Analysis and Visualization of Large-Scale Time Series Network Data

Patricia Morreale, Allan Goncalves, and Carlos Silva

Department of Computer Science, Kean University, Union, NJ USA
{pmorreal,goncalal,salvadca}@kean.edu

Abstract. Large amounts of data ("big data") are readily available and collected daily by global networks worldwide. However, much of the real-time utility of this data is not realized, as data analysis tools for very large datasets, particularly time series data are cumbersome. A methodology for data cleaning and preparation needed to support big data analysis is presented, along with a comparative examination of three widely available data mining tools. This methodology and offered tools are used for analysis of a large-scale time series dataset of environmental data. The case study of environmental data analysis is presented as visualization, providing future direction for data mining on massive data sets gathered from global networks, and an illustration of the use of big data technology for predictive data modeling and assessment.

1 Introduction

Increasingly large data sets are resulting from global data networks. For example, the United States government's National Oceanic and Atmospheric Administration (NOAA) compiles daily readings of weather conditions from monitoring stations located around the world. These records are freely available. While such a large amount of data is readily available, the value of the data is not always evident. In this chapter, large time series data was mined and analyzed using data mining algorithms to find patterns. In this specific instance, the patterns identified could result in better weather predictions in the future.

Building on earlier work [1,2,3,4], two separate datasets from NOAA were used. One was the Global Summary of Day (GSOD) dataset [5], which currently has data from 29,620 stations. The second was the Global Historical Climatology Network (GHCN) dataset [6], which currently has data from 77,468 stations. In previous projects, the datasets were downloaded from NOAA's FTP servers and made locally available on our local database server. Each of the GSOD stations collect 10 different types of data (precipitation, snow depth, wind speeds, etc.), while the GHCN stations can collect over 80 different types of data, although the majority only collect 3 or 4 types. Some stations have been collecting data for over 100 years. Both datasets combined consist of over 2.62 billion rows (records) in our database.

F. Xhafa et al. (eds.), *Modeling and Processing for Next-Generation Big-Data Technologies*,
Modeling and Optimization in Science and Technologies 4, DOI: 10.1007/978-3-319-09177-8_9

Data from both datasets was mined using Weka, RapidMiner, and Orange, which are free data mining programs. Each of these programs has a variety of data mining algorithms which were applied to our data. However, before mining, the data had to be converted into a format which allowed it to be properly mined. The problem was solved by developing custom Java programs to rearrange the data.

In this illustrated example, the goal was to find any patterns existing in the data, help predict future significant weather events (snowstorm, hurricane, etc.), and visualize results in a meaningful format. It is also hoped that this work mining large-scale datasets will help others do the same with any dataset of similar magnitude. Although global data was available, the portion of the dataset used was New Jersey, as it has its own special microclimate. By using data from some of New Jersey's extreme weather events as starting points, this research was able to look for patterns that may assist in predicting such events in the future. Examples of extreme events in New Jersey include the unpredicted snowstorm of October 2011, the December 26, 2010 snowstorm (24"-30" accumulation), and a tornado Supercell that hit the state in August 2008.

2 Big Data Applications

The objective was to use the very large amount of data previously imported into a local database server and run data mining algorithms against it to find patterns. This approach was similar to one taken to find relationships in medical data from patients with diabetes [7]. Fields such as telemedicine and environmental sustainability offer great opportunities for big data analysis and visualization. For environmental big data analysis, a variety of popular data mining software was used to evaluate the data to see if one product provided superior results. The software products used were Weka [8], RapidMiner [9], and Orange [10]. Additionally, the raw data was graphed [11, 12] to see if any patterns were identified through visual inspection which the mining software might overlook. By mining the data, a trend was expected in environment/weather. Possible results could be evidence of global warming, colder winters, warmer summers (heat waves), stronger/weaker storms, or more/less large storms. This chapter is an extension of [1, 2], with a specific application of environmental sustainability.

2.1 Environmental Sustainability

Environmental sustainability encompasses several stages. Initially, environmental sustainability referred to development that minimized environmental impact. However, in established areas, such as urban communities, environmental sustainability includes detection of potential problems, monitoring the impact of potential or actual problems, and working to reduce adverse impact of identified threats to environmental sustainability.

The repetitive nature of threats to environmental sustainability in urban environments, such as the underpass that consistently floods during heavy rains, or the air quality that predictably degrades over the course of a workday, is the stuff of urban

legend. Neighborhood residents and regular visitors to the area may generally know the hazards of a particular urban spot, but sharing the knowledge of destructive or hazardous patterns with organizations which might be able to remediate or prevent such regular environmental degradation is not easily done. Rather, at each instance of an environmental threat, the flooded underpass or poor air is addressed as a public safety crisis and personnel and resources are deployed in an emergency manner to provide appropriate traffic rerouting or medical attention.

In addition to critical events which threaten urban environmental conditions, more insidious, slowly evolving circumstances which may result in future urban crises are not monitored. For example, traffic volume on highly used intersections or bridges is not monitored for increasing noise or volume, which might result in a corresponding increase in hazardous emissions or stress fractures. When urban threats are identified, the response process can be aggravated as traffic comes to a standstill or is rerouted, hindering, or delaying emergency response personnel.

Both immediate and slower threats to urban environmental sustainability are dealt with on an 'as occurring' basis, with no anticipation or preventive action taken to avert or decrease the impact of the urban threat. The result, over the past years, has been an increase in urban crisis management, rather than an increased understanding of how our urban environments could be better managed for best use of all our resources – including resources for public safety and environmental sustainability.

