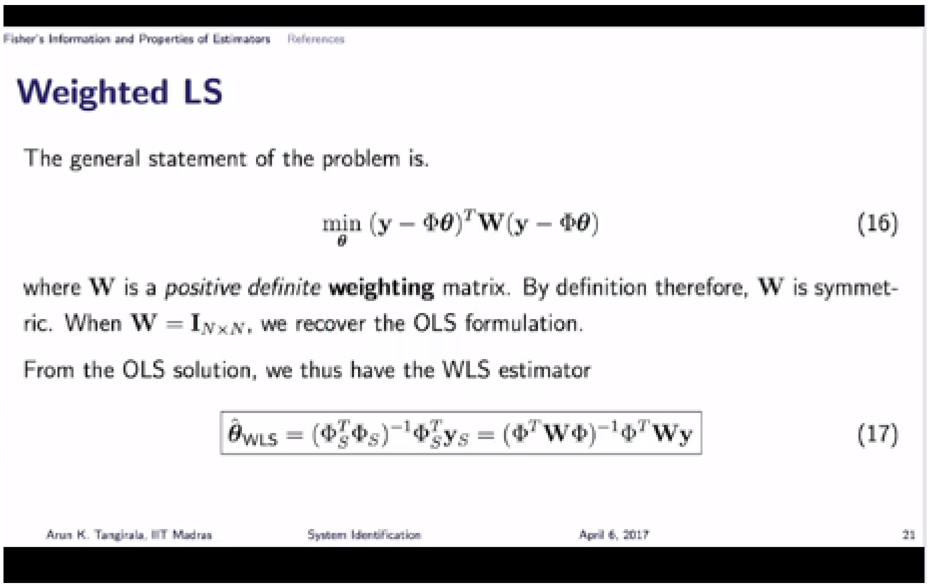
CH5230: System Identification

Fisher's information and properties of estimators

Part 12

Okay, so let's proceed. Hopefully, I've thrown enough light on all of this. Let's now talk about Weighted Least Squares.

(Refer Slide Time: 00:21)


Fisher's Information and Properties of Estimators References

Weighted LS

The general statement of the problem is.

$$\min_{\theta} (\mathbf{y} - \Phi\boldsymbol{\theta})^T \mathbf{W} (\mathbf{y} - \Phi\boldsymbol{\theta}) \qquad (16)$$

where $\mathbf{W}$ is a *positive definite* **weighting** matrix. By definition therefore, $\mathbf{W}$ is symmetric. When $\mathbf{W} = \mathbf{I}_{N\times N}$, we recover the OLS formulation.

From the OLS solution, we thus have the WLS estimator

$$\boxed{\hat{\boldsymbol{\theta}}_{\text{WLS}} = (\Phi_S^T \Phi_S)^{-1} \Phi_S^T \mathbf{y}_S = (\Phi^T \mathbf{W} \Phi)^{-1} \Phi^T \mathbf{W} \mathbf{y}} \qquad (17)$$

Arun K. Tangirala, IIT Madras System Identification April 6, 2017 21

And you know when you watch the-- when you have listen to the lectures on weighted least squares, I've explained to you under what conditions weighted least squares becomes necessary. There are different scenarios in which weighted least squares becomes important. When we talked about ordinary least squares there is one important point that we made, that is we give equal importance to all observations and that is justified under some conditions.

But if your sensor is such that the error varies with the times or various with operating conditions, right? Then you will have to give different importance to different observations or even when your errors are correlated, auto correlated. For example, earlier we discuss this, right? We said if I fit this model to data coming from this process here, what kind of estimates do I get efficient or inefficient? If I fit the model here, let's call this as M 1 and this as M 2. So, if I fit M 1 to the data coming from being generated from that process, I obtain inefficient estimates. Now weighted least squares offers you a solution, where you can obtain efficient estimates still working with the model M 1.But weighted least squares is not just meant for that, it is meant for other scenarios also. What are the other scenarios as we mentioned: Error Characteristic, Error Variance particularly could change with time and we call such error says heteroscedastic errors.

There are many sensors that have that kind of behaviour, where or it may not be sensor itself it may be something else, the sensor may not be contributing much it may be the disturbance that is contributing unmeasured disturbance, whose variance is changing with time and we call such errors as says heteroscedastic errors. By default, in OLS we assume what are known as homoscedastic errors, where the error is the same at any point in time.

That means we are dealing with a particular kind of non-stationary noise, where only the variance is changing with time. It's still white, that is still possible. And the other scenario is that maybe the error actually changing with your operating conditions. That's also possible. So in all these cases, it becomes important to introduce a certain weighting and that is why we end up weighted least squares problem. In fact, there is another scenario, where you run into weighted least squares problem, which is in model updation. Very often models have to be updated when you're implementing them online. You can't have the same model for a process for 20 years.

