Spatial Data Analysis with Python

Song Gao

Email: sgao@geog.ucsb.edu
Goals of Workshop

1. Introduction to the batch processing in ArcGIS;
2. Introduce the Python scripting language and its application in ArcGIS;
3. Become familiar with several methods for writing, and running geoprocessing scripts using Python;
4. Apply Python scripts to automate a GIS workflow;
5. Solve your own domain problem using Python.
I. Introduction

- Primary Data Types
  - **vector**: point, line, polygon
  - **raster**: continuous (e.g. elevation) or discrete surfaces (e.g. land use type)

Common Data Storage Formats

- **vector**: shapefile, geodatabase feature
- **tables**: (.dbf, .xlsx), KML, GeoJSON
- **raster**: ASCII, GeoTIFF, JPEG2000
Why Spatial?

- Discussion: What kinds of spatial variables can you think of for determining the house prices in cities?
Geographically Weighted Regression (GWR)

Discussion: What kinds of spatial variables can you think of for determining the house prices in cities?


\[ \beta_0 + \beta_1 \text{dist to school}^+ + \beta_2 \text{Income} = \text{house price} \]
Geographically Weighted Regression (GWR)

Global model

\[ y = \alpha + \beta X + \varepsilon \]

becomes

\[ y_i = \alpha_i + \beta_i X + \varepsilon_i \]

Where \( i \) indicates that there is a set of coefficients estimated for every observation in our data set.
Geographically Weighted Regression (GWR)
GWR using Python

- GeographicallyWeightedRegression Example (Python Window)
- The following Python Window script demonstrates how to use the GeographicallyWeightedRegression tool.

```python
import arcpy
arcpy.env.workspace = "c:/data"
arcpy.GeographicallyWeightedRegression_stats("CallData.shp", "Calls","BUS_COUNT;RENTROCC00;NoHSDip","CallsGWR.shp","ADAPTIVE","BANDWIDTH PARAMETER","#","25","#","CoefRasters","135","PredictionPoints","#","GWRCallPredictions.shp")
```
Geographically Weighted Regression (GWR)

Table 2. Nonstationarity of parameters in the GWR models.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Euclidean Distance</th>
<th>Travel Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>F value</td>
<td>p-value</td>
</tr>
<tr>
<td>Intercept</td>
<td>6.398</td>
<td>&lt;0.001 *</td>
</tr>
<tr>
<td>LnPlotRatio</td>
<td>1.140</td>
<td>0.178</td>
</tr>
<tr>
<td>LnGreenRatio</td>
<td>5.032</td>
<td>&lt;0.001 *</td>
</tr>
<tr>
<td>LnFloorArea</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LnPropertyFee</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LnEucD_{PriSchool}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LnEucD_{ShoppingMall}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Age</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* The statistically significant

Table 7
Out of sample predictive accuracy: percent of predicted prices within specified range of actual price and $R$-squared between actual and predicted price

<table>
<thead>
<tr>
<th></th>
<th>10%</th>
<th>20%</th>
<th>$R$-squared</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model 1: global</td>
<td>57.1</td>
<td>82.6</td>
<td>0.832</td>
</tr>
<tr>
<td>Model 2: expansion</td>
<td>59.3</td>
<td>85.2</td>
<td>0.867</td>
</tr>
<tr>
<td>Model 3: expansion with lag</td>
<td>59.3</td>
<td>86.7</td>
<td>0.882</td>
</tr>
<tr>
<td>Model 4: GWR</td>
<td><strong>64.6</strong></td>
<td><strong>88.3</strong></td>
<td><strong>0.878</strong></td>
</tr>
</tbody>
</table>
Types of Models in GIS (by function)

- Descriptive models – patterns
- Change models – before and after
- Impact models – what happens
- Explanatory models – process influence
- Predictive models – what will be like
Geoprocessing in GIS

- building data processing chains in GIS:

  *data -> operations -> output*
Geoprocessing in GIS

ModelBuilder

Python Window

Tools

Scripts

Search

Geoprocessing in GIS

ModelBuilder

Python Window

Tools

Scripts

Search
Geoprocessing in GIS

Workflow

- **Goal**: Map parks for citizens
- **Problem**: Park data needs to be organized
  - Park areas lack facility information
  - Activity data spread across multiple point layers
  - Data does not fit web application
  - Need to automate data update process

Geoprocessing in GIS

Population density

60% influence

This model finds the most suitable location for a new park. The model incorporates a Weighted Overlay tool, where weights are assigned to each input based on how much influence each should have in siting a new park.

Distance to parks

Potential park sites

40% influence
import arcpy from arcpy
import env from arcpy.sa
import *
env.workspace = "C:/sapyexamples/data"

outEucDistance = arcpy.EucDistance("rec_sites.shp", 5000, 5, "c:/sapyexamples/output/EucDirOut")
outEucDistance.save("C:/sapyexamples/output/eucdist")
The Need For GIS Automation

- Automation makes work easier. You don't have to remember which tools to use or the proper sequence in which they should be run.
- Automation makes work faster.
- Automation makes work more standardized.
Let us solve a problem!

