Normal vs. Abnormal Behavior

Technical Report

April Jensen and Marina Gavrilova

SPARCS Lab

CPSC Department
University of Calgary

Normal Vs. Abnormal Behavior

The term “outlier” refers to data that deviates significantly from what is considered normal. Outliers are often of interest since they may indicate that data was generated by a mechanism other than the expected one [23], or may indicate abnormal or suspicious behavior. The modeling and analysis of normal vs. abnormal behavior proves useful in security-related areas such as biometrics, video surveillance and intrusion-detection, as well as in other areas such as medical imaging, among others.

There are a number of approaches currently used, such as neural networks, distance-based analysis and clustering methods. Deciding which is the best fit depends on such things as the type of data and how much is known about the normal or expected behavior. Another important consideration is the goals involved, for example whether or not real-time performance is important.

Neural networks can be used to classify data and they use what is called a “single-class classification” system when the goal is just to discern normal behavior from abnormal (as opposed to a “multi-class” system) [5]. In a neural network, a network of processing units work in parallel and a learning machine called a “classifier” makes decisions based on training data and rules (for example a threshold). One drawback is that a lot of training is required to minimize the risk of flagging normal behavior as abnormal [5]. Artificial training data can be used to better train the system [18]. Another potential drawback is that the training data must be assumed to be outlier-free, which may not be possible [5].

There are a number of different possible implementations for neural networks. Computing “single-thresholds” involves training with normal data and choosing a threshold from the data (based on statistics). If a sample exceeds the threshold it is flagged as abnormal. Computing “double-thresholds” involves choosing two such thresholds, and provides a measure of degrees of abnormality, rather than a binary “normal or abnormal” decision. There are a number of other implementations and algorithms that can also be used [5].

The “self-organizing map” is a popular neural-network approach that is often used for spatial proximity problems [5]. Here, a “winning neuron” is the unit whose output is closest to an input vector on some measure. This can then be compared to a threshold to see if it is abnormal [5]. An example of this can be found in [20]. They designed a system meant to detect abnormal pedestrian behavior in surveillance videos. The neural network was trained on normal trajectories, and when presented with input, the “winning neuron” was the one with the smallest Euclidean distance from the input vector. This was then compared to a threshold and flagged as abnormal if it exceeded it. The system could be used to examine trajectories on a point-by-point basis or to examine partial trajectories (which makes real-time performance possible). In practice, they found that while the performance was satisfactory, a lot of training was needed to avoid false alarms [20].

In addition to neural networks, other “training” models may be used. When a normal or expected behavior is known, it can be stored and the inputs then compared to it in order to determine whether or not they deviate. A potential example is described in a white paper for the Oracle 10g database [19]. They suggest that expected routes for ships could be stored and new records inserted in real-time as ships’ location reports come in. Each new record would be compared to the expected route, and a database trigger would flag the behavior as abnormal if it deviated significantly from the expected [19]. Here, an acceptable level of deviation would need to first be determined.

Distance-based approaches are common, particularly when dealing with spatial data since the physical neighborhood plays an important role in its analysis [23]. Spatial data deals with points, lines, polygons and location [1] and the spatial neighborhood can be defined based on attributes like location, distance and adjacency [23]. The approach was proposed by Knorr and Ng (1998, as cited in [21]), and they defined a point p as an outlier if there are no more than k points in the set that are at a distance of d or less from p (where k and d are some chosen values).

To determine if a point is an outlier by this definition requires the computation of the point’s k-nearest-neighbors (k-NN) and their distances (often Euclidean) from the point. There are various ways to do this, for example nested loop algorithms, which check the distance from the point to all other points. More efficient nested loop algorithms (such as the one in [2]) can make use of “pruning”, which detects points that can’t be outliers (due to small distances) and removes them from the test [2]. Another nested loop algorithm [21] computes the distance from each point to its kth nearest neighbor, and the n points with the highest distances are defined as outliers [21].

