



UNIVERSIDADE DE AVEIRO

DEPARTAMENTO DE ELECTRÓNICA, TELECOMUNICAÇÕES E
INFORMÁTICA

47064- DESEMPENHO E DIMENSIONAMENTO DE REDES

Network Statistical Analysis

8240 - MESTRADO INTEGRADO EM ENGENHARIA DE
COMPUTADORES E TELEMÁTICA

António Rafael da
Costa Ferreira
NMec: 67405

Rodrigo Lopes
da Cunha
NMec: 67800

Docentes: Paulo Salvador,
Susana Sargento

Abril de 2016
2015-2016

Conteúdos

1	Metrics	2
1.1	Exercício 1	2
1.2	Exercício 2 e Exercício 3	3
2	Probability Density Functions (PDF) and Cumulative Distribution Function (CDF)	4
2.1	Exercício 4	4
2.2	Exercício 5	6
2.3	Exercício 6	7
2.4	Exercício 7	8
3	Multivariate Distributions	9
3.1	Exercício 8	9
4	Aggregation Effect	10
4.1	Exercício 9	10
5	Events Correlation	11
5.1	Exercício 10	11
6	Periodicity	12
6.1	Exercício 11	12
6.2	Exercício 12	12
6.3	Exercício 13	13
7	Variable Reduction	15
7.1	Exercício 14	15
7.2	Exercício 15	16
7.3	Exercício 16	17
8	Anomaly Identification	18
8.1	Exercício 17	18
9	Single Distribution Models	19
9.1	Exercício 18	19
10	Machine State Modulated Distributions	20
10.1	Exercício 19	20
11	Trend/Growth Models	21
11.1	Exercício 20	21

1 Metrics

1.1 Exercício 1

codeFiles/metrics.py

Neste primeiro exercício, era pedido apenas para se analisar os tipos de perfis existentes. Os perfis que foram dados pelos professores no ficheiro *data1*, estão divididos em dois grupos, os primeiros 20 utilizadores com um tráfego mais periódico e os restantes com um tráfego mais irregular, tal como podemos verificar na imagem seguinte:

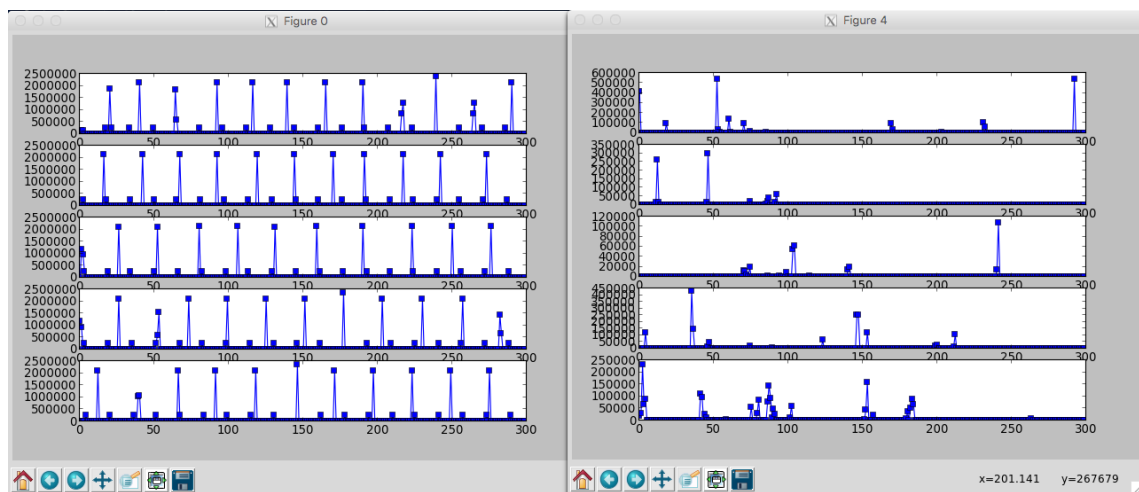


Figura 1: Tipos de Perfis dos utilizadores

Foi também pedido que se fizesse a representação do nosso perfil, obtido no guia passado, quando se receberam os pacotes do YouTube. O perfil que se obteve 2, enquadra num dos dois tipos de perfis acima mostrados.

É possível então verificar que o perfil obtido, se integra no grupo dos 20 primeiros perfis, tendo um tráfego mais periódico que os restantes.

