POLITECNICO DI MILANO
Corso di Laurea MAGISTRALE in Ingegneria Informatica
Dipartimento di Elettronica e Informazione

Ontology Matching Enhanced with Similarity Measures for Georeferenced Datasets

ADVIS Research Laboratory
Advances in Data, Visual, and Information Science Research Laboratory

Relatore: Prof. Emanuele della Valle
Correlatore: Prof. Isabel F. Cruz

Tesi di Laurea di:
Claudio Caletti, matricola 782939

Anno Accademico 2013-2014

To my family

thanks for your love and support

Acknowledgments

I would like to thank my advisor, Prof. Isabel F. Cruz, both for giving me the opportunity to face a challenging and exciting problem, and for all the time spent discussing about possible solutions.

A thank you goes to Matteo Palmonari, for advising me in the meetings we had in Italy, and to Emanuele Della Valle, for supporting me when I was stuck with my work. Your suggestions have been an invaluable resource for my thesis.

I want to thank to all the guys of the ADVIS lab; especially Pavan Reddy and Venkat R. Ganesh, for the time spent together working on GIVA framework and the help they provided to me.

Thank to my friends, and thank to all the people who shared with me these last years of study. Thanks to `buildo` guys, whose enthusiasm and motivation tremendously contributed to my personal growth.

Finally, thanks Marta for bearing and supporting my staying in the United States.

Sommario

I dati geospaziali stanno cambiando sempre più il modo in cui interagiamo con il mondo. In particolare, comportano notevoli trasformazioni nel modo in cui viaggiamo, prendiamo decisioni e pensiamo nuovi prodotti. Il loro uso in campo applicativo è straordinariamente vario: spazia da divertimento e "social" (come twitter o foursquare) a studi ambientali e urbanistici.

Un settore emergente di particolare interesse per l'uso dei dati geospaziali è quello dello "urban metabolism". Il termine "urban metabolism" esprime la necessità di pensare ad una città nello stesso modo in cui si pensa ad un organismo vivente: una realtà complessa, composta dall'interazione di svariate sottocomponenti. L'analisi di una città richiede l'integrazione di diversi indici di salute come flussi d'acqua, materiali ed indicatori socio-economici. L'integrazione di concetti così eterogenei, insieme alla volontà di non risolvere il problema sviluppando modelli specifici ad un'unica situazione, è un obiettivo molto ambizioso, che richiede ricerca e sviluppo di nuove tecnologie. Al fine di contribuire a questo scopo, nell'ADVIS lab di Chicago abbiamo sviluppato GIVA, una piattaforma che permette ad utenti esperti di analizzare dati geospaziali in modo trasparente rispetto alle eterogeneità che li caratterizzano. Il fine di questa tesi è progettare e sviluppare le tecnologie necessarie allo sviluppo del cuore di tale sistema.

In particolare, il mio lavoro si concentra sul potenziamento di algoritmi di "ontology matching" per migliorarne l'efficacia nell'identificazione di similitudini fra ontologie di dati georeferenziati. L'obiettivo finale è quello di sviluppare tecniche che permettano di identificare una relazione qualsiasi fra i concetti rappresentati da dati georeferenziati: che essa sia di similitudine, inclusione o quant'altro. Per il momento ci concentriamo sul primo passo, ossia l'identificazione di corrispondenze fra entità simili.

Da un punto di vista tecnico, procediamo in due diverse fasi. La prima fase consiste nello sviluppo di una misura di similitudine per le istanze di ontologie di dati georeferenziati. La seconda fase consiste nella sua integrazione in un algoritmo di "ontology matching".

Se vogliamo confrontare diversi dataset dobbiamo prima ridurli ad una rappresentazione comune. Per questo motivo creiamo una griglia sopra lo spazio che vogliamo analizzare: discretizziamo lo spazio partizionandolo in un insieme di celle. Ad ogni cella assegnamo un valore ottenuto considerando le istanze in essa contenuta. Le singole istanze possono essere trattate in modo diverso, in base al concetto che rappresentano. Al fine di trovare il numero ottimo di celle per rappresentare il dataset in analisi abbiamo sviluppato una tecnica che coinvolge l'uso dell'autocorrelazione spaziale (l'indice di Moran). Per finire, confrontiamo le strutture dati così ottenute utilizzando l'indice di correlazione di Pearson.

La misura di similitudine descritta viene successivamente integrata in un algoritmo sintattico di "ontology matching", in modo da potenziarlo e renderlo più efficace nell'accoppiamento di ontologie di dati georeferenziati. Test su numerosi dataset dimostrano l'efficacia del nostro approccio. In particolare la metodologia descritta permette di trattare il "MAUP problem", che consiste nell'analisi di dati provenienti da unità amministrative di diversa natura (ad esempio province e regioni), e confrontare dati raccolti a diverse risoluzioni.

Per concludere, discutiamo diversi possibili sviluppi futuri, soppesando accuratamente pregi e difetti di ogni soluzione alternativa.

Summary

In this work we present a technique to improve the capability of the current data management systems to deal with geospatial data. In particular, we focus on enhancing ontology matching algorithms in order to make them more effective when identifying similarities between geospatial ontologies.
This work is meant to define the core techniques for creating a framework capable of identifying any kind of relationships between geospatial datasets.
We proceed following two steps: first, we define similarity measures for comparing the instances of geospatial ontologies; second, we integrate the result into a matcher.
To compare the datasets we create a tessellation to reduce them to a common format. Maximizing the spatial autocorrelation among the cells we are able to identify the tessellation that best expresses the degree of clustering of the data. Finally, Person's R is used as similarity measure to compare the distributions. We propose a few different ways to integrate the obtained similarity measure into an ontology matching algorithm.
We show the effectiveness of each of the used techniques with tests performed both on synthetic and real datasets. We also suggests how to compare datasets collected in different places in different time intervals. Our approach allows to address the MAUP problem and to integrate datasets having different resolutions.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Geospatial data are changing the way we look at the world. In particular, they involve two major transformations in how we do two things: make decisions and manage data[56]. geospatial data are also becoming increasingly important to improve the effectiveness of an application. The uses of such information are really diverse one to the other, ranging from everyday life to environmental studies. This diversity together with the complexity that involve the study of geospatial data requires new mechanisms to perform their integration and analysis.

Urban metabolism is an emerging field that well expresses the need and the issues involved by the integration of geospatial data[27]. Urban metabolism makes an attempt to put together not only the flows of water into a city, materials and nutrients, but also social, health and economic indicators[27]. This huge amount of data requires technologies to support their integration[12]. The main issue when dealing with geospatial data lies on their intrinsic heterogeneity. geospatial data presents heterogeneities in: format, resolution, spatial and

temporal representation, shape, and unit of measure. Designing a framework to deal with all these problems is incredibly complex, and requires great efforts in scientific research.

In the ADVIS Lab[1] we recently designed our framework called GIVA[12]. Given georeferenced datasets, GIVA addresses the problem of accessing them simultaneously and of establishing mappings between the underlying concepts, using automatic methods[12]. In my thesis, we aim at setting up the basics for the implementation of the core matching engine of such a system. We focus on two points: first, formalizing the problems that involve the integration of geospatial data; second, describing a very general way to address the problems.

1.2 Research Challenge

The first great problem we addressed is about the contextualization of our work with respect to the ocean of the publications related to the same subject. The problem of managing geospatial data is huge, and many researchers are doing several attempts to solve it. Therefore, we began by identifying the weaknesses of the current systems looking for areas where we could contribute.

First, we analyzed the most common systems for managing geospatial data, such as PostGIS[47] and QGIS[2] for understanding the limits of the spatial databases. Then, we looked at the technologies used by semantic storage systems for geospatial data. Considering the state of the art, we noticed that there are no studies about how to effectively matching georeferenced datasets. A lot of work has been done on improving the performances of storage systems, but those techniques lacks in consistency, when dealing with highly heterogeneous

[1] http://www.cs.uic.edu/Advis
[2] http://www.qgis.org/

dataset.

Therefore, the second problem we addressed is the classification of the possible heterogeneities in geospatial data. We identified several sources of heterogeneity: resolution, format, time and spatial representation, units of measure and so on. In Chapter 3, we provide a detailed description of those heterogeneities. In Chapter 4, starting the most simple situation and adding heterogeneities step by step, we are going to discuss techniques to identify similarities between georeferenced datasets.

1.3 Results

The result we obtained is setting up the basics for developing a system capable of identifying strong relationships between the concepts represented by georeferenced datasets. In this work we design a very general way to compare geospatial datasets addressing the problems of their heterogeneity in unit of measure, resolution and format. We also suggest how to compare datasets temporally and spatially disjoint. In this way, we laid the foundations for developing technologies to deal with more complex use cases.

1.4 Document Structure

This document is organized as follows. Chapter 2 contains a brief introduction to the technologies used in this thesis and also a discussion about the different approaches used in related works. Chapter 3 presents a detailed description of the problem and clarifies how we want to contribute. In Chapter 4 we start with an overview about the method we want to use to solve the problem, and then we provide a detailed description of the involved techniques. Chapter 5

contains both experiments to show the effectiveness of the used techniques and the description of the implementation in GIVA[12] and AM[14]. Finally, Chapter 6 summarizes the obtained results and discuss possible ways to continue with the work.

Chapter 2

State of the art

2.1 Semantic Web

The Semantic Web has been defined from the Semantic Web community[33] as "the extension of the World Wide Web that enables people to share content beyond the boundaries of applications and websites". The vision of the Semantic Web is the one of an intelligent web, where data are semantically rich and computers are capable of reasoning and analyzing those data. In the past, it has been considered in different ways: a "Web of Data", a "Utopic Vision" or also as a "natural paradigm shift" from the current status of the web[33].

The reason why the Semantic Web is often referred to as a "Web of Data" is simply because it is mainly concerned with data. Figure 2.1 shows the stack of the formats and technologies that enable the Semantic Web. Notice that the bottom half of the figure is strictly related to data. Syntax, ontologies, data interchange formats and querying languages are the core elements of the Semantic Web. This gives us an idea of the importance of processing and managing data.

The Semantic Web has also been described as a "Utopic Vision". The rea-

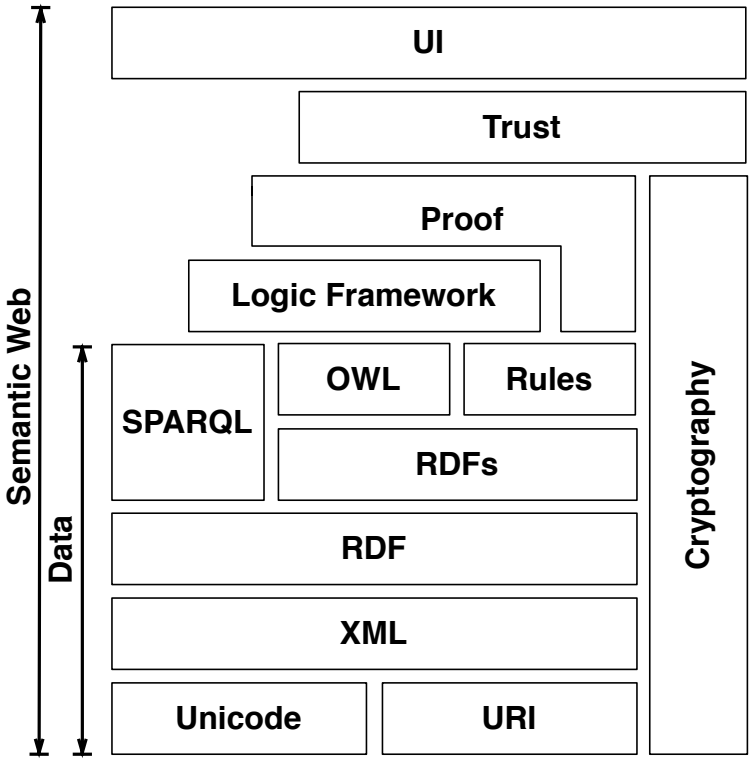


Figure 2.1: Stack of the semantic web (`http://www.w3.org/`).

son is that researchers in this field often deal with very hard problem. An intelligent web requires semantic richness and very general approaches, that are really difficult goals to achieve. Despite its complications, it is true that the web is getting more and more semantic and intelligent, as it is described by the Semantic Web.

The most important aspect of the Semantic Web is that it has inspired and engaged many people to create innovative semantic technologies and applications[33]. Quoting Tim Berners-Lee:

> I have a dream for the Web [in which computers] become capable of analyzing all the data on the Web - the content, links, and transactions between people and computers. A "Semantic Web", which

> makes this possible, has yet to emerge, but when it does, the day-to-day mechanisms of trade, bureaucracy and our daily lives will be handled by machines talking to machines. The "intelligent agents" people have touted for ages will finally materialize[1].

The term "Semantic Web" is often used to refer to the formats and technologies that enable it[33]. The collection and storage of semantic data are enabled by technologies that provide a representation of concepts in a given knowledge domain. The structuring of the information is fundamental for obtaining datasets semantically rich and easily exploitable by diverse applications. These technologies are formalized as W3C standards. The Semantic Web Stack, depicted in Figure 2.1, illustrates the components of the Semantic Web. The overall architecture can be summarized as follows[42]:

- XML is a markup language that provides a syntax for the content of semi-structured documents. XML does not provide any semantic to its content, and it is not a necessary technology for the Semantic Web. Another, very used, format expressing data is Turtle (Terse RDF Triple Language[2]). Despite the fact it has not been formally standardized yet, Turtle is de-facto the standard syntax language of the Semantic Web.

- RDF[3] is a standard model for conceptual description and the representation of information in the Web[36]. It is based upon the idea of making statements about resources in the form of triples: *subject-predicate-object*. It extends the linked structure of the Web using URIs to identify all the three mentioned elements. This allows structured and semi-structured

[1] Berners-Lee, Tim: The next web, at `http://www.ted.com/`
[2] `http://www.w3.org/TR/turtle/`
[3] `http://www.w3.org/RDF/`

data to be put together, exposed and shared. It can be represented in very different ways.

- RDF properties represent relationships between resources. However, RDF provides no mechanisms for describing these properties. RDFs[4] extends RDF and its vocabulary for describing properties and classes of the triples, with semantics for generalized-hierarchies.

- OWL[5] provides additional vocabulary along with a formal semantics. It introduces relations between classes, cardinality, equality, enumerated classes, richer typing and characteristics of properties.

- SPARQL[6] is a query language for semantic web data sources. SPARQL can be used to express queries whether the data is stored natively as RDF or viewed as RDF using a wrapper.

Now that the structure of the Semantic Web has been outlined, we need to answer an important question: where does "Ontology Matching Enhanced with similarity measures for Georeferenced Observations" fits in this context? The title of this thesis is mainly composed by two components: ontology matching and georeferenced observations. In Section 2.1.1 and Section 2.1.2 we discuss data integration and ontology matching, while in Section 2.2 and Section 2.2.1 and Section 2.2.2 we discuss the structure and importance of geospatial data and we also introduce datasets of georeferenced observations.

We already discussed the importance of processing and managing data, but we did not mention the underlying issue: on the web there are a huge amount of heterogeneous data. Many people upload data on the web using different

[4] http://www.w3.org/TR/rdf-schema/
[5] http://www.w3.org/TR/owl-features/
[6] http://www.w3.org/TR/rdf-sparql-query/

formats, languages and methods. It is incredibly important, and challenging, to put this information together, and create a unified view of them. This is the core enabler of an intelligent web.

2.1.1 Data Integration

The problem of data integration involves combining data coming from multiple data sources, providing the user with a unified vision of the data and detecting correspondences between similar concepts[39]. Data integration problems arise even in the simple situation of a unique, centralized databases, and it becomes more and more complex up to the extreme case of transient, dynamic, initially unknown data sources. Several techniques and methods deal with the problem, is such a way complication is added as the situation becomes more complex. Figure 2.2 shows an overview of the possible classifications of those complexities.

In the Semantic Web we deal with a particularly hard situation: data coming from multiple data sources without an a-priori global schema. With respect to Figure 2.2, we are either in case of P2P data integration or P2P with materialization. A major concern in Data Integration for the Semantic Web is linking similar concepts.

Linked Data is about using the Web to connect related data that was not previously linked, or using the Web to lower the barriers to linking data currently linked using other methods. This introduces a better way to share data between different people and organizations. Thanks to the LOD[7] effort we are getting closer to what we initially called the "Web of Data", that is a major milestone in realizing the Semantic Web vision.

[7] `http://linkeddata.org/`

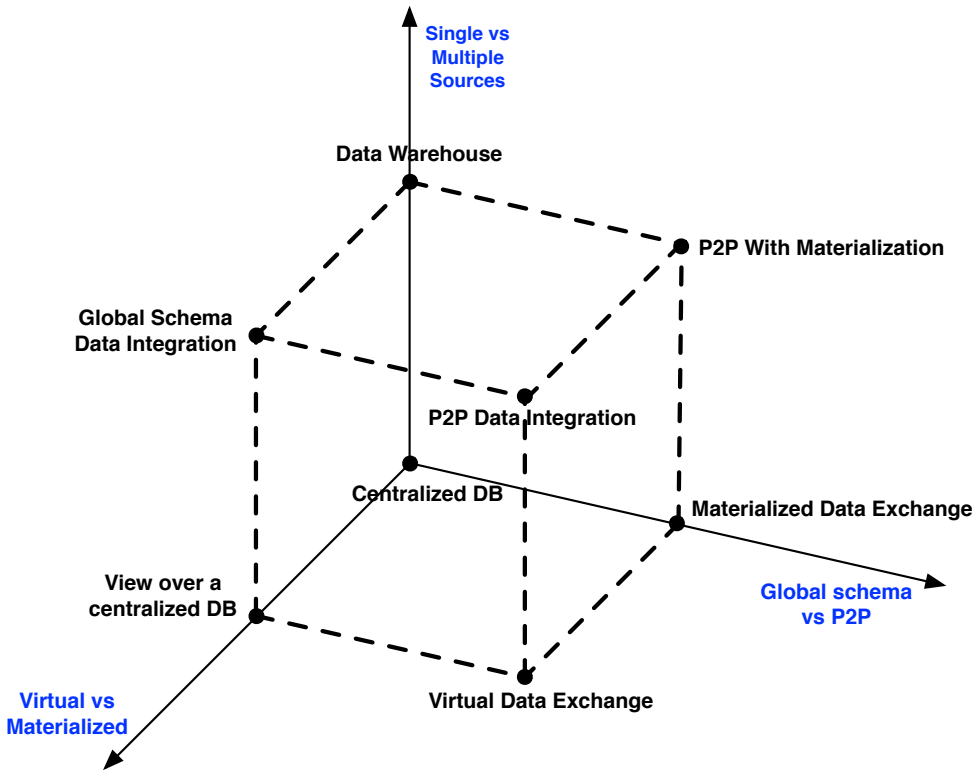


Figure 2.2: Data Integration complexity.

2.1.2 Ontology Matching

Ontology matching is "the process of determining correspondences between concepts in two different ontologies (the source and the target)"[66]. We refer to an ontology matching algorithm as to a matcher[14]. The strength of the mapping is given by a similarity value, that is a number in the interval $[0, 1]$ and that is measured considering heterogeneities of three kinds[14]:

- Syntactic heterogeneity (different models).
- Structural heterogeneity (same model, different organization).
- Semantic heterogeneity (same term, different meaning, or same meaning different terms).

AgreementMaker (AM) is an example of ontology matching tool[14]. AM is a software tool that is used to create semantic links between the global ontology and a local ontology and generate an agreement document, this document is used by the query processor that maps a query expressed in the terms used in the global ontology to the local ontologies[14].

Georeferenced datasets are a very particular kind of data to work with. In Section 2.2 we are going to explore them more deeply.

2.2 Geospatial Data

Using the term geospatial data we refer to datasets having an attribute describing the geographical feature of their instances. The geographical feature is the geographical representation of the instance. It can either be explicit, for instance a point or a shape, or implicit, for instance the name of a state. In this section we discuss why we are particularly interested in treating geospatial data, and we see what is their common structure.

2.2.1 The Importance of Geospatial Data

Geospatial data are changing the way we look at the world. A major transformation includes the way we do these two things: make decision and create maps. In the last few years, concepts of space and technologies, designed to leverage location information, have made huge advances in both two of these areas. The past decade has seen a complete change in how people are able to use and think geography[56].

Nowadays everybody has access to interactive maps, and moving in a big city easier than in the past. However, the revolution of geospatial data is much more than just moving from Point A to Point B. Geospatial data are having a

major influence in making decisions and analyzing problems using geography. Consider, for instance, applications such as *Yelp* and *TripAdvisor*. Today we can use them to find the best and the closest restaurants in the neighborhood. That question can be answered in just a few seconds and we can get directions to the destination almost immediately[56]. We can consider much more complex problems than this one. For instance, we might want to predict the impact of a natural disaster, or analyze the consequences of a business decision. Those problems requires the use of geography, and the capability of making sense of geospatial data.

One of the most impressive examples of the importance of geospatial data in decision making comes from the 1854, when John Snow depicted a cholera outbreak in London using points to represent the locations of some individual cases. His study of the distribution of cholera led to the source of the disease: a contaminated water pump[60]. Figure 2.3 shows the map depicted by John Snow.

