TÉCNICO LISBOA

Engenharia de Tráfego
2018-2019

Mestrado em Engenharia Telecomunicações e Informática

Relatório

Automatic Classification

quarta-feira, 12 de dezembro de 2018

Grupo 5
João Correia - 81990
João Costa - 82528

Introdução

Este laboratório tem como objetivo criar e testar um algoritmo de aprendizagem automática para identificar fluxos *Peer-to-Peer* num *link* de uma rede. Desta maneira, um gestor de redes é capaz de prever o uso "injusto" da largura de banda por parte de um endereço IP.

Este algoritmo baseia-se num classificador *Naive Bayes*, o qual considera que os acontecimentos são todos independentes entre si.

São testados 4 cenários diferentes (**1-labeled.dat**, **2-labeled.dat**, **3-labeled.dat** e **4-labeled.dat**), onde cada cenário possui 10 000 amostras de fluxos entre dois pontos da rede. Cada amostra possui cinco parâmetros:

- Endereço da rede de origem dos pacotes
- Número de conexões simultâneas por endereço IP
- Largura de banda média usada por cada IP nos últimos 60 segundos
- Tamanho médio dos pacotes de todo o tráfego proveniente de um endereço IP
- Hora do dia a que foi registado estes parâmetros

Para estes 4 cenários é também indicado se o fluxo de cada amostra é ou não *Peer-to-Peer*.

Com base nestas informações é feita uma aprendizagem tendo como base uma percentagem de amostras aleatórias, sendo de seguida testado o algoritmo com as restantes amostras.

Estrutura do Código

Para implementar este algoritmo foi usada a linguagem de programação *Python3* uma vez estarmos mais familiarizados com esta, tal como para tirar proveito da estrutura de dados. O ficheiro onde o algoritmo se encontra é o **mainFunction.py**.

Começámos por criar uma classe cujos atributos são:

- Número de fluxos de um determinado tipo ($P2P$ ou $\overline{P2P}$) [**class.number**]
- Dicionário com objetivo de contar o número de vezes que um parâmetro foi visto num determinado tipo de fluxo [**class.dictionaryValues**]
- Dicionário com as probabilidades de cada parâmetro sabendo que o fluxo é $P2P$ ou $\overline{P2P}$ [**class.dictionaryProbabilities**]

Dadas duas instâncias da classe, uma para os fluxos $P2P$ [**peer**] e outra para os fluxos $\overline{P2P}$ [**notPeer**], procedeu-se ao processo de aprendizagem tendo como base uma percentagem do conjunto de amostras aleatórias [**learningProcess()**].

Para obter as amostras aleatórias, foi usada a função **getVectorLines()** que retorna dois vetores, um com as posições das amostras a serem usadas para o processo de aprendizagem, e outro com as posições das amostras a serem usadas para teste. Esta função calcula o número de amostras a serem usadas para aprendizagem com base na percentagem de amostras previamente estabelecida.

Depois do processo de aprendizagem, são calculadas as probabilidades de um determinado parâmetro ser do tipo Y sabendo que o fluxo é do tipo $P2P$ ou $\overline{P2P}$ [**class.createProbabilities()**].

Por fim é efetuado o teste de avaliação do sistema tendo como base as restantes amostras (as não utilizadas no processo de aprendizagem) [**testProcess()**]. Neste teste são calculadas as probabilidades condicionadas do fluxo ser do tipo $P2P$ ou $\overline{P2P}$ sabendo que possuem determinados parâmetros.

Estas probabilidades são calculadas através das seguintes fórmulas, onde X é o conjunto de parâmetros de um determinado fluxo:

$$P(P2P \mid X) = [P(Address \mid P2P) * P(Connection \mid P2P) * P(Bandwidth \mid P2P) * P(PacketSize \mid P2P) * P(Time \mid P2P)] * P(P2P)$$

$$P(\overline{P2P} \mid X) = [P(Address \mid \overline{P2P}) * P(Connection \mid \overline{P2P}) * P(Bandwidth \mid \overline{P2P}) * P(PacketSize \mid \overline{P2P}) * P(Time \mid \overline{P2P})] * P(\overline{P2P})$$

A classificação do fluxo tem em consideração a maior probabilidade entre as duas indicadas em cima, sendo se seguida comparada a solução.

Após esta comparação, é contabilizado o número de *truePositives, trueNegatives, falsePositives* e *falseNegatives*, sendo respetivamente o número de $P2P$ e $\overline{P2P}$ corretamente identificados, o número de *flows* $\overline{P2P}$ mas identificados como $P2P$ e por fim o número de *flows* $P2P$ mas identificado como $\overline{P2P}$.

