Anomaly Detection and Recommender Systems

Machine learning

Soufiane Salama, michel banken

2019

Index

[Anomaly detection 2](#_Toc9425890)

[Exercise 2](#_Toc9425891)

[Part 2: Estimate the dataset statistics 2](#_Toc9425892)

[Part 3: Find outliers 2](#_Toc9425893)

[Part 4: Multidimensional Outliners 2](#_Toc9425894)

[Additional questions 3](#_Toc9425895)

[1. F1 score as a function of epsilon 3](#_Toc9425896)

[2. F1 score as a function of epsilon (selectThreshold2) 3](#_Toc9425897)

[3. Gscatter/Contour 4](#_Toc9425898)

[4. F1 score as a function of epsilon 4](#_Toc9425899)

[Recommender system 5](#_Toc9425900)

[Exercise 5](#_Toc9425901)

[Additional questions 5](#_Toc9425902)

[1. Top 10 recommendations 5](#_Toc9425903)

# Anomaly detection

## Exercise

In this exercise an anomaly detection algorithm is implemented to detect anomalous behavior in server computers. There are 307 examples (m) and the 2 features are the server throughput (mb/s) and latency (ms) of response. These examples are unlabeled, but there is the labeled (y=0/y=1) validation set that also exist of 307 examples.

### Part 2: Estimate the dataset statistics

In the estimateGaussion() function, the average and variance are calculates along next formulas:

Then we use the multiVariateGaussion() function to calculate the ‘p’ value of the training set using the previous calculated mu and sigma² along the next formula that returns a vector of 307 rows.

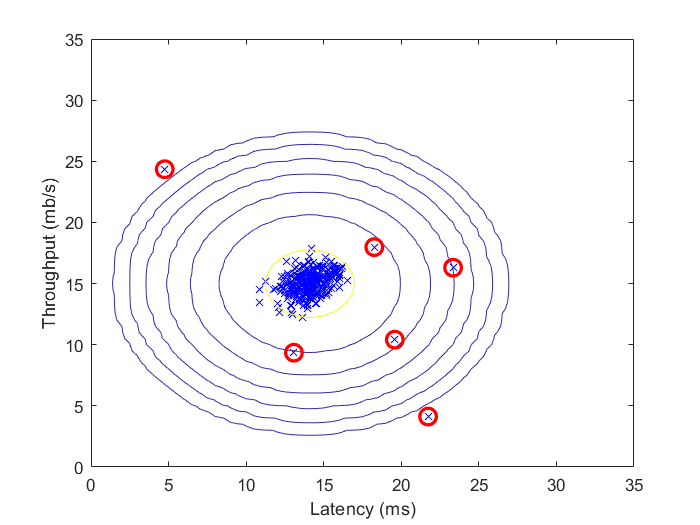
Using the visualizeFit() function we plot the dataset and its estimates distribution.

### Part 3: Find outliers

Instead of using the training set, the validation set is used to calculate the “p” values.

To find the outliers, a value for epsilon needs to be found that provides the highest F1 score. Concretely this is done by:

* going over each “p” value starting from min(p) to max(p) with a step of 8.9909e-05 -> **epsilon**
* checking if the “p” value matrix (307) is smaller then the current epsilon -> **predictions**
* calculating the F1 score based on the predictions, labeled data (yval) and the value for epsilon that provides the highest F1 score is returned
* all the F1 scores and epsilons are kept in a matrix for future plotting

And now we can find our outliers by comparing if the “p” values of the training set are lower than our highest epsilon value which indicates on an anomaly. This is done by Matlab function find().

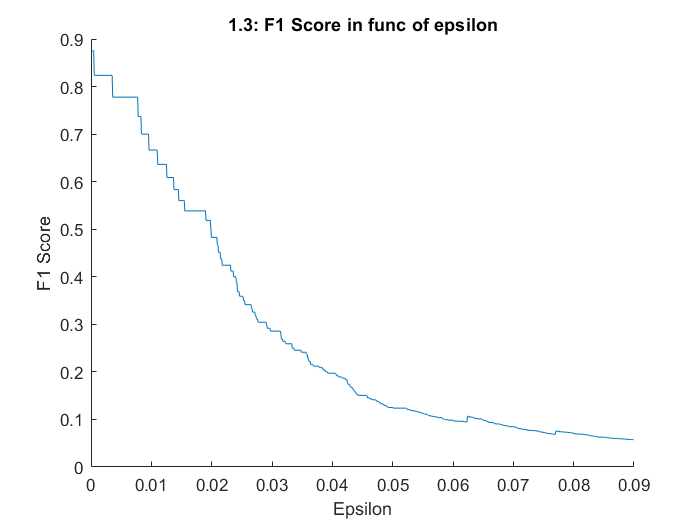
### Part 4: Multidimensional Outliners

Instead of using 2 features we are now going to use 11 features which capture many more properties of the servers. It’s very much the same as using 2 features, but now we aren’t able to plot the outliers.

First the values for mu and sigma² are calculated for the training set (11 features) by using the estimateGaussion() function, then the “p” values for the training as the validation set are calculated using the multiVariateGaussion() function. Finally the epsilon value is calculates based on the “p” value (validation set) and the labeled validation data (y).

