Chapter VIII: Outlier analysis

Knowledge Discovery in Databases

Luciano Melodia M.A.
Evolutionary Data Management, Friedrich-Alexander University Erlangen-Nürnberg
Summer semester 2021

Chapter VIII: Outlier Analysis

What are Outliers?

Outlier:

A data object that **deviates significantly** from the normal objects as if it were generated by a different mechanism.

I.e. unusual credit card purchase, or in Sports: Michael Jordon, Wayne Gretzky, . . .

Outliers are different from noise.

Noise is a random error or variance in a measured variable.

Noise should be removed before outlier detection.

Outliers are interesting.

They violate the mechanism that generates the normal data.

Outlier detection vs. novelty detection:

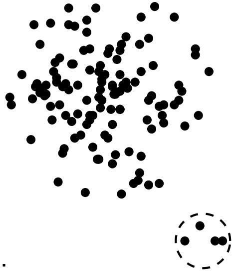
Early stage: outlier; but later merged into the model.

Where to use it?

Applications:

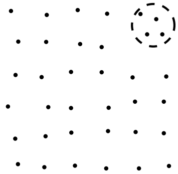
Credit-card-fraud detection.
Telecom-fraud detection.
Customer segmentation.
Medical analysis.

Types of Outliers



- Three kinds: global, contextual, and collective outliers
- **Global** outlier (or **point anomaly**):
  - Significantly deviates from the rest of the data set.
    - I.e. intrusion detection in computer networks.
  - Issue: Find an appropriate measurement of deviation.
- **Contextual** outlier (or conditional outlier):
  - Deviates significantly based on a selected context.
    - I.e. 80°F in Urbana outlier? (Depending on summer or winter).
  - Attributes of data objects divided into two groups:
    - **Contextual attributes**: define the context, e.g., time & location.
    - **Behavioral attributes**: characteristics of the object, used in outlier evaluation, e.g., temperature.
  - Can be viewed as a generalization of local outliers – whose density significantly deviates from its local area.
  - Issue: How to define or formulate meaningful context?

Types of Outliers (2)



Collective outlier:

A **subset** of data objects that collectively deviates significantly from the whole data set.
Ex.: intrusion detection – a number of computers keep sending denial-of-service packages to each other.

Detection of collective outliers:

Consider not only behavior of individual objects,but also that of groups of objects.
Need to have the background knowledge on the relationship among data objects, such as a distance or similarity measure on objects.

A data set may have multiple types of outliers.

One object may belong to more than one type of outlier.

Challenges of Outlier Detection

Modeling normal objects and outliers properly.

Hard to enumerate all possible normal behaviors in an application.

The border between normal and outlier objects is often a grey area.

Application-specific outlier detection.

Choice of distance measure among objects and the model of relationship among objects are application-dependent.

E.g. clinical data: a small deviation could be an outlier;

while in marketing analysis: larger fluctuations.

Challenges of Outlier Detection (II)

Handling noise in outlier detection.

Noise may distort the normal objects and blur the distinction between normal objects and outliers.

It may hide outliers and reduce the effectiveness of outlier detection.

Understandability.

Understand why these are outliers: justification of the detection.

Specify the degree of an outlier:

the unlikelihood of the object being generated by a normal mechanism.

Chapter VIII: Outlier Analysis

How can we detect outliers?

Two ways to categorize outlier-detection methods:

Based on whether **user-labeled examples of outliers** can be obtained:
I.e. supervised, semi-supervised vs. unsupervised methods.
Based on **assumptions** about normal data and outliers:
I.e. statistical, proximity-based, and clustering-based methods.

Outlier Detection I

Supervised Methods:

- Modeling outlier detection as a **classification problem**:
- Samples examined by domain experts used for training & testing.
- Methods for learning a classifier for outlier detection effectively:
  - Model normal objects & report those not matching the model as outliers.
  - Model outliers and treat those not matching the model as normal.
- **Challenges:**
  - Imbalanced classes, i.e., outliers are rare:
  - Boost the outlier class and make up some artificial outliers.
  - Catch as many outliers as possible,
  - i.e., recall is more important than accuracy
  - (i.e., not mislabeling normal objects as outliers).

Outlier Detection II

Assume the normal objects are somewhat "clustered" into multiple groups, each having some distinct features.

An outlier is expected to be far away from any group of normal objects.

Weakness: Can't detect collective outliers effectively.

Normal objects may not share any strong pattern,
but the collective outliers may have high similarity in a small area.

I.e., in some intrusion or virus detection, normal activities are diverse.

Unsupervised methods may have a high false-positive rate,
but still miss many real outliers.
Supervised methods can be more effective,
e.g., identify attacking some key resources.

Outlier Detection III

Many clustering methods can be adapted for unsupervised methods:

- Find clusters, then outliers: not belonging to any cluster.
- **Problem 1:** Hard to distinguish noise from outliers.
- **Problem 2:** Costly since first clustering, but far less outliers than normal objects.
- Newer methods: tackle outliers directly.

Outlier Detection IV

Situation:

In many applications, the **number of labeled data objects is small**:
Labels could be on outliers only, on normal objects only, or on both.

Semi-supervised outlier detection:

Regarded as application of semi-supervised learning.

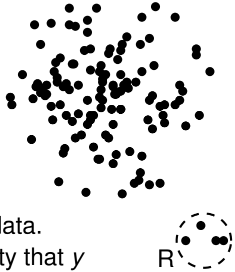
If some labeled normal objects are available:

Use the labeled examples and the proximate
unlabeled objects to train a model for normal objects.
Those not fitting the model of normal objects are detected as outliers.

