A Data-driven, High-performance and Intelligent CyberInfrastructure to Advance Spatial Sciences

by

Hu Shao

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved September 2018 by the
Graduate Supervisory Committee:

Wenwen Li, Co-Chair
Sergio Rey, Co-Chair
Ross Maciejewski

ARIZONA STATE UNIVERSITY
December 2018

ABSTRACT

In the field of Geographic Information Science (GIScience), we have witnessed the unprecedented data deluge brought about by the rapid advancement of high-resolution data observing technologies. For example, with the advancement of Earth Observation (EO) technologies, a massive amount of EO data including remote sensing data and other sensor observation data about earthquake, climate, ocean, hydrology, volcano, glacier, etc., are being collected on a daily basis by a wide range of organizations. In addition to the observation data, human-generated data including microblogs, photos, consumption records, evaluations, unstructured webpages and other Volunteered Geographical Information (VGI) are incessantly generated and shared on the Internet.

Meanwhile, the emerging cyberinfrastructure rapidly increases our capacity for handling such massive data with regard to data collection and management, data integration and interoperability, data transmission and visualization, high-performance computing, etc. Cyberinfrastructure (CI) consists of computing systems, data storage systems, advanced instruments and data repositories, visualization environments, and people, all linked together by software and high-performance networks to improve research productivity and enable breakthroughs that are not otherwise possible.

The Geospatial CI (GCI, or CyberGIS), as the synthesis of CI and GIScience has inherent advantages in enabling computationally intensive spatial analysis and modeling (SAM) and collaborative geospatial problem solving and decision making.

This dissertation is dedicated to addressing several critical issues and improving the performance of existing methodologies and systems in the field of CyberGIS. My dissertation will include three parts: The first part is focused on developing methodologies to help public researchers find appropriate open geo-spatial datasets from millions of records provided by thousands of organizations scattered around the

world efficiently and effectively. Machine learning and semantic search methods will be utilized in this research. The second part develops an interoperable and replicable geoprocessing service by synthesizing the high-performance computing (HPC) environment, the core spatial statistic/analysis algorithms from the widely adopted open source python package – Python Spatial Analysis Library (PySAL), and rich datasets acquired from the first research. The third part is dedicated to studying optimization strategies for feature data transmission and visualization. This study is intended for solving the performance issue in large feature data transmission through the Internet and visualization on the client (browser) side.

Taken together, the three parts constitute an endeavor towards the methodological improvement and implementation practice of the data-driven, high-performance and intelligent CI to advance spatial sciences.

To

My wife Wei Kang

And

My mother Fengxia Jia

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my advisors, Prof. Wenwen Li and Prof. Sergio Rey, for their enlightenment and support throughout my PhD study. It is a great fortune to have such knowledgeable advisors and it was a great time while working with them in last four years. My gratitude also goes to my committee member, Prof. Ross Maciejewski, for his constructive advice on the dissertation.

Table of Contents

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

1 INTRODUCTION

1.1 Background and Research Motivation

Geographic Information Science (GIScience) and System (GISystem) have been booming in recent decades and achieved great development. On one hand, they have borrowed a lot of theories, concepts and approaches from many other disciplines, including Mathematics, Physics, Computer Science, Economics, Psychology etc. On the other hand, GIScience as a practical science continuously plays a critical role in numerous fields such as climate change, ecology, environmental sciences, public health and archaeology to help solve scientific problems and improve decision-making practices with significant societal impacts (Wang 2013). In the foreseeable future, such interaction between GIScience and other disciplines will be afoot.

In the field of GIScience, we have witnessed the unprecedented data deluge resulting from the rapid advancement of high-resolution data observing technologies (Kitchin, 2013; Li, Hodgson, & Li, 2018). For example, with the advancement of Earth Observation (EO) technologies, a massive amount of EO data including remote sensing data and other sensor observation data on earthquake, climate, ocean, hydrology, volcano, glacier, etc. are being collected on a daily basis by a wide range of organizations. Besides, human-generated data including microblogs, photos, consumption records, evaluations, unstructured web pages and many other Volunteered Geographical Information (VGI; Goodchild, 2007) are incessantly generated and shared on the Internet (Yang, Huang et al., 2017).

Meanwhile, the emerging cyberinfrastructure rapidly increases our capacity for handling such massive data with regard to data collection, management, high-performance computing, data integration and interoperability, data transmission and visualization, etc. (Zhang and Tsou 2009; Yang et al. 2010; Wright and Wang 2011; Rey et al. 2015; Li,

Cao, and Church 2016a; Li, Wang, Bhatia 2016b; Li et al. 2016c; Song et al. 2016). Cyberinfrastructure consists of computing systems, data storage systems, advanced instruments and data repositories, visualization environments, and people, all linked together by software and high-performance networks to improve research productivity and enable breakthroughs that are not otherwise possible (Stewart et al, 2010, Wang et al, 2013).

The Geospatial CyberInfrastructure (GCI, or CyberGIS), as the combination of Cyberinfrastructure and GIScience has inherent advantages in dealing with complicated tasks like enabling the analysis of big spatial data, computationally intensive spatial analysis and modeling (SAM), collaborative geospatial problem-solving and decision-making, simultaneously conducted by a large number of users. According to Yang et al, (2010), the main functions of CyberGIS could include: 1) Multi-dimensional data processing, 2) Data collection and heterogeneous integration, 3) Data preservation and accessibility, 4) Supporting the life cycle from data to knowledge, 5) Virtual Organizations (VO), 6) Semantic Web and knowledge sharing, 7) High-performance computing (HPC) and associated spatial computing, 8) Location-based service, and 9) Cross-scale and domain management. Figure 1 demonstrates a generic framework of CyberGIS. From this figure, we can see how numerous components couple with each other and form the lifecycle of a CyberGIS system.

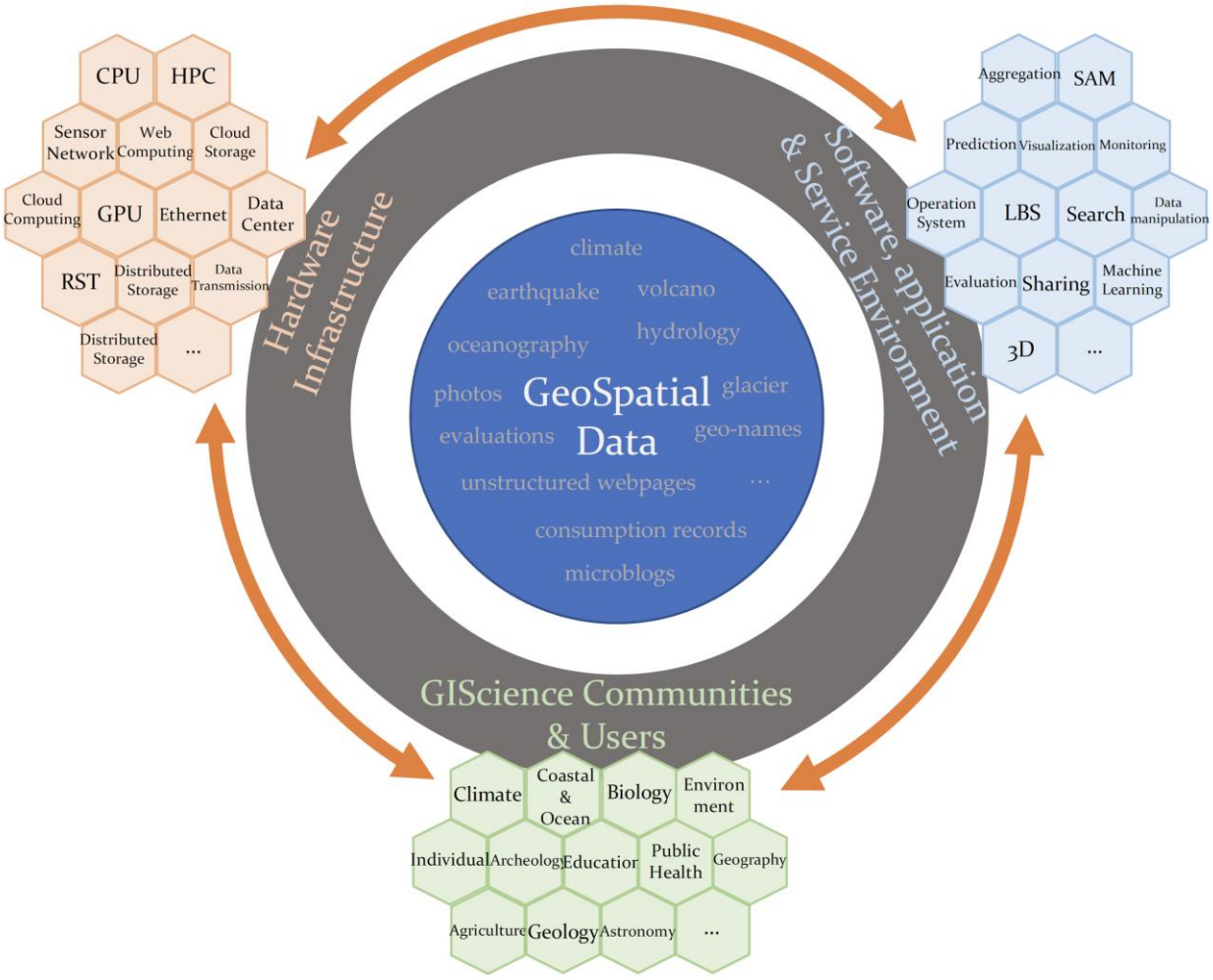


Figure 1 A generic framework for Geospatial CyberInfrastructure. (RST: rapid storage technology; SAM: spatial analysis models; LBS: location-based service)

The advancement of technologies and economy makes it easier for scientists to assemble tremendous resources, workforce, funding, and equipment together to conquer complex and difficult research topics and projects through collaborative working mode. This is also true in the GIScience field. CyberGIS has the potential of providing significant contributions to such scenarios due to its capability of bridging all kinds of distributed resources and providing seamlessly integrated user interface to leverage the collaboration among different teams and disciplines. Such great potential and opportunities have attracted numbers of organizations, teams, and individuals to dedicate to the field of CyberGIS (Anselin & Rey, 2012; Huang et al., 2013; Li, Cao, &

Church, 2016; Wang, 2010; Wang et al., 2013; Yang, Huang, Li, Liu, & Hu, 2017; Yang, Raskin, Goodchild, & Gahegan, 2010; Yu, Yang, & Li, 2018).

This dissertation introduces my systematic research works related to CyberGIS during my Ph.D. period. Three specific research topics are identified and studied: 1) open geospatial data discovery; 2) geospatial and spatial-temporal analysis service integration; 3) high-performance spatial data transmission and visual analytics.

Although massive geospatial data sets are collected and shared on the Internet, most of them are widely distributed on different data repositories hosted by various organizations. Not only the User Interface (UI) provided by those repositories are quite diverse, but also the data sets hosted on those repositories vary a lot in format, time representation, accuracy, coverage, attribute, projection etc. The 20/80 theorem also fits in this situation: compared to the time been spent on data analysis (20% of all), environmental scientists are spending much more time (80%) in finding appropriate data and organizing them (Li et al, 2010). Data integration is the basic ability of CyberGIS to build the bridge between data providers and end users (Horsburgh et al., 2009). Facing such a situation of data deluge, the plight remains on how to help users conveniently and efficiently find appropriate datasets.

Numbers of vibrant communities are working on introducing/integrating the most recent and advanced research algorithms/results into open source software and libraries, such as Python Spatial Analysis Library (PySAL) (Luc Anselin & Rey, 2014; Sergio J. Rey, 2014; S. J. Rey & Anselin, 2007), GeoDa (L. Anselin, Syabri, & Kho, 2010), GDAL, GRASS GIS, GeoTools, GeoPython, spaceime (Pebesma, 2012), STARS (Sergio J. Rey & Janikas, 2006), spdep (Bivand et al., 2011) etc. These toolkits play a critical role in promoting the innovation in GIScience. Meanwhile, more and more big geospatial data sets and HPC resources are becoming available with the advancement of theory and technology. Coupling the spatial analytical functionalities with big data and HPC

resources could bring immediate benefits to multi-disciplines in helping solve complex spatial analysis tasks, supporting remote collaboration among participants from distributed groups, and assisting decision making (Shaowen Wang, 2013). However, most of the open source libraries and toolkits as aforementioned are initialized and designed mainly for the desktop working environment. Hence, how to bridge such advanced spatial analysis functionalities form open source libraries with HPC resources to provide researchers with interoperable and replicable geoprocessing APIs remains to be a great challenge. On the other hand, since GIScience has been widely applied in other research disciplines where empirical researchers do not necessarily have enough GIS background knowledge, the steep learning curve for the advanced algorithm and models will hinder their wide adoption. Therefore, during the implementation of a CyberGIS framework, challenges remains to be addressed on how to provide user-friendly graphic user interface (GUI) with abundant instruction and documentation in order to help users better understand and take advantage of such toolkits, then move a step further to foster the collaboration across the Internet.

In the CyberGIS enabled web services, the ability of rapidly transmitting and sharing spatial data over the Internet is critical to meet the demands of real-time change detection, response and decision making. Many data sets are recorded in the form of vector with attributes (point, line, polygon), such as census tract, hydrology dataset, road network, sensor observation data. In many real-world data-driven applications, original vector datasets are essential for developing flexible, expressive and interactive data visualization and analysis functionalities to help users better understand the context of events and make decisions (Zhang and Li 2005; Stollberg and Zipf 2012). For example, in the scenario of disaster management, i.e. earthquake or flood, researchers need to retrieve multiple datasets including Digital Elevation Model (DEM), road networks, hydrology flow, population distribution, real-time observation data, etc. from distributed

Spatial Data Infrastructures (SDIs) and then conduct analysis immediately for developing evacuation and rescue plans. However, the vector dataset could be very large. The large data volume will slow down each data processing step including data encoding, transmitting, analyzing and visualizing, which could result in a failure to meet the time-critical requirements in real word practices. Hence, developing an optimized processing/transmission module to handle spatial data with massive volume within the framework of CyberGIS could be of great importance to the GIScience field.

1.2 Significance and Contributions

This dissertation is comprised of three potentially publishable papers, each focusing on solving aforementioned specific issues related to CyberGIS.

The building blocks of the first research are thousands of data repositories harvested from the Internet, which result from the pioneer studies of Li et al (Li, 2017; Li, Wang, & Bhatia, 2016; Li, Yang, & Yang, 2010). Based on the previous work, more than 70K datasets distributed in ninety-five countries have been found, which host more than millions of data layers mainly published through Open Geospatial Consortium's (OGC) Web Map Service (WMS; de La Beaujardiere 2006) and Web Feature Service (WFS; Vretanos 2004). Each of the datasets has corresponding metadata which describes its content, topic, provider and other aspects of attributes. This chapter introduces my work on developing a synthetic system that exploits the state-of-art semantic search technologies and supplementary approaches for accomplishing the open access geospatial datasets discovery tasks. To be more specific, 1) a metadata enrichment method is introduced to retrieve more information about the datasets from their original website, 2) the phrase embedding method of natural language processing is adopted to automatically catch the semantic relationship among words and phrases, 3) a working

cyberinfrastructure portal that implements the methodologies is established for providing data search functionalities to public users.

The second research is dedicated to developing an interoperable and replicable cyberinfrastructure for online spatial-statistical-visual analytics. More specifically, I focus on the widely used open source python library - Python Spatial Analysis Library (PySAL), the functions/classes of which are published as geoprocessing services - WebPySAL. Meanwhile, a friendly GUI is implemented in a CyberGIS portal named Geospatial CyberInfrastructure (GeoCI). The client side is capable of integrating any open geospatial data shared based on OGC's WFS/WMS standards, and invoking the geoprocessing services from WebPySAL for on-the-fly spatial analysis, which endows great flexibility to users. During the system design and implementation, four challenges list below are addressed:

- **Interoperability between components and services**: the deployed toolsets should be compatible with the mainstream software and other services, and meanwhile could be easily exploited by users under the network environment.
- **Provenance and metadata for spatial analytical workflows**: this could be one of the most critical factors under the "collaboration" working mode, referring to all the information ranging from how the spatial data is produced, to how the geoprocessing steps are chained and conducted, and to how to obtain the results - the key for quality control and reproduction of geospatial analysis (Luc Anselin & Rey, 2012).
- **Granularity of the functionalities to be exposed as Application Programming Interfaces (APIs)**: many open source libraries are designed for the "single-user" working mode, in which the functionalities of each method and class are usually designed to be atomic, facilitating users to combine various methods for the exploratory analysis in a flexible manner. However, when

deploying the functions on the server side, the communication cost between the client side and the server side needs to be taken account of. The most intuitive way to reduce the communication cost is to combine the atomic APIs into non-atomic ones which accomplish a sophisticated operation by accepting several parameter inputs from users at one shot (e.g. the inference about Local Indicators of Spatial Association (LISAs) (Luc Anselin, 1995)).

- **Documentations and supporting materials**: many open source projects serve as a pioneer in implementing and introducing newly developed methodologies of spatial analysis. When deploying these methodologies, how to provide adequate documentation and materials to educate users to appropriately use the APIs, should be carefully considered as well.

In the third chapter, I introduce the design and implementation of a comprehensive optimizing strategy for high-efficiency vector data sharing through OGC's WFS standards. In general, a WFS processing involves the following workflow: when a web server receives a WFS request, it will first parse the request. Then, according to the parameters provided by the client, the WFS server accesses the required data source and conducts data processing. For example, a spatial filter operation will be applied to the raw data to derive a subset within the desired bounding box. After these processing steps, resultant features will be encoded into specific output format before being sent back to the client side. When the client side receives the response stream, it will decode the stream, parse the result, and convert it into a feature collection which could be used for visualization, statistics, and analysis. The strategy for improving WFS data transmission consists of 1. Combination of pre-generalization and real-time generalization for multiple layers; 2. Separated data transmission processes of features' geometries and attributes; 3. Dynamic adoption of data compression/ decompression methods according to the

network status. Significant improvements will be achieved by applying this optimization strategy to conventional WFS approaches.

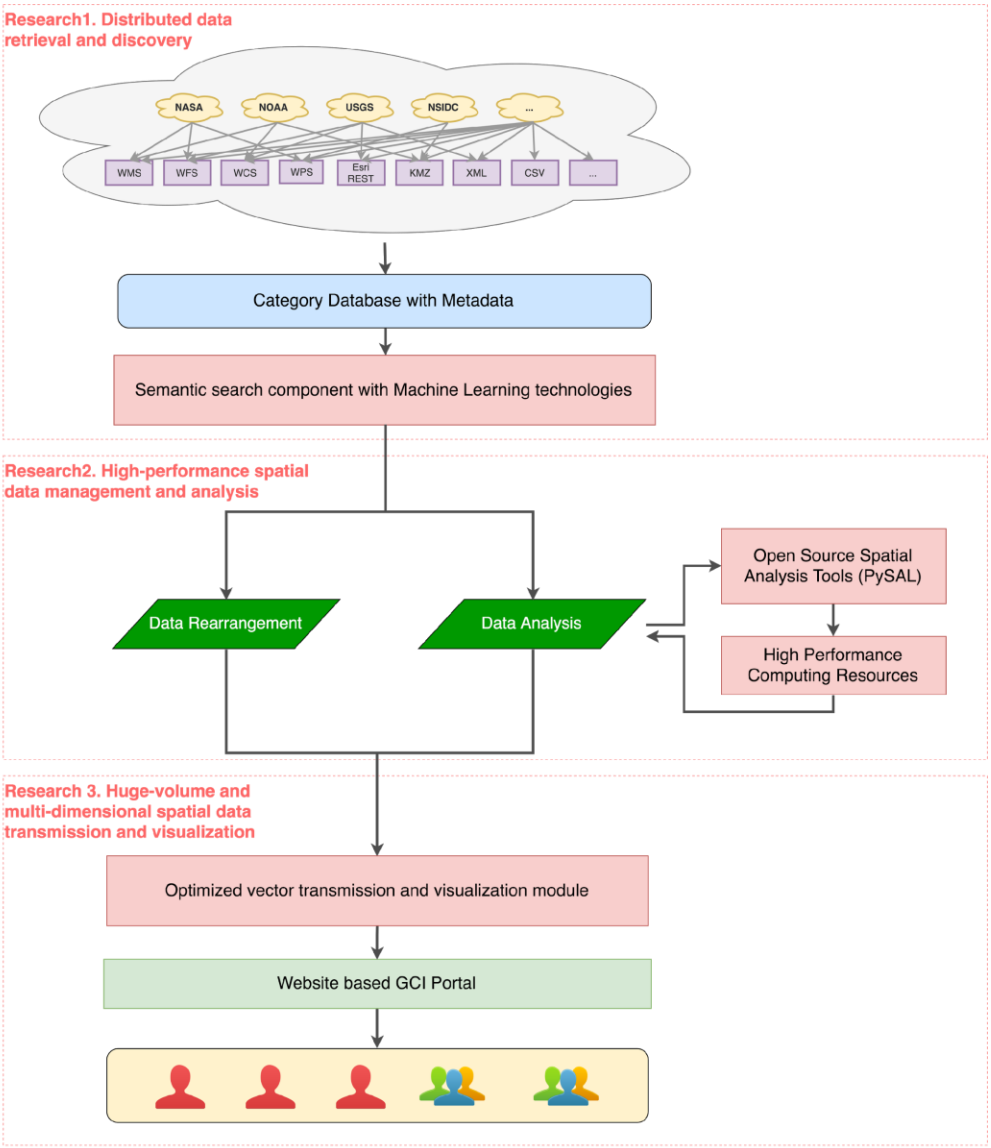


Figure 2 Generic dissertation research framework

Taken together, the three chapters constitute an endeavor towards methodological improvements and implementing practice for the data-driven, high-performance and intelligent CyberInfrastructure to advance spatial sciences. A synthetical and solid working CyberInfrastructure platform named GeoCI will be established as the deliverable outcome, which integrates basic GIS functionalities such as data

management, manipulation, and visualization, as well as all the advanced functionalities achieved in these research works. The relationship of these components and how they interact with each other are illustrated in Figure 2. Hopefully bridging these components together in GeoCI platform could gain the consequence of "1+1>2" in helping public researchers and users efficiently and conveniently discover open geospatial data, conducting exploratory spatial data analysis, and fostering collaboration across different disciplines.

1.3 Organization of the dissertation

The rest of the dissertation is comprised of four chapters. Chapter 2 presents the paper focusing on developing the methodologies to build a synthetic system that enables semantic search for open geospatial datasets. Chapter 3 is the paper on developing an interoperable and replicable cyberinfrastructure for online spatial-statistical and visual-analytics. Chapter 4 presents the paper on designing and implementing a comprehensive optimization strategy for real-time spatial feature sharing and visual analytics under the cyberinfrastructure environment. Chapter 5 introduces the architecture of the comprehensive CyberGIS system GeoCI, as well as how those individual components are integrated into the system and enhance each other in help users solving complex spatial analysis problems. Chapter 6 concludes with the main findings, limitations and potential research directions in future.

2 A SYNTHETIC SYSTEM THAT ENABLES SEMANTIC SEARCH FOR OPEN GEOSPATIAL DATASETS

2.1 Introduction

With the advancement of Earth Observation (EO) technologies, a massive amount of EO data covering the spectrum from remote sensing data to other sensor observation data about earthquake, climate, ocean, hydrology, volcano, glacier etc., are being collected and shared through the Internet on a daily basis by a wide range of organizations. These data play a critical role in the GIScience field in helping scholars gain comprehensive insights into the natural and social phenomena.

However, these rapidly expanding data sources and subsequent processing results are mainly disconnected from each other due to the fact that the organizations which gather and process them are physically distributed around the world (Li *et al.*, 2011). This introduces a great gap between the distributed data sources and users, brings inconvenience to users for searching, retrieving, and mining the massive datasets efficiently before interesting and significant research questions can be raised and answered (Ye, Li & Huang, 2018).

For the distributed geospatial data sources, there are basically two different approaches for archiving, managing and providing them to end users. The **first one** is to build and maintain a synthetical gateway that aggregates as many available data as possible (Li, Goodchild and Raskin, 2012). A number of well-known organizations and agencies are dedicated to gathering and providing high-quality geospatial datasets to public users and professional researchers, including Global Earth Observation System of Systems (GEOSS; Christian, 2005), the INSPIRE geoportal of Europe (Bernard et al., 2005), National Snow & Ice Data Center (NSIDC), the Geospatial Platform of U.S. Federal Geospatial

Data Committee, National Oceanic and Atmospheric Administration (NOAA), the Oak Ridge National Laboratory Distributed Active Archive Center (ORNL DAAC) etc. This approach requires tremendous resources of time, labor, and funding as the long-term input, which might be feasible only with the support from government. Besides, the agreement and collaboration among participants are indispensable. However, the benefits are also obvious - the quality and quantity of datasets, as well as standards used for data maintenance and publishment could all be guaranteed. The **second one** is to develop active web crawler (like Google) to gather datasets provided by various repositories that exist on the Web and provide them through a uniformly designed portal/UI (Li, Yanga and Yang, 2010; Lopez-Pellicer *et al.*, 2011; Patil, Bhattacharjee and Ghosh, 2014; Li, 2017). Li et al. (2017) developed a large-scale web crawling architecture called PolarHub to discover distributed geospatial data and service resources. PolarHub is built upon a service-oriented architecture (SOA) and adopts Data Access Object (DAO)-based software design to ensure the extendibility of the software system. According to the authors, metadata of 40,000 OGC services with 1.5million unique data layers are collected and hosted on their system. The second approach requires sophisticated methods and algorithms to be developed for data crawling and metadata fusion, harmonize and management, and the data quality and consistency are very hard to control. But this approach saves users' time on browsing and searching data across the Internet, improving the accessibility of geospatial data.

Once the huge amount of data is gathered, the following critical task is to provide efficient and friendly search functionalities to help users quickly locate the datasets they desire from hundreds of thousands of records. Both the quality of metadata and the capabilities of searching functionalities could affect the performance of such data searching task (Hu, K. Janowicz, Prasad, Gao, *et al.*, 2015). The metadata is used to describe various aspects of each geospatial dataset, such as its topic, content, extent,

precision, provider, when and how the data is produced, etc. Detailed and accurate metadata is essential for building an effective and efficient data discovery portal. On the other hand, the choice of a data search algorithm and method also matters a lot. The conventional way of data search is based on the full-text keyword-matching technique: only the datasets whose metadata contains identical keywords provided by users will be selected as preliminary candidates, while other datasets which are actually relevant but are described with different keywords will be excluded. The information retrieval community has dedicated a lot of efforts to adopting machine learning and semantic search methods to build the linkages among different keywords and metadata records in order to improve the precision and recall rate of the search results. The methodologies include LSA, LDA et al.

In this chapter I propose to introduce the phrase embedding method for automatically capturing the semantic relationship among various words and phrases in a large number of datasets. The phrase embedding method is based on the recent emerging Word2Vec model for natural language processing (NLP). Word2Vec represents words as vectors in the vector space, while phrases can be represented as the composition of word vectors using compositional models in phrase embedding methods. Then the semantic similarity between words and phrases can be measured.

This chapter develops a synthetic system that enables the state-of-art semantic search technologies with the metadata enrichment approach for accomplishing the open access geospatial datasets discovery task. To be more specific, 1) a metadata enrichment strategy is introduced to retrieve more information about the datasets form their original website, 2) the phrase embedding method is adopted to automatically catch the semantic relationship among words and phrases, and 3) a cyberinfrastructure portal that implements the methodology is established and providing data search functionalities for public users. The rest of this chapter is organized as follows. Section 2.2 introduces

related research in this field, and background knowledge in the information retrieval field. Section 2.3 introduces the phrase embedding methodology. Section 2.4 introduces experiments and the architecture of the cyberinfrastructure system that integrates the data discovery engine. We conclude our work with future directions in Section 2.5.

2.2 Related work

In the scenario of spatial data query on a CyberGIS gateway, when the user inputs the query keyword, the most straightforward way for searching the related spatial data records is to check the database using the full-text keyword-matching technique which finds those datasets whose metadata includes the identical keyword. Such technique has been implemented in the search library such as Apache Lucene and Elastic search, and has been widely adopted in many of the existing geospatial catalogs and portals (McCandless, Hatcher and Gospodnetic, 2010). Its main disadvantage is that during the searching process, the datasets related to the keyword but depicted with synonyms will be excluded from the result candidates. For example, if the searching keyword is “sea”, datasets whose description contains keyword “ocean” or “offshore” may be excluded (Li, Goodchild and Raskin, 2014).

Two factors can be used to measure the performance of a data query system: *precision* and *recall*, which are illustrated in Figure 3. Suppose the blue circle ($A \cup C$) represents the set of true records in the database that are related with the searching keyword. The green circle ($B \cup C$) represents the discovered set of records by using a specific searching mechanism. In a perfect world these two circles should overlap. Precision rate refers to the percent of records in returned datasets are current, which can be calculated as $C/(B \cup C)$, while recall rate means how much percent of the “true” set of records is covered in the returned datasets. Recall rate can be calculated as $C/(A \cup B)$. Obviously

the results acquired by using full-text keyword-matching technique will have relatively low recall rate.

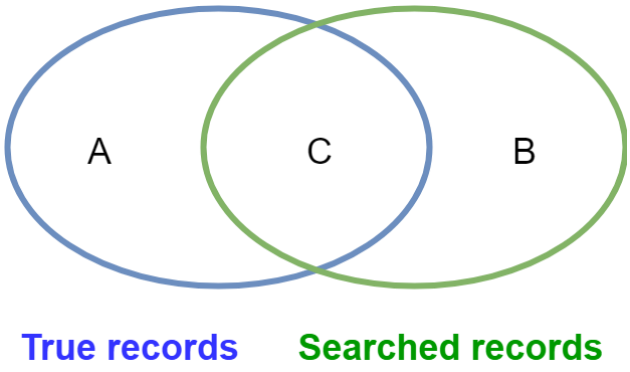


Figure 3 Illustration of the recall and precision

Semantic similarity is widely accepted as a promising solution for improving the precision and recall rates of data discovery tasks. Similarity measures have long been studied in the fields of information retrieval, artificial intelligence etc. Recently, these measures have been extended and reused to measure similarity (Janowicz, Raubal and Kuhn, 2011). When metadata participates in the semantic search, the most useful part is the descriptive text fields, such as title, abstract and keywords. The semantic similarity measure could be based on terms/words which comprise the metadata and have a finer granularity to measure the similarity of concepts. It can also be based on a higher level that treats each individual dataset as an integrated entity and directly measure the similarity among them.

On the terms/words level, domain ontologies can be incorporated to identify associations between concepts (such as polysemes and synonyms) related to users' query, based on which a list of related search terms could be recommended to help refine the search. Relevant work includes WordNet (Miller, 1995), Semantic Web for Earth and Environmental Terminology (SWEET) (Raskin and Pan, 2005), Geosciences Network (GEON) (Bowers, Lin and Ludascher, 2004), Linked Environments for Atmospheric Discovery (LEAD) (Droegemeier *et al.*, 2005), Noesis (Movva *et al.*, 2008), GeoSPARQL

(Battle and Kolas, 2012), GeoLink (Krisnadhi *et al.*, 2015) etc. Ontology knowledgebases usually possess the advantages of high quality, meaningful hierarchical structure and precise relations among ontologies since they are mainly developed under experts' supervision. However, ontology knowledgebases suffer from a limited coverage (Banea *et al.*, 2014). Some researchers also point out that this relies heavily on humans' manual input and definition, which will bring another issue that people with different knowledge background tend to have different perspectives on the categorization of terms as well as their linkages and relations. This would lead to heterogeneous representations and conflicting statements, and eventually influence the effectiveness of a search engine (Li, Wang & Bhatia, 2016).

In addition to building an ontology knowledgebase, many researchers focus on automatically extracting semantic relationships between spatial datasets using machine learning approaches. Li, Raskin and Goodchild (2012) adopted an artificial neural network algorithm called Multiple Layer Feed-Forward Neural Network (MLFFN) to help measure the similarity between datasets. Hu, Ã. K. Janowicz, *et al.* (2015) employed the machine learning method, namely Labeled Latent Dirichlet Allocation (LLDA, a supervised version of LDA, Blei *et al.*, 2003) to extract the topics of each dataset and the similarity between them. Jiang *et al.* (2017) introduced a large volume of user search histories from the PO.DAAC website as the supplementary materials for semantic processing. Similar work can also be found in a number of research (GuoDong, LongHua and QiaoMing, 2009; Gollapalli, Li and Wood, 2013; Liu *et al.*, 2014; Li, Wang and Bhatia, 2016).

The emergence of word embedding technologies in recent years have drawn much attention from researchers. The word embedding models treat words as vectors and train the vectors upon <word, context> pairs in the local window. The basic hypothesis is that words with similar meanings will be embedded into a similar context. Among various

word embedding models, the Word2vect model (Mikolov, Chen, *et al.*, 2013; Mikolov, Sutskever, *et al.*, 2013) has been enjoying wide application due to its effectiveness of automatically capturing semantic meanings of words more precisely than other models, as well as its efficiency of processing extremely large datasets. Besides word2vec, other word embedding models such as GloVe (Pennington, Socher and Manning, 2014) and fastText (Bojanowski *et al.*, 2016) have been widely adopted as well. In addition to the word embedding models, phrase embedding, sentence embedding, paragraph embedding, and document embedding models have been developed recently to measure the semantic relationship among different hierarchical level corpus for different application scenarios (Cho *et al.*, 2014; Zhang *et al.*, 2014; Wieting, Bansal, Gimpel and Livescu, 2015; Gan *et al.*, 2016; Melamud, Goldberger and Dagan, 2016; Conneau *et al.*, 2017; Zhou, Huang and Ji, 2017; Dwivedi, 2017; Jansen, 2017; Sato *et al.*, 2017; Wang, Zhang and Zong, 2017; Young *et al.*, 2017). In this chapter, I adopt a phrase embedding method to help measure phrase/word similarities in our research datasets. The details of the method will be discussed in the next section.

