Project: Detecting failing computer servers on a network

B. Shadrack Jabes

May 20, 2020

1 Introduction

The aim of this project is to write an algorithm to detect the anomalous behavior in computer servers. There are about 307 servers and as the computer servers are functioning two features of the server's are observed: 1. latency which determines transfer speed (ms) of contents within a pipe from the client to the server and back, 2. Throughput of the server (mb/s) which determines the amount of data transferred within a period of time. I build a gaussian model as a function of these two features of the server. I then visualize the density estimation from this gaussian model using an evaluation metric ϵ and identify the anomalous server. Thus using this anomaly detection algorithm one can identify the anomalous server within the dataset.

2 Identifying features

The dataset consists of two features of the server they are throughput (mb/s) and latency (ms). There are a total of 307 servers from which this data is collected. Majority of the servers operate normaly and only few exceptional servers may require the attention of the system administrator (see figure 2).

3 Density estimation: Gaussian model

Firstly I compute the total probability density, $p(X)$, where X are the features of the computer server. The total probability density, $p(X)$ can be written as the product of probability densities of individual features. To obtain the probability densities of individual features we fit each feature X to a gaussian model and estimate the gaussian parameters correspoinding to each feature. Here we assume that each feature is independent therefore can be fitted with a gaussian model. Once we obtain the probability densities of individual features we take the product of each probability densities and compute the total probability density $p(x)$ as follows:

$$p(X) = \prod_{j=1}^{n} P(X_j; \mu_j, \sigma_j^2) \qquad (1)$$

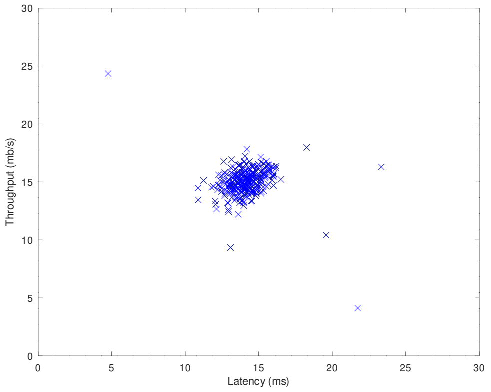

Figure 1: Latency and throughput are the features of the computer server collected over 307 servers.

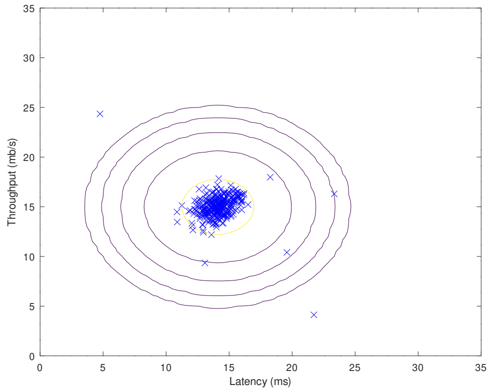

Figure 2: Contour plot shownning the probability density and the features.

where n is the number of features, μ and σ^2 are the mean and the corresponding variance of the features. The mean and variance of each feature i over the total number of servers j are computed as follows:

$$\mu = \frac{1}{m}\sum_{j=1}^{m} X^{(j)} \tag{2}$$

$$\sigma_i^2 = \frac{1}{m}\sum_{j=1}^{m}(X^{(j)} - \mu_j)^2 \tag{3}$$

Where m are the total number of servers.

4 Evaluation metric: ϵ

We now select the evaluation metric for the probability density and predict the anomaly such that if the probability density is below ϵ then the chances that the server is anomalous (flaged as $y = 1$), else categorize the servers to be normal (flagged as $y = 0$). To do that one needs to estimate the threshold ϵ. The problem here is that the data set is skeweed, that is we have more examples with features that are normal and only fewer examples that are anomalous. Therefore from the dataset I evaluate F1-score for various choice of threshold ϵ. I choose the ϵ valuse that corresponds to a high F1-score. The F1-score can be estimated as follow:

$$F1 = \frac{2 * P * R}{P + R} \tag{4}$$

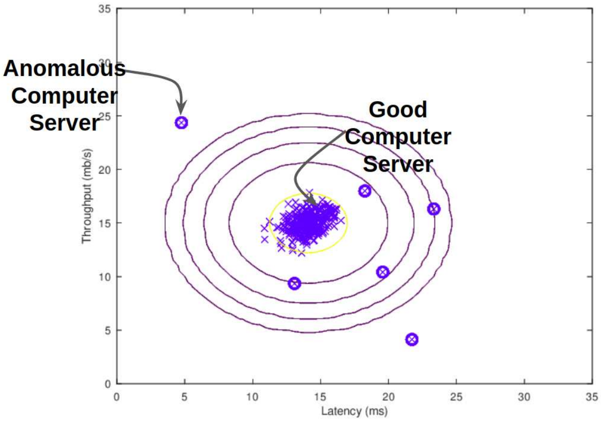

Figure 3: Detecting failing computer servers on a network

Here, precision, $P = \frac{tp}{tp+fp}$ and recall, $R = \frac{tp}{tp+fn}$. The tp,fp and fn are true positive, false positve and false negative respectively. True positive (tp): (i.e.,) when the actual label is anomalous but our gaussian model $p(X) < \epsilon$ predicts it correctly to be anomaly. False positve (fp): (i.e.,) when the actual label is anomalous but our gaussian model $p(X) < \epsilon$ predicts it falsely to be an anomaly. False negative (fn): (i.e.) when the actual label is anomalous but our gaussian model $p(X) < \epsilon$ predicts it falsely to be normal.

5 Contribution

My contribution to this project is writting a high level vectorized programming code in *Octave* to estimate the parameters for the gaussian model and to find the best evaluation metric.