Spatial Statistics and Spatial Econometrics
Prof. Gaurav Arora
Department of Social Sciences and Humanities
Indraprastha Institute of Information Technology, Delhi

Lecture - 15B
Spatial Interpolation and Kriging

Welcome back to the second part of lecture 15. In this lecture, we are going to develop the general idea of Spatial Interpolation right? If you have worked with time series data, you must be aware of interpolation. You know missing values are a typical problem in sample data sets and whenever the values are missing, one of the ways to deal with them is to fill them back using a statistical approximation which is also known as spatial statistical interpolation.

When we conduct this exercise in space, it is called spatial interpolation. Remember, however in space, we have missingness is fundamental because we cannot possibly sample every point in a domain of interest right? So, interpolation or spatial approximation, or spatial prediction on locations that remain unsampled is a fundamental problem with spatial data analysis right ok. Let us move forward without any further delay.

So, spatial interpolation is also called kriging. Kriging is synonymous with optimal prediction. We consider the problem of predicting groundwater level at an arbitrary point in space. So, what you see on the right bottom corner of your figure that you have these 3 sampled locations s_1, s_2, and s_3 where you know what is the groundwater depth and you want to know what is the groundwater depth at s_0. Remember, we did this for one dimension.

Now, we are bringing the problem to the two-dimensional space and we are generalizing it ok. A convenient estimator is a weighted sum of these values that are available right?

If there was no spatial dependence and I was working with a series of data with no dependence, then, G s_0 is just G s_1 multiplied by one-third plus G s_2 multiplied by one-third plus G s_3 multiplied by one-third. That is to say that you know I have these weights lambda 1 equals 1 by 3, lambda 2 equals 1 by 3 and lambda 3 equals 1 by 3 being multiplied by each of the observed values and then, summed together so that I get my value which is unknown a prediction for the unknown value right?

Now, in this case, it is a little bit more complicated in space. The first thing is that G s_0 which is location s_0 which is the unknown location of interest is closer to locations s1 and s_2 than it is to s_3. So, there is this idea that you know s_1 and s_2 may be a better representation of s_0 than s_3 just because they are closer to the two. This will imply that I cannot possibly have the same lambda i's in the case when I am working with non-spatial, non-dependent data sets right?

So, I have provided you here a convenient estimator where these weights are a typical way method of predicting data, you know conducting prediction; but here the weights are not going to be equal. In the sense of what is how each sampled location is weighed in space. We have this extra component which is 1 minus summation i equals 1 to 3 lambda i times G bar; this extra component is just to ensure the unbiasedness of the estimator which is G star s_0.

And this also ensures unbiased through the fact that the sum of weights should equal 1 right? If I do not include this component, I cannot ensure that the sum of you know these weights will be equal to 1 right? If you have worked with constraint optimization before and you know how to sort of include constraints in the Lagrange function, then you can look at G bar as the Lagrange you know multiplier and 1 minus you know summation i lambda i being the constraint that summation i lambda i must equal 1 right?

So, it is the same formulation; just written out conveniently for our purpose. So, lambda i's are data weights and the G bar is the global mean. So, we have this you know convenient estimator. We have data for G s_i, we have data for G bar, what we do not know are these weights. So, the whole problem in this lecture is going to boil down to figuring out what the weights lambda i's are.

So, again, similar to what we did in the previous lecture, what if we are working with non-stationary data? Well, if you are working with let us say there is a domain on the right-hand side that way you have these code values; remember this data are coming from Michael Pyrcz, at UT Austin. We saw a poster, where this figure was there; where you have these scores being dug out of the earth, you know at different locations and there is a domain of interest with you know a black dashed line.

But what if there are these domains, this larger domain is made up of two stationary; two stationary domains when you know, but when they are sort of unionized together, they are clubbed together, all in one is a non-stationary domain because there is a structural break in

between and what if the value that I want to estimate lies in neither of these domains. So, the idea is I must use data in both domains and I must figure out a way to combine them.

So, the way, we go about it is that we write down you know y si as G s_i minus G bar for each domain ok. So, we just reduce or deduct you know reduce the total value observed by a domain mean. Remember G bar will be different in domain 1 and it will be different in domain 2 ok. So, I am simply reducing the observed values by the spatial by the global mean in these sub-domains and the resulting residual is what I am used to as the stationary variable.

