Chapter 4.1 - Curve Fitting

18 MIN · YOUTUBE · HTTPS://WWW.YOUTUBE.COM/WATCH?V=GK60CFJ5G6S
https://www.youtube.com/watch?v=gK60CFJ5g6s

SUMMARY

Curve fitting is a crucial technique for analyzing data trends, allowing for extrapolation and interpolation. The chapter discusses various methods of curve fitting, focusing on least squares fitting, error metrics, and the implications of outliers on fitting accuracy. It also explores polynomial and nonlinear fitting, emphasizing the importance of choosing appropriate error metrics based on data characteristics.

- *Curve fitting simplifies the representation of data points for better interpretability and analysis.*
- *Least squares fitting minimizes the squared difference between data points and the chosen curve.*
- *Different error metrics (maximum error, average error, root mean square error) can significantly affect the fitting outcome, especially in the presence of outliers.*
- *Polynomial fitting can be solved directly using linear systems of equations, while nonlinear fitting requires more complex optimization techniques.*
- *Transformations, such as taking the logarithm of data, can linearize relationships and simplify fitting processes.*
- *The principles of curve fitting are foundational for understanding machine learning, as both involve optimizing parameters to fit models to data.*

we're going to move on to this chapter to talk about curve fitting curve fitting is a pretty important exercise especially when we're looking at data or Trends in data what we'd like is a simple representation of let's say a collection of points in which we can have interpretability potentially we can do extrapolation or interpolation and so what we really want is the ability to think about how we would fit data and or how we fit Trends in data and so that we

can can make some some progress and understanding so what we're going to talk about is a diversity of methods for doing it so the first thing we want to talk about is sort of the most standard way to think about the data analysis process which is leas square fitting and what Le Square fitting does is it basically picks a curve to fit the data and then you find the best fit to that curve and what the Le square is is is

essentially to minimize the Le square error of the curve that you pick to fit the data to the data so let's show this how this works the idea is I have a collection of points and so this data can be something that I collected from sensors or is data that I generated numerically but what the idea here is I have a collection of XY points and they I could plot these things and what I would like is a simple representation of

the trends in this data so what I might do is decide to for instance do a line fit so this is the simplest thing you could do well actually fitting it to a z fit constant Z here but I want to just let's say for instance do a line fit so this is the curve I choose so we'll show this on this data here I'm going to pick to fit a line now the question comes which is how should I pick the

coefficients a which is the slope of the line line or B which is the offset from the Y AIS when it goes through zero so choosing a and b should be done in a principled manner and sort of in a a principal manner where it minimizes some error so we're going to do an optimization problem here where we're going to try to find a and b by minimizing the error of a curve fit so the idea is the following I'm I

know I'm going to get an error so I'm going to take my curve this is the actual data point my curve is not going to go through the data it's going to be the best fit curve through all of this data so you know it doesn't go through the data and so there's an error every time I miss the point that I'm trying to go through I pick up an induc an error so what I would like to do is minimize

this error According to some Metric so this is where it becomes important because now we start getting into this idea of Le Square fitting which is the common way to think about curve fitting so in fact there's lots of different metrics we can use for curve fitting and so what I'm showing you here is a set of metrics that we can use to decide what error should I minimize so you can have things like the maximum error which is

this first one here which is you look to minimize Whoever has the largest eror push it down so that you across all the data the the maximum error is minimized or you can minimize the average error so notice in the average error I'm just trying to collect all the errors across all the data and then trying to minimize that sum or we can do what's often called the root mean square error which is you take the difference in the data

Square it sum it and then you square root that so this is the mean root mean square error which is a very common commonly used metric for error assessment right so it's it's not only used in sort of machine learning training often but it is if if if you don't specify an error most people will probably assume that you're going to use the root me Square and this is how it's defined here and by the way it's

related directly to the Pythagorean theorem we're looking at a length of a vector which is the square of all the sums square root so that's where it comes from it's from this L2 norm and is from something like Pythagorean theorem where we're going to try to minimize the length of a v the the error Vector as it were so these are different metrics by the way it's not clear that we have any moral highground to say this is the

right metric except for that it's just commonly practiced although it could be more appropriate to use one of the other errors here or even something else that would be relevant to the problem at hand so by the way if we use these errors let me talk about some of the different advantages or disadvantages of these so so here is a way to Picture This the Dots here are the actual data and what I've done here with the data is