A field in which the management of geospatial data is particularly interesting is urban metabolism. Urban metabolism involves the concurrent use of huge quantity of data coming from different sources. The idea is that the "increasing urbanization of human societies combined with intense energy demands of modern economies"[51] requires more powerful methods to study and analyze the behavior of the urban systems. It involves any kind of data that, somehow, is helpful in describing the life of a city. For instance, it involves the flow of water into a city, materials and nutrients. However, it is also related to social, health and economic indicators[27]. Urban metabolism not only emphasizes the importance of geospatial data, but also well expresses the need and the challenge represented by their integration. The problem here goes far beyond than simply identifying similarities between the concepts: we

Figure 2.3: Clusters of cholera observations by John Snow[60].

need to compare and put together things that are completely different in terms of representation, format and concept.

2.2.2 The Structure of Geospatial Data

Geospatial data are represented in a variety of different ways. However, whatever the data, it begins measuring a location. The most widely used method for measuring location nowadays is through the GPS. GPS is used to create any kind of geospatial data.

Using GPS it is possible to record positions of objects. We can put together sets of single positions to create lines and polygons. The way we represent spatial data is a fundamental block in GIS. This kind of data are called "vector data". The idea behind the name is simply that those data represent geometri-

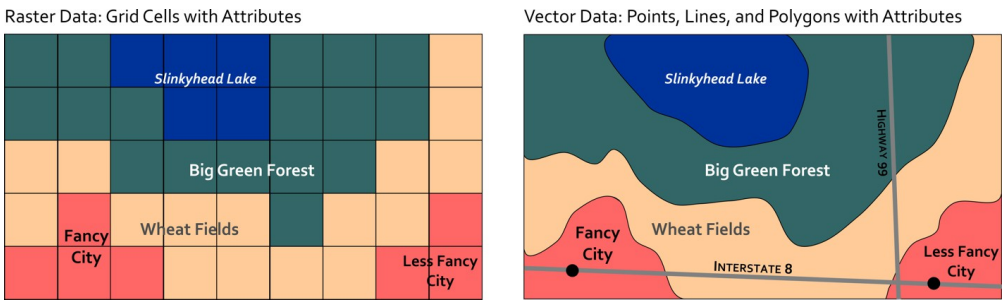


Figure 2.4: Vector and raster data[56].

cal figures. The other major data type is raster data.

Geographic image data are called "raster data", which captures information by assigning values to cells in a grid. A satellite in space photograph the earth and assigns values to each grid cell to develop an image. The size of those grid cells has an impact on the resolution of the final image. Tools like *Google Earth* have made imagery of the Earth more accessible than ever. However, in this work, we are mainly going to deal with "vector data". The reason why we consider "vector data" is that the vectorial representation is much more general. As a matter of fact, it is possible to convert "raster data" to "vector data"[8].

A very important source for geospatial data in the United States is the U.S. Census Bureau. At each population census the Census collects data at different resolutions. In Chapter 4 we will briefly mention about the dataset organized by administrative units.

From a Semantic Web perspective we are mainly interested in ontologies of geospatial data. Geospatial ontologies are characterized by three elements:

- Spatial component (*geo* in Figure 2.5).
- Metadata (*tags* in Figure 2.5).
- Descriptive attribute (*value* in Figure 2.5).

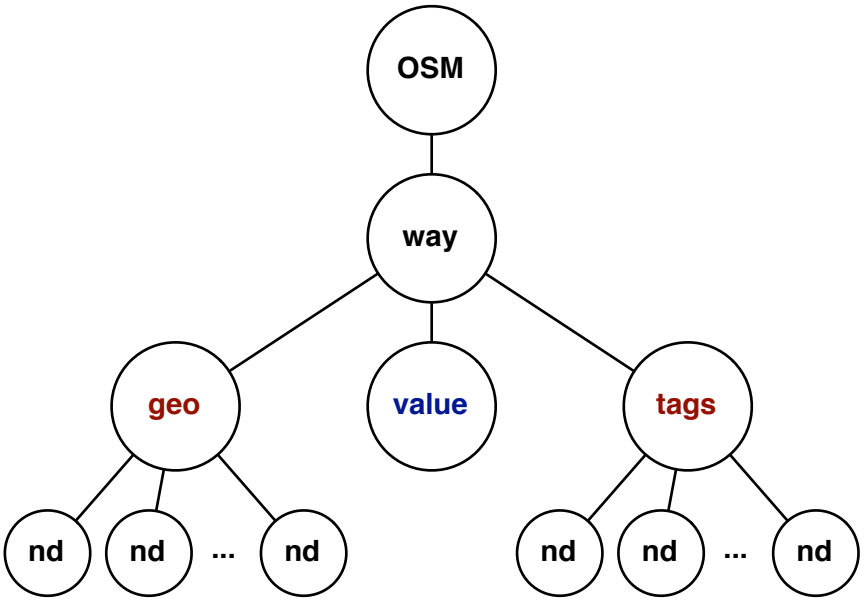


Figure 2.5: Example of geospatial ontology.

Figure 2.5 shows a general example of geospatial ontology extracted from Open Street Map[24] and converted from the XML/RDF representation to a tree. In the picture, red nodes are attributes that really exists in open street map, while the blue node, the descriptive attribute, is added by me. It is very important for our purpose. The descriptive attribute is not always present in geospatial data. We are going to call those datasets that have a descriptive attribute georeferenced datasets. In section 2.2.3 we are going to discuss georeferenced datasets and define their relationship with geospatial data. In particular, in this work we are mainly going to deal with datasets of points, and not generic shapes. This allows us to focus on the comparison between the datasets instead of dealing with the fact they are represented in different ways. However, the described techniques are general enough to be extended to deal with any kind of shape: a more detailed discussion and the implementation of this improvement is left as future development.

Geospatial data can be extended with a temporal feature. The temporal feature can be represented with really different granularities. For instance, data could be recorded every second, once a week or once a year. Time adds great complexity to the problem of the integration of geospatial data. However, the data represented in geospatial datasets are often time-dependent, and it would be very unwise to avoid considering time. Dealing with geospatial data is particularly difficult due to their heterogeneity of their possible shapes and units of measure, the level of resolution of the data. In this work we focus on how to use two of the three described features, the geospatial feature and the value, for our purpose of finding similarities between heterogeneous datasets. Also metadata are a very important source of information for performing alignment between different datasets[22].

However, in this work we do not develop any new method to use them in order to improve our performances in matching geospatial datasets, and thus we consider them just as simple nodes of the ontology that can be either linked or not depending on the decision of the matcher.

2.2.3 Geospatial Data and Georeferenced Observations

So far we only discussed geospatial data, but in the title and throughout the rest of the thesis we refer to georeferenced observations. The reason is because in this work we are not considering any possible kind of geospatial data. We just focus on those datasets that have related values, those that are quantifiable.

For instance, we will be discussing datasets about rainfall, population, car crashes and so on, but we are not going to talk about comparing the shapes of two roads, or two rivers. Figure 2.6 shows the relationship between geospatial data and georeferenced observations. Generally we call geospatial data all the datasets with a geospatial feature. Since also georeferenced observations

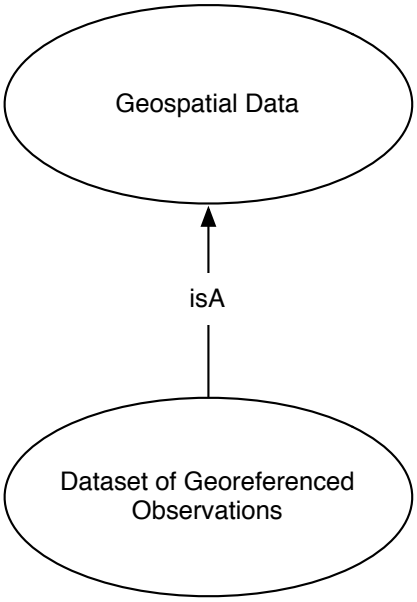


Figure 2.6: Hierarchy of geospatial data.

require a geospatial feature they are a subclass of geospatial data. However, since they require a further constraint, i.e. an attribute representing a value, they are a more specific class with respect to geospatial data.

2.3 Related Works

Over the past decade there have been many different attempts to create a standard semantic representation for geospatial data[4]. In 2003, a W3C Semantic Web Interest Group created the Basic Geo Vocabulary[4,63] which provided a way to represent WGS84 points in RDF. This work was further extended in 2007 to obtain an OWL representation of geospatial data[4,64].

A major issue when storing geospatial semantic data is about performances: without an appropriate indexing a search would require a sequential scan of every instance in the database. The first geospatial index for Semantic Web

data was used by Parliament[30], a high-performance triple store designed for the Semantic Web, in the 2007. There have been developed other triple stores such as Ontotext's OWLIM-SE[8] and OpenLink Virtuoso[9] that supports W3C Basic Geo Vocabulary. All these systems provide fundamental components in the evolution of Geospatial Data management, but they do not provide any feature for integrating them. Other than storing geospatial data, there have also been interest in developing new techniques for querying them. For instance, Perry proposed an extension of SPARQL to SPARQL-ST for complex spatio-temporal queries[4,49]. Other more trivial attempts to provide SPARQL with topological predicates are proposed by Battle and Kolas[4], Xiao, Huang, and Zhai[70], Zhai, Huang, and Xiao[71]. An interesting approach is described by Koubarakis and Kyzirakos[32], who propose stSPARQL to extend SPARQL to includes additional operators for querying RCC[53] relationships and introduces a new syntax for specifying spatial variables[4]. A relevant emerging standard is Geo-SPARQL[11] from the Open Geospatial Consortium (OGC) [10]. This standard aims to resolve issues in geospatial data representation and access[4].

A more recent attempt to describe geospatial data has been done by Salas, Harth, Norton, Vilches, León, Goodwin, Stadler, Anand, and Harries[58], 2011. Starting from the observation that "no consense had been achieved for developing an RDF vocabulary with enough descriptive power to satisfy most requirements of these datasets", Salas et al.[58] tried to define a common vocabulary for representing geospatial data exposing different ways to serialize them enhancing the compatibility with GIS systems. Another interesting approach to improve the current standard in managing semantic geospatial data

[8] http://www.ontotext.com/owlim
[9] http://www.w3.org/2001/sw/wiki/OpenLink_Virtuoso
[10] http://www.opengeospatial.org/

comes from Battle and Kolas[4], who tried to update the triple store Parliament to support GeoSPARQL. In 2012 Kyzirakos, Karpathiotakis, and Koubarakis[35] presented a new version of the data model stRDF and the query language stSPARQL. They also implemented a new system called Strabon which implements those new functionalities.

That said, our main interest lies on the integration of geospatial data, that we did not discuss so far. TELEIOS is an example of project for real-time monitoring using semantic web and linked data technologies[31]. Its goal is the effective discovery of knowledge contained in them[34]. A more recent attempt to browse and make sense of geospatial data coming from different sources is Sextant[45]. Sextant is a web tool that, quoting its authors: "enables exploration of linked geospatial data as well as creation, sharing, and collaborative editing of thematic maps by combining linked geospatial data and other geospatial information available in standard OGC file formats"[12,45].

It focuses on two tasks: querying different RDF storages and manipulating geospatial data. In GIVA, we aim at doing something more: We want to create "a semantic framework that assists domain experts in integrating highly heterogeneous datasets and in analyzing and visualizing dependencies among them"[12]. The main difference lies in the semi-automatic integration of geospatial data. In order to do that, we strongly rely on ontology matching, that is discussed in Section 2.1.2. Our focus is on creating links between similar concept represented by georeferenced datasets. Isaac, Van Der Meij, Schlobach, and Wang[25] showed that instance-based matching has excellent results when applied on general ontologies, just by using simple similarity measures. Geospatial data are a peculiar type of data to work with, and thus we need to develop ad hoc measures for comparing them.

In literature it is possible to find several attempts of comparing geospatial

features, that involve techniques for the comparison between the geographical features of different datasets. For instance, Dunkars[19] presents a method to link objects that represent the same real world feature. Walter and Fritsch[65] also propose a statistical approach for comparing the geospatial features of heterogeneous geospatial datasets. Duckham and Worboys[18] explore the use of instance-level (extensional) information within the fusion process through the classification of the areas composing a raster image. All those works focus on matching geospatial datasets; however, we are interested only in a particular class of geospatial datasets: georeferenced datasets. Matching geospatial features and matching georeferenced datasets are processes that involve different techniques. When analyzing geospatial features the focus is on the shape, while our focus is on the distribution of the dataset.

The result we want to achieve is to develop the core engine for a framework capable of integrating geospatial data. In order to achieve this result, we need an effective way to compare geospatial data.

Chapter 3

Problem Definition

3.1 Goal

Our final goal is to link geospatial ontologies, identifying relationships between their main concepts. Figure 3.1 qualitatively shows the result we want to achieve. We proceed working on two paths: first, we would like to link similar concepts with a high degree of confidence; second, we would like to identify "strong" relationships (for instance, inclusion) between the related concepts.

Let us suppose, for instance, you have multiple datasets about the same concept in your data storage system: if you are able to identify those datasets, you can merge them in a unique one saving space on your disk. An automatic system to perform this task can significantly improve the efficiency of your storage. Otherwise, suppose you have two different datasets: one about "rainfall" and the other one about "snowfall". If you are able to understand that your concepts are both subclasses of a unique class called "precipitation", you can add them in order to create a new dataset about precipitation.

The creation of links between the datasets is just the first step: it is the enabler for analyzing and reasoning about the data. Recalling the stack of the

Semantic Web in Section 2.1 we are working with its lower part: the data part. Currently there are no works describing instance-based approaches to ontology matching algorithm for georeferenced datasets. As discussed in Section 2.1.2 the state-of-the-art systems rely only on structural and syntactical matching to link datasets. In Section 3.2 we are going to see why these methods are not so effective in our case. In this work we aim at creating an instance-based method to match geospatial ontologies. In this way we aim at creating breeding ground for working on how to identify more complex relationships.

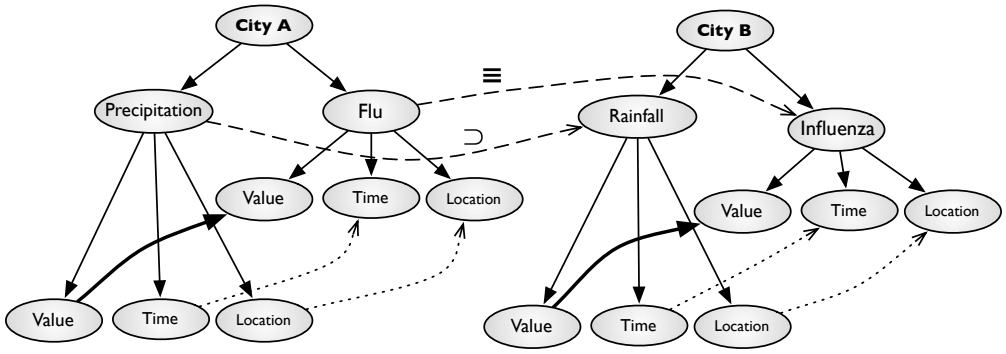


Figure 3.1: Linking geospatial ontology[12].

3.2 Limits of Structural and Syntactical Matching

In Chapter 2 we have discussed geospatial data, Figure 3.2 shows a real example of a geospatial ontology. We already said that geospatial ontologies are characterized by having: a latitude, a longitude, a value and several metadata. Ontology matching works effectively when comparing ontologies of data rich of semantic information and using a shared vocabulary.

The structure shown in Picture 3.2, for instance, is not really different from the ones extracted from flat database tables.

Since the structure we are dealing with is so simple and predictable, running a structural matcher between two ontologies of this type is unlikely going to return a trustworthy alignment. In fact, a structural matcher is probably just going to boost the matching between the root concepts (even if there is no real reason to match them). For example, suppose you want to compare two datasets: the first about rainfall with latitude and longitude as attributes, and the second about car crashes with latitude and longitude as attributes. Since rainfall and car crashes have very similar structures, it is possible that a structural algorithm will match car crashes with rainfall. In this case a syntactical

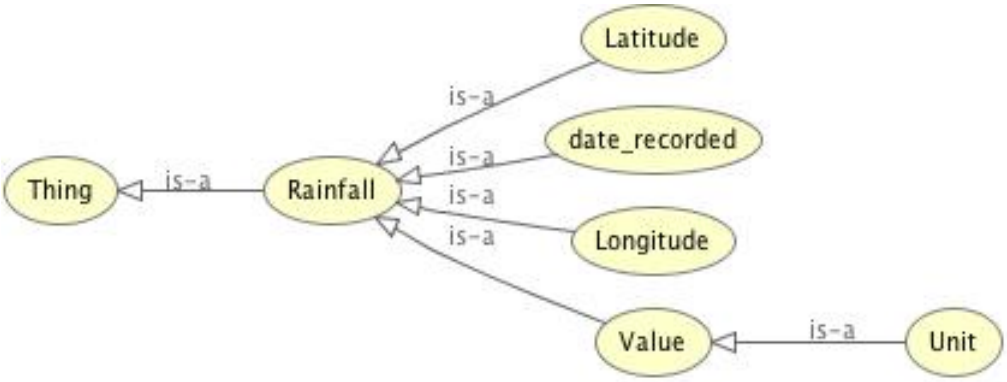


Figure 3.2: Example of flat geospatial ontology[46].

matcher works better. It allows to identify a link between two concepts if they have similar names or if they are synonyms (if we suppose the matcher uses WordNet[43] or similar systems). However, there are situations when the syntactical matcher fails in creating a correspondences between similar concepts.

A matcher, working at syntactical level, would be probably able to link similar simple attributes. Our problem is that it would fail in matching more complex concepts. For instance, suppose you are an analyst who wants to compare a *precipitation* dataset with a *rainfall* dataset. During the summer, when there is no snowfall, reasonably that the two datasets are about the very

Table 3.1: TYPES OF HETEROGENEITIES

Heterogeneity	**Examples of Possible Representations**
Unit of measure	Meters, Centimeters, Inches
Resolution	Country, State, County
Time Representation	Timestamp, Date, Period
Space Representation	Shape, Point, Square
Format	SHP, KML, CSV

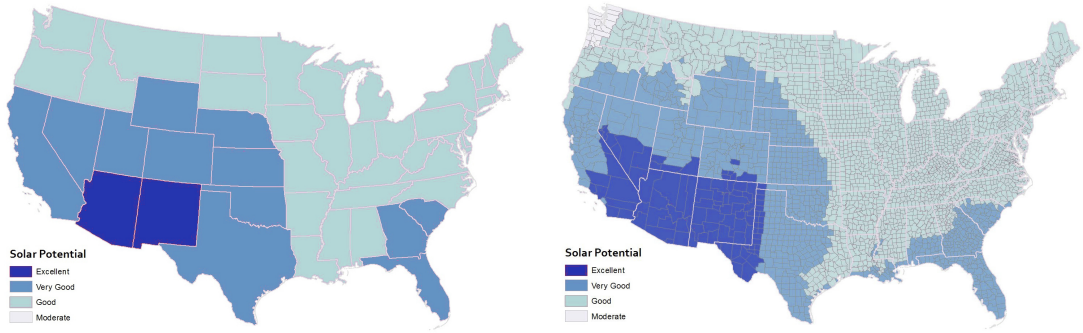
same concept, and you would like an algorithm to link them.

A standard ontology matching algorithm would not be able to do it. Since the structure of the datasets is not going to help us to achieve our goals, and syntactical matching has a limited scope, we decided to proceed with an instance-based ontology matching approach. The instances of the datasets exploit more information we can use to improve the current technologies.

In Section 3.3 we are going to explore the main problem related to geospatial data: their heterogeneity.

3.3 Heterogeneity

Dealing with the instances of geospatial datasets provides us with more information with respect to other datasets, but it involve some drawback. The problem of geospatial data is that they are inherently highly heterogeneous. Geospatial data can be represented with different formats, they can have different resolutions, units of measure and so on. Table 3.1 shows a detailed summary of the kind of heterogeneities we might deal with when using geospatial data.



(a) Solar potential by state. *(b) Solar potential by county.*

Figure 3.3: MAUP problem[56].

3.3.1 Resolution and MAUP Problem

A major pitfall when analyzing geospatial data relates to the resolution of the data. With the term "resolution" we refer to the scale used to represent the data. For instance, data organized by county are at a higher resolution with respect to data represented at state level. A dense dataset of georeferenced observations has an higher resolution with respect to a the same dataset if represented by a few points.

Depending on the scale at which you look at a Geographic pattern, you can derive completely different results from the exact same underlying data[56]. This is called the MAUP[68].

Figure 3.3 shows an example of this problem. The data shown in the picture are about the solar potential analyzed by state and by county. The problem of the resolution is related to the representation of the datasets: if you compare two datasets at different resolutions you need to convert them to an appropriate common representation.