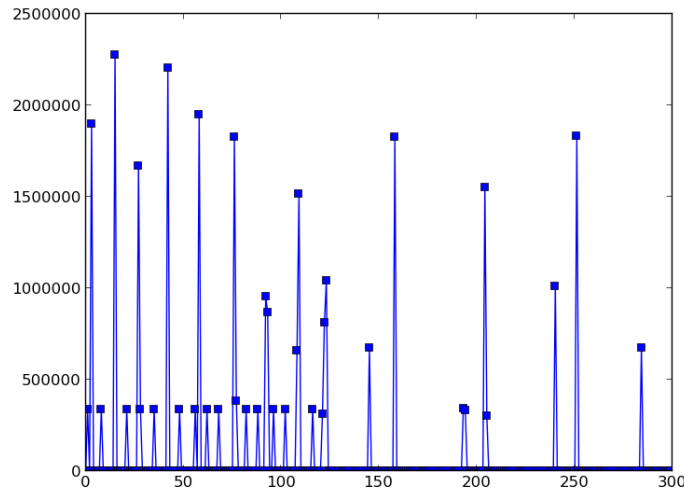


Figura 2: Perfil obtido

1.2 Exercício 2 e Exercício 3

codeFiles/metrics.py (linha 60)

No 2º exercício, foi pedido que se calculassem valores, como por exemplo a média, a variância, entre outros, para todos os perfis, incluindo o que foi obtido no guia anterior. Com estes valores pretende-se saber qual o tipo de perfil foi obtido. O ficheiro usado para o exercício 2 e 3 é o *metrics.py*.

É possível verificar que realmente o perfil obtido, se enquadra no conjunto dos 20 primeiros perfis do ficheiro *data1*, onde o burst de pacotes é mais periódico.

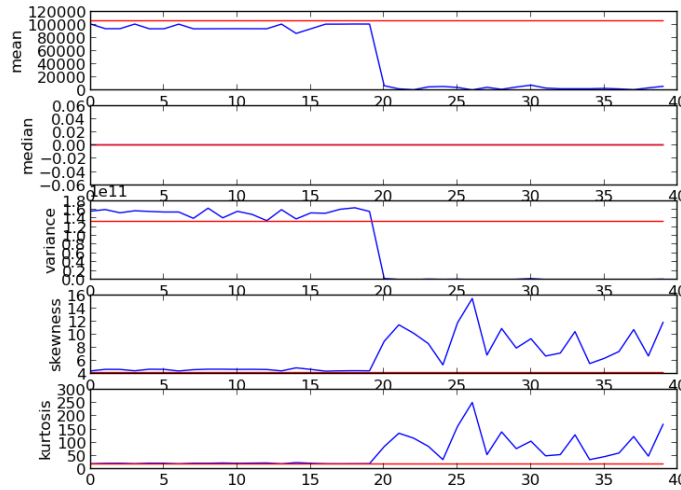


Figura 3: Comparação de Perfis

2 Probability Density Functions (PDF) and Cumulative Distribution Function (CDF)

2.1 Exercício 4

codeFiles/pdf_cdf.py (linha 41)

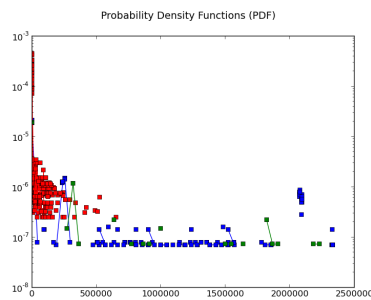


Figura 4: Probability Density Functions (PDF)

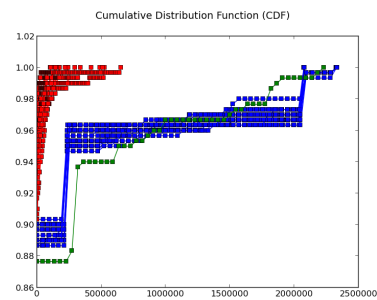


Figura 5: Cumulative Distribution Function (CDF)

Como se pode observar nas figuras 4 e 5, nos gráficos, as PDF dos serviços 0 - 19 são representados pela cor azul, e as 20 - 39 são representadas com a cor vermelha.

Os serviços 0 - 19 estão mais distribuídos pelo eixo dos XX, ou seja, existe mais probabilidade de existir a receção de pacotes com mais frequência.

Os serviços 20 - 39, já existe grande probabilidade de obter zero pacotes, devido aos longos períodos de tempo sem receção de pacotes.

Devido a esta situação, dos serviços 20-39, os valores de kurtosis e skewness são mais elevados, como se pode ver na figura 3.

2.2 Exercício 5

codeFiles/qq_pp_plot.py (linha 39)

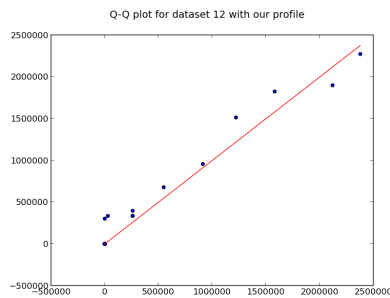


Figura 6: Q-Q plot do dataset 12

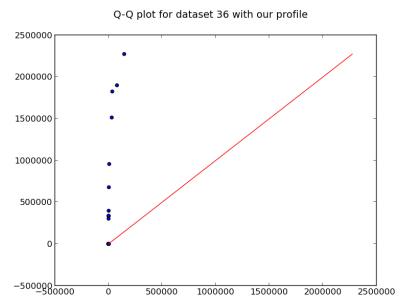


Figura 7: Q-Q plot do dataset 36

É possível ver que através da comparação entre o gráfico Q-Q obtido através do serviço 12 com o gráfico Q-Q obtido do serviço 36, o gráfico pertencente ao primeiro grupo (6) possui um número de pacotes recebidos maior. O perfil obtido através do YouTube, como já tinha sido verificado acima, enquadra-se neste grupo, e é possível observar esse acontecimento pelos pontos que se encontram muito próximos da linha vermelha, ao contrário do serviço 36.