There are a number of possible proximity queries that may be used. Three types are identified by [3]. "Range queries" attempt to find all elements within a certain distance from a point. The "nearest neighbor" query finds the single closest neighbor to the point while a "k-nearest-neighbor" query finds the k nearest neighbors to that point [3]. In addition, there are a number of algorithms that can be potentially used in nearest-neighbor querying, for example the increasing radius algorithm, backtracking methods, priority backtracking, and other algorithms specifically designed for these types of NN queries [3].

Partitioning can also be used to capture proximity information such as k-NN. For example, Knorr et al. [15] divided a 2D space into square cells which they used to detect outliers. They found that this performed better than a nested loop algorithm [15]. The Delaunay tessellation is the basis for many good proximity searching algorithms (according to [3]) and Voronoi diagrams can capture proximity information even more completely (according to [1]).

In a moving situation, the dynamic Voronoi diagram ([9]) can be used in dynamic nearest-neighbor searching. One model [7] simulates sea-surface navigation, using the kinetic Voronoi diagram as a collision-detection mechanism by updating the nearest-neighbors of the object [7]. A similar approach could perhaps potentially be used in the dynamic detection of abnormal behavior as well. A description of the dynamic Voronoi diagram including its properties and how to construct it can be found in [9]. Another approach to the computation of nearest-neighbors is proposed by [14], who use a tree where each node is associated with a cell in space and a recursive algorithm computes the nearest-neighbors. In this dynamic model, when the cell centers move, the data structure is updated [14].

Feature-extraction can be useful in outlier detection and may involve methods based on edges, lines and curves, template methods, and others [29]. The approach was used in early face recognition methods, which involved computing the distance between important points in 2D [29]. Samples can be used as training data and a test input can then be compared to these to check for a match [18].

An example of this is a system designed for analysis of medical images [25]. Here, images are divided into parts and features are extracted from “regions of interest”. The feature vector is compared to each of its k-nearest-neighbors (in a database of normal images) and each neighbor “votes” on whether the image region is normal or not (this is what is called the "consensus scheme" by [18]). The result is a weighted probability indicating the likelihood that the image is normal [25].

Feature vectors can also potentially be used [13] to identify anomalous shipping routes. In the model proposed in [13], they use Voronoi diagrams to partition the area into regions called “micro-neighborhoods”, each of which has a corresponding feature vector (for example it may contain a field called “airport in area” which may contain a “1” or a “0”). Similarly-behaving micro-neighborhoods are grouped to form “macro-neighborhoods” and their feature vectors are averaged to form a “composite feature vector”. Then, it is possible to look for unexpected associations between a path and areas not on its expected trajectory, which may indicate deviations such as a stop-over [13]. In addition, layers such as "drug zones" can be used to search for associations that can’t be found by the traditional methods of spatial autocorrelation [13].

A dynamic approach using features is described by [16], who worked on modeling human trajectories for surveillance using A.I. methods and statistics. They retrieved features from the trajectory (for example distance traveled, position (X and Y), and acceleration) and trained a SVM (Support Vector Machine - a learning and classification mechanism) to classify them as normal or abnormal based on stored normal trajectories. This model could be used on either the entire trajectory, segments of it, or point-by-point, making real-time use possible [16].

Principle Components Analysis (PCA) is a method that has been used to measure the difference between 2 images, for example for use in face recognition [4]. It involves identifying major directions in the vector space as well as corresponding strengths of variation in the data [4]. In face recognition this has been used to yield a well-known result known as the "eigenface" [4]. In face recognition, if 2 images of the same subject are given, the PCA algorithm will identify areas of change, and if 2 different subjects are given will give a result that appears random [4]. A dynamic PCA approach was suggested by [4], and PCA was also used by [8] to determine "anomalous pixels." They chose a threshold for whether a pixel was anomalous or not, then chose regions of interest (ROIs). Feature vectors were extracted and compared to their expected vector to determine if they passed the threshold or not [8].

