Chapter 1 - An Introduction to Outlier Analysis

1 Introduction

An outlier is a point that is so different from the other points that we suspect it is created by a different generating process. The output of outlier analysis is a score or binary label for each point indicating whether the point is an outlier or inlier. Typically, data falls on a range between inlier to noise to outlier.

We can also detect outliers in time series, spatial data, and graphs. Each data point can have categorical or numerical information.

2 The Data Model is Everything

Typically, we create a model for the normal data and the outlier score is how much the point differs from the normal model.

For example, suppose we have one-dimensional points $X_1, ..., X_N$. We can use a Gaussian model where we compute mean μ and variance σ and define the outlier scores as $Z_i = \frac{|X_i - \mu|}{\sigma}$. If N is small, we can use a Student's t distribution instead of a Gaussian. This Z value test fails if the underlying data is not Gaussian or if the outlier is not an extreme value, but is strange compared to its neighbors (contextual outlier).

You can think of outlier detection as binary classification where outliers constitute one class and normal points constitute the other. We can repurpose common machine learning models (e.g. SVM, Neural Net, Random Forest) for outlier detection. In addition to parametric models, we have instance based models like k-nearest neighbors and Local Outlier Factor (LOF), which tend to be more popular.

In unsupervised outlier detection, we don't use a training/validation/test set. This means tuning hyperparameters is hard, so try to use models with few or no hyperparameters.

3 The Basic Outlier Detection Models

We often want interpretable models, which provide hints about why a point is considered an outlier.

To measure the non-uniformity of a set of 1-dimensional points $x_1, ..., x_N$ is to compute mean μ, variance σ, z-scores $z_i = \frac{x_i - \mu}{\sigma}$ and combine them together to get the Kurtosis measure:

$$K(z_1, ..., z_N) = \frac{\sum_{i=1}^{N} z_i^4}{N} \tag{1}$$

To get this to work for multidimensional data, you can compute the Kurtosis measure on the Mahalanobis distances between points and their centroid. You can use the Kurtosis measure to identify features that could be good for outlier detection.

Another way to identify good features is to predict one feature given the others (i.e. make a supervised regression or classification model). If the feature is uncorrelated with the others, drop it.

Extreme Value Analysis can find outliers by looking for points that have very high or low values of a feature. In a multidimensional case, you can consider one feature at a time or do the analysis on the Mahalanobis distances to the centroid.

You can fit a probability distribution (e.g. Gaussian Mixture Model trained with Expectation Maximization) and use the PDF values as outlier scores. You can then run Extreme Value Analysis on the PDF values.

In linear methods, we project the data onto a lower dimensional hyperplane and use the distance from point to hyperplane as outlier score. This, and other dimensionality reduction techniques are not very interpretable.

Spectral methods, which are related to matrix factorization, are good for graph data.

Proximity methods are the most popular way to do outlier detection. They are clustering (segments data points), density (segments data space), or nearest-neighbor based. The distance to the kth nearest neighbor can be treated as the outlier score. k nearest neighbors is slow ($O(N^2)$), but we can prune out many points if we just need binary labels. In clustering, we cluster the points and take each point's distance to nearest cluster as outlier score (we run many copies of the algorithm and average the scores to overcome bad random initializations). Density methods like histograms are interpretable.

Information theoretic methods consider outliers to be points that increase the minimum code length required to describe the points. Frequent pattern mining, histograms, probabilistic models, and PCA can be used as information theoretic models.

For high dimensional data, it's hard to compute distances. So, we do subspace outlier detection, where we look for subspaces where anomalous behavior is exhibited. The subspaces are typically interpretable (e.g. Age ¡ 20 and Disease = 1).

4 Outlier Ensembles

We can combine many algorithms into an ensemble. A sequential ensemble (e.g. boosting) runs the algorithms one after the other where they depend on the previous algorithm. An independent ensemble runs the algorithms in parallel and averages their results.

5 The Basic Data Types for Analysis

Data points can have categorical, numerical, or mixed attributes. To deal with categorical variables, we can one-hot encode them, use Latent Semantic Analysis (LSA) (my note - you could also train an entity embedding).

Data points can be related to each other through a time series, graph, or spatial distribution. A point that is an outlier because of its neighbors is a contextual (or conditional) outlier. A group of points that are strange are called a collective anomaly.

For a time series, anomalies can be spikes, repetitions, changing frequencies, etc. Temporal data typically includes a trend (or concept drift), but this is slow and expected, so we should make sure not to let this fool us. Time series may be continuous vectors or be a sequence of events. For the latter, Hidden Markov Models work well. The former can use autoregressive techniques.

Data can also have a spatial distribution (e.g. temperature or pressure field).

Data can take the form a graph. A node outlier is a data point that is strange. An edge outlier is a relationship between points that is strange. We can also have a temporal component to graphs as they evolve over time.

6 Supervised Outlier Detection

If we have known outliers, we can use supervised learning. This is harder than regular supervised learning because the classes are heavily unbalanced. Some challenges are the Positive-Unlabled Classification problem (i.e. we have known outliers and a set of points whose outlier status is unknown), limited datasets (e.g. we do not know all the kinds of outliers, so we have a subset of them), and active learning (we select possible outlier points that we can send to a labeler who tells us for sure whether they are outliers or not).

7 Outlier Evaluation Techniques

If you have labeled examples, use that. If you have a specific task where you are using the model, use task specific metrics. You can also use internal validity metrics (e.g. mean squared radius of cluster), but it's easy to overfit if you rely on this.

If you have labeled points, use precision and recall as your metrics. Given a threshold t on outlier scores we can compute their predicted binary labels $S(t)$ and compare against the true labels G. We then have

$$\text{Precision} = \frac{|S(t) \cap G|}{|S(t)|} \tag{2}$$

$$\text{Recall} = \frac{|S(t) \cap G|}{|G|} \tag{3}$$

By varying t, you can plot a precision-recall curve. Alternatively, you can plot the true positive rate vs. false positive rate to get a Receiver Operating Characteristic (ROC) curve, which is easier to interpret.

$$TPR(t) = \frac{|S(t) \cap G|}{|G|} \tag{4}$$

$$FPR(t) = \frac{|S(t) - G|}{|D - G|} \tag{5}$$

The lift above the line $TPR = FPR$ indicates the superiority to a random method. If you want a single number, you can compute the area under the ROC curve (ROC AUC). The ROC AUC is the probability that a randomly selected outlier-inlier pair is ranked correctly. A random method gets ROC AUC of 0.5.

Usually, the initial part of the ROC curve matters more than the later parts. For example, it may not matter if a point is ranked 501 or 601, but it matters if it is 1 or 101. Use the normalized discounted cumulative gain in this case.

Avoid tuning hyperparameters to maximize your ROC AUC. So, you should predefine a set of hyperparameters and compute the median ROC AUC. You might also compute the variance to ensure that your algorithm is robust to different hyperparameter choices.

8 Conclusions and Summary

Outlier analysis has many applications and algorithms.