Projeto de Prospeção de Dados

Pedro Oliveira
Faculdade de Ciências, Universidade de Lisboa
fc52754@alunos.fc.ul.pt

Rodrigo Basílio Ferreira
Faculdade de Ciências, Universidade de Lisboa
fc51032@alunos.fc.ul.pt

Rui Roque
Faculdade de Ciências, Universidade de Lisboa
fc57588@alunos.fc.ul.pt

1 Introdução

Neste projeto são analisados dois datasets, um deles contém as impressões digitais de 1101 moléculas (FPS Dataset), sendo que estas impressões consistem em 2048 atributos binários (0 se a característica não estiver presente e 1 se estiver), e o outro com os alvos biológicos conhecidos para cada molécula (ACTS Dataset), apresentado de forma similar a um ficheiro de transações (cada linha corresponde a uma molécula e cada item é um dos alvos). No ACTS Dataset, foram encontradas as association rules mais fortes, de forma a percebermos quais os alvos que têm relações mais fortes entre si, para além disto foi realizado um problema de aprendizagem supervisionada (supervised learning), onde foi selecionado um atributo, com o objetivo de o modelo treinado prever se para novas moléculas, esta contém o gene selecionado ou não. No FPS Dataset foram encontradas as moléculas com valores mais altos de semelhança, de acordo com a semelhança de Jaccard.

2 Conjuntos de dados

2.1 Conjunto de dados FPS

Em primeiro lugar, foi carregado o ficheiro fps.txt. Um dataset que tem 1101 linhas e 2048 colunas, selecionando a coluna das moléculas como índice do dataframe. Foi feita uma pesquisa na qual verificámos que nenhuma coluna tem apenas valores 0, isto é, todas as colunas podem ter informação útil. Por fim verificámos que não existem valores nulos no conjunto de dados.

Foi ainda verificado quais foram os atributos ativos em maior e menor número de moléculas, tendo-se obtido para ambos os seus top 5. Como os atributos estão definidos por valores binários, foi utilizada a métrica média para identificar a percentagem de moléculas com o atributo ativo. Os atributos mais ativos foram os seguintes:

1. Atributo 927, está ativo em 50.02% das moléculas
2. Atributo 1918, está ativo em 50.01% das moléculas
3. Atributo 1089, está ativo em 49.54% das moléculas
4. Atributo 1020, está ativo em 49.31% das moléculas
5. Atributo 936, está ativo em 48.57% das moléculas

Enquanto que os genes menos ativos foram:

1. Atributo 45, está ativo em 3.01% das moléculas
2. Atributo 1876, está ativo em 4.26% das moléculas
3. Atributo 660, está ativo em 5.22% das moléculas
4. Atributo 1321, está ativo em 5.22% das moléculas
5. Atributo 383, está ativo em 5.22% das moléculas

2.2 Conjunto de dados Acts

Inicialmente foi carregado o ficheiro acts.txt. Um dataset que tem 1101 linhas e 201 colunas (utilizando as moléculas como indíce do dataframe), de seguida os valores do dataframe foram convertidos para valores booleanos. As colunas passaram a ser os genes, aumentando assim o número de colunas de 201 para 1244, onde para cada molécula é colocado 1 no gene que contém e 0 no gene que não contém. Após esta formatação foram calculadas as 5 moléculas que têm o maior número de genes e obteve-se os seguintes resultados:

1. PDFDA0131, com 201 genes ativos
2. PDFDA0965, com 179 genes ativos
3. PDFDA0907, com 131 genes ativos
4. PDFDA0263, com 129 genes ativos
5. PDFDA0772, com 110 genes ativos

Também foram calculadas as moléculas com menor número de genes ativos e verificou-se que todas têm pelo menos 1 gene ativo mas existem 182 com apenas 1 gene ativo. De seguida são apresentadas 5 das 182 moléculas que estão ativas em menos genes:

1. PDFDA0551, com apenas 1 gene ativo
2. PDFDA0670, com apenas 1 gene ativo
3. PDFDA0155, com apenas 1 gene ativo
4. PDFDA0666, com apenas 1 gene ativo
5. PDFDA0648, com apenas 1 gene ativo

Além disto, foram identificados os 5 genes mais e menos ativos em moléculas, para isto foram convertidos os valores booleanos do conjunto de dados para valores binários, de forma a obtermos a média dos valores de cada coluna.Média esta que nos indicará os genes mais ativos e menos ativos.