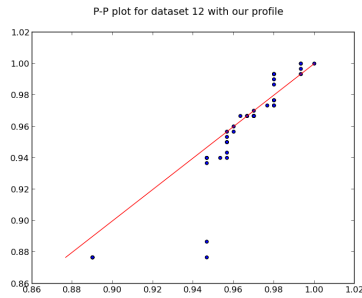


Figura 8: P-P plot do dataset 12

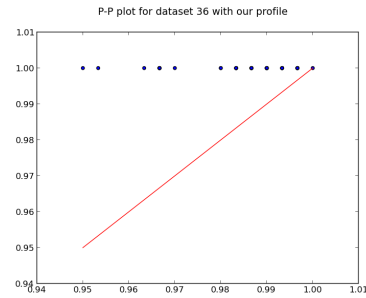


Figura 9: P-P plot do dataset 36

O P-P plot, consiste no confronto entre duas distribuições cumulativas (CDF's) de dois perfis, neste caso, entre o perfil do dataset 12 e 36 com o perfil do YouTube. É possível verificar que tal como nas CDF's dos serviços entre 20 e 39 (5), rapidamente ascende ao valor 1, enquanto que no caso dos serviços entre 0 e 19, onde o perfil YouTube se enquadra, os pontos acompanham a linha, com uma proximidade relativamente baixa, pelo que é possível verificar as conclusões retiradas acima, em relação ao tipo de perfil.

2.3 Exercício 6

codeFiles/pdf_cdf.py (linha 128)

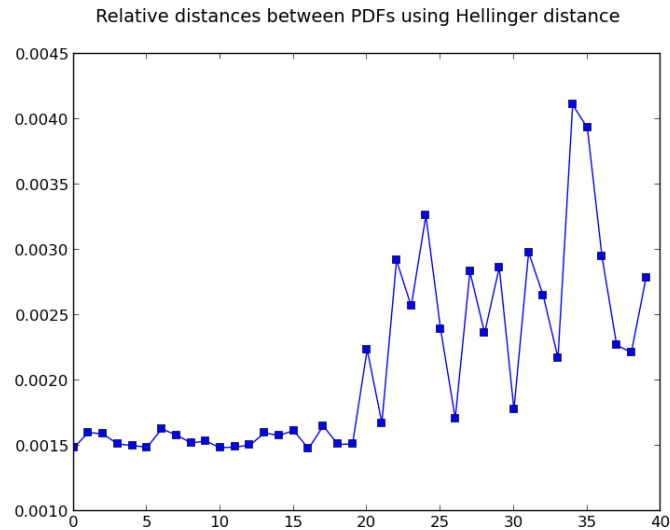


Figura 10: Distâncias relativas entre PDF's

Através do gráfico de distâncias relativas entre PDF's, utilizando a distância de Hellinger, é possível observar, que entre os primeiros datasets 0-19 e o dataset do YouTube, a distância é relativamente pequena. Já nos datasets 20-39, a distância sofre um aumento bastante significativo, pelo que é possível concluir mais uma vez, que o perfil de YouTube, se enquadra mais nos primeiros 20 datasets.

2.4 Exercício 7

codeFiles/pdf_cdf.py (linha 135)

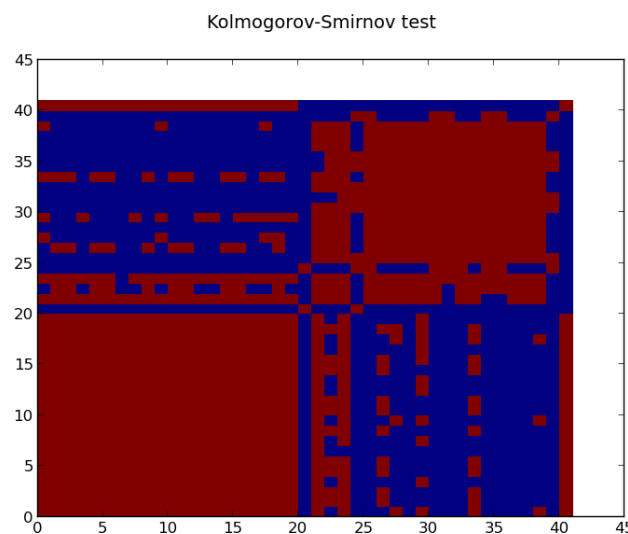


Figura 11: Teste de Kolmogorov-Smirnov

Neste teste, o objetivo era saber se dois perfis seguiam a mesma distribuição. Através do cálculo de um p -value, obtido através de dois datasets. Caso este valor, fosse inferior ao nível de significância (neste caso 5%), a hipótese de os dois perfis seguirem a mesma distribuição era rejeitada. É possível verificar na imagem acima (11) que nos datasets pertencentes ao mesmo grupo, a hipótese é aceite, ficando preenchido a vermelho. Quando pertencem a grupos diferentes, é possível verificar que a maioria das vezes a hipótese é rejeitada (ilustrado a azul).

