Artificial Intelligence in Industry

Last update: 19 September 2024

Academic Year 2024 – 2025

Alma Mater Studiorum · University of Bologna

Contents

1 Preliminaries

Problem formalization Defines the ideal goal.

Problem formalization

Solution formalization Defines the actual possible approaches to solve a problem.

Solution formalization

Occam's razor Principle for which, between two hypotheses, the simpler one is usually correct.

Occam's razor

Remark. This approach has less variance and more bias, making it more robust.

2 Anomaly detection: Taxi calls

Anomaly Event that deviates from the usual pattern. Anomaly

Time series Data with an ordering (e.g., chronological). Time series

2.1 Data

The dataset is a time series and it is a `DataFrame` with the following fields:

`timestamp` with a 30 minutes granularity.

`value` number of calls.

The label is a `Series` containing the timestamps of the anomalies.
An additional `DataFrame` contains information about the time window in which the anomalies happen:

`begin` acceptable moment from which an anomaly can be detected.

`end` acceptable moment from which there are no anomalies anymore.

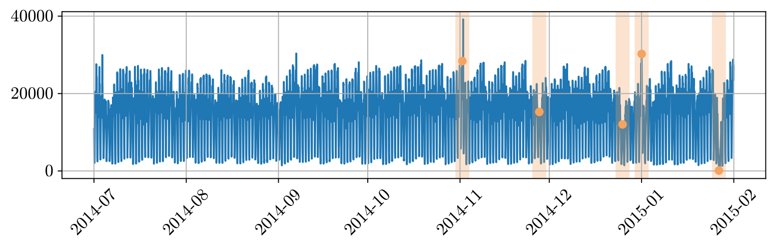

Figure 2.1: Plot of the time series, anomalies, and windows

2.2 Approaches

2.2.1 Gaussian assumption

Assuming that the data follows a Gaussian distribution, mean and variance can be used to determine anomalies through a threshold. z-score can also be used.

2.2.2 Characterize data distribution

Classify a data point as an anomaly if it is too unlikely.

Problem formalization Given a random variable X with values x to represent the number of taxi calls, we want to find its probability density function (PDF) $f(x)$.

An anomaly is determined whether:

$$f(x) \leq \varepsilon$$

where ε is a threshold.

Remark. A PDF can be reasonably used even though the dataset is discrete if its data points are sufficiently fine-grained.

Remark. It is handy to use negated log probabilities as:

- The logarithm adds numerical stability.
- The negation makes the probability an alarm signal, which is a more common measure.

Therefore, the detection condition becomes:

$$-\log f(x) \geq \varepsilon$$

Solution formalization The problem can be tackled using a density estimation technique.

2.2.2.1 Univariate kernel density estimation

Kernel density estimation (KDE) Based on the assumption that whether there is a data point, there are more around it. Therefore, each data point is the center of a density kernel.

Kernel density estimation (KDE)

Density kernel A kernel $K(x, h)$ is defined by:

- The input variable x.
- The bandwidth h.

Gaussian kernel Kernel defined as:

$$K(x, h) \propto e^{-\frac{x^2}{2h^2}}$$

where:

- The mean is 0.
- h is the standard deviation.

As the mean is 0, an affine transformation can be used to center the kernel on a data point μ as $K(x - \mu, h)$.

Given m training data points $\bar{x}_i$, the density of any point x can be computed as the kernel average:

$$f(x, \bar{x}, h) = \frac{1}{m} \sum_{i=0}^{m} K(x - \bar{x}_i, h)$$

Therefore, the train data themselves are used as the parameters of the model while the bandwidth h has to be estimated.

Remark. According to some statistical arguments, a rule-of-thumb to estimate h in the univariate case is the following:

$$h = 0.9 \cdot \min\left\{\hat{\sigma}, \frac{\texttt{IQR}}{1.34}\right\} \cdot m^{-\frac{1}{5}}$$

where:

- IQR is the inter-quartile range.
- $\hat{\sigma}$ is the standard deviation computed over the whole dataset.