The increasing age of urban infrastructures, and the further awareness of environmental hazards in our midst highlights that the management of environmental threats on an 'as needed' basis is no longer feasible, particularly as the cost of managing an environmental crisis can exceed the cost of preventing an environmental crisis. With the potential to gather data in a real-time manner from urban sites, the opportunity for anticipatory preparation and preventive action prior to urban environmental events has become possible. Specifically, street level mapping, using real-time information, is now possible, with the integration of new tools and technology, such as geographic information systems and sensors.

Environmental sustainability in an urban environment is challenging. While numerous measures of environmental sustainability, including air quality, rainfall, and temperature, are possible, the group assessment of these measured parameters is not as easily done. Air quality alone is composed of a variety of measurements, such as airborne particulate matter (PM10), nitrogen dioxide (NO2), Ozone (O3), carbon monoxide (CO) and carbon dioxide (CO2). Road traffic is the main cause of NO2 and CO. While simple environmental solutions such as timing traffic lights are identified as saving billions in fuel consumption and reducing air-pollution (i.e., improving air quality) by as much as 20%, the technological underpinnings to accomplish this have not been developed and deployed on an appropriate scale for urban data gathering and correlation. Furthermore, the use of predictive models and tools, such as data mining, to identify patterns in support or opposed to environmental sustainability is not commonly done in an urban setting. While hurricane, earthquake, and other extreme weather events occur and the aftermath is dramatically presented, more mundane but not less impacting events such as urban flash flooding, chemical spills on city roads, and other environmental events are not anticipated or measured while occurring or

developing, with intent to reduce in scope and damage. By gathering data locally, for assessment and prediction, areas, and events can be identified that might harm environmental sustainability. This knowledge can be used to avoid or disable what might have previously been an urban environmental disaster.

2.2 Data Mining for Trend Identification

Data mining [13] can be referred to as 'knowledge discovery in databases', and is a key element of the "big data" analysis project. Of the four core data mining tasks:

- Cluster analysis
- Predictive modeling
- Anomaly detection
- Association analysis

both predictive modeling and anomaly detection are used in the big data analysis project detailed here. "Predictive modeling" can be further defined by two types of tasks: classification, used for discrete target variables, and regression, which is used for continuous target variables.

Forecasting the future value of a variable, such as would be done in a model of an urban ecosystem, is a regression task of predictive modeling, as the values being measured and forecast are continuous-valued attributes. In both tasks of predictive modeling, the goal is to develop a model which minimizes the error between predicted and true values of the target variables. By doing so, the objective is to identify crucial thresholds that can be monitored and assessed in real-time so that any action or alert may be automatic and high responsive.

"Anomaly detection" is also crucial to the success of big data modeling. Formally stated anomaly detection is the task of identifying events or measured characteristics which are different from the rest of the data or the expected measurement. These anomalies are often the source of the understanding of rare or infrequent events. However, not all anomalies are critical events, meriting escalation, and further investigation. A good anomaly detection mechanism must be able to detect non-normal events or measurements, and then validate such events as being outside of expectations – a high detection rate and low false alarm rate is desired, as these define the critical success rate of the application.

2.3 Sensor Networks and Visualization

Sensor networks have become part of our everyday lives and attract wide interest from industry due to their potential diversity of applications, with a strong expectation that outdoor and environmental uses will dominate the application space [14]. However, actual deployment experience is limited and application development has been further restricted [15]. Previous work in sensors for structure monitoring [16], urban flash flood awareness [17], and mobile emissions monitoring [18] has been initiated, but not to the extent and geographical contextual presentation outlined here.

The overall objective of the big data environmental network implementation is the gathering of environmental information in real-time and storing the data in a database so that, the data can be visually presented in a geographic context for maximum understanding. Ideally, the big data environmental network is an implementation of a wireless environmental sensing network for urban ecosystem monitoring and environmental sustainability. By measuring environmental factors and storing the data for comparison with future data gathered, the changes in data measured over time can be assessed. Furthermore, if a change in one measured variable is detected, examination of another measured variable may be needed to correlate the information, and determine if the measured conditions are declining or advancing over time. Known as 'exception mining,' this assessment can also be visually presented in a geographical context, for appropriate understanding and preventive or divertive action.

2.4 Network Application Design

The network design used was that of a system of distributed sensors, reporting to a base station, which was then connected to a server between the network and the application. The focus was on total application design, from data storage in the relational database, to final interpretation, and presentation in the geographical context.

A. Visual Presentation

Application development included a collection of data, which was archived into a database. This was accomplished by using SQL, a relational database system. To clearly understand and visualize the importance, functionality, and advantages provided by the wireless sensor network, the data must be clearly represented. In order to do so, a programming language or framework is needed that provides the ability to quickly gather data and accurately represent each of our sensors.

After consideration of availability, scalability, and recognition, as well as understanding the well-defined API and number of tutorials available, the Google Maps framework was selected. The Google Maps API allows the user to use Google Maps on individual websites, with JavaScript. In additional, a number of different utilities are available.

Google Maps provides an additional advantage, as the nodes represented on the map and the data contained in each one of the nodes can be setup using XML and additional nodes or markers can be added with ease. Once the decision to use Google Maps was made, the software development effort shifted from data representation efforts to working on accessing the actual real-time data from the wireless sensor network server. For Google Maps to process this new data and properly represent it, an XML file is generated. Fig. 1 shows the exchange between the web server and the wireless sensor network server.

As data is sent in from the nodes, it is passed through the base station to the Perl XML Parser, which parses the incoming data and filters out unwanted packets. The result remaining is the desired data packet set.

