Mahanalobis Outliers Detection: Jonathan Ndamba

jonathan.ndamba.pro@gmail.com

july 2019

Summary

1 Introduction **1**

2 Reminder **1**
2.1 Gaussian mixture 1
2.2 Expectation-Maximisation algorithm 2

3 Mahanalobis **5**
3.1 Mahanalobis Distance 5
3.2 Scoring 5

1 Introduction

The MDO package stands for Mahanalobis Detection Outliers, which enables the detection of abnormal values by measuring the distance between a point and a distribution. We will explain the mathematical background behind the use of this package. Firstly, we will do a brief reminder dealing with the Gaussian mixture and the Expectation-maximisation algorithm, and finally we will define the MDO (Mahanalobis Detection Outliers) and its hypothesis.

2 Reminder

2.1 Gaussian mixture

A gaussian mixture is a linear combination of gaussian with this constraints

- $\sum_{i=1}^{n} \pi_i N(\mu_i, \sigma_i)$
- $\sum_{i=1}^{n} \pi_i = 1$

The π_i are latent variables, which means they can't be directly observed and need to be inferred.

2.2 Expectation-Maximisation algorithm

The log-maximisation likelihood can be used in order to infer parameters, however it would mean to solve a difficult calculus which is to solve a summation in a logarithm.

$l_0(Z) = ln(\sum_{i=0}^{n} N(\mu_i, \sigma_i))$

In order to simplify the algorithm comprehension, we will use a mixture with only two components (this example comes from the book ESL [Element of Statistical Learning] by Trevor Hastie et al.).

We have :

- $y_1 \rightarrow N(\mu_1, \sigma_1)$
- $y_2 \rightarrow N(\mu_2, \sigma_2)$
- $y = (1 - \Delta) * y_1 + \Delta * y_2$ with $\Delta \in \{0, 1\}$ and $Pr(\Delta = 1) = \pi$
- $g(y) = (1 - \pi) * N(\mu_1, \sigma_1) + \pi * N(\mu_2, \sigma_2)$

Now we will assume that we know the value of Δ which is an other latent variable, then we obtain this form :

- $l_0(Z, \Delta) = (1 - \Delta)ln((1 - \pi) * N(\mu_1, \sigma_1)) + \Delta * ln(\pi * N(\mu_2, \sigma_2))$
- $l_0(Z, \Delta) = (1 - \Delta)[ln((1 - \pi) + ln(N(\mu_1, \sigma_1))] + \Delta[ln(\pi) + \ln(N(\mu_2, \sigma_2))]$

with $Z = \{\pi, (1 - \pi)\}$
and in case we know the value of Δ we have :

- Case $\Delta = 0$
  - $l_0(Z, \Delta = 0) = ln((1 - \pi) + ln(N(\mu_1, \sigma_1))$
- Case $\Delta = 1$
  - $l_0(Z, \Delta = 1) = ln(\pi) + ln(N(\mu_2, \sigma_2))$

with this form we can use the log-maximisation likelihood and determine μ_1 the sample mean and σ_1 the sample variance for those data with $\Delta = 0$ and μ_2 the sample mean and σ_2 the sample variance for those data with $\Delta = 1$. In real cases, we can't use Δ because we don't know the exact value so we use its expected value:

$\gamma_i = E(\Delta | \mu_i, \sigma_i, Z)$

which is also called the responsibility. We can execute the EM algorithm until convergence :

Algorithm 8.1 *EM Algorithm for Two-component Gaussian Mixture.*

1. Take initial guesses for the parameters $\hat{\mu}_1, \hat{\sigma}_1^2, \hat{\mu}_2, \hat{\sigma}_2^2, \hat{\pi}$ (see text).

2. *Expectation Step*: compute the responsibilities

$$\hat{\gamma}_i = \frac{\hat{\pi}\phi_{\hat{\theta}_2}(y_i)}{(1-\hat{\pi})\phi_{\hat{\theta}_1}(y_i) + \hat{\pi}\phi_{\hat{\theta}_2}(y_i)}, \; i = 1, 2, \ldots, N. \tag{8.42}$$

3. *Maximization Step*: compute the weighted means and variances:

$$\hat{\mu}_1 = \frac{\sum_{i=1}^N (1-\hat{\gamma}_i) y_i}{\sum_{i=1}^N (1-\hat{\gamma}_i)}, \qquad \hat{\sigma}_1^2 = \frac{\sum_{i=1}^N (1-\hat{\gamma}_i)(y_i - \hat{\mu}_1)^2}{\sum_{i=1}^N (1-\hat{\gamma}_i)},$$

$$\hat{\mu}_2 = \frac{\sum_{i=1}^N \hat{\gamma}_i y_i}{\sum_{i=1}^N \hat{\gamma}_i}, \qquad \hat{\sigma}_2^2 = \frac{\sum_{i=1}^N \hat{\gamma}_i (y_i - \hat{\mu}_2)^2}{\sum_{i=1}^N \hat{\gamma}_i},$$

   and the mixing probability $\hat{\pi} = \sum_{i=1}^N \hat{\gamma}_i / N$.

4. Iterate steps 2 and 3 until convergence.

Figure 1: EM algo From ESL

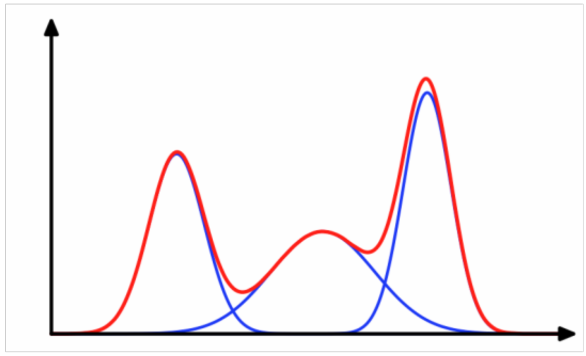

Figure 2: EM-Algo in the case of univariate

Figure 3: EM-Algo in the case of Multivariate

In the case of multivariate analysis our μ_i become vector of means and our σ_i become Covariance matrix (Σ_i).

3 Mahanalobis

3.1 Mahanalobis Distance

The Mahanalobis distance is a distance which "standardise" variables which means any variables dominate others. When a point is a large value then it's for the majority of variables, and not for only one variable dominate by scale. Here we use the Precision matrix the inverse of covariance matrix.

$D_i(x) = \sqrt{(x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i)}$
(the equivalent form for univariate $D_i(x) = \sqrt{(\frac{(x-\mu_i)}{\sigma_i})^2})$

each μ_i and Σ_i of each cluster are inferred by the EM algorithm.

3.2 Scoring

In our case we will be using the average of each Distance of Mahanalobis defined by each Cluster inferred by the EM Algorithm, this can be seen as a bagging method, each cluster have a partition of data (Boostrap) and so each cluster "vote" for an output by the average (like random forest).

$\sum_{i=0}^{n} D_i(x) = \sum_{i=0}^{n} \sqrt{(x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i)}$

In our case we added an assumption which is to add the π_i inferred on the EM algorithm it's equivalent to say "it is not as bad to be far from a small cluster as it is from a large cluster"
Then we use the weighted average :

scoring $= \sum_{i=0}^{n} \pi_i D_i(x) = \sum_{i=0}^{n} \pi_i \sqrt{(x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i)}$