- Find the population living within the 500 meter of HW 101 in Santa Barbara

**Population Data:**
https://www.census.gov/cgi-bin/geo/shapefiles/index.php

**Roads:**
http://geog.ucsb.edu/~sgao/data/santabarbara.zip
2. What is Python?

Python is a programming language that lets you work quickly and integrate systems more effectively. https://www.python.org/
1. Automate workflows;
2. Batch process data;
3. Manipulate data tables, geometry, and map docs;
4. Use functions accessible only by scripts.
Advantages of Python

1. Less restricted data types;
2. Open source support;
3. Cross-platform;
4. Object-orientated (A data structure that combines data with a set of methods for accessing and managing those data).
Python Editors

- Integrated Development Environment (IDE): A software application for programming and software development
- Source code editor: A text editor for software code, with features specially designed to simplify and speed up writing and editing of code
- Suggested Python editors:
  1) IDLE
  2) PythonWin
  3) IPython
  4) Others (wiki.python.org/moin/PythonEditors)
User Resources

1) Books

2) Websites

   www.python.org
   forums.arcgis.com

3) ArcPy site package (online)
Open Source Python Packages

WinPython for Python 2.7:
- numpy 1.9
- scipy 0.15
- PySAL: not included
- pandas 0.15
- shapely: not included
- fiona: not included
- six 1.8
- Windows only

Anaconda for Python 2.7:
- numpy 1.9
- scipy 0.14
- PySAL 1.6
- pandas 0.14
- shapely: not included
- fiona: not included
- six 1.8
- Virtual Machine images
- Windows, Mac, Linux

Enthought Canopy for Python 2.7:
- numpy 1.8
- scipy 0.15
- PySAL 1.7 (in academic option)
- pandas 0.15
- shapely: 1.5.1 (in academic option)
- fiona: 1.4.8 (in academic option)
- six 1.9
- Windows, Mac, Linux

Other Python distribution options listed at: [http://www.scipy.org/install.html](http://www.scipy.org/install.html)
Useful Open Source Python Spatial Libraries

**Data Handling:**
- shapely
- GDAL/OGR
- pyQGIS
- pyshp
- Pyproj

**Analysis:**
- shapely
- numpy, scipy
- pandas, GeoPandas
- PySAL
- Rasterio
- scikit-learn

**Plotting:**
- matplotlib, prettyplotlib
- descartes
- cartopy, it-image
3. Python Structure and Syntax

- Programs are composed of modules
- Modules contain statements
- Statements contain expressions
- Expressions create and process objects
3. Python Structure and Syntax

- **Object**: A piece of memory, with values and associated operations; also known as variables

- **Types of objects**:
  - Numbers
  - Strings
  - Lists
  - Dictionaries
  - Files
3. Python Structure and Syntax

- Expression: Processes an object: \( x \times 2 \)
- Statement: Performs a task, via an expression: \( y = x \times 2 \)
- Types of statements:
  - Assignment: \( x=5 \)
  - Call: `open('DataFile.csv')`
  - import
  - print
  - if/elif/else
  - for, while
3. Python Structure and Syntax

- **Module:** A library of tools; permanent file of code, composed of statements.

- **Types of modules:**
  - Standard library modules: os, sys, string … (module index)
  - Specialized modules or site-packages: arcpy (site package)
3. Python Structure and Syntax

- Case sensitivity (DataFile ≠ datafile)
- Indentation (4, 6, 8…)
- File paths (/, \ or r’string’)
- Quotation marks (“, ‘)
- Commenting (#)
4. Running A Python Script In ArcGIS

- Provides Python access to all geoprocessing tools and extensions in ArcGIS
  
a. All geoprocessing tools in ArcMap are provided as functions in ArcPy
  
b. ArcPy also includes several functions not available as tools in ArcMap
- ArcPy has several sub-modules with related sets of functions (e.g., spatial analyst, mapping)
4. Running A Python Script In ArcGIS

1) Include a header
2) Import modules
3) Specify environment settings
4) Define variables
5) Run geoprocessing tools
Three Ways to Run a Python Script In ArcGIS

1) In a Python editor

2) In the Python window in ArcMap

3) As a script tool in ArcToolbox
Three Ways to Write a Python Script In ArcGIS

1) Edit an existing script
2) Export a script from ModelBuilder
3) Build a script in the Python window
1) Python is case sensitive;
2) Python is sensitive to indentation
3) Filepaths use single forward slash (/), double back slash (\\), or raw string supression (r""filepath")
4) May need to hard code filepaths (workspace + os.dir + "filename")
5) Save scripts with the .py file extension
6) Avoid schema lock: remove datasets from ArcMap
Understanding your GIS
This "hello world" style notebook shows how to get started with the GIS and visualize its contents.
- Get started with the GIS class

Managing your GIS
The ArcGIS API for Python provides APIs and samples for ArcGIS Online administrators to manage their online organization.
- Clone a portal

Performing Spatial Analysis
Call sophisticated spatial analysis tools that work with online content, using a few lines of code.
- Chennai floods analysis

https://developers.arcgis.com/python/