2.3 Methodology

2.3.1 Geospatial Metadata

The building blocks of this research are thousands of data repositories harvested from the Internet, which result from the pioneer studies of Li et al (Li, 2017; Li, Wang, & Bhatia, 2016; Li, Yang, & Yang, 2010). Based on the previous work, more than 70K geospatial data providing services distributed in ninety-five countries have been found, hosting more than millions of data layers mainly published through Open Geospatial Consortium's (OGC) Web Map Service (WMS; de La Beaujardiere 2006) and Web Feature Service (WFS; Vretanos 2004). WMS is the standard protocol for serving georeferenced map images through the Internet while WFS is the standard protocol for

serving geographical features (vector) data (Shao and Li, 2018). Both the WMS and WFS standards support the “get-capabilities” operation, which provides the “get-capabilities” XML file describing series of both human- and machine-readable information about the service, including 1) information about the data providing service itself (Service Identification), 2) metadata about the organization providing the service (Service Provider), 3) metadata of the supported operations (Operation Metadata), and 4) a metadata list describing all the data layers hosted on the service, etc.

Table 1 demonstrates an example of the “get-capabilities” XML file extracted from a WFS data layer’s metadata section. In this metadata section, properties of the layer, such as name (as id), title, abstract, keywords, and bounding box are provided. Such information plays a critical role in helping users get the perception of the layer’s content and characteristics. It is also essential for data retrieval in later steps.

Table 1 Example of a WFS Layer get-capability content

```
<FeatureType xmlns:epi="http://sedac.ciesin.columbia.edu/data/collection/epi">
  <Name>epi:epi-environmental-performance-index-2010_water-effects-on-ecosystems</Name>
  <Title>EPI 2010: Water Effects on Ecosystems</Title>
  <Abstract> Environmental Performance Index, 2010 Release (1994-2009): Water Effects on
Ecosystems displays the indicators within the water effects on ecosystems policy category of EPI.
See more information at http://dx.doi.org/10.7927/H4D21VHT. </Abstract>
  <ows:Keywords>
    <ows:Keyword>agriculture</ows:Keyword>
    <ows:Keyword>climate</ows:Keyword>
    <ows:Keyword>conservation</ows:Keyword>
    <ows:Keyword>governance</ows:Keyword>
    <ows:Keyword>health</ows:Keyword>
    <ows:Keyword>marine-and-coastal</ows:Keyword>
    <ows:Keyword>sustainability</ows:Keyword>
    <ows:Keyword>water</ows:Keyword>
    <ows:Keyword>epi-environmental-performance-index-2010</ows:Keyword>
    <ows:Keyword> epi-environmental-performance-index-2010_water-effects-on-ecosystems
    </ows:Keyword>
  </ows:Keywords>
  <DefaultSRS>urn:x-ogc:def:crs:EPSG:4326</DefaultSRS>
  <ows:WGS84BoundingBox>
    <ows:LowerCorner>-180.0 -55.792</ows:LowerCorner>
    <ows:UpperCorner>180.0 83.667</ows:UpperCorner>
  </ows:WGS84BoundingBox>
  <MetadataURL type="FGDC" format="text/plain"> http://sedac.ciesin.columbia.edu/data/set/epi-
environmental-performance-index-2010/metadata </MetadataURL>
</FeatureType>
```

2.3.2 Metadata Enrichment

Metadata is the primary material for building the data search system. Hence its quality heavily affects the performance of the searching result (Hu, K. Janowicz, Prasad and Gao, 2015). The spatial datasets used in this research are collected from a large number of data providing services scattered around the world. Thus, the metadata quality varies. OGC's geospatial data sharing standards do not regulate quality of the metadata. Attributes of metadata such as title, abstract, and keywords provide descriptive information of the data content, which can be used for data search. Unfortunately, such attributes are incomplete, or even missing in a certain proportion of the datasets.

As shown in Table 1, the metadata includes a <*MetadataURL*> section whose content is a URL link pointing to some external metadata resource, which usually contains much more detailed information about the data layer. The external metadata is expected to follow some specific standards, such as Digital Geospatial Metadata (CSDGM) from the Federal Geographic Data Committee (FGDC), ISO TC211 19115, or ISO TC211 19139 (Vretanos, 2004; de La Beaujardiere, 2006), making them relatively easy to be parsed. Such external metadata provides a possible solution for improving the situation of a lack of appropriate metadata in some data layers: on one hand, the information extracted can be harmonized into the layer's original metadata to improve the metadata's quality; on the other hand, the description document can be used for training the phrase representation model in the next step.

According to our experience, the organizations who provide geospatial datasets usually host corresponding web portals as well. Rich context information about the geospatial dataset can be found in the web portals, such the description about their ongoing project, their study area, data acquisition methods, working background etc. Although such information cannot be directly used to enrich the metadata, they can still be used for training the phrase representation model in the next step.

2.3.3 Measuring semantic relationships in the metadata

The recently developed word embedding technology – Word2Vec – will be adopted for learning the word representation. The word embedding is the generous name for those language modeling and feature learning techniques which project words into a vector space (Bartusiak *et al.*, 2017). Word2vec is based on probabilistic prediction approach, which trains the word vectors based on their contextual neighbors inside a specific window size (usually around 5). The basic assumption is that semantically related words are more frequently co-occurring in the training corpus, and similar words have similar contexts. After the training by Word2Vec, a word is represented by a vector and its context property is preserved in the vector space. That is to say, for those words which co-occur more frequently in the corpus, their representing vectors will also have shorter distances in the vector space. Therefore, given a specific word, it will be very easy to find its semantically related words by looking for its close vectors in the vector space.

The Word2Vec contains two core architectures for learning distributed representations of words, namely Skip-gram and CBOW (the continuous bag-of-words model). These two models are similar in the algorithm, while Skip-gram targets to find word representations which are useful for predicting the surrounding words in the context, CBOW does it in a reversed fashion, which tries to predict the current word based on its context. The performance of these two models varies across corpus (Liu and Gao, 2017). The Skip-gram model is adopted here. The training objective is to maximize the log probability:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c}\log p(w_{t+j}|w_t)$$

where c is the size of context window (which is set as 9 in our practice), training words are represented as $w_1, w_2, \ldots, w_T$. In the Skip-gram model, the original probability function p is a softmax function:

$$p(w_O|w_I) = \frac{\exp\left({v'_{w_O}}^T v_{w_I}\right)}{\sum_{w=1}^{W} \exp\left({v'_w}^T v_{w_I}\right)}$$

where v_w and v'_w are the input and output vector representations of the word w, and W is the total number of words in the corpus. The computation cost of the full softmax function is very expensive, Mikolov et al. adopted a hierarchical softmax as the approximation in Skip-gram which significantly improved its efficiency. Besides, Word2Vec can also well preserve the linear regularities among words compare with other models such as LSI or LDA (Mikolov, Chen, *et al.*, 2013), making it possible to apply binary operations on word vectors to extend the model.

In many NLP scenarios, it is more reasonable to treat both words and phrases as the basic composite units of sentences, paragraphs and documents. For example, 'New York', 'green house' and 'point of interest' are more semantically integrated as phrases than separate words. For the spatial data discovery task, it is also more meaningful and common for users to provide phrases instead of single words during data search, such as 'wild fire', 'sea surface temperature', and 'US annual economic data'. Therefore, it should be more appropriate to learn both words and phrases representations and use such information to assist data discovery.

There exist two popular strategies for learning phrase representations. The first one treats phrase as an indivisible term (pseudo-word) and learns phrase embedding based on its external context similar to the word embedding methods. The second one acknowledges the meaning of words which comprise the phrase, and uses compositional methods to learn phrase representations. While the first method is suitable for learning short phrases (e.g. bi-word phrase) with a very large corpus (Mikolov, Sutskever, *et al.*,

2013; Peng and Gildea, 2016), it cannot take advantage of the information embedded in the words which comprise the phrase. Besides, it suffers from data sparseness for those multi-words phrases which rarely appear in the corpus (M. Li *et al.*, 2018). Hence, more efforts have been dedicated to developing the compositional models to jointly learn word and phrase representations in recent years (Anoop and Asharaf; Socher, Manning and Ng, 2010; Zhang *et al.*, 2014; Yin and Schuetze, 2014; Zhao, Liu and Sun, 2015; Lebret and Collobert, 2015; Yin and Schütze, 2016; Hashimoto and Tsuruoka, 2016; Zhou, Huang and Ji, 2017; Dwivedi, 2017; Sato *et al.*, 2017; B. Li *et al.*, 2018). Simple operations on word vectors such as add (additive model) and point-wise multiplication (multi model) could be very efficient and produce well-performed phrase representations to fulfill general NLP tasks (Mitchell and Lapata, 2010; Blacoe and Lapata, 2012; Lebret and Collobert, 2015; Wieting, Bansal, Gimpel, Livescu, *et al.*, 2015; Wang and Zong, 2017). While more complicated methods for learning phrase representations, such as Matrix, RNN (recurrent neural network), and LSTM (Long short-term memory) are proposed to improve the accuracy (Socher, Manning and Ng, 2010; Cho *et al.*, 2014; Yu and Dredze, 2015; Zhao, Liu and Sun, 2015; Hashimoto and Tsuruoka, 2016; Dwivedi, 2017; B. Li *et al.*, 2018; M. Li *et al.*, 2018), they usually need to be fed with high-quality training data, such as positively related phrase pairs, and carefully tuned in order to achieve high accuracy. What's more, the training time is significantly longer than the additive model and multi model. Other factors such as the profile of training data, how the word representation is pre-trained, and how the objective function is selected could all affect the performance (Wang and Zong, 2017).

In this chapter, the additive model is adopted for automatically calculating the phrase representations in the same vector space as words. Then, the word and phrase representations will be used in two places: 1) when a user types a keyword, the relevant phrases and words will be quickly extracted and provided to the user for selection,

encouraging the user to provide more specific and unambiguous query criteria to help improve the query performance; 2) given a query word or phrase, the related words/phrases will be found by calculating the cosine similarity among their representing vectors, followed by the full-text matching with these words/phrases in the database to find the appropriate datasets. This step will significantly improve the recall rate of searching results.

2.4 Experiments and Results

2.4.1 Experimental Dataset Profile

A subset of data services is separated from the massive database for the experiments. The criteria for selecting the samples include: 1) The language used by the service should be English, 2) Each data service contains no less than 200 spatial data layers. For all the services that meet the criteria, 303 of them are randomly selected for experiments, which contains 163,285 data layers.

Figure 4 demonstrates the summary statistics about the experimental datasets. From Figure 4.a we can see most of the services contain less than 1000 data layers. As shown in Figure 4.b, even after excluding layers whose abstracts contain less than 8 words, we can observe a large proportion of layers with short abstracts, an indicating of poor quality.

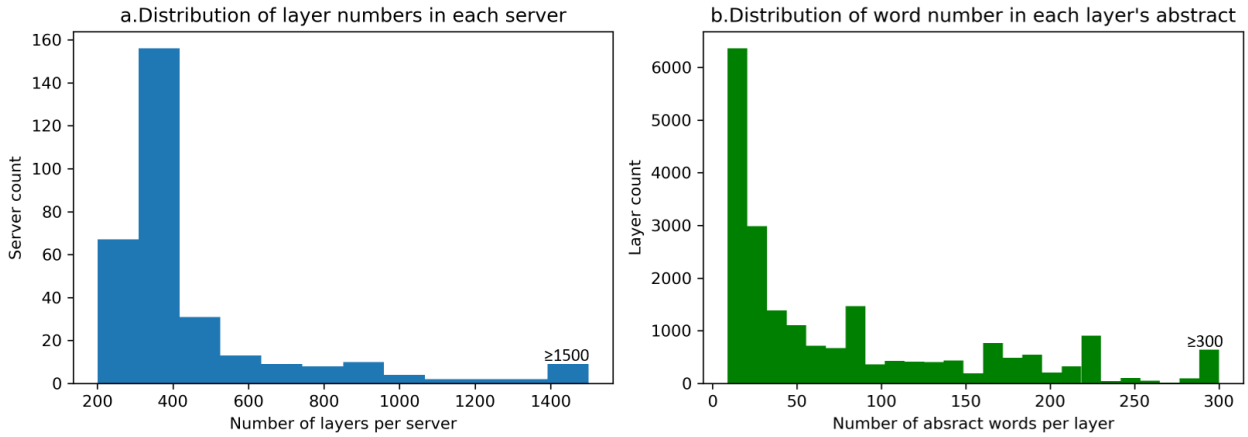


Figure 4 Static profile of experimental datasets

Table 2. demonstrates the summary statistics concerning missing attributes of our experimental layers. We can observe that there is a large proportion which have incomplete keywords and abstracts.

Table 2 Statistic of missing attributes in experimental layers

Attributes	Number of missing layers (in percentage)
Title	193 (0.12%)
Keywords	51600 (31.6%)
Abstract	105601 (64.7%)

2.4.2 Metadata Enrichment

After parsing the metadata, I detected 13199 external metadata URLs from 9162 layers, accounting for 5.6% of the experimental datasets. Figure 5 illustrates how different metadata standards are supported by the metadata of geospatial data layers. From the graph, we can see ISO TC211 19115 is the most popular standard. After the metadata records are crawled, they will be used for enriching original metadata of each layer and training the phrase representation model.

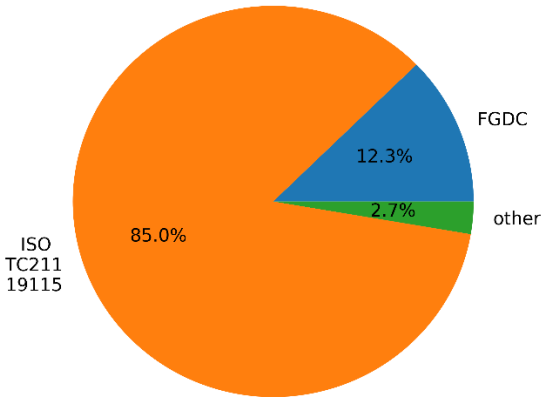


Figure 5 Supported external metadata standards by experimental data layers

For the experimental services, if their organizations also host website portals, the documents in the websites can potentially provide rich context information about the spatial dataset. After manual check, I located 74 websites which are directly related with the experimental datasets. Then I employed Apache Nutch[1] to crawl the websites and retrieve the documents. Finally, 146,482 web pages were acquired, from which 119MB text-based documents are extracted. These documents will be used for the phrase representation model training.

2.4.3 Word and Phrase Representation Training

Word representations are calculated using Word2Vec in the first step. The corpus for model training consists of 1) titles, abstracts, and keywords extracted from all experimental layers' metadata and 2) webpage documents crawled from the portals of data providers. The Word2Vec model in Gensim[2] library is employed for training the word representations. Basic text preprocessing steps are conducted before the training, include removing stop words and lowercasing all words. Configurations for the training process include 1) training algorithm: Skip-gram, 2) window size: 9, 3) minimum count of vocabulary: 2, 4) word vectors dimension: 100.

[1] http://nutch.apache.org/

[2] https://radimrehurek.com/gensim/

In the second step, I first extracted all the phrases from the corpus. The part of speech (POS) annotation methods in Stanford CoreNLP library (Manning *et al.*, 2014) was employed. Figure 6 demonstrates the statistical information of extracted phrases. Figure 6.a shows the distribution of each phrase's appearances in the corpus. A lot of phrases appear rarely here since the size of the corpus is not very large. Hence, it will be not suitable to use the pseudo-word strategy for phrase representation calculation. Figure 6.b shows the distribution of phrase length, we can find that the majority of the phrases are relatively short (containing less than 5 words).

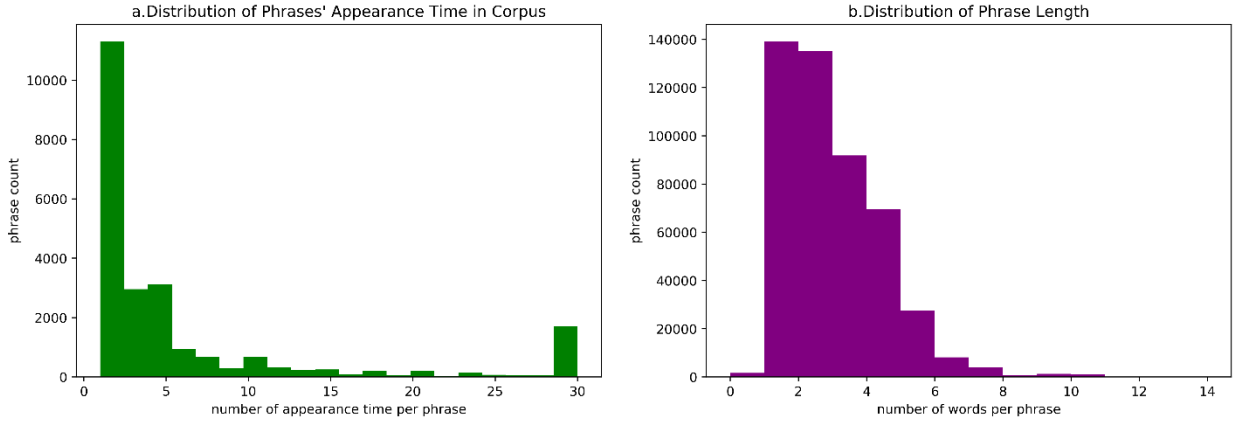


Figure 6 Statistical information of extracted phrases

After the phrases are extracted, the representing vector of each phrase can be calculated by averaging its component word vectors:

$$V_p = \frac{\sum_{i=1}^{n} V_{w_i}}{n}$$

where V_p is the vector of phrase p, which contains a word sequence of $w_1, w_2, \dots, w_n$. For word w_i, its representing vector is represented as V_{w_i}. The calculated phrase representations and the previous word representations belong to the same vector space. Similarity of any pair of word or phrase can be calculated by using their cosine similarity:

$$sim(X, Y) = \frac{\sum_{i=1}^{d} X_i Y_i}{\sqrt{\sum_{i=1}^{d} X_i^2} \sqrt{\sum_{i=1}^{d} Y_i^2}}$$

where X and Y are the vectors of a pair of units (word or phrase) with dimension d ($d = 100$ in our experiment). X_i and Y_i are the components of vector X and Y respectively.

I trained the LSI model, Word2Vec model for single word and pseudo-word model on the corpus and compared them with the additive word and phrase representation model. The comparison results are presented in Table 3: for some query term examples, the top 20 most similar terms in each model are listed. For the LSI model, since similarity measurement can only apply to single words, no result will be returned for phrases. The results in Table 3 indicate: 1) The pseudo-word model performs poor for the similarity measure task, 2) For single word similarity measurement, the Word2Vec model performs better than LSI, 3) Comparing the Word2Vec model with the additive model, for single words, the former performs better in some cases ('coastline', 'rice', 'road') and worse in others ('train', 'marine', 'forest', 'earthquake', 'rain'), while for the multi-word phrases, the latter performs much better than any other models.

Table 3 Comparison of top 20 most similar terms returned with different models

Query term	LSI	Word2Vec	Pseudo-word model	Additive model
train	lines, transmission, features, point, buffers, electric, data, boundary, article, chesapeake, anonymised, stations, tline, uae, ais, quality, admin, spatial, bay, railroads	stops, ride, bus, tube, railway, commuter, trains, tram, rails, centreline, passenger, buses, passengers, routes, ptv, wettbewerb, wagons, electric, riders, carpark	lightrail, quartermile, junctions, junction, prek, parent, permissions, crossings, wildfire, housed, tubes, grades, ridgeline, srilanka, sfpd, landfills, spdes, kgra, paths, trains	a train or tram, train and road, a railway centreline, stops, ride, train stations, bus, rail station pnt, tube, railway, major bike facilities, commuter, a gazetted railway, trains, trains or trams, the bus lines, tram, rails, station point locations, centreline
marine	marine, legacy, areas, zones, line, habitat, conservation, points, licences, distribution, consents, point, licenses, hawaii, applications, data, algae, polygon, species, ocean	doñana, psac, mussel, detecting, hab, breeding, whale, birds, mss, habour, frithjof, reintroduction, stock, sjølve, harbours, argo, fjords, undercover, sightings, namibian	ecological, major threats, restore, environment, wildlife, reserves, applications, managed, reserve, nature, communities, sometimes, community, importance, vulnerable, scotland, generations, enhancement, programme, trophic	discontinued marine, marine beacons, this marine refuge, marine obstruction, capad 2012 marine, marine turtles, marine polys, the marine portions, the marine ecoregions, marine faunal distributions, the marine portion, the marine animals, marine mammal, marine mammals, marine biogeographic patterns, benthic marine, the marine reserves, marine components, marine plants, marine biology
coastline	ne, data, admin, natural, earth, features, new, zealand, boundaries, nz, linz, areas, provided, graticules, abstract,	coastlines, portions, margins, estuaries, navigable, cliffs, lakes, ridges, depicting, extends, shelves, surrounding, segments, submerged, lines, continent, lagoons,	regions, all, geodatabase, graves, reflects, located, various, damaged, simply, nwhi, were, specified, one, intended, represents, under, imcra, aggressors, inhabited, possibly	the victorian coastline, shoreline and coastline, indonesia coastline, the coastline definition, coastlines, 10m minor islands coastline, portions, the coastline component, island polygons, osm coastlines, submerged portions, provinces lakes, ponds lakes and dam boundaries, a man made coastline, gbrmpa reefs gbr

	govt, http, marine, information, land	shoreline, beaches, continents		features coast, glacial lake boundary, coastal lines, 0 boundary lines, boundary lines, the boundary lines
rice	geotiff, wcs, features, ton, biomass, billion, update, energy, area, dry, county, global, btu, national, nrel, data, wind, program, office, crops	wheat, millet, yield, smallholder, pasture, crops, cassava, maize, migration, potatoes, sorghum, oats, crop, farmers, barley, agriculture, urbanization, tilapia, livelihoods, miraca	wheat, maize, production, cassava, residues, sugarcane, demand, yield, cereal, severity, wood, crop, cereals, sugarbeets, corn, potatoes, total, worldwind, ha, phl	rice or maize, maize rice and wheat, rice maize cassava and sweet potatoes, wheat rice maize barley oats rye millet sorghum buckwheat, wheat, millet, cereal yield, crops or crop varieties, cereal crops, yield, smallholder, pasture, crops, cassava, maize, in migration minus out migration, migration, potatoes, sorghum, oats
forest	land, cover, poly, alb, features, wcs, areas, geotiff, landuse, data, forest, globcover, gc, flood, adg, regional, abstract, provided, en, aus	pasture, deciduous, deforestation, crops, coniferous, forests, conifers, timber, fires, grassland, growing, rangeland, grasslands, peatlands, cropland, wildland, vegetation, trees, evergreen, agricultural	cropland, pasture, sparsely, evergreen, broadleaved, dominantly, irrigated, filling, local, sddc, tenure, deciduous, croplands, acts, vegetated, suited, forests, dominant, enhancement, irrigation	a dense swamp forest, forest reserves, forest cover indicator, forest cover, denr ncr mini forest established, forest category, forest types, forest interior habitat, dry land forest, forest and snow areas, forest conservation easements, ecps fdps forest conservation plats, pasture, deciduous, deforestation, natural vegetation, crops, coniferous, forest areas data, forests
earthquake	provided, abstract, baikalgis, data, ocean, level, coastal, water, rise, sea, pacific, earth, inundation, hawaii, science, model, global, area, mhhw, high	seismology, complementary, tsunamis, landslide, landslides, epicenters, fatalities, wales, quake, liquefaction, aftershocks, tsunami, experienced, spontaneous, southern, iceland, kyriopoulos, cyclones, intense, floods	mw, frequency, post, chile, risks, kamchatka, localised, 1952, 1960, 8 2 mw, liquefaction, events, mortality, hazard, cyclone, philippines, volcano, 9 5 mw, 9 0 mw, 1957 aleutian earthquake	earthquake epicenters, the kaikoura earthquake 2016, the christchurch earthquake, 1957 aleutian earthquake, i 1 the 1946 aleutian earthquake 8, the 1964 alaska earthquake, significant earthquake, the recent earthquake, the 22 february 2011 earthquake, earthquake hazard, post earthquake, seismology, complementary, global earthquake hazard, earthquake mortality loss estimates, landslide fatalities, tsunamis, global earthquake hazard frequency, landslide and drought, landslide
rain	forecast, precipitation, geotiff, wcs, article, land, probability, global, lightning, radar, hourly, model, panam, map, landuse, unit, climatestop, temperature, system, rain	cloudiness, observations, numbers, rains, torrential, nuuksio, snowfall, lightning, qpf, overflowing, europe, auroras, finland, precipitation, inches, rainy, flash, temp, thunderstorms, winds	wheat, maize, cereal, fed, satiation, cereals, total, hectare, harvested, ago, rainfed, precipitations, worldwind, kilograms, accum, ferman, sweet, rice, ha, m3	freezing rain, cloudiness, rain count, rain days, rain std error, observations, levelling observations, buffalo numbers, reduced observations, other reduced observations, most reduced observations, the daily precipitation observations, zebra numbers, wildebeest numbers, elephant numbers, numbers, giraffe numbers, rains, odd numbers, the adjusted reduced observations
road	roads, england, road, noise, data, lden, national, laeqh, lnight, network, rail, features, nz, reserves, special, linz, topo, ortho, muni, layer	railway, roads, vehicular, roundabout, rail, street, highway, frontage, footpaths, muswell, pedestrians, euston, crossing, patrols, hgv, lanes, lane, bus, archway, lawn	roads, mot, addresses, name, locality, identifier, logistics, railroad, id, street, landonline, records, electoral, network, wfp, railway, connections, places, centreline, referencing	the road distant, the road markings, a road embankment, road shapes, frederick road, a road centreline, some road centreline geometries, the electoral road, addressing road, electoral road subsection, these road centrelines, the road centrelines, train and road, road layout, road and railway centrelines, road labels, a road or track, road sections, some road sections, seasonability road condition and practicability
road network		roads, railway, interchange, wayfinding, rail, euston, lanes, patrols, bus, hgvs, railroad, odenton, markings, roundabout, resurfacing, lighting, trains, entrances, cyclists, railways	roads, mot, logistics, subsections, practicability, openstreetmap, geometries, landonline, wfp, transportation, places, tracks, electoral, thana, addresses, code, identifier, cadastral, mooring, railways	the road network, road network, this road network, constrained road network, the wfp road network, madagascar road network, cameroon road network, the main road network, nigeria road network, the railway network, strategic road network, roads network, nepal road network, the railways network, addressing road, the emerald network, the road markings, the road distant, a road embankment, road and hydro
parking space		car, servicing, doors, freight, cars, unreasonable, dropped,	fema, nal, corridors, neighborhoods, retail, schemes, works, mot,	parking signes, paved parking, no parking signs, paved parking lots, residential parking, other open space, open space, open

	taxi, spaces, kerb, cpz, lighting, entrances, commuter, buses, trains, passengers, suspension, walls, railings	frontage, strategy, paths, necessarily, aeronautics, fod, walking, food, locally, england, opsnatgs, transportation	space comprising, controlled parking zones, parking features, car, passengers or freight, open space uses, servicing, doors, time and space, freight, cars, unreasonable, dropped
wild fire	micronutrient, populations, insects, tissue, mussels, pose, breeding, farmed, milk, regime, juveniles, survival, animals, protein, crustaceans, farms, eggs, mixtures, feeding, hippoglossus	drp, leisure, dist, tracts, hbc, cen, tenure, appl, fod, comprised, gateshead, polling, mixed, recycling, cockles, ebtjv, centres, rst, 1977, frontage	fire hydrants, selman fire, starbuck fire, fire arms practise, fire districts, the fire districts, fire and rescue, fire districts centroids, live fire training, the fire departments, cots populations, fire stations, micronutrient, wild areas, fish farm poly, shellfish farms, livestock and wildlife, populations, regime breakdown, plant or wildlife
remote sensing	sens, proximal, ieee, spaceborne, satellitbilleder, longterm, networks, xlinks, fibers, geosci, sensors, µm, situ, sensed, remotely, multispectral, xiaoguang, challegens, anvendelser, optical	authorities, gamma, drsrs, consquences, health, action, stewardship, late, surveys, government, department, cleanup, ecl, help, serviceprovider, reduce, environmental, conservation, requires, responders	remote sensing, semi automated methods and remote sensing images, satellite remote sensing products, its remote location, remote areas, the most remote areas, sens, proximal, the most remote coral atolls, ieee, spaceborne, satellitbilleder, longterm, networks, xlinks, fibers, geosci, optical sensors, sensors, µm
satellite imagery	multispectral, worldview, avhrr, panchromatic, hyperspectral, microwave, radiometer, acquired, landsat, rapideye, polarimetric, orthorectified, rectified, orthoimagery, sensors, gsd, ikonos, aerial, mosaic, eo	interpretation, retrievals, utilizes, supplied, aqua, derives, worldview, robust, transverse, instrument, ikonos, visual, sensor, solar model, visible, wavelengths, ann, meteorology and solar energy global data, mercator, photography	orthorectified satellite imagery, the satellite imagery, quickbird satellite imagery, recent satellite imagery, an high resolution satellite imagery, the worldview 2 satellite, satellite retrievals, multipectral ikonos satellite data, the 2006 quickbird imagery, a imagery was captured fo, a imagery was captured for t, a imagery was captured for, t imagery was captured for the, s imagery was captured for the, high spatial resolution satellite imagery, aerial imagery, visible imagery, hourly satellite, the satellite era, the supplied imagery
surface temperature	salinity, temperatures, °c, depth, calculated, measured, emitted, vertically, anomalies, constant, relative, cooler, simulated, variations, wavelength, velocity, concentration, swe, humidity, assuming	pressure, global summer, maximum, runoff, forecasts, daytime, skin, downward, salinity, optical, winds, nighttime, celsius, mph, flux, mbar, humidity, instru, ‰, plays	surface temperature, minimum surface temperature, maximum surface velocity, temperature and salinity, surface gravitational acceleration, average summer daytime maximum surface temperature, a quantitative surface, measuring temperature, skin temperature, the surface craft, sufficient temperature, surface roughness, temperature isolines, the maximum daytime land surface temperature, temperature and depth ranges, surface pressure, surface cells, the minimum nighttime land surface temperature, smu temperature, sea surface temperature

2.4.4 System Implementation for the Geospatial Data Search Engine

The Geospatial data search engine is implemented and integrated into our cyberinfrastructure portal – GeoCI. Figure 7 demonstrates the architecture of the search engine and how it interacts with other components of the system.

In GeoCI, users can provide their search keywords and filtering conditions to find specific geospatial datasets. While the user is typing, the semantically related terms of

those keywords will be quickly retrieved and presented to users for selection. Through such interaction between user and GUI, more detailed and unambiguous searching keywords will be produced. Once the user is satisfied with their searching keywords, the keywords and their semantically relevant terms will be calculated based on the additive model and delivered to a full-text matching engine to find their related geospatial data records. Elastic search[3] is employed here for the full-text matching task. When the metadata records are discovered, the filtering conditions such as boundary box and data collecting time range will be applied to the datasets. Those records that do not meet the filtering condition will be removed. After the filtering, the record will be ranked according to their similarity distance to the query keywords. The cosine similarity distance will be used here for calculating the value. Finally, the ranked records will be returned to the user as geospatial data layer candidates. Figure 8 demonstrates the interactive data search GUI implemented in GeoCI.

Note that till this step, the returned candidates are still metadata records. When the user selects some of these candidates and adds them to his/her working space in GeoCI, the system will automatically go to the data providing services to acquire the real datasets on-the-fly. After the real datasets are transmitted to the web portal, they can be used for visualization and conducting exploratory spatial-temporal data analysis. Figure 9 demonstrates a snow depth statistic scenario in the north polar region by using the discovered dataset and built-in spatial analysis tool in GeoCI.