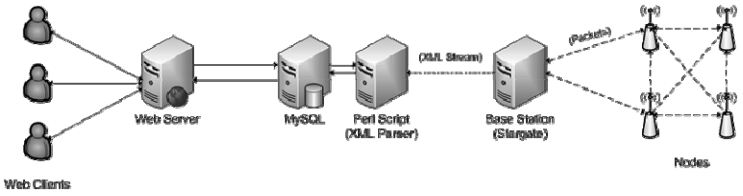

Fig. 1. Data exchange between web server and wireless sensor network server

The wireless sensor network data is gathered in a database, which is then presented in a visual context. Google Maps can be used to present the information in a geographical context. By clicking on the sensor node, real-time information is presented to the viewer, in appropriate context. The raw data, collected from the sensor, is appropriately converted to standard units for display and understanding.

In addition to real-time data presentation in a geographical context, a temporal presentation, using time and date information, has also been developed. An illustration of this can be seen in Fig. 2. A query by sensor presents the specific details of that sensor at one point in time, as well as providing a comparison of the sensor's status for prior dates and times. More than one sensor measurement can be overlaid on the chart, which permits correlation of events and times precisely with sensors.

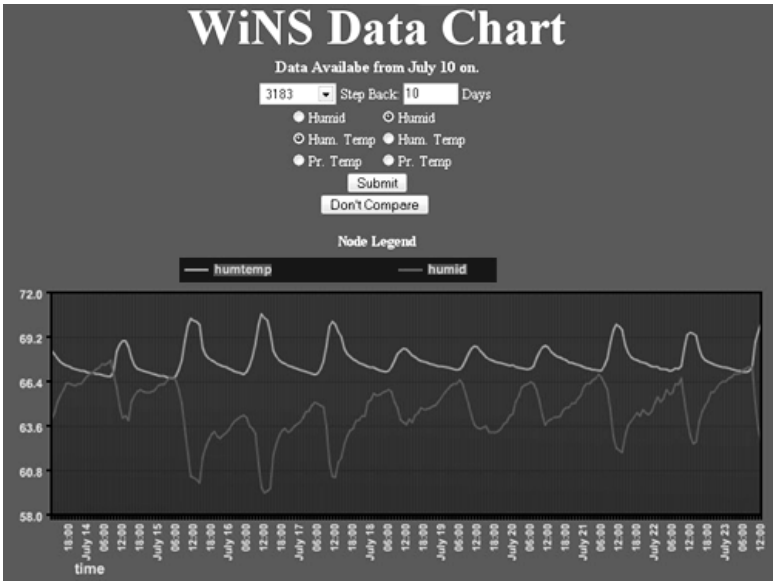

Fig. 2. Visual presentation of data from two sensors with date

While data has been gathered by sensors before, the correlation and presentation of this environmental real-time information in a geographical context, mined from a very large dataset, in addition to providing support for historical and temporal comparison, is innovative.

B. Integration of Geovisualization and Data Mining

The presentation of data in real-time for contextual understanding is only one aspect of the big data application. The archived information in the database permits before and after animations to be developed, using time gradients to show how the measured variables have changed in the preceding time, or are expected to change in the future, based on predictive algorithms, taking into account the reported variables, such as wind speed, in the case of a chemical spill and the potential migration over an urban community, for example.

Once the data of the multiple network nodes has been collected for a long period of time, it is desirable to reveal patterns, if any, in the data. For example, with regard to spatial-temporal variations of air quality along the streets or at intersections, patterns regarding daily and seasonal cycles of air quality, change, and decay over distances from the intersection can be directly visualized through an interactive and animated visualization environment. Furthermore, the visualization tool can be used together with numerical data mining algorithms to model quantitative relationships between air quality and other factors such as weather and traffic conditions. Numerical data mining algorithms such as supervised learning can be easily integrated here. The findings can then be applied to simulate the spatial-temporal air quality variations given arbitrary weather and traffic conditions for the purpose of predictive modeling.

Human visual perception offers a broadband channel for information flow and excellent pattern recognition capabilities that facilitate knowledge discover and the detection of spatial-temporal relations [19]. An effective tool to explore geographic data and communicate geographic information to private or public audiences, geovisualization has long been used for data exploration and pattern recognition. The approach presented and discussed here integrates geovisualization with data mining to reveal spatial-temporal patterns embedded in the data collected by the sensor network over time. While prior work [20] has approached such an idea, the work presented here is the first to visually present the data correlations.

2.5 Big Data Analysis for Environmental Sustainability

Wireless sensor network applications, and the very large dataset of gathered data associated with them, are an emerging area of technology which will benefit organizations and governments with valuable real-time data. In order to properly use such data, a strong, dynamic, and user-friendly interface is needed which allows individuals to clearly see how measured conditions, such as environmental circumstances, are changing over time. The visual depiction of urban environmental events, for example, will permit anticipatory or preventive actions to be taken in advance of adverse human and ecological impact. The use of data mining techniques on the data gathered by the wireless sensor network permits the identification of past patterns and developing trends in air quality or urban flooding, for example. The network and interface illustrated here accomplishes the goals of real-time information gathering and display for environmental sustainability and further work is underway to improve and refine the solution presented here.

Additional research underway includes a case study where a number of exploratory spatial data analysis (ESDA) techniques will be tested to facilitate the visual detection of spatial-temporal patterns of air quality in relation to weather conditions. ESDA techniques being tested in the case study include temporal brushing [21] and temporal focusing [22], temporal reexpression through multiscale data aggregation [23] and static visual bench marking [24]. Animation of the temporal data will enable common users to visualize the change of air quality over space and time. With temporal brushing and focusing, the use is not only a passive viewer of the information, but can interact with the animation and learn actively. Temporal reexpression through multiscale data aggregation provides an opportunity to directly visualize the daily and seasonal cycles of air quality change. Finally, using static visual benchmarking, the air quality level at any recorded time spot can be compared visually with health standards and give the viewer a direct alert on how high air quality is affecting human health. Efforts continue to integrate the geovisualization environment with a knowledge discovery procedure for data mining.

