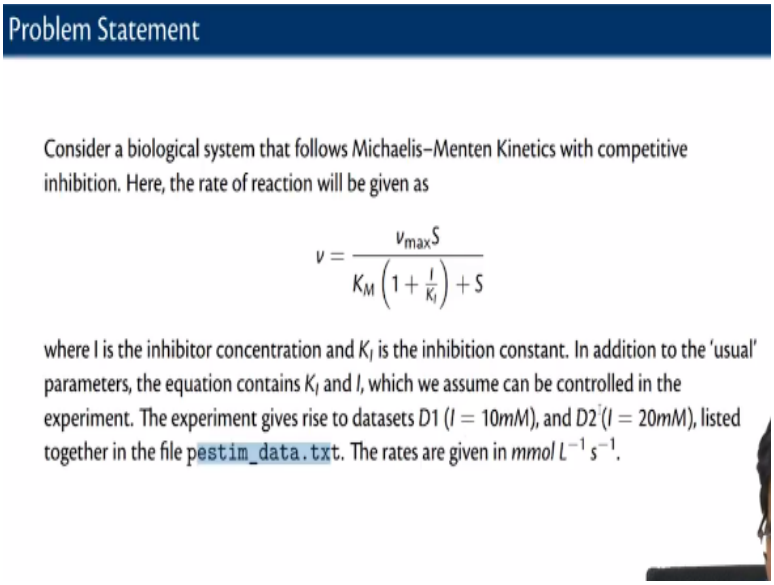
Computational Systems Biology
Karthik Raman
Department of Biotechnology
Indian Institute of Technology – Madras

Lecture – 50
Lab: Parameter Estimation

So, welcome to this lab on parameter estimation, today we look at how we set up the objective function using Matlab, you know for a simple problem like Michaelis - Menten and also study how we can use the classic Matlab optimiser which is called fminsearch.

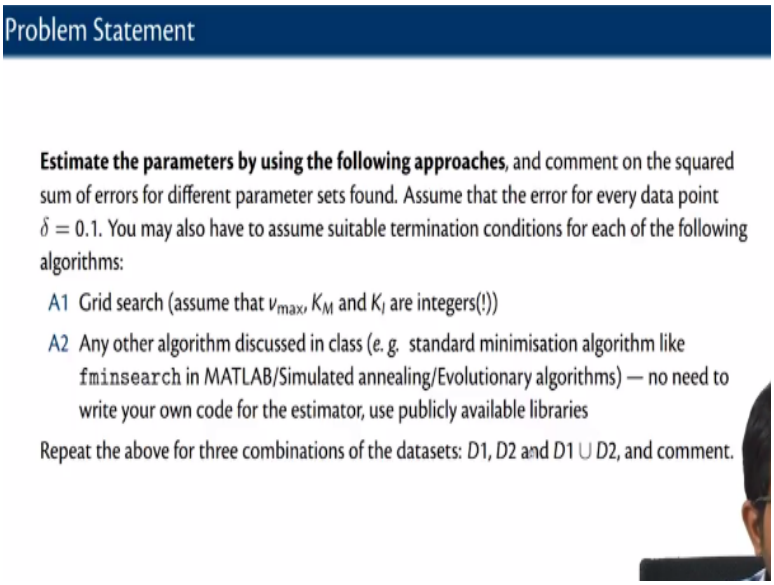
(Refer Slide Time: 00:27)



Today, let us solve a simple problem, let us consider a biological system that follows Michaelis - Menten kinetics with competitive inhibition, you might be called as the rate is given as v is v max * S/ Km * 1 + i/ Ki + S as you see here, where I is the inhibitor concentration, K is inhibition constant and there are few additional parameters here, so you have a total of 4 parameters; v max, Km, i and K, I is not a parameter essentially.

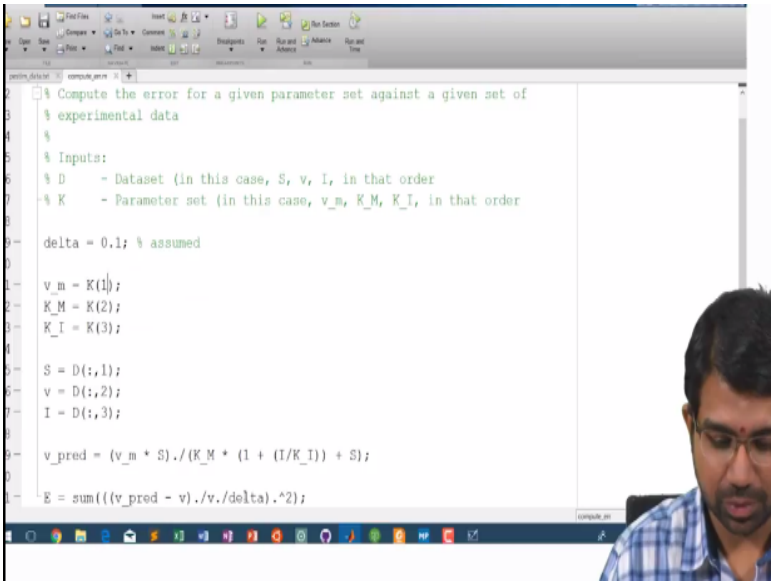
It is the experimental condition under which the; so it is the experimental condition under which the experiment is perform, so and there are 2 data sets in this given file, parameter estimation data dot text for I = 10 millimolar and 20 millimolar.