Cluster-based methods can be useful as well in outlier detection, for example when analyzing point data. It involves partitioning data into groups so that there is a high degree of similarity between objects within the group and the data is dissimilar to objects outside the group [28]. In spatial data, this often involves a Euclidean distance measure, but data may clustered based on other measures as well, for example in the previous example involving micro- and macro-neighborhoods [13], features were used. An object is an outlier if it does not belong to the cluster.

There are various methods for clustering data. One approach is to link objects by edges if they are within a certain proximity to each other, for example using a Delaunay triangulation [1]. Density-based clustering forms a cluster of points where the density is above a certain threshold; then outliers are points where the data is less dense. Another method of clustering is partitioning. Here, data is divided into a number of groups, and objects are exchanged among the groups until the cluster quality stops improving [28]. In the “k-means” algorithm, for example, the cluster is represented by a mean and the exchanging stops when the average distance of objects from that mean converges to a minimum [28].

A description of how to deal with network based clustering can be found in [28]. They present a storage method for network data, as well as a partition-based algorithm with density-based and hierarchical clustering (which involves the bottom-up merging of clusters [28]). They experimented with it using real road networks, and according to [28] it can use different types of weights on the edges, for example "time to travel," cost or Euclidean distance [28].

An algorithm using nearest-neighbor clustering has been proposed [6], and makes use of Voronoi diagrams and a neural network, meant for when problems must be solved in real-time [6]. The Voronoi cell is constructed for each point; then these are grouped into clusters representing each class of patterns [6]. A neural network is used for classifying points.

Many approaches involve modeling density and rejecting test patterns that fall in regions of low density [18]. Hypothesis testing can also be used to determine whether a given test sample comes from the same distribution as the training data, for example using a t-test [18]. These and a number of other anomaly-detection approaches use statistical methods, such as means, and assume that the data has a normal (Gaussian) distribution (and that the outliers are known if they are present in the data), for example density-based algorithms. They are termed “parametric” approaches. Others, such as the k-nearest-neighbors method, are called “non-parametric” and do not require such statistical assumptions [18]. Non-parametric algorithms, for example those involving a threshold, may require the experimenter to choose the value for the threshold themselves based on what they believe is an acceptable level of deviation and their own definition of what constitutes an outlier in the particular situation.

In addition to the main categories of outlier-detection methods, there are a number of others as well. The OUTLAW system [12], for example, analyzes cargo routes for abnormal behavior by correlating several parameters in different layers and by analyzing spatial proximity. It also uses overlaid maps and rules, and flags a route as abnormal if the number of flags crosses a certain threshold [12]. This method differs from traditional methods which are often deviation-based, distance-based or density-based, according to [12]. The authors draw attention to potential problems with traditional methods. For example, where clustering is used it is possible that the entire cluster itself could be an outlier (which would then not be detected) [12]. In addition, most outlier detection approaches yield a binary result with the object in question being flagged as either an outlier or not, rather than providing degrees of deviation [12].

Shape analysis is another method that is suggested in [26] and has been used in face recognition. It uses shape theory to model activity with a polygonal shape, based on the idea that each object is a moving mass or particle. “Static shape activity” refers to when the shape formed by moving particles stays relatively similar over time, and “dynamic shape activity” is the pattern of global motion over time [26]. It is possible to learn the dynamics of deviations, and then look for temporal or spatial abnormalities by calculating their probabilities (if low then it is likely abnormal). This was used on the example of people getting off of an airplane. Since there is an expected route that they follow while walking toward the airport, the overall shape they will create as they do so will follow a certain expected pattern. If deviations occur (for example someone walks erratically off the path) this can be detected and measured [26].