If only some labeled outliers are available,
that small number may not cover the possible outliers well.

To improve the quality of outlier detection: get help from models for normal objects learned from unsupervised methods.

Outlier Detection V: Statistical Methods



(Also known as model-based methods)

Assume that the **normal data follow some statistical model**.

The data not following the model are outliers.

Example (right figure):

First use Gaussian distribution $\mathcal{N}_D(x \mid \mu, \sigma)$ to model the normal data.
For each object y in region R, estimate $\mathcal{N}_D(y \mid \mu, \sigma)$, the probability that y fits the Gaussian distribution.
If $\mathcal{N}_D(y \mid \mu, \sigma)$ is very low, y is unlikely generated by the Gaussian model, thus an outlier.

Effectiveness of statistical methods:

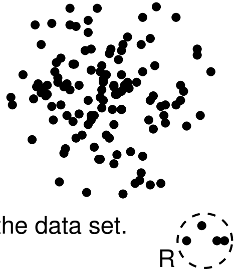
Highly depends on whether the assumption of statistical model holds in the real data.

There are many kinds of statistical models.

E.g., parametric vs. non-parametric.

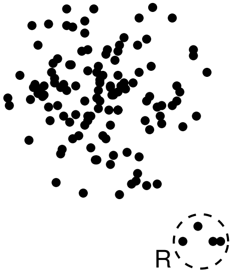
Outlier Detection (2): Proximity-Based Methods

An object is an outlier if the **nearest neighbors of the object are far away**, i.e., the proximity of the object significantly deviates from the proximity of most of the other objects in the same data set.



- **Example (right figure):**
  - Model the proximity of an object using its 3 nearest neighbors.
  - Objects in region R are substantially different from other objects in the data set.
  - Thus the objects in R are outliers.
- **Effectiveness of proximity-based methods:**
  - Highly relies on the proximity measure.
  - In some applications, proximity or distance measures cannot be obtained easily.
  - Often have a difficulty in finding a group of outliers which are close to each other.
- **Two major types of proximity-based outlier detection:**
  - Distance-based vs. density-based.

Outlier Detection (3): Clustering-Based Methods



Normal data belong to large and dense clusters, whereas outliers belong to **small or sparse clusters**, or do not belong to any cluster.

- **Example (right figure): Two clusters.**
  - All points not in R form a large cluster.
  - The two points in R form a tiny cluster, thus are outliers.
- **Many clustering methods:**
  - Thus also many clustering-based outlier detection methods.
- **Clustering is expensive.**
  - Straightforward adaptation of a clustering method for outlier detection can be costly and does not scale up well for large data sets.

Chapter VIII: Outlier Analysis

Statistical Approaches

Assume that the objects in a data set are **generated by a stochastic process** (a generative model).

Idea:

Learn a generative model fitting the given data set, and then identify the objects in low-probability regions of the model as outliers.

Methods divided into two categories:

Parametric vs. non-parametric.

Parametric method

Assumes that the normal data is generated by a parametric distribution with parameter θ.
The probability density function of the parametric distribution $f(x, \theta)$ gives the probability that object x is generated by the distribution.
The smaller this value, the more likely x is an outlier.

Statistical Approaches (2)

Non-parametric method:

Do not assume an a-priori statistical model
and determine the model from the input data.
Not completely parameter-free,
but consider number and nature of the parameters to be flexible and
not fixed in advance.
Examples: **histogram** and kernel-density estimation.

Parametric Methods I: Detection of Univariate Outliers Based on Normal Distribution

Univariate data:

- A data set involving only one attribute or variable.

Assumption:

- Data are generated from a normal distribution.

Learn the parameters from the input data, and identify the points with low probability as outliers.

- Use the **maximum-likelihood method** to estimate μ and σ.

The Maximum Likelihood Estimate of μ

Assumption:

Data is generated by an underlying Gaussian process.
Thus, the likelihood function $\mathcal{L}$ is the Gaussian process itself:

$$\mathcal{L}(\mathbf{X}) = \mathrm{P}(\mathbf{X} \mid \theta) = \mathcal{N}(\mathbf{X} \mid \theta) = \mathcal{N}(\mathbf{X} \mid \mu, \sigma). \tag{1}$$

We need to find good estimates for μ and σ:

$$\mu_{\mathrm{MLE}} = \mathrm{argmax}_{\mu}\ \mathcal{N}(\mathbf{X} \mid \mu, \sigma), \tag{2}$$

$$\sigma_{\mathrm{MLE}} = \mathrm{argmax}_{\sigma}\ \mathcal{N}(\mathbf{X} \mid \mu, \sigma). \tag{3}$$

To make computation easier, as the product of probabilities $\prod$ turns into sums $\sum$ under the log-function, we apply the logarithm. As log is monotonically increasing it holds that $\mathrm{argmax}_{\theta}\ \log f(\theta) = \mathrm{argmax}_{\theta}\ f(\theta)$.

