Chapter 13

Applications of Outlier Analysis

> “The study and knowledge of the universe would somehow be lame and defective, were no practical results to follow.”– Marcus Tullius Cicero

13.1 Introduction

Outlier analysis has numerous applications in a wide variety of domains, such as the financial industry, quality control, fault diagnosis, intrusion detection, Web analytics, and medical diagnosis. The applications of outlier analysis are so diverse that it is impossible to exhaustively cover all possibilities in a single chapter. Therefore, the goal of this chapter is to cover many problem domains at a higher level and show how they map to the various techniques discussed in earlier chapters. The practical issues and challenges in the context of real data sets will also be discussed. This will provide a broader understanding of the issues involved in *problem domain to technique mapping.* The main application domains covered in this chapter are as follows:

- Quality control applications
- Financial applications
- Web log analytics
- Intrusion detection applications
- Medical applications
- Text and social media applications
- Earth science applications

In addition, a section will also be devoted to miscellaneous types of data (such as images and trajectories). Within each domain and problem formulation, the diversity of the problems and data types are significant. Therefore, a few “core” formulations will be studied for each

C.C. Aggarwal, *Outlier Analysis*, DOI 10.1007/978-3-319-47578-3_13

application domain, rather than the wide gamut of specific and detailed variations. The formulations will be mapped to broad classes of techniques covered in this book. The goal of this chapter is to teach practitioners how to *use* outlier analysis methods by defining an appropriate mapping from the problem formulation to the specific class of techniques. The specific challenges of different problem domains will also be discussed, with related research being integrated into the discussion of each application.

At this stage, a few practical insights about outlier analysis will be mentioned. These insights seem to be common across many problem domains, and therefore provide an understanding of the areas that would benefit the most from further research:

- *Dependency-oriented data is ubiquitous:* Although traditional multidimensional outlier detection is applicable in many domains, an increasing number of domains generate *dependency-oriented data* such as time series, sequences, spatial data, or network data. Such data is extremely complex and more challenging to analyze as compared to multidimensional data. This is because anomalies are usually defined in a contextual or collective sense in such data. The ability to distinguish between noise and anomalies is limited, because large amounts of data are required in order to obtain a sufficient level of statistical significance about the frequency properties of collective groups of data items.

- *Supervision is often critical in distinguishing between noise and application-specific anomalies:* A significant amount of research has been devoted towards unsupervised outlier analysis in the literature. However, when examining *what really works* in many applications, the presence of supervision is critical. This is because outliers found by unsupervised methods often correspond to noise, and may greatly outnumber the number of interesting anomalies. In such cases, the incorporation of supervision is critical. Even when labels are not available, *indirect* supervision can be incorporated into unsupervised methods by using domain knowledge during feature extraction and specific details of algorithm design. Furthermore, active learning methods can be used in order to efficiently *create* labels by combining unsupervised methods with human feedback.

- *Very simple algorithms work surprisingly well:* Even though a large number of complex algorithms have been defined for outlier detection, many simple algorithms (like the exact k-nearest neighbor method and the Mahalanobis method) seem to work better than many of these complex techniques. Greater complexity in algorithm design is often not adequately rewarded for outlier analysis. In many cases, the complex algorithms may work well for a small subset of benchmarked data sets, which gives a false sense of security about their effectiveness.

- *Ensembles should be used where possible:* Even though simple algorithms work very well, when used on a *standalone* basis, the main advantage of complex algorithms can be gained when they are used in conjunction with ensemble methods. For example, subspace outlier detection is *inherently* ensemble-centric [4, 31] and it makes little sense to use such methods on a standalone basis. In fact, almost all the accurate algorithms for subspace outlier detection, such as feature bagging, rotated bagging, subspace histograms, and isolation forests use ensembles in one form or the other. Furthermore, ensemble methods like variable subsampling and rotated bagging can be combined with virtually any ensemble method because of their generic nature. Detailed experiments and analyses of different ensemble-centric algorithms are presented in [35].

This chapter is organized as follows. Quality control applications are discussed in section 13.2. A discussion on financial applications is provided in section 13.3. Methods for using outlier analysis in Web log analytics are discussed in section 13.4. The problem of outlier analysis in the context of security and intrusion detection applications is studied in section 13.5. Medical applications are studied in section 13.6. Text and social media applications are studied in section 13.7. Earth science applications are presented in section 13.8. A number of miscellaneous applications are studied in section 13.9. A broader discussion of the key guidelines for practitioners is provided in section 13.10. A description of the software resources available for the practitioner is provided in section 13.11. The conclusions are provided in section 13.12.

13.2 Quality Control and Fault Detection Applications

Quality control applications arise often in the context of manufacturing. Outliers can be detected in such applications either in terms of the characteristics of individual objects, or in terms of the aggregate characteristics of the manufacturing process. Some examples of applications in such domains are discussed below.

Application 13.2.1 (Quality Control) *A manufacturing process is designed to produce widgets, which are defective with probability p_i. A specific batch of widgets of size n contains q defective widgets. The goal is to estimate the probability that the manufacturing process is behaving in an anomalous way.*

Discussion: This is one of the standard formulations for quality-control analysis. In many variations of this problem, specific parameters of the widget (e.g., physical parameters) may be tracked and compared to an expected mean and standard deviation. In some cases, multiple parameters may be tracked simultaneously. Depending upon the specific parameters being tracked, a variety of extreme value analysis methods from Chapter 2 can be used.

- The Chernoff bounds discussed in section 2.2 of Chapter 2 can be used to provide tight bounds on the tail probabilities when specific fractions are being tracked (e.g., fraction of defective widgets).

- The Hoeffding inequality discussed in section 2.2 of Chapter 2 can be used to provide tight bounds when strict upper and lower limits exist on tracked values.

- The t-value and normal value distributions discussed in section 2.2 of Chapter 2 can be used in order to provide approximations for the level of significance of an aggregate value (e.g., fraction of defective widgets in a sample).

- In cases, where multiple parameters are tracked (eg. a combination of length, width, weight), the multivariate distance distribution methods discussed in section 2.3.4 of Chapter 2 may be used, either on the individual parameters (dimensions), random subsets of parameters (dimensions), or the entire set of parameters (dimensions). In many instances, the anomalous behavior may be reflected simultaneously in many dimensions, as a result of which the use of multivariate analysis may provide better insights.

■

While aggregate analysis is an important application in quality control, numerous scenarios require the examination of a specific object for faults. This is related to the problem of fault diagnosis.

Application 13.2.2 (Fault Detection and Systems Diagnosis) *A running engine or industrial system is continuously being monitored on a variety of parameters such as rotor speed, temperature, pressure, performance, and so on. It is desired to detect a fault in the engine system as soon as it occurs.*

Discussion: The data in this domain are usually in the form of sensor data, in which continuous sensor values are tracked over time. Thus, methods for time-series analysis may be used in this scenario. However, the specific methods being used may depend upon the specific application at hand.

- In many applications, extreme values of the sensor data may correspond to anomalies. For example, very high temperature or pressure may precede the bursting of a pipe. In such cases, even the simple extreme-value analysis methods of Chapter 2 may be used without accounting for the temporal aspect of the data. However, in order to perform early detection, sudden and unusual changes are more relevant. For example, a sudden and unusual *rise* in temperature may be relevant, even when the *absolute* value of the temperature is not high. In such cases, the abrupt change detection methods discussed in section 9.2 of Chapter 9 may be used.

