes called a, a relative concept called the loss function. There is lots of difference once that could work, you can do the absolute error, you can do the squared error, you can do various kinds of squashed errors where you know. The errors count different depending on how, how much deviation there is. It turns out that this one is particularly well behaved because of this reason that I'm explaining now that that because this error function is smooth as a function of the constant C, we can use calculus to actually find the minimum error value. But there's lots of other things that could work and they actually do find utility in various different machine learning settings. >> Okay. >> So just now using the chain rule, if you want to find how do this error function output change as a function of input c. We can take the derivative of this sum you know, bring the two over. Times this, times the derivative of the inside, which is negative one in this case. And now this gives us a nice, smooth function saying what the error is as a function of c. And if we want to find the minimum, what do we want to do to this quantity? >> Set it equal to zero, because that's what I remember from Calculus. >> That's right. So in particular if the error you know, the error function is a nice smooth thing the derivative is negative and then zero and then positive. When it hits zero that's when the thing has

bottomed-out. Alright. So now we just need to solve this, this equation for c . So we have one equation and one unknown. Alright, so that gets us this. But, this quantity, it's just the constant added to itself n times. So it's n times c . We move that to the other side. We get n times c . N is the number of data points as you recall. Is the sum of the y_i 's. We divide two by n and what do we see? So what is it Charles? >> The best constant is the average of all your y 's. >> Great, it's the mean. The mean comes back. Right, so in the case of finding the best constant here, we just have to average the y , the y 's together and that catches thing that minimizes the squared air. So squared air is this really nice thing because it tends to bring things like mean back into the picture. It's really very convenient. And, it generalizes to higher, higher order of function tier, not higher functions, but more variables like, like lines. Sorry. Lines that have some, some non constant slope. By doing the same kind of process and things actually work really nicely.