The Maximum Likelihood Estimate of μ

We seek for the best parameters $\theta = \{\mu, \sigma\}$ for some dataset $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ of n data points. Thus we take the sum of the respective logarithms applied to the Gaussian:

$$\log\left(\mathcal{N}(\mathbf{X} \mid \theta)\right) = \sum_{i=1}^{n} \log\left(\mathcal{N}(\mathbf{x}_i \mid \theta)\right) = \sum_{i=1}^{n} \log\left(\mathcal{N}(\mathbf{x}_i \mid \mu, \sigma)\right). \tag{4}$$

The log-likelihood function then reads as:

$$\sum_{i=1}^{n} \log\left(\mathcal{N}(\mathbf{x}_i \mid \mu, \sigma)\right) = \sum_{i=1}^{n} \log\left(\frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp\left(-\frac{1}{2}\left(\frac{(\mathbf{x_i} - \mu)^2}{\sigma^2}\right)\right)\right). \tag{5}$$

Note, that we took a simplification here. The full covariance matrix Σ is replaced keeping only the diagonal elements σ^2, which is the variance. This is known as the assumption of diagonal covariance matrices.

The Maximum Likelihood Estimate of μ

Next, we use some algebra to get the log-likelihood, denoted by $\log \mathcal{L}(\mathbf{X})$, into a nicer form:

$$\log\left(\mathcal{L}(\mathbf{X})\right) = \sum_{i=1}^{n} \log\left(\frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp\left(-\frac{1}{2}\left(\frac{(\mathbf{x}_i - \mu)^2}{\sigma^2}\right)\right)\right) \tag{6}$$

$$= \sum_{i=1}^{n} \log\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) + \log\left(\exp\left(-\frac{1}{2}\left(\frac{(\mathbf{x}_i - \mu)^2}{\sigma^2}\right)\right)\right) \tag{7}$$

$$= \sum_{i=1}^{n} \log(1) - \log(\sqrt{2\pi\sigma^2}) + \log\left(\exp\left(-\frac{1}{2}\left(\frac{(\mathbf{x}_i - \mu)^2}{\sigma^2}\right)\right)\right) \tag{8}$$

$$= \sum_{i=1}^{n} \log(1) - \log\left(\sqrt{2\pi\sigma^2}\right) - \frac{1}{2}\left(\frac{(\mathbf{x}_i - \mu)^2}{\sigma^2}\right) \cdot \log(e). \tag{9}$$

We simplify the computation taking $\log$ with base e. Thus $\log_e e = 1$.
It also applies, regardless of base, that $\log(1) = 0$.

The Maximum Likelihood Estimate of μ

Applying the logarithm with base e yields:

$$\log\left(\mathcal{L}(\mathbf{X})\right) = \sum_{i=1}^{n} -\log\left(\sqrt{2\pi\sigma^2}\right) - \frac{1}{2}\left(\frac{(\mathbf{x}_i - \mu)^2}{\sigma^2}\right) \tag{10}$$

$$= \sum_{i=1}^{n} -\frac{1}{2}\log\left(2\pi\sigma^2\right) - \frac{1}{2}\left(\frac{(\mathbf{x}_i - \mu)^2}{\sigma^2}\right) \tag{11}$$

$$= -\frac{n}{2}\log\left(2\pi\sigma^2\right) + \sum_{i=1}^{n} -\frac{1}{2}\left(\frac{(\mathbf{x}_i - \mu)^2}{\sigma^2}\right) \tag{12}$$

$$= -\frac{n}{2}\log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(\mathbf{x}_i - \mu)^2. \tag{13}$$

The Maximum Likelihood Estimate of μ

In order to get $\operatorname{argmax}_\mu \log\left(\mathcal{L}(\mathbf{X})\right)$ we have to do two things:

1. Derive the partial derivative of the function with respect to the parameter.
2. Set the partial derivative to zero, and solve for μ.

In the same way we get $\operatorname{argmax}_\sigma \log\left(\mathcal{L}(\mathbf{X})\right)$. Thus,

$$\operatorname{argmax}_\mu \log\left(\mathcal{L}(\mathbf{X})\right) := \frac{\partial \log\left(\mathcal{L}(\mathbf{X})\right)}{\partial \mu} = 0. \tag{14}$$

We need to find the following partial derivative:

$$\frac{\partial \log\left(\mathcal{L}(\mathbf{X})\right)}{\partial \mu} = \frac{\partial}{\partial \mu}\left(-\frac{n}{2}\log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(\mathbf{x}_i - \mu)^2\right). \tag{15}$$

The Maximum Likelihood Estimate of μ

We start to simplify our partial derivative:

$$\frac{\partial \log\left(\mathcal{L}(\mathbf{X})\right)}{\partial \mu} = \frac{\partial}{\partial \mu}\left(-\frac{n}{2}\log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(\mathbf{x}_i - \mu)^2\right) \tag{16}$$

$$= \frac{\partial}{\partial \mu}\left(-\frac{n}{2}\log\left(2\pi\sigma^2\right)\right) + \frac{\partial}{\partial \mu}\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(\mathbf{x}_i - \mu)^2\right) \tag{17}$$

$$= 0 + \frac{\partial}{\partial \mu}\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(\mathbf{x}_i - \mu)^2\right) \tag{18}$$

$$= \frac{\partial}{\partial \mu}\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(\mathbf{x}_i - \mu)^2\right). \tag{19}$$