Data split Time series are usually split chronologically:

Train Should ideally contain only data representing the normal pattern. A small amount of anomalies might be tolerated as they have low probabilities.

Validation Used to find the threshold ε.

Test Used to evaluate the model.

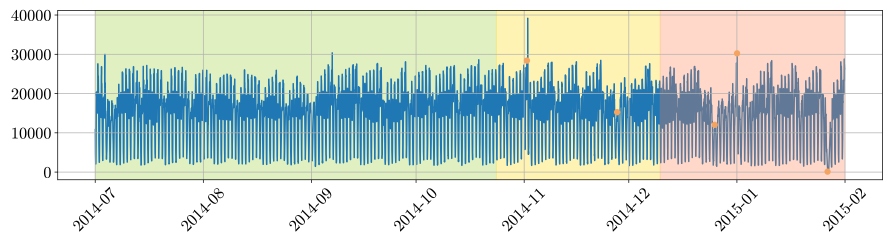

Figure 2.2: Train, validation, and test splits

Metrics It is not straightforward to define a metric for anomaly detection. A cost model to measure the benefits of a prediction is more suited. A simple cost model can be based on:

True positives (TP) Windows for which at least an anomaly is detected;

False positives (FP) Detections that are not actually anomalies;

False negatives (FN) Undetected anomalies;

Advance (adv) Time between an anomaly and when it is first detected;

and is computed as:

$$(c_{\text{false}} \cdot \texttt{FP}) + (c_{\text{miss}} \cdot \texttt{FN}) + (c_{\text{late}} \cdot \texttt{adv}_{\leq 0})$$

where c_{false}, c_{miss}, and c_{late} are hyperparameters.

Threshold optimization Using the train and validation set, it is possible to find the best threshold ε that minimizes the cost model through linear search.

Remark. The train set can be used alongside the validation set to estimate ε as this operation is not used to prevent overfitting.

Remark. The evaluation data should be representative of the real world distribution. Therefore, in this case, to evaluate the model the whole dataset can be used.

Remark. KDE assumes that the Markov property holds. Therefore, each data point is considered independent to the others.

2.2.2.2 Multivariate kernel density estimation

Remark. In this dataset, nearby points tend to have similar values.

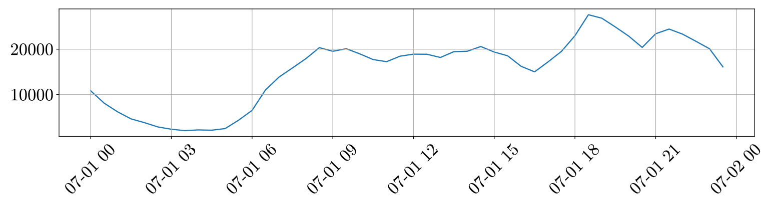

Figure 2.3: Subset of the dataset

Autocorrelation plot Plot to visualize the correlation between nearby points of a series. Given the original series, it is duplicated, shifted by a lag l, and the Pearson correlation coefficient is then computed between the two series. This operation is repeated over different values of l.

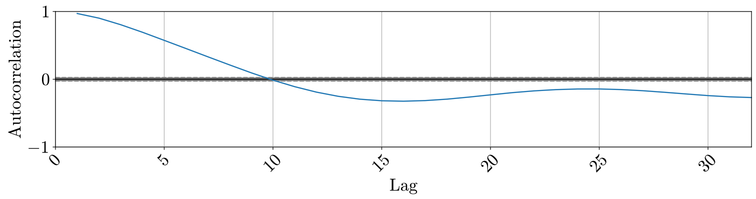

Figure 2.4: Autocorrelation plot of the subset of the dataset. There is strong correlation up to 4-5 lags.

Sliding window Given a window size w and a stride s, the dataset is split into sequences of w continuous elements.

Remark. Incomplete sequences at the start and end of the dataset are ignored.

Remark. In `pandas`, the `rolling` method of `Dataframe` allows to create a slicing window iterator. This approach creates the windows row-wise and also considers incomplete windows. However, a usually more efficient approach is to construct the sequences column-wise by hand.

Multivariate KDE Extension of KDE to vector variables.