Because the process characteristics must have changed, you must have brought in a lot of changes in operating conditions. Maybe the equipment is degrading, many things can happen over a period of time, where you're compelled to update the model. And you're collecting this data that is being known acquired online and you use this data to update your model. What do you do in order to update your model, as I say, you know, try to live in the present, forget the past and so on. But don't do that for the course. Okay?

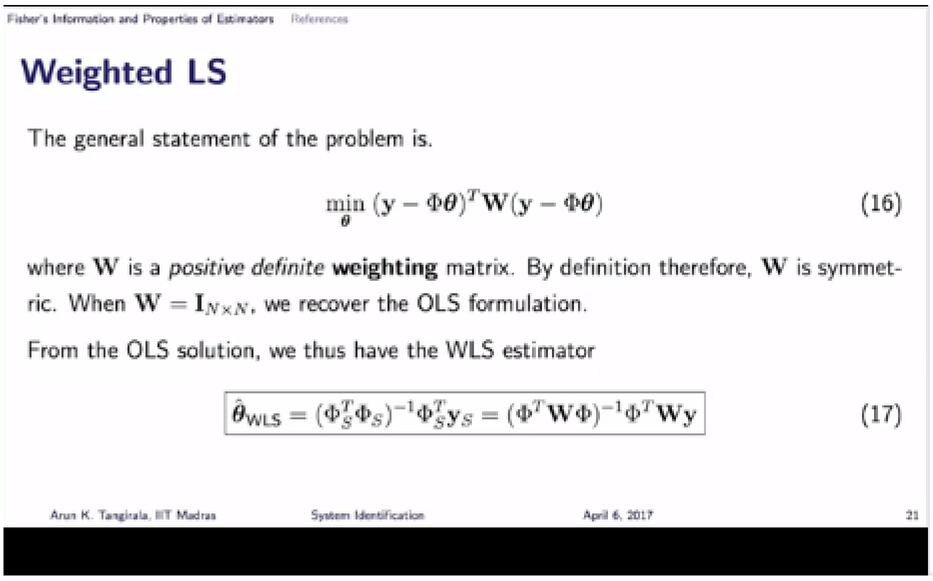
But don't apply to every aspect of your life. In model updation, we apply that in forgetting way. So, for example, instead of solving, I mean, instead of minimizing sigma epsilon square k, k running from let's 0 to n minus 1, I may want to minimize this objective function. I can say 1 over n, doesn't matter. So, why did we introduce this lambda, lambda to the power of n minus k? Or since it runs from 0 to n minus 1, I'm just going for simplicity, I am going to change this. I could have change n minus 1, n to n minus 1 either way it's the same. So what do you see there? The most recent observation acquires more-- the maximum importance. Lambda is a value between 0 and 1, if lambda is 1 you get back to your original ordinary least squares, OLS is also original least squares. When lambda is very small what are you implying? Let say lambda is 0.01. What is the implication? What are you trying to tell the algorithm?

So what happened? When lambda is 0.01 or you know, small values of lambda? What are you trying to convey to the algorithm?

Present [5:53 inaudible].

So very quick forgetting. Short-term memory. Don't really look at the data in the past. Almost, you're kind of ignoring it, right? Because lambda is being raised to n minus k, and as n, grows large, and lambda becomes small, the past data practically have no say in a parameter estimation. So through this lambda, you can control how long you want, how much what is a width of the window that you're willing to consider for the model. So, on an online basis, you run this algorithm and there is a recursive version of the least squares which I'll also talk about later on. So it is a recursive least squares that you implement with this forgetting factor, so that you don't have to really estimate work out a full least squares problem all the time. You can use the previous model and just add some correction factor to the parameters and the computation is much easier. But if you just look at it as a offline problem, what are you doing here? You're actually solving a weighted least squares problem. Where you are giving different weights to different observation. Each observation gets a different waiting. This is one of the simplest weight least squares problem that you would be solving. And the weighted least squares problem can be solved as I've explained in the lecture. Using the OLS thing, I'm not going to go through the solution procedure, but this is the final solution that you get for the forgetting factor case, for this model updation case. Lambda is called a forgetting factor. How does w look like?

(Refer Slide Time: 07:43)


Fisher's Information and Properties of Estimators References

Weighted LS

The general statement of the problem is.

$$\min_{\boldsymbol{\theta}} (\mathbf{y} - \Phi\boldsymbol{\theta})^T \mathbf{W}(\mathbf{y} - \Phi\boldsymbol{\theta}) \quad (16)$$

where $\mathbf{W}$ is a *positive definite* **weighting** matrix. By definition therefore, $\mathbf{W}$ is symmetric. When $\mathbf{W} = \mathbf{I}_{N\times N}$, we recover the OLS formulation.