3.3.2 Datasets Disjoint in Space and Time

Your datasets might be recored in different places and in different time intervals. A possible classification of this kind of heterogeneity is shown in Figure 3.4. Clearly, the way the datasets are disjoint strongly influences the techniques you can use to analyze them. In Chapter 4 we will show differ-

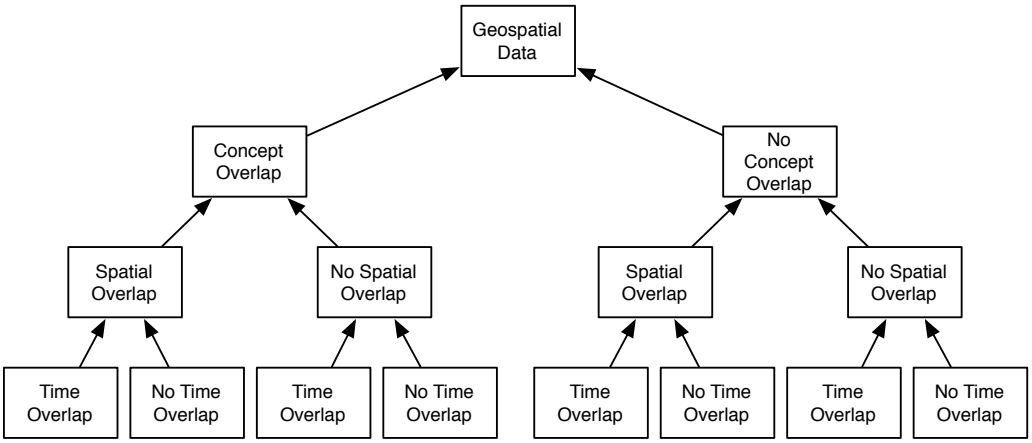


Figure 3.4: Hierarchy of the possible relations between geospatial datasets.

ent approaches to compare geospatial datasets. Starting from the most simple case, we try to extend our method to deal with more complex situations.

3.3.3 Uncertainty

Geographic locations can include administrative units like states and counties, natural areas like forests and lakes that can sometimes be formally defined by their observable features, and cultural regions with uncertain boundaries like neighborhoods. Understanding people's conception of geospatial entities is not easy. The example shown at Figure 3.5 by Andy Woodruff and Tim Wallace at `http://www.bostonography.com` shows how people in Boston perceive of their city's neighborhoods. It is obtained using crowdsourcing. It is imprecise,

and parts of the map are empty. This is a much more faithful representation of what we can actually know about these types of places than the neat and tidy borders we can define for administrative units[69]. This picture effectively

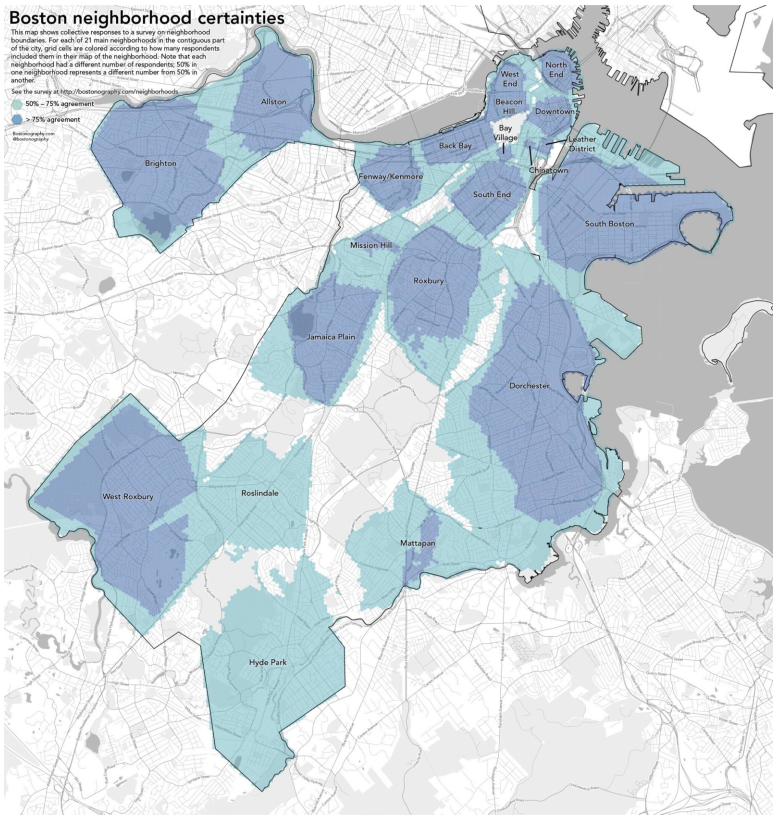


Figure 3.5: Neighborhoods boundaries of Boston[69].

gets the reader across the uncertain nature of geospatial data. However, the conception of the geospatial entities by people is not the only or the major source of uncertainty. Uncertainty is generated in different steps during the life of a dataset. Measuring a phenomena generates uncertainty, the placement

of the sensors to measure a phenomena generates uncertainty, and so on. The fact that uncertainty is so widespread in geospatial data is not a real problem for us, and we are not going to discuss this problem. However, we need to be aware of it.

3.4 Conclusions

To conclude, we want to improve the capability of the current matchers to link geospatial entities. For this purpose, we proceed by rethinking about how to build an ontology matching system, so that the analysis of the instances is included.

In order to do that, we need to address the different heterogeneities that we might occur when analyzing geospatial data. In Chapter 4, considering the different types of heterogeneities, we are going to propose a process to solve this problem.

Chapter 4

Problem Solving

4.1 Overview

In this section we provide the reader with an overview of our solution to the problem. In Subsection 4.1.1 we contextualize our solution and in Subsection 4.1.2 we describe the design of the system.

4.1.1 Contextualization

We have seen that the problem of integrating geospatial data is very complex and faceted. For these reason, before to start discussing a solution, we need to contextualize our approach with respect to the other possibilities. We start by discussing the differences between semantic data integration" and "instances integration". With "semantic data integration" we refer to the integration performed at concept level, while with "instances integration" we refer to the integration performed at instance level. Considering a data integration terminology, "semantic data integration" is similar to "virtual data integration" and "instances integration" is similar to "materialized data integration". In this

work we want to achieve "semantic data integration": we want to create links between similar concepts, not to obtain a unique dataset from heterogeneous sources. "Instance integration" can be obtained by merging two datasets that are known to be about the very same concept, however, it is not our main goal. For our purpose, "instance integration" can be seen merely as a possible opportunity enabled by the "semantic data integration".

That said, it does not mean that we are not going to use instances; on the contrary, as mentioned in Chapter 3, we are going to use an Instance-Based Approach, that of course involves the use of instances. To conclude, we are going to use instances to find useful semantic information about the concept represented by the dataset.

Another major clarification that needs to be done is about the type of datasets we are going to consider. We already discussed in Chapter 2 the difference between geospatial data and georeferenced data, and we pointed out that we are going to deal with georeferenced data. The most common class of georeferenced datasets is the one of datasets having a point as geographical feature. For this reason, we are mainly going to deal with "datasets of points". However, there are also other very important geospatial datasets that are represented as shapes: despite we are not going to use them explicitly, we need to develop an approach that can be easily adapted also to work with such datasets.

We also said that a major issue that arises in treating geospatial instances involves their heterogeneity in space and time. In order to deal with this problem, we start focusing on the most simple possible situation: datasets in the same area, at the same time. We are also going to suggest possible ways to extend this approach in time and space, so that we include a wider range of possible use cases.

4.1.2 System Design

We aim at enhancing the current architecture of an ontology matching algorithm in order to work better with georeferenced datasets. Figure 4.1 shows the standard architecture of a matcher.

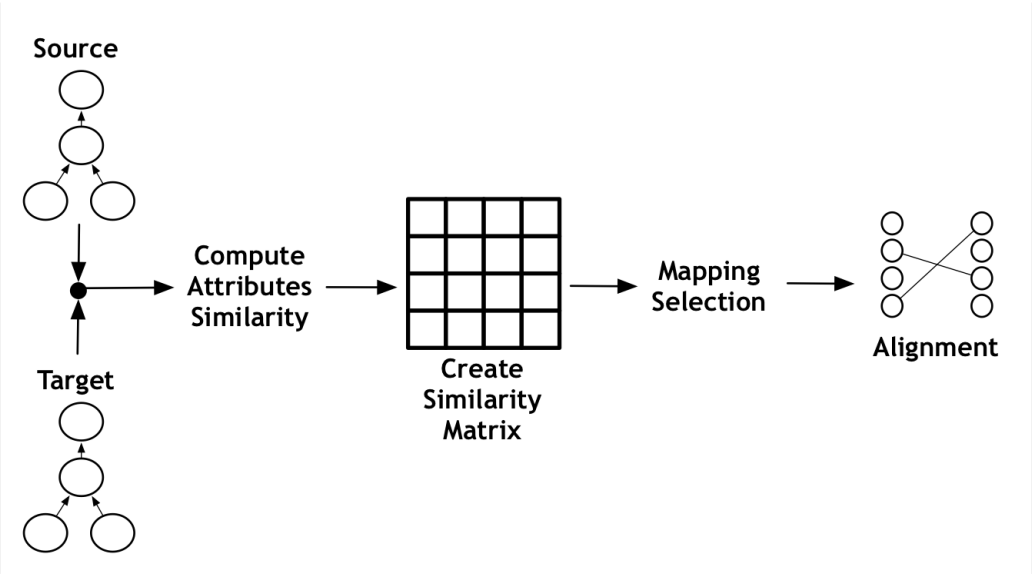


Figure 4.1: General matcher architecture[14].

Our approach consists in using instance-based matching to overcome the limits of syntactical matchers. Useful information we can extract from georeferenced datasets is the distribution of the data over the space. The distribution of a georeferenced dataset is obtained using three attributes: latitude, longitude and value. Since the distribution feature is not going to be present in the final alignment, unless it was already present in the initial ontologies, we decided to explicitly add it to each of the compared ontologies. The distribution attributes added to the ontologies are obviously guaranteed to be the same entity. For this reason, we create a correspondence between the added nodes. We decided to assign to the correspondence a similarity measure based on the similarity of the distributions. For the moment, the process of adding

a distribution feature can be seen as a post-processing step performed after a standard ontology alignment. The desired result is shown in Picture 4.2.

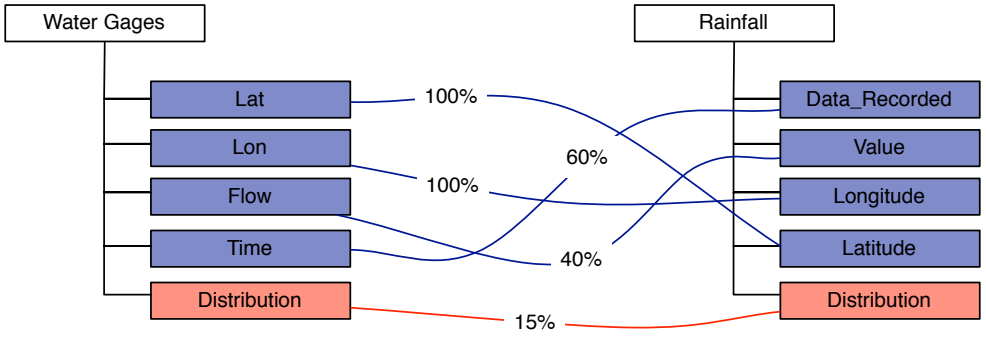


Figure 4.2: Addition of distribution.

What we obtain is not a standard ontology alignment for two reasons:

1. We always have a correspondence between the distributes nodes. It is not possible to link other attributes to them.

2. We need to use an ad-hoc method to obtain the similarity value between the distribution nodes.

That said, why is that representation appropriate?

First, the obtain alignment provides useful information, that can be further used for data integration. It is fine to modify the meaning of the alignment provided that we obtain an improvement.

Second, the obtained alignment can be used as an intermediate step of an enhanced standard ontology matching system. Under the assumption that "Datasets with a similar distributions are likely going to be about the same concept, datasets with dissimilar distributions are unlikely going to be about the same concept", we can use the similarity between distributions to obtain a similarity measure between the root concepts.

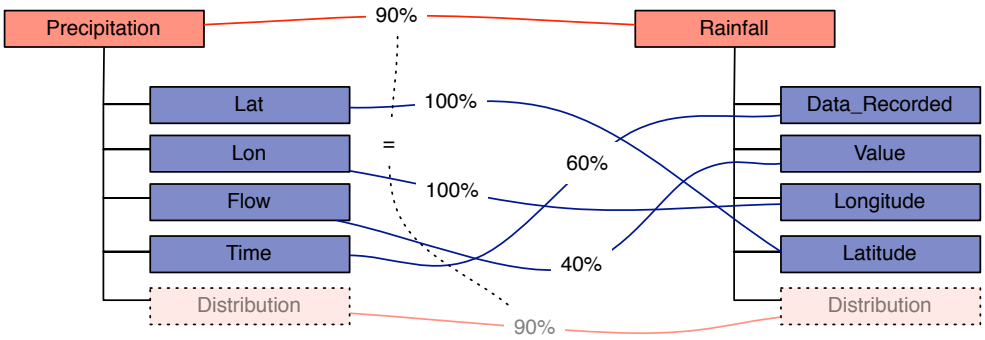


Figure 4.3: Enhanced ontology matching.

An example is shown in figure 4.3. We already discussed the fact that a structural matcher is not going to behave really well in our case, since our ontologies are too poor from a structural point of view.

However, when using a structural matcher, an improvement performed on a link is going to influence also the other links. After an improvement in a link we obtain a new similarity matrix, and we need to iterate again to identify the best mappings. For this reason, we can stop considering this approach as merely a post-processing step.

Figure 4.4 shows the overall architecture of the obtained new ontology matching algorithm.

4.2 Data Processing

In this section we discuss three major steps we need to process geospatial data: data extraction, ontology extraction and data translation[12]. We begin discussing data extraction. Processing geospatial data is a fundamental step for their integration. Even though in this work we are not going to contribute with new techniques we describe in detail the once we intend to use.

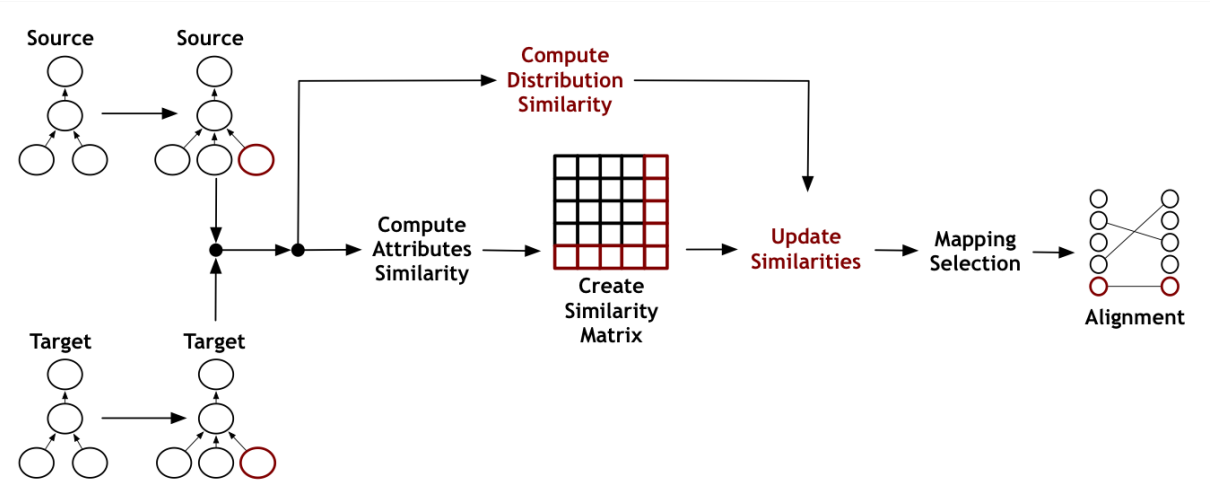


Figure 4.4: Enhanced matcher architecture.

Geospatial data, as we can see in Figure 4.5, are represented in different formats. The geospatial feature in these data formats uses geodetic systems such as WGS84 and geometric objects (e.g., polygon, polygonal chains and so on)[12].

However, geospatial data are also often represented in unstructured or semi-structured formats, for instance web tables or text, and thus they need an ad-hoc processing. For example, quoting our GIVA paper, "web tables are primarily constructed using the ¡table¿ tags for a variety of purposes such as, HTML forms, calendars, page layout, and relational data"[12]. This information can be exploited to automatize the process of extraction of metadata. In many cases web tables are poor in features, and they do not contain easily extractable metadata and, in order to extract the corresponding feature-rich tables, we need to first identify the headers (which are sometimes nested) and then store it, together with the table, in structured file[13]. For this kind of

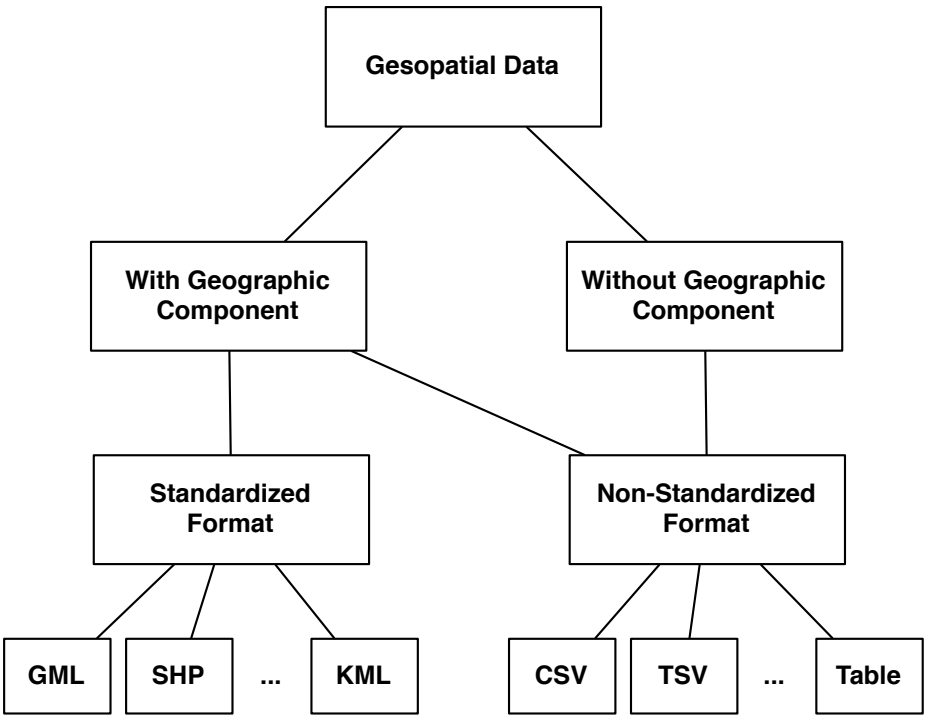


Figure 4.5: Hierarchy of the formats of geospatial data[12].

extraction we can use machine learning approaches: in GIVA, for instance, we use a decision tree classifier model (C4.5)[52] using 20 different heuristics (including number of columns, rows, font size, and color) and trained on 100 web tables with geospatial data[12].

Another major problem when processing geospatial data involves data translation, i.e. the process of translating data from one format to another. Before we attempt to create geospatial mappings between these data, they need to be translated into a common spatial data format. Generally we can use GDAL[67] to convert datasets from a standard format to another. For instance, we can use GDAL to convert datasets from GML to PostGIS dump format. In this case, we face with the problem that non-standardized formats require semantic processing to identify the appropriate column headers that contain information

about spatial coordinates, timestamps and values. We decided to simply use string matching on the attributes in order to identify longitude, latitude and value. Next in this chapter, we are going to work under the assumption that those attributes are correctly identified.

Further, data in non-standardized formats may contain implicit geographic components (e.g., the word "Illinois")[12]. Special processing and techniques are required to identify these implicit geographic components. The hierarchical characteristics of geospatial classification schemes can be modeled using a part-of or is-a relationship [15]. We can also use methods to extract ontologies from a variety of formats, including relational tables, XML, and RDF documents considering a global ontology [17].

So far we only discussed datasets of points, but there are other types datasets we might be interested in. In particular, we would like to deal with datasets organized in administrative units. An example of dataset organized in administrative units is a dataset containing the amount of population of the U.S. organized by state.

The reasons why we are interested in administrative units are mainly two: first, dataset organized this way are very common; second, integrating data collected by different administrations involves dealing with heterogeneity. It often happens that those data do not have an explicit geographical feature, and that they are simply stored by name of the administratie units. An example is shown in Picture 4.6. We can deal with the problem by mapping the name of the administratie units to a given table containing pairs (administratie units, shape), providing the dataset with a geographical feature.