- In many applications, thousands of time-series may be monitored, and unusual deviations in *specific* combinations of time-series provide information about *different types of* anomalous behavior. Section 9.2 of Chapter 9 discusses the case in which a sensor failure scenario can be distinguished from a pipe rupture scenario with the use of supervision. Such scenarios almost always require supervision in order to provide the learning necessary for specific kinds of anomalies.

- Unusual *shapes* in the time-series may often provide clues about anomalous behavior. For example, an unusual vibration in the engine system may cause oscillations in the pressure values, which are abnormal. In such cases, the unusual time-series shape detection methods of section 9.3 in Chapter 9 can be used. These methods can also be generalized to multivariate time-series by transformation to trajectory data, as discussed in section 9.3.1.3 of Chapter 9.

- Unusual shapes can also be related to *specific* diagnosis of system faults by using supervised methods. In such cases, training data relating similar faults with the corresponding time series may be available. Supervised methods for unusual shape detection in time series are discussed in section 9.3.5 of Chapter 9.

The work in [241] designs a method for novelty detection, such as the detection of shorted turns in the field-windings of operating synchronous turbine-generators. Signature patterns of signals are extracted from the running motor, and are compared with the normal signals in order to detect novelties. A probability density estimation method for detection of abnormal conditions in engineering is discussed in [167]. A kernel method is used for the density estimation process. Kernel density-estimation methods are discussed briefly in Chapter 4 of this book. The method in [428] uses principal component analysis to transform the data, and then identifies the anomalies as the points that lie far away from the primary hyperplanes of projection.

Many of the above methods can also be applied to problems such as structural damage detection, in which faults in mechanical units may be diagnosed with the use of different kinds of time-series data. PCA methods have also been used for anomaly detection in spacecraft components [206]. A method that combines unsupervised and supervised learning methods for fault detection in automobile data is discussed in [207]. A detailed discussion of the use of the wavelet transformation for machine-health monitoring is provided in [432]. The use of neural networks for motor fault detection is discussed in [146]. A supervised method for motor-bearing damage detection was proposed in [482]. In many cases, streaming and online detection methods are desirable [9].

■

The diagnosis of computer systems also requires real-time anomaly detection techniques. However, the data in such cases is often discrete, and the methods used are typically similar to those in intrusion detection and security applications. This will be discussed in a later section of this chapter. For example, methods for system monitoring of large computer clusters are discussed in [470].

A related topic is that of structural defect detection, which attempts to determine structural defects in 2-dimensional or 3-dimensional objects such as a fabric, or a beam [123, 526, 527]. In such cases, measurements may be associated with each spatial location. For example, for the case of fabric fault detection, a 2-dimensional image of the fabric may be analyzed for faults [123].

Application 13.2.3 (Structural Defect Detection) *Given a 2- or 3-dimensional surface associated with measurements at each spatial location over time, the goal is to determine significant structural defects, either at a given point in time, or significant changes that occur over time.*

Discussion: The nature of the measurements in this case is specific to the problem domain. This problem is inherently spatiotemporal, since multiple spatial measurements are available at different instants in time. Different methods are possible for defining outliers.

- Unusual (spatial) changes in the attribute values on the basis of spatial locations can be used in order to detect anomalies. For example, the neighborhood algorithms discussed in section 11.2.1 of Chapter 11 can be used in order to detect outliers.

- Unusual shapes in the images implied by the attribute values often provide insights about significant patterns of defects in the data. These methods are discussed in section 11.2.4 of Chapter 11.

- The spatial analysis in the two cases can be combined with temporal analysis in order to determine significant changes in the underlying data. This corresponds to the methods discussed in section 11.3 of Chapter 11.

- In many cases, previous examples of specific defects may be available. In such cases, the supervised techniques discussed in Chapter 11 may be used.

Methods for structural defect detection are discussed in [123, 274, 526, 527].

■

A broad review of fault detection methods is provided in [554].

13.3 Financial Applications

Financial fraud is one of the more common applications of outlier analysis. Such outliers may arise in the context of credit card fraud, insurance transactions, and insider trading. This section will discuss a number of financial applications in the context of outlier analysis.

Application 13.3.1 (Credit-Card Fraud) *A credit-card company maintains the data corresponding to card transactions by different users. Each transaction contains a set of attributes, such as the user identifier, amount spent, geographical location, and so on. The card company may also have labeled data containing previous examples of fraudulent transactions. It is desirable to determine fraudulent transactions from the data.*

Discussion: One desirable aspect of credit-card applications is that labeled data is often available in order to relate the transactions with the underlying anomalies. Nevertheless, both supervised and unsupervised methods can be used for anomaly detection in such cases.

Many domain-specific characteristics of the data are used for fraud detection. For example, it is well known that transaction amounts consisting of large absolute values may correspond to anomalies. The most common technique is to build user profiles on short segments of transaction sequences. Typically, the ordering among a short segment of the transactions is immaterial. If desired, a single transaction of the user can also be used. Either a single transaction or a short sequence of transactions can be converted into a feature vector, which is compared to the user's profile. The key is to design a similarity function, which can encode the wide diversity of attribute types, the collective profile within a short segment, and domain-specific knowledge (e.g., large transactions or sudden bursts of high-value transactions are more likely to be fraudulent). It is also possible to use geographical location in order to determine the anomalousness of a sequence of transactions with respect to other sequences from the same spatial location.

The major challenge with anomaly detection in credit card data is that false positives are extremely common, and false negatives are expensive, even when rare. In other words, the receiver operating characteristic (ROC) curve usually suggests very noisy behavior of purely unsupervised methods. The quality of the inference can be improved in two ways, both of which incorporate some form of supervision:

- Domain-specific knowledge needs to be encoded into the similarity function in order to account for the differential nature of fraudulent transactions.

- When labels are available, supervision should be used in order to relate the profiles to fraudulent behavior.

In many cases, the automated analysis is combined with manual inspection in order to determine significant cases of fraud. Recently, discrete-sequence methods of Chapter 10 based on hidden Markov models have also been used [509]. In these methods, symbolic values are extracted by discretizing the credit card amounts. A survey on supervised fraud detection methods is provided in [440]. The issue of class imbalance in supervised fraud detection methods is discussed in [441]. A wide variety of methods [45, 124, 218, 440, 513, 514] are available for credit-card fraud detection, although the general experience has been that supervised methods are the most effective. This is not surprising, since supervised methods are better able to distinguish between true anomalies and noise.

■

Application 13.3.2 (Insurance Claim Fraud) *In this case, claims are made by different entities on the basis of insurance policies. Significant anomalies need to be discovered from the data on this basis.*

Discussion: Although this application shares some resemblance to credit-card fraud detection at a higher level, it is also significantly different in many ways. For example, user-specific profiles cannot be constructed, because a particular user may rarely make a claim. On the other hand, repeated claims by a single user is often an indicator of fraud, and should be incorporated as a feature during pre-processing. Unlike credit fraud applications, geographical location is often *contextually* not relevant.