[3] https://www.elastic.co/products/elasticsearch

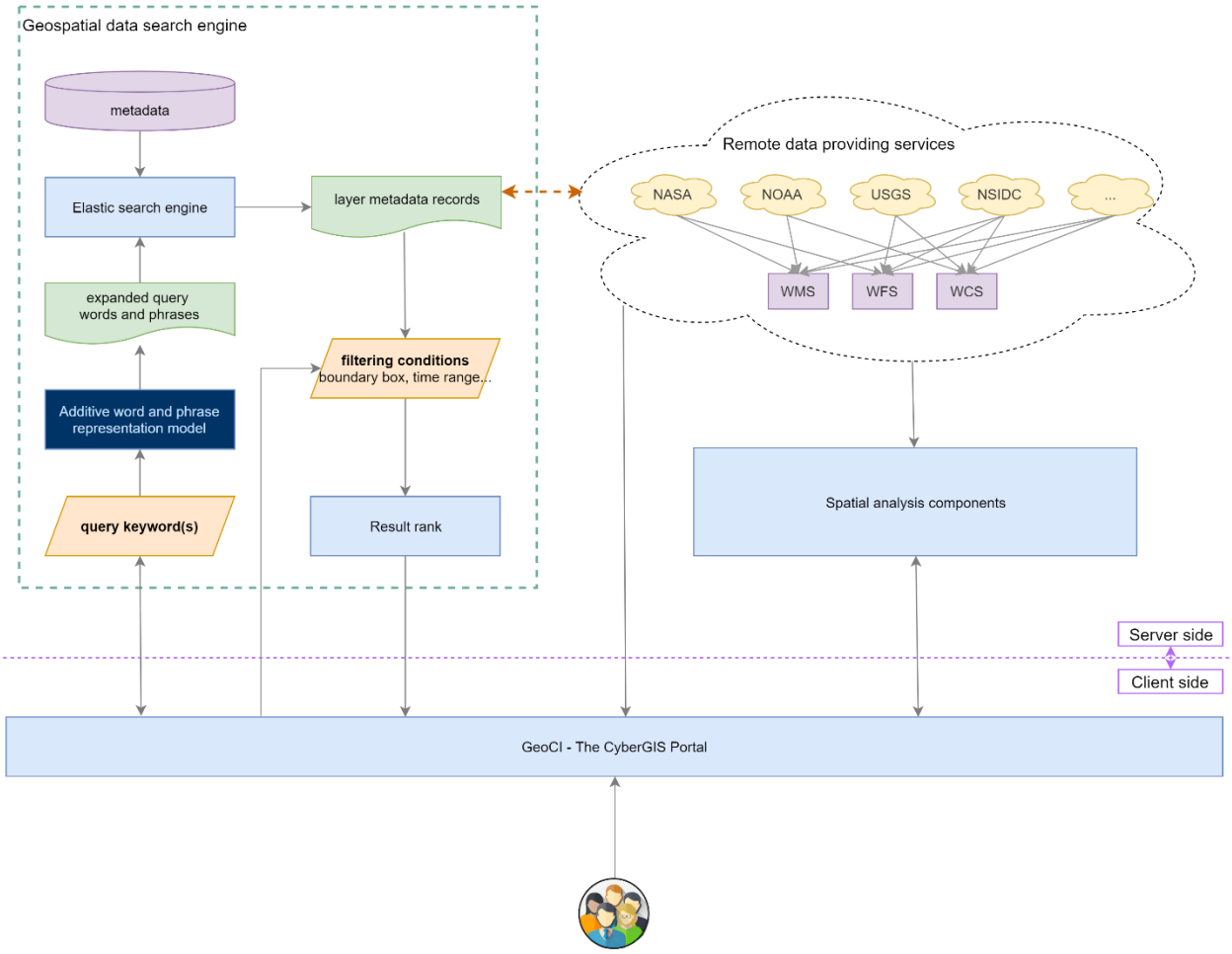


Figure 7 Architecture of the semantic search system

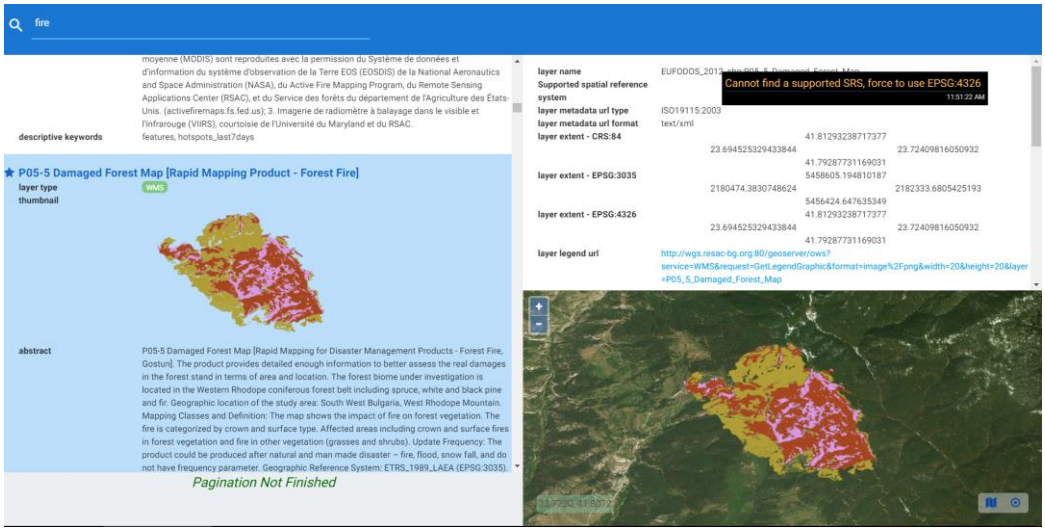


Figure 8 GUI for semantic enhanced geospatial data search

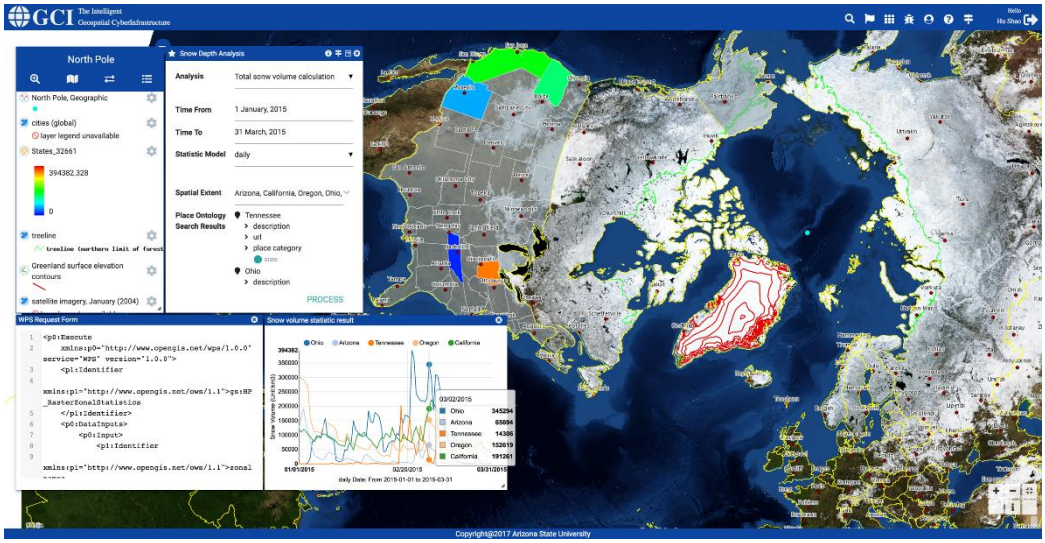


Figure 9 Exploratory spatial-temporal analysis with discovered dataset

2.5 Discussion and Conclusion

With the technological advancement, numerous geospatial datasets are being collected and shared on the Internet by different organization scattered around the world. These massive geospatial datasets introduce great research opportunities to the GIScience field. Faced with the data ocean, there exists a critical but challenging task to develop a geospatial data discovery mechanism to help researchers and public users efficiently and conveniently find appropriate datasets from millions of data records.

This chapter is focused on developing a semantically enhanced data discovery system to assist users in finding geospatial datasets from hundreds of thousands of geospatial data layers provided by thousands of organizations. The state-of-art word and phrase representation methodologies from the NLP filed are adopted to automatically extract semantic relationships among individual words and phrases in our metadata. A metadata enrichment strategy is adopted to improve the data quality and enhance the model training results. The data discovery system is implemented and integrated into a cyberinfrastructure portal named GeoCI for providing the search functionalities to public users.

Future research could be focused on the implementation of a more effective evaluation system for comparing the precision and recall rates of our system with the baseline system based on full-text match search and LSI method. In this research, the POS method is adopted for extracting phrases from our metadata. In the future, more sophisticated entity recognition methods based on the neural network models could be adopted to improve the search result. Besides, adopting the high-quality geospatial ontology knowledgebases (e.g. GCMD) in the result filtering and ranking stages could potentially improve the search result.

3 WHEN PYSAL MEETS GEOCI: TOWARDS AN INTEROPERABLE AND REPLICABLE CYBERINFRASTRUCTURE FOR ONLINE SPATIAL-STATISTICAL-VISUAL ANALYTICS

3.1. Introduction

The Geographic Information Science (GIScience) has ushered tremendous development in recent decades. Meanwhile it continuously contributes to multidiscipline by means of providing modern theories, methodologies, softwares and tools to help solve scientific problems and improve decision-making practices (Shaowen Wang, 2013). With the advancement of GIScience, there exist numbers of vibrant GIScience teams working on integrating the most advanced algorithms and methodologies into open source libraries or software toolkits (Li, Di, Han, Zhao, & Dadi, 2010; Steiniger & Hunter, 2013; Swain et al., 2015), such as Python Spatial Analysis Library (PySAL) (Luc Anselin & Rey, 2014; Sergio J. Rey, 2014; S. J. Rey & Anselin, 2007), GeoDa (L. Anselin, Syabri, & Kho, 2010), GDAL, GRASS GIS, GeoTools, GeoPython, spaceime (Pebesma, 2012), STARS (Sergio J. Rey & Janikas, 2006), spdep (Bivand et al., 2011) etc. These toolkits play a critical role in promoting the innovation in GIScience.

Among various working modes in the GIScience field, there are two typical types. The first one is “single-user” oriented, which is most suitable for individual researchers who possess professional domain knowledge. They generally conduct research and experiment from the exploratory perspective and in the back-and-forth manner. Since this working mode gives researchers absolute control on what data and materials to

prepare, as well as what analytical methods and software to adopt, it is very popular among individual researchers.

On the other hand, with the advancement of technologies, the "single-user" working mode in a localized computing environment is infeasible in scenarios including:

- For those very large projects that require the collaboration of participants from different domains and physical locations, the dataset, documents and knowledge must be simultaneously shared among the team in an efficient way (Rinner, Keßler, & Andrulis, 2008; Sun & Li, 2016).
- In the time critical and data intensive scenarios, e.g. when nature disaster happens, massive dataset including basic terrain, hydrology, transportation data and real-time observation data need to be gathered for spatial-analysis on-the-fly, the results should be required to decision makers to make sure that rapid response and evacuation plans could be executed(Huang, Cervone, Jing, & Chang, 2015; Wu, Convertino, Ganoe, Carroll, & Zhang, 2013).
- In the cases of mobile working or field investigation, the architecture of system could be distributed: the server side is responsible for data storage and computation, while the tasks of client side for mobile phones and tablets could just be data collection and visualization(Cerón, Fernández-Carmona, Urdiales, & Sandoval, 2018).
- For the scenario of demonstration and education, e.g. for the cases of dashboard system to visualize live stream data and display the patterns of these data, or to educate the usage of very complicated dataset or newly developed data analysis methods through demonstration, the web based application could be the

appropriate choice (Harris, 2003; Purves, Medyckyj-Scott, & Mackaness, 2005; Veenendaal, 2015).

The rapid development of geospatial technologies in recent decades enables scientists to gather massive high-quality georeferenced data from the physical world, society, economy, social-media, web pages, etc. Such data deluge introduces GIScience researchers the great opportunity to obtain a closer and deeper insight into the phenomena happening in nature and human-society. Consequently, the development and achievement of theories, methods, softwares and discoveries are in an accelerating rate driven by the richness of data in the last few decades.

In addition to the big data deluge, the high-performance computing (HPC) theories and technologies have been greatly developed recently, and numerous commercial or academic HPC products and platforms have been widely accepted, such as Amazon Cloud, Microsoft Azure, Google Earth Engine, Hadoop, Apache Spark, NoSQL database, Cloud storage etc. These HPC facilities are capable of hosting big data sets and conducting large scale analysis and simulations which are infeasible on an individual desktop. All these factors together make the second "collaborative" working mode increasingly popular nowadays (Rinner et al., 2008).

Harnessing these open source toolkits on the big data and HPC environment and making them accessible to the "collaborative" working mode could bring immediate benefits to the GIScience community. Nevertheless, most of the aforementioned open source libraries are initiated merely for the desktop environment, instead of the "collaborative" working mode. Developing sophisticated web-based middleware to wrap these libraries and expose their analysis functionalities as geoprocessing services could be a feasible solution. However, four challenges need to be addressed in the integration process:

1) Interoperability between components and services: the deployed toolsets should be compatible with the mainstream software and other services, and meanwhile could be easily exploited by users under the network environment.
2) Provenance and metadata for spatial analytical workflows: this could be one of the most critical factors under the "collaborative" working mode, referring to all the information ranging from how the spatial data is produced, to how the geoprocessing steps are chained and conducted, and to how to obtain the results - the key for quality control and reproduction of geospatial analysis (Luc Anselin & Rey, 2012).
3) Granularity of the functionalities to be exposed as Application Programming Interfaces (APIs): many open source libraries are designed for the "single-user" working mode, in which the functionalities of each method and class are usually designed to be atomic, facilitating users to combine various methods for the exploratory analysis in a flexible manner. However, when deploying the functions on the server side, the communication cost between the client side and the server side needs to be taken account of. The most intuitive way to reduce the communication cost is to combine the atomic APIs into non-atomic ones which accomplish a sophisticated operation by accepting several parameter inputs from users at one shot (e.g. the inference about Local Indicators of Spatial Association (LISAs) (Luc Anselin, 1995)).
4) Documentations and supporting materials: many open source projects serve as a pioneer in implementing and introducing newly developed methodologies of spatial analysis. When deploying these methodologies, how to provide adequate

documentation and materials to educate users to appropriately use the APIs, should be carefully considered as well.

This article addresses these challenges and introduces our research in developing an interoperable and replicable cyberinfrastructure for online spatial-statistical-visual analytics. More specifically, we focus on the widely used open source python library - PySAL, the functions/classes of which are published as geoprocessing services - WebPySAL. Meanwhile, a friendly graphic user interface (GUI) is implemented in a Geospatial CyberInfrastructure named GeoCI. The client side is capable of integrating any open geospatial data shared based on OGC's WFS/WMS standards, and invoking the geoprocessing services from WebPySAL for on-the-fly spatial analysis, which endows great flexibility to users.

The rest of the chapter is organized as follows: Section 3.2 introduces related research in this field, the background of PySAL, Web Processing Service (WPS) - the standard employed in our platform for publishing services, and GeoCI. Section 3.3 introduces the architecture of WebPySAL and a GUI of WebPySAL on GeoCI. How the aforementioned challenges were addressed in our practice will be particularly elucidated. Section 3.4 uses two case studies of exploratory spatial/spatiotemporal data analysis to demonstrate how the server side and the client side could be coordinated to assistant users for accomplishing spatial analytical tasks. We conclude our work with future directions in Section 3.5.

3.2. Related Work and Background

3.2.1 The development of PySAL and its submodules

PySAL is an open source library of spatial analytical functions written in Python intended to support the development of high level applications (Sergio J. Rey & Anselin,

2010). PySAL was initially released in July 2010 and has been continually updated under a 6 month release cycle under the BSD-3 License (Sergio J. Rey et al., 2015). Core team of PySAL is in a vibrant status implementing newly developed or widely adopted geospatial and space-time analytics in PySAL to benefit the scientific community.

Since late 2016, the PySAL team has initialized the code base refactoring process, which aims to reorganize PySAL's functionalities into submodules. Each submodule is/will be released as an independent python package which accomplishes a specific set of spatial analytical tasks. The purpose of the code base refactoring is 1) to better expose the various spatial analytical functionalities of PySAL to the general public, making them clearer and easier to be understood and utilized from a user's perspective; 2) to relieve the developers from the burden of maintaining a giant metapackage as it is much easier to introduce new features to and maintain the much smaller submodules from a developer's perspective. After the refactoring, the submodules (or packages) of PySAL can be roughly classified into four groups:

1. **Lib**: provides core functionality used by other submodules to work with spatial data in Python, including *libpysal*[4];

2. **Explore**: contains exploratory spatial data analysis of clusters, hotspots, and spatial outliers, plus spatial statistics on graphs and point patterns, including *esda*, *giddy*, *pointpats*, *inequality*, *region* and *spaghetti*;

3. **Model**: contains spatial modeling tools including *spreg*, *mgwr*, *spvcm*, *spint*, and *spglm*;

4. **Viz**: provides methods for visualizing spatial datasets as well as the output of spatial statistics, including *mapclassify*, *splot* and *legendgram*.

[4] https://github.com/pysal/libpysal

Many different derivative forms of PySAL's application have been implemented, including desktop applications such as Crime Analytics in Space-Time (CAST), Space-Time Analysis of Regional Systems (STARS) (Sergio J. Rey & Janikas, 2006) and GeoDaSpace (Luc Anselin & Rey, 2014), PySAL toolkits and plugins for Desktop GIS such as ArcMap and QGIS, interactive computing tool such as Jupyter Notebook.

3.2.2 Web Process Service (WPS) standards

The Open Geospatial Consortium (OGC) Web Processing Service (WPS) interface standard provides rules in terms of how to provide inputs (requests) and handle outputs (responses) for geospatial processing services. It defines an interface that facilitates the publishment of geospatial processes from a developer's side, the discovery of and binding to those processes from a client's side, and the invocation and monitor of the geoprocessing APIs. The input/output of a WPS execution can be raster, vector, coverage and/or non-spatial data.

The three most important operations of WPS are:

- *GetCapabilities*: provides a human- and machine-readable *xml* file depicting details of the service, including service metadata and metadata describing the available processes.
- *DescribeProcess*: provides detailed description of the processes available on the service and the definitions of the inputs/outputs of each process.
- *Execute*: the operation to invoke the processes with specified input values and required output data items. The requests are mainly HTTP POST with *xml* request documents, since the requests usually have complex structures.

The WPS standards are widely accepted across the geospatial science community. Many software, libraries, web portals and services adopt the WPS as their geoprocessing standards, such as ArcMap, QGIS, GeoTools, GeoServer, 52° North, and Zoo-Project. To ensure the interoperability between WebPySAL and other existing platforms and

components, the geoprocessing services of WebPySAL are published according to WPS standards as well.

3.2.3 The GeoCI Portal

Initiated in 2012, the GeoCI web portal plays the role of testbed for hosting and demonstrating all the cutting-edge technologies and methodologies developed by our research team. A spatial data search engine is integrated into GeoCI, enabling it to discover a huge number of open geospatial data shared on the Internet. Rich data visualization and exploration functions have been integrated into GeoCI as well. In this article, we will develop several spatial analytical components on GeoCI as study cases. These components will exploit the geoprocessing APIs provided by WebPySAL.

3.2.4 Related works

Coupling spatial analysis models with HPC resources to support collaborative research under the web environment could bring immediate benefits in accelerating solving complex spatial problems and supporting decision making process.

A number of related research and practices have been done recently with different emphasis (Luc Anselin & Rey, 2012). Some of them are dedicated to deploying sophisticated spatial analysis models on a HPC environment to solve specific issues related to hydrology (Rajib et al., 2016), ecology (Dubois, Schulz, Skøien, Bastin, & Peedell, 2013; Sugumaran, Meyer, & Davis, 2009), environment (Delipetrev, Jonoski, & Solomatine, 2014; Swain et al., 2015) and natural disaster (Huang et al., 2015) et al, while others focus on technical solutions such as the design and implementation of GyberInfrastructure (CI) working environment to handle and manipulate big geospatial and conduct analysis and simulations (Astsatryan et al., 2015; Mihon, Colceriu, Bacu, & Gorgan, 2013; Shaowen Wang & Liu, 2009), or the development of the parallel

computing capacity of a HPC environment(Laura, Li, Rey, & Anselin, 2015; F. Z. Wang et al., 2009; S. Wang & Armstrong, 2009).

In this article, we target at the popular and advanced spatial data analysis library – PySAL. We first enable its spatial analysis functionalities under the web environment based on the widely accepted processing API standard, and then seamlessly adopt and integrate these APIs into a GeoCI portal so that the advanced spatial analysis functionalities are combined with abundant geospatial data (as well as time series data) shared on the internet. The deployment strategy and the architecture of various components will make our system extremely interoperable from the users' perspective and extensible from the developers perspective.

3.3. Methodology and System Implementation

3.3.1 The architecture of WebPySAL

The WebPySAL platform is aimed at providing PySAL's core spatial and spatiotemporal analytical functionalities as services on the server side. Figure 10 shows the system architecture of WebPySAL. The classes and functions from PySAL family's submodules including *libpysal*, *pointpats*, *giddy*, *mapclassify* and *esda* are extracted and reorganized as the geoprocessing APIs of WebPySAL. In the following we will expound on how the aforementioned challenges are addressed in the system design and implementation for providing spatial analysis functionalities as services.

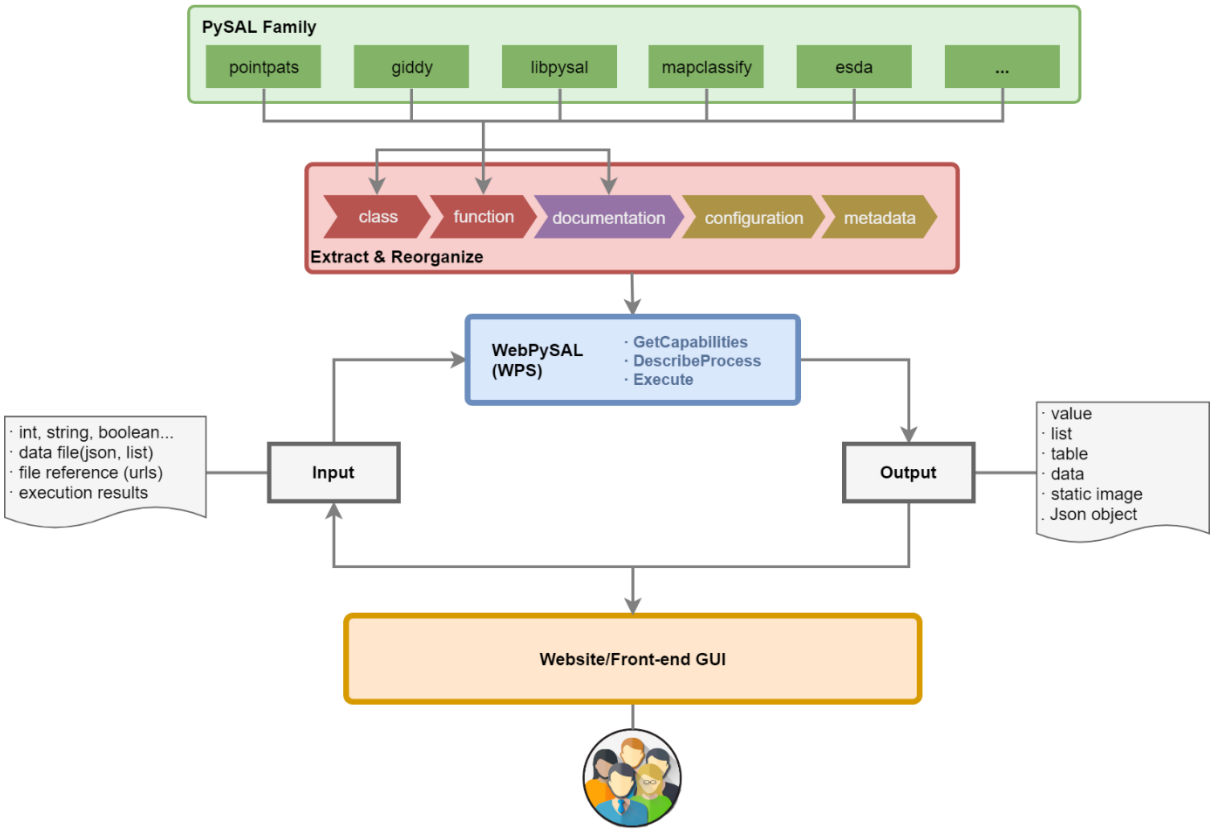


Figure 10 The architecture of WebPySAL

3.3.1.1 Interoperability

The WPS standard is adopted for providing the geoprocessing services which ensures the interoperability between WebPySAL and other existing systems.

The python implementation of the WPS standard - *PyWPS*[5]*(Čepický, 2007)* is employed for the platform development. PyWPS is an open source project for utilizing OGC's WPS standard on the server side. In the implementation, PyWPS acts as the middleware for transforming the functionalities from PySAL into WebPySAL. Each spatial analysis functionality is wrapped into an individual class with predefined inputs and outputs according to the rules of PyWPS. Additional documentation, configuration and metadata are provided to PyWPS as well. Then PyWPS will publish these functionalities as geoprocessing APIs through the WPS standards. During this process, we do not change

[5] http://pywps.org/

the original codebase in PySAL, on purpose of guaranteeing the code consistencies of PySAL on one hand, and facilitating the rapid development of WebPySAL one the other hand.

In WebPySAL, the execution operations are capable of accepting a wide range of inputs, including 1) *literal data* such as numbers, strings, booleans; 2) *complex data* such as GML, JSON, text file, etc; 3) *file references* such as URLs (the system will automatically go fetch the data set according to the URLs on the server side for geoprocessing); and 4) *the result/output of other operations*. This provides the flexibility for users to chain multiple operations together to make up and execute a complex geoprocess task at one time.

Table 4 displays a simplified execution request form for the statistical inference about the widely adopted global spatial autocorrelation statistic - Moran's I on the first column. Four input parameters (highlighted in orange background) are assigned to the API, where the first parameter 'spatial_data' is assigned with a URL reference, which is actually a WFS service. WebPySAL system will download the data set at the backend before executing the process. The second parameter 'weights' is assigned with the result of another execution, that is, constructing a k-nearest-neighbor (KNN, $k = 4$ here) spatial weight matrix (highlighted in green background). Hence the data section for 'weights' is another independent execution request form instead of a value. WebPySAL will execute this process firstly, get the result and use the result as the input of the 'weights' parameter. The third and fourth parameters are assigned with a string and an integer respectively. The result of the processing result is presented on the second column of Table 1.

Table 4 Example WPS POST request for the statistical inference about Moran's I

```
<wps:Execute
xmlns:wps="http://www.opengis.net/wps/1.0.0" …
xmlns:wfs="http://www.opengis.net/wfs" >
  <ows:Identifier>esda:Moran</ows:Identifier>
```

```
{
    "I": {
        "value": 0.45036780970104806,
        "title": "I",
```

```
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>spatial_data</ows:Identifier>
      <wps:Reference
xlink:href="http://cici.lab.asu.edu/geoserver910/wfs?serv
ice=WFS&amp;version=1.1.0&amp;request=GetFeature&amp;type
Name=it.geosolutions%3Aus48&amp;srsName=urn%3Ax-
ogc%3Adef%3Acrs%3AEPSG%3A3857&amp;outputFormat=json"
method="GET" mimeType="application/vnd.geo+json" />
    </wps:Input>
    <wps:Input>
      <ows:Identifier>weights</ows:Identifier>
      <wps:Reference
xlink:href="http://cici.lab.asu.edu:5002/wps"
method="POST" mimeType="application/gal">
        <wps:Body>
          <wps:Execute
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd"
version="1.0.0" service="WPS">
            <ows:Identifier>libpysal:KNN</ows:Identifier>
            <wps:DataInputs>
              <wps:Input>
                <ows:Identifier>data</ows:Identifier>
                <wps:Reference
xlink:href="http://cici.lab.asu.edu/geoserver910/wfs?serv
ice=WFS&amp;version=1.1.0&amp;request=GetFeature&amp;type
Name=it.geosolutions%3Aus48&amp;srsName=urn%3Ax-
ogc%3Adef%3Acrs%3AEPSG%3A3857&amp;outputFormat=json"
method="GET" mimeType="application/vnd.geo+json" />
              </wps:Input>
              <wps:Input>
                <ows:Identifier>k</ows:Identifier>
                <wps:Data>
                  <wps:LiteralData>4</wps:LiteralData>
                </wps:Data>
              </wps:Input>
            </wps:DataInputs>
            <wps:ResponseForm>
              <wps:RawDataOutput
mimeType="application/gal">
                <ows:Identifier>weights</ows:Identifier>
              </wps:RawDataOutput>
            </wps:ResponseForm>
          </wps:Execute>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>column_name</ows:Identifier>
      <wps:Data>
        <wps:LiteralData>y2009</wps:LiteralData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>permutations</ows:Identifier>
      <wps:Data>
        <wps:LiteralData>99</wps:LiteralData>
      </wps:Data>
    </wps:Input>
  </wps:DataInputs>
</wps:Execute>
```


```
        "abstract": "value of Moran's
I"
    },
    "EI": {
        "value": -0.02127659574468085,
        "title": "EI",
        "abstract": "expected value
under normality assumption"
    },
    "VI_norm": {
        "value": 0.008391070042774918,
        "title": "VI_norm",
        "abstract": "variance of I
under normality assumption"
    },
    "seI_norm": {
        "value": 0.09160278403397419,
        "title": "seI_norm",
        "abstract": "standard deviation
of I under normality assumption"
    },
    "z_norm": {
        "value": 5.148799901876373,
        "title": "z_norm",
        "abstract": "z-value of I under
normality assumption"
    },
    "p_norm": {
        "value": 2.621583647943737e-7,
        "title": "p_norm",
        "abstract": "p-value of I under
normality assumption"
    },
    "VI_rand": {
        "value": 0.006250746750777324,
        "title": "VI_rand",
        "abstract": "variance of I
under randomization assumption"
    },
    "seI_rand": {
        "value": 0.07906166422974743,
        "title": "seI_rand",
        "abstract": "standard deviation
of I under randomization assumption"
    },
    "z_rand": {
        "value": 5.9655258972941,
        "title": "z_rand",
        "abstract": "z-value of I under
randomization assumption"
    },
    "p_rand": {
        "value": 2.4384736452276456e-9,
        "title": "p_rand",
        "abstract": "p-value of I under
randomization assumption"
    }
}
```

3.3.1.2 Provenance and metadata

Tracking the provenance of operations is a key factor of guaranteeing analysis result quality and ensuring the full replicability of data analysis and interoperability with other systems, which are critical under the increasingly popular collaboration context nowadays (Luc Anselin, Rey, & Li, 2014). Two strategies for keeping the provenance are adopted in WebPySAL's implementation: geoprocessing API version and execution form. In WebPySAL, since each geoprocessing API wraps some specific functionalities from the submodules of PySAL, the development version of these submodules will be automatically extracted and used by WebPySAL. In terms of the open source libraries which are developed and upgraded rapidly, this strategies can help users get a better sense about which version of libraries they are using and whether they can obtain identical results to the older versions. At the time of writing, the version of submodules integrated into WebPySAL are *libpysal 3.0.5*[6], *esda 1.0.1.dev0*[7], *giddy 1.1.1*[8], *pointpats 1.1.0*[9], and *mapclassify 1.0.1*[10].

The API description form of WebPySAL contains the version info and all the essential parameters needed to execute the API. After specific parameters and configurations are provided from user side, they will be injected into the execution request form and submitted to the server side to initialize the analysis process. These forms are in XML format, which are designed to be both human- and machine- readable. Properly saving all the relevant metadata could guarantee the provenance of an geoprocessing execution, so that users can replicate the process anytime later to get the identical results.

---

[6] https://pypi.org/project/libpysal/3.0.5/

[7] https://pypi.org/project/esda/1.0.1.dev0/

[8] https://pypi.org/project/giddy/1.1.1/

[9] https://pypi.org/project/pointpats/1.1.0/

[10] https://pypi.org/project/mapclassify/1.0.1/

Let's take the spatial weight construction, which is essential to many spatial analytical tasks, as an example. WebPySAL provides 6 different types of spatial weights, which are distinguished by the identifier of the execution. Table 5 shows the API description form for KNN spatial weight construction, which is extracted from *libpysal 3.0.5*. The requirements for the input parameter "Data" as well as two optional input parameters "Number of nearest neighbors" and "Id Variable" with default values are listed in the description form. In the execution form (Table 6), the input geometry data is provided to get the result weights. Once storing this execution form, users can re-submit it anytime later to get the identical results.

Table 5 Example API description form for KNN spatial weight construction

```
<wps:ProcessDescriptions … service="WPS" version="1.0.0" xml:lang="en-US">
  <ProcessDescription wps:processVersion="3.0.5" storeSupported="true"
statusSupported="true">
    <ows:Identifier>libpysal:KNN</ows:Identifier>
    <ows:Title>K Nearest Neighbor Weights Calculation</ows:Title>
    <ows:Abstract>Calculate the KNN weights object from a collection of geometries. Classes
and functions used in this API include libpysal.weights.Distance.KNN. For more information,
see the metadata</ows:Abstract>
    <ows:Metadata xlink:title="KNN"
xlink:href="https://github.com/pysal/libpysal/blob/master/libpysal/weights/Distance.py"
xlink:type="class"/>
    <DataInputs>
      <Input minOccurs="1" maxOccurs="1">
        <ows:Identifier>data</ows:Identifier>
        <ows:Title>Data</ows:Title>
        <ComplexData>
          <Default>
            <Format><MimeType>application/vnd.geo+json</MimeType></Format>
          </Default>
          <Supported>
            <Format><MimeType>application/vnd.geo+json</MimeType><Format>
            <Format><MimeType>application/gml+xml</MimeType><Format>
          </Supported>
        </ComplexData>
      </Input>
      <Input minOccurs="0" maxOccurs="1">
        <ows:Identifier>k</ows:Identifier>
        <ows:Title>Number of nearest neighbors</ows:Title>
        <ows:Abstract>Number of nearest neighbors for querying, default value is
2</ows:Abstract>
        <LiteralData>
          <ows:DataType
ows:reference="urn:ogc:def:dataType:OGC:1.1:integer">integer</ows:DataType>
          <ows:AnyValue/>
        </LiteralData>
      </Input>
      <Input minOccurs="0" maxOccurs="1">
        <ows:Identifier>idVariable</ows:Identifier>
        <ows:Title>Id Variable</ows:Title>
        <ows:Abstract>The name of the column to use as IDs. If nothing is provided, the
dataframe index is used. (Note: the ids should be unique and Integer type is
preferred.)</ows:Abstract>
```

```
          <LiteralData>
            <ows:DataType
 ows:reference="urn:ogc:def:dataType:OGC:1.1:string">string</ows:DataType>
            <ows:AnyValue/>
          </LiteralData>
        </Input>
      </DataInputs>
      <ProcessOutputs>
        <Output>
          <ows:Identifier>weights</ows:Identifier>
          <ows:Title>Result Bundle</ows:Title>
          <ows:Abstract>The calculated weights by using this method.</ows:Abstract>
          <ComplexOutput>
            <Default>
              <Format><MimeType>application/json</MimeType></Format>
            </Default>
            <Supported>
              <Format><MimeType>application/json</MimeType></Format>
              <Format><MimeType>application/gal</MimeType></Format>
              <Format><MimeType>application/gwt</MimeType></Format>
              <Format><MimeType>application/swm</MimeType></Format>
            </Supported>
          </ComplexOutput>
        </Output>
      </ProcessOutputs>
    </ProcessDescription>
  </wps:ProcessDescriptions>
```

Table 6 Example API execution form for KNN spatial weight construction

```
<wps:Execute … version="1.0.0" service="WPS">
    <ows:Identifier>libpysal:KNN</ows:Identifier>
    <wps:DataInputs>
        <wps:Input>
            <ows:Identifier>data</ows:Identifier>
            <wps:Reference mimeType="application/vnd.geo+json" xlink:href=
"http://sedac.ciesin.columbia.edu/geoserver/wfs?service=WFS&amp;version=1.1.0&amp;request=GetFeatu
re&amp;typeName=epi%3Aepi-environmental-performance-index-2010_climate-change&amp;srsName=urn%3Ax-
ogc%3Adef%3Acrs%3AEPSG%3A4326&amp;outputFormat=application%2Fjson" method="GET"/>
        </wps:Input>
    </wps:DataInputs>
</wps:Execute>
```

3.3.1.3 Abstraction and aggregation of PySAL functions to provide synthetical APIs

PySAL was originally designed for the desktop working environment. During the implementation, the object-oriented strategy is adopted meaning that class objects are widely used for hosting analysis functions and relevant variables. When users are exploring the library under the desktop environment (e.g. in a Jupyter notebook), intermediate results such as class instances and variables can be easily stored in the RAM and re-used for the next-step analysis. Consequently, it is appropriate to make each method atomic which is only responsible for performing a single task since this enable

users flexibly combine different methods in the exploratory analysis. Nevertheless, WebPySAL will be mainly used under the internet working environment, interacting and transforming data between the server and client side will be much more time consuming than under a local environment, and this will also bring more burden to the UI design on the client side. Therefore, for WebPySAL, the data transmitted between the server and client sides via network should not be too fragmented and the interaction should not be too frequent. During the implementation of WebPySAL, we adopt a "synthetical" strategy which enables each WebPySAL API to take combined input parameters, conduct the whole geoprocessing workflow and return complete results that can be directly used for visualization and interpretation on the client side.