Top 5 dos genes mais ativos em moléculas:

1. Gene REP, está ativo em 49.99% das moléculas
2. Gene SLCO1B1, está ativo em 46.11% das moléculas
3. Gene SLCO1B3, está ativo em 46.07% das moléculas
4. Gene LMNA, está ativo em 41.55% das moléculas
5. Gene CYP3A4, está ativo em 33.93% das moléculas

Top 5 dos genes menos ativos em moléculas:

1. Gene ZAP70, está ativo em 3.01% das moléculas

2. Gene MME, está ativo em 3.01% das moléculas
3. Gene S1PR1, está ativo em 3.01% das moléculas
4. Gene S1PR3, está ativo em 3.01% das moléculas
5. Gene S1PR4, está ativo em 3.01% das moléculas

3 Association Rules

3.1 Introdução

O primeiro objetivo definido, utilizando o conjunto de dados do ficheiro Acts.txt, passa pela descoberta de associações entre os genes, de forma a saber por exemplo, qual ou quais os genes que uma molécula poderá ter se tiver um outro conjunto de genes.

3.2 Desenvolvimento

Para identificação destas associações é verificado em primeiro lugar o desempenho dos diferentes algoritmos utilizados (Apriori, FP-growth e ECLAT) para diferentes valores de threshold, de forma a selecionar o melhor algoritmo com o melhor threshold que será depois utilizado no processo de identificação das regras. Esta verificação é feita através de um ciclo for que para cada um dos thresholds dados (0.2; 0.1; 0.05; 0.02; 0.015), calculará o tempo de processamento de cada algoritmo, guardando a informação num dicionário.No final de todas as iterações do ciclo, este dicionário será convertido num dataframe, onde pode ser encontrada toda a informação do desempenho de cada algoritmo para diferentes valores de threshold.

Após a definição do dataframe que contém a informação do desempenho de cada algoritmo, são realizadas algumas representações gráficas, onde concluímos que o número de itemsets (conjuntos de genes) vai aumentando à medida que o valor de threshold vai dimininuindo.

Desta forma foi selecionado o threshold de 0.02 que contém cerca de 50 mil conjuntos de genes, este valor de threshold foi selecionado pelo facto de não apresentar o maior número de conjuntos de genes, cerca de 1 milhão de conjuntos de genes utilizando um threshold de 0.015, nem ter o menor número de conjuntos, com apenas 5 conjuntos de genes utilizando um threshold de 0.2.

A outra representação gráfica foi criada para verificar o desempenho de cada algoritmo à medida que o número de conjuntos vai aumentando. Esta representação ajudou-nos a escolher o algoritmo ECLAT como aquele que será utilizado para identificar os conjuntos de genes frequentes, por não só apresentar o melhor desempenho para o threshold selecionado mas por também apresentar o melhor desempenho independentemente do número de conjuntos de genes existentes.

Depois de selecionado o algoritmo que obteve o melhor desempenho (ECLAT), foi então feita a identificação dos conjuntos de genes frequentes utilizando também o threshold selecionado (0.02). De seguida foram identificadas regras de associação para os conjuntos de genes frequentes, utilizando a métrica - confiança com um threshold mínimo de 70%.

3.3 Resultados obtidos

Das regras obtidas foram selecionadas as regras que obtiveram um lift superior ou igual a 44 e uma convicção superior a 13, obtendo assim 177 regras. Com este conjunto de regras podemos dizer, por exemplo, que:

- Se uma molécula tiver os genes ADRA2B, SLC6A2, CHRM4, HTR2A, DRD1 e CHRM2 também terá os genes ADRA2A, DRD3, CHRM3, HTR6;
- Se uma molécula tiver os genes SLC6A2, CHRM4, HTR2B, ADRA1D, HTR2A e CHRM2 também terá os genes ADRA2A, HTR6, DRD3, CHRM3, CHRM1;
- ...