Once a multidimensional representation of the claims has been created, the problem is essentially an application of multidimensional (point) anomaly detection. The key step is to extract the correct features from the insurance claim documents, which can be used in order to create an unsupervised or supervised anomaly detection system. Feature extraction in insurance claim scenarios is highly domain specific, since it requires the identification of indicators that are highly specific to the particular type of claim. For example, in a life-insurance scenario, a low lag between the initiation of the policy and the death of the subject is sometimes correlated with homicide. In a medical insurance claim scenario, the statistical distribution of the claims over different types of diseases coming from a single medical provider may be skewed when the provider is engaging in fraud. Depending upon the application, such features need to be extracted in a domain-specific way. Therefore, any of the methods in Chapters 2, 3, 4, 5, 6, and 7 may be used once the feature extraction phase has been performed.

Labels are usually available since previous examples of fraud are available. In order to obtain high quality prediction, it is critical to encode the information about previous examples, either in the form of the learning algorithms discussed in Chapter 7, or indirectly by using feature representations that distinguish fraudulent claims from normal ones. Methods for insurance fraud detection are discussed in [166, 440, 556]. In particular, a comprehensive bibliography may be found in [440].

■

Many financial organizations also track the user behavior at their Web sites. These correspond to discrete sequences that can be analyzed in order to determine significant instances of fraud. These cases will be analyzed in section 13.4 of this chapter on Web log analytics.

Application 13.3.3 (Stock Market Anomalies) *The financial tickers of the different stocks and options correspond to time-series data streams. In some cases, significant anomalies may be created by external events. The* early *detection of such events may be useful in the determination of* unknown *influencing factors such as insider trading, or automated stock trading glitches (e.g., the flash crash of May 2010).*

Discussion: In many cases, external information such as news streams (e.g., *Google News*) are also available for event detection. This is particularly useful for applications such as insider trading detection, where unusual temporal ordering between events in the news and events in the stock tickers can be used in order detect insider trading. For example, if an anomalous change (in value or transaction volume) of the relevant stock ticker precedes an event for the ticker in the news stream, then this can be used as an indicator of insider trading.

In other cases, such as the unusual behavior of the stock market during the flash crash of May 6, 2010, direct time-series analysis may be used. Such methods are discussed in Chapter 9. Both deviation-based contextual point anomalies and time-series shape-based collective anomalies provide insights about the unusual interactions. Deviation-based anomalies are more useful for early detection, whereas shape-based anomalies are more useful for detailed diagnosis, which is slightly delayed. In many cases, it may be desirable to use streaming methods [9] in order to determine the anomalies in real time. Methods for early detection of insider trading in financial markets are discussed in [172]. Some of the methods for multidimensional change detection discussed in Chapter 9 are useful for tracking other aspects of stock activity, such as specific stock orders or the volume distribution of stock orders. Methods for distribution change detection in stock order data streams are discussed in [372].

Another interesting method for detecting *regime anomalies* [76] from time-series data streams can be used in order to detect sudden changes in the dependencies among different streams. This provides an idea of scenarios in which the relationships among the different stocks have changed over time.

■

Financial entities often interact with one another. Examples include producers and suppliers, customers with sellers, and customers with each other. In such cases, anomalous patterns of interaction can provide useful insights. This leads to the interesting problem of detecting anomalies in financial and customer interaction networks.

Application 13.3.4 (Financial Interaction Networks) *A set of financial entities V are continuously interacting with one another over time. Values on the edges may correspond to the intensity or volume of the interactions. Unusual anomalies may need to be determined in such cases.*

Discussion: Financial interaction networks are ubiquitous, and the interactions between different financial participants are often tracked in order to obtain competitive knowledge about the interactions. In some cases, such as mobile phone fraud, the interactions may not specifically correspond to financial transactions, but rather to an interaction between two customers. In most of these cases, values on the edges are available for analytical purposes. The temporal change detection methods in section 12.5 of Chapter 12 can be used in order to identify the relevant regions of change in the network. The challenge in using such methods is that the values on the edges are often critical to the anomaly detection process. For example, high values of the transactions may correspond to anomalies. It is relatively easy to generalize the spectral methods of Chapter 12 to include the values on the edges in the analysis. Such methods are discussed in [280, 519, 520, 522]. Different methods for fraud detection in the context of mobile phone networks are discussed in [275, 418, 440, 441, 536].

■

13.4 Web Log Analytics

Web logs often contain significant information about security breaches and other kinds of anomalous activity. For example, a bank may keep logs of its Web site accesses. Unusual patterns of accesses may correspond to anomalous activity.

Application 13.4.1 (Web Log Anomalies) *Given a sequence of accesses in a Web log, the goal is to determine the unusual patterns of accesses in this log.*

Discussion: Web logs are typically pre-processed into a set of user-specific discrete sequences. These discrete sequences correspond to the identifiers of the pages accessed by the users. Many challenges arise during pre-processing, since users can often be distinguished only at the level of their IP addresses. Nevertheless, even in such cases, user-sessions can often be mined from the logs. The initial phase of pre-processing is crucial in such applications, because a single undifferentiated log is provided. This log needs to be decomposed into user-sessions, and then further decomposed into test sequences, and comparison units, as discussed in Chapter 10. General issues related to Web log data preparation are discussed in [150].

In many cases, additional domain knowledge is available about *relevant* sequences. For example, a repeated sequence of accesses to *login* and *password* pages may be indicative of anomalous behavior. In other cases, examples of anomalous discrete sequences may be available. Where possible, such domain knowledge should always be used, because it significantly improves the quality of the underlying results.

Numerous methods discussed in Chapter 10 are applicable to this case, depending on the types of outliers that need to be found.

- Position outliers can be used to determine unexpected accesses. These are contextual anomalies detected by the method, and correspond to a single unpredictable access, which is an outlier because of its *relationship* to adjacent and neighboring values. Markovian and rule-based models are typically used for outlier detection in discrete sequences. Such methods are useful for early anomaly detection, when a single unexpected web access is sufficient to arouse suspicion.

- Combination outliers are useful for determining unusual subsequences in the test sequence. This can be achieved using either unsupervised or supervised methods. In the case of supervised methods, the extraction of relevant features such as k-grams is crucial for effective anomaly detection. In the case of unsupervised techniques, window-based nearest neighbor and hidden Markov models are typically used.

Methods for anomaly detection in web logs are discussed in [169, 329].

■

Web logs are often analyzed in the context of a wide variety of security and intrusion detection algorithms. These methods will be discussed in this section. In the former case, operating system call traces are analyzed on a particular computer system, whereas in the latter case, the network data is analyzed for anomalies.

13.5 Intrusion and Security Applications

Intrusions correspond to different kinds of malicious security violations in a computer system. The data is typically streaming, and arrives at a high rate. Intrusions correspond to anomalous events, which need to be inferred from the underlying data. Denning [165] classified intrusions into *host-based intrusion detection systems* and *network-based intrusion detection systems.* These cases are somewhat different both from the perspective of data representation and temporal locality. In general, the former involves the analysis of discrete

Table 13.1: Some examples of user commands in a UNIX system

Time	Hostname	Command	Arg1	Arg2
AM	Sayani-T61	mkdir	dir1	
AM	Guardian	more	file1	
AM	Guardian	cp	file1	file2
AM	Sayani-T61	cd	dir1	
AM	Guardian	find	dir2	-print
AM	Sayani-T61	vi	file1	

sequences with high temporal correlations, whereas the latter involves the analysis of multidimensional streams with (relatively) limited temporal correlations. Therefore, different models are typically used in these cases. Each of these cases will be discussed in this section.