Order of polynomial

$k=0$: constant

$k=1$: line

$k=2$: parabola

$$\underline{f(x) = c_0 + c_1 x + c_2 x^2 + \dots + c_k x^k}$$

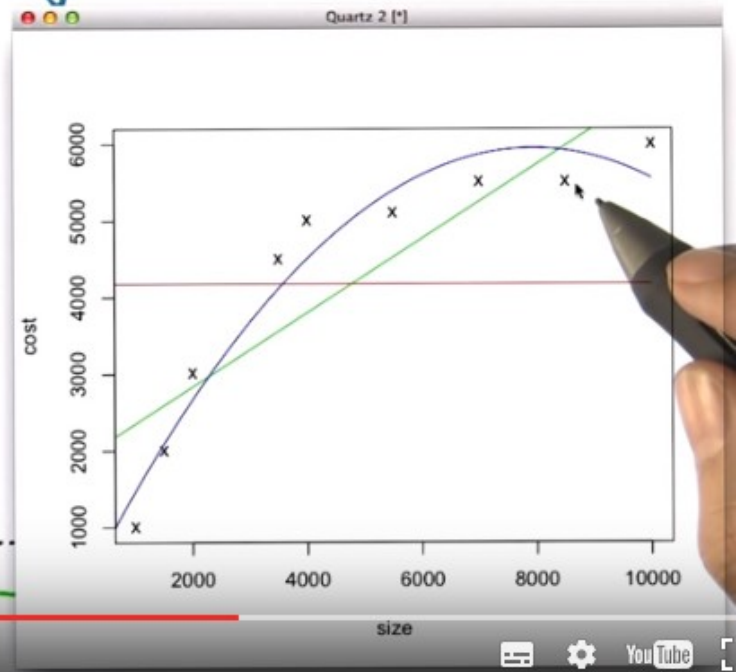
Order of polynomial

$k=0$: constant

$k=1$: line

$k=2$: parabola

$$f(x) = c_0 + c_1 x + c_2 x^2 + \dots$$



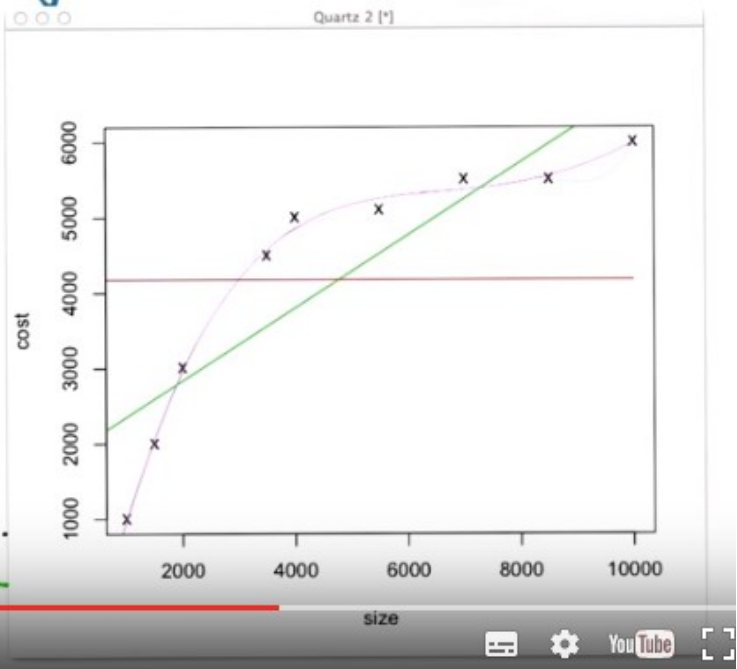
Order of polynomial

$k=0$: constant

$k=1$: line

$k=2$: parabola

$$f(x) = c_0 + c_1 x + c_2 x^2 + \dots$$



上圖中那個彎的曲線是 order 4

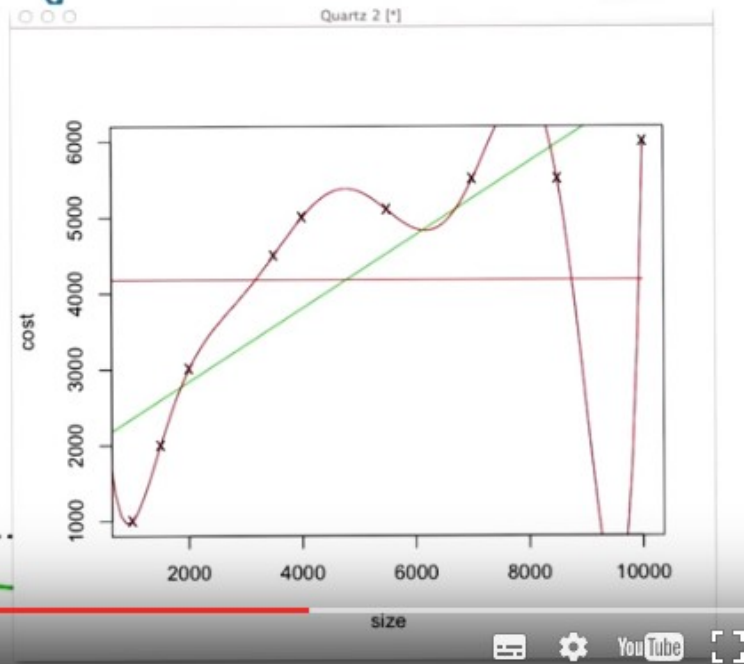
Order of polynomial

$k=0$: constant

$k=1$: line

$k=2$: parabola

$$f(x) = c_0 + c_1 x + c_2 x^2 + \dots$$



上圖中那個彎的曲線是 order 6

Order of polynomial

$k=0$: constant

$k=1$: line

$k=2$: parabola

$$f(x) = c_0 + c_1 x + c_2 x^2 + \dots$$



7. Alright, so now let's, let's get back to our data set that we were looking at before. So again, the ideas

that we're, we're going to try to find a way of predicting the value for various points along the way on this curve. And one thing we could do is find the best line. But we also talked now about finding the best constant. Turns out these all belong to a family of functions that we could fit. Which are functions of this form. Alright. We've got x is our input and what we're going to do is we're going to take some constant and add that to some scaled version of x times some scaled version of x squared plus some scaled version of x cubed, all the way up to some order k . And we've talked about k equals zero, the constant function. And k equals one, the line. But there's also k equals two, parabola. Would it probably be a good choice at this particular case? >> Yes. >> It does seem like it's got, sort of, curvy downy nature, right? >> Mm hm. >> It's going up and it's kind of flattening out and maybe we could imagine that it starts coming down again? At least, over the course of these points, it doesn't come down again but at least it sort of flattened out. So let's take a look at that. Let's take a look at the. The best parabola to fit this. Alright, so, so here we go. We've got the, the best line now, the best constant function which is just the average. We have the best line with some slope to it. That's the green one. We have now the best parabola and look at it, it does, it does a nice job, right? Kind a gets, gets tucked in with all those other points. so, so what do you think? Is this the best way of, of capturing this. This particular set of points? >> Well, if the only thing we care about is minimizing the sum of squared error, my guess is that the parabola has less squared error. >> Yeah. It ha, there's more degrees of freedom so at the worst we could have just fit the parabola as a line. Right. We can always just set any of these coefficients to, to zero. So if the best fit to this really was a line. The best fit to this data point was a line, then the parabola that we see here wouldn't have any curve to it. So, yeah. Our arrows going down. Hm, As we have gone from order zero to order one to order two. So can you think of any other way getting there in order to getting down even more. >> How about order 14 million. >> Interesting, while in this particular case, given the amount of data that we have, we can't go past the number of data points, yeh after that. They're really unconstrained. >> Okay. Then how about order nine? >> Order nine is a good idea. But just to give you an idea here, we're going to step up a little more. This is order four and look at, look at how lovely it can actually capture the flow here. That's, very faded. Order six and in fact the best we can do here is of the, of the, sorry. The most, the highest order that, that works is order eight. And, son of a gun, look what it did. It hit every single point dead on in the center. Boom. Boom. Boom. It used all the degrees of freedom it had to reduce the error to essentially zero. >> Excellent. >> So [LAUGH], one could argue that this is a really good idea. Though, (order 6 時) if you look at what happens around 9000, there's some craziness. Do you see that? >> I do. >> At the Yeah, the To try to get this particular parabola to hit that particular point, it sent the curve soaring down with an up again. >> We also did that between 6500 and 8500, it sent [CROSSTALK]. >> Yes good point >> [UNKNOWN] [CROSSTALK] >> Right, right, that's right went off the top of the plot. So Yeah, that's kind of [INAUDIBLE]. But let's just, just to show that we really are, as we have more degrees of freedom we're fitting the error better. Let me show you what it looks like, the amount of error for the best fit for each of these orders of k . Alright and so, so what you see when we actually plot the, the squared error, this function that we're trying to minimize. As we go from order zero to order one, order two, order three, order four, order five, all the way to eight. By eight, there is no error left because it nailed every single point. So you know its kind of a good, but it doesn't feel quite right, like the curves that we're looking there looked a little bit crazy.

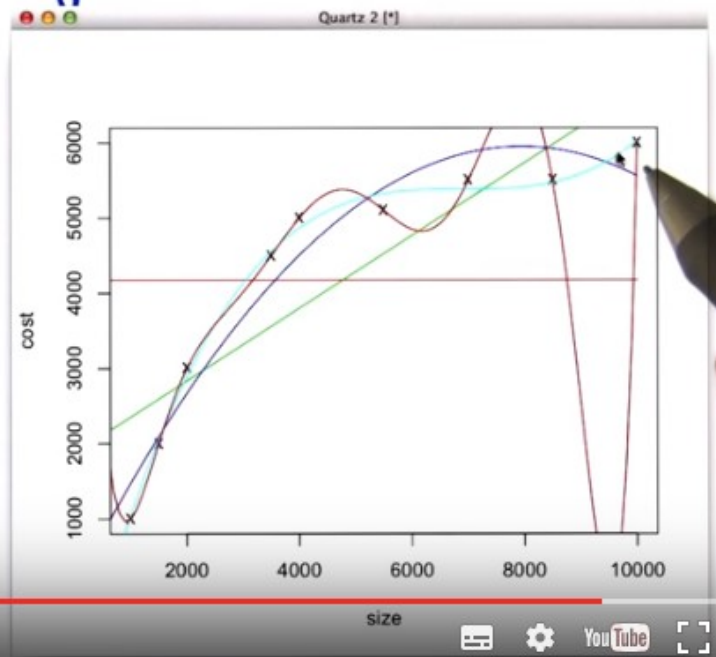
8. All right, so let's, let's do a quiz. Give you a chance to kind of think about what where are these trade-offs are actually going to be. So we're going to pick the degree for the housing data, and your choices are going to be the degree zero, one, two, three, or eight. So a constant, that's the first choice. Or a line that has some slope that, you know, sort of increases with the data, that's your second choice. Or it could be we use a degree two parabola. So sort of goes up and then levels off. Or you can it might be a little hard to see but here's a cubic that that goes up flattens out a little bit and then rises up again at the end. Or we could go with the full monty, the octic. You can see that might not be spelled correctly.

That actually has enough degrees of freedom that it can hit each of these points perfectly. Like that. >> The authority line. >> [LAUGH] Good, you got that in.

Pick the degree

Which degree should we choose for this data?

- ☐ $K=0$, constant
- ☐ $K=1$, line
- ☐ $K=2$, parabola
- ☒ $K=3$, cubic
- ☐ $K=8$, octic



9. So Charles, how would we go about trying to figure this out? >> How would we go about trying to figure this out? >> Yeah, what do you think? Which one would you choose and how would you choose? >> so, well that's a good question. Well just given what you, what you've given me, I'm going to ask. I think smartly guess, that probably k equals 3, is the right one, and >> K equals 3 >> And I'll tell you why. It's because zero, one and two seem to make quite a few errors. >> Mh-hm >> Three does a pretty good job but doesn't, doesn't over commit to the data. >> Hm. >> And that's the problem with eight, is that eight says, you know, the training data that I have is exactly right and I should bend. And move heaven and earth in order to, to match the data. And that's probably the wrong thing, certainly if there's any noise or, or anything else going on in the data. >> Right. So it sort of seems like it's overkill, especially that it's doing these crazy things between the points. Whereas the cubic one, even though it clings pretty close to the points, it stays between the points, kind of between the points. >> Yeah. >> Which seems like a really smart thing. So yeah so, so that turns out to be the right answer but lets actually lets actually evaluate that more concretely.