The Maximum Likelihood Estimate of μ

Next, we move the partial operator inside the sum:

$$\frac{\partial \log\left(\mathcal{L}(\mathbf{X})\right)}{\partial \mu} = \frac{\partial}{\partial \mu}\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(\mathbf{x}_i - \mu)^2\right) \tag{20}$$

$$= \frac{\partial}{\partial \mu}\left(\sum_{i=1}^{n} -\frac{1}{2\sigma^2}(\mathbf{x}_i - \mu)^2\right) \tag{21}$$

$$= \sum_{i=1}^{n} \frac{\partial}{\partial \mu}\left(-\frac{1}{2\sigma^2}(\mathbf{x}_i - \mu)^2\right) \tag{22}$$

$$= \sum_{i=1}^{n} \left(\frac{\partial}{\partial \mu}\left(-\frac{1}{2\sigma^2}\right) \cdot (\mathbf{x}_i - \mu)^2 + \left(-\frac{1}{2\sigma^2}\right) \cdot \frac{\partial}{\partial \mu}(\mathbf{x}_i - \mu)^2\right) . \tag{23}$$

The Maximum Likelihood Estimate of μ

After having applied the product rule, some of the terms drop out:

$$\frac{\partial \log\left(\mathcal{L}(\mathbf{X})\right)}{\partial \mu} = \sum_{i=1}^{n} \left(\frac{\partial}{\partial \mu} \left(-\frac{1}{2\sigma^2} \right) \cdot (\mathbf{x}_i - \mu)^2 + \left(-\frac{1}{2\sigma^2} \right) \cdot \frac{\partial}{\partial \mu} (\mathbf{x}_i - \mu)^2 \right) \tag{24}$$

$$= \sum_{i=1}^{n} \left(0 + \left(-\frac{1}{2\sigma^2} \right) \cdot \frac{\partial}{\partial \mu} (\mathbf{x}_i - \mu)^2 \right) \tag{25}$$

$$= -\frac{1}{2\sigma^2} \sum_{i=1}^{n} \frac{\partial}{\partial \mu} (\mathbf{x}_i - \mu)^2. \tag{26}$$

Using the chain rule we yield:

$$\frac{\partial \log\left(\mathcal{L}(\mathbf{X})\right)}{\partial \mu} = -\frac{1}{2\sigma^2} \sum_{i=1}^{n} 2(\mathbf{x}_i - \mu) \cdot -1 = \frac{1}{\sigma^2} \sum_{i=1}^{n} (\mathbf{x}_i - \mu). \tag{27}$$

The Maximum Likelihood Estimate of μ

Finally, we set this equation to 0 and solve the fixpoint equation according to μ:

$$\frac{1}{\sigma^2}\sum_{i=1}^{n}(\mathbf{x}_i - \mu) = 0, \tag{28}$$

$$\sum_{i=1}^{n}(\mathbf{x}_i - \mu) = 0, \tag{29}$$

$$\sum_{i=1}^{n}\mathbf{x}_i - \sum_{i=1}^{n}\mu = 0, \tag{30}$$

$$0 = \sum_{i=1}^{n}\mathbf{x}_i - n\mu, \tag{31}$$

$$n\mu = \sum_{i=1}^{n}\mathbf{x}_i, \tag{32}$$

$$\mu = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i. \tag{33}$$

The Maximum Likelihood Estimate of σ

We proceed similar for σ, and solve the partial differential equation according to our second parameter. But, first we simplify again the equation to something more handy:

$$\frac{\partial \log(\mathcal{L}(\mathbf{X}))}{\partial \sigma^2} = \frac{\partial}{\partial \sigma^2}\left(-\frac{n}{2}\log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(\mathbf{x}_i - \mu)^2\right) \tag{34}$$

$$= \frac{\partial}{\partial \sigma^2}\left(-\frac{n}{2}\log\left(2\pi\sigma^2\right)\right) + \frac{\partial}{\partial \sigma^2}\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(\mathbf{x}_i - \mu)^2\right). \tag{35}$$

We apply the product rule to the log-likelihood function:

$$\frac{\partial \log(\mathcal{L}(\mathbf{X}))}{\partial \sigma^2} = \frac{\partial}{\partial \sigma^2}(-\frac{n}{2}) \cdot \log\left(2\pi\sigma^2\right) - \frac{n}{2} \cdot \frac{\partial}{\partial \sigma^2}\log\left(2\pi\sigma^2\right) + \frac{\partial}{\partial \sigma^2}\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(\mathbf{x}_i - \mu)^2\right).$$

The Maximum Likelihood Estimate of σ

Again some of the terms drop out:

$$\frac{\partial \log\left(\mathcal{L}(\mathbf{X})\right)}{\partial \sigma^2} = -\frac{n}{2} \cdot \frac{\partial}{\partial \sigma^2} \log\left(2\pi\sigma^2\right) + \frac{\partial}{\partial \sigma^2}\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(\mathbf{x}_i - \mu)^2\right) . \tag{36}$$

Using the chain rule we yield:

$$\frac{\partial \log\left(\mathcal{L}(\mathbf{X})\right)}{\partial \sigma^2} = -\frac{n}{2} \cdot \frac{1}{2\pi\sigma^2} 2\pi + \frac{\partial}{\partial \sigma^2}\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(\mathbf{x}_i - \mu)^2\right) , \tag{37}$$

$$= -\frac{n}{2\sigma^2} + \frac{\partial}{\partial \sigma^2}\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(\mathbf{x}_i - \mu)^2\right) . \tag{38}$$