3 Visualization and Pattern Identification in Big Data

Visualization of massively large datasets presents two significant problems. First, the dataset must be prepared for visualization, and traditional dataset manipulation methods fail due to lack of temporary storage or memory. The second problem is the presentation of the data in the visual media, particularly real-time visualization of streaming time series data. Visualization of data patterns, particularly 3D visualization, represents one of the most significant emerging areas of research. Particularly for geographic and environmental systems, knowledge discovery and 3D visualization is a highly active area of inquiry. Recent advances in association rule mining for time series data or data streams makes 3D visualization and pattern identification on time series data possible.

In streaming time series data the problem is made more challenging due to the dynamic nature of the data. Emerging algorithms permit the identification of time-series motifs [25] which can be used as part of a real-time data mining visualization application. Geographic and environmental systems frequently use sensor networks or other unmanned reporting stations to gather large volumes of data which are archived in very large databases [26]. Not all the data gathered is important or significant. However the sheer volume of data often clouds and obscures critical data which causes it to be ignored or missed.

The research presented here outlines an ongoing research project working to visualize the data from national repositories in two very large datasets. Problems encountered include dataset navigation, including storage and searching, data preparation for visualization, and presentation.

Data filtering and analysis are critical tasks in the process of identifying and visualizing the knowledge contained in large datasets, which is needed for informed decision making. This research is developing approaches for time series data which will permit pattern identification and 3D visualization. Research outcomes include as-

sessment of data mining techniques for streaming time series data, as well as interpretive algorithms, and visualization methods which will permit relevant information to be extracted and understood quickly and appropriately.

3.1 Large-Scale Data for Visualization

This research works with datasets from the National Oceanic and Atmospheric Administration (NOAA), a federal agency in the U.S., focused on the condition of the oceans and the atmosphere. The purpose of the research project is to take meteorological data and analyze it to identify patterns that could help to predict future weather events. Data from the GHCN (Global Historical Climatology Network) dataset [3] was initially used. Earlier research had worked with NOAA's Integrated Surface Dataset (ISD) [4]. Both of these datasets are open access and the volume of streaming time series data was significant and growing.

The GHCN dataset consists of meteorological data from over 76,000 stations worldwide with over 50 different searchable element types. Examples of element types include minimum and maximum temperature, precipitation amounts, cloudiness levels, and 24-hour wind movement. Each station collects data on different element types.

3.2 Time Series Data Analysis

Searching for temporal association rules in time series databases is important for discovering relationships between various attributes contained in the database and time. Association rules mining provides information in an "if-then" format. Because time series data is being analyzed for this research, time lags are included to find more interesting relationships within the data. A software package from Universidad de La Rioja's EDMANS Groups was used to preprocess and analyze the time series from the NOAA datasets. The software package is called *KDSeries* and was created using *R*, a language for statistical computing.

The *KDSeries* package contains several functions that preprocess the time series data so knowledge discovery becomes easier and more efficient. The first step in preprocessing is filtering. The time series are filtered using a sliding-window filter chosen by the user. The filters included in *KDSeries* are Gaussian, rectangular, maximum, minimum, median, and a filter based on the Fast Fourier Transform. Important minimum and maximum points of the filtered time series are then identified. The optima are used to identify important episodes in the time series. The episodes include increasing, decreasing, horizontal and over, below, or between a user-defined threshold.

After simple and complex episodes are defined, each episode is view as an item to create a transactional database. Another R-based software package, *arules*, makes this possible. *Arules* provide algorithms that seek out items that appear within a window of a width defined by the user. From there, temporal association rules are then extracted from the database.

The first algorithm being used to extract the temporal association rules is the Eclat algorithm. Eclat (Equivalence Class Clustering and Bottom-up Lattice Traversal) is an efficient algorithm that generates frequent item sets in a depth-first manner. Other

algorithms such as Aproiri and FP-growth will then be used to extract association rules and compared and contrasted with each other. This work is ongoing.

3.3 Methodology

The data used is located on NOAA's FTP site in the form of .dly files. Each station has its own .dly file which is updated daily (if the station still collects data). Each .dly file has all the data that has ever been collected for that station. Whenever new data is added for a station, it is appended to the end of the current .dly file, which presents the problem that each file must be downloaded over again to keep our local database current. To obtain the data, a Java-based program was built that would download every file in the folder holding the .dly files. The Java program used the Apache Commons Net library to download the files from the NOAA FTP server.

After downloading all of the .dly files, the Java program opens an input stream to each of the downloaded files (one at a time). Each line of the .dly files contains a separate data record, so the Java program would read in each line and use it to form a MySQL "INSERT" statement that would be used to place data into a local database. At one point, space was exhausted on the local machine, and researchers had to upgrade the hard disk from ~200 GB to 256 GB to continue inserting data into the relational database.

Once all of the data was placed into the local database, a web interface was built that allowed users to search the dataset (Fig. 3). The interface allows users to search by country, state (within the United States), date range, and values that are <, <=, >, >=, !=, or == to any chosen value. Because the dataset contains the value -9999 for any record that is invalid or was not collected, the web interface also has the option to exclude any -9999 values from the results. The results are output with each line containing a different data result, and each result consisting of month, day, year, and data value.

Fig. 3. Query screen of web interface to NOAA GHCD dataset

For visualizing the data, the first step was to plot the location of each station using Google Earth. The NOAA FTP server also has a file that lists the longitude, latitude, and elevation of each station, so this information was placed into a separate table in our database. Next, a PHP script from the Google Earth website was customized to support queries to the database for the location of each station and then format the results in KML. This KML data is then loaded into the browser-based version of Google Earth on the same webpage. A separate PHP script was built that allows the user to search for stations in the entire world, by country, or by state (if searching within the US) (Fig. 4) and results from the query are graphed (Fig. 5).