From the OLS solution, we thus have the WLS estimator

$$\boxed{\hat{\boldsymbol{\theta}}_{\text{WLS}} = (\Phi_S^T\Phi_S)^{-1}\Phi_S^T\mathbf{y}_S = (\Phi^T\mathbf{W}\Phi)^{-1}\Phi^T\mathbf{W}\mathbf{y}} \quad (17)$$

Arun K. Tangirala, IIT Madras System Identification April 6, 2017 21

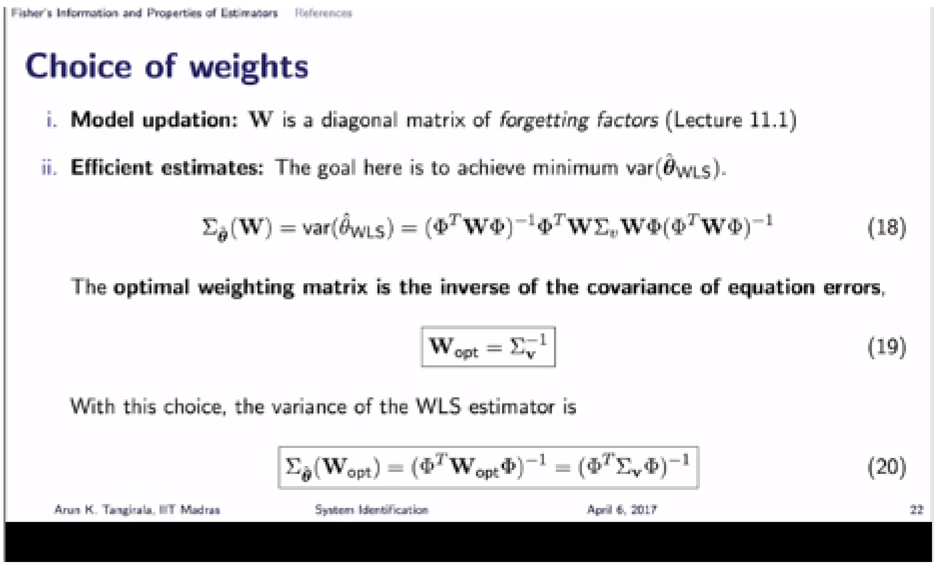
What is w? W is a matrix in general. It's a weighting matrix. Diagonal. What are the diagnosis consist of? Which weights? For that problem, what is w exactly, here? In this case, what is the W matrix? It's Diagonal matrix made up of? Yeah. So right from 1 to Lambda power, right. Okay. So that's how it is. So W is very easy in such cases. W, by the way, has to be always positive definite, so that you get a, you still solve the convex optimization problem. But in general, suppose this is not the model kind of problem that I'm solving, I'm actually solving something else. Some other weighted least squares problem where w is chosen based on some other consideration, correct? That is, I'm just trying to recap whatever I've said in the lecture before. Hopefully, so that the concepts are reinforced. So the choice of weights is an important thing.

(Refer Slide Time: 9:00)

<Type notes for the currently selected dictation here>

In model updation, it's very easy. W is simply a matrix, a diagonal matrix of forgetting factors. But, in a general scenario, I would like to choose W, such that I get efficient estimates. See, suppose, I have heteroscedastic errors. Again, heterscedastic areas are case where the variance keeps changing with time. And that is not, therefore, truly a white noise because in a true white noise is stationary, variance is the same at all times. Therefore, I know I'll run into inefficient estimates, but I want to still get efficient estimates, or maybe there is some other error characteristic that is not allowing me to obtain efficient estimates. In all such cases, I would like to choose W, in general, such that the estimates are efficient. Now it turns out that the optimal weighting is nothing but sigma v z inverse, that is, in general if you have y[k], equals psi transpose theta naught, plus z k, to make sure that I get efficient estimates all the time, I have written here v, but it should be actually z there.

(Refer Slide Time: 10:32)



Fisher's Information and Properties of Estimators References

Choice of weights

i. **Model updation:** $\mathbf{W}$ is a diagonal matrix of *forgetting factors* (Lecture 11.1)

ii. **Efficient estimates:** The goal here is to achieve minimum var($\hat{\theta}_{\text{WLS}}$).

$$\Sigma_{\hat{\theta}}(\mathbf{W}) = \text{var}(\hat{\theta}_{\text{WLS}}) = (\Phi^T\mathbf{W}\Phi)^{-1}\Phi^T\mathbf{W}\Sigma_v\mathbf{W}\Phi(\Phi^T\mathbf{W}\Phi)^{-1} \quad (18)$$