Depois de obter estes valores, é calculada a performance do sistema na função **verifyPerformance()** analisando a *Accuracy, Error Rate*, *Precision, Recall, True Negative Rate* e *F-measure.*

Para uma melhor contextualização destas métricas, a *Accuracy* e o *Error Rate* são a percentagem de *flows* corretamente e incorretamente identificados, respetivamente. Estas métricas servem para saber quão fiável um sistema pode ser, contudo não são suficientes para indicar se estamos perante o uso indevido de largura de banda.

A *Precision* indica-nos a percentagem de fluxos que foram corretamente identificados do conjunto $P2P$ previsto pelo sistema, ou seja, de todos os fluxos $P2P$ identificados pela máquina, a precisão é a percentagem de amostras que eram realmente $P2P$.

A *Recall* e o *True Negative Rate* referem-se à percentagem de amostras corretamente identificadas para os fluxos $P2P$ e $\overline{P2P}$ respetivamente, ou seja, é a *Accuracy* individual para cada tipo de *flow*. Estas métricas são boas para indicar o quão capaz foi o sistema de identificar corretamente fluxos do tipo $P2P$ e $\overline{P2P}$.

Por fim, e para balancear a *Precision* e o *Recall*, é usada a métrica *F-measure*. Assim podemos saber se o nosso sistema identifica corretamente fluxos $P2P$ tendo sempre em consideração os erros (*False Positive* e *False Negative*).

Na Figura 1 pode-se ver a relação entre algumas métricas e os classificadores binários, *truePositives, trueNegatives, falsePositives* e *falseNegatives.*

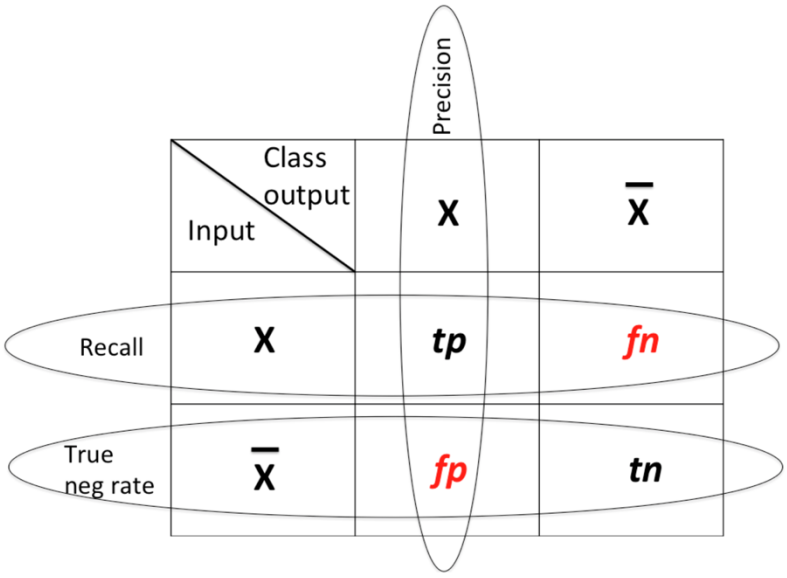

Figure 1 - Relação entre algumas métricas e classificadores binários.

Análise de Resultados

Depois de implementar todo o algoritmo referido na secção anterior foi feita a análise da performance do sistema nos 4 casos de estudo (**1-labeled.dat**, **2-labeled.dat**, **3-labeled.dat** e **4-labeled.dat**).

De modo a tirar conclusões mais precisas, repetimos as simulações 50 vezes, através do ficheiro **testPerformance.py**, e calculámos a média dos resultados. As métricas nas tabelas em baixo foram calculadas com os valores médios das 50 repetições para cada experiência.

Para perceber a variação da performance do sistema foram efetuados diversos testes, sendo que a diferença entre eles limita-se somente ao número de amostras usadas para aprendizagem tal como para teste do sistema. As percentagens das amostras utilizadas para o processo de aprendizagem foram de 90%, 75%, 50%, 25% ,10% em relação ao número total de amostras.

Nas Tabelas 1, 3, 5 e 7 encontram-se o número médio de *True Positives*, *True Negatives*, *False Positives* e *False Negatives* para os ficheiros 1-labeled.dat, 2-labeled.dat, 3-labeled.dat e 4-labeled.dat, respetivamente. Nas Tabelas 2, 4, 6 e 8 encontram-se os valores para a *Accuracy*, *Error Rate, Precision, Recall, True Negative Rate* e *F-measure,* também para os 4 ficheiros *labeled*.