So, instead of G s_i which is a non-stationary random variable, I am now using y s_i which is a stationary residual. So, the problem then boils down to predicting y star s_i and after I get my y star s_i, I will add this global mean back to the data right? But so, my variogram estimation of my spatial dependence measure estimation will happen with the stationary values, right? So, the idea of intrinsic stationarity is still holding ok. So, let us move forward.

So, now, the basic query as I said is what should go into lambda i. Here, I have the first thing was you know the closeness. So, how close is s_0 and s_1 to each other; how close are s_0 and s_2 to each other and how close are you know s_0 and s_3 to each other? Clearly, s_1 is the closest you know quite similar not so farther apart is s_2, and quite farther apart is s_3. Data redundancy is very interesting. Now, data redundancy is the idea of spatial dependence. G s_1, see that s_1 and s_2 are quite near to each other.

In fact, the distance between s_1 and s_2 is smaller than the distance, each of their distance from the unknown location. If there is strong spatial dependence in data, then these two values are going to be correlated with each other. That would mean that although I am looking at two unique observations, the information that can be derived from these observations is not equivalent to two unique observations.

This is an idea that we discussed at the beginning of this you know one of the early lectures where we said what is the effective n prime when rho equals 0.26. Remember that lecture, you know earlier in this class, right? You can go back and check, where we worked with a one-dimensional Z_1 to Z_n data, we first did not have spatial dependence; then, we introduce spatial dependence and we found the data size.

The data size reduces, there is a reduction in the data and this reduction is equivalent to the fact that there is a correlation in the data which is reducing the effective amount of

information contained in the data. This idea is called data redundancy. And finally, there is the direction of spatial contiguity. Now, the data may have strong East-West contiguity than North-South.

There is anisotropy for example. In that case, you are going to have a problem that even though s_3 is farther apart or s_1 is very nearby it's really you know in the North-South direction from you know s_0. So, these are some of the interesting factors that must come into account, when we decide lambda i. So, lambda i should be ideally a function of A, B, and C right ok.

Now, what are the common weighting schemes? If you have worked with you know spatial data, commonly what people do is they use these equal weights 1 over n; like I said if you have non-spatially dependent data, you go ahead and you put in 1 over 3, 1 over 3 and 1 over 3 to each of these values. Calculate the mean and say this is my prediction; this is my best guess. The other thing is inverse distance squared. So, here, you are you know controlling for distance in a deterministic sense, right?

So, you are taking the distance and squaring it, providing an index of dependence between these variables that are solely dependent on the distance. The first commonly used weighing scheme which is just the mean, the global mean in the domain of observed values does not account for either of the spatial factors which is the closeness of the data redundancy or the direction of spatial continuity. The second one which indeed sort of accounts for closeness does not account for redundancy and the direction of a spatial dependence ok.

So, you know our task is to construct the weights that account for all the above criteria and yield an optimal prediction of G 0 which we call G star as 0 right? That is the prediction.

So, to do that, let us work with stationary data and consider a linear estimator for de-meaned groundwater location at a level at the location as 0. y star i s_0 which is the prediction at s_0 is equal to the summation of all the weights multiplied by the observed locations stationary or de-meaned groundwater level data implies that expectation of y is equal to 0. That is why we do not need the constraint which is 1 minus summation i equals 1 to n lambda i times y bar. This y bar is equal to 0.

So, the constraint is automatically satisfied that the sum of the weights will auto now intrinsically, implicitly not intrinsically the weights will implicitly sum to 1 ok. And then, I

have my intrinsic stationarity you know formulation which gives me my variogram formulation as well which is 2 gamma h equals expectation of the difference square.

Sorry for the typo there, it is the expectation of difference squared, where we are taking the difference between observed values at two locations s_i and s_i plus h which can be also called s_j in general. The estimation variance; so, the truth and the prediction, when I take a difference between them and I square them and take the expectation, I get what is called estimation variance. This is the variance of my estimate.

This estimation variance can be written as expectation y star squared minus twice of expectation y star into y 0 plus expectation y squared. If you are having any trouble you know visualizing this, just take what is inside the expectation operator and expand it, just take the squared; you have y star s_0 squared plus y s_0 squared minus 2 y star s_0 y s_0. Because the expectation is a linear operator, it enters the brackets and applies to each of the terms individually and that is it; that is all that you are seeing on the right-hand side here.

I am going to expand on these things a little bit further. So, what I am going to do is I am going to take y star s_0 and replace it with y star s_0 is nothing but summation i lambda i y si. So, I am going to just substitute the unknown value with the weighted sum everywhere that I see it.