Polynomial Regression

X	y
x_1	y_1
x_2	y_2
\vdots	\vdots
x_n	y_n

$$\underline{C_0 + C_1 X + C_2 X^2 + C_3 X^3} \approx \boxed{y}$$

$$\begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 \\ 1 & x_2 & x_2^2 & x_2^3 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 & x_n^3 \end{bmatrix} \begin{bmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{bmatrix} \approx \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}$$

$X \quad w \approx Y$

$$Xw \approx Y$$

$$X^T X w \approx X^T Y$$

$$\cancel{(X^T X)^{-1}} \cancel{X^T X} w \approx \cancel{(X^T X)^{-1}} X^T Y$$

$$w = (X^T X)^{-1} X^T Y$$

10. Alright. So we talked through how it works when you've got you're trying to fit your data to a constant function, to a zero order polynomial. But let's, let's at least talk through how you do this in the more general case. This is, this is what I've been doing to, to fit various curves to the data at least implicitly. So, what we're really trying to do is we've got a set of data, x and y . Set n , n examples of x 's and their corresponding y 's. And what we're trying to find is these coefficients, C_0 , C_1 , C_2 , C_3 . Let's say if we're trying to do cubic regression where C_0 gets added to C_1 times x , which gets added to C_2 times x squared. Which gets added to C_3 times X cubed and we're trying to get that to look a lot like y . Now we're not going to get to exactly equal y but let's pretend for a moment that we could. We have a

bunch of these examples and we want it to work for all of them. So we can arrange all of the, all these constraints, all these equations into matrix form. If you're familiar with linear algebra. So the way that we can write this is here are the, here are the coefficients that we're looking for, the C's, and here are what we're going to multiply them by. We're going to take the X one and look at the zeroth power, the second power, the third power. And that equation I'll use my hands cause that's I always, I always need to use my hands when I do matrix multiplication. So you're going to across here and down there to multiply these and add. And that needs to correspond to y_1 . And same thing this now the second row. Multiplied by these coefficients. Need to give us our y_2 and so forth. Alright. So if we arrange all these x values into a matrix, and we'll call it, you know, x . And then we have these other guys. And we'll call this w , like the coefficients. Obviously w stands for coefficient. And we want that to sort of equal This vector of y 's. And we basically just need to solve this equation for the w 's. Now, we can't exactly solve it (因為未知數比方程少) because it's not going to exactly equal, but we can solve it in a least squares sense. So let me just step through the steps for doing that. Alright, so let's, so here's how we're going to solve for w . So what we're going to do is premultiply by the transpose of x . Both sides. I mean really what we wanted to do at first is if we are solving for Y , we need to multiply by the inverse of X , but this isn't really going to be necessarily well behaved. But if we pre multiplied by the X transpose then this thing ($X^T X$) is going to have a nice inverse (另一個原因應該是 $X^T X$ 是一個方陣, 注意 X 不是方陣, 所以跟本沒有逆). So now we can pre multiply by that inverse. All right. Now, conveniently because this has a nice inverse, the inverses cancel each other. [NOISE] We get that the weights we're looking for can be derived by taking the x matrix times its own transpose, inverting that, multiplying by x transpose and then multiplying it by the y . And that gives us exactly the coefficients that we need To have done our polynomial regression. And it just, it just so happens that we have some nice properties in terms of these x transpose x . Not only is it invertible, but it does the right thing in terms of minimizing the least squares. It does it as a projection. Now, we're not going to go through the process by by which we argue that this is true. >> Does it have something to do with calculus? >> It most likely has something to do with calculus. And we'll get back to calculus later. But in this particular case we can, we're just using projections and linear algebra. And most importantly the, the whole process is just we take the, the data we arrange it into this matrix with whatever sort of powers that we care about. And then we just compute this quantity and we're good to go. >> Okay.

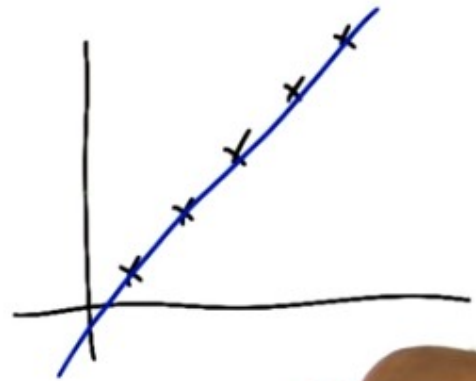
Errors

Training data has errors

Not modeling f , but $f + \epsilon$

where do errors come from?

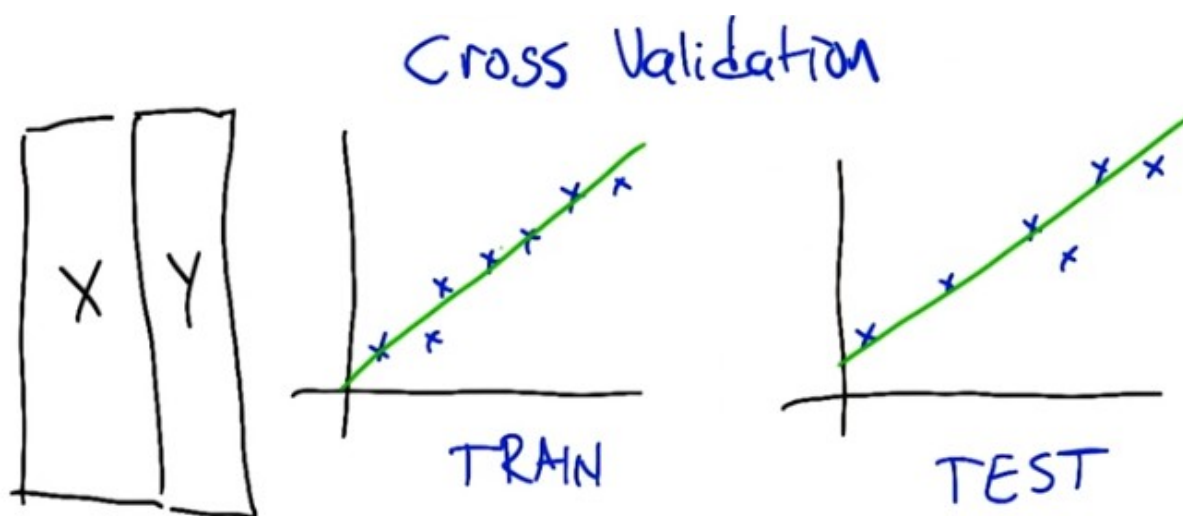
- ☒ Sensor error
- ☒ Maliciously - being given bad data
- ☒ transcription error
- ☒ unmodeled influences



11. Alright, now, part of the reason, we can't just solve these kinds of problems by solving, a system of linear equations and just being done with it, the reason we have to do these squares is because of the presence of errors. The training data that, that we are given, has errors in it. And it's not that we're actually modelling, a function, but ,the thing that we're seeing is the function plus some, you know, some error term on each piece of data. So, I think, it's reasonable to, to think about where did these errors come from? So, I don't know, what do you think ,Charles, why, why is it we're trying to fit data, that ,has error in it, can't we just, can't we just have no errors? [LAUGH] >> I would like to have no errors. Certainly ,my code, has no errors. [CROSSTALK] well, so let's see where might errors come from. So, they could come from, sensor error, right? Just ,somehow you're, you're getting inputs and you're getting outputs and that output's, being read by, some machine or by a camera or by something and you just, there's just error in the way that you read the data. Just an error in the sensors. >> Alright, can you think of other ways. I guess, I guess ,in this case you're imagining that the data came by actually measuring something, with the machine. So that, that makes, a lot of sense. What other ways, can we put together the data? >> I don't know I could think of a bunch. I mean the error, well, the errors could come, maliciously. There could be some, something out there, that is trying to give us bad data. >> Alright, that seems like a possibility, that, when the data set was collected, let's say that we're collecting, various, Oh, maybe if I. Oh, this happens, this happens a lot. So, so, if you're trying to collect data from other Computer Science departments and you're trying to put together, some kind of collection of, you know, how much do you spend on your. Graduate students ,say,uh, sometimes ,these departments will actually misrepresent the data and give you give you, things that are wrong. Because, they don't want to tell you the truth, because they're afraid of what you are going to do. >> Yeah, I've noticed that everyone does that except, for Georgia Tech and Brown University. >> Yeah, there are highly honest and reputable universities in my experience. >> Yeah, that's what I feel. >> well, another time that you can get data, is if somebody, is, copying stuff down. So, what about sort of the