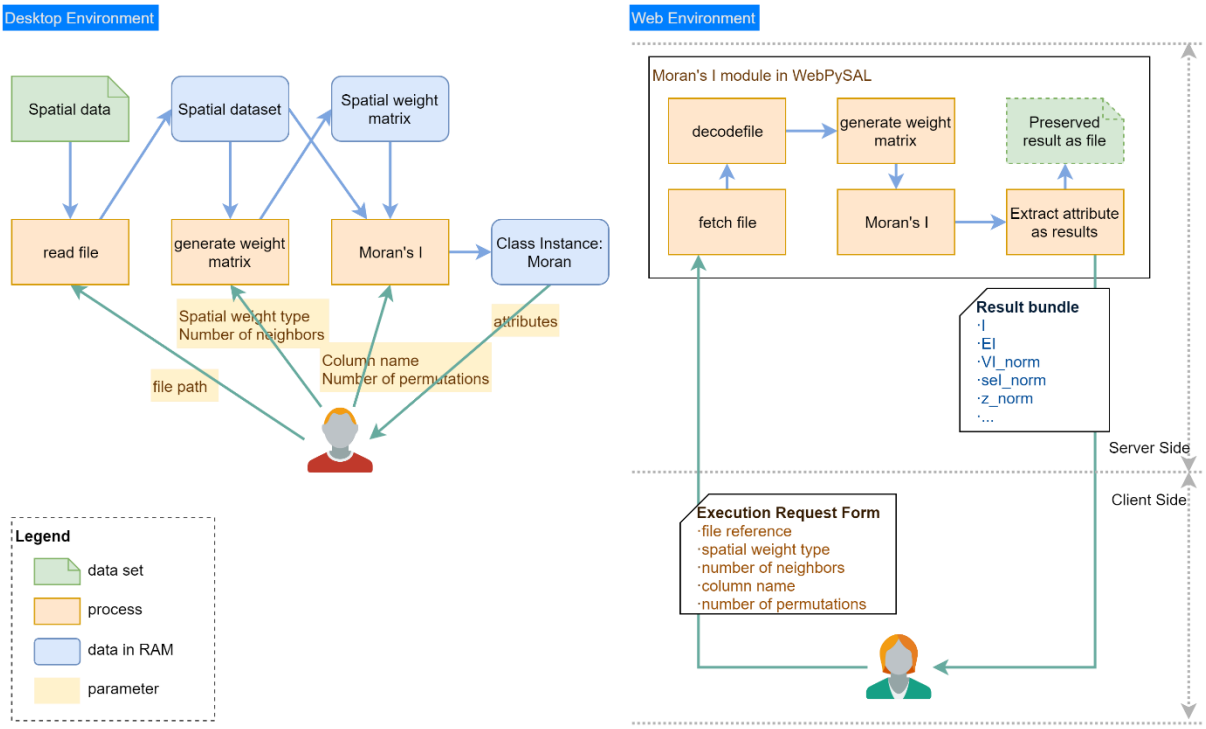


Figure 11 Comparison of the interaction modes with PySAL and WebPySAL under the desktop environment vs. web environment

Figure 11 illustrates the difference of interaction modes with PySAL and WebPySAL for calculating the Moran's I statistic. Under the desktop environment, the user needs to invoke three functions sequentially in order to read the geospatial file, generate weight

matrix, and initialize the Moran's I class. Different parameters should be provided to these functions separately during the process. Results concerning Moran's I are assigned to the Moran object as attributes. All the intermediate results are temporarily stored in the local computer's memory for quick access. Under the web environment, the "synthetical" API takes the inputs of all the parameters needed to produce the final results at one shot. After the parameter inputs are submitted through the execution request form to the server side, they will be assigned to atomic functions separately to execute the process chain. When the process is finished, the resulted attributes will be extracted and injected into the result bundle (usually a JSON object) and returned to the user or stored on the server side as files for later access. After the results are returned, the memory for preserving the intermediate results will be freed on the server side. From the graph we can see that the user only needs to interact with WebPySAL once for invoking the API and she/he still has the flexibility of providing different values of parameters.

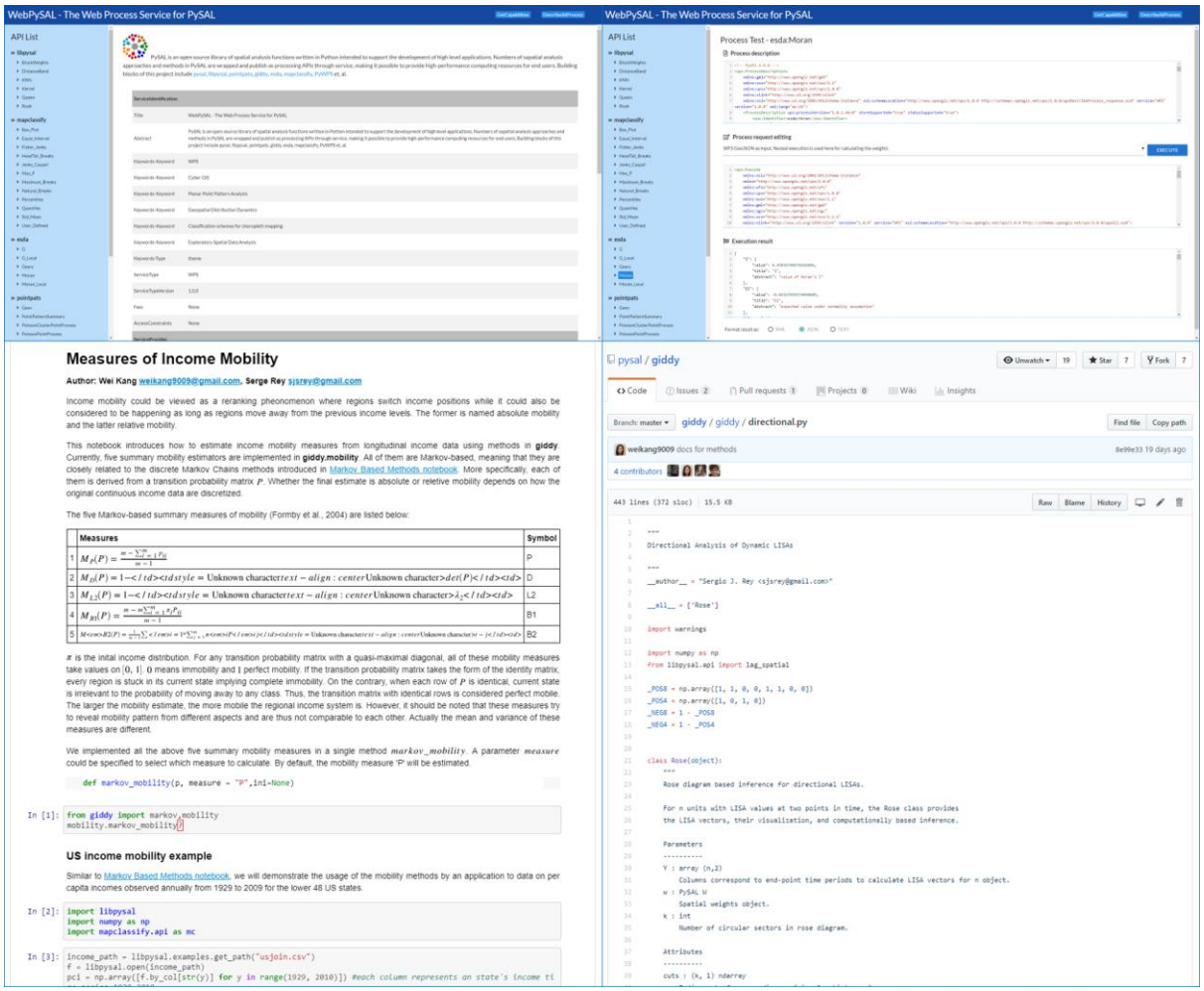


Figure 12 WebPySAL demonstrations

3.3.1.4 Documentations and supporting materials

Adequate documentation and materials are highly necessary to educate users appropriately take advantage the APIs and interpret the geoprocessing results, especially for the open source libraries which release frequently and continuously introduce new functionalities. In WebPySAL, the documentation and supporting materials are provided to users through the following 3 approaches.

- There are many use case demonstration pages created for the functionalities of PySAL. When these functionalities are wrapped into WebPySAL, the URLs pointing to the demonstration page and source code will be automatically injected into the metadata sections of each API.

- Adequate description information is presented as abstraction sections for WebPySAL's APIs, the input parameters of each API and the result variables of each output JSON object.
- A GUI portal of the WebPySAL is implemented. In the portal, metadata about WebPySAL is presented, all the APIs are listed. For each API, there is at least one execution request example to demonstrate how to invoke the API.

The GUI and documentation of WebPySAL are presented in Figure 12.

3.3.2 Implementation of spatial analysis modules in GeoCI

Nowadays, a large number of organizations are collecting and sharing geospatial datasets on the Internet through OGC's WFS and WMS standards for public and scientific use. In our previous work, we developed a geospatial data discovery engine named PolarHub, which is capable to collect hundreds of thousands of geospatial dataset's metadata information. The metadata information is stored in a relationship database and integrated into GeoCI's system. An geospatial data search engine is implemented in GeoCI to help users conveniently find their desired datasets by using keywords and/or spatial extent filtering. The selected datasets can be easily included into GeoCI under user's account for later visualization and analysis.

WebPySAL's geoprocessing APIs have been fully integrated into GeoCI. Specific exploratory data analysis modules are designed and implemented to help users take advantage of WebPySAL's spatial analysis models (SAM) and functionalities. The architecture of the integrated systems is presented in Figure 13. The data analysis modules/functions include *esda* (exploratory spatial data analysis), Rose (directional analysis of dynamic LISAs in *giddy*), Markov analysis (spatially explicit Markov methods in *giddy*), and Rank Based Analysis (rank based methods in *giddy*).

A typical working flow for a user in GeoCI is as follows:1) Browse the datasets in GeoCI, select the interested ones which are stored in GeoCI under specific workspace; 2) Select a spatial analysis module in GeoCI, provide the spatial dataset and other input parameters in the GUI; 3) Invoke the geopossing API in WebPySAL and obtain the results; 4) Visualize and demonstrate the spatial analysis results through interactive maps, graphic charts and reports and meanwhile store the provenance information for later use.

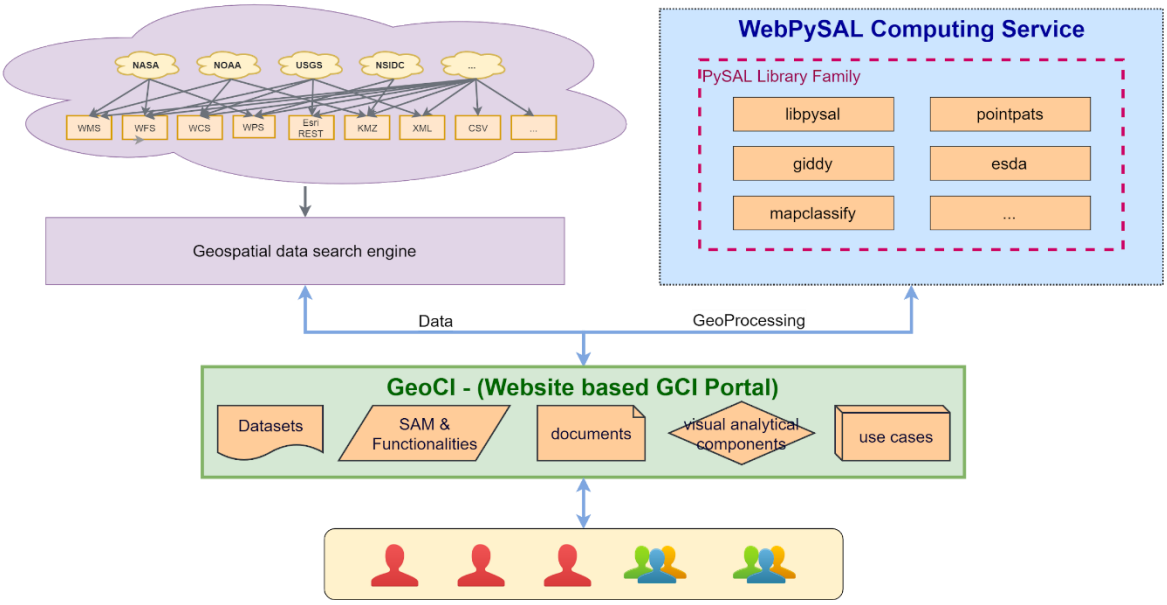


Figure 13 The architecture of GeoCI

Figure 14 illustrates the user interface implemented in GeoCI for setting parameters for the Markov analysis module. The abstract description of the module is presented below the analysis method, followed by the metadata tags. Each of the tags are URLs pointing to the original documentation of PySAL library. There are four parameters requested for this analysis method. Those starting with star (*Time Periods Data, *The name of columns as input) are required. The rest are optional meaning that a default value will be supplied if the user leaves them blank. After setting the parameters, the user can click the EXECUTE button to trigger the execution. Results of the reports and charts will be automatically appended below the EXECUTE button after the calculation is finished.

While newly calculated variables of the geometries will be attached to the spatial dataset for later visualization. The execution form been submitted can be opened by clicking the button at the upper-right corner for viewing and later reusing.

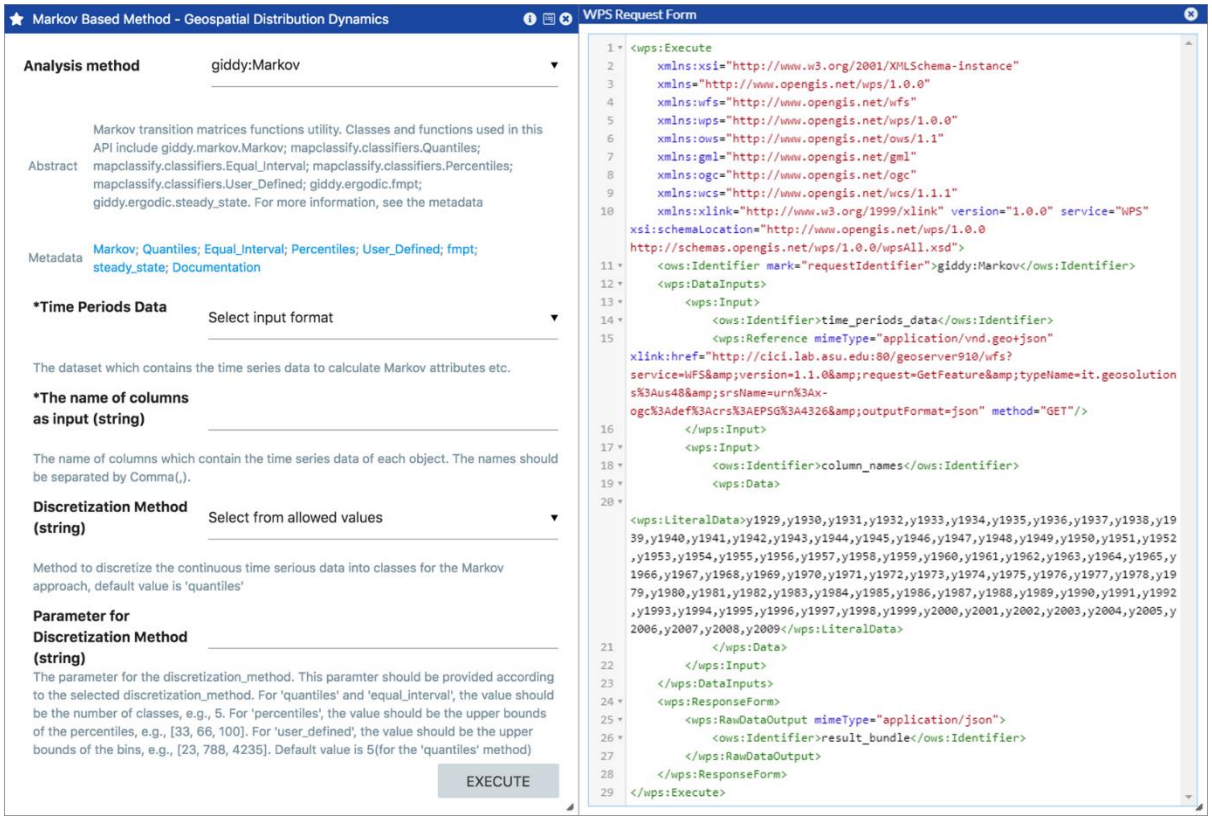


Figure 14 Graphic user interface for the Markov chain analysis module

3.4. Illustration and Experiments on Spatial & Spatiotemporal Statistics

In this section, we use two case studies to illustrate how GeoCI and WebPySAL are tightly coupled to help users fulfil spatial analytical tasks with visual aid in a convenient and efficient fashion. Both cases utilize Exploratory Spatial Data Analysis (ESDA) methods which is an extension to Exploratory Data Analysis (EDA) to uncover underlying structures in spatial data. EDA is a concept proposed 40 years ago which postpones assumptions about the underlying theory/model followed by the data with a wide array of quantitative methods and statistical graphics (Tukey, 1977). ESDA extends

EDA to incorporate spatial attributes (location). While the first case represents a general first step in exploring global and local spatial patterns of lattice data at a time point, the second case explores the role of space in shaping the evolution of a variable over time.

3.4.1 Global and Local indicators of spatial association

Global and local indicators of spatial association are the most important tool for exploring the spatial distribution of a given variable at a time point. Both pertain to the question of spatial randomness by examining whether or to what degree location similarity and attribute similarity coincide. While the global indicators operate at the global level, meaning that a single summary statistic is produced, the local indicators operate at the local level by decomposing the global ones to provide insights in the local patterns such as hot and cold spots, as well as the instability of spatial associations (Luc Anselin, 1995).

The PySAL submodule *esda* implements a wide array of global indicators including Moran's I, Geary's c, Getis-Ord G and join count statistics together with their respective local decompositions. All of them have also been integrated in WebPySAL and GeoCI. Here, we detail the usage of Moran's I and local Moran's I which are the most widely used in empirical settings as an illustration.

Given n spatial observations with attribute y, the global indicator of spatial association, Moran's I (Cliff & Ord, 1981), is defined in Equation (1):

$$I = \frac{n}{S_0} \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} z_i w_{i,j} z_j}{\sum_{i=1}^{n} z_i z_j} \quad (1)$$

where $z_i = y_i - \overline{y}$ is the deviation from the global mean, W is the (n, n) spatial weight matrix formalizing the spatial relationship between any pair of spatial units and $S_0 = \sum_{i=1}^{n} \sum_{j=1}^{n} w_{i,j}$. Inference could be made under the normality assumption or based on spatial permutations. For the proper estimation and inference of this statistic, it is required that the user supplies the attribute, the spatial weight matrix, and the number

of permutations if the randomization-based inference is desired. We shall see that WebPySAL provide options for setting these parameters in a convenient fashion.

Local Moran's I is a spatial decomposition of Moran's I shown in Equation (2) which has a value for each spatial unit. As suggested by (Luc Anselin, 1995), a pseudo p-value could be obtained for I_i based on conditional randomization. The required parameters are similar to the global indicator.

$$I_i = \frac{(n-1)z_i \sum_{j=1}^{n} w_{i,j} z_j}{\sum_{j=1}^{n} z_j^2} \quad (2)$$

3.4.1.1 Data

We applied the global and local indicators of spatial association to the U.S. county average median household incomes in 2016. The county boundaries were acquired from U.S. Census Bureau's MAF/TIGER geographic database[11] and the "Unemployment and median household income for the U.S., States, and counties, 2007-17" which included the county-level median household incomes 2016 were downloaded from the U.S. Department of Agriculture (USDA)'s website[12]. These two datasets are joined and hosted on our testbed as a standard WFS data service for public use[13]. The spatial distribution of the median household incomes can be conveniently visualized in GeoCI as shown in Figure 15. It seems that similar values tend to be geographically closer to each other.

[11] https://www.census.gov/geo/maps-data/data/cbf/cbf_counties.html (data accessed by Aug/09/2018)

[12] https://www.ers.usda.gov/data-products/county-level-data-sets/download-data/ (data accessed by Aug/09/2018)

[13] http://cici.lab.asu.edu/geoserver910/wfs?service=WFS&version=1.1.0&request=GetCapabilities

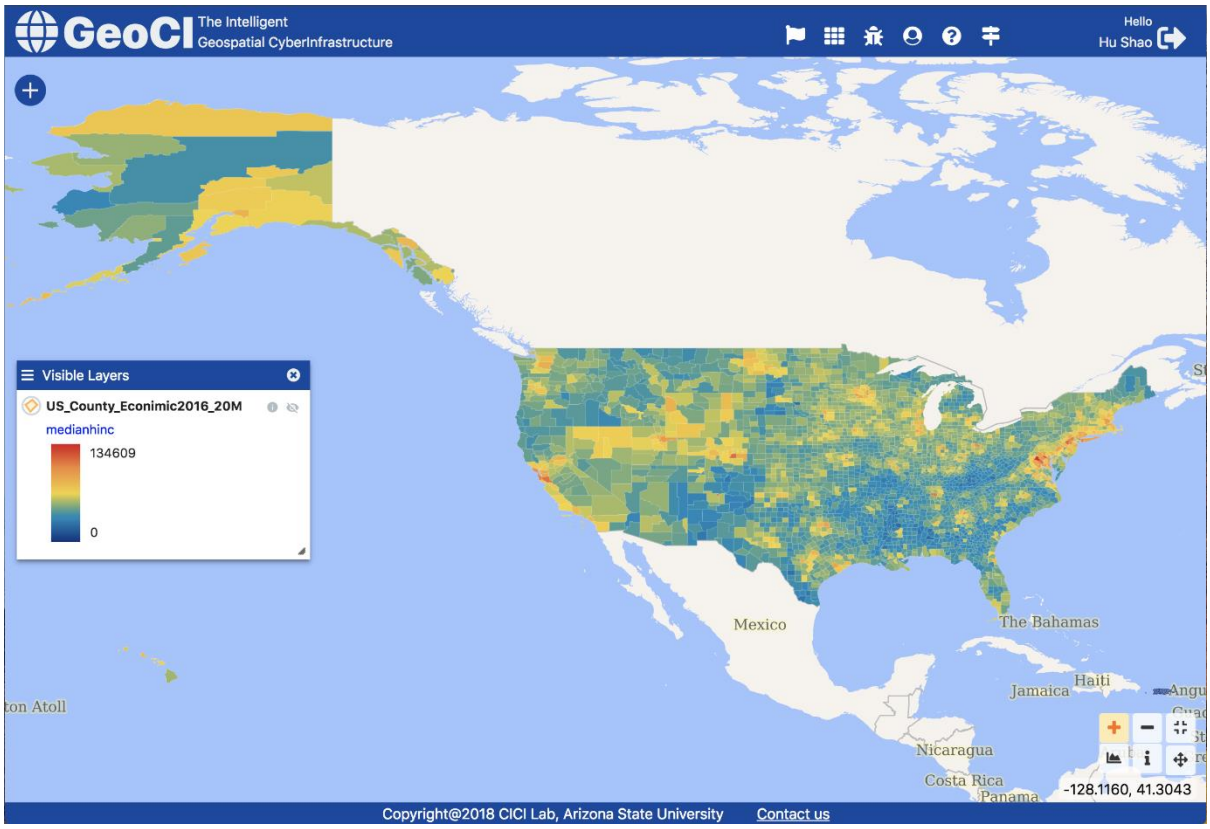
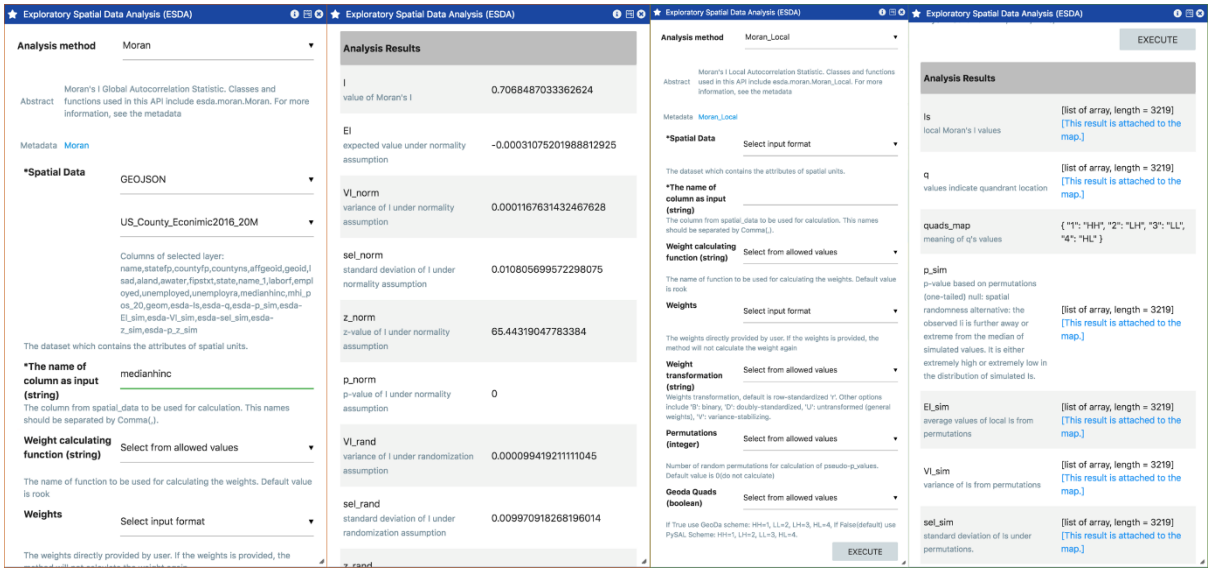


Figure 15 Map of the U.S. county-level median household incomes in 2016

3.4.1.2 Empirical Results and Visualization

Global and local Moran's Is are applied to the U.S. 2016 county-level median household incomes to explore its spatial distribution, or more specifically, whether the observed incomes are spatially random and whether there are hot spots of high incomes or cold spots of low incomes which deserves further investigation. We start with global Moran's I. As displayed in the left of Figure 16(a), GeoCI provides a GUI for selecting values for all the relevant parameters. There are two ways to specify the spatial weight matrix W: choose a weight type (queen/rook contiguity, KNN, etc) so that a spatial weight matrix is constructed for the GEOJSON geometries using functions in *libpysal*, or supply a spatial weight file. Users also have the option to leave them blank so that the default value is used which builds a row-normalized rook contiguity weight matrix where spatial units sharing an edge are considered neighbors. Here, we use the default value for the spatial

weight. 999 permutations are selected for randomization-based inference. The same values are selected for the inference about the local Moran's Is as shown in the left of Figure 16(b).



(a) (b)

Figure 16 Moran's I and Local Moran's I in WebPySAL and GeoCI

Results about Moran's I will be appended to the analysis method window once the calculation is completed (Figure 16 (a)). The visual impression of spatial clustering of similar values is confirmed by the positive and significant Moran's I of 0.707 with p-value of 0 under the normality assumption and pseudo p-value of 0 based on the 999 spatial permutations. Since results about local Moran's Is are almost always n-dimensional (a list of results are shown in Figure 16(b)), they are appended to the original data set to facilitate geovisualization in GeoCI. Figure 17 displays the spatial clusters of cold spots (low-low) and hot spots (high-high) of county-level mean household incomes.

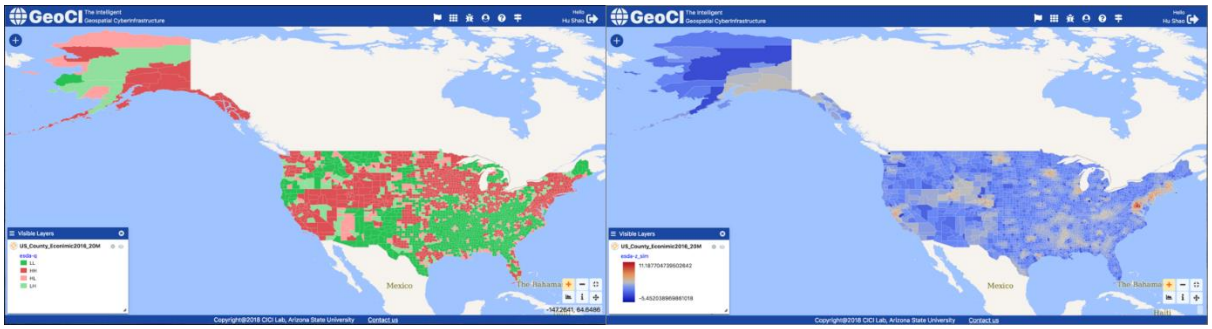


Figure 17 Visualization of Local Moran's Is in GeoCI

3.4.2 Spatial Markov Tests

The first-order discrete Markov chains model is a widely used stochastic model in which the current status is only dependent on its status at the immediately preceding time period. It has been widely applied to provide insights into the underlying dynamics of land use and land cover change, crime patterns and income distribution dynamics (McMillen & McDonald, 1991; Quah, 1993; Sergio J. Rey et al., 2014). By further assuming time homogeneity, the transitional dynamics for the whole study time span could be summarized in a (k, k) stochastic matrix P in which each element $p_{i,j}$ presents the probability of transitioning from state i to j over two consecutive time periods. The maximum likelihood estimator $\hat{p}_{ij}$ is displayed in Equation (3):

$$\hat{p}_{ij} = \frac{n_{ij}}{\sum_{j=1}^{k} n_{ij}} \quad (3)$$

where n_{ij} is the number of transitions from state i to j across two consecutive time periods. The conventional application of the Markov chains model to a spatial setting assumes that the dynamics are identical across all spatial units. Thus, P is estimated from the pooled data. However, the ignorance of space in shaping the dynamics could lead to false conclusions. The spatial Markov tests which tests for spatial dependence in the discrete Markov chains framework have been proposed and their properties have been evaluated for the study of regional income distribution dynamics (Bickenbach &

Bode, 2003; Kang & Rey, 2018; Sergio J. Rey, Kang, & Wolf, 2016; S. J. Rey, 2001). The alternative of spatial Markov tests contends that the underlying dynamics is too complex to be summarized in a single transition probability matrix. Rather, the transition probability is context-sensitive in that it is also dependent on the current status of neighbors. The so-called spatial lag shown which is the weighted average of neighbors' values (e.g. income) in Equation (4) is usually used to quantify neighbors' status:

$$z_t = W y_t (4)$$

where z_t is the n-dimensional spatial lag at t. Following the similar discretization strategy to the original time series, the time series of spatial lags could also be discretized into k categories on which transition probabilities are conditional, resulting in k spatially dependent transition probability matrices. The likelihood ratio (LR), χ^2 and Kullback information-based (Kullback, Kupperman, & Ku, 1962) tests can be formed by comparing them with the single matrix estimated from the pooled data.

To conduct a spatial Markov test, the longitudinal data, the spatial weight matrix, and the quantile number (for discretization) k are required. We shall see how WebPySAL and GeoCI provide convenient interface for the user to setting the parameters.

3.4.2.1 Data

The average per capita incomes for the lower 48 U.S. states from year 1929 to 2009 are used for demonstration. The data set was acquired from Bureau of Economic Analysis, U.S. Department of Commerce. The state-level cartographic boundary data was downloaded from United States Census Bureau's MAF/TIGER geographic database. These two datasets are bound together and hosted on our testbed as a standard WFS data service. The map of U.S. state per capita incomes in 2009 can be easily visualized in GeoCI. We can also interactively explore the time dimension with the help of the time series plot shown in

Figure 18: as the user move the vertical dotted line in the time series plot, the map on the right will be updated to the chosen year (e.g. 1973) and the colors of the time series plot will be updated to match the color scheme of the map.