Fig. 4. Web interface to station selection from GHCD dataset

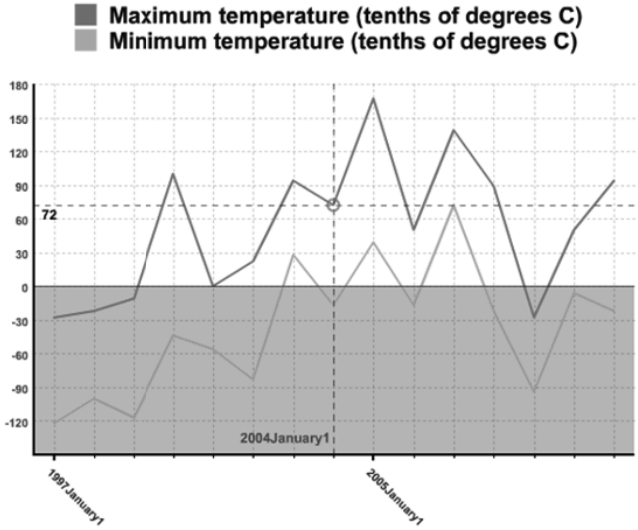

Fig. 5. Real-time visualization of a user query to the GHCD dataset

This visualization can be integrated into a Google Earth display (Fig. 6).

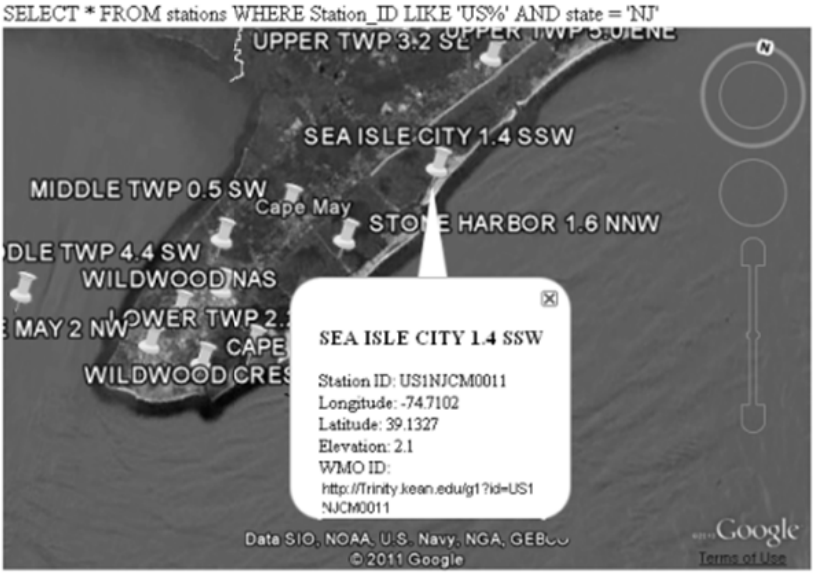

Fig. 6. NOAA GHCD reporting stations in Google Earth

4 Preparing Big Data for Analysis

Data type dissimilarities in big data are common. Significant time can be spent on data cleaning and validation. The effort presented here resulted in a methodology to pull data from the database and convert it to a data mining friendly format, with conservative use of computer memory and storage.

The two NOAA datasets were stored in separate tables. For the GSOD dataset, each station always collects the same 10 types of weather data (wind speed, precipitation, etc.), each type of data was given its own column in a table within the database. Unfortunately, the GHCN stations can collect any of over 80 different types of weather data, so it would be impractical to assign each data type its own column. For this reason, one column was used to store the type of data the record would hold, and a second column to store the actual value collected.

After populating the database, it was learned that the GHCN data needed to be reorganized so that it could be mined. In the GSOD dataset, all types of data are stored in the same record, which makes it very easy for mining software to compare all aspects of one day to another day. In contrast, the GHCN dataset has multiple records for each day to accommodate for its wide variety of data types, which we cannot use with data mining algorithms. This is primarily because the GHCN data types are stored as different measurements and cannot be directly compared to each other. For example, snowfall is measured in millimeters, while maximum temperature is measured in tenths of Celsius degrees.

It would be a massive undertaking to remake the GHCN table within the local database (it would also be quite inefficient if we made a column for each data type), so a Java program was developed to pull data from the database and convert it to a

data mining friendly format. The program queries for data within a specific date range and station range (numbers were assigned to the stations), and for the specific data types which we wanted to retrieve. The program then compiles the results into a file, in which each station has a single record for each day. The file can be read like a table in which each data type has its own column. This lets data be efficiently retrieved from the database, while still allowing the data to be properly organized for mining.

A subset of the data is requested at one time as the conversion process can take hours or days if too much data is requested. Most mining programs cannot handle such large amounts of data, and not all element types are commonly used. Some mining algorithms will not run if a data type is missing too many entries, which would be the case if some of the less commonly collected data types were used. This program is used to retrieve data from both the GSOD and GHCN datasets to reduce time wasted on retrieving the same data multiple times, and to remove the need to have mining programs connect to our database server.

A second Java program was written that converts commas to tabs. The mining program Orange does not read CSV files, but does read tab-delimited files. Originally, commas were used to separate data values, so a second program was made that would quickly convert all commas to tabs in order to analyze the data with Orange.

5 RapidMiner vs. Weka vs. Orange

Three programs were used to mine the data. The most success came from RapidMiner, with quite a bit less success experienced with Weka and Orange. All the three programs have a similar setup in which different functions are dragged and dropped onto the interface screens and then connected together to run the chosen algorithms. Each of the programs gave operational issues at times, but RapidMiner seemed to be the most stable and useful.