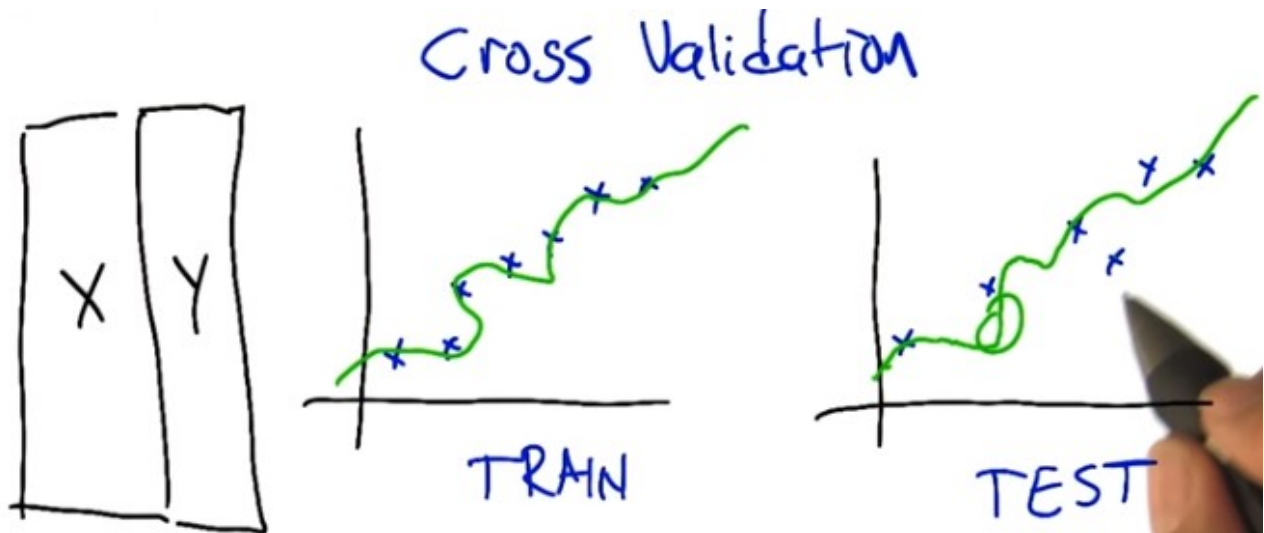
idea of a transcription error. >> Uh-huh. >> So we're just, you know, we've copied everything, but, you know, there's, there's just some of the, some of the lines that got filled in just got mistyped. >> Yeah, and you think ,that's different from sensor error? >> Well, it's, it's maybe a slightly different kind of sensor error, right? So ,sensor errors were actually saying there's something physical, that's being measured and there's just noise in that. Transcription error, is similar except it's a person. [LAUGH] >> Mm. >> Right? The, the there's a little blips in the person's head and they can do, it can be a very different kind of error. You can get, like transpositions of digits, maybe instead of ,um, just you know, noise. >> Okay, how bout, how bout one more? How about ,uh, there's really, just noise, in the process. So how about that, that we took in input X, but there's something else going on in the world, that we weren't measuring, and so the output ,might depend on other things besides, simply ,the input that we're looking at. So what would be an example of that? >> So an **un-modeled influence**, might be, well, if we're. >> [CROSSTALK] Let's look at the housing data. >> That's what I, that's what I was thinking exactly. So, **in the housing data, we were just trying to relate, the size of the houses, to the price, but, there's a lot of other things like** change of the houses to the price and >> **Location**, location. >> Location and location, right those are three really good reasons, that are not in the particular regression, that we did, that could ,actually influence the prices. So right, that and, you know, the quality of the house and who, who built it, and, you know, the colors, the colors. >> Even, even, even time of day, or what the interest rates were that morning, versus the, what people thought they might be the next day. Who knows? >> Right and so all these different things are being considered ,in that particular regression, so we're just kind of imagining ,that it's noise, that it's just having a, a ,uh, bumpy influence on the whole process. >> Sure. >> All right. **So, so what I'd like you to do is select ,the ones that you think actually are important, the ones that, that, that could actually come up, when you're using machine learning and regression to solve your problems.**

12. All right, and if you know, if you were paying attention as we were going through this, these are all very common, and realistic things. So, you know these are all true, **these are all sources of error. And this is why we really need to be careful when we fit our data. We don't want to fit the error itself, we want to just fit the underlying signal.** So let's talk about how we might be able to figure that out. How can we, **how can we get a handle on what the underlying function really is apart from the errors and the noise** that are, that are in it.