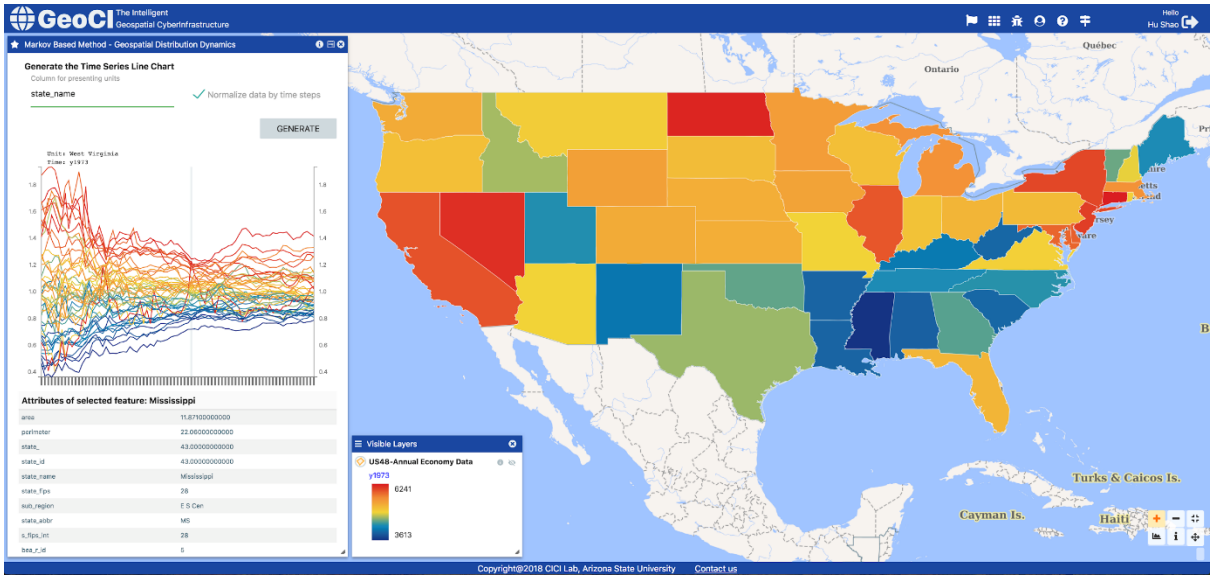
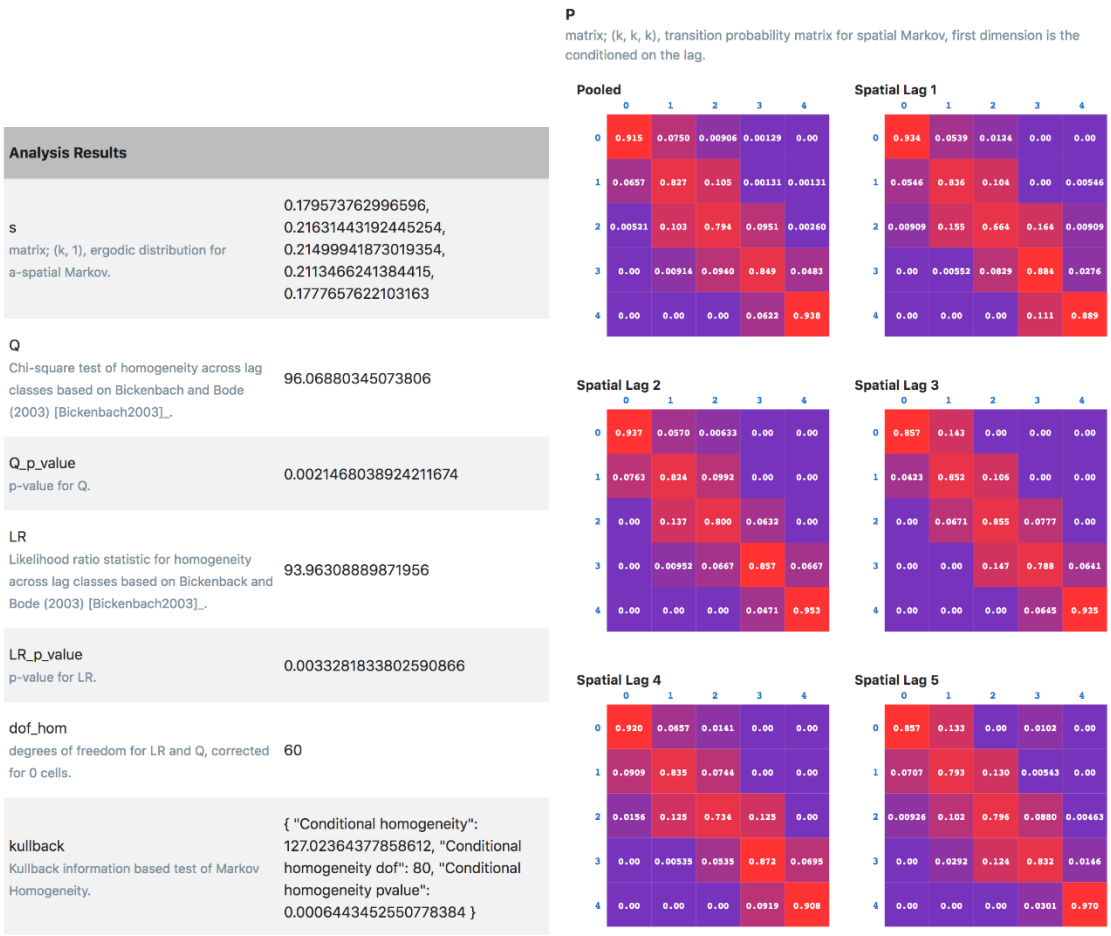


Figure 18 Interactive visualization of average per capita income series for the lower 48 U.S states 1929-2009



Analysis Results	
s matrix; (k, 1), ergodic distribution for a-spatial Markov.	0.179573762996596, 0.21631443192445254, 0.21499941873019354, 0.2113466241384415, 0.1777657622103163
Q Chi-square test of homogeneity across lag classes based on Bickenbach and Bode (2003) [Bickenbach2003]_.	96.06880345073806
Q_p_value p-value for Q.	0.0021468038924211674
LR Likelihood ratio statistic for homogeneity across lag classes based on Bickenback and Bode (2003) [Bickenbach2003]_.	93.96308889871956
LR_p_value p-value for LR.	0.0033281833802590866
dof_hom degrees of freedom for LR and Q, corrected for 0 cells.	60
kullback Kullback information based test of Markov Homogeneity.	{ "Conditional homogeneity": 127.02364377858612, "Conditional homogeneity dof": 80, "Conditional homogeneity pvalue": 0.0006443452550778384 }

Figure 19 Output of Spatial Markov Tests

3.4.2.2 Empirical Results and Visualization

The default value for the discretization, year-specific quintiles, are used as the cutoffs to discretize the continuous per capita incomes and their spatial lags, giving rise to a (5,5) transition probability matrix under the null of spatial randomness of dynamics and 5 (5,5) transition probability matrices under the alternative of spatially dependent dynamics.

Analysis results are appended to the interface of the analysis method once the calculation is completed. We display part of the results here for illustration purpose. As shown in the right of Figure 19, the transition probability matrix estimated from the pooled data and the 5 matrices estimated from the spatial lag - conditioned subsamples are visualized

where red represents high probability and purple low probability. As “Spatial Lag 1” is the first quintile (poor) of the spatial lags, the probability of staying poor is 0.934, which is higher than 0.857 where the neighbors are rich (“Spatial Lag 5”). All of the three test statistics are strongly significant (LR test: 93.96 (p-value: 0.003), χ^2 test: 96.07 (p-value: 0.002), Kullback test: 127.01 (p-value: 0.0006)), confirming the role of space in shaping the U.S. state per capita income dynamics. This could also have important regional policy implications.

3.4.3 Comparison of Computational Time between WebPySAL and PySAL

Compared with the desktop-based data analysis working mode, there is an overhead of communication time between server side and client side when conducting the analysis on WebPySAL. In this section, we conduct some experiments to see if the overhead of communication time will significantly affect WebPySAL’s performance in terms of computational time.

We conducted a series of experiments to compare the performance under different working environments: the desktop-based PySAL against server-client WebPySAL. The variations of the experiment include: 1. Different analysis methods: Local Moran’s I and interregional and intraregional indicators of exchange mobility – the inter-and intra-regional Tau statistics (Rey, 2016); 2. Different datasets: a dataset of 48 U.S. states and a dataset of 3,141 U.S. counties; 3. Different numbers of permutations for the inference: [99, 499, 999].

The performance of the experiments is obtained under the computing environment as follows: the WebPySAL is hosted on a server machine with two 12-core 2.1 GHz 64-bit Xeon CPUs and 64 GB RAM running Ubuntu 16.04.4. The client side is tested on a laptop machine with a 4-core 2.50 GHz 64-bit Intel i-7 CPU and 8GB RAM running Windows 10. The Internet speed environment for experiment is relatively high (50Mbps).

The geospatial datasets used for PySAL are stored locally in the same laptop, while the datasets for WebPySAL are provided as a WFS service hosted on the same server.

A series of tests are conducted with the combinations of different methods, datasets and simulation times. For the PySAL calculation, since the data loading time is very short, we only record the total time in each calculation. For WebPySAL, we record 1. the total calculation time and 2. the time used for communication and data transmission. Figure 20 presents the comparison results. The orange solid lines represent time consumption in PySAL, the blue solid lines and dash lines represent total calculation time and communication time respectively in WebPySAL. From the graphs we can find that:

- For the small state-based dataset (the first column), all the calculation can be finished very quickly within 1 second. Hence the differences of time consumption won't be noticed by users.
- When calculating Local Moran's I with the large county-based dataset (top-right), the total time cost on WebPySAL is a little bit longer, which are mainly resulted from the communication between the server and client sides.
- When calculating the more complex Regional Tau with county-based data, since the time been used for simulation is very long, the communication time is negligible in such cases. To be noted, since the Numpy library is adopted for the matrix calculations, which is parallelized, hence, the calculation time will be much shorter on a powerful machine.

In summary, for all the experiments, the communication time is stable and relatively short (contingent on the data size) while the calculation time in WebPySAL environment will be much shorter due to its high computing performance. Hence, putting all factors together, we can see that WebPySAL could be a feasible solution for handling complex computing tasks.

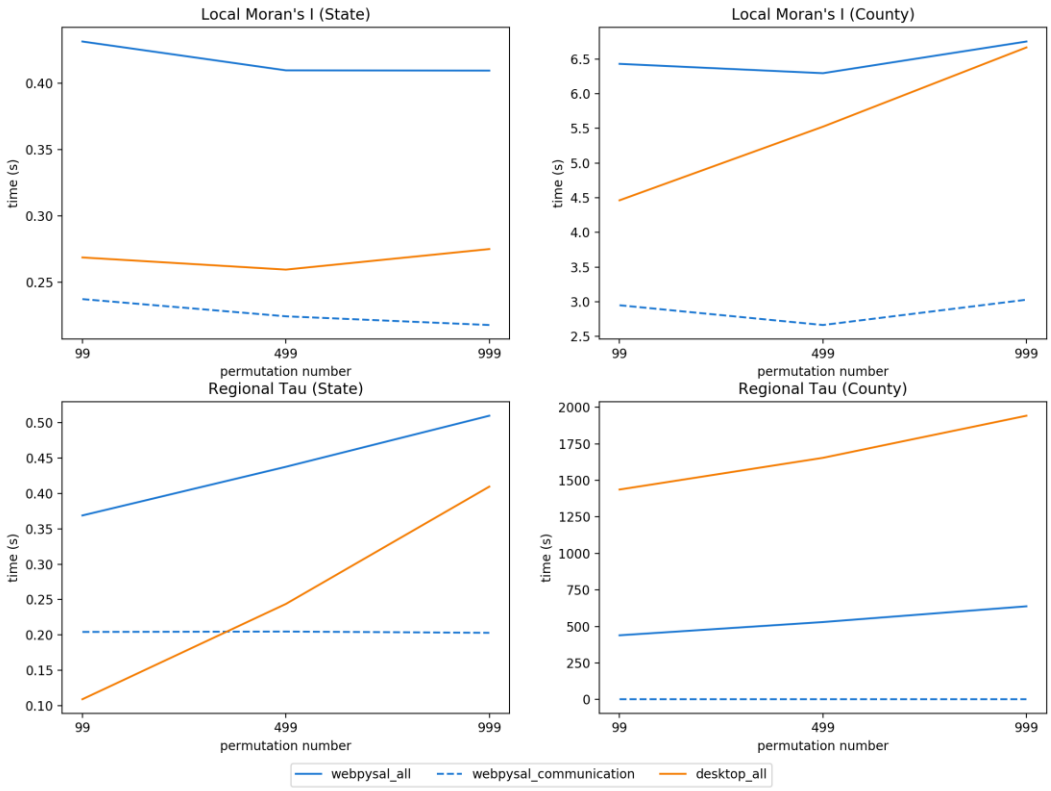


Figure 20 Comparison of time consuming on PySAL against WebPySAL in different experiments

3.5. Discussion and Conclusion

Nowadays, there are a number of vibrant teams focusing on introducing and implementing newly developed and advanced spatial analysis methodologies in open source libraries, which contribute a lot to the GIScience field and related disciplines. This article introduces our research in designing and implementing an interoperable and replicable cyberinfrastructure for online spatial-statistical-visual analytics - WebPySAL based on the popular open source library - PySAL. Many popular and advanced spatial analysis functionalities are provided through the standard WPS APIs. A CyberGIS portal - GeoCI - is bridged with WebPySAL in order to harness the spatial analysis modules

with massive geospatial data and the HPC environment to address the research challenges in the real world.

The contributions of our research include: 1) Established the WebPySAL as a working instance instead of a prototype to benefit the GIScience community; 2) Presented strategies and methodologies about how to guarantee the interoperability and replicability in the practice of implementing a standard geospatial web processing service; 3) Implemented an interactive and user-friendly GUI in our web portal GeoCI to assist users in conducting exploratory spatial/spatiotemporal data analysis with massive open access geospatial data sets. In addition to potential benefits this work brings by bridging spatial analysis toolkits with CyberInfrastructure, the design and implementation of this system could potentially help users who are lack of GIScience background knowledge or programming skills to better understand and adopt advanced spatial analytical methodologies.

The WebPySAL will be published as a member of PySAL's family on GitHub[14], and the integration work of PySAL's advanced spatial analysis functionalities will be continued. An active instance of WebPySAL is currently available at http://cici.lab.asu.edu:5002. Parallel spatial analysis modules will be integrated into WebPySAL to leverage the HPC resources in CyberInfrastructure to help solve more challenging tasks in the future.

[14] https://github.com/pysal

4 A COMPREHENSIVE OPTIMIZATION STRATEGY FOR REAL-TIME SPATIAL FEATURE SHARING AND VISUAL ANALYTICS IN CYBERINFRASTRUCTURE

4.1 Introduction

With the advancement of Earth Observation (EO) technologies, a massive amount of EO data including remote sensing data and other sensor observation data such as earthquake, climate, ocean, hydrology, volcano, glacier etc. are being collected on a daily basis by a wide range of organizations and shared through the Internet. These datasets act as fundamental materials to help scientists study and understand various geophysical and social phenomena. The emerging geospatial cyberinfrastructure (GCI) rapidly increases our capacity for handling such massive data with regard to data collection, management, high-performance computing (HPC), data integration and interoperability, data transmission and visualization, etc. (Zhang et al., 2009; Yang et al., 2010; Wright et al., 2011; Rey et al., 2015; Li et al., 2016a; Li et al., 2016b; Li et al., 2016c; Song et al., 2016). These advancements of GCI make it a promising instrument for building science gateways under the environment of Internet to handle the time-critical tasks such as real-time environment monitoring, disaster management and decision-making (Zhang et al., 2005; Stollberg et al., 2012; Li et al., 2013).

Web service is a key element in GCI to foster interoperation of data from disparate sources. In these GCI enabled web services, the ability of rapidly transmitting and sharing spatial data over the Internet is critical to meet the demands of real-time change detection, response, and decision-making. In terms of geospatial data sharing, there exist many community-driven data sharing standards, among which the Open Geospatial Consortium's (OGC) Web Map Service (WMS; de La Beaujardiere, 2006) and Web

Feature Service (WFS; Vretanos, 2004) are mostly adopted in GCI applications. WMS is the standard protocol for serving georeferenced map images through the Internet while WFS is the standard protocol for serving geographical features (vector) data. Raster datasets such as remote sensing imageries are usually shared through the WMS protocol; while vector datasets could be shared through either WMS or WFS. If shared through WMS, the vector datasets will be pre-rendered as static images before being transmitted to users. If shared through WFS, the geometries and properties of the vector dataset will be directly disseminated with no information loss.

In many real-world data-driven applications, original vector datasets are essential for developing flexible, expressive and interactive data visualization and analysis functionalities to help users better understand the context of events and make decisions (Zhang et al., 2005; Stollberg et al., 2012). For example, in scenarios of disaster management, i.e. earthquake or flood, researchers need to retrieve multiple datasets including DEM (Digital Elevation Model), road networks, hydrology flow, population distribution, real-time observation data etc. from distributed Spatial Data Infrastructures (SDIs) and then conduct analysis immediately for developing evacuation and rescue plans. In other scenarios of real-time environment monitoring and traffic flow monitoring, massive real-time and historical environmental and traffic monitoring data in vector format need to be retrieved continuously. Then, such datasets will be used for analysis at the backend, in the cloud for example, and providing animated and interactive data visualization functionalities at the frontend, i.e. any browser which has an Internet connection.

Although sharing vector datasets through WFS brings a lot of benefits, it can also introduce serious performance issues: the data processing time through WFS mainly depends on original data sizes – if a vector layer is large, the data processing time in each stage of WFS will increase accordingly, including data preparation and encoding on the

server side, data transmission through the Internet and data decoding and visualization on the client side. Comparatively, for WMS, no matter how complex the original vector datasets are, they will be pre-rendered into static images and cached on the server side in advance. Once the preparation work is finished, the WMS processing time will be stable and irrelative to the original data size. This makes WMS a very efficient data sharing strategy. Such performance bottlenecks have hindered the widespread integration of WFS into a cyberinfrastructure, especially for those time-critical and data-massive applications. Currently, although many of datasets have been shared by different SDIs, only a small proportion of them are published through WFS.

In this study, we introduce our design and implementation of a comprehensive optimizing strategy for high-efficiency vector data sharing through WFS. The strategy consists of (1) Combination of pre-generalization and real-time generalization for multiple layers; (2) Separated data transmission processes of features' geometries and attributes; (3) Dynamic adoption of data compression/ decompression methods according to the network status. Significant improvements are achieved after applying this optimization strategy to conventional WFS approaches. The rest of this article is organized as follows: section 2 introduces the related work of this topic. Section 3 discusses the optimization strategies in detail. Section 4 introduces our experiments for performance comparison. In the last section, the conclusion and directions of future work are given.

4.2 Related work

Geospatial data sharing is now becoming a popular trend along with the increase in people's capability in collecting all kinds of EO data. Well-known organizations and agencies including the Global Earth Observation System of Systems (GEOSS; Christian, 2005), the INSPIRE geoportal of Europe (Bernard et al., 2005), National Snow & Ice

Data Center (NSIDC), the Geospatial Platform of U.S. Federal Geospatial Data Comittee, National Oceanic and Atmospheric Administration (NOAA), The Oak Ridge National Laboratory Distributed Active Archive Center (ORNL DAAC) etc. are continually collecting and providing a wide range of geospatial datasets to users. On the other hand, the developments of standards and technologies in recent decades have profoundly promoted the process of data sharing. The OGC Web Services (OWS) standardize how geospatial data and processing services could be published and shared through the internet. Service Oriented Architectures (SOA) are developed for wrapping functionalities into independent, interoperable, loosely-coupled and standard interfaces in purpose of sharing and reusing (Papazoglou et al., 2007; Giuliani et al., 2013). The revolution of Internet technologies such as Web 2.0, AJAX (Asynchronous JavaScript and XML) and HTML5 make it possible to build context-rich, interactive and user-friendly web applications which empowered the process of information transmission, data visualization, user communication and collaboration (Sayar et al., 2006; Rinner et al., 2008; Pierce et al., 2009; Boulos et al., 2010; Hall et al., 2010; Longueville et al., 2010; Li et al., 2011b).

Consequently, the number of SDIs and geospatial services has been increased rapidly (Sahin et, al. 2008) for providing various datasets, computation services and visualization tools for different user groups (Giuliani et al., 2013). Li et al. (2011a) developed a virtual Arctic SDI that introduces cross-catalog data harvesting, service chaining and online visualization to enhance understanding of the Arctic climate and ecosystem. Han et al. (2012a) developed an SDI which can help users conveniently retrieve DEM data of a customized region and conduct related data analysis on these datasets on the server side. Granell et al., (2010) introduced a web application designed under the principle of SOA for providing reusable hydrological models. Han et al. (2012b) introduced a web application which provides US conterminous geospatial cropland data

to users. Corresponding statistics and analysis tools are integrated to support decision making. Ames et al., (2012) built a web services-based software aiming at discovering, retrieving and analyzing hydrologic and climate data. Kulawiak et al., (2010) developed a web GIS application for serving the scenario of marine oil pollution monitoring, simulating and decision-making support. Raup et al., (2007) introduced a project named GLIMS (Global Land Ice Measurement from Space) which is the collaboration result of many institutions across the world. This project could provide rich glacier datasets, analyzing tools and services. Wang et al., (2016) developed a web application which is capable to identify polar cyclones and provide interactive 3D visualization tools for the cyclones.

As noted before that the original data size could greatly affect the performance of WFS process, many attempts have been made trying to resolve such issue. Yang et al. (2005) introduced interesting methods for improving the performance of web-based GIS, including data caching, multi-thread processing on the server side, and dynamic data requesting on the client side. Michaelis et al. (2012) introduced their implementation of WMS and WFS in a desktop application where some optimizations were tested, including data querying by the envelop and feature complexity reduction operations. Data generalization was suggested and implemented for providing map service through WMS and Web Processing Services (WPS, Schut et al., 2007; Foerster et al., 2010). Zhang et al. (2013) designed a parallel data query method to reduce the data retrieval time on the server side. Li et al. (2015) introduced the optimization strategy of data compression and decompression on the server and client sides to reduce the time for data transmission. However, this method can only be used for some specific scenarios.

Although progress has been made, a comprehensive study in improving WFS performance is still lacking. In our research, a comprehensive optimization strategy which contains multiple independent optimization steps will be introduced and

embedded into a WFS processing pipeline for performance enhancement. These independent optimization steps can be dynamically combined to fulfill requirements of different application scenarios.

4.3 Methodology

In general, a WFS processing involves the following workflow: when a web server receives a WFS request, it will first parse the request. Then, according to the parameters provided by the client, the WFS server accesses the required data source and conducts data processing. For example, a spatial filter operation will be applied to the raw data to derive a subset within the desired bounding box. After these processing steps, resultant features will be encoded into specific output format before being sent back to the client side. When the client side receives the response stream, it will decode the stream, parse the result, and convert it into a feature collection which could be used for visualization, statistics, and analysis. Figure 1 demonstrates the main components for WFS processing. In this workflow, we propose to integrate three stacked optimization strategies to further enhance its real-time performance: (1) Combination of data pre-generalization and real-time generalization to reduce the data complexity; (2) Separated data transmission processes of features' geometries and attributes; (3) Dynamic adoption of data compression and decompression methods according to various network conditions (boxes with orange borders in Figure 1 showcase where the integrations happen)

4.3.1 Geometry generalization for vector data

The most critical factor affecting the performance of WFS is the size of the data source. The larger the dataset is, the longer time it will take for data processing at each stage. On the other hand, one of the main purposes of WFS data retrieval is for visualization. If a complex geometry (e.g. the boundary of U.S.) containing tens of thousands of vertices is

drawn on the screen with a fixed resolution, many adjacent vertices will fall into the same pixel. This process not only burdens computation but also becomes less helpful in feature understanding. This means in practice it is unnecessary to draw very complex geometries for visualization. Therefore, it becomes an involuntary idea to simplify the geometries of vector layers for WFS processing.

Indeed, the topic of map generalization itself has long been studied by scholars (Weibel, 1997; Oosterom, 2009). Much effort has been dedicated to developing new generalization methods to present clear and accurate maps to audiences at different spatial scales. The implementation of our optimizing strategy is not restricted to any specific generalization methods. Users should select the appropriate algorithm according to their application requirements. For demonstration and experiments, we chose two efficient and robust algorithms -- Douglas-Peucker (DP; Douglas et al., 1973; Shen et al., 2008) and Topology Preserving (TP; Bajaj et al., 1998) in this research. Both algorithms simplify a line by recursively deleting some of its containing points while keeping the main shape of the line. A distance tolerance could be specified for controlling the simplicity of the result. The difference between DP and TP algorithm is DP executes much faster than TP, but TP preserves the topology relationship for features in a map.

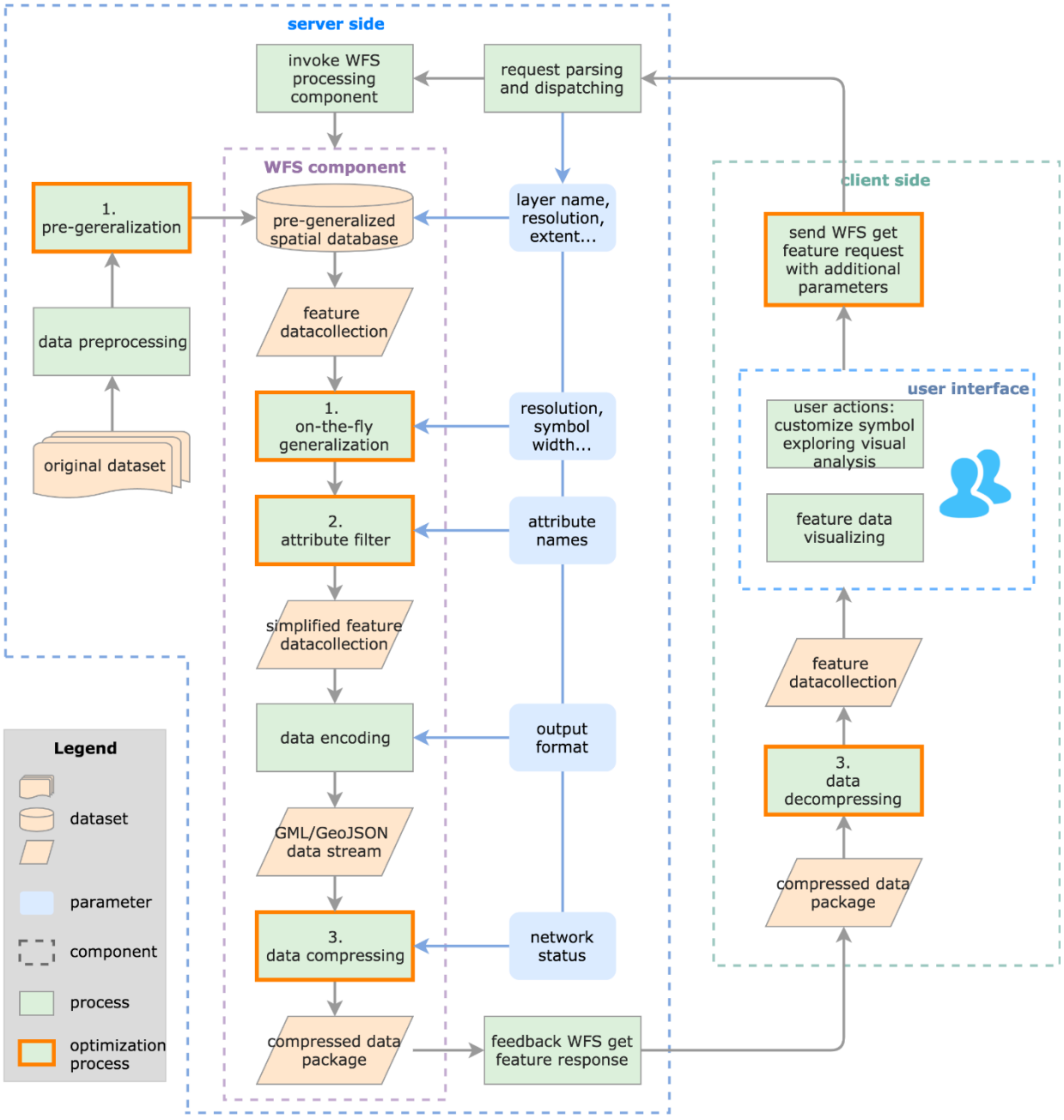


Figure 21. WFS workflow with optimization strategies

Finding the appropriate criterion for layer generalization is important: If the layer is generalized to be too simple, it will bring significant visual changes to the map; on the other extreme, it will impose unnecessary computing and transmitting burden to the system. In this research, we introduce the *Appropriate Distance Tolerance (ADT)* calculating method for a vector layer. The principle of ADT is maximizing the distance

tolerance for generalization but not significantly affecting the visualization results, meanwhile considering the web application's computing and transmitting capacity. The following factors should be considered for deciding the ADT:

- The scale of each pixel in the target visualization screen: this is the key factor for the generalization. If adjacent points of a line fall into the same pixel when presented, one of them can be safely deleted without affecting the visualization result.
- Symbolization scheme: considering the width of lines and polygon boundaries that are presented, large width means a coarse requirement of accuracy, then the distance tolerance can be loosened accordingly.
- Network speed and data processing capacity of the client side: under the circumstance of low network bandwidth or limited computing speed on the client side, the distance tolerance can be increased as well.

Respecting these three factors, we define the formula for calculating the ADT for a vector layer:

$$ADT = \alpha \cdot PiS \cdot \min_{f \in L}(w(f))$$

In this formula, α denote the coefficient for controlling the ADT according to network speed, data processing capacity of the client side and users' preference of geometries' detail – the smaller α is, the more detail will be kept. Empirically α could be set between the range of [0.3, 2.0]. PiS denote the scale of each pixel in monitor's canvas. PiS is decided by both the area of map for presenting (visible region) and the resolution of canvas. That says, if the resolution of canvas is N_h (horizontal) by N_v(vertical) and the visible region in map is D_h (horizontal) by D_v(vertical), then $PiS = \min\left(\frac{D_h}{N_h}, \frac{D_v}{N_v}\right)$. Finally, if the layer L adopts specific symbolization scheme, e.g. setting the width of roads with a certain width or setting varying widths according to the traffic-flow attribute, this

formula will use the minimum value of every feature f's symbol width w in L for calculating the ADT.

Although generalizing vector layers for WFS brings substantial benefits to the system, layer generalization itself could be time-consuming, especially when dealing with large datasets or processing frequent incoming requests from multiple clients. To solve this problem, we propose a two-step data generalization strategy – pre-generalization plus on-the-fly generalization. For a given data layer, generalization is performed using a sequence of distance tolerances (DTs) and the results are preserved locally. Upon receiving a request, the server will first calculate the ADT and select the pre-generalized layer whose DT is closest to but no greater than the ADT. Then on-the-fly generalization is further conducted on top of the selected pre-generalized data layer. Through this way, the entire generalization process can be greatly accelerated.

Note that DT sequence should be carefully selected for pre-generalization. Narrowing the interval of DTs could help improve the performance of on-the-fly generalization component. However, this will lead to the side-effect of consuming much storage space on the server side. Considerations on deciding the appropriate DT sequence include: (1) DT sequence should be designed according to map's varying presentation scales, which begins with the value that could fit the whole layer into the visible region. (2) Due to the limitation of visible region in the monitor, small map scale means that a large number of features in a layer will be presented. Therefore, more pre-generalized layers should be prepared for small scales. When it comes to large scales, a small number of features will be left after the spatial filtering by the visible region, therefore it will not take much time to finish the job of on-the-fly generalization. Then, fewer pre-generalized layers are needed in such case. 3) The smaller the DT is given, the fewer points in the features will be deleted during the generalization process. When deleted points are fewer than 20% of

total points in the original dataset, or too many layers have been created, the pre-generalization procedure will be ceased.

In summary, following is the criterion for determining layer pre-generalization DT sequence:

$$DT = \begin{cases} ADT_L & \dots l = 1 \\ ADT_M & \dots l = 2 \\ \dfrac{ADT_F}{2^{l-3}} & \dots 3 \le l \le 7 \\ \dfrac{ADT_F}{4^{l-5}} & \dots 8 \le l \le 10 \end{cases}$$

Here DT is calculated at different levels by using the formula of ADT. ADT_x means the calculated ADT value for a layer been fitted into a monitor with specific resolution, with $\alpha = 0.5$ and minimum symbol width equals 1. Here ADT_L, ADT_M and ADT_F represent the ADT for monitor with low (800×600), medium (1280×720) and full (1920×1080) resolutions respectively. l denotes different levels. According to this formula, the DT sequence has an accelerated descending trend, which is set as 2 after level 2, and becomes 4 after level 7. The pre-generalization procedure will stop at the 10th level, or the level where total deleted points do not exceed 20% of the original data points.

4.3.2 Attribute filtering according to users' demands

Both the geometry and attribute information of vector layers can be provided through WFS. While the attribute information of a vector layer is informative, it inevitably increases the data size. In many situations not all attributes are necessary, and different users have different preferences about the attributes. Therefore, the performance of WFS request could be improved by filtering out unnecessary attributes.

In this proposed implementation of a new WFS workflow, in order to avoid transmitting unnecessary attributes of vector layer, the metadata and statistical information of vector layer's attributes are provided by an independent API (Application Programming Interface). Such information is much smaller than the original attributes and is

accessible for users at any time. At the first time of a WFS request, only geometries and specified attributes of a vector layer will be returned. After reviewing the metadata, if users need any specific attributes for exploratory visualization or analysis, they can retrieve them separately by using the attribute filtering. When additional attributes arrive the client side, they could be added to the existing vector layer by matching their unique ids.

Table 7 demonstrates the examples of WFS request with different filtering strategies: the spatial filtering uses a boundary box to request features inside of a certain region, which is widely adopted; the attribute filtering specifies a certain list of attributes (i.e. the geometry and "STATE_NAME" in this example) for retrieval. These two filtering strategies can be applied in a joint way as well.

