UTILIZANDO MINERAÇÃO DE REGRAS DE ASSOCIAÇÃO SOBRE UMA BASE DE DADOS MUSICAL

Fábio Oliveira Tempesta

Instituto Federal de Minas Gerais (IFMG) – Campus Bambuí

fabio.oliveira.tempesta@gmail.com

RESUMO

A mineração de conjuntos de itens frequentes é uma tarefa da área de mineração de dados com maior número de aplicação prática. O trabalho consiste em encontrar relações de artistas escutados por usuário em uma grande base de dados para servir como recomendação de um novo artista. A base de dados foi passada por uma preparação antes de ser feita a mineração com o intuito de melhorar os resultados e diminuir processamento. Foi utilizado o algoritmo Apriori com a biblioteca Apyori, ajustando os parâmetros para gerar o melhor resultado viável. O algoritmo encontrou 43 relações de conjunto de itens frequentes de tamanho 2 e 3 com uma relevância considerável para o contexto da aplicação.

Palavras-chave: Regras de Associação; Gosto Musical; Apriori

1 INTRODUÇÃO

A mineração de conjuntos de itens frequentes é uma tarefa da área de mineração de dados com maior número de aplicação prática. Ela compartilha muitos recursos de indução de regra de classificação, mas são diferentes, enquanto classificação foca em adquirir capacidade de fazer previsões, as regras de associação concentra em fornecer uma visão clara daqueles dados processados (YE, 2003). O trabalho pioneiro nessa área foi a análise de carrinhos de compras de clientes, conseguindo encontrar relações interessantes que conseguiram induzir as pessoas a comprarem mais sobre os determinados produtos que estavam associados a regra. A análise de associação é responsável por descobrir essas interessantes relações escondidas nas grandes base de dados, elas são representadas na forma de regras de associação ou conjunto de itens frequentes (TAN; STEINBACH; KUMAR, 2005).

Geralmente os algoritmos projetados para essa finalidade encontram os conjuntos de itens frequentes utilizando métricas de suporte, que mede a razão entre a presença do conjunto e o número total de transações, e de confiança, que determina a acurácia da indução realizada. Existem outras medidas relevantes utilizadas pelos algoritmos, como *Lift* e Convicção, sendo utilizado em alguns determinados tipos e a escolha do autor.

O presente trabalho tem o propósito de recomendar estilos musicais, ou o conjunto de itens frequentes de interesse, para os usuários a partir de uma base de dados contendo as bandas escutadas individualmente por dezena de milhares de usuários. Utilizando o algoritmo *Apriori*,

será encontrado uma relação entre os artistas, podendo recomendar novas de acordo com o estilo musical em comum descoberto na base de dados dos usuários.

2 ANÁLISE E PREPARAÇÃO DA BASE DE DADOS

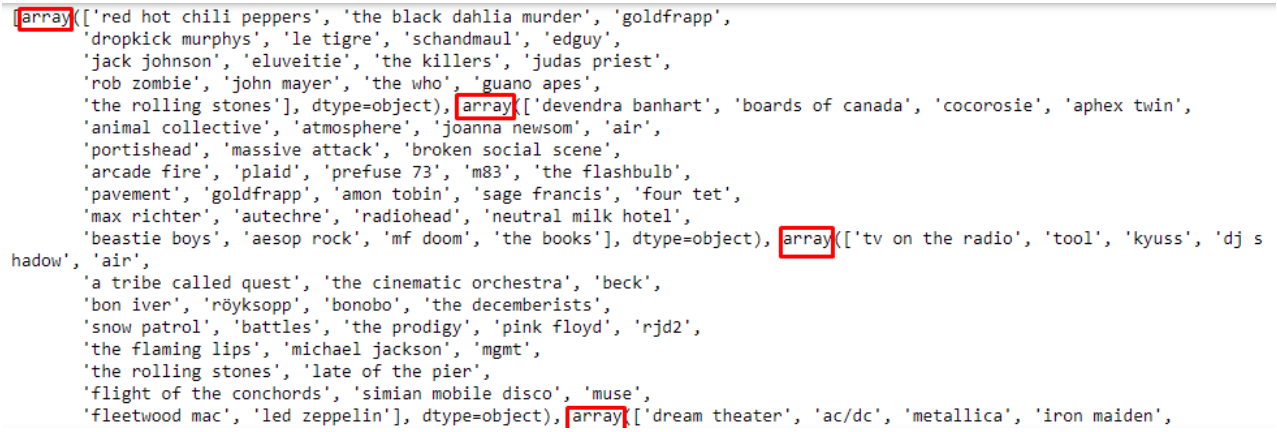
Observando a figura 1, percebe-se algumas irregularidades que poderia vir a complicar a mineração, como o modo em que as informações estão distribuídas e *features* com pouca relevância para a finalidade, como *sex* e *country*.

	user	artist	sex	country
0	1	red hot chili peppers	f	Germany
1	1	the black dahlia murder	f	Germany
2	1	goldfrapp	f	Germany
3	1	dropkick murphys	f	Germany
4	1	le tigre	f	Germany
...	...	...	..	...
289950	19718	bob dylan	f	Canada
289951	19718	pixies	f	Canada
289952	19718	the clash	f	Canada
289953	19718	a tribe called quest	f	Canada
289954	19718	radiohead	f	Canada

```
Data columns (total 4 columns):
 #   Column   Non-Null Count   Dtype
---  ------   --------------   -----
 0   user     289955 non-null  int64
 1   artist   289955 non-null  object
 2   sex      289955 non-null  object
 3   country  289955 non-null  object
dtypes: int64(1), object(3)
```

Figura 1 – Informações sobre a base de dados original.
Fonte: Elaborada pelo autor.

