ORDEN: Outlier Region Detection and Exploration in Sensor Networks

Conny Franke
Department of Computer Science
University of California at Davis, CA, U.S.A.
franke@cs.ucdavis.edu

Michael Gertz
Institute of Computer Science
University of Heidelberg, Germany
gertz@informatik.uni-heidelberg.de

ABSTRACT

Sensor networks play a central role in applications that monitor variables in geographic areas such as the traffic volume on roads or the temperature in the environment. A key feature users are often interested in when employing such systems is the detection of unusual phenomena, that is, anomalous values measured by the sensors. In this demonstration, we present a system, called ORDEN, that allows for the detection and (visual) exploration of outliers and anomalous events in sensor networks in real-time. In particular, the system constructs outlier regions from anomalous sensor measurements to provide for a comprehensive description of the spatial extent of phenomena of interest. With our system, users can interactively explore displayed outlier regions and investigate the heterogeneity within individual regions using different parameter and threshold settings. Using real-world sensor data streams from different application domains, we demonstrate the effectiveness and utility of our system.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*data mining, spatial databases and GIS*; H.5 [**Information Interfaces and Presentation**]: User Interfaces

General Terms

Algorithms, Experimentation, Management

Keywords

outlier detection, sensor data, stream processing

1. INTRODUCTION

Major advancements in sensor technology have led to numerous systems that perform computations over measured sensor values. One of the key functions of such systems is the detection of anomalous events, that is, sensor readings that deviate from the rest of the sensor readings and thus can be considered *outliers*. Such outliers can simply be caused by individual malfunctioning sensors but can also be due to a larger geographic area that exhibits some unusual phenomena or event over time. Especially the latter type of scenario, that is, detecting an area where a collection of sensors measures anomalous values, is of interest to the end user, because the detection of such outlier regions helps to explore time-variant, regional aspects.

Our system for the detection and exploration of outlier regions in sensor data streams, called ORDEN, provides this functionality. ORDEN is realized as a modular system in which one module receives sensor readings from a sensor network in real-time and is responsible for detecting outliers in the stream of sensor data. In contrast to existing, binary approaches, our outlier detection module employs the concept of *degree-based outliers*, i.e., it assigns a degree between 0 and 1 to each measurement depending on the intensity of its "outlierness". Such an outlier degree allows for a more fine-grained and context sensitive analysis of anomalous sensor readings.

The second module then constructs outlier regions from the sensors where anomalous measurements were detected. A novel feature of the region construction is that because of the concept of degree-based outliers, outlier regions can be better delineated from normal regions and are represented as heterogeneous regions, given that different sensors in such a region may exhibit different outlier degrees. The latter aspect leads to a very useful suite of exploration concepts and techniques for outlier regions over time, as we will demonstrate with our system. Important functionality of ORDEN include investigating correlations between individual outliers, the interactive exploration of outlier regions with respect to different outlier degree thresholds, and the demonstration of the context sensitive region boundary placement, which depends on the outlier degrees of nearby sensors. In summary, ORDEN enriches the functionality of today's applications for monitoring in sensor networks and for the detection and exploration of outlier regions in particular.

2. OUTLIER REGION DETECTION

This section outlines the techniques realized in the backend of our demonstration system. First, an outlier detection algorithm is applied to the incoming sensor data stream in order to identify outlier sensors. Then, outlier regions are constructed based on the outlier sensors. A detailed discussion of the underlying algorithms can be found in [4].

Our system uses a degree-based outlier detection technique called *DSTORM*, which assigns an outlier degree $OD \in [0, 1]$ to each incoming sensor measurement. Such OD values are more meaningful, because they take into account that some sensor readings are more clearly outliers than others. $OD = 0$ indicates that a measurement is normal, whereas a high OD value implies that a measurement is highly anomalous compared to other, previously observed

SIGMOD'09, June 29–July 2, 2009, Providence, Rhode Island, USA.
ACM 978-1-60558-551-2/09/06.

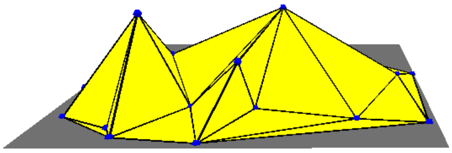


Figure 1: TWS representation of a sensor network

measurements. The current OD values of all sensors in the network are determined based on a sliding window over the previous measurements of all sensors.

Once outliers have been identified, a 2D sensor network can be represented as a three-dimensional surface, as illustrated in Figure 1. This surface is called *Triangulated Wireframe Surface* (TWS) and is generated based on a triangulation of the sensors where the OD values of sensors are added as z-coordinates, i.e., the elevation of sensors in the TWS. In a TWS, outlier regions are shown as "hills". Differences in the elevation of the TWS indicate differences in the outlier degree of sensors that make up an outlier region.

The TWS is employed to construct polygonal outlier regions with respect to an intensity threshold $\varphi \in [0, 1]$. This is done by intersecting the TWS with a plane parallel to the x/y plane at height $z = \varphi$, as illustrated in Figure 2. Thus, the user specified parameter φ is used to select a subset of all detected outlier sensors, i.e., only sensors having $OD \geq \varphi$ are included in the outlier region. The described method places the region boundary in such a way that a measurement taken at a location right next to the boundary would have an OD value close to φ.