4 Supervised Learning

4.1 Introdução

Como segunda tarefa, decidimos resolver um problema de aprendizagem supervisionada, selecionando um gene como o alvo (SLCO1B1) e que terá como objetivo, identificar se uma determinada molécula contém ou não este gene, utilizando novamente os dados do ficheiro Acts.txt.

Por consequência de ser um problema de aprendizagem supervisionada, o conjunto de dados será dividido em conjunto de treino e em conjunto de validação independente. Posteriormente foram treinados, utilizando validação cruzada, vários modelos de classificação:

- BernoulliNB;
- RandomForest;
- DecisionTree;
- AdaBoost.

Destes apenas é selecionado aquele que obtém o melhor resultado de f1-score, métrica utilizada por tratar-se de um problema binário. De seguida o melhor modelo é treinado mas desta vez para todo o conjunto de treino e avaliado pelo conjunto de validação idependente.

4.2 Desenvolvimento

Em primeiro lugar, começou-se por deslocar a coluna do gene alvo selecionado para a última posição do dataframe, convertendo depois todos os valores do conjunto de dados para valores binários.

De seguida foi realizada uma pequena exploração do alvo selecionado, verificando em quantas moléculas este gene encontra-se ativo e não ativo. Desta exploração identificámos que existem 764 moléculas que não têm o gene SLCO1B1 ativo e 337 moléculas têm este gene ativo.

Posteriormente converteu-se o dataframe para NumpyArray, separando o conjunto de dados em X(todos os valores de todos os genes excepto o gene SLCO1B1) e y(todos os valores do gene SLCO1B1), que depois foram divididos em conjunto de treino e conjunto de validação independente.

Para a descoberta do melhor modelo foi definida uma função que recebe como argumentos o conjunto de treino e o modelo. Na função é realizada a validação cruzada de 5 ficheiros utilizando o conjunto de treino recebido como argumento, retornando depois o conjunto de teste gerado e os resultados previstos para esse conjunto de teste. Esta função é chamada dentro de um ciclo for que em cada iteração manda como argumento um novo modelo, fazendo depois a comparação do f1 score retornado pela função, com o f1 score obtido na iteração anterior.

4.3 Resultados obtidos

Depois de calculados os scores obtidos em cada modelo, chegámos à conclusão que o melhor modelo de classificação treinado foi o Random Forest, que obteve os seguintes resultados no treino inicial do modelo:

- F1 Score - 0.97%
- Recall - 0.96%
- Precision - 0.97%
- Matthews Correlation Coefficient - 0.95%

Estas foram as métricas utilizadas para avaliação do modelo devido a tratar-se de um problema de classificação binária.

Após a seleção do modelo Random Forest como o melhor modelo treinado, treinou-se novamente este modelo mas desta vez utilizando todo o conjunto de treino e utilizando o conjunto de validação independente para avaliação do modelo final.

Após o treino do modelo com todo o conjunto de treino, obteve-se os seguintes resultados utilizando o conjunto de validação independente:

- F1 Score - 0.97%
- Recall - 0.99%
- Precision - 0.96%
- Matthews Correlation Coefficient - 0.96%

Como se pode ver, para além dos bons resultados obtidos pelo modelo de classificação, é possível verificar que não existe qualquer sinal de overfitting, pois os resultados obtidos para novos dados continuam semelhantes aos obtidos no treino inicial do modelo.

5 Similar Molecules

5.1 Introdução

Para finalizar e por ainda não termos utilizado o conjunto de dados do ficheiro fps.txt, para esta última tarefa foi utilizado este conjunto de dados. Por este já se encontrar no formato binário e não sabermos o que cada atributo significa foi utilizada esta última tarefa para descobrir eventuais semelhanças que possam haver entre as várias moléculas do conjunto de dados.