Initially, mining began with Weka, but a few key issues made it clear that Weka should be dropped early on. First, Weka's "Knowledge Flow" program, which contains mining functions, was not able to connect to our local database server (before we built our conversion program). Second, Weka has a lot of mining algorithms, but little explanation of how to use them. Frustrated with these problems, RapidMiner was tried.

RapidMiner has a large amount of mining functions, and it has an extension that gives it some functions from Weka. It also has a file import wizard that helps ensure that data is correctly imported into the program. Most importantly, there is a small guide for each mining algorithm on the bottom right-hand corner of the screen that is automatically shown whenever a function is selected. The guide explains exactly what a function is for, how to use it, and what its inputs and outputs are. RapidMiner also has a search feature that lets users quickly find the algorithms they want to use. In addition, it has a wizard called "Automatic System Construction" that runs different algorithms on data to determine which ones may yield results [11]. We did not have much success with this feature. Out of all the programs, only RapidMiner provided results.

Despite those positive features, there were two main problems with RapidMiner. The first problem was that it would crash if functions with more than ~5MB of input

data were used. The second problem was that many functions would only run if numeric data were converted to nominal data. This means that those numeric values would be treated as though they were words, thus entirely removing their important numeric properties. The nominal values have no relation to each other, such as difference (between two values), but are treated as separate, equal instances.

Orange was the last program we tried using. Before using Orange the CSV files were converted to tab-delimited files, but this was not really an issue. A feature that stands out in Orange is that Orange will only permit the addition of a new function if it can be attached to one that has already been chosen. This removes some guesswork, allows us to easily see what we can use, and identify functions that we may have otherwise overlooked. Like RapidMiner, Orange would sometimes crash if it received too much data.

6 Mining the Data

In an effort to find patterns, a variety of algorithms were used. There are three main algorithms that provided some form of results, and those were the

- association rules algorithms
- various decision tree algorithms
- Naïve Bayes algorithm

All algorithms were run on a computer using a 2.66 GHz Intel Core 2 Duo E8200 processor, 3 GB of RAM, and the 32-bit version of Windows XP. Times listed below will be for algorithms analyzing a 5.39MB file containing 238,839 records from the GHCN dataset.

The purpose of an *association rules algorithm* is to find relationships between different columns of data (data types). An example could be if a day's minimum temperature was above 75˚F then the month is June, July, or August (summer months in New Jersey). Weka had originally given some very simple relationships like this, but nothing of significance. RapidMiner required conversion of most numeric values to nominal values, so when these algorithms were used the rules produced were not helpful. The average time for running an association rules algorithm in RapidMiner on the GHCN file was 8.3 seconds.

The purpose of the *decision tree algorithms* is very similar to the association rules in that it tries to find relationships between different data columns. These relationships are then used to form a tree that leads from one column to another until you arrive at a leaf node. A data column which will be a leaf node in the tree must be selected (assign it as a label), as this will be the value which you are trying to predict. RapidMiner always assembles the decision trees in any arrangement it chooses, so it is not possible to designate the position of specific data columns in the traversal of the tree.

RapidMiner did not produce any significant results with these algorithms. The program produced trees, but they were not very useful, most likely because it again required numeric values to be converted to nominal values. Some of the different decision trees tried included regular decision trees, CHAID (shown in Fig. 7), and ID3. The average time for running a regular decision tree algorithm in RapidMiner on the GHCN file was 12.5 seconds.

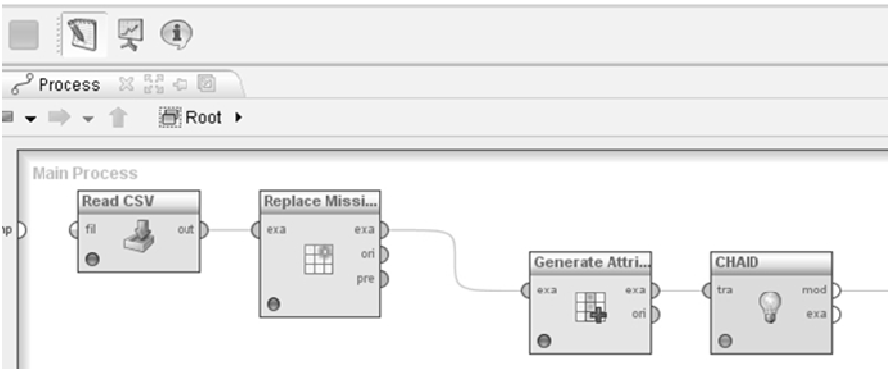

Fig. 7. Connection of functions used with the CHAID decision tree algorithm in RapidMiner

The *Naïve Bayes algorithm* is a classification algorithm used to group temperature ranges. Using RapidMiner's "Generate Attribute" function, we were able to create new data columns that took the minimum and maximum temperature values and grouped them into ranges of very cold, cold, middle, warm, and very warm. The different temperature ranges used were picked by the group. The results are shown in Fig. 8. After running the Naïve Bayes algorithm on this new data, it was determined that the number of very cold days in New Jersey has increased significantly over the past 7 years, jumping from ~2.3% of days in 2005 to ~24.7% of days in 2011. The average time for running the Naïve Bayes algorithm in RapidMiner on the GHCN file was 7.1 seconds.