Application 13.5.1 (Host-based Intrusions) *Operating-system call traces are available in a computer system, which are symbolic sequences. Anomalous subsequences in these traces correspond to malicious computer programs. It is desired to determine anomalous sequences from these traces.*

Discussion: The data in this case are similar to Web logs at a conceptual level, in that it corresponds to symbolic sequences. In this case, operating-system call traces are used instead of web log traces. The calls could correspond to either operating system calls or user calls. Thus, the calls form the base alphabet Σ over which the mining is performed. Different kinds of programs execute different sequential combinations of calls. Therefore, the sequential ordering of the calls provides critical information in order to distinguish between normal and malicious programs. These calls could either be at the user-command level, or they could be at the operating system level. The latter is much more granular, and that can sometimes make it more difficult to mine such sequences. Some examples of user-level calls in an operating system are illustrated in Table 13.1.

During the feature-extraction phase, the logs are transformed into symbolic sequences. In many cases, when the commands are coming from multiple sources, they may need to be separated out into their different hostnames. For example, in the case of Table 13.1, the sequences of commands from the hostnames *Guardian* and *Sayani-T61* need to be separated out into different sequences in order to examine the malicious behavior of a particular host. This is similar to Web log analytics, in which sequences that are specific to each Web user are constructed in the pre-processing phase. Sometimes, the choice of the symbols used in the sequence may also depend upon the application at hand. For example, should only the user command be used, or should some combination of user command and argument be used? These choices are highly application-specific, and the quality of the final results may be sensitive to such choices.

Subsequently, any of the discrete sequence methods discussed in Chapter 10 can be used. In fact, the different kinds of scenarios are very similar to those in Application 13.4.1 on Web log analytics. The reader may refer to the details of the specific methodologies used in Application 13.4.1. In spite of the very different data domains in these cases, it is interesting to see that the underlying methods are often interchangeable.

A comparison of different methods for intrusion detection in these scenarios is provided in [158]. Numerous methods have been proposed in the literature for this scenario, and are discussed in [158, 188, 197, 198, 199, 200, 201, 216, 217, 272, 338, 339, 340, 349, 350, 351].

■

A second class of methods is that of network-intrusion detection systems, in which the intrusions are inferred from network data. The data on the network can be of different types, depending upon the level of abstraction at which it is presented. For example, the data could correspond to the underlying packets on the network, and the intrusions may result in subtle changes in the multidimensional features extracted from these packets.

Application 13.5.2 (Network Intrusion Detection) *Given a stream of network packets or data records, the goal is to determine network intrusions.*

Discussion: The temporal relationships between data records are much weaker in this case than in the case of host-based systems, in which the sequential ordering of calls is critical in identifying intrusions. Furthermore, each individual record in this case is multidimensional, and contains the features extracted from the unit of network data (e.g., packet), or raw *tcpdump* data. For example, a common feature used is the number of bytes transferred, which is continuous. Other attributes are discrete. Thus, this problem can be modeled as a multidimensional stream of records, containing both continuous and categorical attributes. An example of a network-intrusion data set is illustrated in Table 13.2. Only five of the basic features are shown. This is the well known *KDD Cup 1999 Intrusion detection data set* [203], which contains a combination of symbolic and continuous attributes. These features are of three types that correspond to the basic characteristics of the connections (service, protocol, bytes transferred etc.), the content characteristics of the connections suggested by domain knowledge (e.g., number of "hot" indicators, failed login attempts), and the traffic characteristics (e.g., number of connections, or the number of connections with specific kinds of errors).

Most of the unsupervised multidimensional outlier detection methods can be generalized to this case. Furthermore, in cases where stream processing is required, the multidimensional streaming outlier detection methods of Chapter 9 may be used. In particular, aggregate change-points may often be helpful in identifying network-wide traffic anomalies. Such change points often correspond to network intrusions and attacks [377, 334, 335]. Since the data is often of a mixed nature, many of these algorithms need to be modified using the general methodologies discussed in Chapter 8 for adapting unsupervised algorithms to mixed data sets.

In some cases, a subset of the data may be labeled, and may correspond either to normal data or to intrusions. The labeled intrusions can never be exhaustive, since new intrusions may always arise over time. Nevertheless, labeled data about existing intrusions is useful for identifying repetitive attacks. At the same time, it is important to also detect novel classes as new intrusions arise. Such scenarios can be addressed using the streaming and supervised novel-class detection methods discussed in section 9.4.3 of Chapter 9. Methods for network-intrusion detection are discussed in [55, 56, 70, 71, 72, 145, 188, 288, 352, 330, 331, 381, 382, 420, 484, 544, 589]. A comparative study of network-intrusion detection schemes may be found in [345].

■

Table 13.2: Examples of five basic features from the network connection records from the *KDD Cup 1999 Network Intrusion Data Set* [203]

Duration	**Protocol**	**Service**	**Src_Bytes**	**Dest_Bytes**
5	*tcp*	*telnet*	183	3855
5	*tcp*	*telnet*	183	3855
0	*tcp*	*http*	298	2239
0	*udp*	*private*	105	146
0	*udp*	*domain_u*	44	44
0	*tcp*	*http*	188	2199
0	*icmp*	*ecr_i*	508	0

13.6 Medical Applications

Medical applications typically use different types of diagnostic tools for predictive modeling. Two of the most common kinds of data that are encountered for predictive modeling of medical data are in the form of sensor data (e.g., electrocardiography), and spatial data (e.g., positron emission tomography). Both of these are different types of contextual data. Each of these different cases will be addressed by a different application definition.

Application 13.6.1 (Medical Sensor Diagnostics) *Given a set of sensor readings from a given patient, the goal is to determine if the patient has a disease condition.*

Discussion: Both supervised and unsupervised methods can be used in order to process medical data. For the unsupervised case, the problem formulation is similar to that of fault diagnosis, except that the nature of the sensor readings are specific to the medical domain. Therefore, all the methods from Chapter 9 can be used for this case as well. In the simplest case, extreme or unexpected value analysis on medical time-series or data distributions may be used [276, 343, 463, 502] in order to determine anomalous values. An approach that uses a probabilistic mixture model is discussed in [528]. Time-series containing subsequences of unusual shapes [360] may also be useful for identifying more complex anomalous conditions.

In the context of medical data, the cost of missing a positive is high, and the diagnosis needs to be specific. Furthermore, any anomalous conditions may need to be reported in real time. In many cases, a specific diagnosis may be distinguished from another potentially incorrect diagnosis by using the signals from multiple sensor data streams. In this context, a supervised method for deviation-based anomaly detection in multivariate time-series data streams was proposed in [9]. Methods for shape-based supervised anomaly detection are discussed in Chapter 9.

Supervised methods are particularly desirable in the medical domain because of the specificity requirements of a diagnosis. In many cases, expert knowledge [50] may need to be combined with the mining algorithm in order to ensure the most effective results. Semi-supervised methods for medical time-series classification are discussed in [564]. The problem of supervised shape discovery in time-series data is discussed in section 9.3.5 of Chapter 9.

■

Another common diagnostic tool used in the medical domain is that of *imaging*. In these cases, an magnetic resonance imaging (MRI) or positron emission tomography (PET) scan is used to create a 2-dimensional or 3-dimensional image of a part of the body such as the brain. Anomalies in these shapes correspond with significant medical conditions.