And then, I am going to expand it. So, I am going to keep the expectation operator you know as it was, and I am going to now expand these things. So, I have because y star you know I have expectation yi s_i squared minus twice sorry sum expectation or summation of lambda i y s_i whole squared plus summation of lambda i y s_i times y s_0 which is the truth remember plus expectation y s_0. Now, this summation y i lambda i y i squared can be written as summation double summation y lambda i lambda j y s_i and y $s_{j,}$ and similarly, I expand this further.

Expanding this thing further, I take the expectation operator in. It is a linear operator just like the summation. So, it sort of starts entering in and it goes and applies it to the random variables, right? So, these are all constants lambda i lambda j are constants and you know so the expectation operator moves right in. Similarly, in the next you know as the next component of this summation, the expectation operator just keeps moving in as you see it.

Now, finally, this expectation y i y s_j when, remember the mean of y at s_i is 0 right because you know it is a residual, I am working with a residual. So, because the mean is 0, the

expectation of the product of y s_i and s_j can be written as the correlation; sorry the covariance between s_i and s_j. Similarly, in the second component, I can write is this expectation y si and y as 0 as the covariance between s_i and s_0 and because again y bar is 0, I can imagine all of these as written as y s_0 minus y bar the whole squared right?

So, because of that I can write this as simply variance of y at s_0 which is the unknown location being c 0 because you are working with stationary data, the large-scale variation is the same everywhere. It is a stationary variance scenario, right? So, c 0 applies to unknown locations as well as known locations. So, what happens is that the last term turns out to be the large-scale stationary variance which is the cell of the variogram, the first term is about data redundancy and the second term is about data closeness.

So, we have accounted for data variation which is the second moment in data, we accounted for redundancy and we also accounted for the data closeness. So, we have all the factors now that I wanted to account for. So, let us move forward.

So, what is the next step in retrieving optimal Kriging weights by you know optimal Kriging weights? What we do is now that we have a sample estimation variance of the predicted y s 0, what we do is, we want to minimize this variance? The lesser the variance, the more accurate my prediction, and when I minimize this variance, I basically choose these weights lambda i.

And automatically, these weights will be a function of redundancy, closeness, and the large-scale variation in data because the estimation variance is a function of these three factors right? So, I set up my objective function here, I write down my first-order conditions, I have n simultaneous equations, and I have n unknown weights. So, I have n equations and n unknowns ok.

It turns out that you know you will have a more very convenient sort of formulation of weights which we will see in the matrix form going forward. But the idea is that I will have the lambda i star which are the optimal weights and remember, these weights account for data redundancy, they account for closeness and they account for large scale variation in data right?

So, unlike the deterministic weights which are the inverted distance squares, although they are counting for the distance, they are not accounting for large-scale variation in data, they are not a random variable-based understanding. It is a deterministic understanding of the

world; it is a physical understanding of the world that just by looking at the distance between two locations, I can say, what values will be realized at those locations. That is a very limiting idea even intuitively, I suppose you know, I am sure all of you can understand that.

So, let us move forward and look at an example and go back to our example, where we started. So, that we can understand this process with the example as well. So, we have these three sample data points s_1, s_2, and s_3 locations; where the sample data points are G of s_1, G of s_2, and G of s_3. The main objective that I have as an analyst is to be able to predict the value of groundwater level at location s_0.

The first thing I must do is assume spatial stationarity; if it is non-stationary, I should create a filter, construct a filter, de-filter, apply the filter to my data, de-trend it, de-sort of mean it, remove all the non-stationarity, and then you know work with the residual. So, here, I am going to start by assuming stationarity in the data. But it's an imperfect assumption that almost always need not hold with real-world data.

Then, G of s_0 is nothing but the weighted sum of all the available values which is lambda i G s_i summed across all three values that is, i summation i equals 1 to 3. Our objective here is to figure out the optimal weights lambda i star. The first thing that we hint, that we have from our exposition previously is that we have to minimize the estimation variance.

So, we have to then figure out the estimation variance and try and minimize it. So, these first-order conditions as they were in the previous slide, will turn out to be very convenient. So, the first condition is lambda 1 C s_1 s_1; remember this is nothing but you know C 0. This is C s_1 s_2; so, this is C s_1 minus s_2 right? This is going to be dependent. This is the co-variogram at lag s_1 minus s_2 and this is the co-variogram at lag s_1 minus s_3 which is equal to the total co-variogram of lag between s_1 and the unknown location s_0.