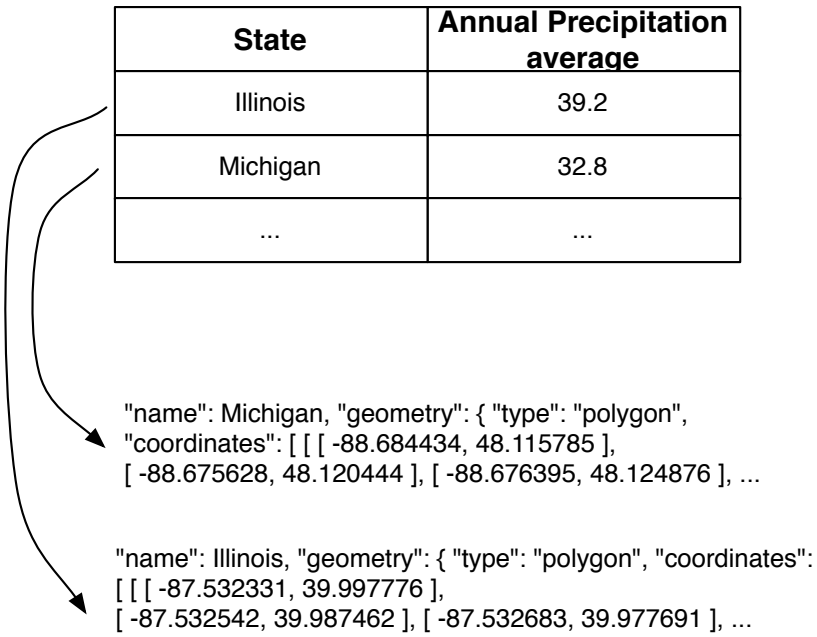


Figure 4.6: How to manage administrative units.

4.3 Distribution Similarity

We need now to describe a way to compute a similarity measure between geo-referenced datasets. What we ideally want to achieve is to compare geospatial datasets in the same way we compare words. We want to create a metric that is for geospatial datasets what *Levenshtein Distance* is for strings of characters. We need a very general and computationally inexpensive approach: we are not going to describe a model to predict the future behavior of the analyzed phenomena. It would not be really useful for our purpose. We just need a simple measure to say whether the given datasets are similar or not.

Since we cannot rely on the topology of the geospatial feature, suppose you are analyzing datasets of points, in order to identify similarities we decided to focus on the distribution over the space of the instances. Recalling the problem of time and space heterogeneity, we start working on the simple assumption

that the two datasets are about the same time interval and the same area. The basic idea of the approach lies in the simple observation that: if two datasets have a common pattern in their distributions of values, they likely describe the same concept; if their distributions are very dissimilar, they unlikely describe different concepts.

The most common techniques in geostatistics focus on finding trends within a single dataset. A commonly used technique for analyzing trends in spatial datasets is spatial autocorrelation, that is a statistic to measure and analyze the degree of dependency among observations in a geographic space. However, spatial autocorrelation does not fit to our case, since it does not help in finding similarities across different datasets[37,38]. To conclude, we decided to build our own spatial similarity metric relying on a very basic statistical relationship: correlation.

4.3.1 Tessellation

In order to compare different datasets, we need to first reduce them to a common representation. We proceed partitioning the space where both datasets lie in d non-overlapping regions, obtaining a tessellation.

For sake of completeness, here is a formal definition of a tessellation:

Tessellation 1. *Let S be a closed subset of $\mathbb{R}^d$, $\Im = \{s_1, s_2, ..., s_n\}$ where s_i is a closed subset of S, and $s_i^{'}$ the interior of s_i. If the elements of $\Im$ satisfy:*

1. $s_i^{'} \cap s_j^{'} = \emptyset$ *for* $i \neq j$
2. $\bigcup_{i=1}^{n} s_i = S$

then the set $\Im$ is called a tessellation of S. Property (1) means that the interiors of the elements of $\Im$ are disjoint and (2) means that collectively the elements of $\Im$ fill the space S[5].

We briefly discuss the possible types of tessellations we can use. For instance, a type of tessellation that is widely used in artificial intelligence are Voronoi tessellations. Given a set of points, also called seeds, for each seed we identify a corresponding region consisting of all points closer to that seed than to any other[5]. In our case the seed will be the single observation. The problem with Voronoi diagrams is that they are built on a specific dataset. In our case we need to compare multiple dataset, and thus this technique does not suit to our situation. We move on and look at regular tessellation. regular tessellation are uniform and isohedral tessellations (i.e. those consisting of regular triangles, squares, or hexagons)[5]. A uniform tessellation is a tessellation for which the vertices of the tessellation are of the same type[5]. An isohedral tessellation is a monohedral tessellation (in which all the cells are of the same size and shape) in which the ordered sequence of the number of edges meeting at the ith corner of a cell is the same for every cell[5]. In short, the cells are completely interchangeable. Notice that the three regular tessellations induce different distributional characteristics of Moran's I. It has been proved that tessellations of triangles and squares are more appropriate if our main objective is the investigation of properties of spatial aggregation and rasters, whereas tessellations of hexagons are more pertinent if we wish to generalize our results to empirical tessellations[6]. To conclude, it is more appropriate to use hexagons than squares to investigate the distributional properties of spatial test statistics in empirical maps.

We define "tessellation dimension" as the number of cells that compose the tessellation. For instance considering the tessellations in Figure 4.7 we have $TessellationDimension = 36$.

Despite the fact we have shown that using tessellation of hexagons is a better solution for our situation, for sake of simplicity, we proceed using a

squared tessellation. Squared tessellations are easier to implement and they can also be used to deal with border line problems with longitude values in a simpler way (for instance, dealing with datasets are across -150 and +150 degrees of longitude). In short, a squared tessellation allows us to explore the effectiveness of our approach in an easier way. However, a real world implementation would better use an hexagon tessellation.

Once we tessellated the space of the datasets, for each of the datasets we create a new data structure containing a value for each cell in the tessellation (a float matrix). The value of each cell can be obtained in different ways; for instance, simply by averaging the observations contained in the cell or computing their sum. The function used to obtain the value of each cell depends on the analyzed concept. For instance, analyzing rainfall and precipitation we would use an "average function", while analyzing car crashes we would use a "sum function".

Once we have a uniform data structure containing our georeferenced datasets we can proceed comparing their distributions.

4.3.2 Correlation as Similarity Measure

The most common measure of relation between two quantities is the *Pearson's R*. There are several benefits in using this metric. The first is that this metric can be used when quantities (i.e. scores) varies, since the accuracy of this score increases when data is not normalized[59]. Another benefit is that the Pearson's R is the same for any scaling within an attribute. Thus, objects that describe the same data but use different units of measure can still be used[59]. This allows us to address the problem of the heterogeneity in units of measures. *Pearson's R* is obtained by dividing the covariance of the two variables by the product of their standard deviations, that can be simply computed as

$$\rho_{X,Y} = corr(X,Y) = \frac{\sigma_{X,Y}}{\sigma_X \cdot \sigma_Y}$$

In practice, for each cell for each dataset we compute its deviation from the mean value. For each cell we multiply the result obtained with one dataset times the result obtained with the other dataset and we sum the resulting values. Finally, we divide by the multiplication of the standard deviations of the two tessellated datasets. In formula:

$$r = \frac{\sum_{i=1}^{n}(X-\bar{X})(Y-\bar{Y})}{\sqrt{\sum_{i=1}^{n}(X-\bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y-\bar{Y})^2}}$$

Person's R returns a value such that $-1 \leq \rho_{X,Y} \leq 1$. A negative correlation is meaningful for geospatial analysis, and it could be used to infer a relationship between the datasets. However, for the moment we are only interested in similarity, and thus we simply consider negatively correlated datasets as very dissimilar one to the other. The obtained value can be integrated with other measures of similarity between the ontologies, such as syntactical similarity and so on. It is possible that two completely unrelated concepts are highly positively correlated: in this case, this approach is going to produce false positive links. Nevertheless we can integrate this method with other similarity measure to address this issue. To conclude, Figure 4.7 shows a complete overview of the process of finding a similarity measure between georeferenced data.

4.3.3 Similarity Measure Properties

In this section we briefly discuss whether it is legitimate or not to use Person's correlation as a similarity measure, considering the properties of similarity measures. Recalling the fact that a similarity measure is usually defined as inverse measure of a distance metric, it should have the following properties[61]:

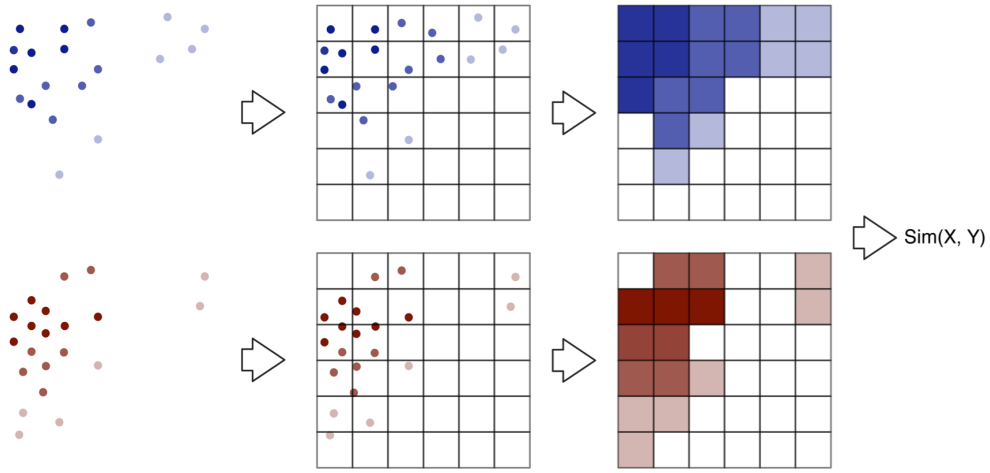


Figure 4.7: Computing a similarity measure for georeferenced datasets.

Similarity Measure 1. *Similarity Measure properties:*

1. $\exists s_0 \in \mathbb{R} : -\infty < sim(x,y) \leq s_0 < +\infty, \forall x,y \in \mathbb{R}$
2. $sim(x,x) = s_0$ *(sim is reflexive)*
3. $sim(x,y) = sim(y,x)$ *(sim is symmetric)*
4. $sim(x,y)sim(y,z) \leq [sim(x,y) + sim(y,z)]sim(x,z)$ *(triangle inequality holds)*

Pearson's R takes values on the interval $[-1,\ 1]$ can easily be transformed to one taking values on the interval $[0,\ 1]$, and vice versa. For this purpose we can use the transformation[3,40]:

$$[-1,\ 1] \rightarrow [0,\ 1] : t \rightarrow (t+1)/2$$

Again, the choice of this transformation depends on the fact that we are searching for similar concepts. A strongly negative correlation would be very interesting when searching for related concepts. Thus, property 1 is respected by our metric. Also properties 2 and 3 are trivially respected by Person's R. The

only property that we cannot respect using correlation is the 4th. Triangle inequality does not hold for Pearson's R, this is the reason why Pearson's R is often referred to as a semi-metric. Although similarity measures usually respect the triangular inequality, we can relax this constraint and proceed being aware of this issue.

4.3.4 Approach Issues

There are three main issues concerning the described approach:

1. Correlation is weak.

2. This method is useful only to find similarities between ontologies in the same area in the same time interval.

3. The measure of the correlation strongly depends on the dimension of the tessellation (see Section 5.3.2).

"Correlation does not imply causation" is a commonly used phrase in science and statistics to emphasize that the correlation is a weak concept. In our case we are not searching for a strong relationships such as causation, thus it seems to be fine for our approach. That said, we anyway need to take into account the fact that correlation is a weak relationship.

The second problem is the strict working hypothesis we are considering: the fact that we are comparing dataset with temporal and spatial overlap. Of course, there is no reason why the distribution of the rainfall on a sunny day should be similar to the distribution of the rainfall on a rainy day; this method fails when considering different time lapses.

The third problem is that the result strongly depends on the process used to compute it. The lower the granularity of the tessellation the higher the

correlation value, the higher the granularity the better the approximation of the distribution (see Section 5.3.2 for more details). In Section 4.3.5 we are going to address this problem.

4.3.5 Tessellation Dimension

In this section we deal with the problem of finding a correct resolution for a given dataset. In particular, we analyze the data distribution of each of the datasets in order to find a proper dimension for the tessellation. Our goal is to identify properties of the dataset that we would like to preserve, and situations that we would like to avoid. In this way we reduce the set of possible choices of tessellation dimension to a subset for which the obtained Pearson's R score is meaningful.

Oversampling

We define "oversampling" the process of sampling a dataset using a tessellation that is too fine for the given dataset. An example of oversampling is shown in Figure 4.8. The picture shows the application of two different tessellations to the same input dataset. "Tessellation 1" clearly fails in representing the given dataset, while "tessellation 2" gives a qualitatively good representation of the data. The problem with "tessellation 1" is that we are trying to represent the data with a tessellation that is finer than the resolution of the dataset. In order to obtain a good representation of the dataset we would like to identify the finer tessellation possible that does not involve oversampling.

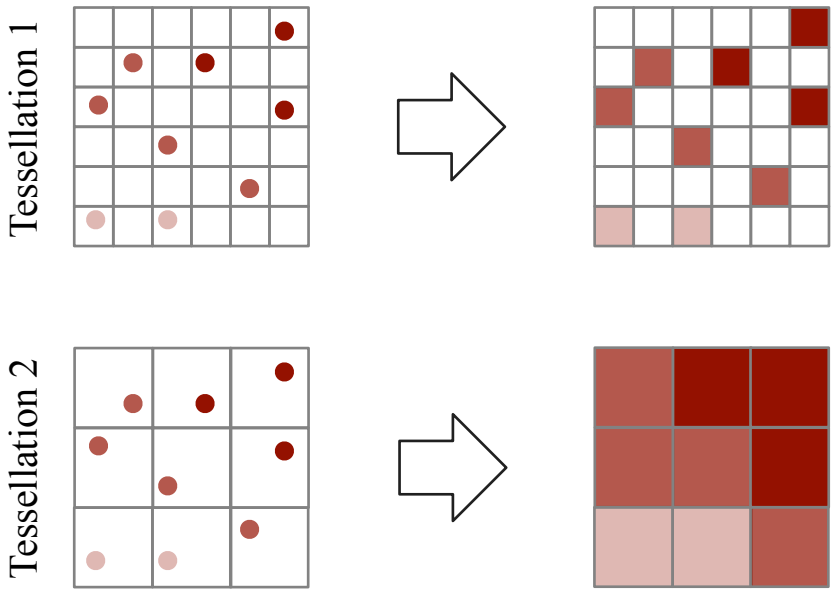


Figure 4.8: Example of oversampling.

Spatial Autocorrelation

Spatial autocorrelation is a widely used measure to analyze the degree of dependency among observations in a geographic space. Three commonly used spatial autocorrelation statistics are: *Moran's I*, *Geary's C* and *Getis's G*.

In order to compute these statistics, we need first to compute a spatial weights matrix, that represents the the geographic relationship between observations in a neighborhood. The influence between geographic observations is usually measured considering a distance measure. In our case, for instance, considering a contiguity matrix, we suppose that only adjacent cells influence the the given cell.

Spatial autocorrelation is represented by a value in the interval $[-1, 1]$, in which a value close to 0 indicates that the distribution of the datasets is close to random. Spatial autocorrelation that is significantly more positive than expected from random is an index of high degree of clustering across the

space, while significant negative spatial autocorrelation indicates that neighboring values are more dissimilar than expected by chance, and suggests a "chessboard-like" distribution. An example is shown in Figure 4.9. In partic-

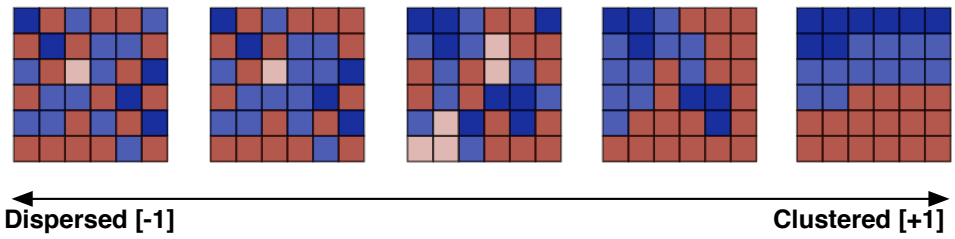


Figure 4.9: Spatial autocorrelation (`www.arcgis.com/features`).

ular we use Moran's I that is a widely used measure of spatial autocorrelation developed by Patrick Alfred Pierce Moran[44]. Moran's I is defined as:

$$I = \frac{N}{\sum_i \sum_j w_{ij}} \frac{\sum_i \sum_j w_{ij}(X_i - \bar{X})(X_j - \bar{X})}{\sum_i (X_i - \bar{X})^2}$$

where N is the number of spatial units indexed by i and j; X is the variable of interest; $\bar{X}$ is the mean of X; and w_{ij} is an element of a matrix of spatial weights[48]. The matrix of spatial weights represents the influence that each element has on any another element. Following standard convention, here we exclude "self influence" by assuming that 0 $w_{ii} = 0$ for all $i = 1, ...n$ (so that W has a zero diagonal). It can be computed in different ways: we decided to use, due to its simplicity and convenience from a computational point of view, a k-Nearest Neighbors algorithm (Appendix A provides details about how to compute a k-Nearest Neighbors Weights matrix).

Best Tessellation Dimension Identification

Oversampling is strictly related to the degree of clustering of the dataset. The tessellations for which we obtain a high degree of clustering involve a good

representation of the dataset (for example "tessellation 2"), the tessellations for which we obtain a low degree of clustering involve a bad representation of the dataset (for example "tessllation 1") A high score indicates a high degree of clustering among the instances, that is in interesting information for who analyzes the data, and it is a property that we would like to preserve. In statistics the degree of clustering is measured with spatial autocorrelation. In particular, we use a measure of spatial autocorrelation called Moran's I, that is described in the previous Subsection (4.3.5).

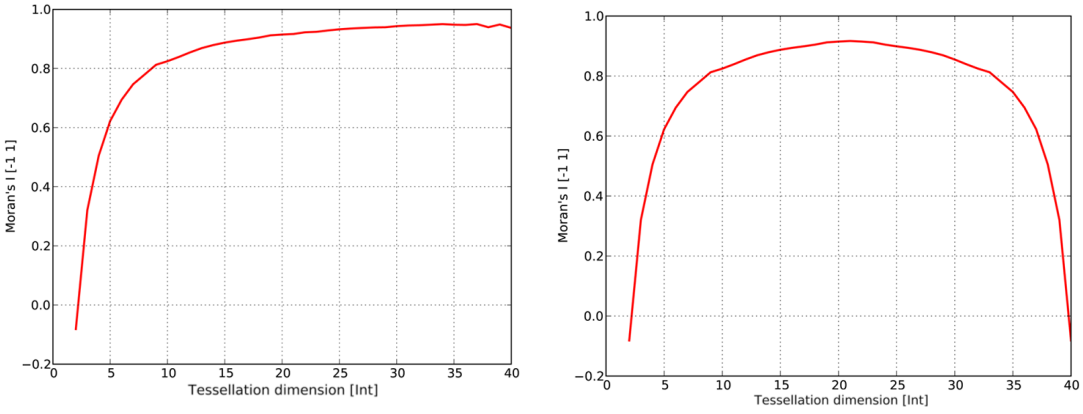
In order to identify the tessellation dimension that better represents the data we select the tessellation that maximizes the Moran's I score of the dataset. In Chapter 5.3.2 we will discuss experiments to further justify this idea.

The process of identifying the best tessellation dimension is very similar to the one used in incremental spatial autocorrelation in the ArcGIS spatial analysis tool[57]. The incremental spatial autocorrelation allows to identify the best distance band, that is the distance within a spatial entity is considered neighbor of another spatial entity. The difference with our approach is that instead of maximizing on distance band we maximize on the tessellation dimension. The distance band is related to the way the spatial autocorrelation is computed, while the tessellation dimension is related to the topology of the dataset.

Best trade-off Point

Since we are studying a trend in a discrete and finite interval, one could just use a simple maximum function to select the maximum value of the curve. However there are situations when this is not our best choice.

For instance, considering the examples shown in Figure 4.10, taking the maximum value would perfectly work for the curve in Figure 4.10b, but it would involve some issues in the case of Figure 4.10a.



(a) *(b)*

Figure 4.10: Curve trends for tessellation dimension.

Considering a monotonically increasing curve, selecting the maximum we are also taking the maximum tessellation side dimension possible: the result depends on our choice of the maximum tessellation side dimension and having a high tessellation dimension is computationally more expensive than having a low tessellation dimension.

A better way to choose the tessellation dimension would be to select a dimension such that the derivative of the curve is "sufficiently" close to zero, we do not need it to be exactly zero. For instance, in the case in Figure 4.11 we could use the elbow method[28] to identify the point where the derivative of the curve significantly change. In this way we are able to identify a tessellation that qualitatively represents well the data. A possible way to find the best trade-off point on the curve works as follow: for each point on the curve, we find the one with the maximum distance from the line linking the first and the last point. Figure 4.11 graphically shows how the distance is computed. We first project the vector p on b obtaining $\hat{p}$, and then we take as distance the norm of difference of p and $\hat{p}$[50]. This technique effectively works both in the case of a monotonic curve and in the case of a curve that is not monotonic.