1º Caso de Estudo

Aprendizagem (%)	*True Positive*	*True Negative*	*False Positive*	*False Negative*
10	131.42	6702.70	58.60	2107.28
25	46.24	5619.00	13.96	1820.80
50	6.12	3752.08	1.90	1239.90
75	1.16	1876.62	0.56	621.66
90	0.10	750.12	0.02	249.76

Tabela 1 – Média do número de True Positives, True Negatives, False Positives e False Negatives para o 1º caso de estudo

Aprendizagem (%)	*Accuracy*	*Error Rate*	*Precision*	*Recall*	*True Negative Rate*	*F-measure*
10	0.75935	0.24065	0.69161	0.05870	0.99133	0.10822
25	0.75537	0.24463	0.76811	0.02477	0.99752	0.04799
50	0.75164	0.24836	0.76309	0.00491	0.99949	0.00976
75	0.75111	0.24889	0.67442	0.00186	0.99970	0.00371
90	0.75022	0.24978	0.83333	0.00040	0.99997	0.00080

Tabela 2 – Valores para Accuracy, Error Rate, Precision, Recall, True Negative Rate e F-measure do 1º caso de estudo

Com a observação da Tabelas 2, podemos ver que existe um aumento considerável da *Precision* em função da percentagem de aprendizagem. Isto deve-se ao facto de haver um maior conjunto de amostras para o sistema aprender, no entanto, uma percentagem grande de aprendizagem leva a que o sistema tenha poucos testes para fazer, podendo levar deste modo a uma errada classificação do mesmo.

Da análise dos valores do *Recall* e do *True Negative Rate,* podemos concluir que o sistema aprende melhor fluxos do tipo $\overline{P2P}$ do que do tipo $P2P$ (dado este conjunto de amostras), uma vez que com o aumento da percentagem de amostras de aprendizagem, os valores do *True Negative Rate* tendem a aumentar, algo que não acontece com o *Recall*. Isto pode ser justificado pela variedade de fluxos $\overline{P2P}$ e pelo facto de o sistema tratar de cada parâmetro dos fluxos individualmente.

Era também de esperar que a *Accuracy* aumentasse com o acréscimo da percentagem de aprendizagem, uma vez termos um conjunto de amostras mais amplo para o sistema aprender.

Apesar do sistema melhorar a sua *Precision*, devido a uma melhor aprendizagem dos fluxos $\overline{P2P}$, a percentagem de fluxos $P2P$ devidamente identificados é muito baixa, tornando deste modo o sistema pouco credível para identificar fluxos $P2P$ (baixo valor do F-measure).

2º Caso de Estudo

Aprendizagem (%)	*True Positive*	*True Negative*	*False Positive*	*False Negative*
10	399.80	7817.38	18.62	764.20
25	337.34	6514.72	14.62	633.32
50	233.34	4339.18	10.50	416.98
75	123.80	2172.76	5.62	197.82
90	53.16	869.42	2.40	75.02

Tabela 3 - Média do número de True Positives, True Negatives, False Positives e False Negatives para o 2º caso de estudo

Aprendizagem (%)	*Accuracy*	*Error Rate*	*Precision*	*Recall*	*True Negative Rate*	*F-measure*
10	0.91302	0.08698	0.95550	0.34347	0.99762	0.50530
25	0.91361	0.08640	0.95846	0.34754	0.99776	0.51011
50	0.91450	0.08550	0.95694	0.35881	0.99759	0.52192
75	0.91862	0.08138	0.95658	0.38493	0.99742	0.54895
90	0.92258	0.07742	0.95680	0.41473	0.99725	0.57864

Tabela 4 – Valores para Accuracy, Error Rate, Precision, Recall, True Negative Rate e F-measure do 2º caso de estudo

Para o 2º caso de estudo, podemos ver que a *Accuracy* deste sistema é superior ao do 1º caso, tal como este valor tende a aumentar com o número de amostras utilizadas para aprender.

É também de referir que tanto a *Precision* como o *True Negative Rate* encontram-se praticamente constantes para todas as percentagens de aprendizagem, contudo o *Recall* tende a aumentar. Isto pode ser explicado por um decréscimo do número de falsos negativos.

