Data Processing Using Python Script and ArcGIS ModelBuilder

PIL KWON
Department of Civil and Environmental Engineering
Seoul National University
35-209
1 Gwanak-ro, Gwanak-gu, Seoul 151-742
REPUBLIC OF KOREA (SOUTH KOREA)
pil0706@snu.ac.kr http://pilkwon.efoliomn.com

Abstract: - The amount of data that one can acquire is limitless. And, it can be easily purchased from commercial companies. In most cases, the data are exported or delivered in spreadsheet, which are compatible with ArcGIS, the most widely used Geographic Information System (GIS) software in GIS sector. However, the data from commercial data providers contain unnecessary data that do not need for a certain project. Using the raw data can increase processing times and reduce performance of geoprocessing tools. This study shows how the raw data in Excel format is processed using ArcGIS ModelBuilder and Python script.

Key-Words: Python, ModelBuilder, ArcGIS, GIS, Python Script, ArcMap, Local Moran's *I*

1 Introduction

ArcGIS, developed by ESRI, inc., is the most frequently used software in GIS field. The powerful GIS software is compatible with many different types of file format, such as Excel, dBASE, Shapefile, and more.

Shapefile displays point, line, polygons, and attributes. Among the displayable objects, the attribute in Shapefiles is based on dBASE file format, which is one of types of spreadsheet, and can be compatible with Microsoft Excel.

In many cases, GIS researchers and engineers work with spreadsheet and then import the results to ArcGIS to join the spreadsheet and Shapefiles. That is because ArcGIS is not fully designed to work with other data types but Shapefiles. This is not only inefficient, but also able to cause mistakes.

Furthermore, editing raw data is troublesome if the data are big and the data processing with two different software.

One of the greatest functions of ArcGIS is that the software allows users to modify or create new geoprocessing tools using Python programming language. Also, the software lets the users to create their own tools with mixing with embedded ArcGIS Tools using ModelBuiler. According to Dobesova, the data flow model in ModelBuilder can be designed easily by drag and drop any geoprocessing function from ArcToolBox [1].

The intuitive user interface of ModelBuilder will make accessing and editing big spreadsheet data in ArcGIS environment easier and enhance the accuracy and efficiency for GIS engineers.

This study is going to show how the gigantic size of Microsoft Excel data are going to be edited and organized under the ArcGIS environment using Python script and ArcGIS ModelBuiler. This advanced GIS techniques of using ArcGIS ModelBuilder and Python scripting for handling big data within ArcGIS are going to reduce processing times for data analysis.

2 Data Preparation

To show workflows and examples with ModelBuilder and Python programming language, a hypothetical test was conducted. The hypothetical test for this project was arranged to see cluster or disperse of residential places' housing prices over time using Local Moran's *I* (residential real estate price' spatial distribution).

For this project, housing address, prices, and sale date were required. A simple workflow for this project is described below.

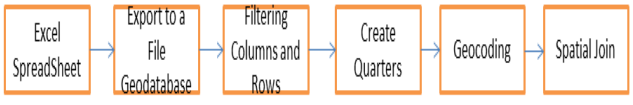

Figure 1: Workflow for Testing

The data used for this test were purchased from DataQuick® in San Diego, California, USA. The data contain a hundred thousand housing transaction records from 2000 to 2009.

ISBN: 978-1-61804-150-0

2.1 Removing Irreverent Fields

The data contain California real estate transactions records such as address, owner, loan type, zone, previous owner, owner's mailing address, and more. For this project, only required fields were address, price, and date among 134 fields. Unnecessary fields, most of the fields, needed to be removed under the ArcGIS environment. The raw data (Excel spreadsheet) was imported into a file geodatabase table to have better performance.

Officially, the limitation of rows in a table in a file geodatabase is 4,294,967,295 (Maximum limit of Microsoft Excel rows is 65,536) [2] [3], the data for this project were not over 4.2 billion records, it was do-able.