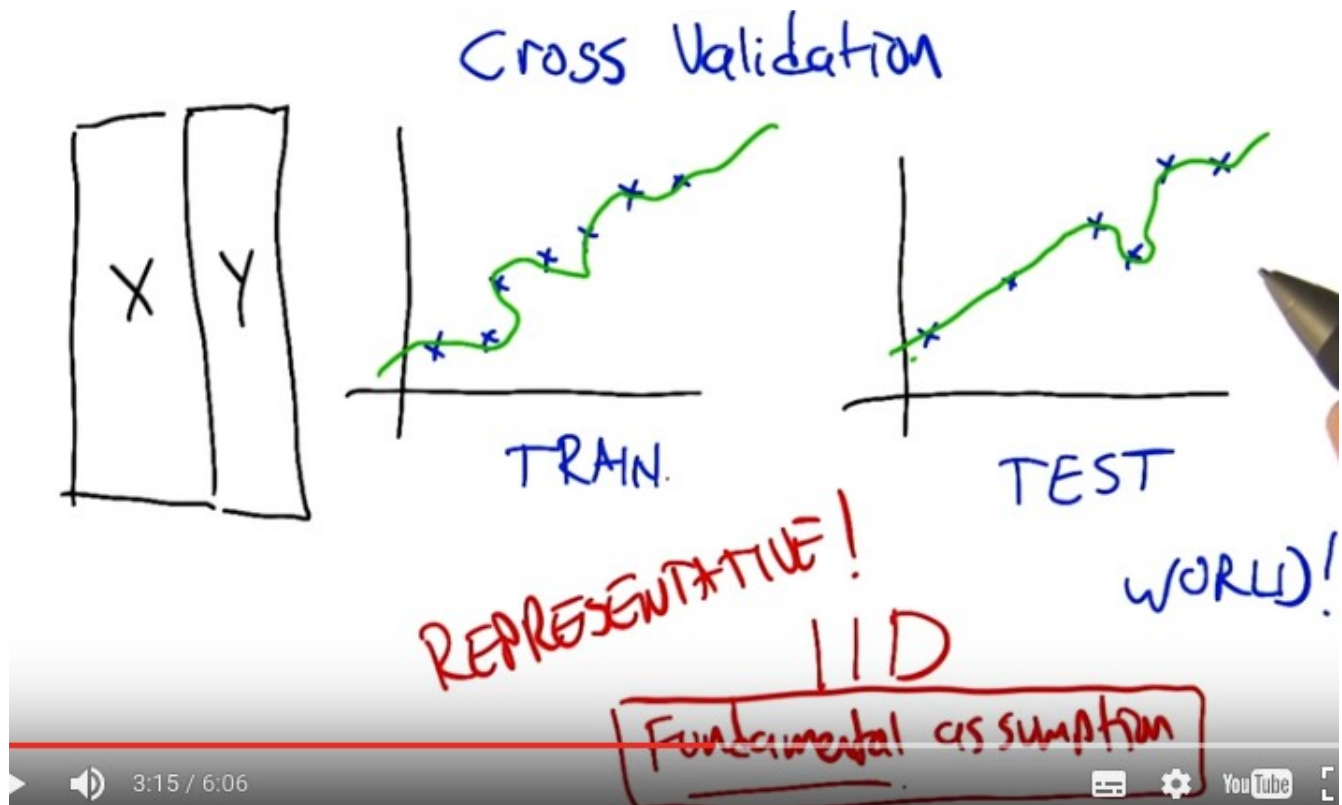


13. Alright, so let me try to get to this concept of cross validation. So, imagine that we've got our data, this is our training set. We can, again, picture geometrically in the case of regression. And, ultimately

what we're trying to do is find a way of predicting values and then testing them. So, **what we imagine is we do some kind of regression (in the training set) and we might want to fit this too a line.** And, you know, the line is good, it kind of captures what's going on and **if we apply this to the testing set, maybe it's going to do a pretty good job.**



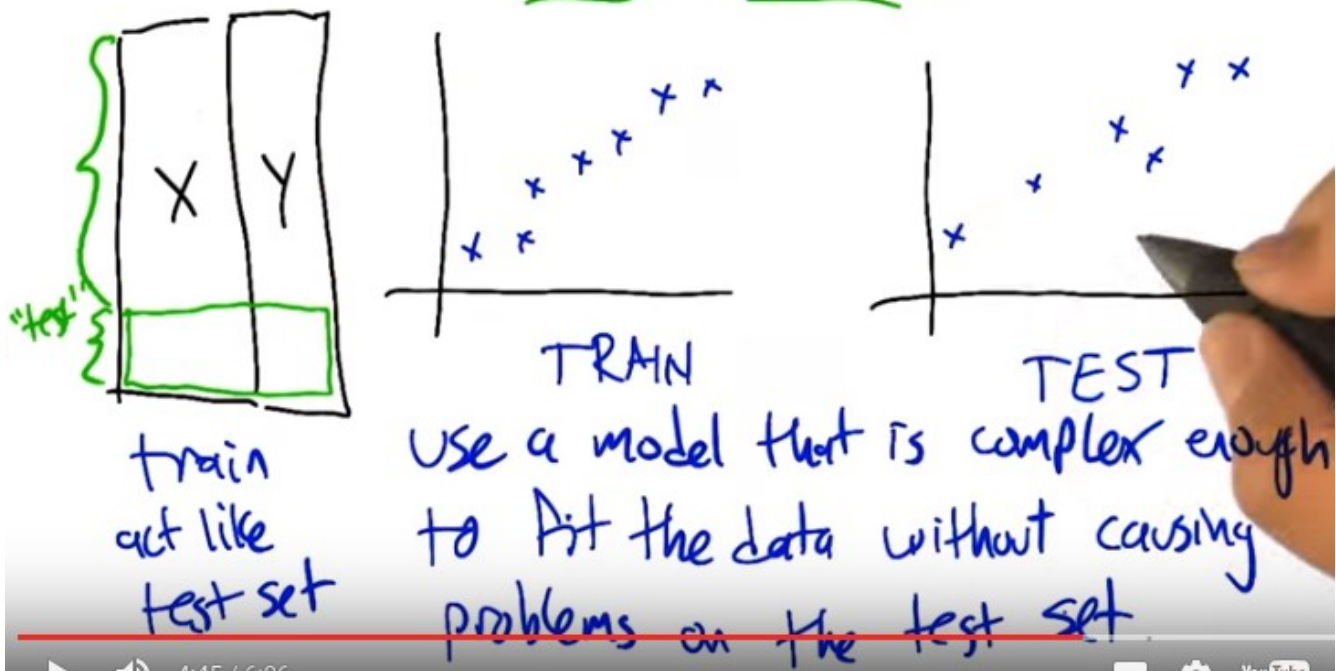
But, if we are, you know, feeling kind of obsessive compulsive about it we might say well in this particular case we didn't actually track all the ups and downs of the data. So what can we do in terms of if we, if we fit it with the line and the errors not so great. What else could we switch to Charles? >> We could just use the test. No, sorry. What, what I mean is if we fit, we fit this to a line and we're sort of not happy with the fact that the line isn't fitting all of the points exactly. We might want to use ,uh, maybe a higher order polynomial. >> Oh, I'm sorry, totally misunderstood you. >> **To fit this better.** So if we, we can fit this with a higher order polynomial and maybe it'll hit, all these points much better (in the training set) . You know, so we have this kind of, kind of other shape, and now it's doing this, **it's making weird predictions in certain places (in the testing set).** So, really what we'd like to do is, and what was your suggestion? **If we trained (did regression) on the test set, we would do much better on the test set, wouldn't we?** >> Yes. >> **But that, that, that's definitely cheating.**



IID: independent and identically distributed

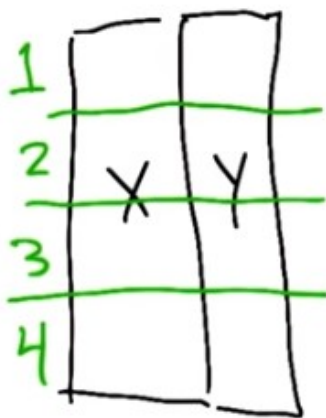
>> **Why is cheating?** >> Is there some, why is it cheating? Well, if we exactly fit the error, the, the test set. That's not a function at all, is it? [LAUGH] **If we exactly fit the, the test set, then again that's not going to generalize to how we use it in the real world.** >> **So the goal is always to generalize. The test set is just a stand-in for what we don't know we're going to see in the future.** >> Yes, very well said. Thank you. >> Actually that suggests something very important, right, it suggest that ,um, nothing we do, on our training set or even if we cheat and use the test set .Actually makes sense unless we believe that somehow the training set and the test set represent the future. >> Yes, that's a very good point, that we are assuming that this data is representative of how the system is ultimately going to be used. In fact, there's an abbreviation that statisticians like to use. That the data, **we really count on the data being independent and identically distributed,** >> Mm-hm. >> **which is to say that all the data that we have collected, it's all really coming from the same source, so there is no, no sort of weirdness that the training set looks different from testing set looks different from the world** but they are all drawn from the same distribution. >> So would you call that a fundamental assumption of supervised learning? >> I don't know that I'd call it a fundamental of supervised learning per se, but it's a fundamental assumption in a lot of the algorithms that we run, that's for sure. >> Fair enough. >> There's definitely people who have looked at, well what happens in real data if these assumptions are violated? Are there algorithms that we can apply that still do reasonable things? But the stuff that we're talking about? Yes, this is absolutely. A fundamental assumption. Alright, but here's, here's where I'm trying to get with this stuff.