Another model [27] analyzes “change points” which are the times when significant deviation from the normal occurred, and may be of interest. Their model uses auto-regression statistical methods and it updates parameters to give past examples less weight. Each data/time point is given a score (higher meaning it is more likely to be an outlier) and the model can detect both change points and outliers simultaneously [26]. Data can be modeled using a time series or an independent model. Change points can be computed from the outlier scores over the time series. They tested their approach on stock market data and were able to pinpoint a number of significant change points [26]

The modeling and detection of abnormal behavior has a number of important applications, especially in areas such as safety and security. New and advanced systems and algorithms make it possible to detect and respond to events in real-time as well as to predict and prevent potential incidents from occurring in the future using insight gained from past data.

References

[1] Adam, Nabil R.; Janeja, Vandana Pursnani; Atluri, V. (2004). Neighborhood based detection of anomalies in high dimension spatio-temporal sensor datasets. Proceedings of the 2004 ACM Symposium on Applied Computing. 2004. Pg. 576-583. Nicosia, Cyprus.

[2] Bay, Stephen D.; Schwabacher, Mark (2003). Mining distance-based outliers in near linear time with randomization and a simple pruning rule. Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2003. Washington, D.C. Pg. 29-38.

[3] Chavez, Edgar; Navarro, Gonzalo; Baeza-Yates, Ricardo; Marroquin, Jose Luis (2001). Searching in metric spaces. ACM Computing Surveys (CSUR), Volume 33, Issue 3 (Sept. 2001). Pg. 273-321.

[4] Chen, Tsuhan; Jessie, Hsu Yufeng; Liu, Xiaoming; Zhang, Wende. (2002). Principal component analysis and its variants for biometrics. Proceedings of the 2002 International Conference on Image Processing. Volume 1, 22-25. Sept. 2002. Pg. 1-61-1-64. vol. 1.

[5] Frota, Rewbenio A.; Barreto, Guilherme A.; Mota, Joao C.M. (2004). An evaluation of neural network methods and data preparation strategies for novelty detection. Draft, submitted to IEEE TKDE-Special Issue on Intelligent Data Preparation. Nov. 24, 2004. www.deti.ufc.br/~guilherme/IEEE_TKDE.pdf

[6] Gentile, C.; Sznaier, M. (2001). An improved Voronoi diagram based neural net for pattern classification. IEEE Transactions on Neural Networks, Vol. 12(5), 2001. Pg. 1227-1234.

[7] Gold, C.M.; Chau, M.; Dzieszko, M.; Goralski, R. (2004). The "Marine GIS"- Dynamic GIS in action. Proceedings of the ISPRS 2004 – XXth congress. Istanbul, Turkey.

[8] Goldman, A.; Cohen, I. (2004). Anomaly detection based on an iterative local statistics approach. Proceedings of 2004 23rd IEEE Convention of Electrical and Electronics Engineers in Israel. 6-7 Sept. 2004. Pg. 440-443.

[9] Gowda, Ihor G.; Kirkpatrick, David G.; Lee, Der Tsai; Naamad, Amnon (1983). Dynamic Voronoi diagrams. IEEE Transaction on Information Theory, Vol. IT-29, No. 5, September 1983. Pg. 724-731.

[10] Hu, Weiming; Tan, Tieniu; Wang, Liang; Maybank, Steve (2004). A survey on visual surveillance of object motion and behaviors. IEEE Transactions on Systems, Man and Cybernetics – Part C: Applications and Reviews, Vol. 34, No. 3, August 2004.

[11] Jain, A.K.; Murty, M.N.; Flynn, P.J. (1999). Data clustering: a review. ACM Computing Surveys, Vol. 31, No. 3 (1999). Pg. 264-323.

[12] Janeja, V.P.; Alturi, Vijayalakshmi; Adam, Nabil R. (2002). OUTLAW: Using geo-spatial associations for outlier detection and visual analysis of cargo routes. Proceedings of the Second National Conference on Digital Government (dg. o 2002), May 1, 9-22. 2002. Los Angeles, CA.

[13] Janeja, V.P.; Atluri, Vijayalakshmi; Adam, Nabil R. (2004). Detecting anomalous geo-spatial trajectories through spatial characterization and spatio-semantic associations. The Fifth National Conference on Digital Government (dg. o 2004), Seattle, WA.