The Maximum Likelihood Estimate of σ

Moving the partial operator inside the sum and applying the product rule yields:

$$\frac{\partial \log\left(\mathcal{L}(\mathbf{X})\right)}{\partial \sigma^2} = -\frac{n}{2\sigma^2} + \sum_{i=1}^{n} \frac{\partial}{\partial \sigma^2}\left(-\frac{1}{2\sigma^2}(\mathbf{x}_i - \mu)^2\right) . \tag{39}$$

$$= -\frac{n}{2\sigma^2} + \sum_{i=1}^{n} \left(\frac{\partial}{\partial \sigma^2}\left(-\frac{1}{2\sigma^2}\right)(\mathbf{x}_i - \mu)^2 - \frac{1}{2\sigma^2} \cdot \frac{\partial}{\partial \sigma^2}(\mathbf{x}_i - \mu)^2\right) . \tag{40}$$

$$= -\frac{n}{2\sigma^2} + \sum_{i=1}^{n} \left(\frac{\partial}{\partial \sigma^2}\left(-\frac{1}{2\sigma^2}\right)(\mathbf{x}_i - \mu)^2\right) . \tag{41}$$

We switch the notation a little, such that the next steps become more intuitive:

$$\frac{\partial \log\left(\mathcal{L}(\mathbf{X})\right)}{\partial \sigma^2} = -\frac{n}{2\sigma^2} + \sum_{i=1}^{n} \left(\frac{\partial}{\partial \sigma^2}\left(-\frac{1}{2} \cdot \sigma^{-2}\right)(\mathbf{x}_i - \mu)^2\right) . \tag{42}$$

The Maximum Likelihood Estimate of σ

Once again we use the chain rule to simplify our log-likelihood derivative:

$$\frac{\partial \log\left(\mathcal{L}(\mathbf{X})\right)}{\partial \sigma^2} = -\frac{n}{2\sigma^2} + \sum_{i=1}^{n} \left(\frac{\partial}{\partial \sigma^2}\left(-\frac{1}{2}\right) \cdot \sigma^{-2} - \frac{1}{2}\frac{\partial}{\partial \sigma^2}\sigma^{-2} \cdot (\mathbf{x}_i - \mu)^2 \right) \tag{43}$$

$$= -\frac{n}{2\sigma^2} + \sum_{i=1}^{n} -\frac{1}{2}\frac{\partial}{\partial \sigma^2}\sigma^{-2} \cdot (\mathbf{x}_i - \mu)^2 \tag{44}$$

$$= -\frac{n}{2\sigma^2} + \sum_{i=1}^{n} -\frac{1}{2}\frac{\partial}{\partial \sigma^2}(\sigma^2)^{-1} \cdot (\mathbf{x}_i - \mu)^2 \tag{45}$$

$$= -\frac{n}{2\sigma^2} + \sum_{i=1}^{n} -\frac{1}{2} \cdot -1 \cdot (\sigma^2)^{-2} \cdot 1 \cdot (\mathbf{x}_i - \mu)^2 \tag{46}$$

$$= -\frac{n}{2\sigma^2} + \sum_{i=1}^{n} \frac{1}{2\sigma^4}(\mathbf{x}_i - \mu)^2. \tag{47}$$

The Maximum Likelihood Estimate of σ

We can simplify the equation a little more:

$$\frac{\partial \log(\mathcal{L}(\mathbf{X}))}{\partial \sigma^2} = -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_{i=1}^{n}(\mathbf{x}_i - \mu)^2 = -\frac{1}{2\sigma^2}\left(-n + \frac{1}{\sigma^2}\sum_{i=1}^{n}(\mathbf{x}_i - \mu)^2\right). \tag{48}$$

Finally, we set the equation to 0 and solve the fixpoint equation:

$$-\frac{1}{2\sigma^2}\left(-n + \frac{1}{\sigma^2}\sum_{i=1}^{n}(\mathbf{x}_i - \mu)^2\right) = 0, \tag{49}$$

$$-n + \frac{1}{\sigma^2}\sum_{i=1}^{n}(\mathbf{x}_i - \mu)^2 = 0, \tag{50}$$

$$\frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \mu)^2 = \sigma^2. \tag{51}$$

Parametric Methods I: Detection of Univariate Outliers Based on Normal Distribution (2)

Example:

Average temperature: $\{24.0, 28.9, 28.9, 29.0, 29.1, 29.1, 29.2, 29.2, 29.3, 29.4\}$.
For these data with $n = 10$, we have

$$\widehat{\mu} = 28.61, \quad \widehat{\sigma} = \sqrt{2.29} = 1.51. \tag{52}$$

Then the most deviating value 24.0 is 4.61 away form the estimated mean.
$\mu \pm 3\sigma$ contains 99.7% of the data under the assumption of normal distribution.
Because $\frac{4.61}{1.51} = 3.04 > 3$,
the probability that 24.0 is generated by a normal distribution is less than 0.15%.

Each *tail* to the left and to the right of the 99.7% has 0.15%.

Hence, 24.0 identified as an outlier.

Parametric Methods I: The Grubb's Test

Univariate outlier detection: The Grubb's test (maximum normed residual test).

Another statistical method under normal distribution

For each object x in a data set, compute its z-score: $z = \frac{|x - \overline{x}|}{s}$,

where $\overline{x}$ is the mean and s the standard deviation of the input data.

x is an outlier, if

$$z \geq \frac{N-1}{\sqrt{N}} \sqrt{\frac{t^2_{\frac{\alpha}{2N}, N-2}}{N - 2 + t^2_{\frac{\alpha}{2N}, N-2}}}, \tag{53}$$

where $t^2_{\frac{\alpha}{2N}, N-2}$ is the value taken by a t-distribution

at a significance level of $\frac{\alpha}{2N}$, and N is the number of objects in the data set.