There are python processes like delete fields and update cursor. Yet, rather than deleting 131 unnecessary fields; keeping fields that are required for this project was implemented using for loops and delete fields. In other words, keeping 'address', 'sate date', 'sale value' and delete everything is easier to implement and faster than deleting each of 131 fields. In addition, housing prices transaction less than $50,000 US Dollars were removed, for the records were not meaningful (It was assumed that prices less than $50,000 were transaction between family members).

Figure 2 represents a snap shot of the python code of deleting unnecessary fields and rows.

```
import arcpy
arcpy.env.overwriteOutput = True

table = arcpy.GetParameterAsText(0)

keepFields = ["Address", "Sale_Date", "Sale_Value"]
deleteFields = []

fields = arcpy.ListFields(table)

for field in fields:
    if not field.name in keepFields and not field.required:
        deleteFields.append(field.name)

arcpy.DeleteField_management(table, deleteFields)

rows = arcpy.UpdateCursor(table)

for row in rows:
    if row.Sale_Value <= 50000:
        rows.deleteRow(row)
    elif not row.Address:
        rows.deleteRow(row)

arcpy.AddMessage("Finished")
```

Figure 2: Fields and Row Deletion

Python programming language, therefore, can provide faster and more efficient process.

2.2 Creating Quarter

In social science, real estate analysis is often conducted in quarter. Based on the data's sate date field, one can extract month. And with the month information, quarter data can be created in another field. For this, a new function was defined and the function will update each row using update cursor. If sale months are falling one through three, it will return one (first quarter). Four through six will return two (second quarter), and so forth. The summary of script is described in Figure 3.

```
import arcpy
arcpy.env.overwriteOutput = True

table = arcpy.GetParameterAsText(0)
Outputtable = arcpy.GetParameterAsText(1)

import datetime


def Pil(salemonth):
    if ((salemonth == 1) or (salemonth==2) or (salemonth ==3)):
        return 1
    if ((salemonth == 4) or (salemonth==5) or (salemonth ==6)):
        return 2
    if ((salemonth == 7) or (salemonth==8) or (salemonth ==9)):
        return 3
    elif ((salemonth == 10) or (salemonth==11) or (salemonth ==12)):
        return 4

arcpy.AddField_management (Outputtable, "Quarter", "DOUBLE", "", "",
                           "", "", "", "", "")

tableRows = arcpy.UpdateCursor(Outputtable)

for tableRow in tableRows:
    saleDate=tableRow.getValue("Sale_Date")
    tableRow.setValue("Quarter",Pil(saleDate.month))
    tableRows.updateRow(tableRow)

del tableRow
del tableRows
```

Figure 3: Define a New Function, Quarter and Update Cursor

This figure also shows the greatest flexibility of Python script in ArcGIS.

2.3 Geocoding

The tools introduced in previous sections will provide tables with only necessary fields. Based on the address fields, there are two ways to geocode. One is he or she can create address locator. Another option is he or she can utilize ArcGIS online geocoding services. The latter method was chosen for this study because the online services use NAVTEQ's 2012 data for its address locator [4] and it is widely used in commercial GPS market. It also offers flexibility of finding locations with minor typos. The online address locator can be freely acquired from ESRI's website, '*http://tasks.arcgisonline.com/arcgis/services*' is going to be added to ArcGIS server under the GIS Server directory by the given URL. The online address locator was embedded to ArcGIS ModelBuilder (see Figure 4).

Once the address locators are embedded, it can be easily dragged to ModelBuilder. In order to apply this tool to his or her data, internet access is required.