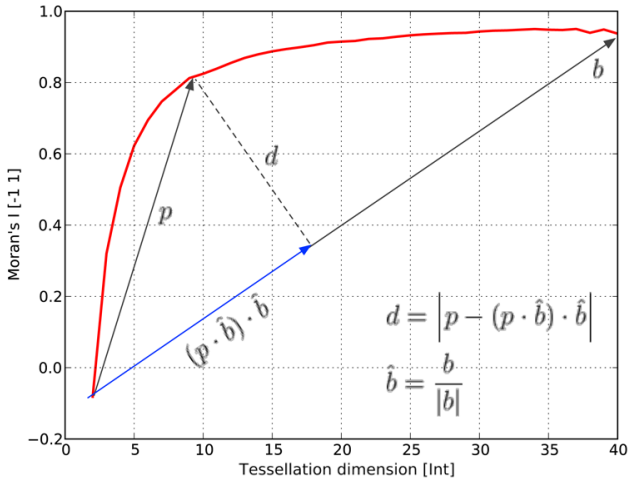


Figure 4.11: Finding the best trade-off point on the curve[50].

Once we identified the best tessellation dimension for each of the two dataset, we compare the datasets using the lower one. In this way we avoid oversampling in both the datasets, and we select the tessellation dimension that is less computationally expensive.

4.3.6 Spatial Autocorrelation as Similarity Measure

We already discussed the issues involved by using correlation as similarity measure, and we already said that the method works only when considering data about the same concept at the same time in the same place.

We discuss now another possible solution to deal with spatial and temporal heterogeneities. Basically, our hypothesis is the following: "Considering a sufficiently wide area and a sufficiently wide time interval, a particular concept has a same spatial autocorrelation score, i.e. degree of clustering, independently of where and when it was measured".

Proving that the previous statement is true is a very hard challenge. In

Chapter 5 we provide some examples to support this idea. Further considerations are left as future developments.

Chapter 5

Experiments

5.1 Overview

This chapter contains the results obtained applying my approach and a description of the tools we produced to show its effectiveness. Next, in this chapter, Section 5.2 describes the results obtained using Moran's I to identify the best tessellation dimension. Section 5.3 contains the experiments performed using the whole approach for comparing georeferenced datasets. Section 5.4 describes GIVA and the integration of my approach for comparing georeferenced datasets into AgreementMaker[14]. Finally, Section 5.5 shows the results obtained using spatial autocorrelation as similarity measure.

5.2 Tessellation Dimension

A major issue about the approach we described in Chapter 4 is the choice of the tessellation dimension. In this section we perform experiments to study the trend of the Moran's I score when the tessellation dimension varies. As we have seen in Chapter 4 this problem is strictly related with the problem of

oversampling. We tested our approach on real datasets about precipitation in Illinois from the National Weather Service.

5.2.1 Implementation

The experiments have been performed using a python script, whose structure is described in the UML class diagram shown in Figure 5.1. The same organization is then translated in JavaScript and used for the visualization tool of GIVA. The dataset is taken from a PostGIS database using the module `PostGISConnection`, relying on the `psycopg2` library[1]. A `GeoferencedDataset` object is created for the loaded dataset. Starting from the `GeoreferencedDataset` object, by using the method `tessellate(int tessellationDimension, TessellationMethod method)`, we create a new `TessellatedDataset` object of a given dimension. The squared tessellation is represented in the class by a `float` matrix and by its bounds.

The method `computeMoransI(int k)` takes the k value as input, used to compute the spatial weights using k-Nearest Neighbor, and returns the Moran's I score. We tested different k in order to see how the trend of the Moran's I changes with respect to it. The results are plotted using the `matplotlib` library[2].

5.2.2 Results

The results obtained analyzing the relationship between Moran's I and oversampling are shown in Figure 5.2.

In order to clarify the obtained results we visualize on a map the tessellations obtained with different tessellation dimensions in Figure 5.3. The tessel-

[1] `https://pypi.python.org/pypi/psycopg2`
[2] `http://matplotlib.org/`

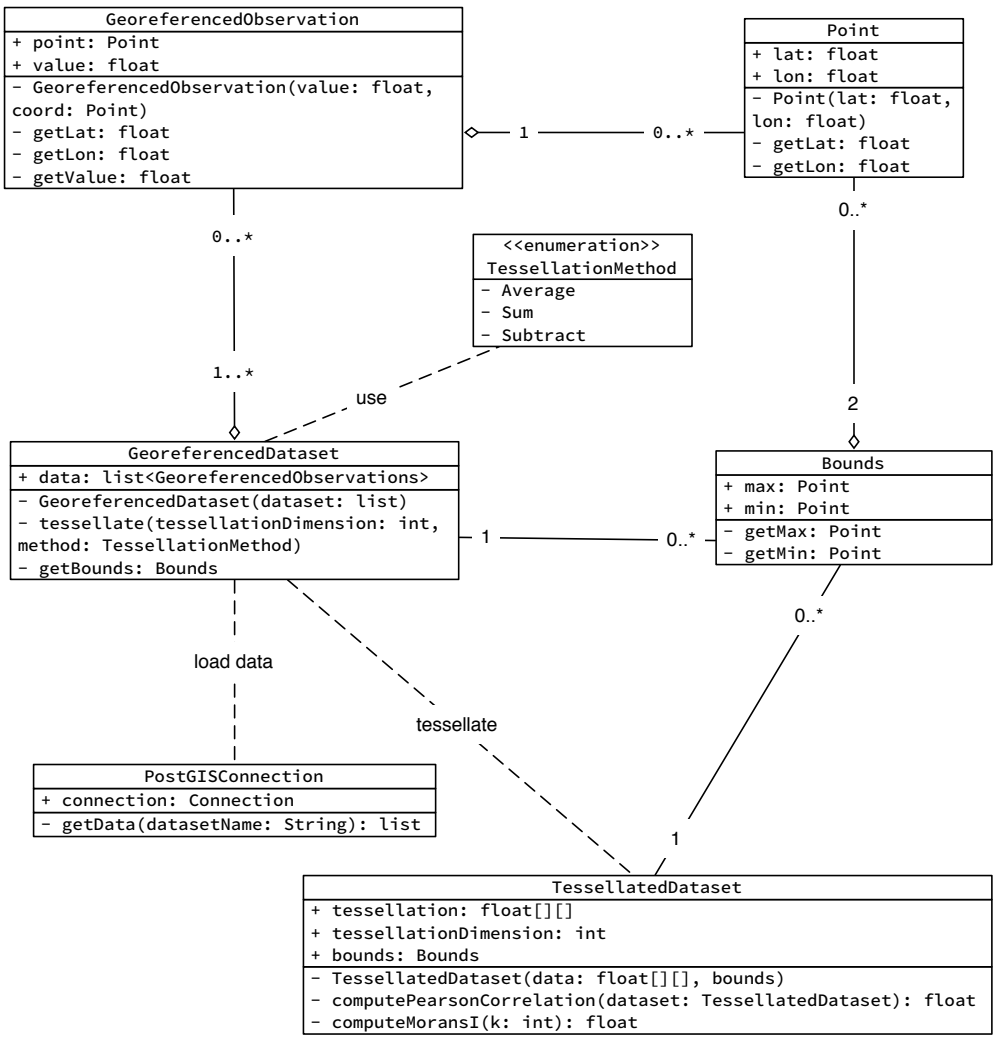


Figure 5.1: Data structures class diagram.

lation on the left corresponds to the tessellation dimension identified by our approach (225), the tessellation on the right corresponds to a higher tessellation dimension (900). Three main observations arise from this visualization. First, our hypotheses is confirmed: it is true that using the tessellation dimension that maximizes the Moran's I we have a qualitatively good representation of the dataset. Second, a too dense tessellation involves a low Moran's I value and results in a qualitatively bad representation of the dataset. Third, the max-

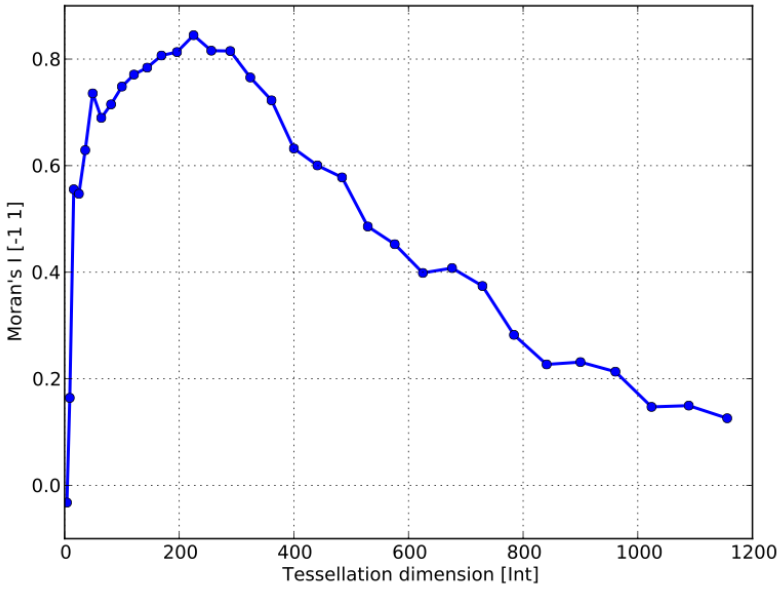


Figure 5.2: Moran's I trend for rainfall dataset

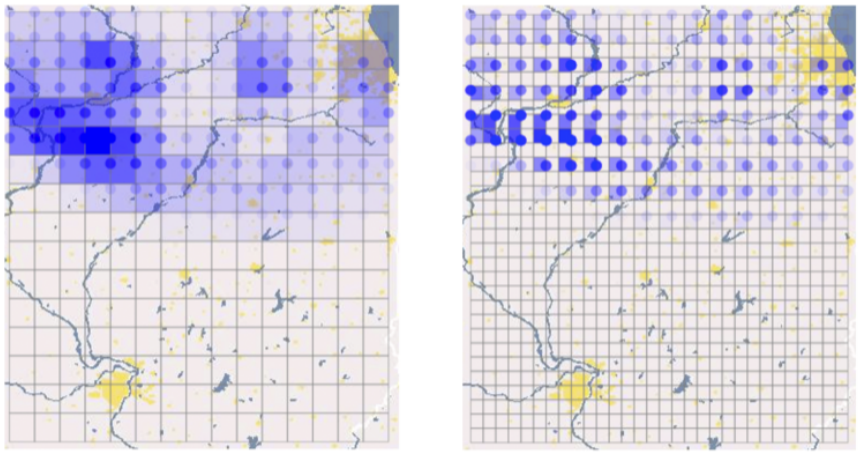


Figure 5.3: Real world example of oversampling

imum value of the Moran's I converges to the situation in which we have only one observation per cell.

5.3 Correlation and Tessellation Dimension

This section contains the results we obtained using correlation as similarity measure for georeferenced datasets. The experiments focus on showing the effectiveness of the use of the correlation together with the identification of the best tessellation dimension, but not yet of the overall effectiveness of the geospatial ontology matching system. We test our approach with both synthetic and real datasets.

5.3.1 Implementation

The tests have been performed using the same data structures used for testing the tessellation choice, and are shown in Figure 5.1. The method `computePersons Correlation(TessellatedDataset secondDataset)` of the `TessellatedDataset` Class returns the correlation value. For the evaluation on the synthetic data we avoided passing through `GeoreferencedDataset`, thus we just created ad-hoc `TessellatedDataset` objects.

5.3.2 Synthetic Datasets

Figure 5.4 shows a simple graphic of the results obtained computing correlation on synthetic datasets. The synthetic datasets are created in such a way to represent meaningful use cases, from which we expect a particular behavior a-priori. The tessellation used in the experiment is a squared tessellation of dimension $5x5$. The results are computed on dataset distributed uniformly over the space and with datasets distributed in a smoother way. This results are meant to validate both the implementation and the soundness of decision comparing tessellated georeferenced datasets using correlation. Next section contains more extensive experiments on real data.

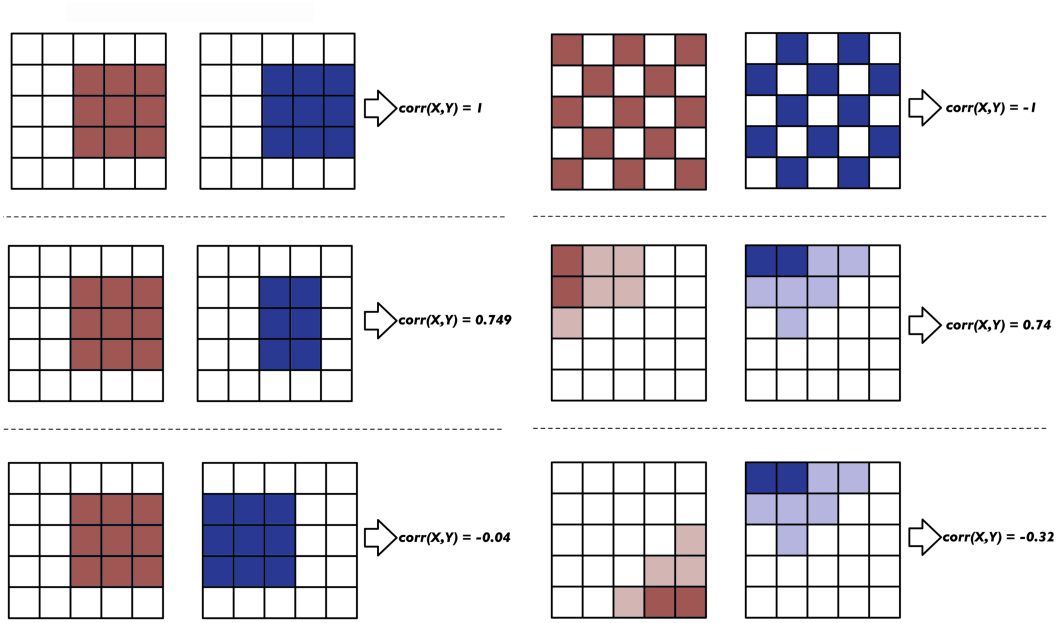
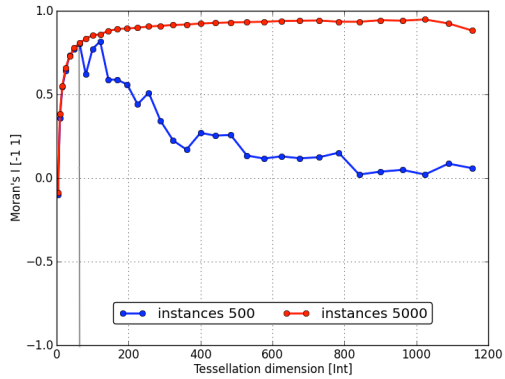


Figure 5.4: Synthetic examples of correlation as spatial similarity measure.

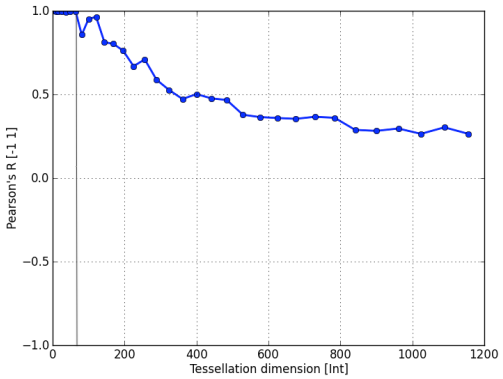
5.3.3 Real World Datasets

In this section we show the results we obtained applying our approach to datasets from National Weather Service (NWS). The idea is to show that the tessellation dimension identified maximizing the Moran's I allows to obtain a good value of Pearson's R. Two different kinds of test have been performed.

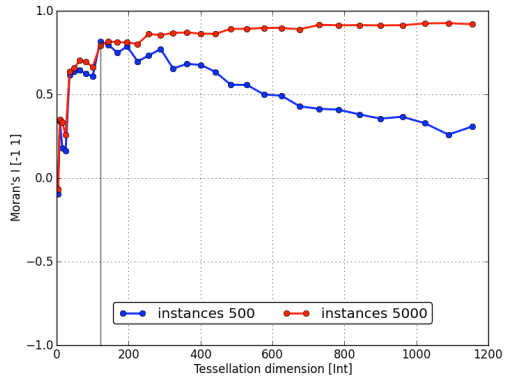
The first test consists in computing Moran's I and Pearson's R for datasets that are expected to be highly correlated. In this way it is shown that we are able to obtain a tessellation dimension for which the two datasets are correctly highly correlated. From NWS we downloaded datasets about precipitation in Illinois, Michigan and Indiana during February 2013. For each state we created two datasets randomly sampling 500 and 5000 instances from the same source. The two obtained datasets are compared using the approach described in Section 4.3.2 of Chapter 4, Figure 5.5 shows the results. The datasets come from the very same dataset, and we expect them to be highly correlated.



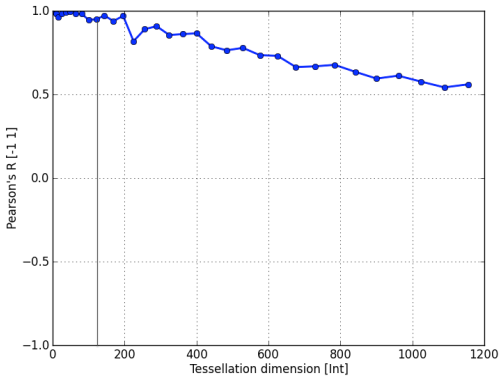
(a) Moran's I, Illinois



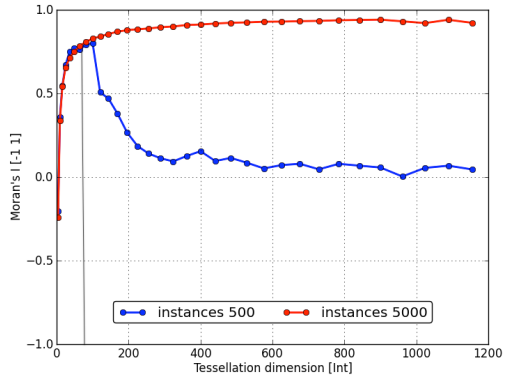
(b) Pearson's R, Illinois



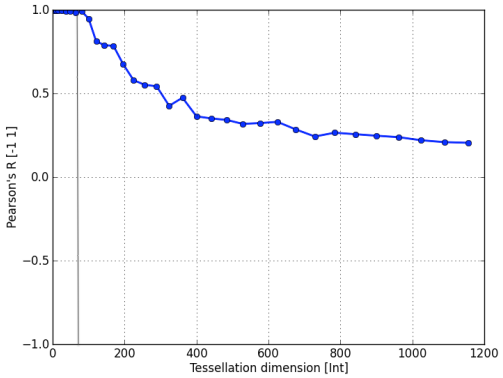
(c) Moran's I, Michigan



(d) Pearson's R, Michigan



(e) Moran's I, Indiana



(f) Pearson's R, Indiana

Figure 5.5: Highly correlated datasets about precipitation.

The results we obtained are:

1. Pearson's R score strongly depends on the tessellation dimension.

2. Using my approach to find the best tessellation dimension, by considering Moran's I curve, the correlation between the two datasets has a very high score.

The second test consists in computing Moran's I and Pearson's R for datasets that are expected to have a low correlation. In this way we show that my approach allows to identify a tessellation dimension for which the two datasets correctly have a low correlation score. From NWS we downloaded datasets about precipitation in Illinois, Michigan and Indiana during February 2013 and March 2013. In this way, we obtained two datasets for each state. The two obtained datasets are compared using the approach described in Section 4.3.2 of Chapter 4, Figure 5.6 shows the results. In this case, there is no reason why precipitation in March should be highly correlated to precipitation in February, thus, we suppose the two dataset to be lowly correlated.

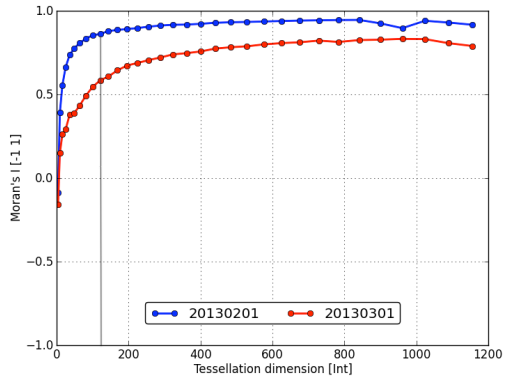
Results:

1. Pearson's R score strongly depends on the tessellation dimension.

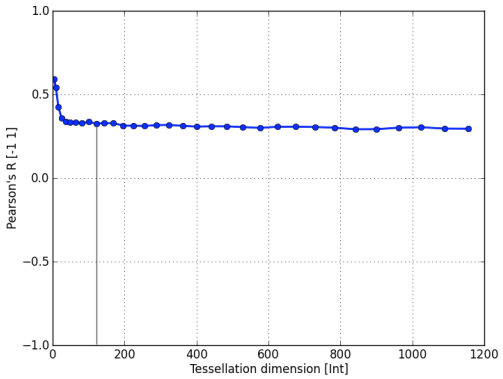
2. Using my approach to find the best tessellation dimension, by considering Moran's I curve, the correlation between the two datasets has a very low score.