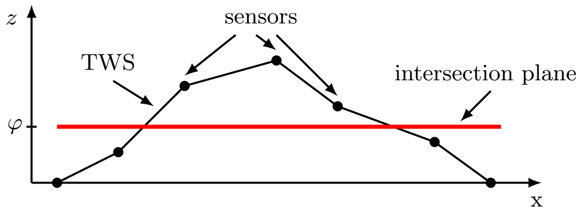


Figure 2: Intersecting a TWS with a plane at height φ to construct an outlier region

3. SYSTEM ARCHITECTURE

In this section, we outline our implementation of the ORDEN system. The architecture of the overall system is depicted in Figure 3. Measurements from a sensor network arrive at the system as a continuous data stream. The system is able to parse various formats of sensor data, which we will describe in Section 4. Parsing is done in the *Input preprocessor*. This component is also responsible for filtering out erroneous values from the data, using the information provided by the operator of the sensor network. Missing measurements in the data are flagged by a predefined value, and our basic data cleansing method replaces such a flagged value by the last valid value observed by that sensor. This approach can be easily replaced by a more sophisticated data cleansing technique according to the specific requirements of the application.

Metadata of the observed sensor networks, like sensor locations, are stored in a PostgreSQL database. Outliers and

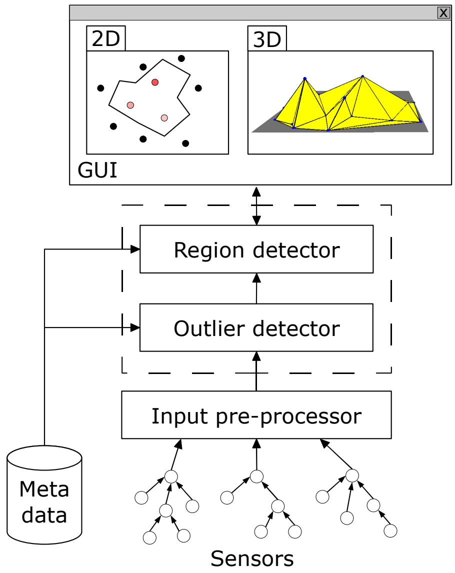


Figure 3: Architecture of the demonstration system

outlier regions are computed in the two main backend components *Outlier detector* and *Region detector*, which both utilize the stored metadata. The region detection component is decoupled from the outlier detector, which allows to use degree-based anomaly detection techniques different from the DSTORM algorithm mentioned in Section 2. Finally, the GUI visualizes the computed outlier regions in real time, either in a 2D sensor network view or an interactive 3D view where the current TWS of the sensor network is depicted. The threshold parameter φ can be adjusted by the user to interactively explore the heterogeneity of the regions.

4. DEMONSTRATION

The objective of our demonstration is twofold: First, we use ORDEN to demonstrate how outlier region detection works with the techniques described in Section 2. Second, we show how ORDEN, in particular its visual interface, can be used to interactively explore outlier regions on-line.

In our demonstration, we use real-world sensor data from three different domains. The first data set, the Intel lab sensor data [2], provides measurements from 54 sensors deployed in the Intel Berkeley Research lab for a period of two months in 2004. The sensors measured three different variables: temperature, illumination, and humidity. As a second real world data source we use CIMIS data [1], containing measurements from about 120 weather stations across California. CIMIS data contain variables such as air and soil temperature, wind speed, and precipitation. The third source of sensor data is TAO, the Tropical Atmosphere Ocean Project [3]. TAO provides measurements from sensors that are attached to about 100 buoys mooring in the Atlantic, Pacific, and Indian Ocean. Measured variables include subsurface temperature, velocity of ocean water, and salinity.

Our region detection algorithm is applied to each of the real-world data sets described above. We show how different parameter settings of the outlier detection algorithm DSTORM influence the detected outliers and their OD values. It will be obvious in the demonstration that DSTORM