Table 7 Example of WFS request with different filtering strategies

Query Type	Example
Query with spatial filtering (use boundary box)	<wfs:GetFeature service="WFS" version="1.1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wfs" xmlns:wfs="http://www.opengis.net/wfs" xsi:schemaLocation="http://www.opengis.net/wfs" xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc" > <wfs:Query typeName="wps_pattern:NAT" srsName="urn:x-ogc:def:crs:EPSG:4326"> <ogc:Filter><BBOX> <ogc:PropertyName>the_geom</ogc:PropertyName> <Envelope srsName="urn:x-ogc:def:crs:EPSG:4326"> <lowerCorner>32.1 -125.1</lowerCorner> <upperCorner>42.0 -114.7</upperCorner> </Envelope> </BBOX></ogc:Filter> </wfs:Query></wfs:GetFeature>
Query with attribute filtering (use attribute names)	<wfs:GetFeature service="WFS" version="1.1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wfs" xmlns:wfs="http://www.opengis.net/wfs" xsi:schemaLocation="http://www.opengis.net/wfs" xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"> <wfs:Query typeName="wps_pattern:NAT" srsName="urn:x-ogc:def:crs:EPSG:4326"> <wfs:PropertyName>the_geom</wfs:PropertyName> <wfs:PropertyName>STATE_NAME</wfs:PropertyName> </wfs:Query> </wfs:GetFeature>

4.3.3 Data compression/decompression of encoded vector data

For data exchange and interoperability across different platforms, some commonly used vector layer output formats are supported by WFS, including GML (Geography Markup Language; Cox et al., 2002), KML (Keyhole Markup Language), GeoJSON (Butler et al., 2008), CSV (Comma-Separated Values) etc. Among these formats, GML and GeoJSON are the most commonly used. GeoJSON is designed based on the JSON (JavaScript Objective Notation) format. In GeoJSON, each feature is encoded into an object which consists of a list of key-value pairs that correspond to the name and value of feature attributes. GML (Geography Markup Language) is defined by the OGC to express geographic features. Inside of the GML document, the features are organized as a list of XML (eXtensible Markup Language) nodes, where the geometry and attribute information are stored in different tags.

Table 8 demonstrates how a single polygon feature is encoded into GeoJSON and GML formats. This feature has 3 attributes, viz., “Land”, “CFCC” and “LANAME”. The polygon of the feature consists of 4 vertices. Both of GeoJSON and GML are text based and contain many duplicated tags in their output files. Therefore, applying compression processes to the output data can reduce their size, and further reduce the time for data transmission.

Table 8 Example of using different output formats to encode a feature

GeoJSON	**GML**
{"type": "Feature", "id": "poly_landmarks.1", "geometry": { "type": "MultiPolygon", "coordinates": [[[[40.730647,-73.996035], [40.72999,-73.996449], [40.730437,-73.997356], [40.730834,-73.998047], [40.730647,-73.996035]]]] },	<tiger:poly_landmarks gml:id="poly_landmarks.1"> <tiger:the_geom> <gml:MultiSurface srsDimension="2" srsName="urn:x-ogc:def:crs:EPSG:4326"> <gml:surfaceMember> <gml:Polygon srsDimension="2"> <gml:exterior> <gml:LinearRing srsDimension="2"> <gml:posList>40.730647 -73.996035 40.72999 -73.996449 40.730437 -73.997356 40.730834 -73.998047 40.730647 -73.996035</gml:posList> </gml:LinearRing>

```
"geometry_name": "the_geom",
"properties": {
  "LAND": 2,
  "CFCC": "D85",
  "LANAME": "Washington
  Square Park"}}
```

```
        </gml:exterior>
</gml:Polygon></gml:surfaceMember></gml:MultiSurface>
</tiger:the_geom>
<tiger:LAND>2.0</tiger:LAND>
<tiger:CFCC>D85</tiger:CFCC>
<tiger:LANAME>Washington Square Park</tiger:LANAME>
</tiger:poly_landmarks>
```

Text data compression itself is a very active research topic. Classic data compression algorithms include: Run-length encoding (RLE; Robinson et al., 1967), Burrows–Wheeler transform (Burrows et al., 1994), Huffman coding (Huffman, 1952), Prediction by partial matching (PPM; Cleary et al., 1984), LZ77 (Ziv et al., 1977), LZ78(Ziv et al., 1978) etc. Currently, there are dozens of available data compression methods and toolkits derived from these algorithms. In consideration of the requirements for data interoperability and performance optimization, the target data compression methods for WFS should possess the characteristics of (1) robust and well performed in terms of compression speed and compression ratio; (2) widely adopted; (3) have available software development kits (SDK) for both server and client sides integration. The DEFLATE (Deutsch, 1996) and LZMA (Lempel–Ziv–Markov chain; Pavlov, 2007) algorithms are selected for integration and testing in this research as both are widely adopted. The DEFLATE algorithm is a combination of LZ77 and Huffman encoding. While the LZMA algorithm is a derivation of LZ77. Generally, the DEFLATE method compresses files faster than LZMA, but the generated files have less compression ratio (Li et al., 2015).

For a WFS-supported web application, the time (Δt) been saved by integrating the compression process equals to the time reduced on data transmission subtracts time used for compression and decompression. Δt can be expressed as:

$$\Delta t = \frac{Size_{original} - Size_{compressed}}{ts} - \left(\frac{Size_{original}}{cs} + \frac{Size_{original}}{ds}\right)$$

Here $Size_{original}$ means uncompressed data size, $Size_{compressed}$ means compressed data size. cs denotes data compressing speed on the server side, ds denotes data decompressing speed on the client side. ts denotes data transmitting speed through the Internet. In this formula, $Size_{original} - Size_{compressed}$ can be expressed as $Size_{original} \times (1 - 1/cr)$, where cr denotes compression ratio of the algorithm, which equals to $\frac{Size_{original}}{Size_{compressed}}$. For this formula, as long as $\Delta t > 0$, it is worthy to integrate the compression/decompression component.

In practice, factors like the computing capacity of the server system, network speed and the performance of compression algorithm could all affect Δt. Once the data is prepared, the system can adaptively select one compression algorithm with the maximum Δt:

$$\max(\{0, \Delta t_1, \Delta t_2, \dots \Delta t_n\})$$

here 0 means no compression method is needed, Δt_i means saved time by using a certain compression method.

4.4 Experiments and Performance Comparison

We implemented the proposed optimization strategies into the WFS component of the open source software GeoServer. A geospatial cyberinfrastructure portal is developed for data retrieval and visualization. The performance of the experiments is obtained under such computing environment: the GeoServer and the CI portal are hosted on a server machine with a 6-core 3.39 GHz 64-bit Xeon CPU and 8 GB RAM running Ubuntu 14.04.2. The client side is tested in the FireFox browser (version 51.0.1) on a laptop machine with a 4-core 2.90 GHz 64-bit Intel i-7 CPU and 8GB RAM running Windows 10. The screen resolution is 1920×1080. Additionally, the experiments are conducted under a high Internet speed environment (50Mbps).

Table 9 Statistics of the datasets for experiments

ID	Dataset name	Data type	Number of attributes	Total feature number	Total vertex number	Table size before pre-gen (MB)	Table size after pre-gen (MB)
1	Census tract	polygon	12	8057	2903671	53	136
2	WBDHU12	polygon	19	5315	6941774	118	320
3	NHDWaterbody	polygon	12	111653	4091652	94	284
4	NHDArea	polygon	11	11790	3047995	54	140

Four relatively complex geospatial datasets in the region of California State, U.S. are selected for the experiments. These datasets are (1) Census tract regions; (2) Watershed Boundary Dataset (WBD) at the level of 6 (the most detailed level in which the sub-watersheds are recorded); (3) Areal hydrographic waterbody (NHDWaterbody) features and (4) Areal (NHDArea) hydrographic landmark features (U.S. Geological Survey and U.S. Department of Agriculture, Natural Resources Conservation Service, 2013). Details of the datasets are listed in Table 9 All the datasets are of polygon type and contain multiple attributes. The number of features in each dataset ranges from a few thousand (dataset 2) to hundreds of thousands (dataset3). The total number of vertices in each dataset exceeds one million. Each dataset is stored in a database table. The last two columns of Table 3 list the data table size before and after pre-generalization. Figure 22 visualized these four datasets.

The proposed optimization strategies could be directly embedded into the WFS process pipeline as shown in Figure 21. In the pipeline, each stage's output becomes the input of its following stage. In the rest of this section, we will introduce the experiments as per their sequential order in the pipeline.

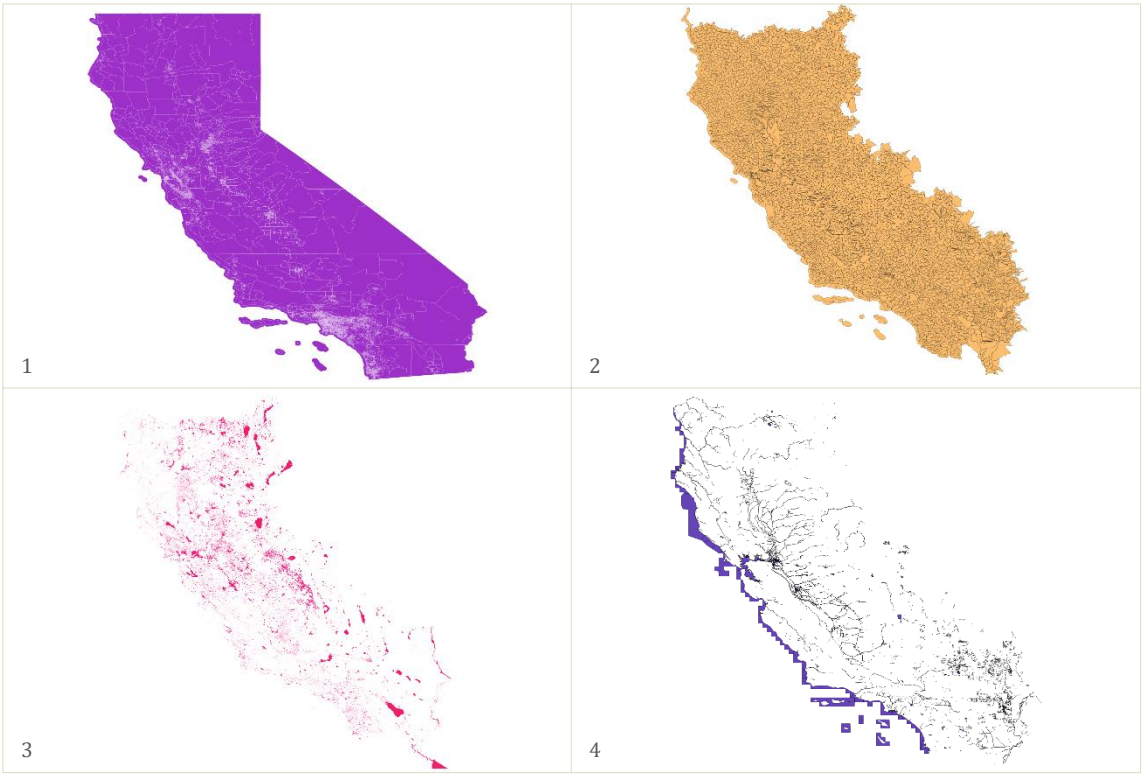


Figure 22. Geospatial data layers for experiments. 1.census track polygons, 2. Watershed Boundary Dataset (WBD), 3. Areal hydrographic waterbody (NHDWaterbody), 4. Areal (NHDArea) hydrographic landmark features

The experiments were conducted on varying scales, which correspond to different zoom levels in the browser: if the zoom level increases by 1, the scale of the map will double, and the visible region in the browser will be reduced to ¼ of the previous level. In our experimental environment, level 6th is the minimum level to fit the whole study area into the visible region. While at higher levels the client side only needs to request partial dataset inside of the visible region to support visualization. To keep the conciseness of the article, we will only introduce our experiments at the 6th level, which is the worst case since the entire datasets will be processed. At the end of this section, we will compare the WFS performance with and without optimizations on the dimension of varying scales.

4.4.1 Generalization

The pre-generalization is conducted by following the rules in section 0 and using the DP algorithm. The original geometries, attributes and generalized geometries of a dataset

are stored as a single data table in a spatial database, i.e. PostGIS. The advantage of using a database table to store such information is to avoid duplicated storage of attribute data when they are stored in independent files.

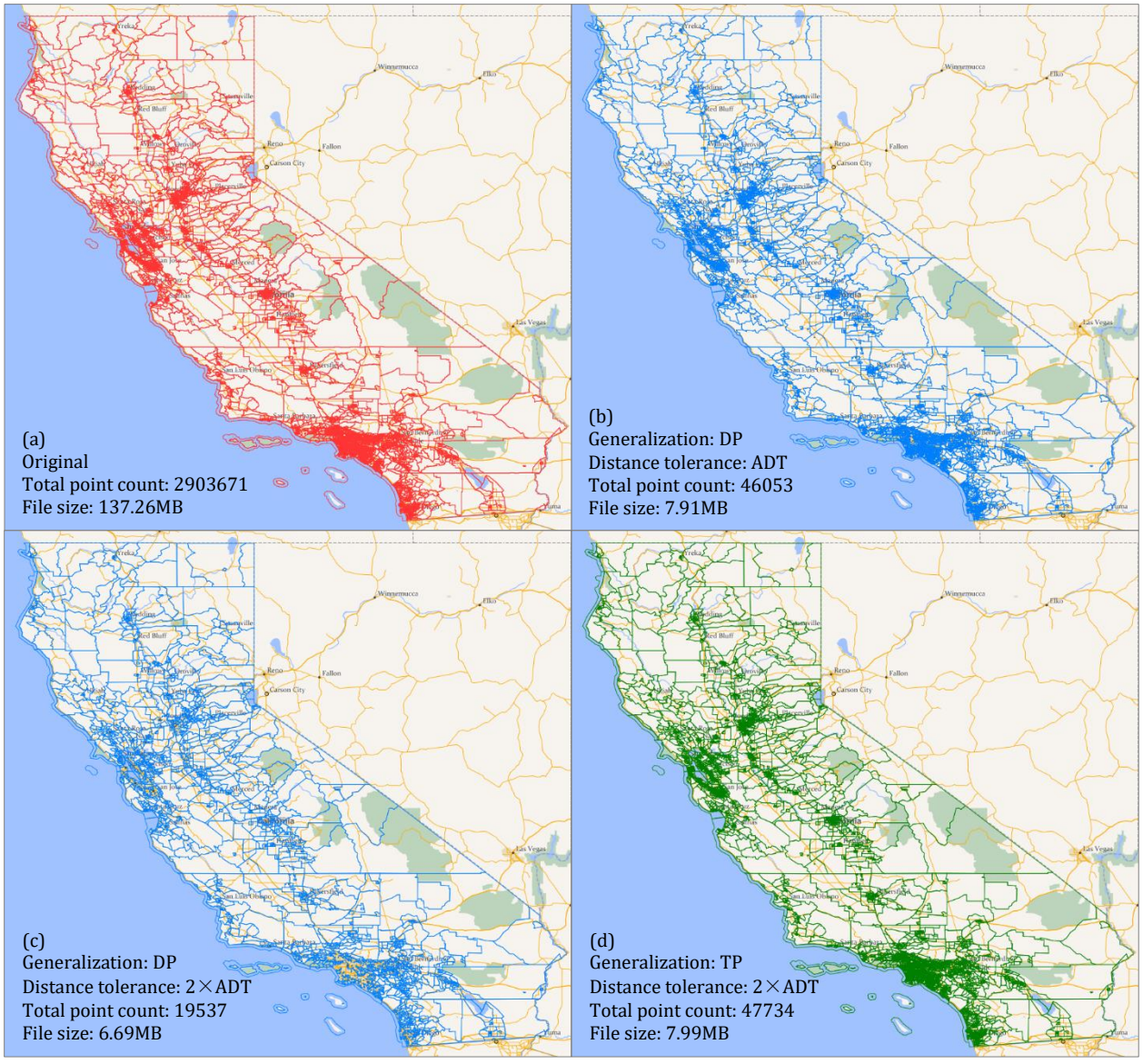


Figure 23. Comparison of vector layer generalization results by using different distance tolerances and different generalization algorithms

Figure 23 demonstrates the generalized census tract layer by using different distance tolerances and generalization algorithms. The ADT is calculated with $\alpha = 0.6$ and the polygon boundaries' width are set to 1 pixel. Figure 23 (a) presents the original layer. In Figure 23(b), the layer is generalized with ADT using DP algorithm. Barely any

difference could be observed from the graphics, but the data size is significantly decreased after the generalization: total points number in Figure 23(b) is only 1.59% of the original data (from 2903671 to 46053) and the file size is only 5.76% of the original file (from 137.26 Mb to 7.91 MB, GeoJSON format). If we increase the DT for DP generalization from ADT to $2ADT$, there will be some obvious differences in the metropolitan areas of California State, including San Francisco, Los Angeles and San Diego – the reason for the hollow areas is because some polygons are deleted entirely due to their tiny size. Figure 23 (d) demonstrates the layer generalized by using TP with 2ADT. The map is comparable with the original one in Figure 23 (a). Since small polygons are preserved after the generalization, the result is suitable for spatial analysis on the client side. However, the tradeoff is time consumption for the TP algorithm is longer than the DP algorithm. Users could select appropriate generalization according to their application requirements. We will mainly use DP for the rest experiments.

Figure 24 demonstrates the decrease of vertex number in each layer by the two stages of generalization: there are very significant vertex reductions in the first stage of pre-generalization (Figure 24(A)); then in the second stage of on-the-fly generalization, it could also achieve approximately 30% points reduction.

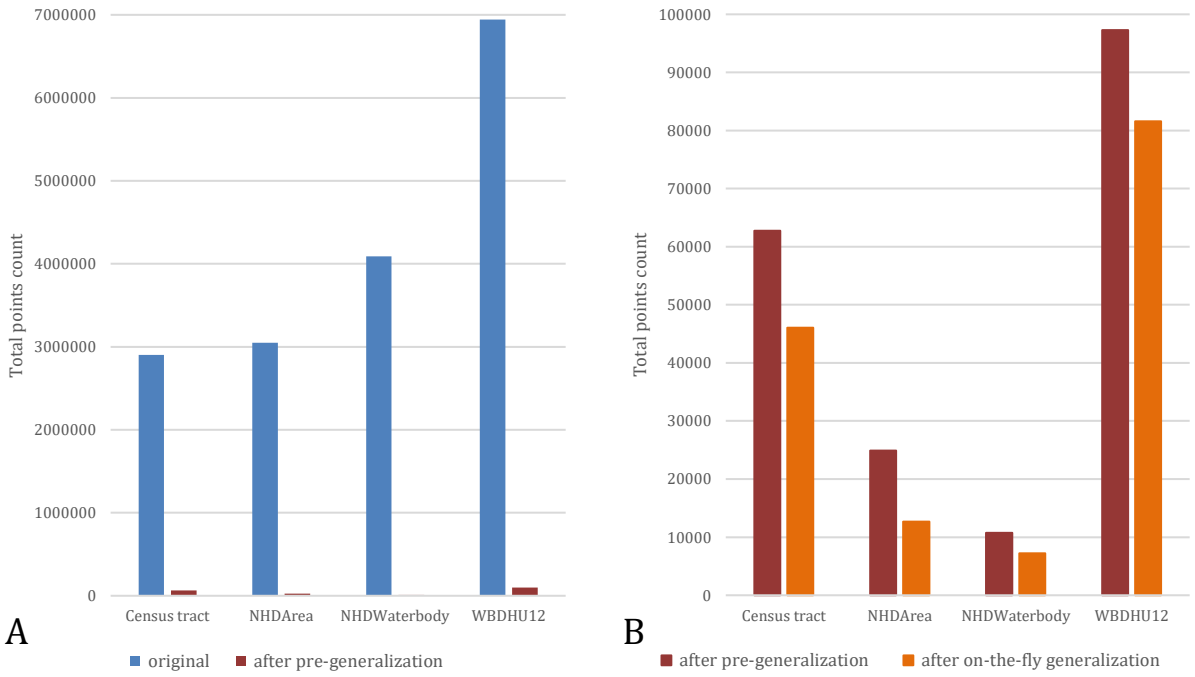


Figure 24. Comparison of total points reduction in two stages of generalization. A: pre-generalization; B: on-the-fly generalization

4.4.2 Attributes filtering

As we discussed in section 0, although the attributes of layer features are informative, it is not always necessary to provide all of them at the first time of a data request. For the purpose of presenting data faster, only the geometries and another one or two key attributes need to be initially retrieved. Other attributes can then be gradually transmitted upon users' demands. Figure 25 demonstrates the comparison of file size before and after the attributes filtering optimization. For all the four testing datasets, the file sizes dropped for more than 60% after the attributes filtering. Indeed, the number of attributes in each layer decides how much of the size could be reduced – significant data size reduction can be achieved in this experiment because all these layers contain more than 10 attributes (Table 9). The experimental results also indicate that the file sizes encoded by GeoJSON are smaller than those encoded by GML.

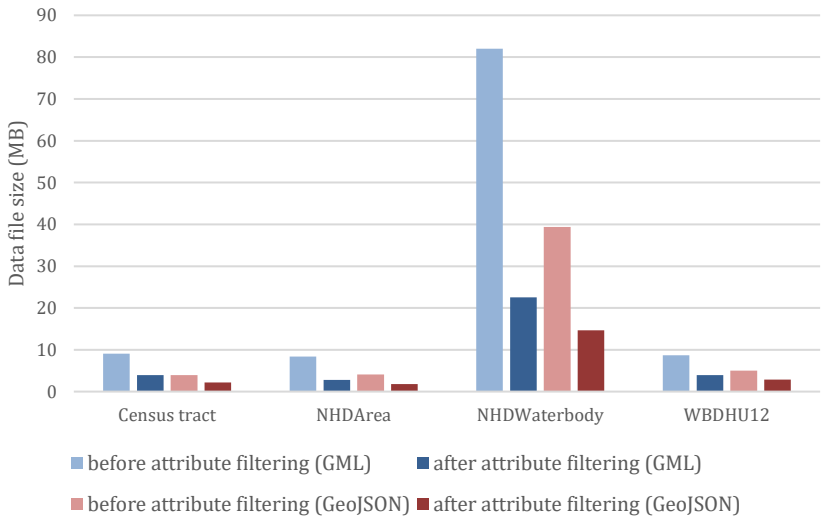


Figure 25. Comparison of file sizes before and after attribute filtering

4.4.3 Data compression

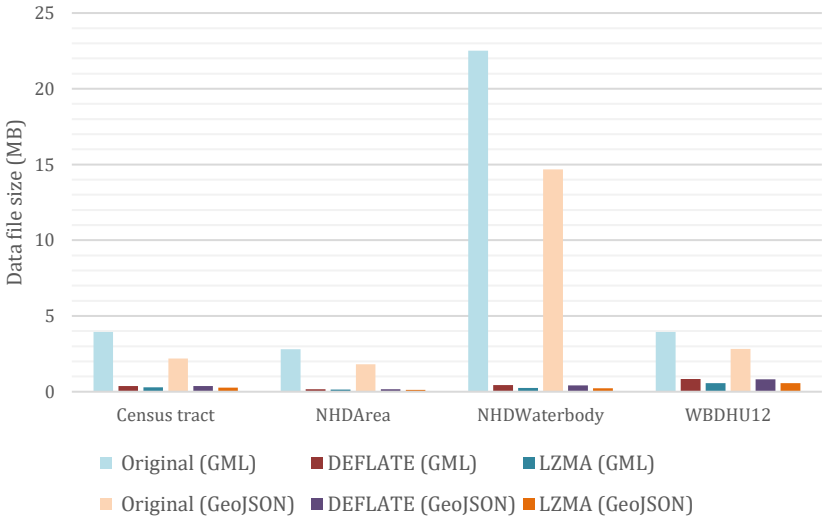


Figure 26. Comparison of file sizes before and after compression

After the last stage of attribute filtering, the content of the layers has been prepared. Before sending those layers back to the client side, compression is conducted on the data files to reduce data size and save network transmission time. Figure 26 presents the file sizes before and after the compression. As the graphic shows, both the DEFLATE and LZMA could achieve very good compression rate, while the LZMA method does a better job in getting smaller compressed files. Another interesting finding is, for the same

dataset and same compression method, the compressed files have approximately the same size, for either GML or GeoJSON encoding. Finally, no matter which encoding method or compression method is used, the file sizes of all 4 layers are less than 1 MB after the compression.

4.4.4 Overall performance comparison

This section introduces the overall performance improvements in terms of time consumptions and file sizes after applying all optimization strategies. The experiments were conducted on varying scales, which begin from zoom level 6th and ends at zoom level 16th in the map.

In a complete a WFS request-response cycle, the raw feature data will go through eight processing stages, including: (1) data preparation (e.g. read original data from driver or database); (2) on-the-fly generalization; (3) encoding the features into specific formats (e.g. GML or GeoJSON); (4) data compression; (5) data transmission through the internet; (6) data decompression on the client side; (7) feature decoding and (8) layer rendering in browser.

Among these stages, the 3rd, 4th and 5th are coupled: At stage 3, the features of a layer are sequentially encoded into an output stream in the memory. While for the compression component at stage 4, it could begin the compressing work as long as there is content in the output stream of stage 3, instead of wait until stage 3 finishes all its work. In other words, the encoding and compression process could work simultaneously. The data transmission component works in the same mode. Consequently, the time used for these three steps cannot be separated from each other. The total time for WFS process could be calculated as:

$$T = t_{pre} + t_{gen} + t_{ect} + t_{dcom} + t_{dcod} + t_{rd}$$

Here, t_{pre} denotes time for data preparing, t_{gen} denote on-the-fly generalization time, t_{ect} means the times used for encoding, compression and transmission. t_{dcom} and t_{dcod} represent decompression and decoding time. And t_{rd} denotes time for layer rendering.

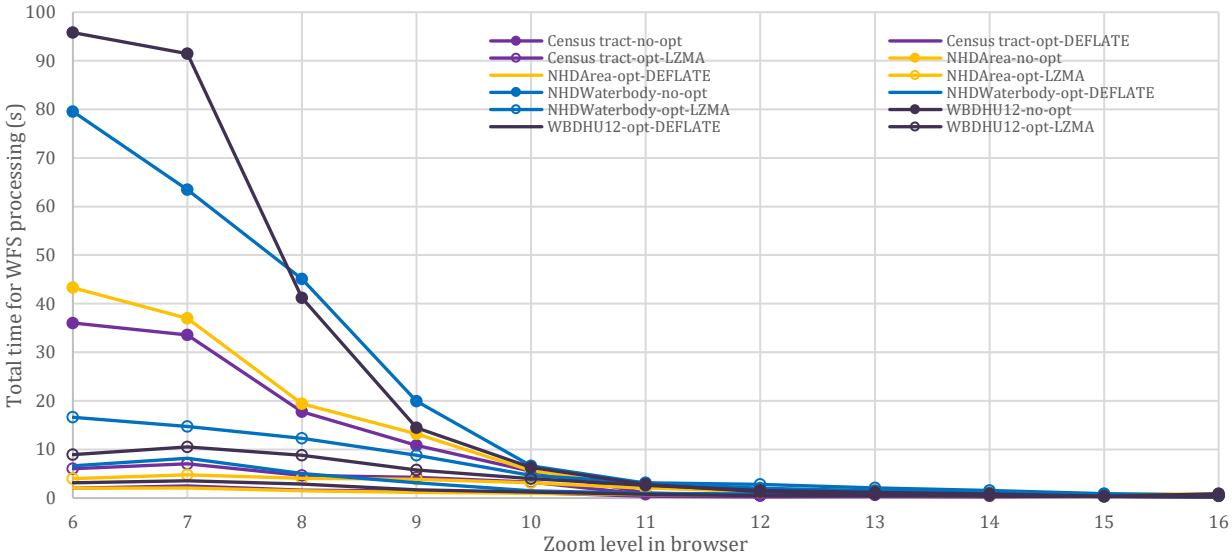


Figure 27. Comparison of time consumption at different zoom levels before and after applying the optimization strategies

Figure 27 demonstrates the total time consumption for WFS processing at different zoom levels. In the graphic, different data layers are presented with different colors. Three types of processing methods are compared: (1) process with no optimization (lines with solid dots); (2) process with all optimization strategies and use DEFLATE for compression (lines without dot); (3) process with all optimization strategies and use LZMA for compression (lines with hollow dots). According to the graphic, as the zoom level increases, since the area of visible region decreases, performance of all processing methods get better. But at low zoom levels, WFS performances are significantly improved after applying the optimization strategies. Especially for the cases using DEFLATE for compression, the WFS process time is controlled under 10 seconds for any dataset at any zoom level.

Figure 28 shows the details of time consumption at different stages of WFS processing at zoom level 6 – where the whole datasets are processed. Obviously, t_{pre}, t_{ect}, t_{dcod} and t_{rd} are much shorter after applying the optimization strategies. Benefiting from the pre-generalization process, t_{gen} is very short as well. Time been used for compression and decompression by LZMA are longer than DEFLATE, which should be the main reason that optimization with DEFLATE performs better than with LZMA. According to the experimental results, DEFLATE is a better choice than LZMA for compression in time-critical application scenarios, such as real-time environemnt monotoring or public data service.

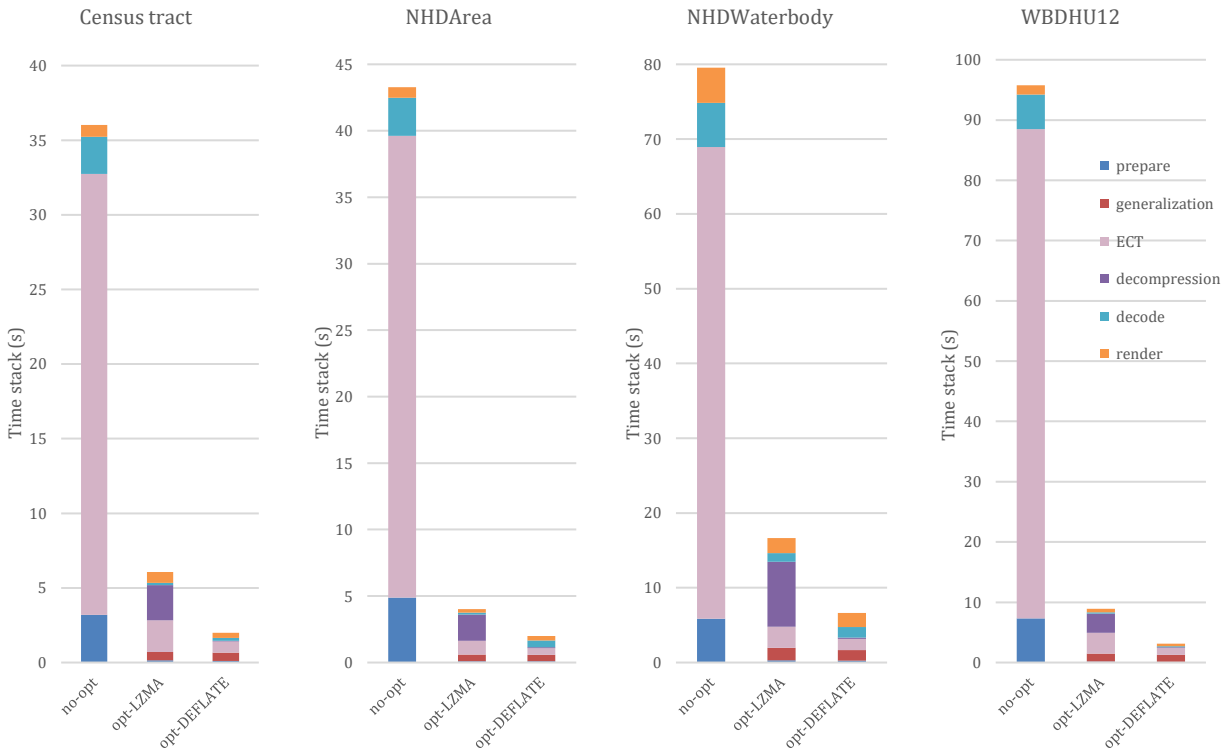


Figure 28. Details of time consumption at different stages of WFS processing (level 6th)

Figure 29 demonstrates how the sizes of layers for transmission are reduced after applying the optimization strategies. The data package for transmission is controlled within 1 MB for any dataset at any zoom level. In fact, for the two compression methods, LZMA could achieve better compression ratio than DEFLATE. Therefore, for the

application scenarios with low or limited network bandwidth, i.e. emergency rescue or field investigation, LZMA is a preferred optmization strategy for compression.

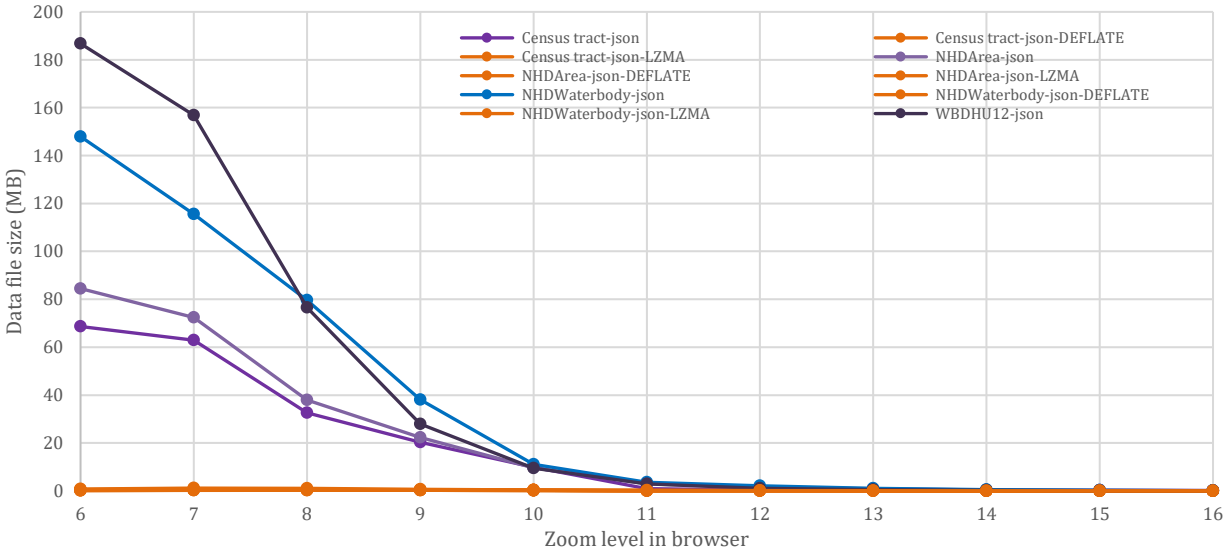


Figure 29. Comparison of data sizes at different zoom levels for transmission

4.4.5 An extension to a nation-wide dataset

In this section, we use a much larger dataset – census tract polygons of the entire United States – to test the capability of our methods. Figure 30 demonstrates the profile of the dataset, which originally contains 73682 polygons and 35.8 million vertices. If all the 53 properties are considered, the original file is larger than 1GB in GML format. Under the former experiment environment, level 4 is the minimum level to fit the layer into a visible region.

