Artificial Intelligence in Industry

Matteo Donati

October 16, 2022

Contents

1 Anomaly Detection via Simple Methods

1.1 Problem and Data

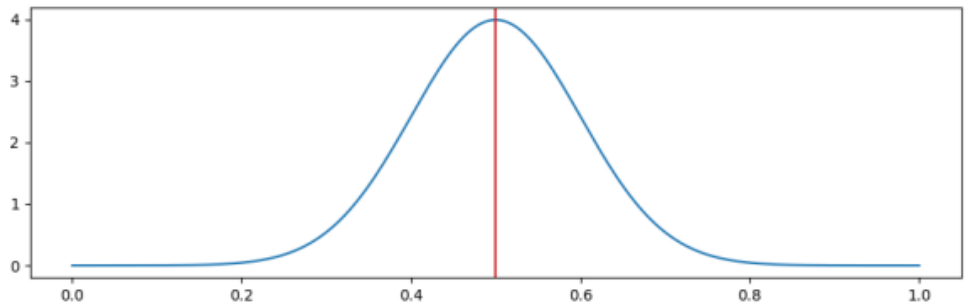
The goal of anomaly detection is to detect, analyze and anticipate abnormal situations (i.e. **anomalies**). Usually, anomaly detection is based on time-series analysis, where a time-series is a sequence whose index represents time. Time-series have one difference with respect to classical table datasets: their row index is meaningful, since it represents the position of the example in the sequence. The labels associated with such a dataset usually indicate an instant of time at which an anomaly occurs. Considering a specific anomaly, one could also compute the window of time inside which the specific anomaly occurs.

1.2 Anomaly Detection and Kernel Density Estimation

A possible approach to detect anomalies is based on the fact that anomalies are often unlikely. If one can estimate the probability of every occurring observation x, then one can spot anomalies based on their low probability. Formally, a detection condition can be states as $f(x) \leq \theta$, where $f(x)$ is a **probability density function** (**PDF**), and θ is a scalar threshold. Given some training data $\hat{\mathbf{x}}$, the true density function $f^*(x) : \mathbb{R}^n \to \mathbb{R}^+$, and a second function $f(x, \omega)$, a supervised learning approach to estimate the probability densities considers a suitable loss function, $L(y, y^*)$, that has to be optimized so to find the best set of parameters ω that minimizes the considered loss:

$$\text{argmin}_\omega \, L(f(\hat{\mathbf{x}}, \omega), f^*(\hat{\mathbf{x}})) \tag{1}$$

However, this approach cannot work, because usually one does not have access to the true density f^*. Thus, density estimation is an unsupervised learning problem. Such problem can be solves via a number of techniques (e.g. via Kernel Density Estimation). In **Kernel Density Estimation** (**KDE**) the main idea is that wherever, in the input space, there is a sample, then it is likely that there are more samples, so one can assume that each training sample is the center for a density kernel. Formally, the kernel $K(x, h)$ is just a valid PDF, where x is the input variable (scalar or vector), and h is a parameter (scalar or matrix respectively) called *bandwidth*. For example, given a single sample $x = 0.5$, then a Gaussian estimator with $h = 0.1$ will produce the following:



Indeed, in `sklearn`, a Gaussian kernel is given by:

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

which is similar to the PDF of the Normal distribution, where the mean can be interpreted as zero, and h controls the standard deviation of the distribution. However, since the mean is zero, the kernel will be centered on zero. To solve this, one can use an affine transformation, $K(x - \mu, h)$, which gives the value of a kernel computed for the value x and centered on μ. Moreover, the estimated density of any point is obtained as a kernel average:

$$f(x, \hat{\mathbf{x}}, h) = \frac{1}{m} \sum_{i=0}^{m} K(x - \hat{x}_i, h) \tag{3}$$

where x is the input for which to compute the estimate, $\hat{\mathbf{x}}$ is the matrix containing the training samples, $x - \hat{x}_i$ is the difference between x and the i-th training sample. Thus, KDE models are not trained in the usual sense: the training set is part of the model parameters. The only thing that one needs to train is h. For the univariate case, one can apply the following rule of thumb:

$$h = 0.9 \min\left(\hat{\sigma}, \frac{IQR}{1.34}\right) m^{-\frac{1}{5}} \tag{4}$$

where IQR is the inter-quartile range.