Spatio-temporal analysis of socioeconomic neighborhoods

Sergio Rey[‡*], Elijah Knaap[‡], Su Han[‡], Levi Wolf[§], Wei Kang[‡]

Abstract—The neighborhood effects literature represents a wide span of the social sciences broadly concerned with the influence of spatial context on social processes. From the study of segregation dynamics, the relationships between the built environment and health outcomes, to the impact of concentrated poverty on social efficacy, neighborhoods are a central construct in empirical work. From a dynamic lens, neighborhoods experience changes not only in their socioeconomic composition, but also in spatial extent; however, the literature has ignored the latter source of change. In this paper, we discuss the development of novel, spatially explicit approaches to the study of longitudinal neighborhood dynamics using the scientific Python ecosystem.

Index Terms—neighborhoods, GIS, clustering, dynamics

Introduction (.5) (SR, EK, SH, LW, WK)

For social scientists in a wide variety of disciplines, neighborhoods are central thematic topics, focal units of analysis, and first-class objects of inquiry. Despite their centrality public health, sociology, geography, political science, economics, psychology, and urban planning, however, neighborhoods remain understudied because researchers lack appropriate analytical tools for understanding their evolution through time and space. Towards this goal we are developing the *open source longitudinal neighborhood analysis program* `(OSLNAP)`. We envisage OSLNAP as a toolkit for better, more open and reproducible science focused on neighborhoods and their sociospatial ecology. In this paper we first provide an overview of the main components of OSLNAP. Next, we present an illustration of selected OSLNAP functionality. We conclude the paper with a road map for future developments.

OSLNAP

Neighborhood analysis involves a multitude of analytic tasks, and different types of enquiry lead to different analytical pipelines in which distinct tasks are combined in sequence. OSLNAP is designed in a modular fashion to facilitate the composition of different pipelines for neighborhood analysis. Its functionality is available through different interfaces from a web-based front end as a service to a library for scripting in Jupyter notebooks or at the shell. We first provide an overview of each of the main analytical components of OSLNAP before moving on to an illustration of how selections of the analytical functionality can be combined for particular use cases.

OSLNAP's analytical components are organized into three core modules: [a] data layer; [b] neighborhood definition layer; [c] longitudinal analysis layer.

Data Layer 0.5

ST databases 0.25 (SH, EK, SR):

Harmonization 0.75 (EK, SR): Like many quantitative analyses, one of the most important and challenging aspects of longitudinal neighborhood analysis is the development of a tidy and accurate dataset. When studying the socioeconomic makeup of neighborhoods over time, this challenge is compounded by the fact that spatial units whose composition is under study often change size, shape, and configuration over time. The `harmonize` module provides social scientists with a set of simple and consistent tools for building transparent and reproducible spatiotemporal datasets. Further, the tools in `harmonize` allow researchers to investigate the implications of alternative decisions in the data processing pipeline and how those decisions affect the results of their research.

Neighborhood demographic and socioeconomic data relevant to social scientists are typically collected via a household census or survey and aggregated to a geographic reporting unit such as a state, county or zip code which may be relatively stable. The boundaries of smaller geographies like census tracts, however, often are designed to encapsulate roughly the same number of people for the sake of comparability, which means that they are necessarily redrawn with each data release as population grows and fluctuates. Since same physical location may fall within the boundary of different reporting units at different points in time, it is impossible to compare directly a single neighborhood with itself over time.

To facilitate temporal comparisons, research to date has proceeded by designating a "target" geographic unit or zone that is held constant over time, and allocating data from other zones using areal interpolation and other estimation techniques. This process is sometimes known as "boundary harmonization" [@logan_2016]_. While "harmonized" data

* *Corresponding author: sergio.rey@ucr.edu*
‡ *Center for Geospatial Sciences, University of California, Riverside*
§ *School of Geographical Sciences, University of Bristol*

is used widely in neighborhood research, the harmonization process also has known shortcomings, since the areal interpolation of aggregate data is subject to the ecological fallacy–the geographic manifestation of which is known as the “Modifiable Areal Unit Problem” (MAUP) [@Openshaw1984]. Simply put, MAUP holds that areal interpolation introduces bias since the spatial distribution of variables in each of the overlapping zones is unknown. A number of alternative approaches have been suggested to reduce the amount of error by incorporating auxiliary data such as road networks, which help to uncover the “true” spatial distribution of underlying variables, but this remains an active area of research [@schroeder_2017; @Sridharan2013; @Tapp2010; @Xie1995].

In practice, these challenges mean that exceedingly few neighborhood researchers undertake harmonization routines in their own research, and those performing temporal analyses typically use exogenous, pre-harmonized boundaries from a commercial source such as the Neighborhood Change Database (NCDB) [@tatian], or the freely available Longitudinal Tract Database (LTDB) [@logan_2014]. The developers of creators of these products have published studies verifying the accuracy of their respective data, but those claims have gone untested because researchers are unable to fully replicate the underlying methodology.