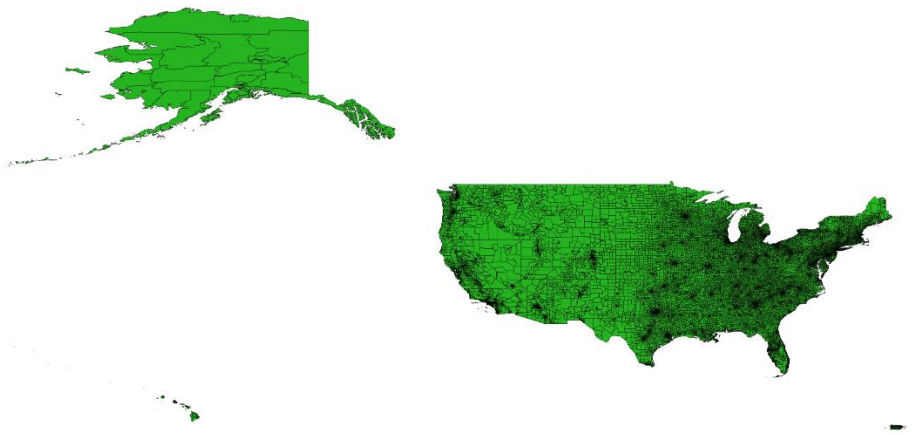


Figure 30 Census tract data of United States

Figure 31 summarizes the experimental results at each data processing stage on the server side for visualization at level 4th. We can observe that in every step the data size gets significantly reduced. The final compressed data files for transmission through the Internet are less than 2 MB.

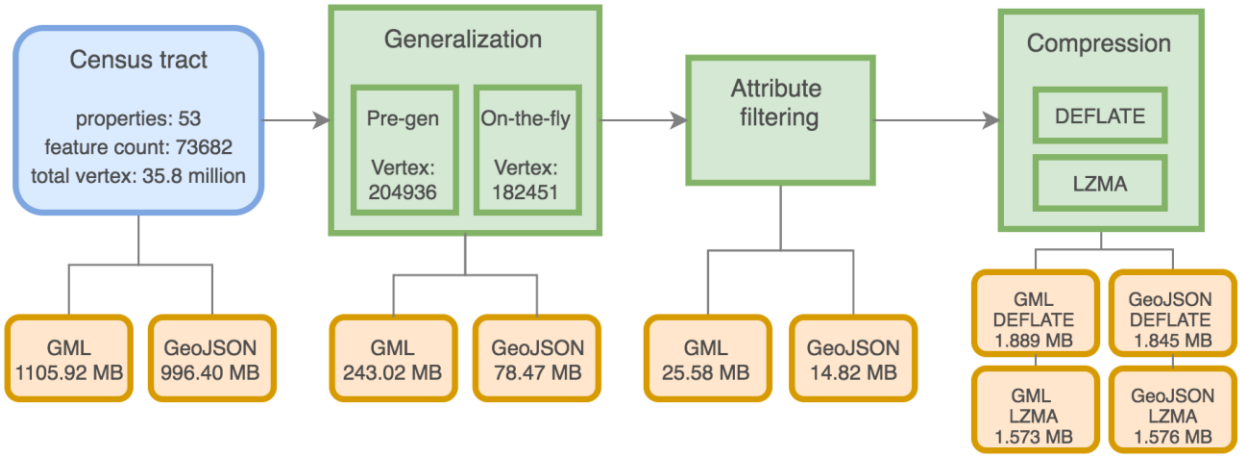


Figure 31 Experiment summary on testing the US census tract data

Figure 32 shows the comparison of overall performance in time consumption. If the WFS model is not optimized (as shown in blue color), at lower zoom levels where many geometries will be returned, it will cause the "memory over flow" exceptions in browser. The data can only be visualized at 8th level or higher. However, after our optimization, the data can be successfully visualized at any level while the time is controlled under a

reasonable range (less than 15 seconds for the worst case by using DEFLATE compression).

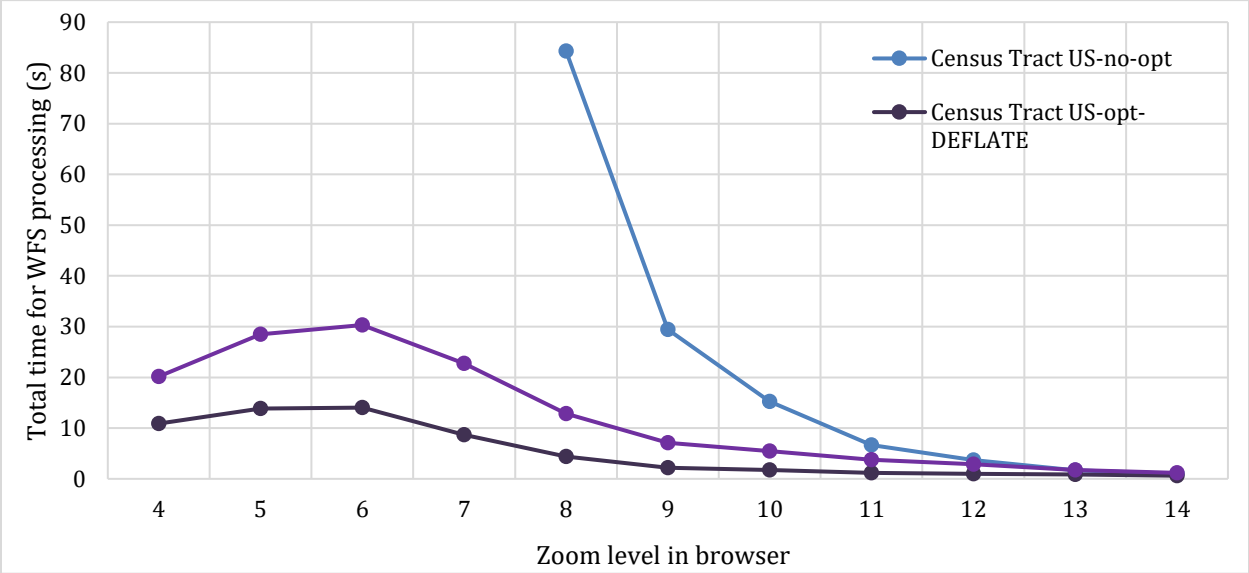


Figure 32 Time consumptions at different zoom levels and using different optimization strategies for US census tract data.

The results show that our optimization methods also work well with large datasets. The main reason is, no matter how large the original datasets are, the size of screen for visualization on the client side is fixed. Larger regions will result in greater ADTs, which means we can use a loosen distance tolerance for data generalization. After generalization, the size of resultant file will be greatly reduced. Since a two-step generalization strategy is adopted, the most time-consuming part is the first step but this is already finished at the data preparing stage. The time used for on-the-fly generalization will not become significant. Hence, this strategy guarantees the efficiency for processing large datasets.

4.5 A cyberinfrastructure implementation and graphic user interface

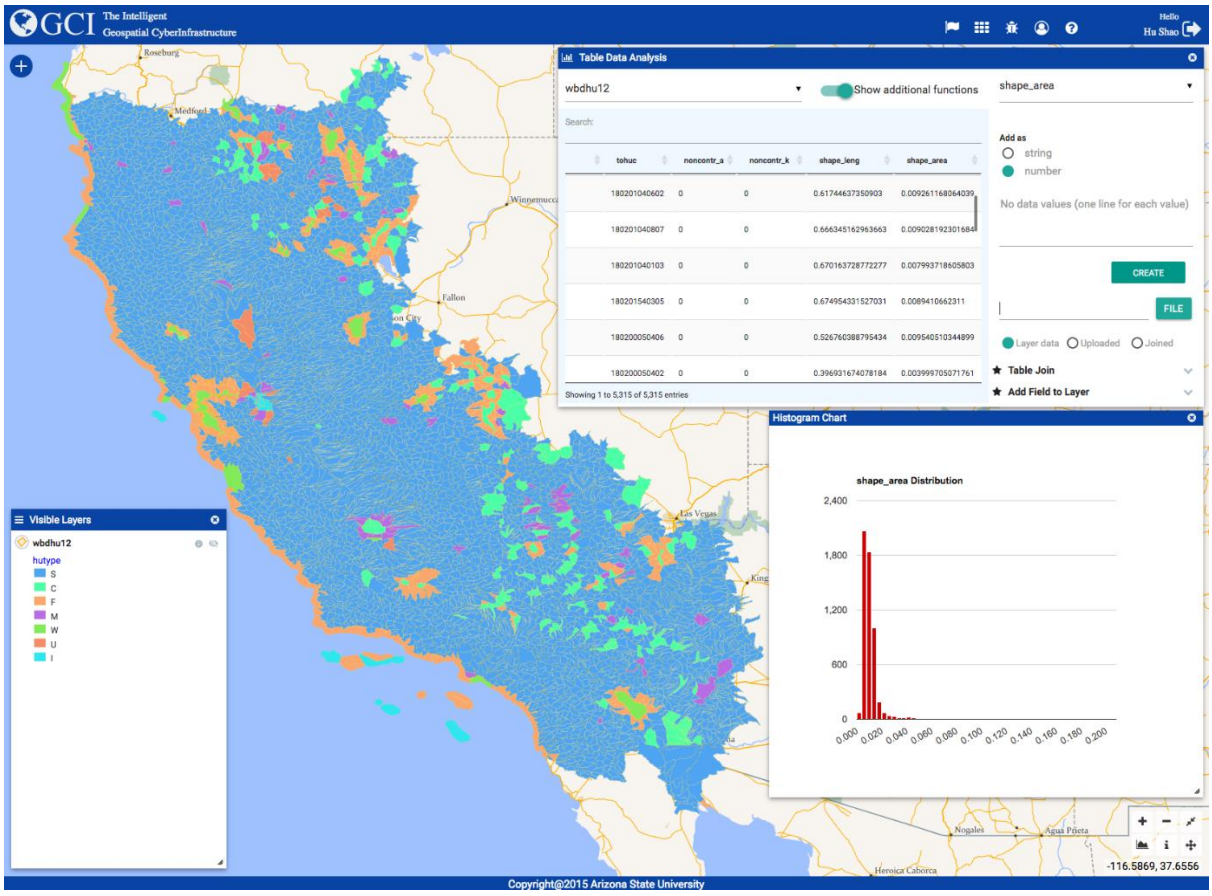


Figure 33. GUI of the CI portal for feature data visualization

Figure 33 demonstrates our GCI web portal. This portal could retrieve, manage and visualize any map or vector layers published by OGC's WMS and WFS standards. The proposed optimization strategies and rich interactive functionalities have been implemented in this portal for feature visualization and analysis. For the WFS server which hosts all the experiment data at the backend, the light-weighted metadata information of all its vector layers will be retrieved and made available to users. According to this metadata information and current computing and network status, the system will calculate the best WFS request parameters. Users could customize the parameters as well. After the features of a layer are requested and delivered, they will be presented in the map immediately. Besides, users could browse the attribute information

of the layer; customize layer's symbology scheme according to their attributes and conduct statistics on the features.

The Watershed Boundary Dataset for the experiment is presented in the map. After applying the optimization strategies, the data layer could be retrieved and visualized rapidly. The color of each feature in the layer is set according to their hydrologic unit type. The "table data analysis" component provides the function of attribute value browsing for each feature (Figure 33, up-right corner). The component could also conduct statistics on a layer's attributes and present the results to users in graphic (Figure 33, bottom-right corner).

4.6. Conclusion

To achieve the goal of supporting real-time spatial feature sharing and visual analytics for massive datasets, this chapter introduces a comprehensive optimization strategy to improve the performance of WFS. The following optimization strategies are introduced and embedded in the WFS process pipeline: 1) Combination of pre-generalization and real-time generalization for multiple layers; 2) Separated data transmission processes of features' geometries and attributes; 3) Dynamic adoption of data compression/decompression methods according to the network status. We have successfully integrated these optimization strategies into a WFS server and conduct corresponding comparison experiments on 4 relatively complex datasets for California area and a large dataset for the U.S. According to the experimental results, significant performance improvements are achieved: in the worst case when the whole dataset is requested, the total WFS processing time is reduced by 90% in general.

Major advantages of the proposed methodology include that all these strategies are independent of each other and can be flexibly assembled. In addition, the data

processing pipeline does not rely on any specific dataset or generalization algorithm or compression algorithm: data providers could select appropriate generalization methods and compression algorithms to be integrated into the pipeline to address different needs. Data consumers could also select the combination of the optimization strategies according to their demands and the network/system environment. For example: in the application scenario in which visualization is the main purpose of data querying, i.e. to develop a real-time water quality visualization and monitoring system, DP generalization algorithm and DEFLATE compression method could be employed. Meanwhile, interactive and user-friendly web portals can be designed and implemented as well to help users better understand and use the vector datasets. On the other hand, if data analyses are needed in addition to visualization, users can choose more rigorous generalization algorithms like topology preserving algorithm, or directly use the non-generalized dataset for their analyses on the client side. Another more feasible way is by taking advantage of geo-cyberinfrastructure, spatial analyses can be conducted on the server side or on cloud with more powerful CPUs and well-designed algorithms. Then analysis results with generalized layer can be returned to the client side using our current proposed strategies for visualization and decision making (Li et al., 2016a; Yang et al., 2010, 2017; Wright et al., 2011).

To leverage the usage of our work, more research will be conducted on the two directions of: (1) employ cloud computing platform and parallel computing strategies to enhance the WFS service capacity and deal with synchronously requests from different users. (2) design and implement interactive and user-friendly web application to help users better understand and use the vector datasets. As the importance of data sharing is well-recognized by scientific community, the demand for building interactive and intelligent geospatial web applications is becoming more urgent in both the fields of scientific

research and daily use. In this context, we expect our work to widen the interoperability of vector data and the adoption of WFS in future.

5 GEOCI: THE COMPREHENSIVE CYBERGIS PLATFORM THAT INTEGRATES ALL THE COMPONENTS TOGETHER

The three research topics of my dissertation have all been successfully implemented and integrated into the comprehensive CyberGIS platform – GeoCI. These components are tightly combined with the basic GIS functionalities in the platform, making it capable of helping users to accomplish a wide range of analysis tasks, such as geospatial data discovery, data integration, data management, user account and workspace management, spatial data visualization, exploratory data visual analytics, spatial and spatial-temporal data analysis, high-volume spatial transmission and visualization etc. Besides these functionalities, rich documentation, tutorials, and well-designed spatial analysis study cases are provided as well to help the beginners to get familiar with the system.

Figure 34 demonstrates the architecture of the GeoCI and how different components interact with each other, including the primary components of 1. User management, 2. Semantic enhanced geospatial data search engine 3. High-performance feature data transmission component and 4. WebPySAL on the server side and the interactive GCI portal on the client side.

Firstly, a complete user management tool is implemented in GeoCI. Each user needs to apply for a user account. In their own account, users can create and manipulate multiple workspaces for different research topics, and specific its spatial reference system. Then, users will be able to add and delete data layers in each workspace. Besides the workspace management functions, users can also customize the system's behaviors and save them to their own configuration file.

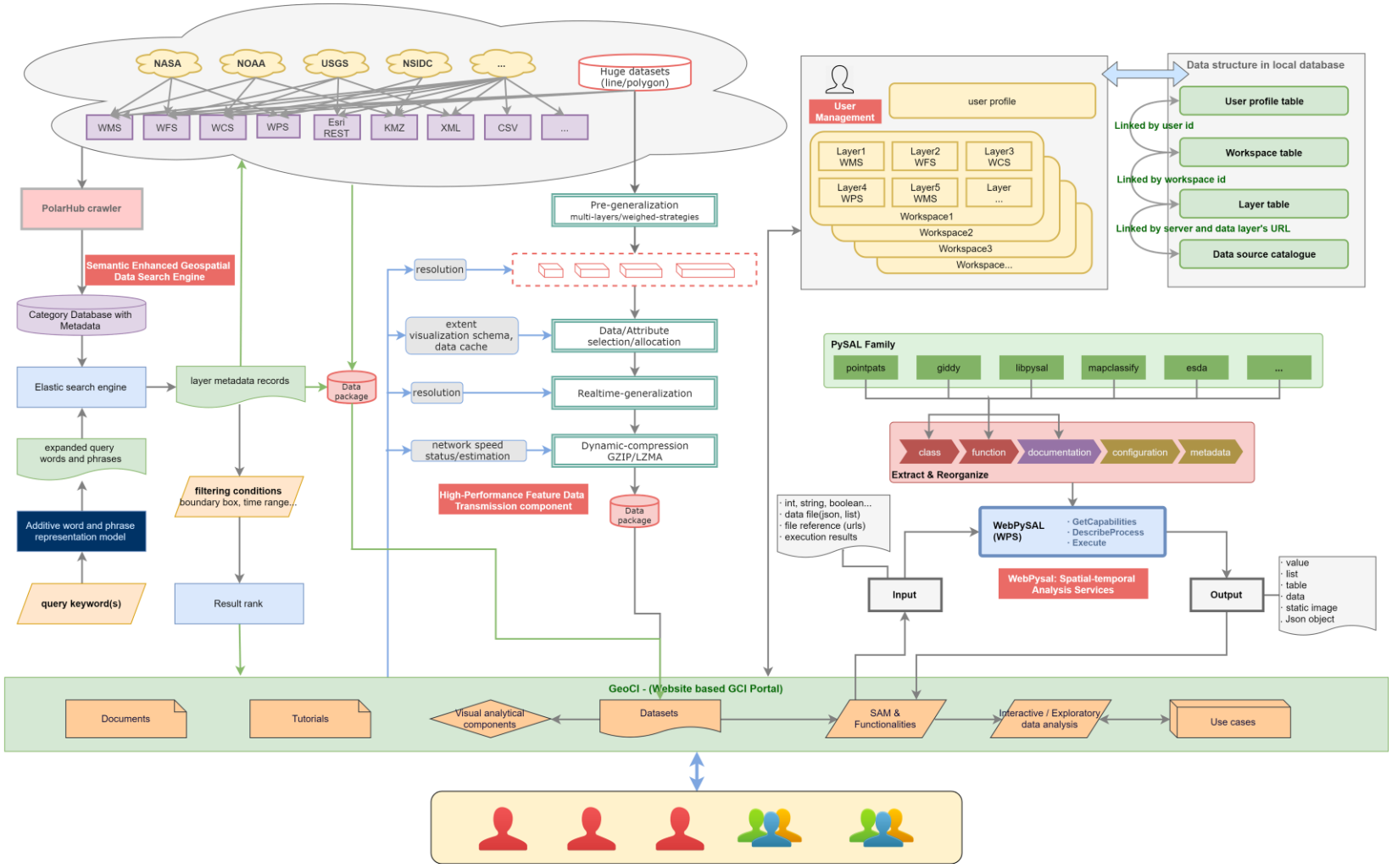


Figure 34 Architecture of the GeoCI Platform

All the information will be stored into the relational database on the server side, making the working environment available to users anywhere and anytime as long as they have access to the Internet. Figure 35 demonstrates the workspace management tool (Figure 35 (a)) and data layer management tool (Figure 35 (b)) in GeoCI.

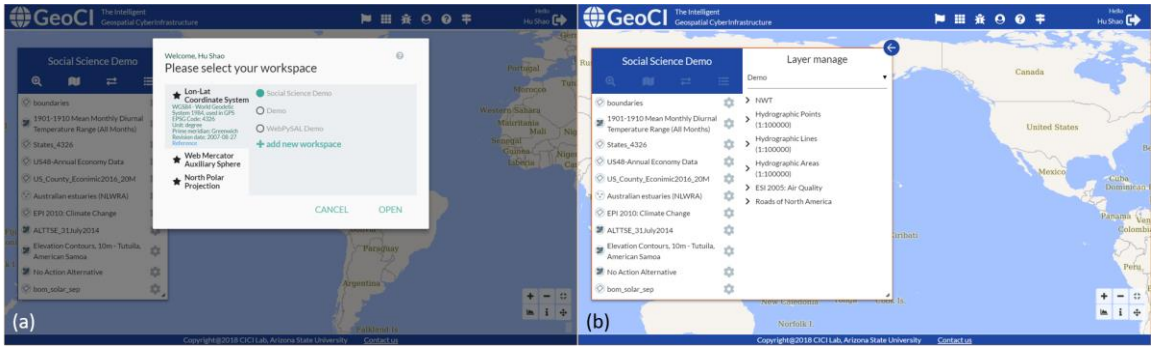


Figure 35 The workspace management tool (a) and layer management tool (b) in GeoCI

One of the most important resources of GeoCI is the metadata dataset collected from thousands of open geospatial data servers around the world. These metadata records are stored in the local database and act as the building concrete of the semantic data search engine (Chapter Two). When the user provides the searching keywords and filtering conditions, the search engine will find the most related geospatial data records and return them to the client side. The metadata information and thumbnail of the data layers will be presented to in an interactive dialogue for the user to select (Figure 8). When the user selects the layers of interest, they will be added to the current workspace for later visualization and analysis (Figure 9).

This platform also provides fused social economic and natural disaster datasets for the spatial analysis showcases. OGC's open geospatial data sharing standards such as WFS and WMS are adopted for sharing the high volume geospatial datasets, making them discoverable in our semantic search engine just like other datasets. More than that, the data publishing service in GeoCI harnesses the optimized spatial feature sharing and

visual analytics technologies developed in Chapter Three in order to transmit, visualize and process large volume geospatial datasets rapidly (Figure 32, Figure 33).

After the steps of data search and management, users can then go further to conduct exploratory visual analytics and spatial/space-time analyses on their selected datasets by using the analysis modules provided by GeoCI. Figure 36 (a) demonstrates one of the basic visual analysis function in GeoCI. Figure 36 (b) presents the advanced space-time analysis modules integrated into GeoCI as a list, including the modules in WebPySAL (Chapter Four). Most of the analysis modules are implemented as standard WPS service, which means they can be seamlessly integrated into third-party GIS platforms as long as they support WPS as well. For these integrated spatial analysis modules, interactive and sophisticated UIs are meticulously designed to help users interpret and understand the analysis results (Figure 17, Figure 18, Figure 19).

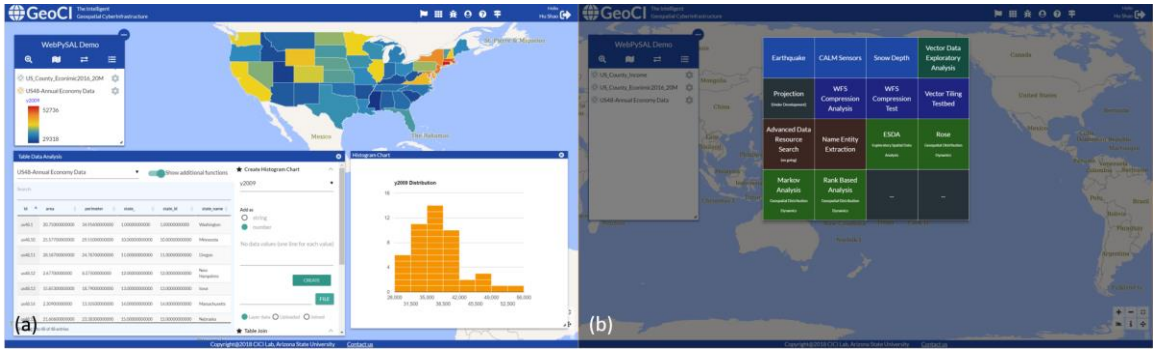


Figure 36 basic visual analysis functions (a) and the list of advanced space-time analysis functions in GeoCI

Finally, abundant documentation, tutorials, and additional functionalities are provided in GeoCI to help users quickly get familiar with the system and educate them to use the spatial analysis modules step by step. Figure 37 (a) presents the static help document introducing the general functions of GeoCI. Figure 37 (b) demonstrates an interactive tutorial which guides users to conduct the geospatial data search step by step.

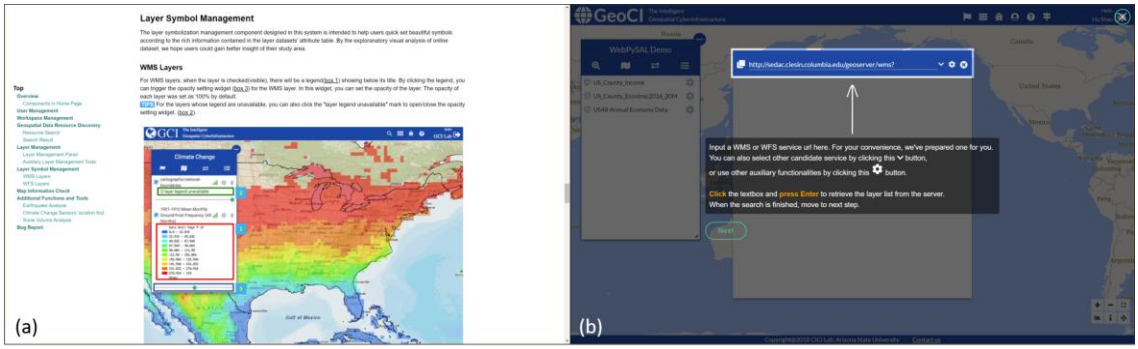


Figure 37 Static help documentation (a) and interactive tutorial (b) provided by GeoCI

As I mentioned before, integrating all the components together as a synthetic system could help enhance the capacity of each other in helping users accomplishing complex tasks. A working instance of GeoCI is hosted on http://cici.lab.asu.edu/gci2. The system is still under development and more functionalities are going to be integrated in near future.

6 CONCLUSION

This dissertation is mainly dedicated to addressing some critical issues or improving the performance of existing methodologies and systems in the field of CyberGIS. Main findings and achievements are listed below.

There are oceans of open geospatial data been shared on the Internet in nowadays, however, the disconnected and heterogeneous nature of the datasets have greatly limited their usage to the potential data consumers. In chapter 2, I designed and implemented a semantic enhanced data discovery system which adopts the state-of-art word and phrase representation methodologies from the natural language processing (NLP) filed to automatically extract semantic relationships among individual words and phrases in the metadata. At the same time, multiple metadata enrichment strategies and result ranking methods are introduced into the system to improve the quality of the data searching result. With the help of this data discovery system, 1. The semantic relationship between words and phrases in the metadata could be extracted and stored into the semantic database. 2. This semantic database could help significantly improve the recall rate of the data search results. 3. With the help of metadata quality enhancement methods and result ranking methods, the precision of data searching result could also been improved. 4. Most of the working flow could be conducted automatically without much labor inputs, making it very suitable to handle large dataset. The data discovery system is implemented and integrated into the GeoCI cyberinfrastructure portal for providing the search functionalities to public users.

Besides the increasement of available geospatial datasets, the GIScience has also ushered tremendous development in recent decades that numerous new methodologies and algorithms have been invented. Meanwhile there exist numbers of vibrant GIScience teams working on integrating the most advanced algorithms and methodologies into

open source libraries or software toolkits. Harnessing these open source toolkits on the big data and HPC environment and making them accessible to public users could bring immediate benefits to the GIScience community. In chapter 3, I established the WebPySAL as a working instance instead of a prototype to fulfill such task. Much efforts are dedicated to introducing the strategies and methodologies to guarantee the interoperability and replicability in the practice of implementing a standard geospatial web processing service. An interactive and user-friendly GUI is developed to assist users in conducting exploratory spatial/spatiotemporal data analysis with massive open access geospatial data sets. In addition to potential benefits this work brings by bridging spatial analysis toolkits with CyberInfrastructure, the design and implementation of this system could potentially help users who are lack of GIScience background knowledge or programming skills to better understand and adopt advanced spatial analytical methodologies.

Feature dataset which contains both geometries and attributes information of the study objects is the most popular data types for visualizations and analyses in scientific research. However, the huge volume nature of the feature datasets has hindered their wide adoption in the web-based working environment, especially in those time critical application scenarios. In chapter 4, I introduce a comprehensive optimization strategy to improve the performance of feature sharing methods through the internet. The following optimization strategies are introduced and embedded in the WFS process pipeline: 1. Combination of pre-generalization and real-time generalization for multiple layers; 2. Separated data transmission processes of features' geometries and attributes; 3. Dynamic adoption of data compression/decompression methods according to the network status. These optimization strategies are successfully integrated into a WFS server and corresponding comparison experiments conducted on different complex datasets. According to the experimental results, significant performance improvements

are achieved: in the worst case when the whole dataset is requested, the total WFS processing time is reduced by 90% in general.

Besides the findings and achievements in each individual work, a CyberGIS portal named GeoCI is established during my Ph.D. period. All the components result from the individual research have been integrated into GeoCI and work as a well-integrated system. In the system, the individual components could interact with each other and enhance each other's capacity in helping users accomplish tasks from geospatial data discovery to exploratory spatial and spatial-temporal analyses. The system implementation work is introduced in chapter 5. Putting all these together, I believe my work possess the great potential in helping users take advantage of the advanced technologies and spatial analysis methods in GIScience field. And a step further, this work could also help leverage the collaboration work among researchers from different discipline in future.

The future working directions of my research will include the following points:

In terms of data discovery, 1) A more precise evaluation system should be implemented to measure the improvement of precision and recall rate of the geospatial data searching system compared with the baseline system based on full-text match search and LSI method. 2) In this research, the POS method is adopted for extracting phrases from our metadata. In future, some more sophisticated entity recognition methods based on the neural network models will be introduced into the system to improve the searching result. 3) There exist a few high-quality geospatial ontology knowledgebases (e.g. GCMD). Introducing these knowledgebases into the result filtering and ranking stages in the system could potentially improve the searching result as well.

For the WebPySAL work. Firstly, WebPySAL will be published as a member of PySAL's family on GitHub[15], and the integration work of PySAL's advanced spatial analysis functionalities will be continued. An active instance of WebPySAL is currently available at http://cici.lab.asu.edu:5002. Parallel spatial analysis modules will be integrated into WebPySAL to leverage the HPC resources in CyberInfrastructure to help solve more challenging tasks in the future.

To leverage the usage of data transmission optimization work, more research will be conducted on the two directions of: 1) employ cloud computing platform and parallel computing strategies to enhance the WFS service capacity and deal with synchronously requests from different users. 2) design and implement interactive and user-friendly web application to help users better understand and use the vector datasets. As the importance of data sharing is well-recognized by scientific community, the demand for building interactive and intelligent geospatial web applications is becoming more urgent in both the fields of scientific research and daily use. In this context, I expect my work to widen the interoperability of vector data and the adoption of WFS in future.

[15] https://github.com/pysal

Bibliography

Ames, D. P., Horsburgh, J. S., Cao, Y., Kadlec, J., Whiteaker, T., & Valentine, D. (2012). HydroDesktop: Web services-based software for hydrologic data discovery, download, visualization, and analysis. Environmental Modelling & Software, 37, 146–156. https://doi.org/10.1016/j.envsoft.2012.03.013

Anoop, V. S. and Asharaf, S. 'Distributional Semantic Phrase Clustering and Conceptualization Using Probabilistic Knowledgebase', 1, pp. 526–534.

Anselin, L. (1995). Local indicators of spatial association-LISA. Geographical Analysis, 27(2), 93–115.

Anselin, L., & Rey, S. J. (2012). Spatial econometrics in an age of CyberGIScience. International Journal of Geographical Information Science, 26(12), 2211-2226.

Anselin, L., & Rey, S. J. (2014). Modern spatial econometrics in practice: A guide to GeoDa, GeoDaSpace and PySAL. GeoDa Press.

Anselin, L., Rey, S. J., & Li, W. (2014). Metadata and provenance for spatial analysis: the case of spatial weights. International Journal of Geographical Information Science, 28(11), 2261-2280.

Anselin, L., Syabri, I., & Kho, Y. (2010). GeoDa: An introduction to spatial data analysis. Handbook of Applied Spatial Analysis, 73–89.

Astsatryan, H., Hayrapetyan, A., Narsisian, W., Saribekyan, A., Asmaryan, S., Saghatelyan, A., … Ray, N. (2015). An interoperable web portal for parallel geoprocessing of satellite image vegetation indices. Earth Science Informatics, 8(2), 453–460.

Bajaj, C. L., & Schikore, D. R. (1998). Topology preserving data simplification with error bounds. Computers & Graphics, 22(1), 3–12.

Banea, C. et al. (2014) 'SimCompass: Using Deep Learning Word Embeddings to Assess Cross-level Similarity', Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), (SemEval), pp. 560–565. Available at: http://alt.qcri.org/semeval2014/cdrom/pdf/SemEval098.pdf.

Bartusiak, R. et al. (2017) 'WordNet2Vec: Corpora agnostic word vectorization method', Neurocomputing, pp. 1–29. doi: 10.1016/j.neucom.2017.01.121.

Battle, R. and Kolas, D. (2012) 'Enabling the geospatial semantic web with parliament and geosparql', Semantic Web. IOS Press, 3(4), pp. 355–370.

Ben-Joseph, E., & Gordon, D. (2000). Hexagonal planning in theory and practice. Journal of Urban Design, 5(3), 237-265.

Bernard, L., Kanellopoulos, I., Annoni, A., & Smits, P. (2005). The European geoportal––one step towards the establishment of a European Spatial Data Infrastructure. Computers, environment and urban systems, 29(1), 15-31.