ISBN: 978-1-61804-150-0

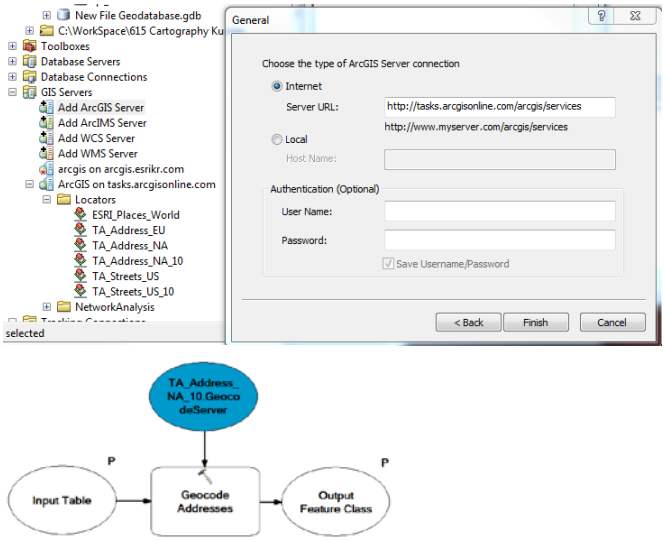

Figure 4: Geocoding Tool Interface

2.4 Spatial Join

Local Moran's *I* finds polygon or areal spatial pattern within a study area. That is, the geocoded points have to be aggregated and joined to spatial boundary.

Spatial boundary was decided to use the United States' census tracts from the United States Census Bureau.

ESRI's definition of spatial join is joining features to other features based on their spatial relationship [5]. Simply, points falling in a census tract will be aggregated and produce median values of the housing prices for the census tract.

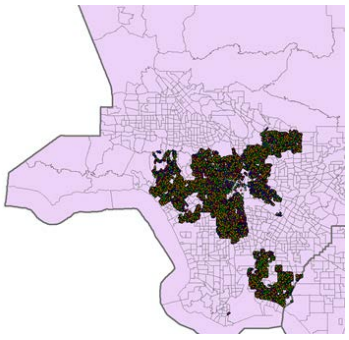

Figure 5: Los Angeles County Housing Points and Los Angeles County Census tracts

The point features' attribute that are on census tracts' will be joined to the census tract polygon.

Since the data contain quarter information and the hypothetical test for this project was to see real estate prices' spatial pattern over different time sequence, the census tracts also have to have the time information. For this reason, the geocoded point features were divided by its quarter and spatial joined independently (one input; for outputs). Using iterate function in ModelBuilder automated this tedious work. Figure 6 shows the model's work flow.