O perfil obtido do YouTube, na imagem, corresponde ao dataset 40, pelo que é facilmente identificável, que este pertence ao primeiro grupo de perfis.

3 Multivariate Distributions

3.1 Exercício 8

codeFiles/pdf_cdf.py (linha 193)

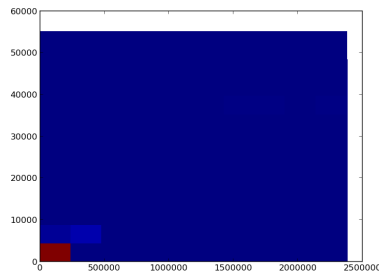


Figura 12: Histograma 2D, Dataset 9

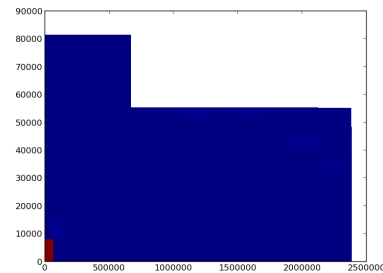


Figura 13: Histograma 2D, Dataset 30

Neste exercício foi pedido para obter as pdf's dos serviços, sendo desta vez utilizadas duas variáveis, upload e download.

Nos gráficos presentes nas figuras 12 e 13, são visíveis vários níveis de cor, sendo que os pontos vermelhos correspondem a um valor da probabilidade maior e nos pontos azuis, a probabilidade é mais pequena.

O gráfico obtido do Dataset 9, pertencente ao primeiro grupo de Datasets, permite verificar que junto ao ponto 0, a probabilidade é maior, isto porque, os pacotes são recebidos em bursts, e a maior parte das vezes, não é recebido qualquer pacote. Os bursts, são representados pelos quadrados azul mais claro.

Já no gráfico do Dataset 30, continua a existir uma maior probabilidade junto ao ponto 0, contudo mais próximo do zero do que o anterior e existem ainda menos quadrados azul claro.

Conclui-se que em ambos os serviços, a probabilidade é maior junto ao ponto (0,0). Verifica-se ainda que o upload é muito inferior ao download, e no segundo grupo de datasets é mais superior do que o do primeiro grupo.

4 Aggregation Effect

4.1 Exercício 9

codeFiles/pdf_cdf.py (linha 206)

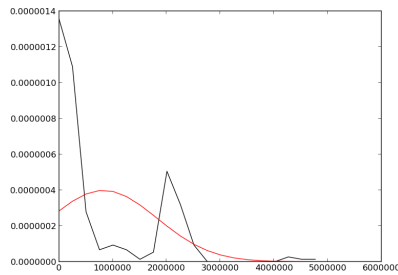


Figura 14: PDF 20 utilizadores

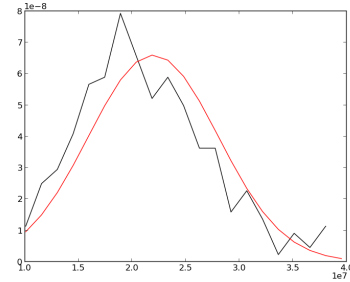


Figura 15: PDF 500 utilizadores

Com este exercício era pretendido verificar que quanto maior o número de utilizadores agregados, maior seria a aproximação à curva Gaussiana. Tal como as imagens (14, 15) demonstram, quando se tem apenas 20 utilizadores, não existe grande aproximação à curva Gaussiana. Quando se tem 500 utilizadores, a aproximação é bastante significativa, tal como pretendido.

5 Events Correlation

5.1 Exercício 10

codeFiles/pdf_cdf.py (linha 224)

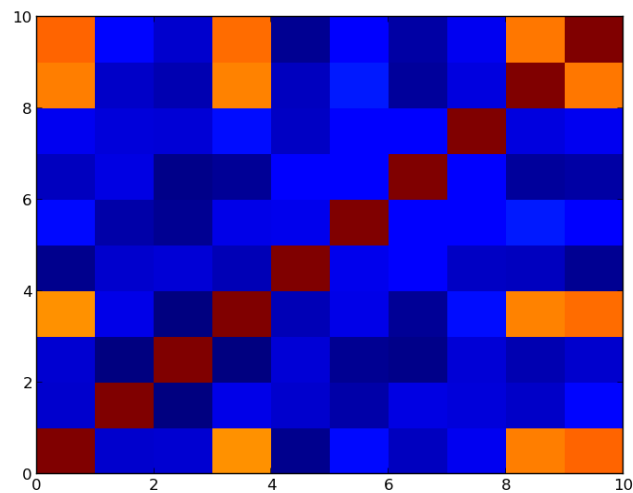


Figura 16: Correlação de eventos

A anomalia no tráfego existente no link 1 fez com que esta, se propagasse pelos links 4 (coluna 3), 9 (coluna 8) e 10 (coluna 9). Através da figura 16, podemos verificar que os links que se encontram a amarelo/laranja, são os que foram afetados pela anomalia do link 1.