To overcome the issues outlined above, `oslnap` provides a suite of functionality for conducting areal interpolation and boundary harmonization in the `harmonize` module. It leverages `geopandas` and `PySAL` for managing data and performing geospatial operations, and the pydata stack for attribute calculations [@rey2009]. The `harmonize` module allows a researcher to specify a set of input data (drawn from the space-time database described in the prior section), a set of target geographic units to remain constant over time, and an interpolation function that may be applied to each variable in the dataset independently. For instance, a researcher may decide to use different interpolation methods for housing prices than for the share of unemployed residents, than for total population; not only because the researcher may wish to treat rates and counts separately, but also because different auxiliary information might be applicable for different types of variables.

In a prototypical workflow, `harmonize` permits the end-user to:

- query the spatiotemporal database created via the `data` module
  - queries may have spatial, temporal, and attribute filters
- define the relevant variables to be harmonized and optionally apply a different interpolation function to each
- harmonize temporal data to consistent spatial units by either:
  - selecting an existing native unit (e.g. zip codes in 2016)
  - inputting a user-defined unit (e.g. a theoretical or newly proposed boundary)
  - developing new primitive units (e.g. the intersection of all polygons)

Neighborhood Identification 1.5
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

cluster 1.0 (EK, LW, SR) (1): Neighborhoods are complex social and spatial environments with multiple interacting individuals, markets, and processes. Despite 100 years of research it remains difficult to quantify neighborhood context, and certainly no single variable is capable of capturing the entirety of a neighborhood’s essential nuance. For this reason, several traditions of urban research focus on the application of multivariate clustering algorithms to develop neighborhood typologies. Such typologies are sometimes viewed as more holistic descriptions of neighborhoods because they account simultaneously for multiple characteristics simultaneously [@galster2001].

One notable tradition from this perspective called “geodemographics”, is used to derive prototypical neighborhoods whose residents are similar along a variety of socioeconomic and demographic attributes [@flowerdew1989; @singleton2014]. Geodemographics have been applied widely in marketing [@farr2005], education [@singleton2009a], and health research [@petersen2011] among a wide variety of additional fields. The geodemographic approach as also been criticized, however, for failing to model geographic space formally. In other words, the geodemographic approach ignores spatial autocorrelation, or the “first law of geography”–that the attributes of neighboring zones are likely to be similar. Another tradition in urban research, known as “regionalization” has thus been focused on the development of multivariate clustering algorithms that account for spatial dependence explicitly. To date, however, these traditions have rarely crossed in the literature, limiting the utility each approach might have toward applications in new fields. In the `cluster` module, we implement both clustering approaches to (a) foster greater collaboration among weakly connected components in the field of geographic information science, and (b) to allow neighborhood researchers to investigate the performance of multiple different clustering solutions in their work, and evaluate the implications of including space as a formal component in their clustering models.

the `cluster` module leverages the scientific python ecosystem, building from `‘geopandas` <http://geopandas.org/>‘__, `‘PySAL` <http://pysal.org>‘__, and `‘scikit-learn` <http://scikit-learn.org>‘__. Using input from the Data Layer, the `cluster` module allows researchers to develop neighborhood typologies based on either attribute similarity (the geodemographic approach) or attribute similarity with incorporated spatial dependence (the regionalization approach). Given a space-time dataset, the `cluster` module allows users to cluster (a) a single time period, (b) a set of time periods cross-sectionally, or (c) a set of periods pooled as a time series. In (b), neighborhood clusters are independent from one time period to the next. This can be a useful approach if researchers are interested in the durability and permanence of certain kinds of neighborhoods. If similar types reappear in multiple cross sections (e.g. if the k-means algorithm places the k-centers in approximately similar locations each time period), then it may be inferred that the metropolitan dynamics are somewhat stable, at least at the macro level, since new kinds of neighborhoods do not appear to be evolving and
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

old, established neighborhood types remain prominent. The drawback of this approach is the type of a single neighborhood cannot be compared between two different time periods because the types are independent in each period.

In the (c), clusters are defined from all observations in all time periods. In this case, the universe of potential neighborhood types is held constant over time, the neighborhood types are consistent across time periods, and researchers can examine how particular neighborhoods get classified into different neighborhood types as their composition transitions through different time periods. While comparatively rare in the research, this latter approach allows a richer examination of socio-spatial dynamics. By providing tools to drastically simplify the data manipulation and analysis pipeline, we aim to facilitate greater exploration of urban dynamics that will help catalyze more of this research.

