Extracção e Análise de Dados da Web

Project Report

Movie-Recommendation System

Sérgio Moura - 70561
João Azevedo - 70614

Grupo 1

Enquadramento

Este projeto baseia-se num sistema de recomendações cinematográficas, que tem em conta os gostos e escolhas prévias dos seus utilizadores. O objectivo foi construir uma aplicação que tomasse como input uma série de utilizadores, uma série de filmes e uma série de votos (dados pelos utilizadores a alguns dos filmes considerados) e retornasse ao utilizador sugestões de outros filmes, por ele não votados, que se crê que sejam do seu agrado.

Antes de mais, como suporte ao armazenamento dos dados necessários ao sistema foi utilizado o Neo4J, uma *graph-database* que permite facilmente explicitar relações entre entidades e visualizar estas mesmas relações numa interface gráfica de fácil entendimento, ou *"Whiteboard Friendly"* segundo os autores. Estas características contribuíram em grande parte para esta nossa escolha, uma vez que um sistema de recomendação se prende em grande medida pelas relações existentes entre as variadas entidades e pela definição de modelos de exploração dessas relações para a obtenção de resultados.

O trabalho foi dividido em 3 partes diferentes, cada qual desenvolvida utilizando variadas ferramentas:

A - Modo online:

No modo online, foi desenhada um aplicação em Node.js para que os utilizadores pudessem votar em filmes e receber recomendações baseadas nos seus votos. Esta aplicação online compreende dois modos, o modo de voto e o modo de obtenção de recomendações.

urls: localhost:3000/rate-movie?id=1234
localhost:3000/get_recommendation?id=4321

No modo de voto, a interface dispõe apenas de uma imagem (capa do filme) e nome do filme que é pedido ao utilizador que avalie, sendo que o utilizador se autentica através do URL da aplicação, um campo para escrever a classificação que se pretende introduzir no sistema para o filme em questão e um botão de submissão da classificação. É de salientar que cada classificação que o utilizador atribui a um filme no modo online é automaticamente tida em conta caso o utilizador peça ao sistema um conjunto de filmes que lhe sejam recomendados.

No modo de obtenção de recomendações, é apenas apresentado ao utilizador a “Wall of Recommendation”, um conjunto de filmes que se revelam adequados ao utilizador e que este provavelmente gostará.

Para fazer isto, a aplicação efectua uma query à graph-database pelos 100 filmes mais cotados nos géneros de que o utilizador é fã. Tendo em conta este resultado, é usada uma medida de semelhança entre as labels atribuídas a cada filme, resultantes das reviews desses filmes.

B - Batch training mode:

Neste modo, o sistema recebe como input um ficheiro contendo triplos de (*user id, movie id, rating*) em que cada um consiste no *rating* dado pelo utilizador identificado pelo *user id* ao filme identificado com o *movie id*.
Antes de mais, é necessário correr o script *addPeople.py*, que faz o *parsing* do ficheiro **ml-100k/u.user** (que contém a informação relativa a todos os utilizadores do sistema) e carrega para a *graph-database* todos os utilizadores sob a forma de nós, criando também nós para as profissões dos utilizadores, necessários à relação que é criada entre estes dois nós (*People* e *Ocupation*) e que dá pelo nome de *Works_In*. Além disso, é atribuída uma *label* a cada utilizador, conforme o seu género (MALE ou FEMALE), presente no ficheiro **u.user**.

De seguida, é necessário executar o ficheiro *addMovies.py* que efectua o carregamento para a *graph-database* dos filmes presentes no ficheiro *movieList.txt*, que não é mais que uma versão corrigida do ficheiro original do ml-100k que continha a colecção de filmes. Esta correcção foi necessária devido a existirem filmes cujo URL do *imdb.com* estava incorrecto no ficheiro original, não conduzindo directamente à página respectiva ao filme.
É então criado um nó para cada filme e um nó para cada género de filme, necessários à criação da relação que dita os géneros de cada um dos filmes na base de dados, que dá pelo nome de *Genre_As*. Os diferentes géneros que podem ser atribuídos aos filmes estão presentes no ficheiro *ml-100k/u.genre* e são carregados através do ficheiro *addGenre.py*

Na nossa solução, foram também consideradas as opiniões de outros utilizadores sobre a forma de *reviews* presentes no site *rottentomatoes.com*. Para cada um dos filmes carregados para a BD, foram pesquisadas no site as *reviews* desse mesmo filme e escolhida, quando possível, a review proveniente do site do New York Times (http://movies.nytimes.com).