6 Periodicity

6.1 Exercício 11

codeFiles/pdf_cdf.py (linha 236)

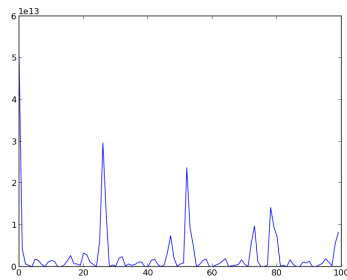


Figura 17: Periodicidade Dataset 3

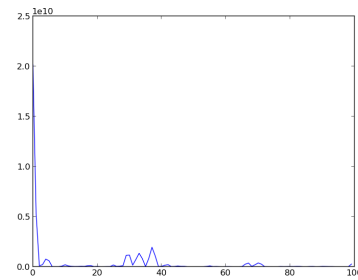


Figura 18: Periodicidade Dataset 22

Através das figuras 17 e 18, é possível verificar que no Dataset 3 existe periodicidade com o decorrer do tempo, sendo neste caso a periodicidade aproximadamente de 26. No outro caso, Dataset 22, verifica-se que não existe periodicidade.

6.2 Exercício 12

codeFiles/pdf_cdf.py (linha 258)

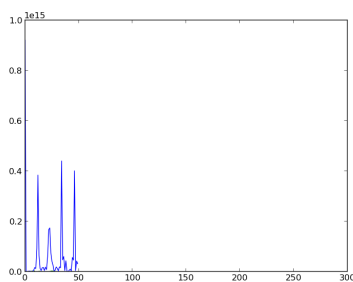


Figura 19: Periodogram Dataset 3

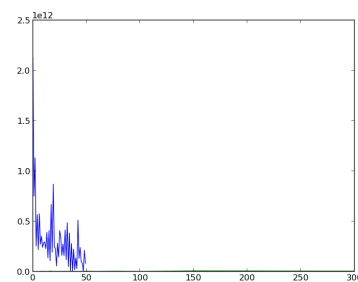


Figura 20: Periodogram Dataset 22

Através dos periodograms 19 e 20, é possível verificar o mesmo que no exercício anterior, existindo uma periodicidade nos datasets pertencentes ao primeiro grupo.

6.3 Exercício 13

codeFiles/pdf_cdf.py (linha 278)

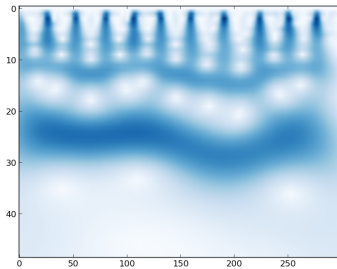


Figura 21: Scalogram FFT Dataset 2

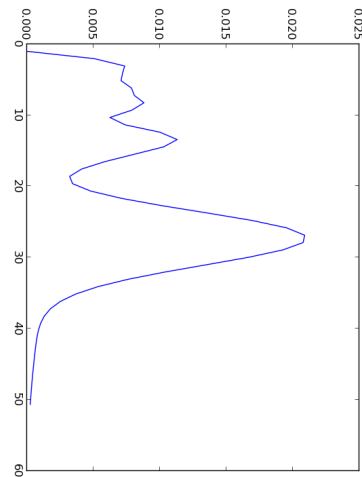


Figura 22: Scalogram Dataset 2

Nas imagens acima, foi feito o Scalogram para um dataset do primeiro grupo de serviços, sendo possível observar que nas zonas mais escuras é um pico de frequência com mais intensidade, podendo perceber-se facilmente a existência de periodicidade.

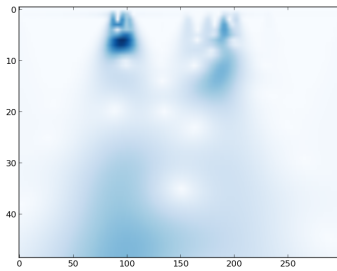


Figura 23: Scalogram FFT
Dataset 31

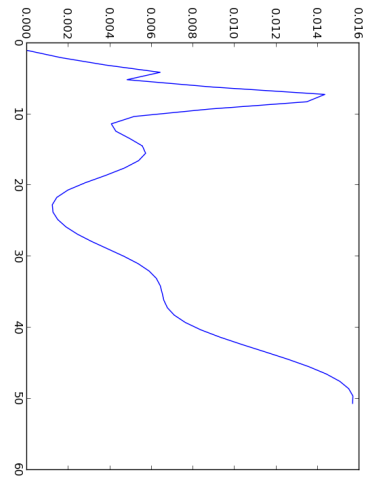


Figura 24: Scalogram Dataset
31

Posteriormente, obteve-se o Scalogram para um dataset do segundo grupo, e obteve-se um valor idêntico mas a periodicidade não é tão evidente como no exemplo anterior.