Para minerar os conjuntos de item frequentes, a organização ideal seria uma lista de transações, de modo em que todos os artistas do usuário ficassem concatenados. Foi utilizado a biblioteca *pandas* para auxiliar essa modificação no *dataset*, ficando como mostra a figura 2, em que cada *array* significa os artistas de um usuário.



```
[array(['red hot chili peppers', 'the black dahlia murder', 'goldfrapp',
       'dropkick murphys', 'le tigre', 'schandmaul', 'edguy',
       'jack johnson', 'eluveitie', 'the killers', 'judas priest',
       'rob zombie', 'john mayer', 'the who', 'guano apes',
       'the rolling stones'], dtype=object), array(['devendra banhart', 'boards of canada', 'cocorosie', 'aphex twin',
       'animal collective', 'atmosphere', 'joanna newsom', 'air',
       'portishead', 'massive attack', 'broken social scene',
       'arcade fire', 'plaid', 'prefuse 73', 'm83', 'the flashbulb',
       'pavement', 'goldfrapp', 'amon tobin', 'sage francis', 'four tet',
       'max richter', 'autechre', 'radiohead', 'neutral milk hotel',
       'beastie boys', 'aesop rock', 'mf doom', 'the books'], dtype=object), array(['tv on the radio', 'tool', 'kyuss', 'dj s
hadow', 'air',
       'a tribe called quest', 'the cinematic orchestra', 'beck',
       'bon iver', 'röyksopp', 'bonobo', 'the decemberists',
       'snow patrol', 'battles', 'the prodigy', 'pink floyd', 'rjd2',
       'the flaming lips', 'michael jackson', 'mgmt',
       'the rolling stones', 'late of the pier',
       'flight of the conchords', 'simian mobile disco', 'muse',
       'fleetwood mac', 'led zeppelin'], dtype=object), array(['dream theater', 'ac/dc', 'metallica', 'iron maiden',
```

Figura 2 – Informações sobre a lista de transação.
Fonte: Elaborada pelo autor.

3 IMPLEMENTAÇÃO: ALGORITMO APRIORI

O algoritmo *Apriori* tem o o objetivo de extrair regras de associação em grandes bases de dados, encontrando todos os conjuntos de itens frequentes (*itemsets* frequentes). Seu funcionamento é dividido em duas funções: uma para gerar os candidatos e eliminar aqueles

que não são frequentes (suporte abaixo do mínimo), a outra utilizada para extrair as regras de associação (ROMÃO et al., 1999). As regras de associação basicamente é uma indução de um elemento, ou seja, se um elemento aparecer, o outro tem grande chances de aparecer (a métrica confiança é responsável por ela). O *Apriori* começa realizando cálculos do suporte de todos *itemsets* frequentes de tamanho unitário (apenas um elemento, k=1), nos passos posteriores, k>1, os *itemsets* frequentes encontrado nos passos anteriores (k-1) são utilizados para gerar os conjunto de itens potencialmente frequentes, sendo verificado se ele passa nas métricas (ROMÃO et al., 1999). Enquanto maior o conjunto de itens frequentes, menor a chance dele satisfazer a métrica proposta, chegando em um ponto em que nenhum conjunto será aprovado, terminando a busca pelos *itemsets frequentes*.

Para utilizá-lo, será importado uma biblioteca do Python chamada *Apyori*, que é uma simples implementação desse algoritmo, podendo determinar o valor das métricas de suporte, confiança e *lift* por meio dos parâmetros. Esses parâmetros deve ser escolhido do modo em que ele consiga gerar bons conjuntos regras e em uma quantidade moderada, podendo deixar os valores das métricas um pouco mais rígido.

A métrica de suporte, observando a figura 3, que é uma captura de tela de *output* do código, mostra que, caso os dados esteja bem distribuídos, a chance de um conjunto de itens constituído por apenas um artista (*itemset* tamanho 1) for maior que 15% é relativamente baixa, devido a razão entre o número de artistas e número de usuários ser de aproximadamente 6,7%. Já é esperado um valor de suporte baixo de parâmetro, pois a tendência é apenas abaixar enquanto maior o conjunto de itens fica.

```
Número de usuários: 15000
Número de artistas: 1004
Média de aristas por usuário: 19.3302
```

Figura 3 – Quantidade de usuários, artistas e média de artistas por usuário.
Fonte: Elaborada pelo autor.

