Outlier Detection

Sharanjit Kaur
University of Delhi, India

INTRODUCTION

Knowledge discovery in databases (KDD) is a non-trivial process of detecting valid, novel, potentially useful and ultimately understandable patterns in data (Fayyad, Piatetsky-Shapiro, Smyth & Uthurusamy, 1996). In general KDD tasks can be classified into four categories i) Dependency detection, ii) Class identification, iii) Class description and iv) Outlier detection. The first three categories of tasks correspond to patterns that apply to many objects while the task (iv) focuses on a small fraction of data objects often called outliers (Han & Kamber, 2006). Typically, outliers are data points which deviate more than user expectation from the majority of points in a dataset.

There are two types of outliers: i) data points/objects with abnormally large errors and ii) data points/objects with normal errors but at far distance from its neighboring points (Maimon & Rokach, 2005). The former type may be the outcome of malfunctioning of data generator or due to errors while recording data, whereas latter is due to genuine data variation reflecting an unexpected trend in data. Outliers may be present in real life datasets because of several reasons including errors in capturing, storage and communication of data. Since outliers often interfere and obstruct the data mining process, they are considered to be nuisance.

In several commercial and scientific applications, a small set of objects representing some rare or unexpected events is often more interesting than the larger ones. Example applications in commercial domain include credit-card fraud detection, criminal activities in e-commerce, pharmaceutical research etc.. In scientific domain, unknown astronomical objects, unexpected values of vital parameters in patient analysis etc. manifest as exceptions in observed data. Outliers are required to be reported immediately to take appropriate action in applications like network intrusion, weather prediction etc., whereas in other applications like astronomy, further investigation of outliers may lead to discovery of new celestial objects. Thus exception/outlier handling is an important task in KDD and often leads to a more meaningful discovery (Breunig, Kriegel, Raymond & Sander, 2000).

In this article different approaches for outlier detection in static datasets are presented.

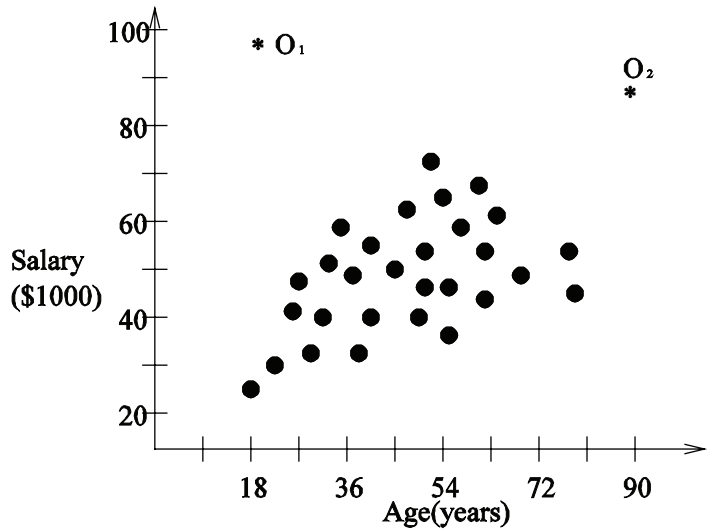
BACKGROUND

Outliers are data points which deviate *much* from the majority of points in a dataset. Figure 1 shows two outliers (O_1 and O_2) in Employee dataset with two attributes age and salary. Points O_1 and O_2 represent employees drawing high salary with age 18 and 90 respectively. These points are considered outliers because i) there is no other point in their neighborhood and ii) they are substantially different from the rest of points. Further exploration of such points may reveal some interesting facts.

Although exact definition of an outlier is application and context dependent, two commonly used general definitions for outliers are as follows. The classical definition is given by Hawkins (Hawkins, 1980) according to which, an outlier is an observation that deviates so much from other observations so as to arouse suspicion that it was generated by a different mechanism. A more recent definition, given by Johnson (Johnson, 1992), defines outlier as an observation which appears to be inconsistent with the remainder of the dataset.

Outlier detection in statistics has been studied extensively both for univariate and multivariate data, where a point not falling inside the distribution model is treated as outlier. Most of the approaches used in statistics are either suitable for univariate data or data with known distribution model e.g. Normal, Poisson etc. (Maimon & Rokach, 2005). In univariate approaches, only one feature is used, whereas in multivariate approaches multiple features are used to distinguish outliers from the normal objects. Multivariate approaches, which

Figure 1. Distinction between normal data and outliers



are computationally more expensive than univariate approaches, yield efficient results for low dimensional numeric dataset with known distribution model. However, for real life multidimensional datasets with unknown distribution, expensive tests for model fitting need to be performed. Data mining approaches do not assume any distribution model and overcome some of these limitations.

DATA MINING APPROACHES FOR OUTLIER DETECTION

Outlier detection is an important area in data mining to reveal interesting, unusual patterns in both static and dynamic datasets. In this article, we focus on non-parametric approaches for outlier detection in static datasets. These approaches detect outliers without any prior knowledge about underlying dataset and view dataset S as consisting of two components.

$$S = P + O \quad (1)$$

Here P represents normal data points and O represents exceptions or outliers. These approaches are categorized as: i) Clustering-based approach, ii) Distance-based approach and iii) Density-based approach, on the basis of their most distinct feature (Breunig, Kriegel, Raymond & Sander, 2000; Papadimitriou, Kitawaga, Gibbons & Faloutsos, 2003) .

Clustering-Based Approach

Clustering is a database segmentation technique which partitions a dataset into unknown groups as per prevalent data characteristics (Han & Kamber, 2006). Grouping is done such that data points in one group are more similar to each other than those belonging to different groups. These groups are called clusters. The data points which are not member of any cluster (Figure 2) are reported as outliers (Maimon & Rokach, 2005).

Clustering algorithms for large datasets like Balanced Iterative Reducing and Clustering using Hierarchy (BIRCH), Density Based Spatial Clustering of Applications with Noise (DBSCAN) and Clustering Using Representatives (CURE) report outliers as a by-product of the clustering process (Dunham, 2004). The objective of these algorithms, however, is to optimize clustering and not outlier detection.

BIRCH maintains compressed information required for clustering in *Cluster Feature Tree* (CF-Tree), which is a balanced tree. Each leaf in CF-Tree represents a set of points with a specified diameter and is treated as a pseudo point during clustering. The algorithm removes a leaf if the number of points is less than a user-defined threshold (λ) and reports all points as outliers.

DBSCAN uses a formal notion of density reachability to discover arbitrary shaped clusters with user defined minimum neighborhood δ and density m. *Density* of a point p is defined as a minimum number of points within a certain distance from p. A point is