5.2 Desenvolvimento

Em primeiro lugar começou-se por definir a função de JaccardSim que calcula a similaridade das moléculas da seguinte maneira:

1. É calculado, para as moléculas em estudo, o número de colunas onde em ambos os atributos (colunas) o valor 1 está presente (interseção de moléculas);
2. É calculado o número total de atributos (colunas) onde pelo menos um valor 1 está presente, não estando obrigatoriamente o valor 1 para ambas no mesmo atributo (União de moléculas);
3. E por último é feita a divisão entre o primeiro passo e o segundo, de forma a obter o valor de similaridade entre as moléculas.

De seguida foi feito um tratamento do conjunto de dados, de forma a que este pudesse ser utilizado pela função Jaccard. Para isso, a primeira tarefa a ser realizada foi definir a coluna com os nomes das moléculas como índice do dataframe, depois foram guardados o número de colunas e linhas do dataframe. Após este pequeno tratamento do conjunto de dados, foi guardada numa lista o índice do dataframe (os nomes das moléculas) e transformado o dataframe num NumPy array.

Concluída esta primeira fase de pré-processamento dos dados, realizou-se de seguida o cálculo dos valores de similaridade entre as diversas moléculas, utilizando a função anteriormente referida (JaccardSim). Este cálculo é feito utilizando dois ciclos for, um dentro do outro, que percorrem o Numpy array que contém todas as moléculas (linhas) e atributos (colunas) do conjunto de dados. O primeiro ciclo percorre todo o array do conjunto de dados uma única vez enquanto que o segundo vai percorrer N vezes o array da posição seguinte ao do ciclo anterior até ao final do array, sendo N o número de moléculas existentes no conjunto de dados. Dentro desse segundo ciclo é chamada a função JaccardSim de maneira a calcular para cada molécula o valor de similaridade em relação a todas as outras molécula do conjunto de dados.

5.3 Resultados obtidos

Dos resultados obtidos pela função JaccardSim apenas aqueles que tivessem um valor de similaridade superior a 70% de semelhança eram considerados.Considerando esta percentagem, a função encontrou 70 possíveis semelhanças entre as diversas moléculas.

Entre elas podemos encontrar por exemplo:

- A molécula PDFDA0115 que é similar à molécula PDFDA0281 com um valor de similaridade de 100%
- A molécula PDFDA0842 que é similar à molécula PDFDA1046 com um valor de similaridade de 94%
- A molécula PDFDA0089 que é similar à molécula PDFDA0216 com um valor de similaridade a rondar os 83%
- A molécula PDFDA0036 que é similar à molécula PDFDA0450 com um valor de similaridade de 75%
- A molécula PDFDA0235 que é similar à molécula PDFDA0682 com um valor de similaridade a rondar os 70%
- ...

De notar ainda que das 70 semelhanças encontradas em moléculas, 17 são consideradas semelhantes com um valor de similaridade de 100%.

6 Conclusão

Em conclusão, e depois de realizadas estas tarefas podemos afirmar que apesar de existirem diversas moléculas, algumas delas poderão ser muito semelhantes, diversas moléculas possuem os mesmos genes, conseguimos determinar se uma nova molécula terá ou não o gene SLCO1B1 como ativo, prever os restantes genes através dos genes já conhecidos. Contudo ainda poderiam ter sido realizados outros tipos de tarefas que poderiam clarificar ainda mais os resultados obtidos como por exemplo a identificação de clusters no ficheiro Acts.txt que nos agruparia as diversas moléculas de acordo os genes ativos de cada molécula, porém também poderia ser mais difícil não só pela sua complexidade mas também na descoberta do número certo de clusters pelo não conhecimento profundo dos dados em estudo.

7 Bibliografia

Python 3 online materials, tutorials.
Scikit-Learn online tutoriais.
A. Downey, “Think Python”, 2nd edition, O’Reilly Media, 2015.
S. Raschka, “Python Machine Learning” Packt Publishing Ltd, 2015.
Andreas C. Muller Sara Guido, “Introduction to Machine Learning in Python – A Guide for Data Scientists”, O’Reilly, 2016