7 Variable Reduction

7.1 Exercício 14

codeFiles/classifier.py (linha 58)

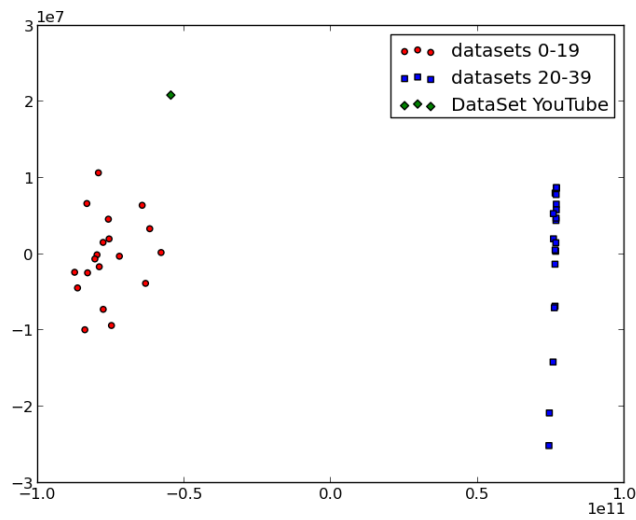


Figura 25: PCA

Utilizando PCA reduzimos variáveis como a média, a mediana, variância, skewness, kurtosis, entre outros, a componentes principais. É possível então através da imagem acima, que o perfil obtido do YouTube, encontra-se de acordo com os perfis do primeiro grupo de perfis. É então feito um fit, com todas as variáveis, seguindo de uma transformação, sendo a diferenciação de perfis, feita com as métricas transformadas e não com as variáveis de média, etc. Quando já existe uma transformação, todas as métricas que chegarem, podem ser transformadas sem ser feito o fit. A transformação origina pontos x , y e α_1 , α_2 , que são utilizados para preencher o gráfico.

7.2 Exercício 15

codeFiles/classifier.py (linha 106)

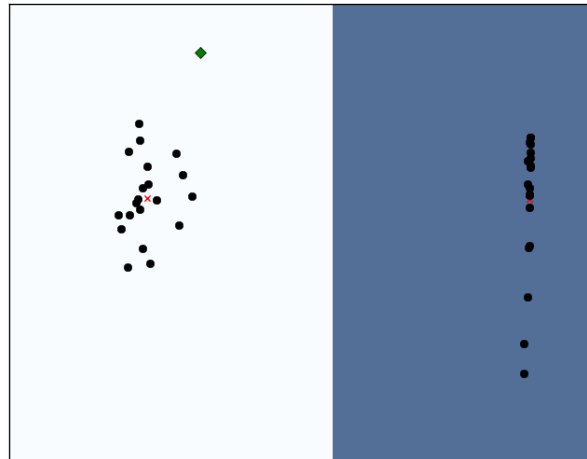


Figura 26: K-Means

Neste exercício, pretende-se diferenciar utilizadores, desta feita, utilizando o método K-means, que requer um conhecimento à priori do número de clusters existentes. É então feito o fit das componentes, que serve para definir/criar grupos de utilizadores. Posto isto é feito o predict, que vai receber os dados e classificá-los. É possível ver na imagem 26, um gráfico de pontos idêntico ao PCA, sendo que neste a divisão dos grupos é mais clara, sendo feita através das cores branco e azul, sendo os pontos da parte branca pertencentes aos datasets 0-19 e os da zona azul pertencentes aos datasets 20-39.

O perfil do YouTube, como esperado encontra-se junto aos pontos do primeiro grupo de utilizadores.

7.3 Exercício 16

codeFiles/classifier.py (linha 157)

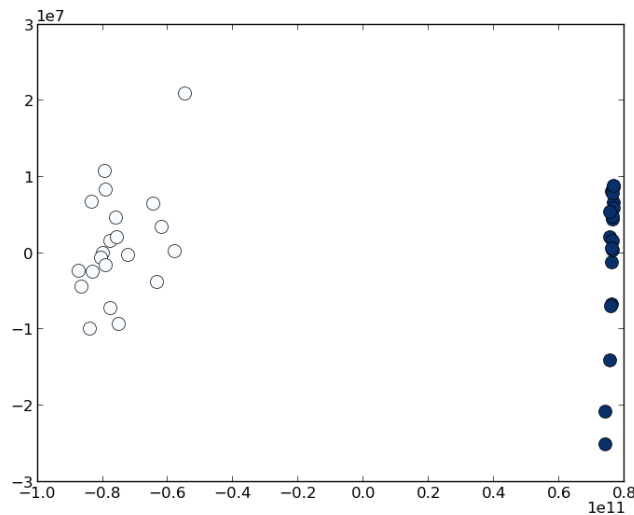


Figura 27: DBSCAN

Um outro método para fazer a diferenciação de utilizadores é o DBSCAN, neste não existe número de clusters. Este método baseia-se nas distâncias relativas, sendo com isto necessário correr sempre o fit, quando recebe algo novo.