## Additional questions

### F1 score as a function of epsilon



### F1 score as a function of epsilon (selectThreshold2)

In this plot we can see a difference with previous plot:

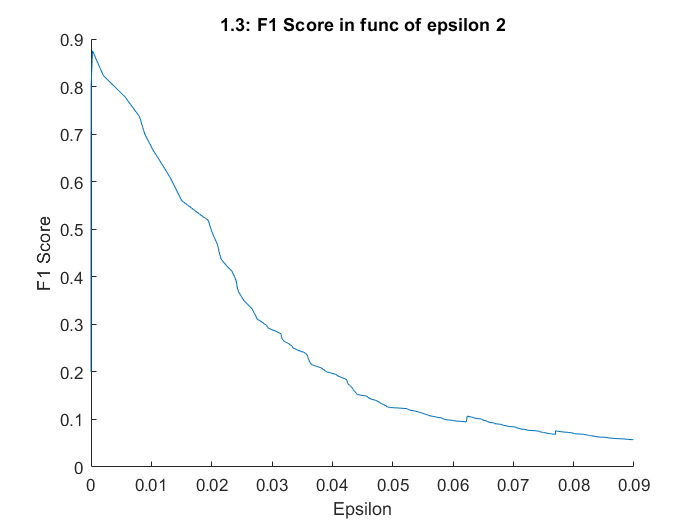
* the plot is smoother, this can be explained by the number of data used. In previous plot, selectThreshold() uses a stepsize of . But here we only use the average of two successful ‘p’ values -> **306** (307-1)
* it looks there are F1 scores for epsilon = 0, but these are just very small values for epsilon this is due by the calculation of the F1Score (precision of recall)



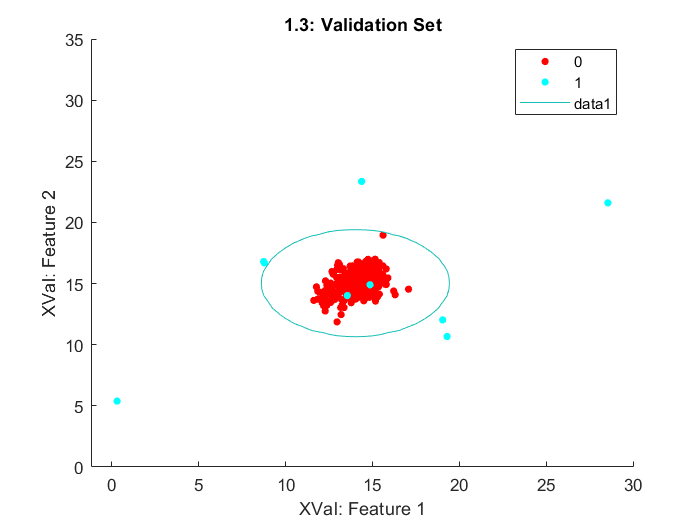
Begint laag, stijgt en zakt terug



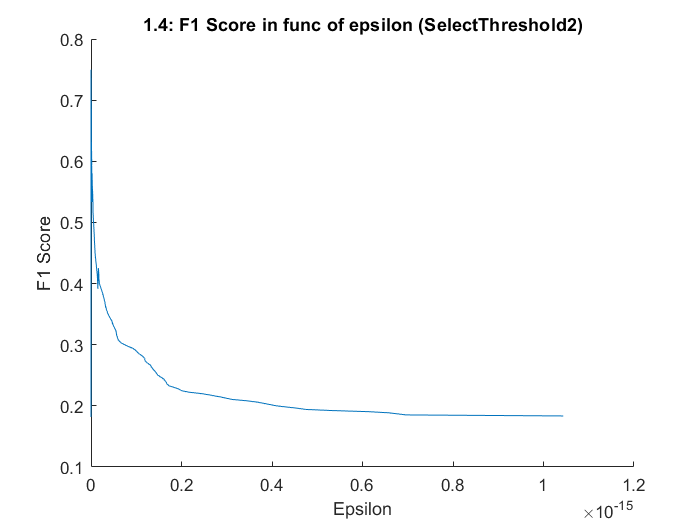
Begint hoog en zakt



### Gscatter/Contour



### F1 score as a function of epsilon

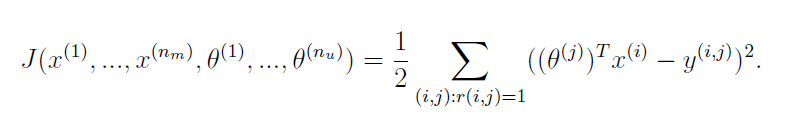


# Recommender system

## Exercise

In this exercise a collaborative filtering learning algorithm is implemented and applied to a dataset of movie ratings. The dataset consist of 943 users (n) and 1682 movies (m).

### Part 2: Cost function

The collaborative filtering cost function looks as follows:

Wat doet de reshape functie precies?

After implementing this cost function in the cosiCostFunc() function we get a cost of 22.22.

Note that we need to add the R parameter in the cost function, this is R value is equal to 1 (else 0) if user *j* has rated movie *i.*

### Part 3: Gradient

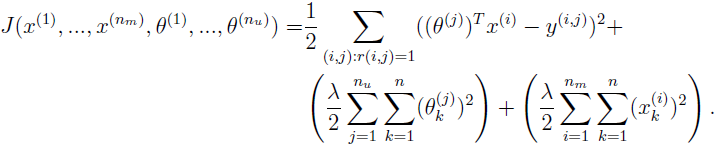
The collaborative filtering gradient function looks as follows:

Note that also here the R parameter needs to be implemented in the gradient functions.



### Part 4: Regularized cost function

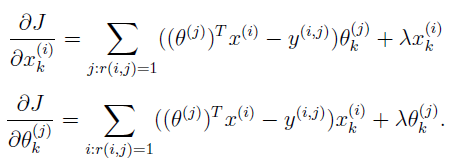
To implement regularization the cost function needs to be modified as follows:



Using regularization the cost increases to 31.34 using a lambda value of 1.5.

### Part 5: Regularized gradient

Also the gradient function needs to be modifies as follows:



### Part 6: Entering ratings for a new user

Gwn een aantal films raten.

### Part 7: Learning movie ratings

### Part 8: Recommendation for you

## Additional questions

### Top 10 recommendations