Application 13.6.2 (Medical Imaging Diagnostics) *Given a multidimensional image of an affected body part, the goal is to determine whether a patient has a disease condition, and what that condition is.*

Discussion: This is a classic example of a spatial application that contains both contextual and behavioral attributes. The discovery of anomalies in such data has been addressed extensively in Chapter 11. The most crucial part of the feature extraction is to convert the shape [602] into a time series using the methods discussed in section 11.2.4 of Chapter 11. Subsequently, a variety of unsupervised or supervised methods can be applied to the extracted time series. The problem typically reduces to one of the following formulations:

- Anomalous shapes may correspond to specific disease conditions such as tumors, multiple sclerosis lesions, mammography, or the degraded brain regions of an Alzheimer patient [448, 553, 505, 537]. In the semi-supervised case, examples of normal shapes may be available, and it may be desirable to determine shapes that are very different from these normal profiles. The method of [602] may be used to convert the shapes into a time series, and then anomaly detection can be applied to this time series. For example, these anomalies may be identified using the approach given in [565]. This method has also been described in section 11.2.4 of Chapter 11.

- In many cases, previous examples of anomalous regions may be available. These examples can be used to train a classifier to learn the relationship between the shape and the specific disease condition. A variety of shape learning methods [69, 115, 375, 602] are available for labeled data. A brief review of such techniques is provided in section 11.2.4.4 of Chapter 11.

- In some cases, a temporal sequence of images from the same patient over many years may be available, and it may be desirable to determine *changes* in the shapes of the images [86]. This corresponds to shape-change detection methods discussed in section 11.2.4.5 of Chapter 11.

A variety of shape analysis methods have been used frequently in the medical domain for image diagnostics [86, 251, 448, 553, 477, 505, 537, 543].

■

13.7 Text and Social Media Applications

Text and social media applications are extremely common because of the ubiquity of text data in social interactions such as email, the Web, and blogs. Some of these methods have already been discussed in Chapter 8.

Application 13.7.1 (Event Detection in Text and Social Media) *Given a document corpus $\mathcal{D}$, the goal in unusual topic detection is to determine unusual documents that differ significantly from the trend. In first story detection, a stream of documents is available, and it is desirable to determine unusual events corresponding to new topics in the stream of documents. In the context of social media applications, such streams may be generated by user activities such as tweets.*

Discussion: This scenario has already been discussed extensively in Chapter 8. Both supervised [584, 586] and unsupervised methods [29, 47, 48, 49, 95, 306, 507, 508, 585, 608, 622] may be used. The reader is referred to Chapter 8 for details.

A particularly important special case is that of social media streams in which the user activities such as tweets may provide early knowledge of unusual events. In many cases, unusual events in localized regions may show up in social media feeds well before traditional news media, because of the wider authorship of social media sites. Such events will typically be manifested as changes in the topical and linkage distributions of the social media feeds. Both supervised and unsupervised techniques are relevant in such cases, depending upon the availability of training data. In the supervised case, it may be desirable to determine events of specific types. Methods for finding unusual events or changes in text and social streams are discussed in [30, 435, 562]. Both supervised and unsupervised methods for event detection in social media streams are discussed in [30].

■

Another common application in email streams is to detect spam in a sequence of emails.

Application 13.7.2 (Spam Email) *Given a stream of emails, the goal is to identify the subset of emails that correspond to spam.*

Discussion: While unsupervised methods for unusual topic detection can be used, the results are often likely to be inaccurate. While spam is still a small fraction of the mail in most cases, the volume is large enough to make it difficult to detect with unsupervised methods. This case is best addressed with the use of supervised methods, where the specific features of the emails are learned, and related to spam labels in the training data. Any of a variety of methods for text classification [24] can be used.

A lot of additional domain knowledge is available, which helps determine whether a particular email message is junk or not. For example, an email would be more or less likely to be junk or spam according to some of the following characteristics:

- The domain of the sender can make an email to be more or less likely to be junk. For example, an email address ending in *.edu* is less likely to be junk.
- Phrases such as *"Free Money"* or overemphasized punctuation such as "!!!" can make an email more likely to be junk.
- Whether the recipient of the message is a particular user or part of a larger mailing list can influence the underlying possibility that the email is junk or spam.

The Bayes classifier for text provides a natural way to incorporate such additional information into the classification process by creating new features for each of these characteristics. A method such as this has been discussed in [469]. A survey of methods for email spam filtering may be found in [85, 152].

■

In many social networks, a significant percentage of the links are noisy, and may not provide any useful insights for analysis. Such links are often caused by spam links on the Web to increase search-engine rating, or by low-quality links across weakly related entities.

Application 13.7.3 (Noisy and Spam Links) *Given a social network with content at the nodes, determine the noisy and spam links with the use of structure and content information.*

Discussion: This problem is discussed extensively in the sections 12.3.2 and 12.4 of Chapter 12. The first section discusses methods for determining linkage outliers on the basis of structure only, whereas the second section discusses methods for determining linkage outliers with the use of both structure and content. Such edges in a social network correspond to relationships that are weak, and often harm the effectiveness of social media algorithms. Techniques for determining such outliers are discussed in [17, 196, 212, 452, 214]. In many cases, supervised methods may be used in order to learn link spam, when labeled information is available.

■

Many types of networks (blogs, bibliographic networks, social networks) contain both texts and link information. Discovering significant patterns of change that constitute anomalous activity in these types of networks may be helpful.

Application 13.7.4 (Anomalous Activity in Social Networks) *Given an evolving network with associated text content at the nodes, the goal is to determine the anomalous regions of activity or change in the network.*

Discussion: This problem is related to that of evolution of communities in the underlying network. It is possible to use a purely structural approach with the use of the community change analysis methods discussed in section 12.5 in Chapter 12. In the case of blogs and social networks, a significant amount of content is also available in the network. In such cases, community detection algorithms can be enhanced with the use of node content [12]. In addition, a variety of spectral methods [280, 519, 520, 522] can be used, when it is required to use only the linkage structure for analysis. A method that specifically monitors evolving blogs for significant change is discussed in [415].

Social networks are particularly useful tools for the discovery of threat activity such as terrorist interactions. A method for finding threat activity in social networks with the use of eigenspace analysis was proposed in [398].

■

13.8 Earth Science Applications

Outlier detection is used in numerous weather, climate, or vegetation cover applications, where anomalous regions are detected in spatial data either at a single snapshot or over time. Therefore, many of these applications are spatial or spatiotemporal in nature. For example, sea surface temperatures are often tracked in order to determine significant and anomalous weather patterns.

Application 13.8.1 (Sea Surface Temperature Anomalies) *The temperatures on the sea surface are tracked continuously over time. It is desired to determine:*

(a) unusual localized spatial variations in temperature on the sea surface,

(b) regions on the sea surface containing unusual shapes with homogeneous temperature,

(c) sudden and unexpected changes in sea surface temperature that are localized to a specific region, and

(d) the relationship of the spatial temperature patterns to known weather events for predictive modeling.

Discussion: This is a typical spatial or spatiotemporal formulation that arises often in the context of meteorological applications. In this case, the spatial and temporal coordinates are the contextual attributes, whereas the temperature is the behavioral attribute. It requires the determination of both contextual and collective outliers from the data. The determination of unusual spatial variations can be performed by finding neighborhood-based outliers. Such outliers are discussed in section 11.2.1 of Chapter 11.