[14] Kanungo, Tapas; Mount, David M.; Netanyahu, Nathan S.; Piatko, Christine; Silverman, Ruth; Wu, Angela Y. (1999). Computing nearest neighbors for moving points and applications to clustering. Proceedings of the Tenth Annual ACM-SIAM Symposium on Discrete Algorithms. Baltimore, Maryland, U.S. Pg. 931-932.

[15] Knorr, Edwin M.; Ng, Raymond T.; Tucakov, Vladimir. (2000). Distance-based outliers: algorithms and applications. The VLDB Journal – The International Journal on Very Large Databases. Volume 8, Issue 3-4 (February 2000). Pg. 237-253.

[16] Lee, Ka Keung; Yu, Maoling; Xu, Yangsheng. (2003). Modeling of human walking trajectories for surveillance. Proceedings of the 2003 IEEE/RST International Conference on Intelligent Robots and Systems. Las Vegas, Nevada, October, 2003.

[17] Liu, Ying; Sprague, Alan P.; Lefkowitz, Elliot. (2004). Network flow for outlier detection. Proceedings of the 42nd Annual Southeast Regional Conference. 2004. Huntsville, Alabama. Pg. 402-403.

[18] Markou, Markos; Singh, Sameon. (2003). Novelty detection: a review – part 1: Statistical approaches. Signal Processing. Volume 83, Issue 12 (December 2003). Pg. 2481-2497.

[19] Oracle Database 10g. Developing spatial applications using Oracle Spatial and MapViewer. An Oracle Technical White Paper. February 2004.

[20] Owens, Jonathan; Hunter, Andrew (2000). Application of the self-organizing map to trajectory classification. Proceedings of the Third IEEE International Workshop on Visual Surveillance. 1 July 2000. Pg. 77-83.

[21] Ramaswamy, Sridhar; Rastogi, Rajeev; Shim, Kyuseok. (2000). Efficient algorithms for mining outliers from large data sets. Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data. Dallas, Texas, U.S. Pg. 427-438.

[22] Rao, S.; Sastry, P.S. (2003). Abnormal activity detection in video sequences using learnt probability densities. TENCON2003. Conference on Convergent Technologies for Asia-Pacific Region. Volume 1, 15-17. Oct. 2003. Pg. 369-372.

[23] Sun, Pei; Chawla, S. (2004). On local spatial outliers. Proceedings of the Fourth IEEE International Conference on Data Mining, 2004. ICDM 2004. 1-4 Nov. 2004. Pg. 209-216.

[24] Theisel, H.; Seidel, H.P. (2003). Feature flow fields. Proceedings of the Symposium on Data Visualization 2003. ACM. Grenoble, France. Pg. 141-148.

[25] Van Ginneken, B.; Katsuragawa, S.; terHaar Romeny, R.M.; Kurrio, Doi; Viergever, M.A. (2002). Automatic detection of abnormalities in chest radiographs using local texture analysis. IEEE Transactions on Medical Imaging. Volume 21, Issue 2, Feb. 2002. Pg. 139-149.

[26] Vaswani, N.; Chowdhury, A.R.; Chellappa, R. (2003). Statistical shape theory for activity modeling. Multimedia and Expo, 2003. ICME ’03 Proceedings. 2003. International Conference on, Volume 3, 6-9, July 2003. Pg. 111-181-4 Vol. 3.

[27] Yamanishi, Kenji; Takeuchi, Jun-ichi (2002). A unifying framework for detecting outliers and change points from non-stationary time series data.

Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Pg. 676-681.

[28] Yiu, Man Lung; Mamoulis, Nikos. (2004). Clustering objects on a spatial network. Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data. Pg. 443-454.

[29] Zhao, W.; Chellappa, R.; Phillips, P.J.; Rosenfeld, A. (2003). Face recognition: A literature survey. ACM Computing Surveys, Vol. 35, No. 4, December 2003. Pg. 399-458.