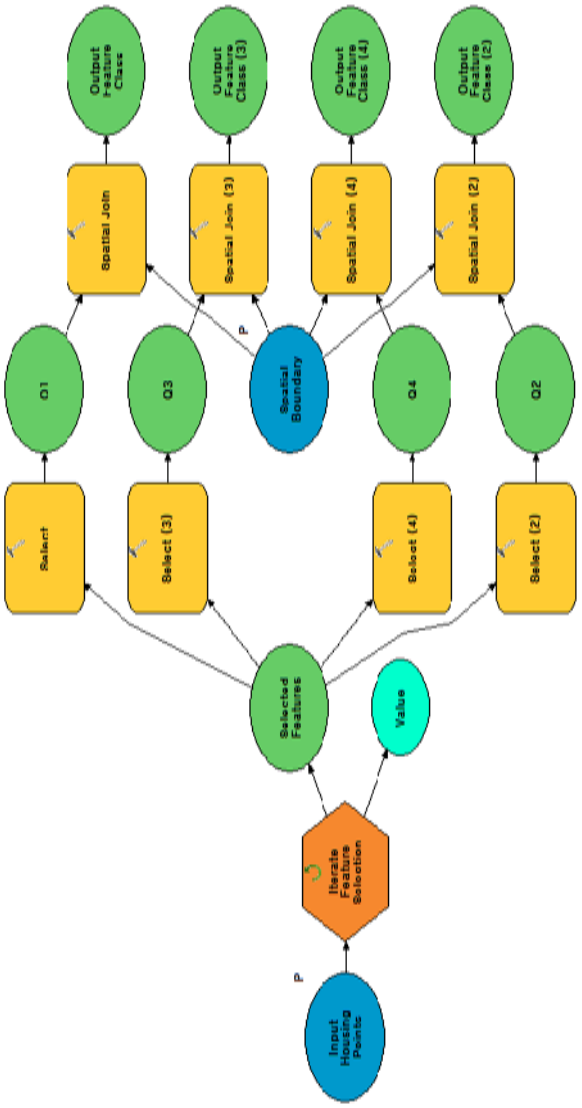

Figure 6: Spatial Join

Orange hexagon in this figure indicates iterator. For instance, to create first quarter's spatial joined polygon features, the iterator will run through each point and select the first quarter. Then, send the feature points to spatial join function and spatial join it with the census tracts.

3 Data Analysis

Figure 7 shows the second quarter in 2004 spatial patterns of housing prices. The red polygons indicate the areas have high housing prices and the blue areas specify the areas have relatively low

ISBN: 978-1-61804-150-0

housing prices. Figure 7 only shows the one of the results.

This shows that inland counties like Riverside and San Bernardino Angeles have relatively low housing prices than coast line areas like Santa Monica or Newport Beach.

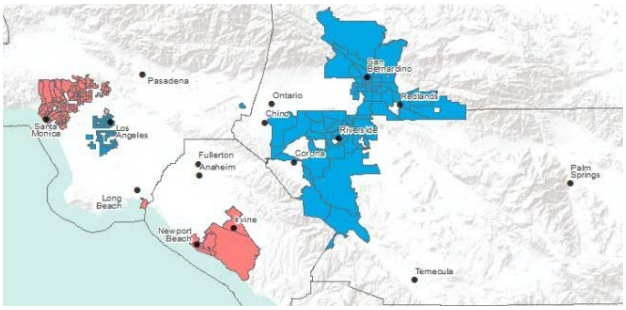

Figure 7: Local Moran's *I* Results in Q2, 2004

This spatial pattern, clusters of higher housing prices areas posed on the Pacific Ocean, did not change over the study time (2000-2009) even during the subprime mortgage crisis in late 2000. However, a few census tracts in Los Angeles had lower housing prices even if they were posed nearby coastal areas.

4 Conclusion

ArcGIS ModelBuilder allows exporting its workflows to Python programming language, which the user can start programming easily without typing all tiresome script. The ModelBuilder also allows users to modify ArcGIS embedded tools as the users' own wish.

Python script also let users to create their own ArcGIS tools. Python script has more flexibility than using implanted tools in ArcGIS or ModelBuilder. For instance, ModelBuilder only allows one iterator function. However, in Python script more than one iterator can be executed. Therefore, it can be very useful for dealing with big and complex data process even under the ArcGIS environment.

If one can manage programming and set correct algorithms of model workflow, no matter how much data he or she gets, all the works will be automated. The automated workflows would reduce labor time and mistakes that can be produced by human (GIS engineers). We have looked through just a small part of benefits of using Python and Modelbuilder to deal with big data processing.

Data are important for analyses or any kinds of researches. But more important thing is how we process the data to meet our needs. You can do much more with better accuracy by these introduced functions.

References:

[1] Ing. Zdena Dobesova, Programming language Python for data processing, *Electrical and Control Engineering*, 2011, pp. 4866 - 4869

[2] Colin Child, The Top Nine Reasons to Use a File Geodatabase, *ArcUser*, 2009, pp. 12 – 15

[3] Mircosoft Corporation, Excel specifications and limits, 2012, http://office.microsoft.com/en-us/excel-help/excel-specifications-and-limits-HP005199291.aspx

[4] ESRI and NAVTEQ, North American Address Locator, *ArcGIS Resource Center Desktop 10*, 2012. http://www.arcgis.com/home/item.html?id=919dd045918c42458f30d2c85d566d68

[5] ESRI, Inc., Spatial Join (Analysis), *ArcGIS Resrouce Center Desktop 10*, 2012, http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#//00080000000q000000

ISBN: 978-1-61804-150-0