To facilitate this work, the `cluster` module provides wrappers for several common clustering algorithms from `scikit-learn` that can be applied . Beyond these, however, it also provides wrappers for several *spatial* clustering algorithms from `PySAL`, in addition to a number of state-of-the art algorithms that have recently been developed [@wolf2018] .

In a prototypical workflow, `cluster` permits the end-user to:

- query the (tidy) space-time dataset created via the `harmonize` module
  - queries may have spatial, temporal, and attribute filters
- define the neighborhood attributes and time periods and on which to develop a typology
- run one or more clustring algorithms on the space-time dataset to derive neighborhood cluster membership
  - clustering may be applied cross-sectionally or on the pooled time-series
  - clustering may incorporate spatial dependence, in which case `cluster` provides options for users to parameterize a spatial contiguity matrix
- clustering results may be reviewed quickly via the builtin `plot()` method, or interactively by leveraging the `geovisualization` module.

Longitudinal Analysis (WK, SR, EK) (.5): The second major component of the analytical layer provides a suite of functionalities for the longitudinal analysis of neighborhoods to uncover how neighborhoods evolve over time. Traditional analysis focuses solely on the changes in the socioeconomic composition, while it is argued that the geographic footprint should not be ignored [**?**]. Therefore, this component draws upon recent methodological developments from spatial inequality dynamics and implements two broad sets of spatially explicit analytics to provide deeper insights into the evolution of socioeconomic processes and the interaction between these processes and geographic structure.

Both sets of analytics take time series of neighborhood types assigned for all the spatial units of analysis (e.g. census tracts) based on adopting a spatial clustering algorithm as the input while they differ in how the time series are modeled and analyzed. The first set centers on *transition analysis* which treats each time series as stochastically generated from time point to time point. It is in the same spirit of the first-order Markov Chain analysis where a (k, k) transition matrix is formed by counting transitions across all the k neighborhood types between any two consecutive time points for all spatial units. Drawbacks of such approach include that it treats all the time series as being independent of one another and following an identical transition mechanism. The spatial Markov approach was proposed by [@Rey01]_ to interrogate potential spatial interactions by conditioning transition matrices on neighboring context while the spatial regime Markov approach allows several transition matrices to be formed for different spatial regimes which are constituted by contiguous spatial units. Both approaches together with inferences have been implemented in Python Spatial Analysis Library (PySAL)[1] [@Rey14]_ and the Geospatial Distribution Dynamics (giddy) package[2]. Our module considers these packages as dependencies and wrap relevant classes/functions to make them consistent and efficient to the longitudinal neighborhood analysis.

The other set of spatially explicit approach to neighborhood dynamics is concerned with *sequence analysis* which treats each time series of neighborhood types as a whole in contrast to *transition analysis*. The optimal matching (OM) algorithm, which was originally used for matching protein and DNA sequences [**?**], is adopted to measure the similarity between every pair of neighborhood type time series. It generally works by finding the minimum cost for transforming one time series to another using a combination of operations including replacement, insertion and deletion. The similarity matrix is then used as the input for another round of clustering to derive a typology of neighborhood trajectory [**?**]. We extend the definition of various operation costs to incorporate potential spatial dependence and spatial heterogeneity.

Geovisualization Layer 1.0

View Types 0.75 (SH, SR):
Interactivity 0.75 (SH, SR):

Illustration (4.5-7.5)

Neighborhood Identification 1.5 (EK, LW, SH, SR)

Neighborhood Dynamics 1 (WK, SR, EK, SH, LW)

Conclusion (0.5)

Future Directions (SR, EK, SH, LW, WK)

In this paper we have presented the motivation for, and initial design and implementation of OSLNAP. At present, we are in the early phases of the project and moving we will be focusing on the following directions.

Parameter sweeps: In the definition of neighborhoods, a researcher faces a daunting number of decisions surrounding treatment of harmonization, selection of variables, and choice of clustering algorithm, among others. In the neighborhood literature, the implications of these decisions

1. https://github.com/pysal/pysal
2. https://github.com/pysal/giddy

remain unexplored and this is due to the computational burdens that have precluded formal examination. We plan on a modular design for OSLNAP that would support extensive parameter sweeps to provide an empirical basis for exploring these issues and to offer applied researchers computationally informed guidance on these decisions.

Data services: OSLNAP is being designed to work with existing harmonized data sets available from various firms and research labs. Because these fall under restrictive licenses, users must first acquire these sources - they cannot be distributed with OLSNAP. To address the limitations associated with this strategy, we are exploring interfaces to public data services such as CenPy and tigris so that users

Reproducibility: A final direction for future research is the development of reproducible workflows as part of OSLNAP. Here we envisage leveraging our earlier work on provenance for spatial anayltical workflows [**?**] and extending it to the full longitudinal neighborhood analysis pipeline.