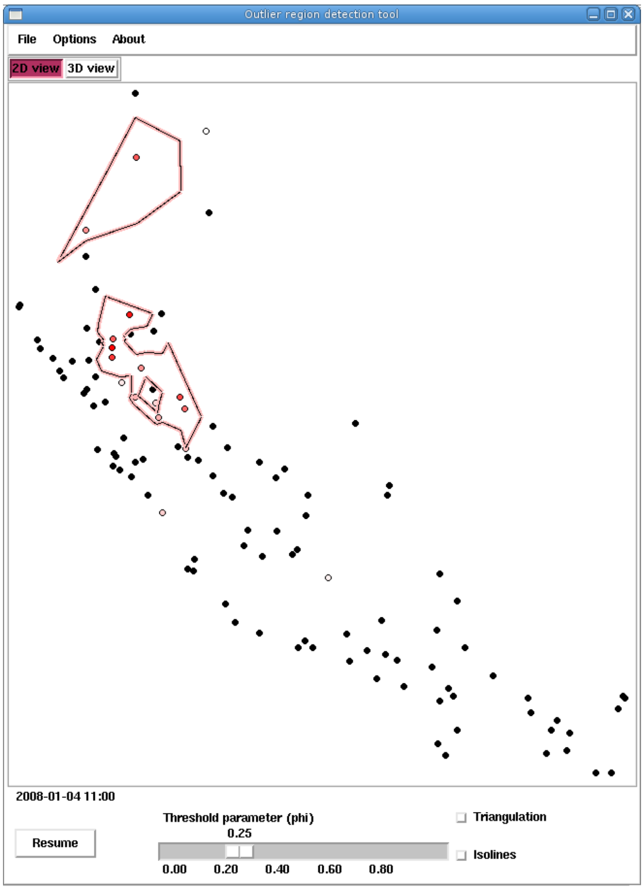


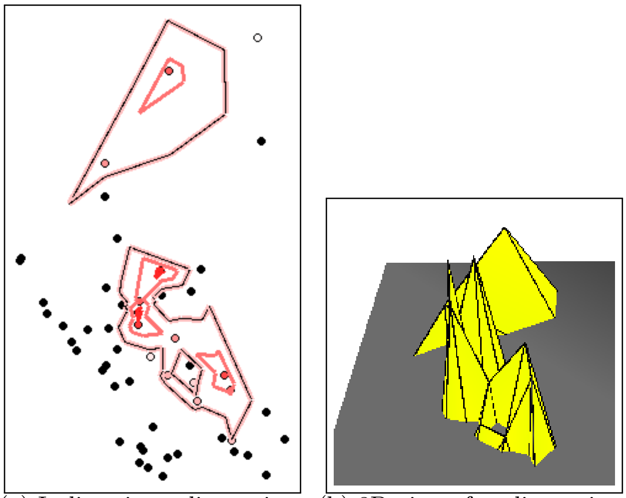
Figure 4: ORDEN system processing CIMIS data

is quite stable, meaning that a wide range of parameter values can be used to detect outliers in the data. Fine-tuned parameters are mostly beneficial for adjusting the distribution of different OD values to the range of outlier sensor measurements.

We will also demonstrate that one can visually distinguish outlying sensor measurements from erroneous values in the data. Outlier values persist in the data set over a period of several measurements and have mostly moderate OD values, whereas erroneous values have an OD value of 1.0 in almost all cases, and are only present for one time unit.

Based on the detected outliers, ORDEN also displays the constructed outlier regions. The triangulation of all sensors can be shown, such that a user can easily see how the region boundary placement was done. The threshold parameter φ can be adjusted dynamically, and subsequently the region boundary is updated to reflect the new value of φ. Figure 4 depicts a screenshot of the ORDEN system while it processes CIMIS wind speed data. In the figure, φ is set to 0.25 and the corresponding outlier regions, which are mostly in the north-west, are outlined by the polygonal boundaries.

Optionally, isolines with adjustable spacing can be displayed in addition to region boundaries. For example, using a spacing of 0.3 and $\varphi = 0.25$, a region boundary will be placed at $\varphi = 0.25$ and additional isolines will be displayed at $\varphi = 0.55$ and $\varphi = 0.85$. This is illustrated in Figure 5(a), which shows the north-west area of the sensor network shown in Figure 4. The adjustable threshold φ in



(a) Isolines in outlier regions (b) 3D view of outlier regions

Figure 5: Outlier region views in ORDEN

combination with isolines is one of the most important features of our tool. They allow for a thorough exploration of outlier regions in sensor data and illustrate the heterogeneity within individual regions. The exploration capabilities of the ORDEN system are augmented by an interactive 3D view, which displays the current TWS representation of the sensor network and the degree plane intersecting the TWS at height φ. Figure 5(b) depicts a 3D view of the TWS that corresponds to the outlier regions shown in the 2D views in Figures 4 and 5(a).

We will also demonstrate the correct boundary placement done by our system. In ORDEN's GUI, we are able to exclude sensors from the computation, such that measurements from these disabled sensors are not considered when computing outlier degrees of sensors. Disabled sensors can therefore be used to verify the correct region boundary placement as described at the end of Section 2, where we stated that sensors that are placed near the region boundary have an OD value close to φ.

In summary, we believe that our demonstration will enable a user to understand the techniques used for outlier region detection and how the results can be explored on-line using the provided parameters.

5. ACKNOWLEDGMENTS

This work is in part supported by the National Science Foundation under Award No. ATM-0619139.

6. REFERENCES

[1] California Irrigation Management Information System (CIMIS). http://wwwcimis.water.ca.gov.

[2] MIT Computer Science and Artificial Intelligence Lab: Intel lab sensor data. http://db.csail.mit.edu/labdata/labdata.html.

[3] Tropical Atmosphere Ocean Project. http://www.pmel.noaa.gov/tao/.

[4] C. Franke and M. Gertz. Detection and Exploration of Outlier Regions in Sensor Data Streams. In *SSTDM '08: Workshop on Spatial and Spatiotemporal Data Mining at IEEE ICDM 2008*, 2008.