I've have put in one outlier right and so just ask the question what happens if I put an outlier in the data what happens to these error metrics so what you're seeing here is three different fits for the three different Norms one is the E Infinity Norm which is the maximum error and if you fit the maximum error because you have this outlier point you get this curve here that moves away from all the other points typically

would' say this is not a very good fit on the other hand I've you know I've made the data with this bad data point so it really emphasizes why this data breaks down so but this is the way we would use these errors so you have to be aware that whatever error metric you pick you're going to have a certain kind of problem that may be associated with it so the E Infinity error or the maximum error moves off the line if you

have an outlier and then what you're looking at here the solid line is a least square error and the dotted line is the E1 error in fact the E1 error kind of does better than all the others partly because of this outlier in fact the L1 Norm often times promotes robustness because the L2 error not only takes this distance right it doesn't just look at the distance and trying to minimize it it takes this distance and

squares it and tries to minimize it so that really has a big impact when you have a large distance here whereas the L one Norm or the just the the distance between the error the the curve fit and the point it's not squared so it's not as pronounced of an error there so these are just things to consider typically when we do robust statistics or robust curve fitting you might want to promote something like the L1 Norm or in

other words the average error versus the root me square but often times if we have a distribution and you don't have outlier or you know things like that in your data a least square fit error is just just perfectly fine to use in fact all three airor curves would almost look identical if we just remove this one bad point so the point is that you should think a little bit about your error metrics depending upon the data

itself so always plot your data first when you do your curvefit to see do you have things like this happening in your data if you do use the appro appropriate metric all right so let's talk about what Le Square fitting does we're going to do a Le square fit and talk about how would we actually get these curves I showed you these curves right here right so if you look at these curves one of the questions is how did I actually get

these because what I actually found for the line is I found the coefficients A and B by minimizing those errors so how do we do it if we have a least square fit so what the least square fit is going to take that line up there and now what we're going to do is we're going to actually look at minimizing the square of the distance between our prediction F of XK and the actual data y of K so

we're going to take this difference we're going to try to essentially minimize the sum of all those squares okay so let's go ahead and put into our function f ofx k the actual function we want to use which is a line fit here which is a of XK plus b and I don't know what A and B are but what I do know is I'm trying to minimize this sum which is the error so whenever Trying to minimize

something your immediate thought should be to go to the derivative I want to take the derivative with respect to a and with respect to B and set it equal to zero to find the minimum error okay because that's what I'm trying to find is the A and B and so if I take the derivative of this error set it equal to zero that should be the minimum how do you know it's not a minimum or maximum

because there is no maximum error in other words I can move keep moving this further and further away from the data points and so you know that if you find something it's going to be a Minima not a Maxima so let's go ahead and go to it so I take that fit and all I do is exactly what we've talked about here so again derivatives play this key role throughout everything we do because it's basically telling us here now how to get

this best fit curve which is you take the error respect respect to a and respect to B set it equal to zero and these are very easy things to differentiate and here's what you get here for these formulas I can rearrange these formulas into a 2X two system of equations which is given by here now this 2 by two system I can actually just go ahead and solve it for a and b and that is the solution so this is so

simple that it's built into things like an Excel notebook or Excel spreadsheet where it says do you want to do a line fit or a quadratic fit or any polinomial fit so if you have a second order fit like a quadratic you you get a 2 x two system if you have a cubic you want to fit it to you get a 3X3 CTIC a 4x4 in other words I can just set this up as a

system of equations to solve for the coefficients that minimize the error and the data itself comes in here in terms of fitting the in terms of making up the Matrix a as well as the right hand side B so you solve that and you get your A and B it's a direct solve there's no optimization this is just simply solving a linear system of equations so that's how you do Le Square fitting okay and again we could do more

advanced curves we can do something like a parabolic fit so now you have a b and c to find but it's exactly the same process you want to find a b and c you take the error and you take the derivative respect to a with respect to B respect to C set it to zero and what you end up with is a 3X3 system of equations and all of these things are worked out so this is all hard coded

into python Excel mat lab any of the pro programming L languages have already done this so whatever degree of polomia you want to pick the system of equations for it has already been worked out and encoded inside of the software itself so you don't have to actually work this out it's been worked out but you should know how to get it as well right which is it comes out to be and not too bad of a form for instance for the 2x two and