5.3.4 Census Dataset and MAUP Problem

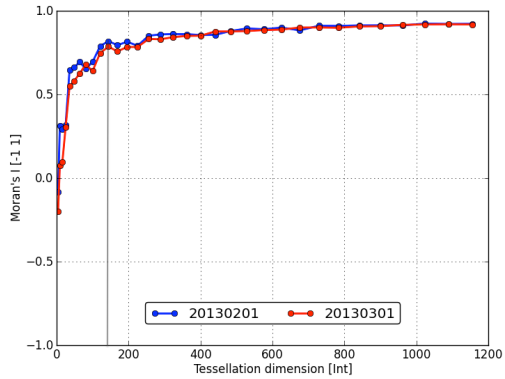
My approach can be used also to deal with the MAUP problem. In order to show that, in this section we compare datasets at county level with datasets at state level. The datasets are extracted from the census dataset, from which we



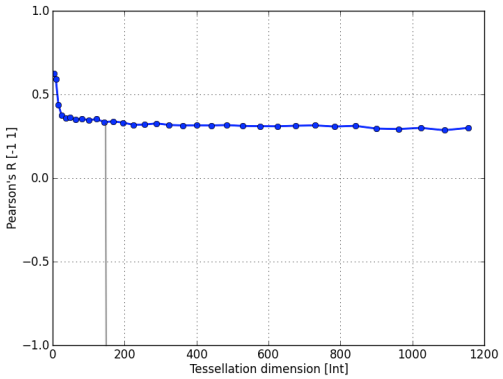
(a) Morans's I, Illinois



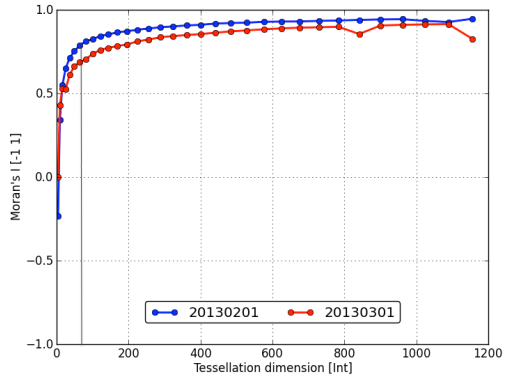
(b) Pearson's R, Illinois



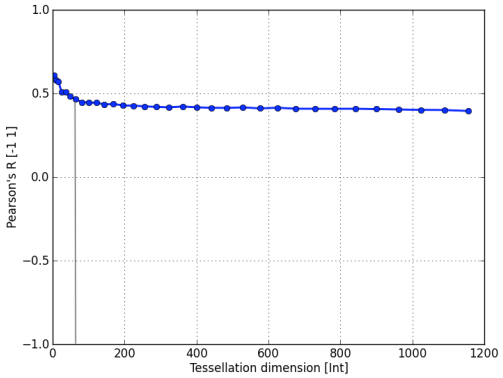
(c) Morans's I, Michigan



(d) Pearson's R, Michigan



(e) Morans's I, Indiana



(f) Pearson's R, Indiana

Figure 5.6: Lowly correlated datasets about precipitation.

considered three attributes: family median income, household mean income and workers median earnings. The dataset at county level is simply obtained using the FIPS code of the counties, while the dataset at state level are obtained aggregating the values extracted from the counties.

The obtained results are shown in Figure 5.7.

As we have already seen for the precipitation datasets from National Weather Service (NWS), also in the case of the census dataset we can easily identify tessellation for which we have a high correlation. In this case, however, the curve representing the Pearson's R with respect to the tessellation dimension is quiet disturbed for low tessellation dimensions.

5.4 Tools

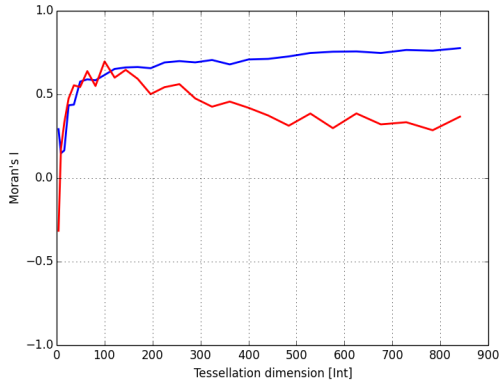
In this section we discuss the implementation of the techniques described in my thesis in AgreementMaker[14] and GIVA[12].

5.4.1 GIVA

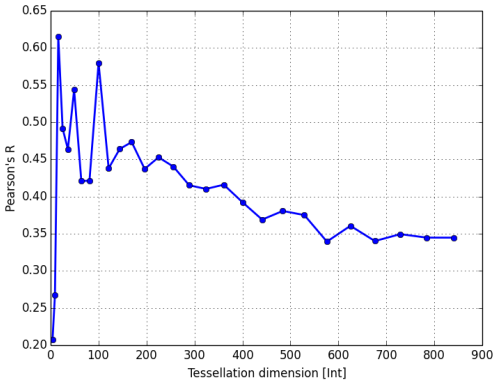
GIVA is a "semantic framework that assists domain experts in integrating highly heterogeneous datasets and in analyzing and visualizing dependencies among them"[12]. GIVA supports different types of users: administrator, domain expert, and casual user, with different types of access[12].

The administrator has the overall control over the platform, he is in charge of inserting new datasets in the system. A domain expert is an analyst who uses our framework for integrating or analyzing data. Finally, the casual user is someone who uses needs to use our framework but who does not have the same skills of the domain expert.

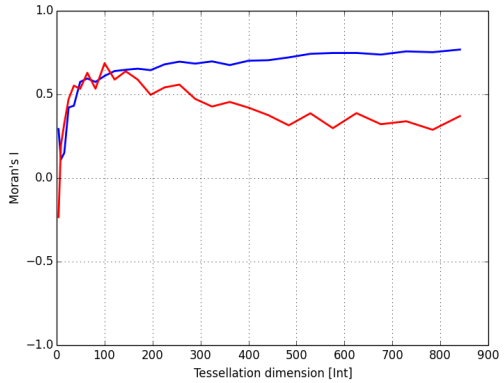
There are a lot of GIS systems out there. However, given the complexity



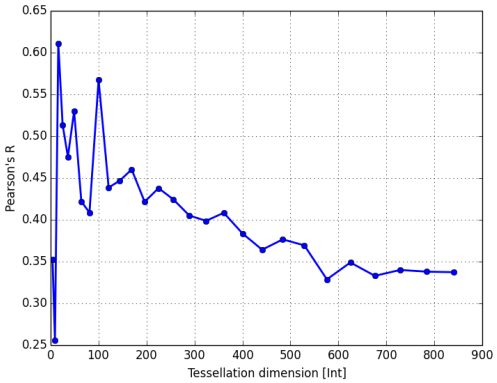
(a) Morans's I, family median income



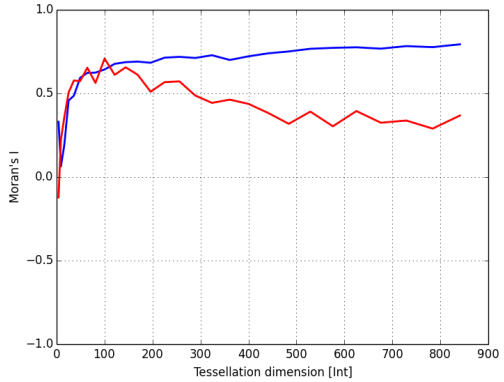
(b) Pearson's R, family median income



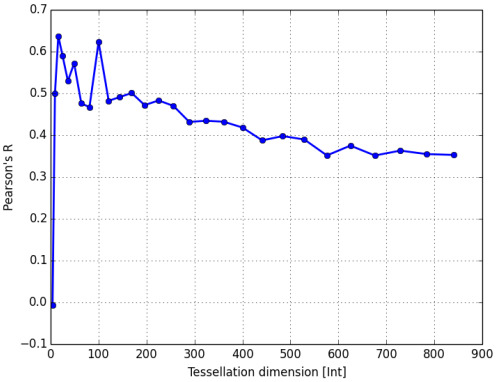
(c) Morans's I, mean income



(d) Pearson's R, mean income



(e) Morans's I, median earnings



(f) Pearson's R, median earnings

Figure 5.7: Experiments related to the MAUP problem.

and the technical competences required to build our framework, we were not able to find similar GIS system with similar capabilities[12]. The visualization tool has been developed as a Java Dynamic Web Project, in order to provide the tool as a service available on the web. A servlet on the server side queries a PostGIS database and passes data, in GeoJSON[3] format, to the client side.

GeoJSON data are efficiently imported in JavasScript, thanks to `eval()` function, and manipulated using the same classes described in Figure 5.1 (that we have reimplemented in JavaScript). The data are finally visualized using OpenLayers library[4] for showing the map, and D3[5] for plotting the Moran's I chart.

OpenLayers is an Open Source JavaScript library that makes it easy to put a dynamic map in any web page. It can display map tiles and markers loaded from any source. It has been particularly helpful since it allows to plot a given dataset just by receiving a GeoJSON, without the need of describing how to draw particular shapes.

D3 is a JavaScript library for manipulating documents based on data. D3 helps you representing data using HTML, SVG and CSS. The emphasis of D3 on web standards gives you the full capabilities of modern browsers without tying yourself to a proprietary framework, combining powerful visualization components and a data-driven approach to DOM manipulation. It has been chosen as main library for visualizing data because of its power and modernity.

Figure 5.8 shows an example of use of the geovisualization tool. In the example, in particular, two datasets are loaded from the database: the red dataset is about water gages, while the blue dataset is about rainfall. A specific

[3] `http://geojson.org/`
[4] `http://openlayers.org/`
[5] `http://d3js.org/`

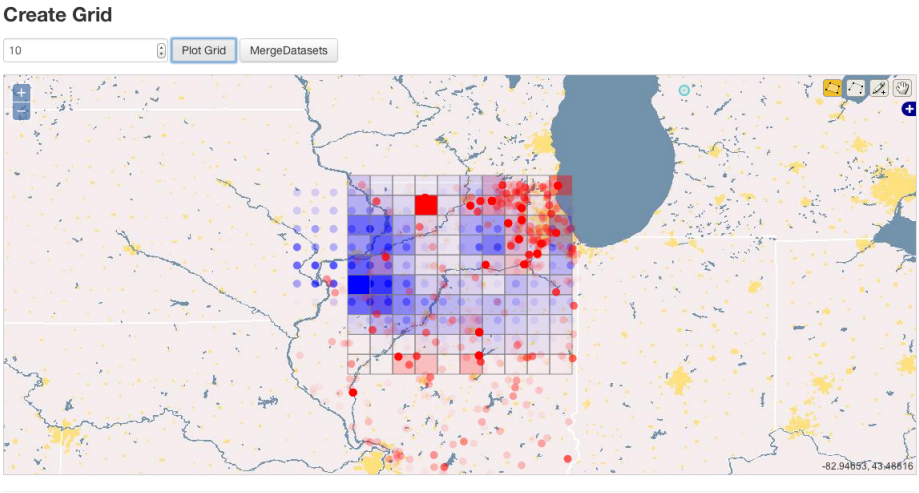


Figure 5.8: Screenshot of the GIVA's visualization tool[12].

area in the map is selected and tessellations of a selected dimension are created. The intensity of the points of the dataset and of the cells of the tessellation is defined in the following way: the higher the intensity the higher the opacity. Figure 5.9 shows the component of the framework that allows to visualize how

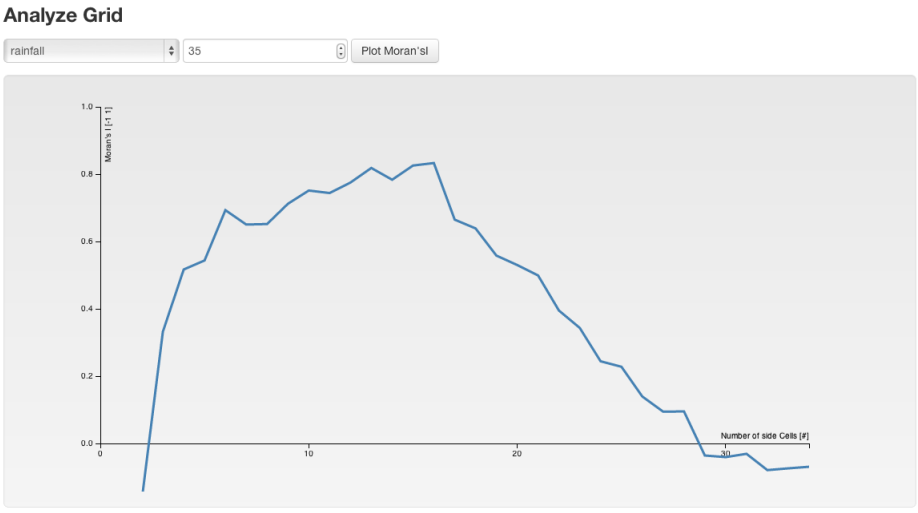


Figure 5.9: Screenshot of the GIVA's Moran's I chart[12].

does the Moran's I value varies when changing the tessellation dimension. The

Table 5.1: UNMODIFIED SYNTACTICAL MATCHER.

Source	Target	Link
Unit	Unit	1.0
Value	Val	0.819
Longitude	Long	0.608
Latitude	Lat	0.539
date_recorded	Precipitation	0.344
Rainfall	time	0.049

user selects the dataset he wants to analyze and sets the maximum value for which computing the Moran's I.

5.4.2 AgreementMaker

In order to provide an example of how the described techniques can be used to improve the performances of an ontology matching algorithm, we built a prototype within AgreementMaker[14] (AM) and we tested it using synthetic ontologies. The used ontologies are described in Appendix B. In order to obtain dataset of georeferenced observations highly correlated, we sampled two datasets from the National Weather Service. The idea is to simulate the situation in which we have two datasets about the same concept.

We modified a syntactical matcher as described in Section 4.1.2 of Chapter 4. This Matcher compares the source and target ontologies by using string matching techniques on the words that compose their attribute. The results obtained by the unmodified matcher are show in Table 5.1. In this case *precipitation* and *rainfall* are not even matched, but they are confused with *time* and *data_recorder* attributes. We proceed by adding the *distribution* node.

On the initialization of the matcher the tree defined in the ontology is modified by adding the distribution node at the same level of *longitude*, *latitude* and *value*. When the matcher compares the *distribution* nodes our matcher imposes

Table 5.2: SYNTACTICAL MATCHER, DISTRIBUTION ADDED.

Source	Target	Link
Unit	Unit	1.0
Distribution	Distribution	0.981
Value	Val	0.819
Longitude	Long	0.608
Latitude	Lat	0.539
date_recorded	Precipitation	0.344
Rainfall	time	0.049

as similarity measure the measure obtained by comparing the instances of the datasets as described in Section 4.3. The obtained result is shown in Table 5.2, and the differences with the previous table are highlighted in green in the table.

Finally, we try to use the *distribution* node to improve the matching between *precipitation* and $rainfall$. After the mapping between the *distribution* similarities is added to the Similarity Matrix, we impose as similarity measure between Precipitation and $rainfall$ the same value of the similarity between the *distribution* nodes. The obtained results are shown in Table 5.3. The updated value in the alignment in highlighted in yellow in the table. Notice that the imposed mapping between $rainfall$ and *precipitation* influences also other attributes: *data_recored* and *time* are now correctly mapped, even if with a low accuracy.

5.5 Spatial Autocorrelation as Similarity Measure

In Chapter 4 we discussed the fact we might use spatial autocorrelation as a feature to identify similar concepts. In this section, using the GIVA framework, we provide some evidence about what discussed. We take our dataset about precipitation from National Weather Service and we selected data in two dif-

Table 5.3: ENHANCED INSTANCE-BASED MATCHER

Source	Target	Link
Unit	Unit	1.0
Distribution	Distribution	0.922
Rainfall	Precipitation	0.922
Value	Val	0.819
Longitude	Long	0.608
Latitude	Lat	0.539
date_recorded	time	0.102

ferent areas: Illinois and Florida. We sampled one dataset about Florida, and two datasets about Illinois. The two datasets in the Illinois area are taken at different times.

In this way we are comparing datasets temporally and spatially disjoint. Computing the trend of the Moran's I score for the different datasets, we noticed that the value of Moran's I, for a sufficiently high tessellation dimension, was close to 0.9. Figure 5.10 shows the trend obtained. The yellow curve is what we obtain plotting water gages, that is used to highlight the difference with respect to the datasets about precipitation. Two of the curves start decreasing after a while, but their maximum value is similar. Those experiments are not enough to definitely state that spatial autocorrelation is good as similarity measure. However, they enforces our belief that it is a good feature for comparing geospatial data, and that it worth to be tested in such a way.

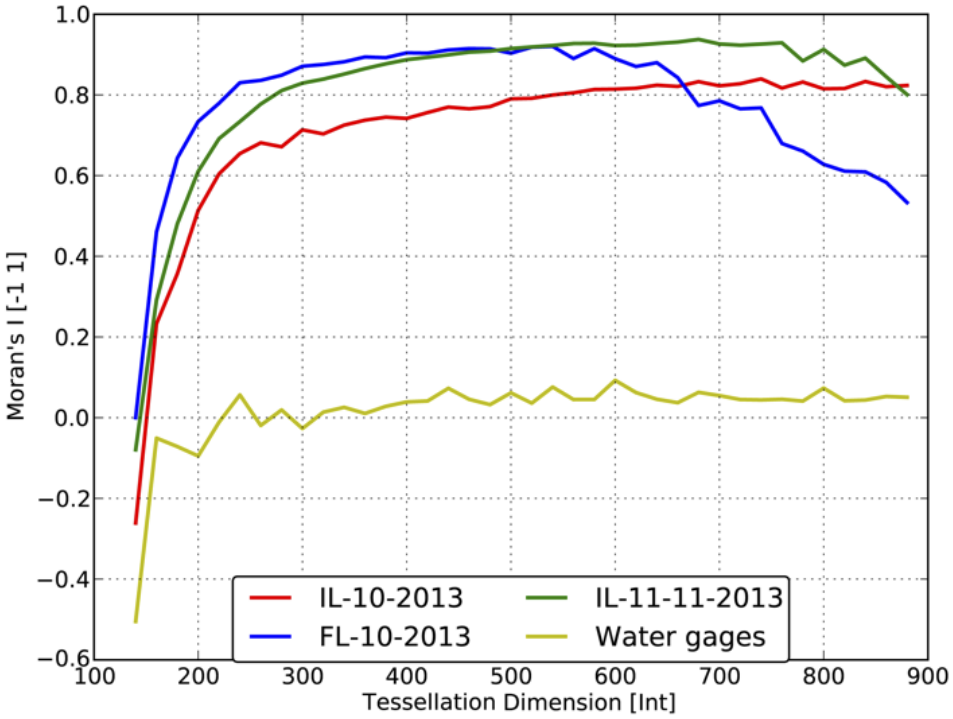


Figure 5.10: Spatial autocorrelation as similarity measure.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

Geospatial data are becoming increasingly important in today's world. They incredibly improve both the our ability in making decisions and the potential of our tools. A fundamental step in the evolution of the management of geospatial data lies in the capability of integrating them, dealing with their heterogeneities and different representations.

In this work, we proposed an instance-based approach to align geospatial ontologies. In particular, my work focuses on enhancing ontology matching algorithms in order to make them more effective when identifying similarities between ontologies of georeferenced observations. In order to overcome the limitation of the standard techniques, we decided to go for an instance-based approach.

The experiments performed on real world datasets show that our method allows to properly compute Pearson's R on datasets with different resolutions.

Table 6.1 shows the heterogeneities we were able to address in this work. We have shown a small prototype of matcher implemented using the described

Table 6.1: ADDRESSED HETEROGENEITIES

Heterogeneity	**Addressed using**
Unit of Measure	Person's R
Resolution	Maximization of Moran's I
Format	GDAL and Data Translation

technologies. We also suggested a way to compare datasets spatially and temporally disjoint using spatial autocorrelation.

6.2 Range

The techniques discussed are very general, and can be also used for other applications: we developed a similarity measure for georeferenced dataset that is not only suitable for ontology matching. Different machine learning techniques relies on similarity measure. For example, we could use this measure to create clusters or similar datasets using a k-means algorithm[41].

Limiting the scope of the concepts we are able to compare we would be able to use more powerful techniques. As future work, we would like to extend our approach. For instance, we would like to be able to compare datasets that are not collected in the same time interval and in the same area.

In order to do that we are currently doing experiments with supervised learning techniques. We first define an ontology of the concepts we want to compare, and than we classify the given datasets as member of the one of the concepts of the ontology. Our hope is that datasets about the same concepts present common features even tough they are not supposed to be distributed in the same way.

6.3 Future Works

This work can be improved along three possible paths: comparing datasets temporally and spatially disjoint, identifying strong relationships between concepts and better integrating the instance-based matching techniques into a structural matcher.

6.3.1 Comparing Datasets Disjoint in Space and Time

A possible feature that can be useful for identifying similarities across concepts in different places is to use spatial autocorrelation.