O perfil do YouTube, foi colocado no grupo dos datasets 0-19, contudo, o esperado, segundo o professor seria a criação de um novo grupo, onde estaria o perfil, mas esta criação de grupo depende da largura de banda com que se obteve os dados, pelo que no nosso caso, não houve a criação de um grupo.

8 Anomaly Identification

8.1 Exercício 17

codeFiles/classifier.py (linha 177)

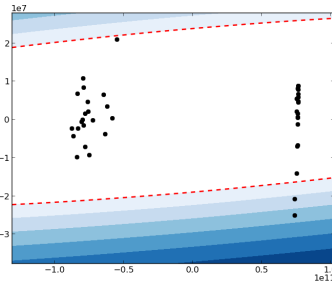


Figura 28: Assumindo que existem 4.5% de anomalias

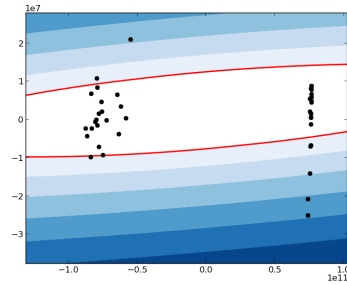


Figura 29: Assumindo que existem 20% de anomalias

É possível através das figuras 28 e 29 verificar num conjunto de utilizadores, os que são considerados anomalias. Para isso, são definidas fronteiras, sendo que tudo o que se encontre fora das mesmas, é considerado anomalia. Neste caso, na figura 28, assumiu-se uma percentagem de anomalias de 4.5%, tornando o intervalo da fronteira maior, sendo que serão menos os casos de anomalia. Na figura 29, assumiu-se uma percentagem de 20% de anomalias, pelo que o intervalo é mais reduzido, e são então, detetadas mais anomalias.

Contudo, esta experiência, deveria ser feita em grupos, de forma a detetar as anomalias dentro de um determinado grupo, e não no conjunto total de utilizadores, onde os perfis dos utilizadores do primeiro grupo, são anomalias do segundo grupo, e vice-versa.

9 Single Distribution Models

9.1 Exercício 18

codeFiles/distribution.py (linha 19)

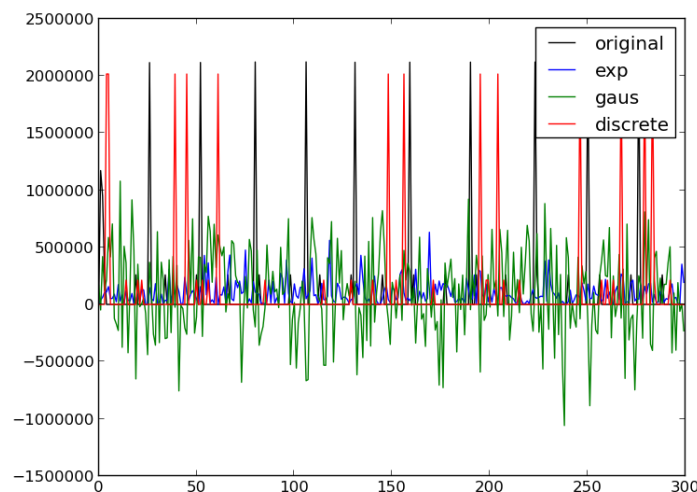


Figura 30: Distribuição original, exponencial, gaussiana e discreta

Neste exercício, era pedido, para através de um perfil qualquer, tentar arranjar uma distribuição que descrevesse o comportamento do mesmo. Era pedido que se ajustasse a distribuição de forma exponencial, normal e discreta, de forma a se comparar com as PDF's do perfil original.

Para gerar a distribuição de forma exponencial apenas era necessária a média dos valores da distribuição original. No caso de gerar uma distribuição normal/gaussiana, seria necessária, para além da média dos valores, a variância.

Neste exercício, apenas se olha para quantidade e não para tempo, pelo que não é possível ter uma noção exata de periodicidade, existe uma discrepância na mesma.