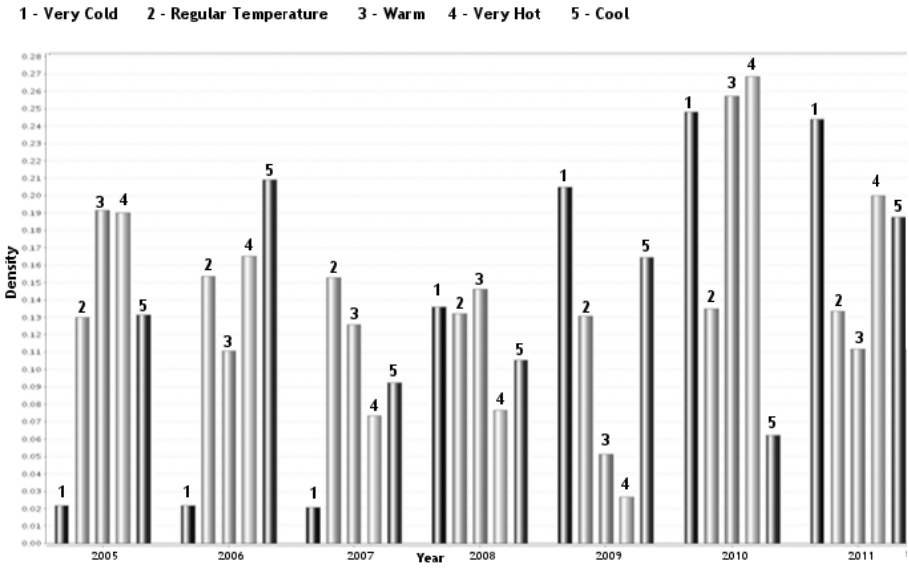

Fig. 8. Graph showing temperature changes in New Jersey over the last 7 years (2005-2012) using data from the GHCN dataset. This graph was generated from the results of the Naïve Bayes algorithm.

7 Visual Inspection of Raw Data

In addition to mining the data using algorithms, raw data was graphed using the program DPlot [12]. Various data values were graphed against each other and against the dates they were recorded. Overall, only one significant pattern resulted from this approach, which was discovered while graphing maximum temperature values for the last 7 years in New Jersey using the GHCN dataset. This graph shows that after every 0.8 or 0.12 Celsius degrees there is a gap of 0.4 Celsius degrees without any recorded values. In addition, the blocks of 0.8 and 0.12 degrees alternate almost continuously along the graph (shown in Fig. 9). This is very odd, and there is no explanation for the missing values.

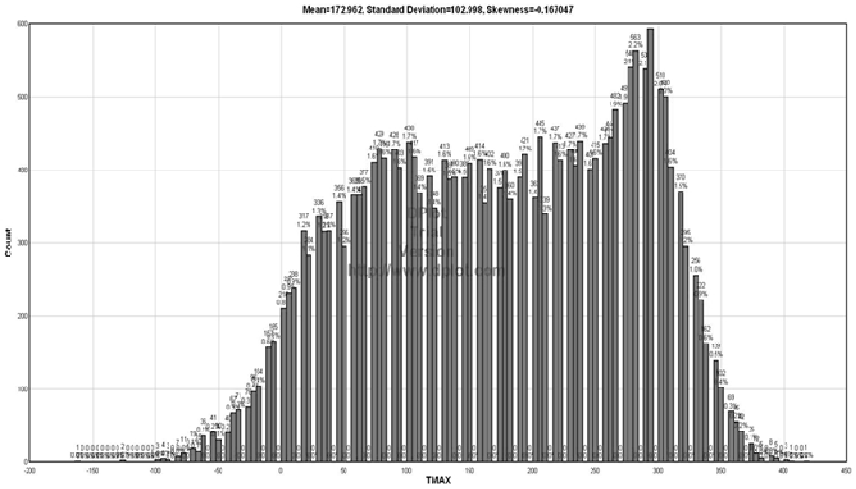

Fig. 9. Histogram showing maximum temperature distributions over the past 7 years in New Jersey (using the GHCN dataset). There is a pattern of missing values shown in the graph.

8 Visualization and Presentation in Context

Overall, the data mining programs did not yield results of great significance on the datasets used thus far. However, the work with the data mining tools yielded more information. There was little new information learned from the algorithms or from the actual results. Most of the association rules were either nonsense due to conversion from numeric to nominal types or very basic rules that are already commonly known (like colder temperatures are seen in the winter). Most decision trees also gave similar results or refused to give anything at all (many times the results were a tree with a single node). Some of this may also be due to the fact that there were only a few commonly collected data types (precipitation, minimum/maximum temperature) in both datasets. To make matters worse, many data types in the GHCN dataset may

have been similar (value of 0) or missing too many values to form accurate associations or predictions. To obtain results of significance or find previously unknown patterns there are two things needed:

1. many more data types that are continuously collected by all weather stations and
2. mining algorithms that support a greater range of numeric data types.

9 Conclusions

Although no new patterns were identified by this project, the greater benefit was learning how to organize data for mining and how to mine large amounts of data. The approach taken here for data mining is valid, however:

1. stronger connections between the data types were needed for pattern identification,
2. the mining programs did not handle the data properly, and
3. a greater range of data types is needed to obtain more significant results.

The methodology outlined here can be applied to a wide range of other fields, as long as a dataset with a large number of continuously collected data types is used. The health industry in particular may benefit from mining patient data to find hidden links between different patients with the same diseases or illnesses. Such medical data is almost continuously being collected in large amounts.

Reorganizing the data for use by mining algorithms is an important part of the process. Writing a program to retrieve the data from the NOAA database and saving it locally in a different format was not difficult, but the conversion process sometimes took a very long time. The bulk of this time was taken to retrieve the files from the database. To reduce such time, researchers can either build the database in the needed format to begin with (not always possible), or convert the data from a file stored on the same machine. The process to request and fetch the data from the database is the most time consuming portion of the large dataset analysis. Storing the converted data in a local file will make sure that the request-and-fetch process only has to be performed once for each set of test data.

As for the different programs used, RapidMiner was far easier to use than Orange and Weka. RapidMiner had its drawbacks with data handling and occasional crashing, but it was simply easier to use thanks to its search feature, its documentation for every function, and the fact that it gave results. Overall, Weka and Orange were lacking as programs, as they were troublesome and not useful.