Cross Validation

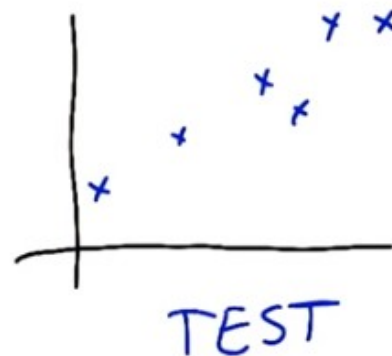
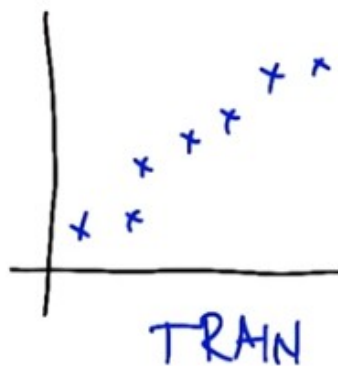


So what we really would like to do, is that we'd like to use a model that's complex enough to actually model the structure that's in the data that we're training on, but not so complex that it's, it's matching that so directly that it doesn't really work well on the test set. But unfortunately we don't really have the test set to play with because that again, is going to, it's too much teaching to the test. We need to actually learn the true structure that is going to need to be generalized. So, so how do we find out. [How can we, how can we pick a model that is complex enough to model the data while making sure that it hasn't started to kind of diverge in terms of how it's going to be applied to the test set.](#) If we don't have access to the test set, is there something that we can use in the training set that we could have it kind of act like a test set? >> Well, we could take some of the training data and pretend it's a test set and that wouldn't be cheating because it's not really the test set. >> Excellent. Indeed, right, so there's nothing magic about the training set all needing to be used to fit the coefficient. It could be that we hold out some of it, as a kind of make pretend test set, a test test set, a trial test set, a what we're going to say cross validation set. And it's going to be a stand in for the actual test data. That we can actually, make use of that doesn't involve actually using the test data directly which is ultimately going to be cheating. So, this cross validation set is going to be really helpful in figuring out what to do.

Cross Validation



Folds



To pick a model....

Train on

1	2	3
2	3	4
3	4	1
4	1	2

pick lowest error.

check

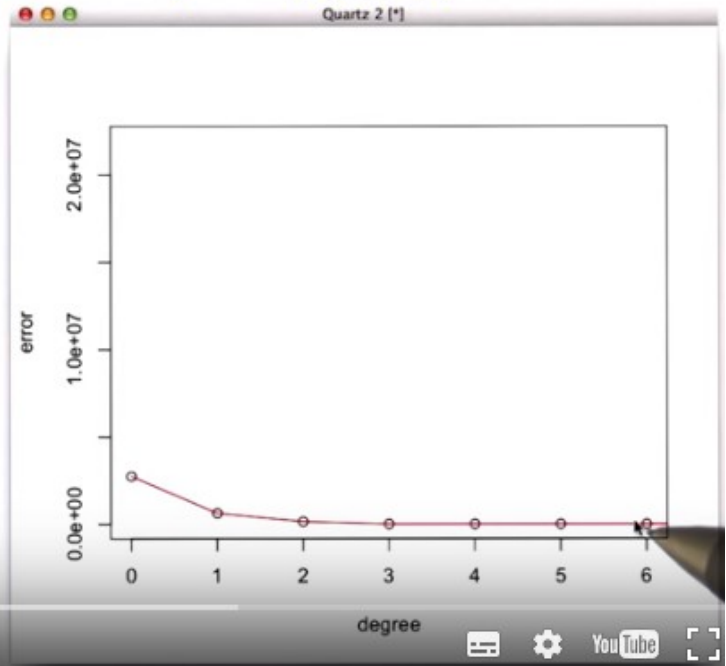
average error together.

So. Alright, so here's how we're going to do this, this concept of cross validation. We're going to take our training data, and we're going to split it into what are called folds. I'm not actually sure why they're called folds. I don't know if that's a sheep reference. >> Why would it be a sheep reference? >> I think there's a sheep-related concept that is called a fold. Like, You know, we're going to bring you back into the fold. >> Oh. >> It's like the, it's like the group of sheep. >> You are just trunk full of knowledge. >> Alright so what we're going to do is train on the first three folds, and use the fourth one to, to see how we did. Train on the second there and fourth fold and check on the first one. And we're going to we're going to try all these different combinations leaving out each fold as a kind of a, a fake test set. And then average these errors. The ,uh, the, the goodness of fit. Average them all together, to see how well we've done. And, the model class, so like the degree of the polynomial in this case that does the best job, the lowest error, is the one that we're going to go with. Alright, so if this is a little bit abstract still let me, let me ground this back out in the housing example.

Housing Example Revisited

Degree of polynomial
per-example error

training error
→ improves



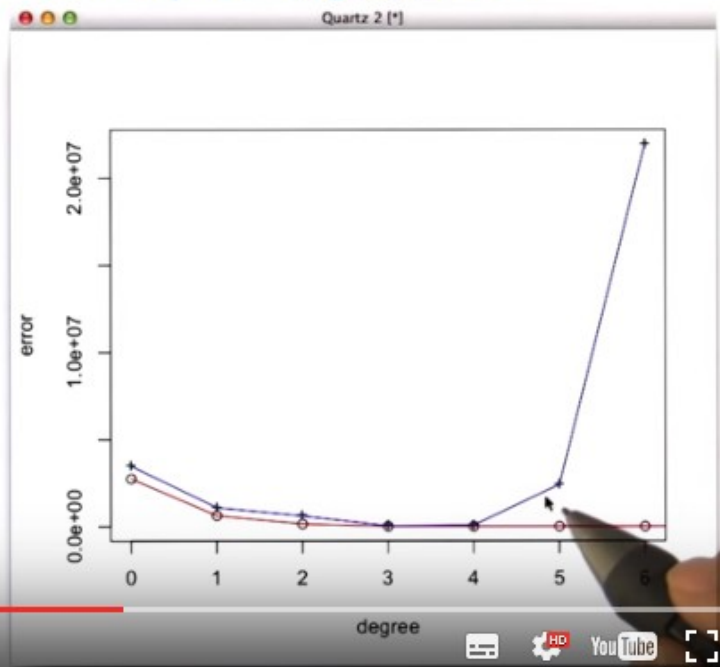
14. Alright so here's how we're going to look at this. So as you may recall, in this housing example. If we look at different degrees of polynomials and how well they fit the data. Let's look at the training error. The per example training error. So how far off is it for each of the data points? And as we increase the degree of the polynomial from constant to linear to quadratic and all the way up to, when this case order six, the error's always falling. As you go up, you have more ability to fit the data, closer and closer and closer, right? because, each of these models is, is nested inside the other. We can always go back. If the zero fits best and I give you six degrees of freedom, you can still fit the zero. So, that's what happens with the training error.