A análise desta *review* foi feita com o objectivo de determinar um máximo de 10 palavras que constituem *labels* que descrevem cada um dos filmes. O ficheiro *testRottenTomatoes.py* utiliza a API do *rottentomatoes.com* para conseguir chegar ao URL de cada uma das *reviews*, de seguida utiliza o BeautifulSoup para obter o texto da *review* e finalmente analisa este texto, comparando-o com um conjunto de palavras que tipicamente caracterizam filmes, presentes no ficheiro *words.txt*.
As palavras que compõem este ficheiro foram descarregadas do site *www.words-to-use.com/words/movies-tv* e depois analisadas com o auxílio do *corpus* do NLTK para remover palavras potencialmente perigosas para a qualidade das *labels* e para considerar apenas nomes próprios e não adjectivos (que apenas indicariam uma opinião qualitativa do filme e não tanto características do mesmo) e também para excluir palavras que constituíssem nomes de géneros de filmes, já associados a cada filme. Estas *labels* foram carregadas para a *graph-database*, associadas a cada um dos filmes.

De seguida, é necessário carregar para a *graph-database* os votos dados pelos utilizadores aos filmes. Para tal, é necessário correr o ficheiro *addVote.py*, que faz o parsing do ficheiro *ml-100k/u*.base*, e para cada linha do ficheiro procura o nó do utilizador e filme em questão e cria uma relação entre ambos que dá pelo nome de *Rates*, contendo a classificação dada.

Para obter as imagens das capas dos filmes apresentadas na interface web do modo Online, é necessário correr o ficheiro getJPG.py, que efectua pedidos através da API do rottentomatoes.com e que descarrega para todos os filmes as respectivas capas, sendo que para os que não dispõem de uma, é utilizada uma imagem que ilustra esta ausência.
Neste momento, a graph-database já terá toda a informação necessária ao cálculo do modelo pelo qual serão dadas as recomendações aos utilizadores…

NOTA: Poderíamos ter usado mais reviews para além das presentes no site cinematográfico do New York Times. No entanto, assumimos que seria desnecessário pois os resultados verificados são muito similares aquando do uso de várias e a complexidade do sistema cresce substancialmente, uma vez que o número de pedidos a efectuar para cada filme também aumenta, sendo os pedidos o principal consumidor de tempo na execução do sistema.

C - Batch Testing Mode:

Neste modo, foi testado o modelo gerado a partir do modo Batch Training Mode, através da sua aplicação sobre os ficheiros *u*.test* e do modelo definido, e comparando o resultado da avaliação do *rating* que cada utilizador daria ao filme em questão com o resultado real, explicitado no ficheiro.

Em primeiro lugar, para o modelo descrito, foi aplicado aos ficheiros u1.base e u1.test, em cada linha do ficheiro de teste, a *query* seguinte, escrita na linguagem *Cypher* (linguagem de *querying* usada em Neo4j), que visa obter a classificação arredondada que o sistema prevê que o utilizador daria ao filme em questão:

```
start a=node:People(people_id="+person_id+"),
      b=node:Movie(movie_id="+movie_id+")
match p=a-[r:Rates]->c<-[s:Rates]-d-[t:Rates]->b,
      c-[u:Genre_As]->g<-[v:Genre_As]-b
where r.rating = s.rating with a,d,t.rating as rate,
      count(distinct(c)) as total
with a,d,rate,total
order by total desc
limit 10 with rate
return avg(rate)
```

Traduzindo esta *query*, o sistema faz o *parsing* do ficheiro e procura na *graph-database* o nó relativo ao utilizador com o ID *person_id* e o nó relativo ao filme com o ID *movie_id*.
De seguida, é calculado o *rate* que o utilizador iria dar através do cálculo da média dos *rates* que outros 10 utilizadores deram ao mesmo filme, sendo que a similaridade entre esses 10 utilizadores e o utilizador em questão é que estes foram os que mais filmes classificaram, filmes estes do mesmo género do filme para o qual se pretende obter classificação, com a mesma nota que o utilizador em questão. Daí ser provável, devido à semelhança entre os filmes considerados e os utilizadores considerados, que o *rate* dado pelo utilizador com o ID *person_id* seja uma média dos *rates* dados pelos seus semelhantes.

Para a query acima, foram obtidos os seguintes resultados após a sua aplicação ao ficheiro de teste, em termos de avaliação qualitativa:

Precision	0.40
Float MAE	0.79
MAE	0.75

Esta *query* constituiu um ponto de partida na nossa análise, sendo que aquele que assumimos como a implementação final apenas previne a situação onde podem não existir dois filmes do mesmo género votados pelo mesmo utilizador. O passo seguinte passa por excluir este parâmetro e descobrir semelhanças mais gerais entre os utilizadores (não tem em conta apenas o(s) género(s) do filme).

Conclusão

Alguns testes paralelos foram tentados, queries simples que consideravam labels em comum, idades dos utilizadores e anos dos filmes, profissão e género. Estas queries apesar de apresentarem algum interesse ao nível da descoberta de padrões, não se mostravam qualitativamente vantajosas.

Baseado somente na idade:

Precision	0.36
Float MAE	0.83
MAE	0.80