(Refer Slide Time: 01:19)



So, there are 2 approaches that we can follow to estimate parameters, one is a simple elementary grid search assuming v max, Km and Ki are integers and the second is any other algorithm, we have discussed in class, so may be a standard minimisation algorithm like fminsearch or stimulated annealing or evolutionary algorithms, there is no need to try and quote this up yourself, just try and use publicly available libraries.

(Refer Slide Time: 02:02)



And I want you to repeat this for 3 combinations, data set one unknown, data set 2 unknown and the combination of the 2 datasets, right, so this is there in the PDF, I share with you. So, you basically get hold of this data set, rest of you download the data set and the; got the data set, you

have problem statement. So, how will you go about solving this; you first need to recall the cost function, so how do you reconstruct this problem?

So, how would you set up a normal parameter estimation problem, first function that takes as input, so you need to essentially write some function like compute error of d, k; d, theta whatever, right, so once you set that up, you now need to pass that on to some direct search algorithm, the simplest direct search algorithm is a really simple, grid search algorithm, you assume all your parameters are integers.

What are your parameters in this case? vm, Km, Ki, you have 3 parameters, assume all of those are integers just for simplicity sake and you scan from 1 to 100, 1 to 100, 1 to 100 and find out what is the best solution and may be you can use one of those as the starting input to fminsearch, right, you will have to understand how to pass this; pass this your new cost function as the argument for fminsearch as well.

Because there is one minor difference, normal fminsearch assumes that your function takes in an input, here you will have an additional parameter as well, so it is usually in this case, I think it might be okay, we do not have initial conditions but sometimes you might have an initial condition as input as well, if you were integrating, here you may not do any integration. Did you see that, we do not need to do integration, why?

There is no differential equation, you already have the for, v is v max* S/ Km dash + S, so all you have to do is to you do not have to do any integration. **"Professor – student conversation starts"** sorry, the best; that will be the starting, you need to give a starting point for fminsearch. What is the starting point you give, you just give 000, if you want but it does not make sense, right, or you can give 111, some random starting point but instead you could take the best output of fminsearch; of grid search and pass it to fminsearch.

You need an x0 right, for doing any explanation, the entire space, do you have lhs in matlab, lhs design that is the one, you can use that too but I only meant for you to use a for loop because we said it all be integers, vm, Km, and Ki. Have you written the cost function; up to you; different

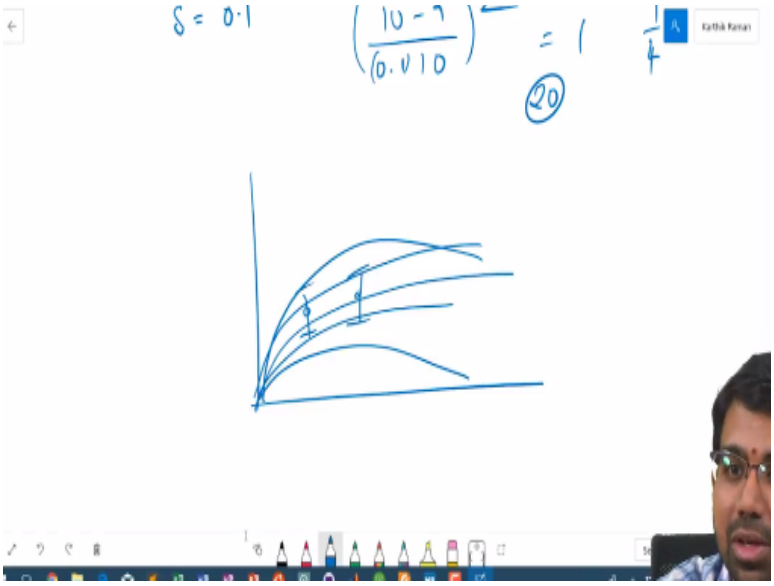
cost functions, right, so you will get a slightly better fit or a slightly worse fit. Here, it may not be too much of difference because the data is not scaled very badly.