Unusual shapes on the sea-surface temperature profile can be performed by applying the unusual shape discovery method discussed in section 11.2.4 of Chapter 11. The temperatures may need to be discretized into buckets in order to convert the continuous values into discrete shape contours. The contour of a region with the same discretized value may be used for the anomaly-detection process.

The neighborhood-based spatiotemporal change detection algorithms of section 11.3 in Chapter 11 may be used in order to determine significant outliers. These correspond to specific *points* in space and time at which there are unusual temporal *or* spatial variations. In some cases, it may be desirable to determine significant changes in the *shapes* of temperature patterns. The detection of significant changes in the patterns can be performed by using the shape-change detection methods discussed in section 11.2.4.5 of Chapter 11.

Such characteristic patterns in different kinds of behavioral attributes such as temperature, pressure, or humidity can often be related to unusual weather events such as cyclones. However, such anomalies are best determined with the use of supervised methods in which previous training data relating the weather patterns to the spatiotemporal data is available. Supervised methods for classification of such data are discussed in detail in [69, 115, 375, 602].

■

While the aforementioned application was presented with the use of temperature as a behavioral attribute, a variety of other attributes such as pressure or humidity may be tracked for anomaly detection. As an example, the work in [569] tracks precipitation patterns in order to determine unusual regions of change. Methods for finding region outliers in meteorological data are discussed in [615]. In many cases, multiple behavioral attributes may be available. This is a more challenging case, because it is desired to determine unusual combinations of behavioral attributes. In such cases, a simple approach is to perform the analysis separately on each attribute and combine the anomaly scores. The real-time spatiotemporal analysis of weather patterns can provide predictions of significant events such as hurricanes.

A closely related problem is that of *land-cover anomalies.* The type of land cover or vegetation is often tracked with the use of remote sensing. Virtually all the aforementioned methods for finding meteorological anomalies can also be applied to the problem of land-cover anomalies.

Application 13.8.2 (Land Cover Anomalies) *The land cover at different spatial locations are tracked continuously over time. It is desired to determine:*

(a) unusual localized spatial variations in land cover type,

(b) regions containing unusual shapes with homogeneous land cover,

(c) sudden and unexpected changes in land cover that are local to a specific region, and

(d) the use of changes in spatial land-cover patterns to uncover unusual and unknown geological, climate, human or wildlife activity.

Discussion: This case is virtually identical to that of uncovering unusual sea-surface temperature anomalies. The major difference is that the land-cover type is the behavioral attribute, which may be discrete. Therefore, the key is to design a similarity function which relates different land cover types to one another. For example, certain types of land cover are more likely to be adjacently located than others. In order to determine such similarity values, the methods [154] for contextual similarity discussed in section 8.4.2 of Chapter 8 may be used. Once such contextual similarity measures have been determined, they can be used in order to determine significant point changes in the data. An example of a recent application to determine significant vegetation changes is discussed in [341]. Methods that correlate land cover changes with other kinds of parameters (such as climate) are discussed in [65, 464].

■

13.9 Miscellaneous Applications

This section briefly covers miscellaneous applications for outlier detection that do not belong to any of the aforementioned categories.

Application 13.9.1 (Data Cleaning) *Given a data set, remove discordants from it. Correct any errors in the data if possible.*

Discussion: This is one of the classical applications of outlier analysis, and is often addressed effectively with unsupervised methods. Virtually all the unsupervised methods discussed in this book can be used for noise *removal.* Many of the autoregressive models introduced in Chapter 9 are used for removal of erroneous values from sensor data. For example, the removal of noisy links in social networks can be considered a data cleaning application. Such methods are discussed in Chapter 12.

For noise *correction*, methods such as PCA can provide the best insights. For example, it has been shown in [21, 425] that the use of PCA and SVD methods can be used in order to improve the representation quality of data sets for mining and retrieval. Methods for removing outliers in the context of regression analysis are discussed in [467].

■

A number of applications are designed to determine anomalies in spatiotemporal data. Such data may correspond to movement patterns for wildlife or vehicular traffic.

Application 13.9.2 (Traffic and Movement Patterns) *Given a set of entities with trajectory patterns, determine significant outliers from the patterns.*

Discussion: This problem arises often in the context of either tracking wildlife with RFID tags, or tracking vehicles with GPS receivers. This problem can be addressed using either spatiotemporal trajectory-mining methods or by network-mining methods depending on how the data is represented. For example, when it is desirable to determine anomalous *entities* in terms of movement patterns, the trajectory mining methods [347] discussed in section 11.4 of Chapter 11 may be used. In some cases, supervision may be used [355] in order to improve the quality of the discovered patterns. Online algorithms for finding outliers in movement patterns are proposed in [102].

In many cases such as traffic data, the movement values are available on an aggregated basis because of privacy concerns. Furthermore, the movement patterns are often associated with a network corresponding to the road network in the underlying data. In such cases, only the flow values on the different road segments may be available. Significant regions of congestion or anomalous behavior may need to be identified. Methods for finding such anomalies are discussed in [400].

■

One of the most common domains for anomaly detection is image analytics. This was discussed earlier in the context of medical image diagnostics.

Application 13.9.3 (Image Data) *Given a set of (possibly labeled) images, or a temporal sequences of almost identical images, it is desirable to determine:*

(a) images with anomalous shapes;

(b) changes in the underlying patterns in cases where temporal snapshots are unavailable; and

(c) prediction of a rare class in cases when some images are labeled with this class using training data.

Discussion: The techniques used for this case are similar to those discussed earlier in this chapter in the context of the medical image diagnostics application. Generic image representations also have a number of attributes such as color or texture, which can be leveraged in order to determine anomalies. As discussed earlier, image diagnosis methods are used frequently in the medical domain for determination of anomalies. In the context of anomalous shape discovery, the feature extraction for *shape representation* is the most important. An example was discussed in section 11.2.4 of Chapter 11, where it was shown how to convert a shape into a time-series for further analysis. An extensive discussion of such feature transformation methods is provided in [602]. General references on image outlier detection may be found in [86, 251, 448, 553, 477, 505, 537, 543].

■

Numerous other applications of outlier analysis may be found in the domains of aviation safety [156], internet routing updates [447], malicious URL detection [380, 613], disease outbreaks [568], spatial linking of criminal incidents [370], failure management of large computer clusters [470], astronomical data [177, 261, 577], disturbance events in terrestrial ecosystems [444] and biological sequences [365].

13.10 Guidelines for the Practitioner

The examples discussed in this chapter illustrate that the diversity in specific problem setting is rather large across different domains. However, many of these models map into the same set of problems. Some critical observations that arise in the context of outlier detection are as follows:

- *Data normalization is important:* A common mistake made by many practitioners is to forget to normalize the data before applying outlier analysis algorithms. Consider an application containing an *Age* attribute (less than a hundred), and a *Salary* attribute (in order of tens of thousands). The use of proximity-based or linear models on such data (without normalization) will be dominated by the *Salary* attribute, and the *Age* attribute will be almost ignored. Typically, each attribute value needs to be divided by its standard deviation (over the entire data set). This ensures that the different attributes are given an equal level of importance in the outlier analysis process. This process is also referred to as *standardization*, in which each attribute is first mean-centered by subtracting the mean. Then, the values in each attribute are scaled to unit variance.