The three main algorithms that were used (association rules, Naïve Bayes, and decision trees) all took approximately the same amount of time to run against a very large amount of data. To remedy most of the crashes experienced, a computer upgrade to a 64-bit operating system, more RAM, and more/faster processors, is planned. Even without a more capable system, this problem was somewhat overcome by splitting the files into subsets that were mined separately.

Future plans for this research include additional comparative experience with larger datasets, involved data from other areas and case studies from other disciplines.

References

1. Holtz, S., Valle, G., Howard, J., Morreale, P.: Visualization and Pattern Identification in Large Scale Time Series Data. In: IEEE Symposium on Large Scale Data Analysis and Visualization (LDAV 2011), Providence, RI, pp. 17–18 (2011)
2. Morreale, P., Qi, F., Croft, P.: A Green Wireless Sensor Network for Environmental Monitoring and Risk Identification. International Journal on Sensor Networks 10(1/2), 73–82 (2011)
3. Shyu, C., Klaric, M., Scott, G., Mahamaneerat, W.: Knowledge Discovery by Mining Association Rules and Temporal-Spatial Information from Large-Scale Geospatial Image Databases. In: Proceedings of the IEEE International Symposium on Geoscience and Remote Sensing (IGARSS 2006), pp. 17–20 (2006)
4. Zhu, C., Zhang, X., Sun, J., Huang, B.: Algorithm for Mining Sequential Pattern in Time Series Data. In: Proceedings of the IEEE 2009 WRI International Conference on Communications and Mobile Computing, pp. 258–262 (2009)
5. NOAA Integrated Surface Database (GSOD), `http://www.ncdc.noaa.gov/oa/climate/isd/index.php` (retrieved June 12, 2013)
6. NOAA Global Historical Climatology Network (GHCN) Database, `http://www.ncdc.noaa.gov/oa/climate/ghcn-daily/` (retrieved June 12, 2013)
7. Han, J., Rodriguez, J.C., Beheshti, M.: Diabetes Data Analysis and Prediction Model Discovery Using RapidMiner. In: IEEE Proceedings of the 2nd International Conference on Future Generation Communication and Networking (FGCN 2008), pp. 96–99 (2008)
8. Weka's website, `http://www.cs.waikato.ac.nz/ml/weka/` (retrieved June 12, 2013)
9. RapidMiner's website, `http://rapid-i.com/content/view/181/190/` (retrieved June 12, 2013)
10. Orange's website, `http://orange.biolab.si/` (retrieved June 12, 2013)
11. Shafait, F., Reif, M., Kofler, C., Breuel, T.R.: Pattern Recognition Engineering. In: RapidMiner Community Meeting and Conference (RMiner 2010), Dortmund, Germany (2010)
12. DPlot's website, `http://www.dplot.com/` (retrieved June 12, 2013)
13. Thuraisingham, B., Khan, L., Clifton, C., Maurer, J., Ceruti, M.: Dependable Real-time Data Mining. In: Proceedings of the 8th IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC 2005), pp. 158–165 (2005)
14. Martinez, K., Hart, J.K., Ong, R.: Environmental Sensor Networks. IEEE Computer, 50–56 (August 2004)
15. Lewis, F.L.: Wireless Sensor Networks. In: Cooke, D.J., Das, S.K. (eds.) Smart Environments: Technologies, Protocols, and Applications. John Wiley, New York (2004)
16. Zimmerman, A.T., Lynch, J.P.: Data Driven Model Updating using Wireless Sensor Networks. In: Proceedings of the 3rd Annual ANCRiSST Workshop (2006)
17. Chang, N., Guo, D.: Urban Flash Flood Monitoring, Mapping, and Forecasting via a Tailored Sensor Network System. In: Proceedings of the 2006 IEEE International Conference on Networking, Sensing and Control, pp. 757–761 (2006)
18. Cordova-Lopez, L.E., Mason, A., Cullen, J.D., Shaw, A., Al-Shamma'a, A.I.: Online vehicle and atmospheric pollution monitoring using GIA and wireless sensor networks. Journal of Physics: Conference Series 76(1) (2007)

19. Gahegan, M., Wachowicz, M., Harrower, M., Rhyne, T.-M.: The Integration of geographic visualization with knowledge discovery in databases and geocomputation. Cartography and Geographic Information Science 28(1), 29–44 (2001)
20. Arici, T., Akgu, T., Altunbasak, Y.: A Prediction Error-Based Hypothesis Testing Method for Sensor Data Acquisition. ACM Transactions on Sensor Networks 2(4), 529–556 (2006)
21. Monmonier, M.: Geographic brushing: Enhancing exploratory analysis of the scatter plot matrix. Geographical Analysis 21(1), 81–84 (1989)
22. MacEachren, A.M., Polsky, C., Haug, D., Brown, D., Boscoe, F., Beedasy, J., Pickle, L., Marrara, M.: Visualizing spatial relationships among health, environmental, and demographic statistics: interface design issues. In: 18th International Cartographic Conference Stockholm, pp. 880–887 (1997)
23. Monmonier, M.: Strategies for the visualization of geographic time-series data. Cartographica 27(1), 30–45 (1990)
24. Harrower, M.: Visual Benchmarks: Representing Geographic Change with Map Animation. Ph.D. dissertation, Pennsylvania State University (2002)
25. Mueen, A., Keogh, E.: Online Discovery and Maintenance of Time Series Motifs. In: Proceedings of 16th ACM Conference on Knowledge Discovery and Data Mining (KDD 2010), pp. 1089–1098 (2010)
26. Morreale, P., Qi, F., Croft, P., Suleski, R., Sinnicke, B., Kendall, F.: Real-Time Environmental Monitoring and Notification for Public Safety. IEEE Multimedia 17(2), 4–11 (2010)