Housing Example Revisited

Degree of polynomial
per-example error

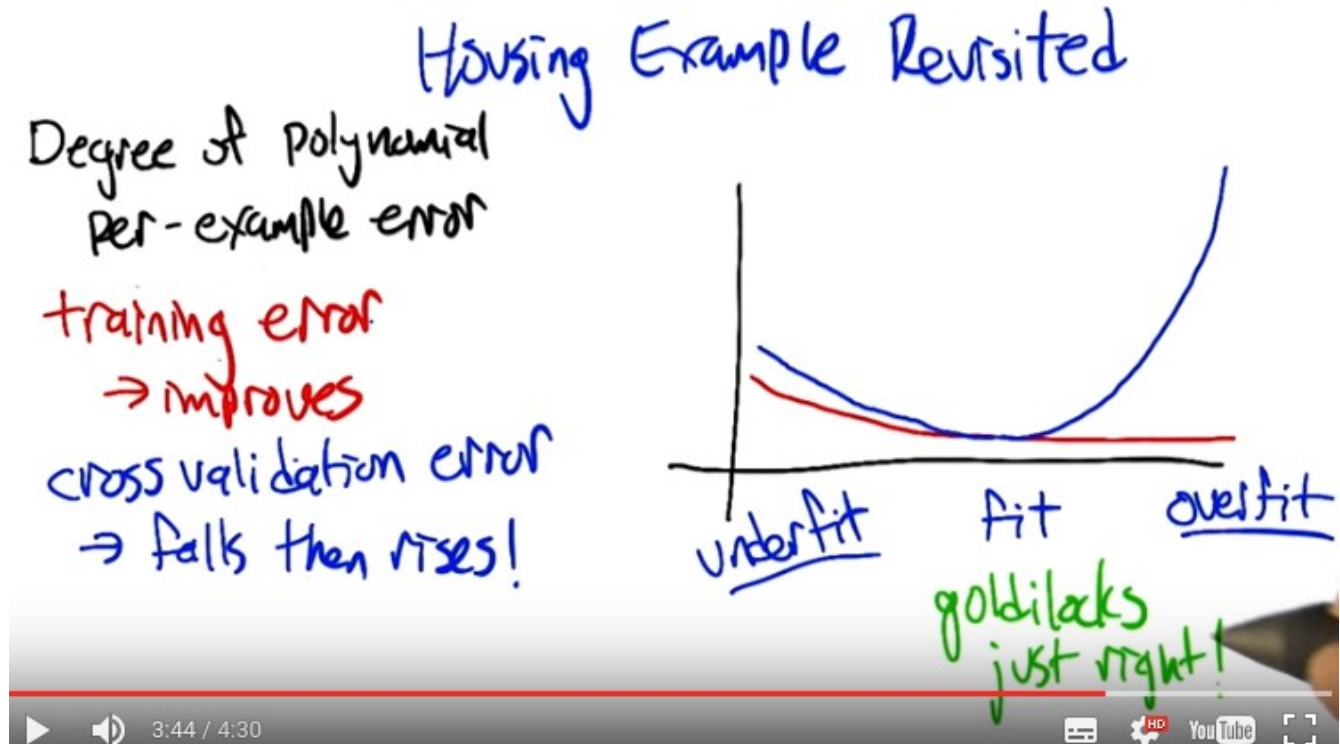
training error
→ improves

cross validation error
→ falls then rises!



But now let's use this idea of cross validation to say what if we split the data up into chunks and have each chunk being predicted by the, the rest of the data? Train on the rest of the data, predict on the chunk. Repeat that for all the different chunks and average together. So, so I actually did that. And this is what I got with the cross validation error. So there's a I don't know there's a couple of interesting things to note about this plot. So that we see, we have this red plot that is constantly falling and the blue plot which is the cross validation error starts out (即 degree 小的时候) a little bit higher than the, the red plot that's got higher error. So, why do you think that is Charles? >> Well that makes sense right? because we're actually training to minimize error. We're actually trying to minimize error on the training set. So the parts we aren't looking at, you're more likely to have some error with. That makes sense if you'd have a little bit more error on the data you haven't seen. >> Right, so, good. So, so, in the, (原因:) on the, this red curve. We're actually predicting predicting all the different data points using all of those same data points. So it is using all the data to predict that data. This blue point, which is really only a little bit higher in this case, is using, in this particular case I used all but one of the examples to predict the remaining example. But it doesn't have that example when it's, when it's doing its fitting. So it's really predicting on a new example that it hasn't seen. And so of course you'd expect it to be a little bit worse. In this particular case, the averages are all pretty much the same so there's not a big difference. But now, let's, let's look at what happens as we start to increase the degree, we've got the ability to fit this data better and better and better, and, in fact, down at you know say, three and four, they're actually pretty close in terms of their ability to, to, to fit these examples. And then what's great, what's really interesting is what happens is now we start to give it more, the ability to fit the data closer and closer. And by the time we get up to, to order six polynomial, even though the error on the training set is really low, the error on this, on this cross validation error, the error that you, that you're measuring by predicting the examples that you haven't seen, is really high. And this is beautiful this, this inverted u, is, is exactly what you tend to see in these kinds of cases. That the error decreases as

you have more power and then it starts to increase as you use too much [LAUGH] of that power. Does that make sense to you?



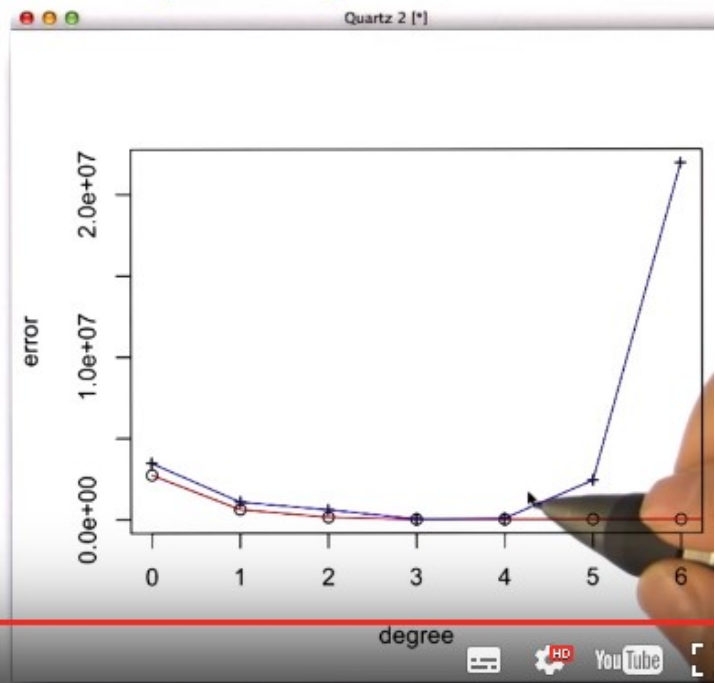
>> It does make sense, so. The, the problem is that as we give it more and more power we're able to fit the data. But as it gets more and more and more power it tends to over fit the training data at the expense of future generalization. >> Right. So that's exactly how we, we referred to this is this sort of idea that if you don't give yourself enough degrees of freedom, you don't give yourself a model class that's powerful enough you will underfit the data. You won't be able to model what's actually going on and there'll be a lot of error. But if you give yourself too much you can overfit the data. You can actually start to model the error and it generalizes very poorly to unseen examples (overfitting 就是因為 model 了 error, 所以不能很好地 generalize to unseen examples). And somewhere in between is kind of the goldilocks zone. Where we're not underfitting, and we're not overfitting. We're fitting just right. And that's the point that we really want to find. We want to find the model that fits the data without overfitting, and not underfitting.