then a 3X3 it's just slightly more complex but same concept so that's how we would do Cur fits with polinomial with the data so it's sort of limiting can we do more complicated things so for instance what if I want to fit something like this an exponential through the data so that might be something like I have a growth curve and instead of fitting a line I see this sort of exponentially growing

so what I want to do now is fit the exponential and of course you're very interested in the parameter a which is the parameter which dictates the exponential growth and the constant C that sits in front of it so you'd like to use the same architecture and say okay great let me do the same thing I did before let me Define my error here's my error E2 eror so now I put my function in here which is c e to the a x

of K and I'm trying to fit this to my data in the Square but now here's where it breaks down because now when you take the derivatives with respect to a and with respect to C what you get is not a system of not a 2X two linear system of equation what you end up getting now is a 2X two nonlinear system of equations so now this is problematic because you got to solve a nonlinear system of

equations so that's that's the problem polinomial always work out but soon as you start picking more complicated forms now you get nonlinear system of equations the problem with nonlinear systems of equations you don't know if there's a solution that exists or if there's multiple solutions that exists and so this problematic in general and this is the system of equations you have here and now you're also left to doing something like a gradient descent algorithm for instance

to try to solve this or something fancier but you know morally they're very much like the gradient descent algorithm walk downhill try to find this and we've already talked about this previously ly in in our lectures however there are tricks to this so people have been doing data fitting for a long time and things like exponentials are things that people would like to use often so we can do instead of trying to fit it to the

exponential what we can do instead is transform the data so I'm going to make a coordinate transformation and here's the coordinate transformation I'm going to go from Little X and Y which is the data I have to Big X and Big Y so little X will just be big X but now notice Big Y will be the log of Y and there's going to be a very specific reason I do it and in fact a coefficient B is going to be

log C because now when I do this essentially this function here that I was trying to find fit to now becomes a line so here it is let's just do the log of Y you can basically plug in here what you want to fit this to and now in this coordinate transformation you are actually doing a line fit in the new variables so what we've done is we've linearized the data so we took data where we're trying to fit an exponential

and we said instead I will take the log of the data and now fit a line to it okay so now that I fit a line to it now I just do exactly a line fit in this new coordinate system which I just showed you how to do so you're going to get a 2X two system of equations right for the a A and B and so once you find those you have the fit and then you can bring it

back to the original coordinate systems to get a and C okay so now you're back at line fitting and so if you can find a trick like this which you often can then you have yourself a situation where you can actually use very simple techniques like polinomial fitting and Le Square regression which is no optimization involved at all it's just a direct solve for solving on more complex data sets but this bring us to a a broader

point which is when we come to fit functions we typically are picking a set of functions here let's so here f is some function that I want to make you know before it was an exponential but it could be some very complex function I want to fit to the data and there could be some reason I have built this complex function right so it can be parameterized by a number of parameters here C1 to CFM so the real question

comes how do I generically go about fitting this to data when it's much more complicated well you do the same procedure you have before you define the error and now the error is a function of these coefficients C1 to C of M but you just say plug in the function I have try to match it to the data take the square of this and minimize the sum of the squares so take the derivative of this error

with respect of each of those coefficients C of J set it equal to zero and what you find is a large nonlinear system of equations and here it is and now you have to basically solve that and again there are methods to solve this including some gradient base NE me there's different algorithms for doing this that are available to you in Python so that we could actually try to solve this nonlinear system of equations so that's kind of the the

basic idea of all of this curve fitting you're going to Define an error that's really important it's going to set up the structure of your optimization and typically if people haven't specified it typically it's a root mean square error which is the square of the difference between your model and the data itself you sum up those squares and try to minimize it and so you get to pick the function and the error and then optimization will give you the values of

those parameters now in this case the optim ization we could work out for polinomial fitting exactly but a lot of times for like the Nar fitting you actually have to build yourself some kind of optimization tool to find the the the best fit in fact when we get to talk about neural Nets that's exactly what's going to happen we're going to do a nonlinear curve fitting and we're going to have to use infrastructure of gradient descent or more

sophisticated ideas to actually find a very large set of parameters which parametrize the fit to the data but for now these are very simple models and interpretable models that we can use and it sets up this structure of curve fitting now the reason too you should really pay attention to a lot of these lectures in curve fitting is that machine learning is essentially a very fancy curve fit and so the key ideas that come out of this curve fitting

lectures are going to apply later all the same concepts are going to be generated except it's going to be at a much bigger scale and that's the only difference really