Parametric Methods II: Detection of Multivariate Outliers

Multivariate data:

A data set involving **two or more attributes** or variables.

Transform the multivariate outlier-detection task into a univariate outlier-detection problem.

Method 1: Compute Mahalanobis distance.

Let $\bar{\mathbf{o}}$ be the mean vector for a multivariate data set. Mahalanobis distance for an object $\mathbf{o}$ to $\bar{\mathbf{o}}$ is $\Delta(\mathbf{o}, \bar{\mathbf{o}}) = (\mathbf{o} - \bar{\mathbf{o}})^T \mathbf{S}^{-1} (\mathbf{o} - \bar{\mathbf{o}})$ where $\mathbf{S}$ is the covariance matrix.
Use the Grubb's test on this measure to detect outliers.

Method 2: Use χ^2 statistic.

$$\chi^2 = \sum_{i=1}^{n} \frac{(\mathbf{o}_i - E_i)^2}{E_i}, \tag{54}$$

where E_i is the mean of the i-dimension among all objects, and n is the dimensionality.
If χ^2 statistic is large, then object $\mathbf{o}_i$ is an outlier.

Parametric Methods III: Using Mixture of Parametric Distributions

Assuming that data are generated by a normal distribution could sometimes be overly simplified.

Example (right figure):

The objects between the two clusters cannot be captured as outliers since they are close to the estimated mean.

Assume the normal data is generated by two normal distributions.

For any object **o** in the data set, the probability that **o** is generated by the mixture of the two distributions is given by

$$\mathrm{P}(\mathbf{o} \mid \theta_1, \theta_2) = f(\mathbf{o} \mid \theta_1) + f(\mathbf{o} \mid \theta_2), \tag{55}$$

where $f_{\theta 1}$ and $f_{\theta 2}$ are the probability density functions of θ_1 and θ_2.
Then use expectation-maximization (EM) algorithm to learn the parameters $\mu_1, \sigma_1, \mu_2, \sigma_2$ from the data.
An object **o** is an outlier if it does not belong to any cluster.

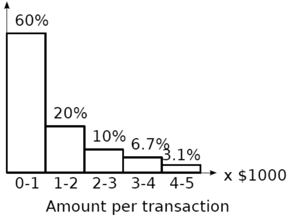
Non-Parametric Methods:Detection Using Histogram

The model of normal data is learned from the input data without any apriori structure.

Often makes fewer assumptions about the data, and thus can be applicable in more scenarios.

Outlier detection using histograms:

Figure shows the histogram of purchase amounts in transactions.
A transaction with the amount of $\$7,500$ is an outlier, since only 0.2% of the transactions have an amount higher than $\$5,000$.



Non-Parametric Methods: Detection Using Histogram (2)

Problem:

Hard to **choose an appropriate bin size** for histogram.

- Too small bin size $\rightarrow$ normal objects in empty/rare bins, false positive.
- Too big bin size $\rightarrow$ outliers in some frequent bins, false negative.

Solution:

Adopt kernel-density estimation to estimate the probability-density distribution of the data.

- If the estimated density function is high, the object is likely normal.
- Otherwise, it is likely an outlier.

Chapter VIII: Outlier Analysis

Proximity-Based Approaches: Distance-Based vs. Density-Based Outlier Detection

Intuition:

Objects that are **far away from the others** are outliers.

Assumption of proximity-based approach:

The proximity of an outlier deviates significantly from that of most of the others in the data set.

Two types of proximity-based outlier-detection methods:

Distance-based outlier detection:

An object **o** is an outlier,
if its neighborhood does not have enough other points.

Density-based outlier detection:

An object **o** is an outlier,
if its density is relatively much lower than that of its neighbors.

Distance-Based Outlier Detection

For each object **o**, examine the number of other objects in the **r-neighborhood** of **o**, where r is a user-specified **distance threshold**.

An object **o** is an outlier if most (taking π as a **fraction threshold**) of the objects in **D** are far away from **o**, i.e., not in the r-neighborhood of **o**.

An object o is a DB(r, π) outlier, iff

$$\frac{||\{\mathbf{o}' \mid d(\mathbf{o}, \mathbf{o}') \leq r\}||}{||D||} \leq \pi. \tag{56}$$

Equivalently, one can check the distance between **o** and its k-th nearest neighbor $\mathbf{o}_k$, where $k = \lceil \pi ||D|| \rceil$.

o is an outlier, if $d(\mathbf{o}, \mathbf{o}_k) > r$.

Distance-Based Outlier Detection (2)

Efficient computation: Nested-loop algorithm:

For any object $\mathbf{o}_i$, calculate its distance from other objects,
and count the number of other objects in the r-neighborhood.
If $\pi \cdot n$ other objects are within r-distance, terminate the inner loop.
Otherwise, $\mathbf{o}_i$ is a $\mathbf{DB}(r, \pi)$ outlier.

Efficiency:

Actually, CPU time is not $\mathcal{O}(n^2)$ but linear to the data set size,
since for most non-outlier objects, the inner loop terminates early.

Distance-Based Outlier Detection (3)

Why is efficiency still a concern?

If the complete set of objects cannot be held in main memory, there is significant cost for I/O swapping.

The major cost:

1. Each object is tested against the whole data set, why not only against its close neighbors?
2. Objects are checked one by one, why not group by group?

Distance-Based Outlier Detection:A Grid-Based Method

CELL:

Data space is partitioned into a multi-D grid.