Housing Example Revisited

Degree of polynomial
per-example error

training error
→ improves

cross validation error
→ falls then rises!



>> So what was the answer on the, housing exam? >> Well, so, it seems pretty clear in this, in this plot that it's somewhere, it's either three or four. It turns out, if you look at the actual numbers, three and four are really close. But three is a little bit lower. So three is actually the thing that fits it the best. And, in fact, if you look at what four does. It fits the data by more or less zeroing out the, the quartic term, right? It doesn't really use the, this power. >> Oh, but that's interesting. So that means it, it barely uses the, the, the extra degree of freedom you give it. But even using it a little bit, it still does worse than generalization. >> Just a tiny bit worse. >> Huh. >> Yup exactly so. >> That's actually kind of cool.

Other Input Spaces

→ scalar input, continuous x

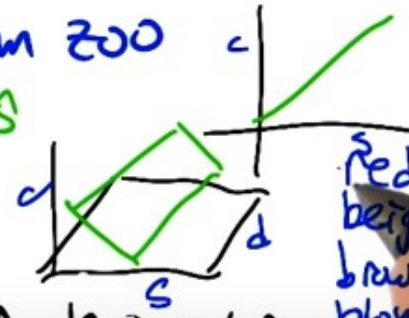
→ vector input, continuous. \vec{x}

include more input features

size, distance from zoo

hyper planes

predict credit score.
job? age? assets?



→ discrete, vector or scalar. hair color

$\{0, 1\}$

red, grey, brown

→ RGB, enumerate

- red 1
- berge 2
- brown 3
- blonde 4
- black 5



5:14 / 5:23



15. Alright. Up to this point I've been talking about regression in the context of a scalar input and continuous output. Sorry. Scalar input and continuous input. So basically this x variable. But the truth of the matter is we could actually have vector inputs as well. So what would might, what might be an example of where we might want to use a vector input? >> A couple of things. One if you look at the housing example, like we said earlier, there are a bunch of features that we weren't keeping track off. So we could have added some of those. >> Great yeah, we could include more input features and therefore combine more things to get it. But how would we do that? So let's say for example, that we have. Two input variables that we think might be relevant for figuring out housing costs. The size, which we've been looking at already, But also let's say the distance to the nearest zoo. We, we think that that's a really important thing. People like to live close to the zoo and so. >> But probably not too close to the zoo. >> [LAUGH] Possibly not too close to the zoo. But let's let's imagine that it's like size, something that actually Or actually, let, let's do it the other way, let's sort of imagine that, that, that the further away from the zoo, you are, the better it is. Just like the bigger the size is, the better it is. >> Mm-hm. >> So how do we combine these two variables into one in the context of the kinds of function classes that we've been talking about? >> Well, if you think about lines, we can just generalize the planes and hyper planes. Right so, in the case of, of a 1 dimensional input. That 1, 1 dimensional input gets mapped to the cost. But in the case of 2 dimensional inputs, like size and distance to the zoo. We have something that's more like a plane, combining these two things together in, in the linear fashion to actually predict what the. Cost is going to be. So right, so these, this notion of linear functions generalizes, this notion of polynomial function function generalizes too very, very nicely (沒講如何

generalize). All right, there is another kind of input that's important too, that, let's think about a slightly different example to help drive the idea home. So let's imagine we are trying to predict credit score, what are some things that we might want to use as features to do that. >> Do you have a job? >> I do, actually. >> [LAUGH] yes. >> Oh, I am sorry, I am sorry, I misunderstood. So you are asking, you are saying one [UNKNOWN] that could be important for predicting someone's credit score is just to know do they currently have a job. Right another thing might be well you, you can ask instead how much money they actually, how, how much, how many assets they have. How much money do they have? Credit cards. >> Great. So, so, right. So things like, what is the value of the assets that, that they own, right? So this is a continuous quantity like we've been talking about. But something like do you have a job, yes or no, is a discrete quantity. And one of the nice things about these kinds of regression approaches that we've talking about, like polynomial regression, is that we can actually feed in these discrete variables as well (沒講如何弄). Certainly if they're, if they're Boolean variables like, do you have a job or not? It, you can just think of that as being a kind of number that's just zero or one. No, I don't have a job. Yes, I have a job. What if it's something like, you know, how many houses do you own? >> Hmm. >> That's pretty easy because that's, you could just treat that as a As a quantity, a scalar type quantity. What about >> Are you. >> Type of job. >> Type of job, I like that. How about hair color? >> So, yeah, how would we do that? If we, if we're trying to feed it in to some kind of regression type algorithm, it needs to be a number or a vector of numbers, and they can be discrete. So right. So how do we encode this as some kind of a numerical value? >> Well, we could do something ridiculous like actually write down the RGB value which would make it kind of continuous. >> Interesting. >> That seems insane, but you could do that. Or you could just enumerate them and just assign them values one through six in this case. Right, 1, 2, 6 or they could be vectors like, is it red, yes or no? Is it beige, yes or no? Is it brown, yes or no? Have it be a vector and actually for different kinds of discrete quantities like this it can make it different, right? So in particular if we just gave the numbers. Then it's kind of signalling to the algorithm that blonde is halfway between brown and black, which doesn't really make sense. We could reorder these. Actually the RGB idea doesn't seem so bad to me. >> [UNKNOWN] of course, you have an interesting question of what's the real RGB value. It implies that somehow interpreting between them Make sense. >> That's right, that's right. >> It also implies an order right. It implies that the scalar order of RGB is somehow mean something that it's no different from saying red is one and beige is two. So, if we multiply it, for example, by a positive coefficient then the more RGB you have The better or the worse, right? >> Hmm. Interesting. Though, in fact what I had in mind here is for RGB, it's three different hair colors. >> I thought the g stood for green. >> There's, people don't have green hair, they have gray hair. >> But I thought the g in RGB stood for green. Yeah it does usually but I'm making a hair joke. [LAUGH] >> Oh oh. I am sorry. I am glad you explained that. You know Michael. >> No problem sir. >> I really, I really like the [UNKNOWN] factor idea. >> Yeah so I think. I think. I imaging that we are going to return to this issues when we start actually encoding problems as mission learning problems. >> I think your right. >> But I think that's I think that's said about regression for the time being. >> I agree.

Conclusion

What did we learn?

- historical facts
- model selection and under/over fitting
- cross validation
- linear, polynomial regression
- best constant in terms of squared error: mean.
- representation for regression

16. So, Charles, what did we learn about regression? >> well, we learned a bunch of interesting historical facts about where the word came from, which I thought was interesting anyway. >> Oh good. >> We learned about model selection and overfitting. And underfitting. And fitting in general, cross validation. >> Talked about, how to do linear and polynomial regression. >> Yeah. >> That the best constant, that the best constant in terms of squared error is the mean. >> Mh-hm >> These little cocktails for that. >> Well we also did the same thing for well we talked about the process for how you do it in general for any polynomial function. >> I think that's everything. >> Well one more thing, and we talked a little bit about representation, and how to make that work in regression. >> Great, we talked about input representations and what some of the issues are. There. >> Yeah. >> Great, I think that's, I think that's a good amount. >> I think so, too.