

1 Anomaly detection

1.1 Introduction

Anomaly detection is an unsupervised learning problem that has some aspects similar to supervised learning problem.

Given m normal(non-anomalous) examples $x^{(1)}, x^{(2)}, \dots, x^{(m)}$, we are supposed to tell whether a new example x_{test} is anomalous. The approach we will take is to build a density model $p(x)$ and choose a threshold ϵ . If $p(x_{test}) \geq \epsilon$, the new example x_{test} is flagged normal; otherwise it is recognized as an anomaly.

Anomaly detection could be used in fraud detection. Also it can be used to decide whether a product is up to the normal quality standard in manufacturing. What's more, it can be used to monitor the status of computers in a data center.

1.2 Algorithm

Suppose the training examples $x^{(i)} \in \mathbb{R}^n$. We will assume that

$$p(x) = \prod_{i=1}^n p(x_i; \mu_i, \sigma_i^2)$$

in which $p(x_i; \mu_i, \sigma_i^2)$ means $x_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$. We are actually taking for granted that different features x_i are independent. Although this is not always true, it does not harm the correctness of the algorithm.

First we will pick out the features x_i that might be indicative of anomalous examples. Then we fit the parameters μ_i, σ_i^2 with

$$\begin{aligned}\mu_j &= \frac{1}{m} \sum_{i=1}^m x_j^{(i)} \\ \sigma_j^2 &= \frac{1}{m} \sum_{i=1}^m (x_j^{(i)} - \mu_j)^2.\end{aligned}\tag{1}$$

Now that we have the model

$$p(x) = \prod_{j=1}^n p(x_j; \mu_j, \sigma_j^2) = \prod_{j=1}^n \frac{1}{\sqrt{(2\pi)\sigma_j^2}} \exp\left(-\frac{(x_j - \mu_j)^2}{2\sigma_j^2}\right),\tag{2}$$

we can calculate $p(x_{test})$ for any new example x_{test} and tell whether it is an anomaly.

1.3 Developing and evaluating an anomaly detection system

Assume we have some labeled data with anomalous and non-anomalous examples ($y=0$ normal, $y=1$ anomalous). In order to choose an superior anomaly

detection algorithm, we will divide the training examples into, as usual, training set, cross-validation set and test set. Note that in this case, the training set contains only normal examples, while the cv set and the test set contain both normal and anomalous examples.

In a typical case, if we have 10000 normal examples and 20 anomalies, we can put 6000 normal examples into the training set, 2000 normal examples and 10 anomalies into the cv set, and 2000 normal examples and 10 anomalies into the test set.

First, we will fit the model $p(x)$ with the training set $\{x^{(1)}, \dots, x^{(m)}\}$. Then, we will make predictions for the examples in the cv set with

$$y = \begin{cases} 1 & \text{if } p(x) < \epsilon \text{ (anomaly)} \\ 0 & \text{if } p(x) \geq \epsilon \text{ (normal)} \end{cases} \quad (3)$$

Since the data is quite skewed (most examples are normal), predication accuracy is not a good evaluation metric. We can calculate the numbers of true positive, false positive, true negative, false negative, and then calculate **Precision/Recall**, and finally evaluate the algorithm with the F_1 score.

We will choose the value of ϵ that provides the best F_1 score on the cv set. Then the algorithm can be evaluated by applying the model on the test set.

1.4 Anomaly detection v.s. supervised learning

In the section above, we labeled anomalies with $y = 1$ in order to properly evaluate the algorithm. This could be confusing since anomaly detection is an unsupervised learning algorithm, while labeling examples is what we do in supervised learning. It seems that the problem could be solved directly with supervised learning methods. Here are some guidelines helping us to decide whether to use anomaly detection or supervised learning when facing a specific problem.

1. Anomaly detection should be used when the number of positive examples is very small while the number of negative examples is large. Supervised learning should be used when there are large number of both positive and negative examples.
2. Anomaly detection should be used when there are many “types” of anomalies, i.e. it seems unlikely that we are able to predict what future anomalies will look like based on the anomalies that have appeared. Supervised learning should be used when there are enough positive examples to help us get a sense of what an anomaly should “look like”, i.e. we are confident that future anomalies will be similar to the ones we have seen.

1.5 Choosing features to use

The choice of features to be used in anomaly detection can be ambiguous and subtle.

When a feature does not “look gaussian” according to its histogram, we can make appropriate transformations to help it become gaussian. For instance, use $\log(x)$, $\log(x+1)$, $x^{1/2}$, $x^{1/3}$, etc instead of x .

When the current choice of features fails to differentiate an anomaly from normal examples, we should examine the anomaly and try to come up with a new feature from it and add it to the model.

A useful general principle for the choice of features in anomaly detection is to choose features that are likely to take unusually large or small values in the case of an anomaly.

1.6 Multivariate Gaussian distribution

As mentioned above, we have been taking for granted that $p(x_i)$ are independent from each other. Instead of using $p(x) = \prod_{i=1}^n p(x_i)$, we can model $p(x)$ all in one go with multivariate Gaussian distribution:

$$p(x; \mu, \Sigma) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu)^\top \Sigma^{-1}(x - \mu)\right) \quad (4)$$

in which Σ is the covariance matrix. This is a better approach when the features chosen are correlated to each other closely.

When using multivariate Gaussian distribution, when given a training set $\{x^{(1)}, \dots, x^{(m)}\}$, we can fit the model with

$$\begin{aligned} \mu &= \frac{1}{m} \sum_{i=1}^m x^{(i)} \\ \Sigma &= \frac{1}{m} \sum_{i=1}^m (x^{(i)} - \mu)(x^{(i)} - \mu)^\top. \end{aligned} \quad (5)$$

When given a new example, we should calculate

$$p(x) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu)^\top \Sigma^{-1}(x - \mu)\right)$$

and flag an anomaly if it turns out that $p(x) < \epsilon$.

It is not difficult to figure out that the original model used above is actually the multivariate gaussian model with

$$\Sigma = \begin{bmatrix} \sigma_1^2 & & & \\ & \sigma_2^2 & & \\ & & \ddots & \\ & & & \sigma_n^2 \end{bmatrix}. \quad (6)$$

When using the original model, we have to manually create extra features to capture anomalies that involve unusual combinations of existing features, while

multivariate gaussian model automatically captures the correlations between features. However, we should be cautious to use multivariate gaussian when there are a lot of features because it is computationally more expensive (calculation of Σ^{-1}). What's more, multivariate gaussian requires $m > n$ to ensure the invertibility of Σ . In practice, it should only be used when m is sufficiently larger than n , e.g. $m > 10n$.

2 Recommender systems