The values are like 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 or something, so they are not like really that badly scaled, so I do not think it should be too bad. What I did was; I did normalise and also took a 0.1% delta, 1/10% delta, denominator, some 0.1 or something like that 0.1, 0.05 something, no not alpha, here we are assuming that it is not a stochastic. Anybody solved it, so the first thing you have to do is to establish a computer err dot n or your objective function, we should look somewhat like this.

It has to basically have your; you should compute the predictive value at each point and then this, just use the same cost function we have discussed in class so far, right, so x measured – x predicted / x measured * delta whole square added across all the points, right. For grid search, there will be 3 for loops, I am first talking about the cost functions, so in cost functions, you do not even need a, for loop, it uses a vectorized addition across all the data points.

And once you have the cost function, you just need to pass into fminsearch which is really simple, here you say, x = fminsearch of at x, so the parameter compute err of d, x and dataset, x and some initial starting point. So, you might find that for different initial points, we will get different x values and you might get weird values as well, so for example you; there is nothing that stops you from getting negative values of the parameters, right.

(Refer Slide Time: 25:10)



So, you need to know that they are not sensible, eliminate them and restart with a different initial point. So, the minimum error I am getting is something like 27.53 for the second dataset and 36.24 for the first dataset, which is not too bad because if you look at the cost function, let us say, x measured = 10 and delta = 0.1, right which means that you do not want to like penalise any number between, you know like 10% error, right.

So, if xp is 9 or 11 between 9 and 11, it is a good thing, right, so for 9 what is the error that you will get? We will get 10 – 9/ 10 * 0.1 whole squared that will actually be 1, right and you have 20 such points, so a total error of 20 is actually very reasonable but then you do not want the; of course, if you got 10 as the exact value, you would have got an 0, right for 9.5, what is it that you will get?

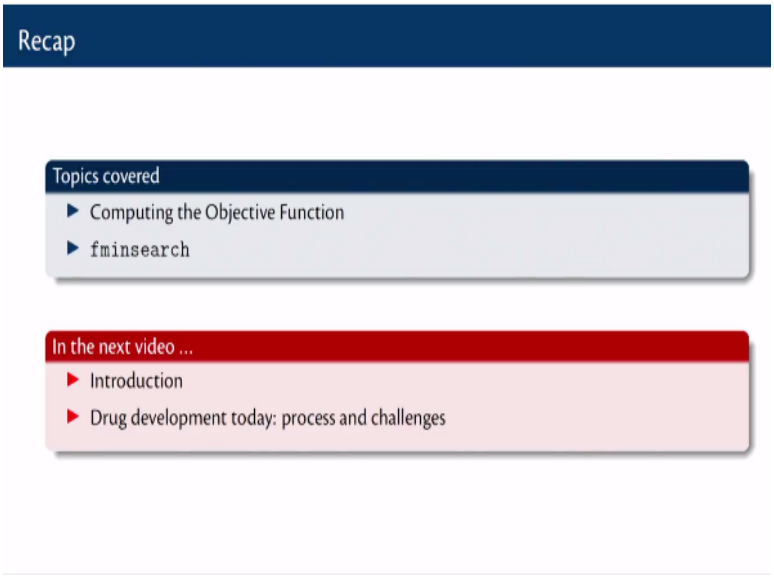
You will get 1/4 as a contribution to error but then you may want to then minimise the; this is post normalisation, right, so if you have 9.5, you will do 10 -9.5 / 10 which is basically 0.5 / 0 .1 (()) (27:19) so, you will actually get; okay, so what happens if out of 28 data points, 18 of them; 19 of them gives zero error and one of them gives a very big error but it still less than this 20, you might still think it is a good data point or a good parameter set.

So, you will have to again worry about it that is why your alpha will come into play or maybe you want to some minimise the median of the residuals or the maximum of the residuals

whatever, no, I am saying you are allowing 10% variation around the mean, right because we were discussing yesterday, right, so if you have a curve that looks like this, right you do not want to penalise something that goes between your error bars, right.

So, this is a bad curve or this is a bad curve but anything that passes, you do not have to pass through the mean to be a good curve, here you have only value of mean, you do not have anything else, right, so you only have; I am not given you multiple points for in at one time; for one values or something like that. If you had that you could use all of them that are one way to go about it. **"Professor – student conversation ends."**

(Refer Slide Time: 29:24)



Or you could compute this standard deviation and then use that to compute a delta and then say I will not penalise if it is better than the delta and things like that. In this video, I hope you got an overview of how we can set up the objective function for optimisation very easily in Matlab, right and you know how we compute the errors in a vectorial fashion importantly and you will also appreciate how we can use fminsearch which is the standard Matlab optimiser for solving such optimisation problems.

In the next video, we will take a jump and I have a interesting guest lecturer who works for this company called Vantage research which consults with the pharma in the space of quantitative

systems pharmacology and Dr. Rukmini from Vantage research will introduce you to drug development and the use of modelling in drug development over the next series of videos.