Our hypothesis is: "considering a sufficiently wide area and a sufficiently wide time interval, a particular concept has a same spatial autocorrelation score, i.e. degree of clustering, independently of where and when it was measured". However, much work needs to be done to prove the real effectiveness of this approach.

Issues lies in the fact that spatial autocorrelation is not a reliable measure for computing the similarity between different datasets, and that this approach might involve too many false positive matchings.

The problem is that its effectiveness depends on: the width of the selected area, the number of the instances, the type of phenomena and so on. Furthermore, probably many different phenomena are characterized by a similar spatial autocorrelation, and this involves our method to generate many false positive matchings even when the other requirements for comparing the datasets are respected.

Another possible way to deal with this problem, is by relying on metadata. When they come together with the dataset, metadata are an incredibly useful source of information. They can be used to identify new features to be used to

compare datasets in different locations in different time intervals.

6.3.2 Identification of Strong Relationships between Concepts

Similarly to what discussed in the last section, the features obtained by an instance-based approach, are probably too few to identify strong relationships between two datasets. Probably our instance-based approach is not enough for this purpose. Another problem is that it is very unlikely that an unsupervised approach is enough to find such a relationships. A possibility is to classify the

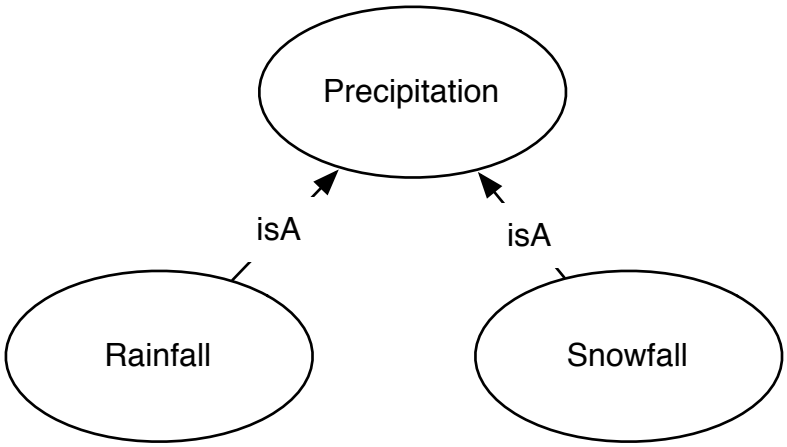


Figure 6.1: Precipitation ontology.

dataset as belonging to a given ontology, and then reason on the hierarchy of the ontology to infer the relationship. For instance, given two datasets (DS1 and DS2) and the ontology depicted in Figure 6.1, we could match each of the datasets to one of the elements in the ontology. If DS1 is matched with "Precipitation" and DS2 is matched with "Rainfall", we can infer the relation $DS2 \subset DS1$. Otherwise, is DS1 is matched with "Snowfall" and DS2 is matched with "Rainfall", we can create the dataset DS3 (about "Precipitation") thanks to the relationship $DS3 = DS1 \cup DS2$.

The problems lies in proving that the used relationships hold in the given situation. We are doing experiments in this direction using supervised machine learning techniques, and the results we are obtaining are encouraging.

6.3.3 Instance-based and Structural Matchers Integration

In Chapter 4 we discussed how we can integrate our instance-based similarity measure for geospatial data into an ontology matching system. We have seen that it is possible to integrate the instance-based component in two ways: as a post-processing step to ontology matching, or in a harder way, iterating after having improved the alignment.

It would be interesting to further explore the effects of the harder approach. Notice that throughout all the work, we relied on the hypotheses that latitude, longitude and the values was correctly identified a priori. It would be interesting to use ontology matching to identify these attributes, and include the whole process in the loop, to iteratively improve each identified link.

Bibliography

[1] Pragya Agarwal. Walter Christaller: Hierarchical patterns of urbanization. *Centre of Spatially Integrated Social Science*, 2007.

[2] Khaldoun Al Agha. Digital cities of the future: Democratic city space through a citizen centric model. URL `http://www.eitictlabs.eu/uploads/media/DigitalCities_web_0711_01.pdf`, visited 2013-01-20.

[3] Per Ahlgren, Bo Jarneving, and Ronald Rousseau. Requirements for a cocitation similarity measure, with special reference to pearson's correlation coefficient. *Journal of the American Society for Information Science and Technology*, 54(6):550–560, 2003.

[4] Robert Battle and Dave Kolas. Enabling the geospatial semantic web with parliament and geosparql. *Semantic Web*, 3(4):355–370, 2012.

[5] B Boots, A Okabe, and K Sugihara. Spatial tessellations. *Geographical information systems*, 1:503–526, 1999.

[6] Barry Boots and Michael Tiefelsdorf. Global and local spatial autocorrelation in bounded regular tessellations. *Journal of Geographical Systems*, 2(4):319–348, 2000.

[7] I Budak Arpinar, Amit Sheth, Cartic Ramakrishnan, E Lynn Usery, Molly Azami, and Mei-Po Kwan. Geospatial ontology development and semantic analytics. *Transactions in GIS*, 10(4):551–575, 2006.

[8] Kang-tsung Chang. *Introduction to geographic information systems*. McGraw-Hill New York, 2010.

[9] Andrew D Cliff and J Keith Ord. *The problem of spatial autocorrelation*. University, 1968.

[10] Andrew D Cliff and J Keith Ord. Spatial autocorrelation: a review of existing and new measures with applications. *Economic Geography*, 46:269–292, 1970.

[11] Open Geospatial Consortium. Ogc geosparql - a geographic query language for rdf data, 2011. URL `http://www.opengeospatial.org/standards/requests/80` visited, 13-11-2013.

[12] Isabel F. Cruz, Venkat R. Ganesh, Claudio Caletti, and Pavan Reddy. GIVA: A Semantic Framework for Geospatial and Temporal Data Integration, Visualization, and Analytics. *ACM SIGSPATIAL*, pages 554–557, 2013.

[13] Isabel F. Cruz, Venkat R. Ganesh, and Iman Mirrezaei. Semantic extraction of geographic data from web tables for big data integration. 2013.

[14] Isabel F. Cruz, Flavio Palandri Antonelli, and Cosmin Stroe. Agreementmaker: Efficient matching for large real-world schemas and ontologies. *PVLDB*, 2(2):1586–1589, 2009.

[15] Isabel F. Cruz and William Sunna. Structural Alignment Methods with Applications to Geospatial Ontologies. *Transactions in GIS, Special Issue on*

Semantic Similarity Measurement and Geospatial Applications, 12(6):683–711, 2008.

[16] Isabel F. Cruz, William Sunna, and Anjli Chaudhry. Ontology alignment for real-world applications. In *National Conference on Digital Government Research (dg.o)*, pages 393–394, 2004.

[17] Isabel F. Cruz and Huiyong Xiao. Ontology Driven Data Integration in Heterogeneous Networks. In Andreas Tolk and Lakhmi Jain, editors, *Complex Systems in Knowledge-based Environments*, pages 75–97. Springer, 2009.

[18] Matt Duckham and Mike Worboys. An algebraic approach to automated geospatial information fusion. *International Journal of Geographical Information Science*, 19(5):537–557, 2005.

[19] Mats Dunkars. Matching of datasets. In *ScanGIS*, pages 67–78, 2003.

[20] Max J Egenhofer. Toward the semantic geospatial web. In *Proceedings of the 10th ACM international symposium on Advances in geographic information systems*, pages 1–4. ACM, 2002.

[21] Max J Egenhofer and Robert D Franzosa. Point-set topological spatial relations. *International Journal of Geographical Information System*, 5(2):161–174, 1991.

[22] Bruce Fraser and Myke Gluck. Usability of geospatial metadata or space-time matters. *Bulletin of the American Society for Information Science and Technology*, 25(6):24–28, 1999.

[23] GDAL Development Team. *GDAL - Geospatial Data Abstraction Library*.

Open Source Geospatial Foundation, 201x. URL http://www.gdal.org, visited 2013-09-10.

[24] Mordechai Haklay and Patrick Weber. Openstreetmap: User-generated street maps. *Pervasive Computing, IEEE*, 7(4):12–18, 2008.

[25] Antoine Isaac, Lourens Van Der Meij, Stefan Schlobach, and Shenghui Wang. An empirical study of instance-based ontology matching. In *The Semantic Web*, pages 253–266. Springer, 2007.

[26] Harry H Kelejian and Ingmar R Prucha. Specification and estimation of spatial autoregressive models with autoregressive and heteroskedastic disturbances. *Journal of Econometrics*, 157(1):53–67, 2010.

[27] C Kennedy, S Pincetl, and P Bunje. The study of urban metabolism and its applications to urban planning and design. *Environmental Pollution*, 159(8):1965–1973, 2011.

[28] David J Ketchen and Christopher L Shook. The application of cluster analysis in strategic management research: an analysis and critique. *Strategic management journal*, 17(6):441–458, 1996.

[29] Craig A Knoblock, Pedro Szekely, José Luis Ambite, Aman Goel, Shubham Gupta, Kristina Lerman, Maria Muslea, Mohsen Taheriyan, and Parag Mallick. Semi-automatically Mapping Structured Sources into the Semantic Web. pages 375–390. Springer, 2012.

[30] Dave Kolas, Ian Emmons, and Mike Dean. Efficient linked-list rdf indexing in parliament. *SSWS*, 9:17–32, 2009.

[31] Manolis Koubarakis, Mihai Datcu, Charalambos Kontoes, Ugo Di Giammatteo, Stefan Manegold, and Eva Klien. TELEIOS: a database-

powered virtual earth observatory. *Proceedings of the VLDB Endowment*, 5(12):2010–2013, 2012.

[32] Manolis Koubarakis and Kostis Kyzirakos. Modeling and querying metadata in the semantic sensor web: The model strdf and the query language stsparql. In *The semantic web: research and applications*, pages 425–439. Springer, 2010.

[33] Markus Krötzsch and Stefan Decker. Semantic web, 2012. URL `http://semanticweb.org/`, visited 18-11-2013.

[34] Kostis Kyzirakos, Manos Karpathiotakis, George Garbis, Charalampos Nikolaou, Konstantina Bereta, Michael Sioutis, Ioannis Papoutsis, Themistoklis Herekakis, Dimitrios Michail, Manolis Koubarakis, et al. Real time fire monitoring using semantic web and linked data technologies. In *International Semantic Web Conference (Posters & Demos)*, 2012.

[35] Kostis Kyzirakos, Manos Karpathiotakis, and Manolis Koubarakis. Strabon: A semantic geospatial dbms. In *The Semantic Web–ISWC 2012*, pages 295–311. Springer, 2012.

[36] Ora Lassila and Ralph R Swick. Resource description framework (rdf) model and syntax specification. 1999.

[37] Sang Lee. Developing a bivariate spatial association measure: an integration of pearson's r and moran's i. *Journal of geographical systems*, 3(4):369–385, 2001.

[38] Pierre Legendre. Spatial autocorrelation: trouble or new paradigm? *Ecology*, 74(6):1659–1673, 1993.

[39] Maurizio Lenzerini. Data integration: A theoretical perspective. In *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 233–246. ACM, 2002.

[40] Loet Leydesdorff. Similarity measures, author cocitation analysis, and information theory. *Journal of the American Society for Information Science and Technology*, 56(7):769–772, 2005.

[41] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, page 14. California, USA, 1967.

[42] Deborah L McGuinness, Frank Van Harmelen, et al. Owl web ontology language overview. *W3C recommendation*, 10(2004-03):10, 2004.

[43] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[44] Patrick AP Moran. Notes on continuous stochastic phenomena. *Biometrika*, 37(1/2):17–23, 1950.

[45] Charalampos Nikolaou, Kallirroi Dogani, Kostis Kyzirakos, and Manolis Koubarakis. Sextant: Browsing and mapping the ocean of linked geospatial data.

[46] Natalya F Noy, Michael Sintek, Stefan Decker, Monica Crubézy, Ray W Fergerson, and Mark A Musen. Creating semantic web contents with protege-2000. *Intelligent Systems, IEEE*, 16(2):60–71, 2001.

[47] Regina Obe and Leo Hsu. *PostGIS in action*. Manning Publications Co., 2011.

[48] J Keith Ord and Arthur Getis. Local spatial autocorrelation statistics: distributional issues and an application. *Geographical analysis*, 27(4):286–306, 1995.

[49] Matthew S Perry. *A framework to support spatial, temporal and thematic analytics over semantic web data*. PhD thesis, Wright State University, 2008.

[50] Christian Perwass. *Geometric algebra with applications in engineering*, volume 4. Springer, 2009.

[51] Stephanie Pincetl, Paul Bunje, and Tisha Holmes. An expanded urban metabolism method: Toward a systems approach for assessing urban energy processes and causes. *Landscape and Urban Planning*, 2012.

[52] John Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[53] David A Randell, Zhan Cui, and Anthony G Cohn. A spatial logic based on regions and connection. *KR*, 92:165–176, 1992.

[54] Philippe Rigaux, Michel Scholl, and Agnes Voisard. *Spatial databases: with application to GIS*. Morgan Kaufmann, 2001.

[55] N Rishe, M Gutierrez, A Selivonenko, and S Graham. Terrafly: A tool for visualizing and dispensing geospatial data. *Imaging Notes*, 20(2):22–23, 2005.

[56] Dr. Anthony C. Robinson. Maps and the geospatial revolution. URL `https://class.coursera.org/maps-001/class`, August 2013. Online Course held at `Coursera.org`.

[57] Lauren Rosenshein and L Scott. Spatial statistics best practices. *Redlands, CA. USA: ESRI*, 2011.

[58] JM Salas, A Harth, B Norton, LM Vilches, AD León, J Goodwin, C Stadler, S Anand, and D Harries. Neogeo vocabulary: Defining a shared rdf representation for geodata. *Public draft, NeoGeo, May*, 2011.

[59] Toby Segaran. *Programming collective intelligence: building smart web 2.0 applications*. O'Reilly Media, 2007.

[60] John Snow. *On the mode of communication of cholera*. John Churchill, 1855.

[61] Sergios Theodoridis and Konstantinos Koutroumbas. *Pattern Recognition, Third Edition*. Academic Press, Inc., Orlando, FL, USA, 2006.

[62] Waldo Tobler. On the first law of geography: A reply. *Annals of the Association of American Geographers*, 94(2):304–310, 2004.

[63] W3C. W3c semantic web interest group - basic geo (wgs84lat/long) vocabulary, 2006. URL `http://www.w3.org/2003/01/geo/`, visited 13-11-2013.

[64] W3C. W3c geospatial vocabulary - w3c incubator group report 23 october 2007, 2007. URL `http://www.w3.org/2005/Incubator/geo/XGR-geo/`, visited 13-11-2013.

[65] Volker Walter and Dieter Fritsch. Matching spatial data sets: a statistical approach. *International Journal of Geographical Information Science*, 13(5):445–473, 1999.

[66] Junli Wang, Zhijun Ding, and Changjun Jiang. GAOM: genetic algorithm based ontology matching. In *Services Computing, 2006. APSCC'06. IEEE Asia-Pacific Conference on*, pages 617–620. IEEE, 2006.

[67] Frank Warmerdam. The geospatial data abstraction library. In *Open Source Approaches in Spatial Data Handling*, pages 87–104. Springer, 2008.

[68] David WS Wong. The modifiable areal unit problem (MAUP). In *World-Minds: Geographical Perspectives on 100 Problems*, pages 571–575. Springer, 2004.

[69] Andy Woodruff. Bostonography: Crowdsourced neighborhood boundaries, July 2012. URL `http://bit.ly/18KfoSm`, visited 2013-09-23.

[70] Zhifeng Xiao, Lei Huang, and Xiaofang Zhai. Spatial information semantic query based on sparql. In *International Symposium on Spatial Analysis, Spatial-temporal Data Modeling, and Data Mining*, pages 74921P–74921P. International Society for Optics and Photonics, 2009.

[71] Xiaofang Zhai, Lei Huang, and Zhifeng Xiao. Geo-spatial query based on extended sparql. In *Geoinformatics, 2010 18th International Conference on*, pages 1–4. IEEE, 2010.

[72] Meng Zhang, Wei Shi, and Liqiu Meng. A generic matching algorithm for line networks of different resolutions. In *Workshop of ICA Commission on Generalization and Multiple Representation Computering Faculty of A Coruña University-Campus de Elviña, Spain*, 2005.

Appendix A

k-Nearest Neighbor Spatial Weights

A spatial weight matrix can be computed using a k-NN algorithm, that is based on the centroid distances, d_{ij} , between each pair of spatial units i and j[10]. Let centroid distances from each spatial unit i to all units $j \neq i$ be ranked

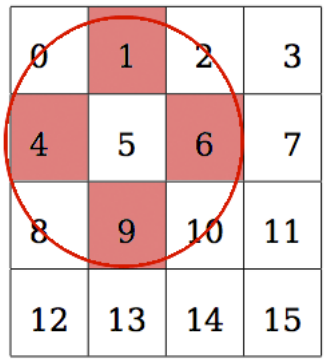


Figure A.1: Neighbors of the cell number 5, with $k = 1$

as follows: $d_{ij_1} \leq d_{ij_2} \leq ... \leq d_{ij_n}$. Then for each $k = 1, 2...n-1$, the set $N_k(i) = \{j(1), j(2), ..., j(k)\}$ contains the k closest units to i (where for simplicity we ignore ties)[26]. For each given k , the k-NN weight matrix, W , then has spatial weights of the form[10]:

$$w_{ij} = \begin{cases} 1 & j \in N_k(i) \\ 0 & otherwise \end{cases}$$

Alternatively, one can consider a symmetric version in which positive weights are assigned to all ij pairs for which at least one is among the k-NN of the other[10,26]:

$$w_{ij} = \begin{cases} 1 & j \in N_k(i) \vee i \in N_k(j) \\ 0 & otherwise \end{cases}$$

Figure A.1 shows an example of how neighbors are selected using k-NN.

Given a $3x3$ squared tessellation, for examples, the matrix of spatial weights computed with $k = 1$ would be:

Given tessellation:

0	1	2
3	4	5
6	7	8

$$, W = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

Appendix B

OWL Ontologies

```
1  <?xml version="1.0"?>
2
3  <!-- Precipitation -->
4
5  <!DOCTYPE rdf:RDF [
6     <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
7     <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
8     <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
9     <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
10 ]>
11
12 <rdf:RDF
       xmlns="http://www.cs.uic.edu/advis/ontologies/precipitation.owl#"
13     xml:base="http://www.cs.uic.edu/Advis/ontologies/precipitation.owl"
14     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
15     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
16     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
17     xmlns:owl="http://www.w3.org/2002/07/owl#">
18
19    <owl:Class rdf:about="#Lat">
```

```
20      <rdfs:label>Lat</rdfs:label>
21       <rdfs:subClassOf>
22         <owl:Class rdf:about="#Precipitation"/>
23      </rdfs:subClassOf>
24   </owl:Class>
25
26   <owl:Class rdf:about="#Long">
27      <rdfs:label>Long</rdfs:label>
28       <rdfs:subClassOf>
29         <owl:Class rdf:about="#Precipitation"/>
30      </rdfs:subClassOf>
31   </owl:Class>
32
33   <owl:Class rdf:about="#Val">
34      <rdfs:label>Val</rdfs:label>
35       <rdfs:subClassOf>
36         <owl:Class rdf:about="#Precipitation"/>
37      </rdfs:subClassOf>
38   </owl:Class>
39
40   <owl:Class rdf:about="#time">
41      <rdfs:label>Time</rdfs:label>
42       <rdfs:subClassOf>
43         <owl:Class rdf:about="#Precipitation"/>
44      </rdfs:subClassOf>
45   </owl:Class>
46
47   <owl:Class rdf:about="#Unit">
48      <rdfs:label>Unit</rdfs:label>
49       <rdfs:subClassOf>
50         <owl:Class rdf:about="#Val"/>
51      </rdfs:subClassOf>
52   </owl:Class>
```

```
53
54 </rdf:RDF>
```

```
1  <?xml version="1.0"?>
2
3  <!-- Rainfall -->
4
5  <!DOCTYPE rdf:RDF [
6     <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
7     <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
8     <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
9     <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
10 ]>
11
12 <rdf:RDF xmlns="http://www.cs.uic.edu/advis/ontologies/rainfall.owl#"
13     xml:base="http://www.cs.uic.edu/Advis/ontologies/rainfall.owl"
14     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
15     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
16     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
17     xmlns:owl="http://www.w3.org/2002/07/owl#">
18
19    <owl:Class rdf:about="#Latitude">
20       <rdfs:label>Latitude</rdfs:label>
21        <rdfs:subClassOf>
22          <owl:Class rdf:about="#Rainfall"/>
23       </rdfs:subClassOf>
24    </owl:Class>
25
26    <owl:Class rdf:about="#Longitude">
27       <rdfs:label>Longitude</rdfs:label>
28        <rdfs:subClassOf>
29          <owl:Class rdf:about="#Rainfall"/>
30       </rdfs:subClassOf>
```

```
31    </owl:Class>
32
33    <owl:Class rdf:about="#Value">
34       <rdfs:label>Value</rdfs:label>
35        <rdfs:subClassOf>
36          <owl:Class rdf:about="#Rainfall"/>
37       </rdfs:subClassOf>
38    </owl:Class>
39
40    <owl:Class rdf:about="#date_recorded">
41       <rdfs:label>date_recorded</rdfs:label>
42        <rdfs:subClassOf>
43          <owl:Class rdf:about="#Rainfall"/>
44       </rdfs:subClassOf>
45    </owl:Class>
46
47    <owl:Class rdf:about="#Unit">
48       <rdfs:label>Unit</rdfs:label>
49        <rdfs:subClassOf>
50          <owl:Class rdf:about="#Value"/>
51       </rdfs:subClassOf>
52    </owl:Class>
53
54 </rdf:RDF>
```

Appendix C

Published Paper

Isabel F. Cruz, Venkat R. Ganesh, Claudio Caletti, and Pavan Reddy. GIVA: A Semantic Framework for Geospatial and Temporal Data Integration, Visualization, and Analytics. ACM SIGSPATIAL, pages 554-557, 2013.