4 RESULTADOS E DISCUSSÃO

Para obter o melhor resultado possível, foi utilizado um suporte mínimo de 0,7%, que, considerando que a base de dados contém 15000 usuários, são no mínimo 105 transações, não sendo um número tão baixo para uma base de dados com tantos artistas. A confiança mínima foi de 60%, não podendo ultrapassar muito desse valor, sendo que muitas relações deixariam de existir. A métrica *lift* foi de 3, um valor bem aceitável para verificar a correlação, dado que o valor deve ser acima de 1. Para os parâmetros utilizados, o tamanho máximo de *itemsets* foi de apenas 3, encontrando 43 relações, como mostra a Figura 4.

```
Foram encontradas 43 relações relevantes!
Relação 0:['keane', 'coldplay']
Relação 1:['nightwish', 'epica']
Relação 2:['sigur rós', 'radiohead', 'air']
Relação 3:['coldplay', 'arctic monkeys', 'oasis']
Relação 4:['radiohead', 'coldplay', 'beck']
Relação 5:['death cab for cutie', 'radiohead', 'beck']
Relação 6:['modest mouse', 'radiohead', 'beck']
Relação 7:['britney spears', 'beyoncé', 'rihanna']
Relação 8:['blur', 'radiohead', 'coldplay']
Relação 9:['bob dylan', 'the beatles', 'coldplay']
Relação 10:['bob dylan', 'the beatles', 'led zeppelin']
Relação 11:['bob dylan', 'the beatles', 'pink floyd']
Relação 12:['death cab for cutie', 'broken social scene', 'radiohead']
Relação 13:['death cab for cutie', 'snow patrol', 'coldplay']
Relação 14:['the killers', 'coldplay', 'kaiser chiefs']
Relação 15:['keane', 'muse', 'coldplay']
Relação 16:['keane', 'radiohead', 'coldplay']
Relação 17:['keane', 'snow patrol', 'coldplay']
Relação 18:['keane', 'the killers', 'coldplay']
Relação 19:['oasis', 'muse', 'coldplay']
Relação 20:['oasis', 'snow patrol', 'coldplay']
Relação 21:['oasis', 'the killers', 'coldplay']
Relação 22:['snow patrol', 'radiohead', 'coldplay']
Relação 23:['the smashing pumpkins', 'radiohead', 'coldplay']
```

Figura 4 – Parte das relações encontradas pelo algoritmo.
Fonte: Elaborada pelo autor.

5 CONCLUSÃO

O algoritmo conseguiu extrair boas relações contidas no *dataset*, podendo servir de recomendações novos artistas para os usuários. Apesar das métricas de confiança e suporte ficarem relativamente baixa, para o contexto do uso que seria ampliar o leque de músicas escutadas pelos usuário fazendo recomendações é aceitável, porque uma recomendação geralmente não tem a obrigação de ser uma dica que faça sentido a todos os usuários, ou seja, uma métrica de confiança muito alta.

Caso fosse utilizar os atributos de região e sexo do usuário, as métricas seriam ainda mais baixas, devido a separação ainda maior dos dados. Essa separação seria uma estratégia desfavorável para essa aplicação.

REFERÊNCIAS

ROMÃO, W. et al. Extração de regras de associação em C&T: O algoritmo Apriori. **XIX Encontro Nacional em Engenharia de Produção**, sn, v. 34, p. 37–39, 1999.

TAN, P.-N.; STEINBACH, M.; KUMAR, V. **Introduction to data mining**. Pearson Education, 2005.

YE, N. **The handbook of data mining**. 1. ed.: Lawrence Erlbaum Associates, Publishers, 2003. (Human factors and ergonomics).

APÊNDICE A - Código de desenvolvimento

```
#imports
import pandas as pd
from apyori import apriori


#base original
source_df = pd.read_csv('lastfm.csv')
#Figura 1
print(source_df)
print(source_df.info())


#Preparação dos dados
transaction_list = []
for i in source_df['user'].unique():
    itemsets = source_df[source_df['user'] == i]['artist'].values
    transaction_list.append(itemsets)

#Figura 2
print(transaction_list)


#Analisando a métrica de suporte
print('Número de usuários:',source_df['user'].nunique())
print('Número de artistas:',source_df['artist'].nunique())
artistsNumber_per_user.clear()
for i in source_df['user'].unique():
    artistsNumber_per_user.append( int(source_df[source_df['user'] == i]['
    artist'].nunique()))

#Figura 3
print('Média de aristas por usuário:',sum(artistsNumber_per_user)/len(
    artistsNumber_per_user))
print('Número total de artistas denominado:',sum(artistsNumber_per_user))


#Minerando relações
relations_set = list(apriori(transaction_list, min_support=0.007,
    min_confidence=0.6, min_lift=3, min_length=2))

#Figura 4
print ('Foram encontradas',len(relations_set),'relações relevantes!')
formatted_relations = [list(relations_set[i][0]) for i in range(0,len(
    relations_set))]
for i in range(len(formatted_relations)):
    print('Relação ',i,':', formatted_relations[i], sep='')
```