- *Noise versus interesting anomalies:* Most application domains contain noisy or incomplete data or data that has errors. For example, sensor data often contains noise because of defects in transmission or failure in the data-collection hardware. As discussed in this chapter, data cleaning is itself a key application of outlier analysis. Therefore, it is critical to design a pre-processing phase that can filter out or correct such noise from the data when possible. This can be achieved using a variety of domain-specific methods, which have knowledge of the noise generation process. For example, in the context of sensor data, such noise can often be corrected or filtered by a variety of methods [22]. Nevertheless, in many cases, the use of such filtering methods can also mask interesting anomalies in the data.

- *The feature extraction phase is crucial:* In many domains, the base data is not necessarily specified in a way that can be used directly with an outlier analysis application. For example, in an insurance application, the documents containing details of the claims may be available. In a credit-card fraud application, raw transaction data may be available. In such cases, feature extraction *should implicitly use domain knowledge* to the extent possible. For example, when it is known that large transactions are more important, a feature corresponding to transaction size should be incorporated. Although numerous dedicated feature selection methods exist for other data mining problems such as clustering and classification, this does not seem to be the case for outlier analysis. This is because feature-selection methods require the determination of aggregate trends relating the features to application-specific aspects of the data. On the other hand, since outliers are based on rare observations, aggregate trends are hard to determine in a way that would be relevant for outlier analysis. Therefore, the incorporation of a domain-specific understanding is often the only way to extract meaningful features for outlier analysis. A proper selection of features can also help distinguish noise from true anomalies.

- *Domain knowledge is often easy to incorporate into unsupervised algorithms:* One of the challenges of outlier analysis is that labeled data is rarely available. However, an *indirect* form of supervision is to incorporate domain knowledge into unsupervised algorithms. Such changes require minor modifications to the details of the underlying algorithm such as the similarity function design, or the design of the transition architecture of a hidden Markov model. Some examples of incorporating domain-specific knowledge are as follows.

  - In a credit-card fraud application, the absolute amount spent is known to be a key indicator of fraudulent behavior. A k-nearest neighbor algorithm can be modified, so as to treat neighbors with higher or lower values of the amount spent in a purchase in a differential way. This knowledge can be incorporated directly into the distance function, without making any other change to the algorithm.

  - In a security monitoring application, specific sequences such as *login password login password* may be known to be indicative of attacks. Such sequences can be provided higher importance during the construction of the comparison units for sequence anomaly detection.

  - The states of a hidden Markov model should reflect an understanding of the process, rather than using a black-box k-state model, in which all transition probabilities are learned.

- *Labeled data should be used where possible:* This is the easiest way to distinguish between noise and anomalies. *Even a small amount of labeled data* can *significantly* improve the effectiveness of outlier analysis algorithms. Unfortunately, in many real applications, labeled data is not available. In fact, the unavailability of labeled data is the raison d'etre for unsupervised outlier analysis.

- *Exploratory and visual analysis can be helpful at all stages of outlier analysis:* One challenge in outlier analysis is that it is often difficult to know which model may work most effectively for a given problem. Should a proximity-based model be used, should a linear model be used, or should a subspace model be used? In this context, visual analysis of the kind introduced at the beginning of Chapter 3 can provide some insights about the distribution of the data. This can often provide an idea of the specific choice of model that might work best for a particular application.

- *A human in the loop can more easily generate labels in conjunction with unsupervised outlier analysis algorithms:* It is hard to generate labeled data in outlier analysis algorithms, because anomalies are rare, and therefore positive examples are often hard to obtain. Furthermore, the manual examination of large amounts of data for anomalies is akin to searching for a needle in a haystack. However, unsupervised and supervised algorithms can be used in an iterative way in conjunction with a human in the loop in order to generate labels. This corresponds to the active learning framework discussed in Chapter 7. Such frameworks can be extremely useful in converting unsupervised outlier detection problems into supervised rare-class detection problems.

- *Outlier ensembles can be used to reduce risk:* The choice of a specific algorithm is a "risk" to the practitioner because the performance of a particular algorithm on a data set cannot be known even after the fact for an unsupervised problem like outlier detection. In such cases, ensembles can be used to combine different models and also reduce the variance of a specific algorithm on a particular data set. By combining a small number of models that are known to work well across a wide variety of data sets (see next section), it is possible to consistently obtain results of good quality. Combining these techniques with methods like subsampling and rotated bagging can further improve the result quality. From this point of view, the ideas in Chapter 6 are very useful to leverage repeatedly across various data domains.

The aforementioned recommendations seem to suggest that the incorporation of supervision and domain knowledge is possible, even when fully labeled data are not available. A careful domain-specific design of the feature selection and algorithmic processes is critical in obtaining the most informative outliers.

13.10.1 Which Unsupervised Algorithms Work Best?

It is impossible to identify the optimal algorithm in the absence of ground truth. Since the outlier detection problem is unsupervised, the effectiveness of a particular algorithm on a data set depends on the "blind luck" of how well the model of normal data reflects the true distribution. Nevertheless, it has been shown [32, 35] that some simple algorithms tend to work well across a broad range of real-world data sets. In fact, testing with real-world data sets seems to show that more complex algorithms do not always yield the performance gains that one would expect from their sophistication. The experiments in [32, 35] seem to suggest that algorithms such as the exact k-nearest neighbor method, the average k-nearest neighbor algorithm, the nonlinear Mahalanobis method, and isolation forests seem to perform extremely well. Although there is some distinction between the base performances of these algorithms, they tend to become more similar when they are combined with a method like variable subsampling. In the case of isolation forests and the nonlinear Mahalanobis method, the use of an ensemble-centric implementation is essential.

It is also noteworthy that the linear Mahalanobis method and (the basic version of the) isolation tree algorithm are virtually parameter-free; this makes their applicability particularly simple. Although the raw distance-based algorithms are not parameter-free, their application in conjunction with variable subsampling (Chapter 6) ameliorates this problem to some extent. This is because the variability in the size of the subsample often eliminates the unpredictable effect of a particular value of the parameter k in the k-nearest neighbor algorithm.

An important point needs to be kept in mind about the relative evaluation of algorithms that depend on parameters. In some cases, a particular algorithm might perform better than another algorithm at the *best* value of the parameter, but it might not perform as well across a reasonable range of parameters. This is particularly true when the algorithm is *unstable* across the choice of parameters, and therefore performs well only at specific values. In such cases, it is particularly important to use a more stable algorithm that shows better performance across a wider range of parameters. For example, as shown in [32], the well-known LOF algorithm [96] does not seem to perform as well as a simple average k-nearest neighbor algorithm, when tested *across a wider range of parameters.* Although the performance at the best value of k for LOF might be better than that at the best value of k for the raw distance-based methods, it is hard for the analyst to know the correct values of k for unsupervised settings. In this sense, the raw distance-based detectors are often superior to LOF in real-world settings because they provide stable and high-quality solutions in a *large range of values* of k. It is also shown [221] that raw distance-based methods tend to be more stable than LOF across the choice of *data sets.* If the data set predominantly contains global outliers, then LOF might give too many false positives.