The **optimal weighting matrix is the inverse of the covariance of equation errors**,

$$\boxed{\mathbf{W}_{\text{opt}} = \Sigma_{\mathbf{v}}^{-1}} \quad (19)$$

With this choice, the variance of the WLS estimator is

$$\boxed{\Sigma_{\hat{\theta}}(\mathbf{W}_{\text{opt}}) = (\Phi^T\mathbf{W}_{\text{opt}}\Phi)^{-1} = (\Phi^T\Sigma_{\mathbf{v}}\Phi)^{-1}} \quad (20)$$

Arun K. Tangirala, IIT Madras System Identification April 6, 2017 22

It's okay. Hopefully, we'll note this correction. So, what we mean by choosing W as sigma v inverse? What do we mean by that? What we are saying is, first of all, what is sigma z there or sigma v? What is the dimension of sigma z?

N/M

N/M, correct. What is it actually? It's a variance-covariance matrix of your errors of this error vector, n observations. So, sigma z is going to look like this. Sigma square z at one. Up to sigma square z at N. When I use the symbol sometimes abroad, there are some students who don't like the z. They don't want to pronounce it. So, they'll say that wiggly thing. That wiggly thing. That's wiggly. Okay. So, the Sigma that wiggly thing is this. And then, of course on the off diagonals, you have up to lag N, auto covariance up to lag N-1. So this is a big sigma that is the optimal weighting. Inverse of this is the optimal weighting. What does it mean? You should learn to interpret as I've explained in the lecture also with an example. Suppose, for some situations, your sigma z is diagonal. Okay? In which case that you're looking at heteroscedastic errors case, where it is diagonal, but the diagonal elements are all different. Suppose that is the scenario. Then, it becomes easy to interpret this result, that the optimal weighting is the inverse of the noise covariance matrix. Suppose this is a scenario, then you the heteroscedastic error case. And there is an example that we worked out in the lecture. In this case, what is W? Optimal weighting. It says optimal weighting should be inverse of the noise covariance matrix.

One by the diagonal elements.

One by the diagonal elements. In other words, I'm going to minimize this problem. Minimize this objective function, one by sigma square, let's say k, epsilon squared k. One to N or whatever. So this is what I'm going to do. What does the objective function now tell me? It says for every term given what kind of importance, scale it down. You can say scale down the error or the squared error by the variance or scale down the error by the standard deviation by the variants of the error at that instant. What does it mean? If I have observations with large errors, then such observations are going to be given lesser importance. We saw the observations which have small errors, because smaller so would mean smaller variants. That makes sense, a lot of sense intuitively. If you have a bag of observations, and each observation comes with a different error variability, obviously the observation with lower variability is more reliable and has to be given more rating, then those observations which have higher variability because then they are less reliable.

So, in that sense, this result makes a lot of sense that the optimal weighting should be the inverse of noise covariance matrix. Now, the other way of looking at this result is equal into scaling the data itself. I earlier said, here, you're giving weighting to epsilon squared k, you're attaching a weightage of one over sigma squared k. But the other way of looking at it is that epsilon k is being scaled in this way. And what does epsilon k consist of? Epsilon k consists of y[k] minus y hat of k. And y hat of k. Let's assume a linear regression is simply psi transpose k theta. So, this is what is happening inside this box. Instead of weighted squares, what is happening? y[k] minus psi transpose k theta is being scaled by sigma k. In other words, y[k] is being scaled by sigma k and so are the regressors. Right? Which means instead of working with original data, what are you working with? Scale data, that's all. So, you can think of weighted least squares as ordinary least squares on scale data. And that is a perspective that we use to derive the solution.

If you, again, go back to the lectures, you will see that is exactly the perspective that we use to derive a solution to the weighted least squares problem. And that's a very, very nice perspective. Because what it says is that, if you scale your data appropriately, prior to estimating the parameters, you're not changing the model structure, you're not changing the data. This is different from filtering the data. This is called scaling. This is not called filtering. Okay, you have to understand the difference between scaling and filtering. In filtering, you also take past output, you apply a filter and so on. Okay. In this case, you're only scaling the data. All it says is if you work with the original data, you may end up with inefficient estimates of the parameters. That's because OLS gives equal importance to all data, right? And when your errors have different variability, you should not have given equal importance in the first place.

But weighted least squares account for that and comes back and says, don't do that, give whatever importance is due to each observation. And the theory tells us that the optimal weighting is in this way. And what does it essentially tell me? It says that instead of working with your data, which has heteroscedastic errors, work with scale data. In the scaled domain the errors are going to be homoscedastic. All the scaled observations will have the same error characteristics. And then you are applying in OLS. That's all. So weighted least squares should be thought of as either giving different weights to your prediction errors or that you're working with scale data. Either perspective is okay. Whichever perspective helps you that's important. That you can work with. Okay, so this is so much about weighted least squares and...