Each cell is a hyper cube with diagonal length $\frac{r}{2}$.

r-distance threshold parameter.

l-dimensions: edge of each cell $r/(2\sqrt{l})$ long.

Level-1 cells:

Immediately next to cell **C**.

For any possible point **x** in **C** and

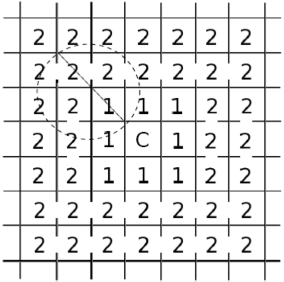
any possible point **y** in a level-1 cell: $d(x, y) \leq r$.

Level-2 cells:

One or two cells away from **C**.

For any possible point **x** in cell **C** and

any point **y** such that $d(x, y) \geq r$, **y** is in a level-2 cell.



Distance-Based Outlier Detection: A Grid-Based Method (2)

Total number of objects in cell **C**: a.

Total number of objects in level-1 cells: b_1.

Total number of objects in level-2 cells: b_2.

Level-1 cell pruning rule:

If $a + b_1 > \lceil \pi n \rceil$, then every object **o** in **C** is not a **DB**(r, π) outlier, because all objects in **C** and the level-1 cells are in the r-neighborhood of **o**,and there are at least $\lceil \pi n \rceil$ such objects.

Level-2 cell pruning rule:

If $a + b_1 + b_2 < \lceil \pi n \rceil + 1$, then all objects in **C** are **DB**(r, π) outliers, because all of their r-neighborhoods have less than $\lceil \pi n \rceil$ other objects.

Only need to check the objects that cannot be pruned.

Even for such an object **o**,
only need to compute the distance between **o** and the objects in level-2 cells.

Since beyond level-2, distance from **o** is more than r.

Density-Based Outlier Detection

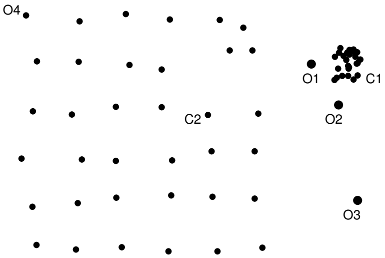
Local outliers:

Outliers compared to their local neighborhoods, not to global data distribution.

In the figure, O1 and O2 are local outliers to C1, O3 is a global outlier, but O4 is not an outlier.
However, distance of O1 and O2 to objects in dense cluster C1 is smaller than average distance in sparse cluster C2.
Hence, O1 and O2 are not distance-based outliers.

Intuition:

Density around **outlier** object **significantly different** from density around its neighbors.



Density-Based Outlier Detection (2)

Method:

Use the **relative density** of an object against its neighbors as the indicator of the degree of the object being outliers

k-**distance of an object o:** $d_k(\mathbf{o})$.

Distance $d(\mathbf{o}, \mathbf{p})$ between **o** and its k-nearest neighbour p.

Test at least k objects **o'** $\in \mathbf{D} - \{\mathbf{o}\}$
such that $d(\mathbf{o}, \mathbf{o}') \leq d(\mathbf{o}, \mathbf{p})$.
at most $k - 1$ objects $\mathbf{o}'' \in \mathbf{D} - \{\mathbf{o}\}$
such that $d(\mathbf{o}, \mathbf{o}') > d(\mathbf{o}, \mathbf{p})$.

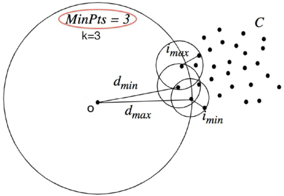
k-distance neighborhood of **o**:

$N_k(\mathbf{o}) = \mathbf{o}' \mid \mathbf{o}' \in \mathbf{D}, d(\mathbf{o}, \mathbf{o}') \leq d_k(\mathbf{o})$.
$N_k(\mathbf{o})$ could be bigger than k
since multiple objects may have identical distance to **o**.

Local Outlier Factor

Reachability distance from o$'$ to o:

$$\text{reachdist}_k(\mathbf{o}' \leftarrow \mathbf{o}) = \max\{d_k(\mathbf{o}), d(\mathbf{o}, \mathbf{o}')\}$$



where k is a user-specified parameter.

Local reachability density of o:

$$\text{ldr}_k(\mathbf{o}) = \frac{||N_k(\mathbf{o})||}{\sum_{\mathbf{o}' \in N_k(\mathbf{o})} \text{reachdist}_k(\mathbf{o}' \leftarrow \mathbf{o})}.$$

LOF (Local Outlier Factor) of o:

The average of the ratio of local reachability of **o** and those of **o**'s k-nearest neighbors.

$$\text{LOF}_k(\mathbf{o}) = \frac{\sum_{\mathbf{o}' \in N_k(\mathbf{o})} \frac{\text{lrd}_k(\mathbf{o}')}{\text{lrd}_k(\mathbf{o})}}{||N_k(\mathbf{o})||} = \sum_{\mathbf{o}' \in N_k(\mathbf{o})} \text{lrd}_k(\mathbf{o}') \cdot \sum_{\mathbf{o}' \in N_k(\mathbf{o})} \text{reachdist}_k(\mathbf{o}' \leftarrow \mathbf{o}).$$

The lower the local reachability density of **o**, and the higher the local reachability density of the k-NN of **o**, the higher LOF.

This captures a local outlier whose local density is relatively low comparing to the local densities of its k-NN.