3º Caso de Estudo

Aprendizagem (%)	*True Positive*	*True Negative*	*False Positive*	*False Negative*
10	607.86	8234.16	0	157.98
25	593.78	6863.96	0	42.26
50	417.66	4576.4	0	5.94
75	212.70	2286.92	0	0.38
90	83.48	916.38	0	0.04

Tabela 5 - Média do número de True Positives, True Negatives, False Positives e False Negatives para o 3º caso de estudo

Aprendizagem (%)	*Accuracy*	*Error Rate*	*Precision*	*Recall*	*True Negative Rate*	*F-measure*
10	0.98245	0.01755	1	0.79372	1	0.88500
25	0.99437	0.00563	1	0.93356	1	0.96564
50	0.99881	0.00119	1	0.98598	1	0.99294
75	0.99985	0.00015	1	0.99822	1	0.99911
90	0.99996	0.00004	1	0.99952	1	0.99976

Tabela 6 – Valores para Accuracy, Error Rate, Precision, Recall, True Negative Rate e F-measure do 3º caso de estudo

Para o 3º caso de estudo, podemos ver que a *Accuracy* tende a aumentar tal como a *Precision* e o *True Negative Rate* tem valor 1. Isto é justificado por não haver falsos positivos, indicando que os únicos erros dados pelo sistema são somente os falsos negativos.

Esta seria uma das soluções ideias, ensinar um sistema de maneira a que este não desse nenhum tipo de erro (nem falsos negativos nem falsos positivos), e pela Tabela 5 podemos ver que os valores dos erros são bastante reduzidos comparativamente com os valores “verdadeiros”.

É também importante salientar que apenas 50% de aprendizagem neste sistema chegaria para o sistema ficar treinado, deixando as restantes amostras para teste.

4º Caso de Estudo

Aprendizagem (%)	*True Positive*	*True Negative*	*False Positive*	*False Negative*
10	193.12	7517.46	19.90	1269.52
25	93.66	6268.86	8.42	1129.06
50	29.36	4181.30	2.18	787.16
75	9.54	2093.64	1.10	395.72
90	2.34	835.98	0.24	161.44

Tabela 7 - Média do número de True Positives, True Negatives, False Positives e False Negatives para o 4º caso de estudo

Aprendizagem (%)	*Accuracy*	*Error Rate*	*Precision*	*Recall*	*True Negative Rate*	*F-measure*
10	0.85673	0.14327	0.90658	0.13204	0.99736	0.23050
25	0.84834	0.15166	0.91752	0.07660	0.99866	0.14139
50	0.84213	0.15787	0.93088	0.03596	0.99948	0.06924
75	0.85127	0.15873	0.89662	0.02354	0.99947	0.04588
90	0.83832	0.16168	0.90698	0.01429	0.99971	0.02813

Tabela 8 – Valores para Accuracy, Error Rate, Precision, Recall, True Negative Rate e F-measure do 4º caso de estudo

Por fim, no 4º caso de estudo, podemos ver uma vez mais uma má *performance* de aprendizagem, uma vez que o valor do *Error Rate* tende a aumentar com o aumento do número de amostras utilizadas pelo sistema para aprender, tal como o valor do *F-measure* tende para 0.

Apesar do sistema ter uma boa precisão relativamente aos fluxos do tipo $P2P$, ou seja, dá poucos falsos positivos, o sistema dá muitos falsos negativos tornando-se assim pouco confiável.

Conclusão

Em suma, podemos verificar que para o primeiro e último caso, o sistema treinou erradamente pois a percentagem de erros tende a aumentar com o acréscimo de amostras de treino. Em contrapartida, no segundo e terceiro caso conseguimos ver uma boa performance pois os números de falso positivos e falsos negativos tende a ser cada vez mais baixo comparativamente com o número de verdadeiros positivos e negativos.

É de referir que o terceiro caso de estudo teve bastante sucesso, sendo que com 25% de amostras de aprendizagem, o sistema já era capaz de ter uma *performance* bastante boa para identificar fluxos $P2P$, cerca de 95%.

O número de falsos negativos e falsos positivos é sempre bastante superior ao número de verdadeiros positivos e verdadeiros negativos, deve-se ao facto dos ficheiros *labeled* conterem mais amostras do tipo **not p2p** do que do tipo **p2p**.

Com este trabalho concluímos que as percentagens de erro e de exatidão não são suficientes para classificar um sistema de aprendizagem binário, sendo sempre necessário o uso da precisão, do *Recall* ou de ambos, dependendo do tipo de sistema.

Os ficheiros **out-1-unlabeled.dat**, **out-2-unlabeled.dat**, **out-3-unlabeled.dat** e **out-4-unlabeled.dat** encontram-se na paste ***Results***, e contêm as decisões dadas pelo sistema para cada fluxo do *trace* tendo sido dado como ficheiro de aprendizagem os ficheiros **1-labeled.dat**, **2-labeled.dat**, **3-labeled.dat** e **4-labeled.dat**, respetivamente. Para a criação destes ficheiros de *output* foi usado o ficheiro **createFunction.py**.