Bickenbach, F., & Bode, E. (2003). Evaluating the Markov property in studies of economic convergence. International Regional Science Review, 26(3), 363–392.

Bivand, R., Anselin, L., Berke, O. al, Bernat, A., Carvalho, M., Chun, Y., ... Others. (2011). spdep: Spatial dependence: weighting schemes, statistics and models. R package version 0.5-31, URL http://CRAN. R-project. org/package= spdep. Retrieved from http://ftp.auckland.ac.nz/software/CRAN/src/contrib/Descriptions/spdep.html

Blacoe, W. and Lapata, M. (2012) 'A comparison of vector-based representations for semantic composition', in Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning, pp. 546–556.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. Journal of machine Learning research, 3(Jan), 993-1022.

Bojanowski, P. et al. (2016) 'Enriching word vectors with subword information', arXiv preprint arXiv:1607.04606.

Boulos, M. N. K., Warren, J., Gong, J., & Yue, P. (2010). Web GIS in practice VIII: HTML5 and the canvas element for interactive online mapping. International Journal of Health Geographics, 9, 14. https://doi.org/10.1186/1476-072X-9-14

Bowers, S., Lin, K. and Ludascher, B. (2004) 'On integrating scientific resources through semantic registration', in Scientific and Statistical Database Management, 2004. Proceedings. 16th International Conference on, pp. 349–352.

Burrows, M., & Wheeler, D. J. (1994). A block-sorting lossless data compression algorithm. Systems Research, Research R(124), 24. https://doi.org/10.1.1.37.6774

Butler, H., Daly, M., Doyle, A., Gillies, S., Schaub, T., & Schmidt, C. (2008). The GeoJSON format specification. Rapport Technique, 67.

Čepický, J. (2007). PyWPS 2.0.0: The presence and the future. Geoinformatics FCE CTU, 2(0), 61–64.

Cerón, M., Fernández-Carmona, M., Urdiales, C., & Sandoval, F. (2018). Smartphone-Based Vehicle Emission Estimation. In Proceedings of the International Conference on Information Technology & Systems (ICITS 2018) (pp. 284–293). Springer International Publishing.

Cho, K. et al. (2014) 'Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation'. doi: 10.3115/v1/D14-1179.

Chris taller, W. (1966). Central places in southern Germany. Prentice-Hall.

Christian, E. (2005). Planning for the Global Earth Observation System of Systems (GEOSS), 21, 105–109. https://doi.org/10.1016/j.spacepol.2005.03.002

Cleary, J., & Witten, I. (1984). Data compression using adaptive coding and partial string matching. IEEE Transactions on Communications, 32(4), 396–402.

Cliff, A. D., & Ord, J. K. (1981). Spatial processes, models & applications. London: Pion.

Conneau, A. et al. (2017) 'Supervised Learning of Universal Sentence Representations from Natural Language Inference Data'. doi: 10.1.1.156.2685.

Cox, S., Cuthbert, A., Daisey, P., Davidson, J., Johnson, S., Keighan, E., … others. (2002). OpenGIS{®} Geography Markup Language (GML) Implementation Specification, version.

de La Beaujardiere, J. (2006) 'OpenGIS{®} web map server implementation specification', Open Geospatial Consortium Inc., OGC, pp. 6–42.

Delipetrev, B., Jonoski, A., & Solomatine, D. P. (2014). Development of a web application for water resources based on open source software. Computers & Geosciences, 62, 35–42.

Deutsch, L. P. (1996). DEFLATE compressed data format specification version 1.3.

Douglas, D. H., & Peucker, T. K. (1973). Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Cartographica: The International Journal for Geographic Information and Geovisualization, 10(2), 112–122.

Dredge, D. (1999). Destination place planning and design. Annals of tourism research, 26(4), 772-791.

Droegemeier, K. K. et al. (2005) 'Service-oriented environments for dynamically interacting with mesoscale weather', Computing in Science & Engineering. IEEE, 7(6), pp. 12–29.

Dubois, G., Schulz, M., Skøien, J., Bastin, L., & Peedell, S. (2013). eHabitat, a multi-purpose Web Processing Service for ecological modeling. Environmental Modelling & Software, 41, 123–133.

Dwivedi, V. P. (2017) 'Beyond Word2Vec : Embedding Words and Phrases in Same Vector Space Beyond Word2Vec : Embedding Words and Phrases in Same Vector Space', (December).

Foerster, T., Lehto, L., Sarjakoski, T., Sarjakoski, L. T., & Stoter, J. (2010). Map generalization and schema transformation of geospatial data combined in a Web Service context. Computers, Environment and Urban Systems, 34(1), 79–88. https://doi.org/10.1016/j.compenvurbsys.2009.06.003

Fotheringham, A. S., Brunsdon, C., & Charlton, M. (2003). Geographically weighted regression: the analysis of spatially varying relationships. John Wiley & Sons.

Franz, T., Schultz, A., Sizov, S., & Staab, S. (2009, October). Triplerank: Ranking semantic web data by tensor decomposition. In International semantic web conference (pp. 213-228). Springer, Berlin, Heidelberg.

Gan, Z. et al. (2016) 'Learning Generic Sentence Representations Using Convolutional Neural Networks'. doi: 10.21437/Interspeech.2016-82.

Giuliani, G., Dubois, A., & Lacroix, P. (2013). Testing OGC Web Feature and Coverage Service performance: Towards efficient delivery of geospatial data. Journal of Spatial Information Science, 7(7), 1–23. http://doi.org/10.5311/JOSIS.2013.7.112

Gollapalli, M., Li, X. and Wood, I. (2013) 'Automated discovery of multi-faceted ontologies for accurate query answering and future semantic reasoning', Data and Knowledge Engineering. Elsevier B.V., 87, pp. 405–424. doi: 10.1016/j.datak.2013.05.005.

Goodchild, M. F. (2007). Citizens as sensors: the world of volunteered geography. GeoJournal, 69(4), 211-221.

Granell, C., Díaz, L., & Gould, M. (2010). Service-oriented applications for environmental models: Reusable geospatial services. Environmental Modelling & Software, 25(2), 182–198. https://doi.org/10.1016/j.envsoft.2009.08.005

GuoDong, Z., LongHua, Q. and QiaoMing, Z. (2009) 'Label propagation via bootstrapped support vectors for semantic relation extraction between named entities', Computer Speech and Language. Elsevier Ltd, 23(4), pp. 464–478. doi: 10.1016/j.csl.2009.03.001.

Hall, G. B., Chipeniuk, R., Feick, R. D., Leahy, M. G., & Deparday, V. (2010). Community-based production of geographic information using open source software and Web 2.0. International Journal of Geographical Information Science, 24(5), 761–781. https://doi.org/10.1080/13658810903213288

Han, W., Di, L., Zhao, P., & Shao, Y. (2012a). DEM Explorer: An online interoperable DEM data sharing and analysis system. Environmental Modelling & Software, 38, 101–107. https://doi.org/10.1016/j.envsoft.2012.05.015

Han, W., Yang, Z., Di, L., & Mueller, R. (2012b). CropScape: A Web service based application for exploring and disseminating US conterminous geospatial cropland data products for decision support. Computers and Electronics in Agriculture, 84, 111–123. https://doi.org/10.1016/j.compag.2012.03.005

Harris, R. (2003). Building a GIScience Community in Cyberspace: reflections on GIScOnline. Journal of Geography in Higher Education, 27(3), 279–295.

Hashimoto, K. and Tsuruoka, Y. (2016) 'Adaptive Joint Learning of Compositional and Non-Compositional Phrase Embeddings', pp. 3–7. doi: 10.1088/0953-8984/28/20/205401.

Hey, T., & Trefethen, A. E. (2005). Cyberinfrastructure for e-Science. Science, 308(5723), 817-821.

Horsburgh, J. S., Tarboton, D. G., Piasecki, M., Maidment, D. R., Zaslavsky, I., Valentine, D., & Whitenack, T. (2009). An integrated system for publishing environmental observations data. Environmental Modelling & Software, 24(8), 879-888.

Hu, Y., Janowicz, K., Prasad, S., & Gao, S. (2015). Metadata Topic Harmonization and Semantic Search for Linked-Data-Driven Geoportals: A Case Study Using ArcGIS Online. Transactions in GIS, 19(3), 398-416.

Huang, Q., Cervone, G., Jing, D., & Chang, C. (2015). DisasterMapper: A CyberGIS framework for disaster management using social media data. In Proceedings of the 4th International ACM SIGSPATIAL Workshop on Analytics for Big Geospatial Data - BigSpatial'15 (pp. 1–6). New York, New York, USA: ACM Press.

Huffman, D. A. (1952). A method for the construction of minimum-redundancy codes. Proceedings of the IRE, 40(9), 1098–1101.

Janowicz, K., Raubal, M. and Kuhn, W. (2011) 'The semantics of similarity in geographic information retrieval', Journal of Spatial Information Science, 2(2), pp. 29–57. doi: 10.5311/JOSIS.2011.2.3.

Jansen, S. (2017) 'Word and Phrase Translation with word2vec'. Available at: http://arxiv.org/abs/1705.03127.

Jiang, Y., Li, Y., Yang, C., Liu, K., Armstrong, E. M., Huang, T., ... & Finch, C. J. (2017). A comprehensive methodology for discovering semantic relationships among geospatial vocabularies using oceanographic data discovery as an example. International Journal of Geographical Information Science, 31(11), 2310-2328.

Kang, W., & Rey, S. J. (2018). Conditional and joint tests for spatial effects in discrete Markov chain models of regional income distribution dynamics. The Annals of Regional Science, 1-21.

Kitchin, R. (2013). Big data and human geography: Opportunities, challenges and risks. Dialogues in human geography, 3(3), 262-267.

Kolda, T. G., & Bader, B. W. (2009). Tensor decompositions and applications. SIAM review, 51(3), 455-500.

Krisnadhi, A. et al. (2015) 'The GeoLink modular oceanography ontology', in International Semantic Web Conference, pp. 301–309.

Kulawiak, M., Prospathopoulos, A., Perivoliotis, L., Łuba, M., Kioroglou, S., & Stepnowski, A. (2010). Interactive visualization of marine pollution monitoring and forecasting data via a Web-based GIS. Computers & Geosciences, 36(8), 1069–1080. https://doi.org/10.1016/j.cageo.2010.02.008

Kullback, S., Kupperman, M., & Ku, H. H. (1962). Tests for contingency tables and Markov chains. Technometrics: A Journal of Statistics for the Physical, Chemical, and Engineering Sciences, 4(4), 573–608.

Laura, J., Li, W., Rey, S. J., & Anselin, L. (2015). Parallelization of a regionalization heuristic in distributed computing platforms – a case study of parallel-p-compact-regions problem. International Journal of Geographical Information Science: IJGIS, 29(4), 536–555.

Lebret, R. and Collobert, R. (2015) '"The Sum of Its Parts": Joint Learning of Word and Phrase Representations with Autoencoders'. Available at: http://arxiv.org/abs/1506.05703.

Li, B. et al. (2018) 'An Adaptive Hierarchical Compositional Model for Phrase Embedding', (1), pp. 4144–4151. doi: 10.24963/ijcai.2018/576.

Li, W. (2017) 'Lowering the Barriers for Accessing Distributed Geospatial Big Data to Advance Spatial Data Science: The PolarHub Solution', Annals of the American Association of Geographers, 0(May), pp. 1–21. doi: 10.1080/24694452.2017.1373625.

Li, W. et al. (2011) 'Semantic-based web service discovery and chaining for building an Arctic spatial data infrastructure', Computers and Geosciences, 37(11), pp. 1752–1762. doi: 10.1016/j.cageo.2011.06.024.

Li, W., Bhatia, V., & Cao, K. (2015). Intelligent polar cyberinfrastructure: enabling semantic search in geospatial metadata catalogue to support polar data discovery. Earth Science Informatics, 8(1), 111-123.

Li, W., Cao, K., & Church, R. L. (2016). Cyberinfrastructure, GIS, and spatial optimization: opportunities and challenges. International Journal of Geographical Information Science, 30(3), 427-431.

Li, W., Goodchild, M. F. and Raskin, R. (2012) 'Towards geospatial semantic search: Exploiting latent semantic relations in geospatial data', International Journal of Digital Earth, 8947(September), pp. 17–37. doi: 10.1080/17538947.2012.674561.

Li, W., Li, L., Goodchild, M. F., & Anselin, L. (2013). A geospatial cyberinfrastructure for urban economic analysis and spatial decision-making. ISPRS International Journal of Geo-Information, 2(2), 413-431.

Li, W., Raskin, R., & Goodchild, M. F. (2012). Semantic similarity measurement based on knowledge mining: An artificial neural net approach. International Journal of Geographical Information Science, 26(8), 1415-1435.

Li, W., Song, M., Zhou, B., Cao, K., & Gao, S. (2015). Performance improvement techniques for geospatial web services in a cyberinfrastructure environment – A case study with a disaster management portal. Computers, Environment and Urban Systems, 54, 314–325. http://doi.org/10.1016/j.compenvurbsys.2015.04.003

Li, W., Wang, S. and Bhatia, V. (2016) 'PolarHub: A large-scale web crawling engine for OGC service discovery in cyberinfrastructure', Computers, Environment and Urban Systems. Elsevier Ltd, 59, pp. 195–207. doi: 10.1016/j.compenvurbsys.2016.07.004.

Li, W., Wu, S., Song, M., & Zhou, X. (2016c). A scalable cyberinfrastructure solution to support big data management and multivariate visualization of time-series sensor observation data. Earth Science Informatics, 9(4), 449–464.

Li, W., Yang, C., Nebert, D., Raskin, R., Houser, P., Wu, H., & Li, Z. (2011). Semantic-based web service discovery and chaining for building an Arctic spatial data infrastructure. Computers & Geosciences, 37(11), 1752-1762.

Li, W., Yang, C. and Yang, C. (2010) 'An active crawler for discovering geospatial Web services and their distribution pattern - A case study of OGC Web Map Service', International Journal of Geographical Information Science, 24(8), pp. 1127–1147. doi: 10.1080/13658810903514172.

Li, X., Di, L., Han, W., Zhao, P., & Dadi, U. (2010). Sharing geoscience algorithms in a Web service-oriented environment (GRASS GIS example). Computers & Geosciences, 36(8), 1060–1068.

Li, Z., Hodgson, M. E., & Li, W. (2018). A general-purpose framework for parallel processing of large-scale LiDAR data. International Journal of Digital Earth, 11(1), 26-47.

Li, Z., Yang, C. P., Wu, H., Li, W., & Miao, L. (2011b). An optimized framework for seamlessly integrating OGC Web Services to support geospatial sciences. International Journal of Geographical Information Science, 25(4), 595–613. https://doi.org/10.1080/13658816.2010.484811

Liu, K. et al. (2014) 'Using Semantic Search and Knowledge Reasoning to Improve the Discovery of Earth Science Records', International Journal of Applied Geospatial Research, 5(2), pp. 44–58. doi: 10.4018/ijagr.2014040104.

Liu, K., Gao, S., Qiu, P., Liu, X., Yan, B., & Lu, F. (2017). Road2Vec: Measuring Traffic Interactions in Urban Road System from Massive Travel Routes. ISPRS International Journal of Geo-Information, 6(11), 321.

Longueville, B. De, De Longueville Bertrand, B., Longueville, B. De, De Longueville Bertrand, B., Longueville, B. De, De Longueville Bertrand, B., & Longueville, B. De. (2010). Community-based geoportals: The next generation? Concepts and methods for the geospatial Web 2.0. Computers, Environment and Urban Systems, 34(4), 299–308. https://doi.org/10.1016/j.compenvurbsys.2010.04.004

Lopez-Pellicer, F. J. et al. (2011) 'Discovering geographic web services in search engines', Online Information Review, 35(6), pp. 909–927. doi: 10.1108/14684521111193193.

Manning, C. D. et al. (2014) 'The {Stanford} {CoreNLP} Natural Language Processing Toolkit', in Association for Computational Linguistics (ACL) System Demonstrations, pp. 55–60. Available at: http://www.aclweb.org/anthology/P/P14/P14-5010.

McCandless, M., Hatcher, E. and Gospodnetic, O. (2010) Lucene in action: covers Apache Lucene 3.0. Manning Publications Co.

McMillen, D. P., & McDonald, J. F. (1991). A Markov chain model of zoning change. Journal of Urban Economics, 30(2), 257–270.

Melamud, O., Goldberger, J. and Dagan, I. (2016) 'context2vec: Learning Generic Context Embedding with Bidirectional LSTM', Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, pp. 51–61. doi: 10.18653/v1/K16-1006.

Michaelis, C. D., & Ames, D. P. (2012). Considerations for Implementing OGC WMS and WFS Specifications in a Desktop GIS. Journal of Geographic Information System, 4(2), 161–167. http://doi.org/10.4236/jgis.2012.42021

Mihon, D., Colceriu, V., Bacu, V., & Gorgan, D. (2013). Grid based processing of satellite images in GreenLand Platform. International Journal of Advanced Computer Science and Applications, 3, 41–49.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems (pp. 3111-3119).

Miller, G. A. (1995) 'WordNet: a lexical database for English', Communications of the ACM. ACM, 38(11), pp. 39–41.

Mitchell, J. and Lapata, M. (2010) 'Composition in distributional models of semantics', Cognitive science. Wiley Online Library, 34(8), pp. 1388–1429.

Movva, S. et al. (2008) 'Customizable search engine with semantic and resource aggregation capability', in E-Commerce Technology and the Fifth IEEE Conference on Enterprise Computing, E-Commerce and E-Services, 2008 10th IEEE Conference on, pp. 376–381.

Oosterom, P. van. (2009). Research and development in geo-information generalisation and multiple representation. Computers, Environment and Urban Systems, 33(5), 303–310. http://doi.org/10.1016/j.compenvurbsys.2009.07.001

Papazoglou, M. P., & van den Heuvel, W.-J. (2007). Service oriented architectures: approaches, technologies and research issues. The VLDB Journal, 16(3), 389–415. http://doi.org/10.1007/s00778-007-0044-3

Patil, S., Bhattacharjee, S. and Ghosh, S. K. (2014) 'A spatial web crawler for discovering geo-servers and semantic referencing with spatial features', Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 8337 LNCS(year), pp. 68–78. doi: 10.1007/978-3-319-04483-5_7.

Pavlov, I. (2007). Lzma sdk (software development kit).

Pearce, D. G. (2001). An integrative framework for urban tourism research. Annals of tourism research, 28(4), 926-946.

Pebesma, E. (2012). spacetime: Spatio-Temporal Data inR. Journal of Statistical Software, 51(7). https://doi.org/10.18637/jss.v051.i07

Peng, X. and Gildea, D. (2016) 'Exploring phrase-compositionality in skip-gram models'. Available at: http://arxiv.org/abs/1607.06208.

Pennington, J., Socher, R. and Manning, C. (2014) 'Glove: Global vectors for word representation', in Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532–1543.

Pierce, M. E., Fox, G. C., Choi, J. Y., Guo, Z., Gao, X., & Ma, Y. (2009). Using Web 2.0 for scientific applications and scientific communities. Concurrency and Computation: Practice and Experience, 21(5), 583–603. https://doi.org/10.1002/cpe.1365

Purves, R. S., Medyckyj-Scott, D. J., & Mackaness, W. A. (2005). The e-MapScholar project—an example of interoperability in GIScience education. Computers & Geosciences, 31(2), 189–198.

Quah, D. T. (1993). Empirical Cross-Section Dynamics in Economic Growth. European Economic Review, 37(2-3), 426–434.

Rajib, M. A., Merwade, V., Kim, I. L., Zhao, L., Song, C., & Zhe, S. (2016). SWATShare – A web platform for collaborative research and education through online sharing, simulation and visualization of SWAT models. Environmental Modelling & Software, 75, 498–512.

Raskin, R. G. and Pan, M. J. (2005) 'Knowledge representation in the semantic web for Earth and environmental terminology (SWEET)', Computers & geosciences. Elsevier, 31(9), pp. 1119–1125.

Raup, B., Racoviteanu, A., Khalsa, S. J. S., Helm, C., Armstrong, R., & Arnaud, Y. (2007). The GLIMS geospatial glacier database: A new tool for studying glacier change. Global and Planetary Change, 56(1–2), 101–110. https://doi.org/10.1016/j.gloplacha.2006.07.018

Rey, S. J. (2001). Spatial empirics for economic growth and convergence. Geographical Analysis, 33(3), 195–214.

Rey, S. J. (2014). Python Spatial Analysis Library (PySAL): An update and illustration. In C. Brunsdon & A. SIngleton (Eds.), Geocomputation. Sage.

Rey, S. J. (2016). Space–Time Patterns of Rank Concordance: Local Indicators of Mobility Association with Application to Spatial Income Inequality Dynamics. Annals of the American Association of Geographers, 106(4), 788-803.

Rey, S. J., & Anselin, L. (2007). PySAL: A Python library of spatial analytical methods. The Review of Regional Studies, 37, 5–27.

Rey, S. J., & Anselin, L. (2010). PySAL: A Python Library of Spatial Analytical Methods. In M. M. Fischer & A. Getis (Eds.), Handbook of Applied Spatial Analysis: Software Tools, Methods and Applications (pp. 175–193). Berlin, Heidelberg: Springer Berlin Heidelberg.

Rey, S. J., & Janikas, M. V. (2006). STARS: Space-Time Analysis of Regional Systems. Geographical Analysis, 38(1), 67–86.

Rey, S. J., Anselin, L., Li, X., Pahle, R., Laura, J., Li, W., & Koschinsky, J. (2015). Open Geospatial Analytics with PySAL. ISPRS International Journal of Geo-Information, 4(2), 815–836.

Rey, S. J., Kang, W., & Wolf, L. (2016). The properties of tests for spatial effects in discrete Markov chain models of regional income distribution dynamics. Journal of Geographical Systems, 18(4), 377–398.

Rey, S. J., Murray, A. T., Grubesic, T. H., Mack, E., Wei, R., Anselin, L., & Griffin, M. (2014). Sex Offender Residential Movement Patterns: A Markov Chain Analysis. The Professional Geographer: The Journal of the Association of American Geographers, 66(1), 102–111.

Rinner, C., Keßler, C., & Andrulis, S. (2008). The use of Web 2.0 concepts to support deliberation in spatial decision-making. Computers, Environment and Urban Systems, 32(5), 386–395.

Robinson, A. H., & Cherry, C. (1967). Results of a prototype television bandwidth compression scheme. Proceedings of the IEEE, 55(3), 356–364. https://doi.org/10.1109/PROC.1967.5493

Sagl, G., Loidl, M., & Beinat, E. (2012). A visual analytics approach for extracting spatio-temporal urban mobility information from mobile network traffic. ISPRS International Journal of Geo-Information, 1(3), 256-271.

Sahin, K., & Gumusay, M. U. (2008). Service oriented architecture (SOA) based web services for geographic information systems. XXIst ISPRS Congress. Beijing, 625–630. http://doi.org/10.1.1.184.3921

Sato, M. et al. (2017) 'Distributed Document and Phrase Co-embeddings for Descriptive Clustering', Eacl, 1, pp. 991–1001. doi: 10.18653/v1/E17-1093.

Sayar, A., Pierce, M., & Fox, G. (2006). Integrating AJAX approach into GIS visualization Web Services. Proceedings of the Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services, AICT/ICIW'06, 2006, 169. https://doi.org/10.1109/AICT-ICIW.2006.114

Schut, P., & Whiteside, A. (2007). OpenGIS web processing service. OGC Project Document.

Shao, H., & Li, W. (2018). A comprehensive optimization strategy for real-time spatial feature sharing and visual analytics in cyberinfrastructure. International Journal of Digital Earth, 1-20.

Shao, H., Zhang, Y., & Li, W. (2017). Extraction and analysis of city's tourism districts based on social media data. Computers, Environment and Urban Systems, 65, 66-78.

Shen, J., Shi, J., Wang, M., & Wu, C. (2008). Building simplification algorithms based on user cognition in mobile environment. In L. Liu, X. Li, K. Liu, X. Zhang, & A. Chen (Eds.), (Vol. 7143, p. 71432T). http://doi.org/10.1117/12.812633

Smith, M. A., & Kollock, P. (Eds.). (1999). Communities in cyberspace. Psychology Press.

Socher, R., Manning, C. D. C. and Ng, A. Y. A. (2010) 'Learning continuous phrase representations and syntactic parsing with recursive neural networks', Proceedings of the NIPS-2010 Deep Learning and Unsupervised Feature Learning Workshop, pp. 1–9. doi: 10.1007/978-3-540-87479-9.

Song, M., Li, W., Zhou, B., & Lei, T. (2016). Spatiotemporal data representation and its effect on the performance of spatial analysis in a cyberinfrastructure environment – A case study with raster zonal analysis. Computers & Geosciences, 87(January), 11–21. http://doi.org/10.1016/j.cageo.2015.11.005

Steiniger, S., & Hunter, A. J. S. (2013). The 2012 free and open source GIS software map - A guide to facilitate research, development, and adoption. Computers, Environment and Urban Systems, 39, 136–150.

Stewart, C. A., Almes, G. T., & Wheeler, B. C. (2010). Cyberinfrastructure Software Sustainability and Reusability: Report from an NSF-funded workshop.

Stollberg, B., & Zipf, A. (2012). OGC Web Processing Service Interface for Web Service Orchestration Aggregating Geo-processing Services in a Bomb Threat Scenario. In Web and Wireless Geographical Information Systems (Vol. 47, pp. 239–251). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-76925-5_18

Sugumaran, R., Meyer, J. C., & Davis, J. (2009). A Web-based Environmental Decision Support System for Environmental Planning and Watershed Management. In Handbook of Applied Spatial Analysis (pp. 703–718).

Sun, Y., & Li, S. (2016). Real-time collaborative GIS: A technological review. ISPRS Journal of Photogrammetry and Remote Sensing: Official Publication of the International Society for Photogrammetry and Remote Sensing , 115, 143–152.

Swain, N. R., Latu, K., Christensen, S. D., Jones, N. L., Nelson, E. J., Ames, D. P., & Williams, G. P. (2015). A review of open source software solutions for developing water resources web applications. Environmental Modelling & Software, 67, 108–117.

Tobler, W. R. (1970). A computer movie simulating urban growth in the Detroit region. Economic geography, 46(sup1), 234-240.

Tukey, J. W. (1977). Exploratory Data Analysis. Pearson College Division.

U.S. Geological Survey and U.S. Department of Agriculture, Natural Resources Conservation Service, 2013, Federal Standards and Procedures for the National Watershed Boundary Dataset (WBD) (4 ed.): Techniques and Methods 11–A3, 63 p., https://pubs.usgs.gov/tm/11/a3/.

Veenendaal, B. (2015). USING THE GEOSPATIAL WEB TO DELIVER AND TEACH GISCIENCE EDUCATION PROGRAMS. ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XL-6/W1, 17–21.

Vretanos, P. A. (2004) 'OpenGIS Web Feature Service Implementation Specification Version 1.1. 0. OpenGIS Project Document 04-094'.

Wang, F. Z., Helian, N., Wu, S., Guo, Y., Deng, D. Y., Meng, L., … Parker, M. A. (2009). Eight Times Acceleration of Geospatial Data Archiving and Distribution on the Grids. IEEE Transactions on Geoscience and Remote Sensing: A Publication of the IEEE Geoscience and Remote Sensing Society, 47(5), 1444–1453.

Wang, F., Li, W., & Wang, S. (2016). Polar Cyclone Identification from 4D Climate Data in a Knowledge-Driven Visualization System. Climate, 4(3), 43. https://doi.org/10.3390/cli4030043

Wang, S. (2010). A CyberGIS framework for the synthesis of cyberinfrastructure, GIS, and spatial analysis. Annals of the Association of American Geographers, 100(3), 535-557.

Wang, S. (2013). CyberGIS: blueprint for integrated and scalable geospatial software ecosystems. International Journal of Geographical Information Science: IJGIS, 27(11), 2119–2121.

Wang, S. and Zong, C. (2017) 'Comparison Study on Critical Components in Composition Model for Phrase Representation', ACM Transactions on Asian and Low-Resource Language Information Processing, 16(3), pp. 1–25. doi: 10.1145/3010088.

Wang, S., & Armstrong, M. P. (2009). A theoretical approach to the use of cyberinfrastructure in geographical analysis. International Journal of Geographical Information Science: IJGIS, 23(2), 169–193.

Wang, S., & Liu, Y. (2009). TeraGrid GIScience Gateway: Bridging cyberinfrastructure and GIScience. International Journal of Geographical Information Science: IJGIS, 23(5), 631–656.

Wang, S., Anselin, L., Bhaduri, B., Crosby, C., Goodchild, M. F., Liu, Y., & Nyerges, T. L. (2013). CyberGIS software: a synthetic review and integration roadmap. International Journal of Geographical Information Science, 27(11), 2122-2145.

Wang, S., Armstrong, M. P., Ni, J., & Liu, Y. (2005, July). GISolve: A grid-based problem solving environment for computationally intensive geographic information analysis.

InChallenges of Large Applications in Distributed Environments, 2005. CLADE 2005. Proceedings (pp. 3-12). IEEE.

Wang, S., Wilkins-Diehr, N. R., & Nyerges, T. L. (2012). CyberGIS-Toward synergistic advancement of cyberinfrastructure and GIScience: A workshop summary.Journal of Spatial Information Science, (4), 125-148.

Wang, S., Zhang, J. and Zong, C. (2017) 'Learning sentence representation with guidance of human attention', IJCAI International Joint Conference on Artificial Intelligence, pp. 4137–4143. doi: 10.24963/ijcai.2017/578.

Weibel, R. (1997). Generalization of spatial data: Principles and selected algorithms. Algorithmic Foundations of Geographic Information Systems, 99–152. http://doi.org/10.1007/3-540-63818-0_5

Wieting, J., Bansal, M., Gimpel, K., Livescu, K., et al. (2015) 'From Paraphrase Database to Compositional Paraphrase Model and Back'. Available at: http://arxiv.org/abs/1506.03487.

Wright, D. J., & Wang, S. (2011). The emergence of spatial cyberinfrastructure. Proceedings of the National Academy of Sciences of the United States of America, 108(14), 5488–5491. https://doi.org/10.1073/pnas.1103051108

Wu, A., Convertino, G., Ganoe, C., Carroll, J. M., & Zhang, X. (luke). (2013). Supporting collaborative sense-making in emergency management through geo-visualization. International Journal of Human-Computer Studies, 71(1), 4–23.

Yang, C., Huang, Q., Li, Z., Liu, K., & Hu, F. (2017). Big Data and cloud computing: innovation opportunities and challenges. International Journal of Digital Earth, 10(1), 13–53. https://doi.org/10.1080/17538947.2016.1239771

Yang, C., Raskin, R., Goodchild, M., & Gahegan, M. (2010). Geospatial cyberinfrastructure: past, present and future. Computers, Environment and Urban Systems, 34(4), 264-277.

Yang, C., Wong, D. W., Yang, R., Kafatos, M., & Li, Q. (2005). Performance-improving techniques in web-based GIS. International Journal of Geographical Information Science, 19(3), 319–342. https://doi.org/10.1080/13658810412331280202

Ye, X., Li, W., & Huang, Q. (2018). A Synthesized Urban Science in the Context of Big Data and Cyberinfrastructure. In Big Data Support of Urban Planning and Management (pp. 435-448). Springer, Cham.

Yin, W. and Schuetze, H. (2014) 'An Exploration of Embeddings for Generalized Phrases', Acl-2014, pp. 41–47.

Yin, W. and Schütze, H. (2016) 'Discriminative Phrase Embedding for Paraphrase Identification'. doi: 10.1109/LSP.2014.2325781.

Young, T. et al. (2017) 'Recent Trends in Deep Learning Based Natural Language Processing', pp. 1–30. doi: arXiv:1708.02709v6.

Yu, J., Wu, J., & Sarwat, M. (2015, November). Geospark: A cluster computing framework for processing large-scale spatial data. In Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems (p. 70). ACM.

Yu, M. and Dredze, M. (2015) 'Learning Composition Models for Phrase Embeddings', Transactions of the ACL, 3, pp. 227–242.

Yue, Peng, Peter Baumann, Kaylin Bugbee, and Liangcun Jiang. "Towards intelligent giservices." Earth Science Informatics 8, no. 3 (2015): 463-481.

Zhang, C., & Li, W. (2005). The Roles of Web Feature and Web Map Services in Real-time Geospatial Data Sharing for Time-critical Applications. Cartography and Geographic Information Science, 32(4), 269–283. https://doi.org/10.1559/152304005775194728

Zhang, C., Zhao, T., & Li, W. (2013). Towards Improving Query Performance of Web Feature Services (WFS) for Disaster Response. ISPRS International Journal of Geo-Information, 2(1), 67–81. https://doi.org/10.3390/ijgi2010067

Zhang, J. et al. (2014) 'Bilingually-constrained Phrase Embeddings for Machine Translation', Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 111–121. doi: 10.3115/v1/P14-1011.

Zhang, T., & Tsou, M.-H. (2009). Developing a grid-enabled spatial Web portal for Internet GIServices and geospatial cyberinfrastructure. International Journal of Geographical Information Science, 23(5), 605–630. https://doi.org/10.1080/13658810802698571

Zhao, Y., Liu, Z. and Sun, M. (2015) 'Phrase Type Sensitive Tensor Indexing Model for Semantic Composition', Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence Phrase, pp. 2195–2201.

Zhou, Z., Huang, L. and Ji, H. (2017) 'Learning Phrase Embeddings from Paraphrases with GRUs', pp. 16–23. Available at: http://arxiv.org/abs/1710.05094.

Ziv, J., & Lempel, A. (1977). A Universal Algorithm for Sequential Data Compression. IEEE Transactions on Information Theory, 23(3), 337–343. https://doi.org/10.1109/TIT.1977.1055714

Ziv, J., & Lempel, A. (1978). Compression of individual sequences via variable-rate coding. IEEE Transactions on Information Theory, 24(5), 530–536. https://doi.org/10.1109/TIT.1978.1055934