Chapter VIII: Outlier Analysis

Summary

Types of outliers:

Global, contextual & collective outliers.

Outlier detection:

Supervised, semi-supervised, or unsupervised.

Statistical (or model-based) approaches.

Proximity-based approaches.

Not covered here:

Clustering-based approaches.
Classification approaches.
Mining contextual and collective outliers.
Outlier detection in high dimensional data.

References

B. Abraham and G.E.P. Box: Bayesian analysis of some outlier problems in time series. Biometrika, 66:229 – 248, 1979.

M. Agyemang, K. Barker, and R. Alhajj: A comprehensive survey of numeric and symbolic outlier mining techniques. Intell. Data Anal., 10:521 – 538, 2006.

F. J. Anscombe and I. Guttman: Rejection of outliers. Technometrics, 2:123 – 147, 1960.

D. Agarwal: Detecting anomalies in cross-classified streams: a Bayesian approach. Knowl. Inf. Syst., 11:29 – 44, 2006.

F. Angiulli and C. Pizzuti: Outlier mining in large high-dimensional data sets. TKDE, 2005.

C.C. Aggarwal and P.S. Yu: Outlier detection for high dimensional data. SIGMOD'01.

R.J. Beckman and R.D. Cook. Outlier...s. Technometrics, 25:119–149, 1983.

I. Ben-Gal: Outlier detection. In: O. Maimon and L. Rockach (eds.), Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers, Kluwer Academic, 2005.

References (2)

M.M. Breunig, H.-P. Kriegel, R. Ng, and J. Sander. LOF: Identifying density-based local outliers. SIGMOD'00.

D. Barbara, Y. Li, J. Couto, J.-L. Lin, and S. Jajodia: Bootstrapping a data mining intrusion detection system. SAC'03.

Z.A. Bakar, R. Mohemad, A. Ahmad, and M. M. Deris: A comparative study for outlier detection techniques in data mining. IEEE Conf. on Cybernetics and Intelligent Systems, 2006.

S. D. Bay and M. Schwabacher: Mining distance-based outliers in near linear time with randomization and a simple pruning rule. KDD'03.

D. Barbara, N. Wu, and S. Jajodia: Detecting novel network intrusion using Bayesian estimators. SDM'01.

V. Chandola, A. Banerjee, and V. Kumar: Anomaly detection: A survey. ACM Computing Surveys, 41:1 – 58, 2009.

D. Dasgupta and N.S. Majumdar: Anomaly detection in multidimensional data using negative selection algorithm. CEC'02.

References (3)

E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo: A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. Int. Conf. of Data Mining for Security Applications, 2002.

E. Eskin: Anomaly detection over noisy data using learned probability distributions. ICML'00.

T. Fawcett and F. Provost: Adaptive fraud detection. Data Mining and Knowledge Discovery, 1:291 – 316, 1997.

V.J. Hodge and J. Austin: A survey of outlier detection methodologies. Artif. Intell. Rev., 22:85 – 126, 2004.

D. M. Hawkins: Identification of Outliers. Chapman and Hall, London, 1980.

Z. He, X. Xu, and S. Deng: Discovering cluster-based local outliers. Pattern Recogn. Lett., 24, June, 2003.

W. Jin, K. H. Tung, and J. Han: Mining top-*n* local outliers in large databases. KDD'01.

References (4)

W. Jin, A. K. H. Tung, J. Han, and W. Wang: Ranking outliers using symmetric neighborhood relationship. PAKDD'06.

E. Knorr and R. Ng: A unified notion of outliers: Properties and computation. KDD'97.

E. Knorr and R. Ng: Algorithms for mining distance-based outliers in large datasets. VLDB'98.

E. M. Knorr, R. T. Ng, and V. Tucakov: Distance-based outliers: Algorithms and applications. VLDB J., 8:237–253, 2000.

H.-P. Kriegel, M. Schubert, and A. Zimek: Angle-based outlier detection in high-dimensional data. KDD'08.

M. Markou and S. Singh: Novelty detection: A review – Part 1: Statistical approaches. Signal Process., 83:2481–2497, 2003.

M. Markou and S. Singh: Novelty detection: A review – Part 2: Neural network based approaches. Signal Process., 83:2499–2521, 2003.

C. C. Noble and D. J. Cook: Graph-based anomaly detection. KDD'03.

References (5)

S. Papadimitriou, H. Kitagawa, P.B. Gibbons, and C. Faloutsos: Loci: Fast outlier detection using the local correlation integral. ICDE'03.

A. Patcha and J.-M. Park: An overview of anomaly detection techniques: Existing solutions and latest technological trends. Comput. Netw., 51, 2007.

X. Song, M. Wu, C. Jermaine, and S. Ranka: Conditional anomaly detection. IEEE Trans. on Knowl. and Data Eng., 19, 2007.

Y. Tao, X. Xiao, and S. Zhou: Mining distance-based outliers from large databases in any metric space. KDD'06.

N. Ye and Q. Chen: An anomaly detection technique based on a χ-square statistic for detecting intrusions into information systems. Quality and Reliability Engineering International, 17:105–112, 2001.

B.-K. Yi, N. Sidiropoulos, T. Johnson, H. V. Jagadish, C. Faloutsos, and A. Biliris: Online data mining for co-evolving time sequences. ICDE'00.

Thank you for your attention.
Any questions about the eighth chapter?

Ask them now, or again, drop me a line:
`luciano.melodia@fau.de`.