GIVA: A Semantic Framework for Geospatial and Temporal Data Integration, Visualization, and Analytics

Isabel F. Cruz, Venkat R. Ganesh, Claudio Caletti, Pavan Reddy
ADVIS Lab
University of Illinois at Chicago
{ifc, vsekar, ccaletti, preddy}@cs.uic.edu

ABSTRACT

The availability of a wide variety of geospatial datasets demands new mechanisms to perform their integrated analysis and visualization. In this demo paper, we describe our semantic framework, *GIVA*, for *Geospatial and temporal data Integration, Visualization, and Analytics*. Given a geographic region and a time interval, GIVA addresses the problem of accessing simultaneously several datasets and of establishing mappings between the underlying concepts and instances, using automatic methods. These methods must consider several challenges, such as those that arise from heterogeneous formats, lack of metadata, and multiple spatial and temporal data resolutions. A web interface lets users interact with a map and select datasets to be integrated, displaying as a result reports where values pertaining to different datasets are compared, analyzed, and visualized.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Spatial databases and GIS*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Query formulation*

Keywords

Spatial databases, GIS, Data integration, User interfaces

1. INTRODUCTION AND MOTIVATION

Spatio-temporal data are a fundamental resource for a variety of applications including those in public administration, transportation networks, and environmental studies. Within environmental studies, a possible scenario entails the study of two indicators: flu and precipitation to detect if they are correlated or if, for example, precipitation is a predictor of flu occurrences. Other scenarios may compare dependencies between these two indicators in two different cities. This scenario is depicted in Figure 1.

Several indicators can be studied at the same time and multiple dependencies considered in the emerging urban metabolism field [9]. To conduct these studies, vast amounts of geospatial information must be accessed and integrated using automatic methods, so that environmental scientists do not have to manually establish connections among highly heterogeneous data. We have been considering several scenarios motivated by two projects in which we collaborate, namely BURST (Building Urban Resilience and Sustainability)[1] and TerraFly [17]. Both projects are intended for experts in a variety of domains including urban metabolism and public health (BURST), hydrology and disaster mitigation (TerraFly), and transportation (BURST and TerraFly).

In this demo, we describe a semantic framework, *GIVA*, for *Geospatial and temporal data Integration, Visualization, and Analytics*. Using this framework, users can select regions in a map, specify time intervals, and select datasets to produce reports where values pertaining to different datasets are compared, analyzed, and visualized.

At the core of GIVA is its capability to deal with data, metadata, and their heterogeneity, by addressing the following issues: (1) *wide variety of formats*, both standardized (e.g., GML, KML, Shapefile, MapInfo TAB) and non-standardized (e.g., HTML tables and flat files); (2) *lack of metadata*, which stems in great part from non-standardized formats; (3) *multiple spatial and temporal resolutions*, due to different data acquisition techniques (e.g., surveys for census data and sensing methods for precipitation); (4) *different vocabularies and schemas*, which are created by diverse organizations (an example in public administration is that of land use codes [18]) and is illustrated for the two cities of Figure 1. In addition, there are overarching issues when dealing with geospatial data, namely that of uncertainty [15, 19].

2. FRAMEWORK

This section introduces our semantic framework (Figure 2) and describes briefly its components.

2.1 Data Extraction

Data of interest to geospatial information appears in a variety of formats, which we represent in the hierarchy of Figure 3. We refer to the formats approved by OGC[2] and that implement its standards as *standardized* and the rest as *non-standardized* data formats. A *geographic component* in these data formats uses geodetic systems such as WGS84 and geometric objects (e.g., polygon, polyline).

However, GIS data that are represented in web tables or text need special processing. Web tables are primarily con-

[1] http://www.burst.uic.edu
[2] http://www.opengeospatial.org/standards/is

SIGSPATIAL'13, Nov 05-08 2013, Orlando, FL, USA
ACM 978-1-4503-2521-9/13/11. http://dx.doi.org/10.1145/2525314.2525324

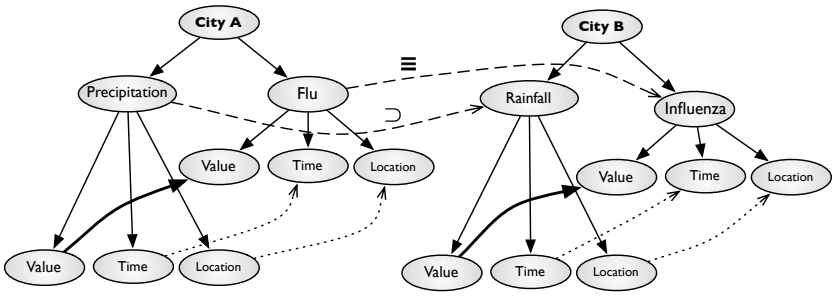


Figure 1: Comparison between two cities. Dashed edges represent concept similarity, dotted edges represent time or location similarity, and solid thick edges represent correlation between values of different concepts.

structed using the <*table*> tags for a variety of purposes such as, HTML forms, calendars, page layout, and relational data. However, in many cases web tables (even if they originate from relational databases) are not feature-rich because they do not contain clearly represented headers. The extraction of the corresponding feature-rich tables entails the identification of the headers (which are sometimes nested) and the storage of the table to produce a feature-rich table, which is stored in a structured file. For this kind of extraction we use a machine learning approach that encompasses a decision tree classifier model (C4.5) [16] using 20 different heuristics (including number of columns, rows, font size, and color) and trained it on 100 web tables with GIS data.

2.2 Data Translation

Data translation is the process of translating data from one format to another. Clean abstraction of data formats and methods to perform data translation are required for a sound solution to data integration [1]. Thus, before we attempt to create geospatial mappings between these data, they are translated into a common spatial data format. One issue is that *non-standardized* formats require semantic processing to identify the appropriate column headers that contain information about spatial coordinates and time stamps. We use string matching on the column headers and perform random sampling on the values to find pattern similarities. For instance, this ensures that an unclearly named column header (e.g., *Pos*) that contains geospatial coordinates (e.g., -85.46, 42.32) will be identified as indeed containing spatial coordinates and its name associated with a correct meaning. Further, data in *non-standardized* formats may contain implicit geographic components (e.g., Illinois). Special processing and techniques are required to identify these implicit geographic components as described in Section 2.4.2.

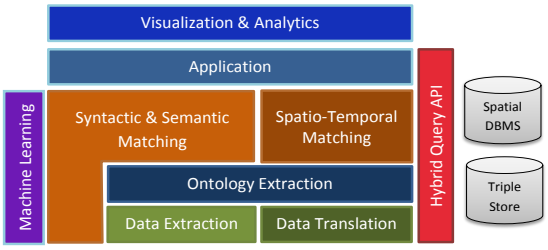


Figure 2: GIVA framework.

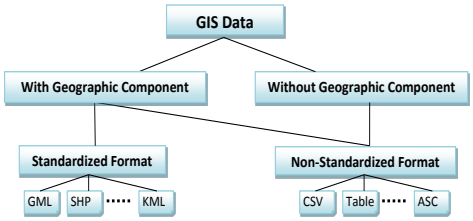


Figure 3: Hierarchy of spatial data types.

2.3 Ontology Extraction

The hierarchical characteristics of geospatial classification schemes can be modeled using a *part-of* or *is-a* relationship [4]. We have also devised methods to extract ontologies from a variety of formats, including from relational tables, XML, and RDF documents and to merge ontologies using matching and a data exchange approach by considering a global ontology [5]. This merging method is further described in Section 2.4.1 but we mention it here because it is related to recent ontology extraction approaches that use data exchange, machine learning, and user interaction [11].

2.4 Matching

The semantic integration of geospatial data requires the identification of correspondences among ontology concepts, properties, and instances, using syntactic and semantic characteristics of the ontologies, a process called *ontology matching* or *alignment*. The output of this process is a set of *mappings*. For spatial and temporal data, the spatial and temporal attributes of the data will also be considered.

2.4.1 Semantic Matching

Ontologies exhibit structural and conceptual heterogeneity, which we attribute to data creation by different organizations. The alignment of these ontologies require the sophisticated combination of various mechanisms geared to the identification of various classes of similarities. We use AgreementMaker [3], which is a proven system for ontology matching. AgreementMaker is also used for the mapping of the ontologies that are extracted from relational, XML, and RDF sources, enabling the mapping of similar concepts independently of where they appear (e.g., titles of relational tables, names of properties, or values). Data integration is achieved by rewriting a query expressed in terms of an ontology to another ontology using the established mappings [5].

AgreementMaker uses machine learning techniques to automatically change its configuration to maximize precision and recall [2].

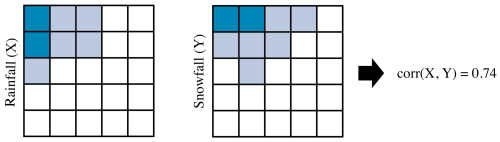


Figure 4: Comparison of values across data sets.

2.4.2 *Spatio-Temporal Matching*

Two important tasks to be addressed by this component are described below:

Resolving implicit geographic component. The process of assigning apposite geographic coordinates is referred to as *geocoding*, and that of identifying a geographic context is referred to as *geoparsing* [13]. For instance, geocoding helps in identifying the word *Illinois* and assigning its respective geographic component (e.g., state boundary of Illinois), if available. However, geospatial ambiguities often exist. For instance, the *Illinois river* may refer to the river in the state of Illinois or to the river of the same name in the state of Oregon. We implement geoparsing using a Named Entity Recognition (NER) technique and use semantic mappings as discussed in Section 2.4.1 for geocoding.

Managing spatial and temporal resolution. Heterogeneities in spatial and temporal resolution are introduced when data are published using different data acquisition techniques. For instance, precipitation data may be published associated with different areas depending on the density of the placement of the gages or the assumed coverage of each of them (e.g., a rectangle in a grid or a circle). We deal with this integration problem by introducing a new spatial resolution method that establishes a grid. The integration is performed by partitioning the space and computing a weighted average of the values in each of the original datasets, as illustrated in Figure 4. This produces a new dataset at a new resolution. Uncertainty increases when the dimensions of the grid are small in comparison with the measurement resolution, hence the grid dimensions can be defined depending on the dataset and the desired level of uncertainty. Temporal resolution can be resolved similarly.

This technique can be used when considering datasets about the same concept, for example *rainfall* or about different concepts, for example if the user wants to build a dataset about *precipitation* starting from two datasets about *rainfall* and *snowfall*. In this case, we can merge the datasets by adding the values of the two original datasets and by introducing an appropriate uncertainty value associated with this merging. Correlation between the datasets (instances) (see Figure 4) can assist the semantic matching process.

2.5 Storage Systems and Application

Our framework includes two different types of storage systems. A *Spatial DBMS* is used for storing and indexing geographic data and a *Triple Store* is used for handling semantic data and also to store the final alignments. A *Hybrid Query API* combines the query functionality of these two systems. An *Application* (web or stand-alone) is necessary to communicate with the other components of the framework and for the user interaction. This application also acts as Web Feature Service (WFS) interface to publish the integrated

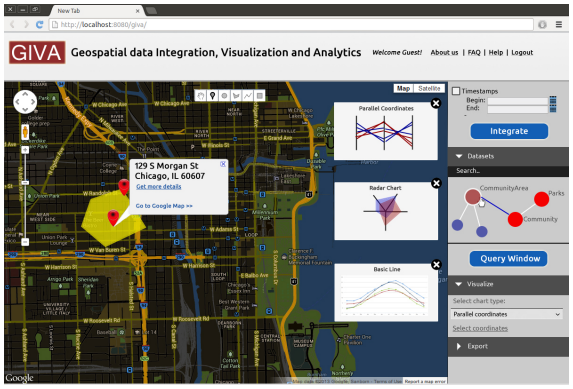


Figure 5: GIVA web interface design.

data to the domain stakeholders. For this demonstration, we develop a web application. The implementation details are described in Section 3.

2.6 Visualization and Analytics

We consider two components: one for visualization and the other one to support analytic methods.

The visualization component is fundamental to develop information processing in the context of different stages of scientific research and decision making. A use-based approach has long been proven to be an effective way to reinforce human understanding of abstract data [12]. We implement both an interactive map and plots for multidimensional visualizations, such as star plots and parallel coordinates graphs, where users will be able to display one or multiple variables simultaneously as shown in Figure 5.

The analytics component aims at providing the scientists with a suite of statistical models for spatial data exploration and multivariate analysis. We offer libraries for spatial autocorrelation and spatial regression as well as for factor analysis. In particular, we implement measures of spatial autocorrelation, such as Moran's I and Geary's C, and libraries to run OLS regression and spatial lag models. However, the analytics tool is meant to be an extensible part of the framework according to the needs of the scientists.

3. IMPLEMENTATION

We use PostGIS, a well-known spatial extension of the PostgreSQL database system, as our *Spatial DBMS* and OWLIM [10], an RDF database management system implemented in Java, as our *Triple store*. We develop a *Hybrid Query API* in Java to interlink PostGIS and SPARQL queries. The *Data Extraction* is developed using WEKA's [8] implementation of C4.5 algorithm to train the model and to extract the feature-rich web tables. The extracted tables are converted to a tab delimited file. *Data translation* is implemented in an XML framework that extends GDAL [7] to extract geospatial data with proper handling of geodetic systems. This module also implements the semantic processing techniques described in Section 2.2 to handle flat files (CSV and TSV). For *Ontology Extraction*, we use Apache OpenNLP[3] as an NLP toolkit and DBpedia[4] to receive suggestions for class names during the ontology construction. Automatic ontology extraction is a complex task and its performance depends on the organization of the schemas.

[3]http://opennlp.apache.org/
[4]http://dbpedia.org/

To overcome this issue, we allow users to optionally review the extracted ontology. The resulting RDF-Schema is used to generate triples. *Semantic Matching* is performed using AgreementMaker [3] and *Spatio-temporal Matching* uses the *Hybrid Query API* and an implementation of a matching mechanism as described in Section 2.4.2. A web interface is developed using the latest web technologies, namely AJAX and jQuery. For visualization and analytics, we use the interactive JavaScript visualization library—D3.js.[5]

4. RELATED WORK

A mobile application for an urban environment is presented by Della Valle et al. [6] to answer semantic queries such as finding the nearest tourist spots. Their data preparation module handles *Point* data from several ESRI Shapefiles, which are then manually processed and converted into an RDF format using PostGIS. These data are used along with an earlier platform that they developed, which provides SPARQL end points and a semantic framework with a reasoner to answer queries.

Urbmet[6] is an interactive map application to analyze urban data. Datasets about energy, material, and population are processed manually to provide reports for the very specific purpose of displaying potential spatial patterns that exist among them. Many similar applications can be found in OpenCityApps.[7] However, each of these applications is limited to providing visualizations or reports for pre-defined purposes and does not support data integration.

Middel presents an integrated framework for visualizing multivariate geodata [14]. The framework stores the spatial data mapped to uniform grids that cannot be changed and uses multinomial logistic regression to estimate characteristics of two different attributes for visualization. The drawbacks with this method are: (1) the possibility of a large amount of generated gridded data that could drastically reduce the performance of the system; (2) the potentially large addition of uncertainty in the partitioned grids that can impact the quality of the visualization.

In all of the systems we reviewed, there is no process that automatically integrates heterogeneous datasets. Also, the heterogeneity that is present in the data formats or metadata is either not resolved or is resolved manually.

5. CONCLUSIONS

We have introduced GIVA, a semantic framework that assists domain experts in integrating highly heterogeneous datasets and in analyzing and visualizing dependencies among them. The system supports three types of users: *administrator*, *domain expert*, and *casual user*, with different types of access. Given the complexity of the overall framework—in fact, we could not find any framework whose overall functionality can be compared in breadth with the one we propose—it is the case that every component of the framework offers opportunities for expansion and for improvement.

Acknowledgments

We would like to thank Roberto Tamassia, Matteo Palmonari, Tom Theis, Ning Ai, Sybil Derrible, Sam Dorevitch, Naphtali Rishe, and Goce Trajcevski for useful discussions. Thanks are due to Matt Dumford and Anna Anderson for help with the software development. This work was supported in part by NSF Awards CCF–1331800, IIS–1213013, IIS–1143926, and IIS–0812258 and by a UIC-IPCE Civic Engagement Research Fund Award.

[5]http://d3js.org/
[6]http://urbmet.org/about/
[7]http://opencityapps.org/

References

[1] S. Abiteboul, S. Cluet, T. Milo, P. Mogilevsky, J. Siméon, and S. Zohar. Tools for Data Translation and Integration. *IEEE Data Engineering Bulletin*, 22(1):3–8, 1999.
[2] I. F. Cruz, A. Fabiani, F. Caimi, C. Stroe, and M. Palmonari. Automatic Configuration Selection Using Ontology Matching Task Profiling. In *Extended Semantic Web Conference (ESWC)*, volume 7295 of *LNCS*, pages 179–194, 2012.
[3] I. F. Cruz, F. Palandri Antonelli, and C. Stroe. AgreementMaker: Efficient Matching for Large Real-World Schemas and Ontologies. *PVLDB*, 2(2):1586–1589, 2009.
[4] I. F. Cruz and W. Sunna. Structural Alignment Methods with Applications to Geospatial Ontologies. *Transactions in GIS, Special Issue on Semantic Similarity Measurement and Geospatial Applications*, 12(6):683–711, 2008.
[5] I. F. Cruz and H. Xiao. Ontology Driven Data Integration in Heterogeneous Networks. In A. Tolk and L. Jain, editors, *Complex Systems in Knowledge-based Environments*, pages 75–97. Springer, 2009.
[6] E. Della Valle, I. Celino, and D. Dell'Aglio. The Experience of Realizing a Semantic Web Urban Computing Application. *Transactions in GIS*, 14(2):163–181, 2010.
[7] GDAL Development Team. *GDAL - Geospatial Data Abstraction Library, Version 1.10.0.* Open Source Geospatial Foundation, 2013.
[8] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA Data Mining Software: An Update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.
[9] C. Kennedy, S. Pincetl, and P. Bunje. The Study of Urban Metabolism and its Applications to Urban Planning and Design. *Environmental Pollution*, 159(8):1965–1973, 2011.
[10] A. Kiryakov, D. Ognyanov, and D. Manov. OWLIM–A Pragmatic Semantic Repository for OWL. In *Web Information Systems Engineering–WISE 2005 Workshops*, pages 182–192. Springer, 2005.
[11] C. A. Knoblock, P. Szekely, J. L. Ambite, A. Goel, S. Gupta, K. Lerman, M. Muslea, M. Taheriyan, and P. Mallick. Semi-automatically Mapping Structured Sources into the Semantic Web. In *International Semantic Web Conference (ISWC)*, pages 375–390. Springer, 2012.
[12] A. M. MacEachren and M.-J. Kraak. Exploratory Cartographic Visualization: Advancing the Agenda. *Computers & Geosciences*, 23(4):335–343, 1997.
[13] K. S. McCurley. Geospatial Mapping and Navigation of the Web. In *International World Wide Web Conference (WWW)*, pages 221–229. ACM, 2001.
[14] A. Middel. A Framework for Visualizing Multivariate Geodata. In *Visualization of Large and Unstructured Data Sets*, pages 13–22, 2007.
[15] D. Pfoser, N. Tryfona, and C. S. Jensen. Indeterminacy and Spatiotemporal Data: Basic Definitions and Case Study. *GeoInformatica*, 9(3):211–236, 2005.
[16] J. R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann Publishers, San Mateo, CA, 1993.
[17] N. Rishe, B. Furht, M. Adjouadi, A. Barreto, E. Cheremisina, D. Davis, O. Wolfson, N. Adam, Y. Yesha, and Y. Yesha. Geospatial Data Management with TerraFly. In *Handbook of Data Intensive Computing*, pages 637–665. Springer, 2011.
[18] N. Wiegand, D. Patterson, N. Zhou, S. Ventura, and I. F. Cruz. Querying Heterogeneous Land Use Data: Problems and Potential. In *National Conference for Digital Government Research (dg.o)*, pages 115–121, 2002.
[19] M. Worboys. Computation with Imprecise Geospatial Data. *Computers, Environment and Urban Systems*, 22(2):85–106, 1998.