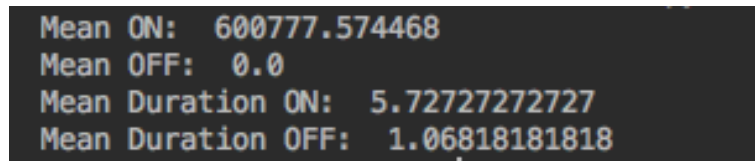
10 Machine State Modulated Distributions

10.1 Exercício 19

codeFiles/distribution.py (linha 47)

No exercício 19, era pedido que se criasse um gerador de perfis através de um gráfico ON/OFF, que consiste numa máquina de dois estados, que saltam de um para o outro. Quando se encontra no estado ON, gera pacotes com uma taxa λ .

É definida uma fronteira, que no neste caso foi com o valor de 1, em que tudo o que se encontra acima encontra-se no estado ON, o resto encontra-se no estado OFF.



```
Mean ON: 600777.574468
Mean OFF: 0.0
Mean Duration ON: 5.727272727
Mean Duration OFF: 1.068181818
```

Figura 31: Médias ON/OFF e Médias Duração ON/OFF

O perfil que se estudou, para uma fronteira com o valor de 1, tinha de médias, como se pode ver na figura acima, 600777.574468 para o estado ON e 0.0 para o estado OFF, o que é o esperado, pois o estado OFF é quando não gera praticamente pacotes nenhuns.

Em relação ao tempo médio que se encontra em cada estado, é possível verificar que se encontra mais no estado OFF, com uma média de 5.727, do que no estado ON, que tem uma média de 1.06818.

Depois da obtenção das médias, gerou-se um novo perfil que seguiria a mesma distribuição, sendo possível verificar, através do gráfico apresentado na figura 32, que o objetivo foi conseguido, pelo que o novo perfil se enquadra na distribuição desejada, com picos como o original, mas na maioria das vezes com pouca produção de pacotes.

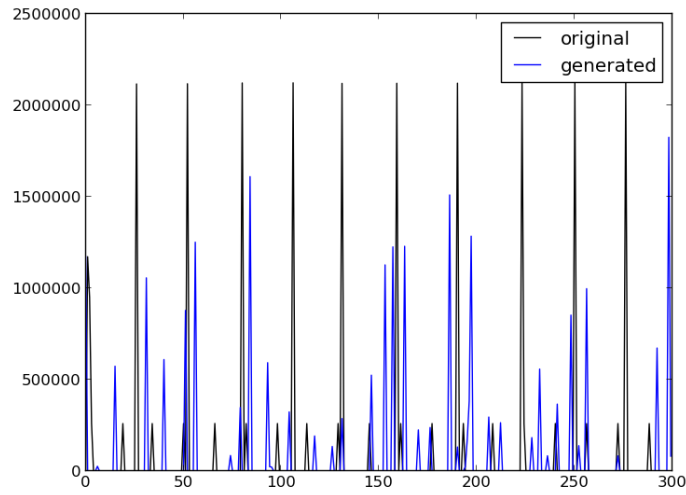


Figura 32: Distribuição do perfil original e do gerado

11 Trend/Growth Models

11.1 Exercício 20

codeFiles/distribution.py (linha 103)

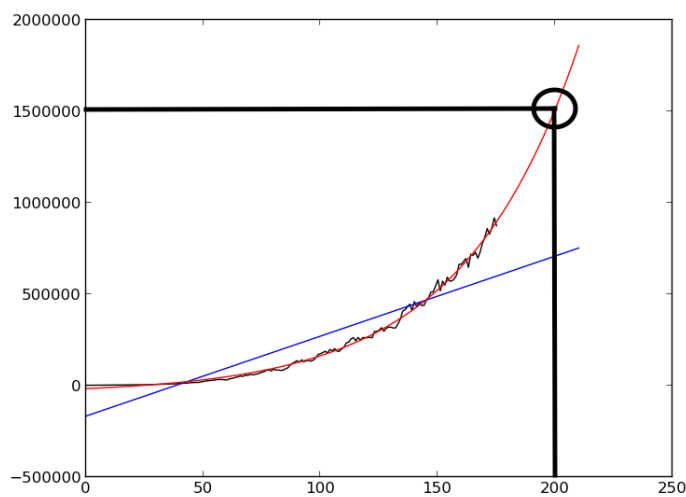


Figura 33: Distribuição original, exponencial, gaussiana e discreta

O objetivo deste último exercício, era prever o tempo que demorará o tráfego de entrada no IX em Amsterdão a atingir 1.5 ExaBytes. Sabendo, que no momento da captura que permitiu a elaboração do gráfico, o tempo ia em 180 meses (fim da linha preta), através do gráfico é possível ver que o valor de tráfego será atingido no mês 200, pelo que em 20 meses o tráfego de entrada, já terá atingido valores nos 1.5 ExaBytes.

É possível ainda verificar através da curva a vermelho, que cada vez mais a subida de valores no tráfego é mais rápida pelo que demorará menos tempo a atingir valores ainda mais elevados.

A curva a vermelho, foi encontrada com um ajuste exponencial, de forma a se aproximar com a realidade, e é possível verificar que está enquadrada com a curva real (linha preta).