A detailed evaluation of some of the key outlier detection algorithms is provided in [35]. In the following, we discuss an overview of three key algorithms that seem to be perform well in many settings, along with their primary advantages:

1. **Linear and Nonlinear Mahalanobis method:** The Mahalanobis method is discussed in section 3.3.1 of Chapter 3. The Mahalanobis distance reports the outlier score of a data point as its Mahalanobis distance from the centroid of the data. The approach combines the benefits of multivariate extreme-value analysis and principal component analysis. Furthermore, the approach can be combined with kernel representations for complex data types or nonlinear distributions; in such cases, the approach tends to outperform many kernel models like one-class SVMs and support-vector data descriptions (SVDD). The nonlinear Mahalanobis method tends to work well in many distributions where conventional methods work poorly. This is because of its ability to adapt to the shapes of various data distributions. The approach can be implemented efficiently as an ensemble-centric method [35] on samples of data with size between 50

and 1000 points. The kernel bandwidth is set at thrice the median pairwise distances between points, which is estimated by sampling. An ensemble-centric implementation of the nonlinear Mahalanobis method has been proposed in [35] and has been shown to achieve excellent results. Although the linear Mahalanobis method can be used efficiently as a standalone method, it is crucial to use the nonlinear method only in an ensemble-centric setting. The nonlinear Mahalanobis method generally tends to be more robust and accurate than the linear Mahalanobis method in such settings.

2. **Raw distance-based algorithms:** The exact and average k-nearest neighbor algorithms are described in section 4.3.1 of Chapter 4. These methods are very simple to implement and often outperform methods like LOF, when tested across a wider range of parameters. However, it is possible for LOF to outperform these methods if the parameters are carefully chosen. Another issue is that raw distance-based algorithms tend to be more robust across a wider range of *data sets*, whether the outliers are global or local [221]. In comparison, algorithms like LOF tend to find too many false positives on data sets containing global outliers. As discussed in section 1.7.2 of Chapter 1, it is more important for unsupervised methods to work well across a wider range of parameter settings and data sets because of the inability to perform model selection and parameter tuning in the absence of ground truth. The merits of raw distance-based methods are discussed in [32, 35].

   There are, however, two primary drawbacks with the entire category of neighborhood-based methods. The first drawback is that such methods have quadratic complexity with the number of points. The second drawback is that these methods can be sensitive to the choice of the parameter k. Both issues can be partially solved with the use of the variable subsampling method discussed in Chapter 6. As shown in [32], variable subsampling blunts the specific effect of parameter choice at least to some extent. Furthermore, by combining variable subsampling with rotated bagging, one can incorporate subspace outlier detection principles to improve both accuracy and efficiency.

3. **Subspace histogram ensembles:** Many subspace histogram methods such as *RS-Hash* [476] (see section 5.2.5) and the isolation forest [367] provide fast and accurate density estimation in subspaces of the data. This can be helpful when many dimensions are locally irrelevant. The ensemble-centric nature of these methods make them robust and accurate. Although the isolation forest does not seem to be a density-estimator at first sight, its scores are intimately related to histogram-based density estimation and are qualitatively similar (see section 5.2.6.3 of Chapter 5). One drawback of such methods is that they tend to be biased against isolated points in the interior regions of the data set. For example, a single outlier at the center of normal data points on a unit sphere in 10 dimensions can receive an inlier-like score with the isolation forest [35]. Furthermore, all histogram-based methods can be challenged by small clusters of anomalies in the data.

In general, one should always wrap an ensemble method (like variable subsampling or rotated bagging) around a base algorithm, unless the algorithm is itself implemented inherently as an ensemble (like *RS-Hash* or isolation forests). As shown in [35], such ensembles have great *equalizing* power; in other words, they can improve base methods (with widely varying quality) to very similar performance that is near-optimal. Furthermore, one can combine the results from these types of robust ensembles to gain further advantages. As a first implementation, we recommend to use a simple ensemble average of the (standardized) scores of algorithms from each of the three types discussed above, which are themselves implemented as ensembles. Such a combination is referred to as *TRINITY* in [35], which combines the nonlinear Mahalanobis method, the k-nearest neighbor detector, and the isolation forest. Each of these base algorithms is used in conjunction with variable subsampling. While it is recognized that more sophisticated algorithms might do better on specific types of data sets, the use of this simple approach can be a reasonably robust option when one does not have a deeper semantic understanding of the underlying data set. After all, lack of visibility and blindness to data characteristics are the hallmarks of all unsupervised settings.

13.11 Resources for the Practitioner

Since the problem of outlier analysis is one of the key problems in data mining, a significant number of software resources exist for this problem. The software available for this problem is both commercial and open-source. Note that the outlier analysis problem is addressed using both supervised and unsupervised methods. A significantly larger number of packages exist for the supervised version of the problem since it is directly related to the problem of classification, which is a much broader field. In this section, the key resources from these two perspectives will be summarized.

A significant number of open source packages exist in the literature for different variations of the problem. A meta-repository containing a description of the different resources for both unsupervised and supervised learning is the *KDD Nuggets* Website [626, 627]. The *Weka* repository [628] is a large general purpose repository, containing different kinds of data mining software for clustering, classification and outlier analysis. In addition, the *ELKI* repository [633] for outlier analysis contains an implementation of many of the advanced algorithms discussed in this book such as *LOF* and its variations, *LOCI*, EM-methods, distance-based methods, and subspace methods. Numerous methods for spatial outlier detection are contained in the same repository. A Python library for outlier detection may be found with *scikit learn* [632]. Packages for outlier detection in R can be found in the *RDataMining* library [634]. Resources for different components of supervised and unsupervised outlier analysis are available from the UCR time-series classification and clustering page [635], and the Symbolic Aggregate Approximation [636] page.

A significant amount of commercial software is also available for outlier analysis. An example is the *IBM Security Network Intrusion Prevention System* [637] for network intrusion detection. The *IBM SPSS Workbench* [638] has numerous tools that can be used for outlier detection in both temporal and non-temporal data. In particular, *IBM SPSS Statistics* [639] contains a significant number of tools which can be used to build models for outlier analysis. The *Oracle Data Miner* [642] has significant data mining capabilities including anomaly detection. *WizSoft* software [643] has designed a software *WizRule*, which can be used for fraud and anomaly detection. *SAS* [640] has developed many different software packages for general statistical modeling and anomaly detection. A particularly relevant one is the *SAS Security Intelligence* [641] software, which is designed to address fraud, compliance and security issues. All of the above can handle both supervised and unsupervised scenarios. Furthermore, for the supervised case, a significant amount of classification software is available both on an open source and commercial basis. An exhaustive list of the different kinds of open source and commercial software may be found in [627]. A number of anomaly detection implementations in Python are available on *scikit-learn* [632].

13.12 Conclusions and Summary

This chapter provides an overview of the applications of outlier analysis. These applications are distributed across a wide variety of domains; nevertheless, many of these domains map to similar formulations for modeling purposes. The goal of this chapter was to provide the practitioner an understanding of the methods used in different domains. The adaptation of various generic outlier analysis methods to specific domains was also discussed.

Outlier analysis brings numerous challenges with it in different application domains, because of the difficulty in distinguishing between noise and anomalies. Typically, the incorporation of human feedback, domain knowledge, ensemble methods, and explicit supervision can address many of these challenges. In cases in which significant visibility does not exist on the effectiveness of various algorithms, it is helpful to try an ensemble of some simple algorithms such as the Mahalanobis method, the k-nearest neighbor algorithm, and subspace histograms.