When you look at the first-order condition, it is going to be very easy to write the second and the third; they are going to be just cyclic. That is, I will just change this s1 to s2 everywhere and I will get my second first-order condition with respect to lambda 2. With respect to lambda t_3, I again just replace the 2 with 3 in the first component of these co-variogram devices and I am done.

So, it's just a convenient form because you know it's just because of the specification of the co-variogram. Now, the point is where do these values come from?

These values are going to come from the variogram model C si s_j is C 0 minus gamma s_i minus s_j. This gamma is my variogram model. This C 0 is the sill which is also estimated from data. So, these values all on the left-hand side are just data, data, data; I can say data-driven, this is data, data, data and these here are unknowns right? so, lambda 1, lambda 2, and lambda 3 are all unknown. These are variables and what about C s_1 s_0? Well, this is two data; I mean this is nothing but C 0 minus gamma s_1 minus s_0.

Although I do not know the G of s_0, but I can calculate the length or the h vector or the distance between s_1 and s_0. That is not too hard. I know these locations deterministically right? So, all the components can be backed out from data by estimating the variogram model that is why we studied lecture, you know in lecture 14 that is before we came to interpolation or training, we first studied the variogram model; the variogram model estimation, you know fitting a model, a goodness of fit criteria and so on and so forth.

Now, for this example, I can write this system of an equation which is written as linear equations into a matrix form which is again a linear form. So, I have all the redundancy factors C s_1 s_1 blah blah blah and large-scale variance sitting in this matrix called a redundancy matrix R. These are my unknown lambda vector right? this is lambda 1, lambda 2 lambda 3 which is the unknown vector. If I get this vector I am done and finally, I have the closeness matrix C which is the closeness between unknown and known locations.

Lambda star is just equal to R inverse C which is pretty clear because we are taking an inverse of R, we need that the R determinant of R is non-zero; otherwise, the inverse cannot be calculated. Of course, you can, I mean there are some advanced topics like pseudo-inverses; but I am not going there in this course. We will also require R to be a positive definite matrix to ensure a unique solution for lambda star.

So, if you are to get a unique solution, we also require our R matrix to be positive and definite. These are very important theoretical underpinnings and you will do all of these things with software, you know when you study R sessions and R $G_{i's}$ session, we will do all of these things on software; you are not going to be actually calculating these things.

But oftentimes when you actually estimate these things, the software just does not stop the loop and it does not give you the weights or does not give the estimate. If the software does not produce a final answer, where do we go? Well, we go back to this black box or the theory and try and figure out what may be going wrong. Well, what may be going wrong are either

of these conditions which is why the functionality of how the process of getting to the optimal weights and the optimal prediction is as important as learning the syntax of software.

So, although you know none of this is done manually; but, going over it manually is very very useful. So, I encourage you to at least twice go through each step of an optimal Kriging estimator, starting from the example to this more generic form in the two-dimension ok alright.

So, some sort of you know notes to end the lecture. First, the Kriging weights and consequently the Kriging estimator account for the distance of information, configuration of data, the spatial configuration of data, and the structural continuity in the data right? It is a very sophisticated estimator ok. The Kriging estimator is unbiased that is the value that you are going to get is going to be in expectation the same as the truth.

The Kriging estimator also minimizes the variance of s_j; this is by definition, right? the definition by which we back out the Kriging estimator or Kriging weights is by minimizing the variance and the measure of the Kriging variance that is sigma squared E s_0 is going to be lower than the cell which is the large-scale variation in data.

So, the variance of the predictor, the optimal predictor at unknown locations is going to be by definition smaller than the large-scale variation in data. It's a property of the Kriging estimator. With that we are done with spatial estimation, going forward we are going to move to the last module in this course which is about spatial regression and I have titled it as you know spatial econometrics.

Here, we are going to first do a recap of a regression model, how to interpret it in space, and what differs and then, we are going to move towards the next step, we will integrate the variogram, and see how the variogram model can be integrated into the spatial regression model?. And then finally, we will move from you know we will learn the theory of moving from a correlation to causation.

We remember again, spatial regression is usually done on software; how do you sort of account for spatial effects is all done on software. But it is equally important to learn the theory; just like in the case of Kriging, in the case of the variogram, although everything is highly computational, and it's very very important to learn the software. At least as important

to learn as it is important to learn the theory if you have to actually use it for any practical purposes, but the theory is very very important as well.

So, I hope this lecture was fun for you and I look forward to having you in the next module which is called spatial econometrics.

Thank you.