Introdução a Regras de Associação de Dados pelo Método Apriori

Hugo Fernandes[1], Flavius Gorgonio[1]

[1]Depto de Computação e Tecnologia - Universidade Federal do Rio Grande do Norte (UFRN)
Caixa Postal: 59.300-000 - Caicó - RN - Brazil

hugo.medeiros.fernandes@gmail.com, flavius@ufrnet.br

Abstract. *This paper seeks to show the essential concepts about the data binding techniques and understand its role as an instrument in decision support. To assist in this task, based on animal registration data will be used in a Zoo.*
Resumo. *Este trabalho busca mostrar os conceitos essenciais a respeito das técnicas de associação de dados e compreender seu papel como um instrumento no suporte a decisões. Para auxiliar nesta tarefa, será utilizada uma base de dados de registro de animais num Zoológico.*

1. Base de Dados

Essa base de dados foi criada por Richard Forsyth no zoológico de Mapperley Park, em 15/05/1990. Teve como proposito o registro das características dos animais.

A base de dados é composta por 101 especies de animais, onde cada linha representa uma nova especie. A maioria das colunas contém características de seres vivos, exemplo, se o animal possui Pelos, Dentição, Pernas, se é domesticável, etc. Esses dados são do tipo booleano, dessa forma, assumindo o valor 0 para indicar a ausência do atributo e o valor 1 para a presença. Os atributos podem ser vistos em mais detalhes na tabela 1 no item Apêndices.

Outras colunas são encontradas na base de dados, onde representam informações relevantes. A coluna '*name*' especifica a especie do animal, sendo seu tipo de dado texto. A coluna '*legs*', mostra o número de patas do animal, e seu tipo de dado é numérico variando de 0 a 8. Finalmente, a coluna '*type*', contém um valor numérico que varia de 1 a 7 que

representa a classe ou grupo biológico do animal, aqui o autor da base determina um desses valores para atribuir o animal a um grupo de especies.

2. Preprocessamento

As principais alterações feitas na base de dados foram em relação a todas as colunas que possuíam dados do tipo booleano, os valores continuaram booleanos, no entanto, foram substituídos os valores 1 pelo nome de suas colunas do tipo texto e os valores 0 pelo atributo vazio, que não representa nada. Essas alterações foram necessárias, pelo fato do algoritmo retornar e trabalhar com atributos texto, não isentando outros tipos de dados, mais para melhor visualização dos resultados nesse trabalho foi definido assim.

Outras alterações feitas, foram em relação as colunas '*name*' e '*type*'. Esses atributos foram excluídos da analise do algoritmo, por se tratarem de informações sem muita relevância para a tarefa de encontrar associações entre atributos.

Por fim, a coluna '*legs*' foi binarizada e convertida para mais novas seis colunas, 'No Legs', '2 Legs', '4 Legs', '5 Legs', '6 Legs', '8 Legs'. Mais uma vez, o mesmo processo realizado com as colunas booleanas, foi repetido nessas novas.

3. Algoritmo Apriori

O algoritmo Apriori é um dos mais populares na tarefa de encontrar associações entre atributos na área de mineração de dados e Web Mining.

Seu funcionamento consiste basicamente em realizar combinações entre atributos, afins de, encontrar padrões ou repetições nas instancias. Geralmente, é atribuído dois indicadores a cada verificação, o Suporte e a Confiança.

O suporte é dado pela relação 1:

$$\frac{N\acute{u}mero\ de\ Repeti\tilde{c}oes\ do\ Padr\tilde{a}o}{N\acute{u}mero\ de\ Instancias} \qquad (1)$$

A confiança é dada pela relação 2:

$$\frac{Número\ de\ Repetições\ do\ Padrão}{Número\ de\ Repetições\ do\ Padrão\ Anterior} \quad (2)$$

Em muitos casos, é exigido dado percentual de suporte e confiança, para se ter um certo nível de credibilidade dos resultados.

Ao final do processo, o algoritmo retorna os atributos associados.

4. Resultados

O algoritmo foi executado com os parâmetros de Suporte Minimo em 30% e Fator de Confiança em 97%, também foi solicitado que só retornasse padrões com no minimo quatro atributos. Os resultados podem ser vistos na tabela 2:

Padrão	**Suporte Minimo (%)**	**Taxa de Confiança (%)**
4 Legs, backbone, hair, milk	30.693069	100.000000
4 Legs, breathes, hair, milk	30.693069	100.000000
backbone, breathes, hair, milk	38.613861	100.000000
backbone, breathes, hair, toothed	37.623762	97.435897
backbone, breathes, milk, toothed	39.603960	97.560976
backbone, hair, milk, toothed	37.623762	97.435897
backbone, milk, tail, toothed	33.663366	97.142857
breathes, hair, milk, toothed	37.623762	97.435897
breathes, milk, tail, toothed	33.663366	97.142857
4 Legs, backbone, breathes, hair, milk	30.693069	100.000000
backbone, breathes, hair, milk, toothed	37.623762	97.435897
backbone, breathes, milk, tail, toothed	33.663366	97.142857

Tabela 2. Resultado da Execução do Algoritmo Apriori

Pode ser visto, que alguns padrões atingiram 100% de confiabilidade, sendo o 10º padrão o mais relevante por ser capaz de relacionar cinco atributos e obter 100% de confiança. Dessa maneira, podemos perceber que animais com '4 Patas', 'Vertebrado', 'Que Respiram' e 'Tem Pelos', para está base de dados também 'Produzem Leite'.

Outra informação relevante para se entender o algoritmo Apriori, é seu tempo de execução. Para o resultado obtido na tabela 2 o algoritmo teve que realizar 2.095.590 iterações na base. Levando em consideração que a base selecionada possui 101 instancias e 21 atributos, ou seja, relativamente pequena, o tempo de execução do algoritmo se mostra um problema, tendo em vista que, a cada iteração o método realiza diversas rotinas aumentando mais ainda seu tempo de espera.

5. Suporte à Decisão

Como pode ser visto, essa técnica é capaz de auxiliar nos mais variados seguimentos, desde relacionar produtos comprados juntos em um supermercado, à buscar causas para doenças onde sintomas estão associados. Dessa forma, os processos de tomada de decisão podem buscar em métodos desse tipo, suporte com alta taxa de confiabilidade, auxiliando cada vez mais profissionais na tomada de decisões.

6. Referencias

- 200_Success, user3084006. **Apriori algorithm using Pandas.** <http://codereview.stackexchange.com/questions/38101/apriori-algorithm-using-pandas>. Acessado em: 09/11/2015.
- Richard Forsyth. **Zoo Data Set.** <https://archive.ics.uci.edu/ml/datasets/Zoo>. Acessado em: 07/11/2015.

7. Apêndices

Atributos	Descrição
Name	Nome do Animal
Hair	Possui Pelos
Feathers	Possui Penas
Eggs	Produz Ovos
Milk	Produz Leite
Airborne	Voa
Aquatic	Nada
Predator	É predador
Toothed	Possui Dentes
Backbone	É Vertebrado
Breathes	Respira
Venomous	É Venenoso
Fins	Possui Barbatanas
Legs	Número de Patas
Tail	Possui Cauda
Domestic	É domesticável
Catsize	Catsize
Type	1 - aardvark, antelope, bear, boar, buffalo, calf, cavy, cheetah, deer, dolphin, elephant, fruitbat, giraffe, girl, goat, gorilla, hamster, hare, leopard, lion, lynx, mink, mole, mongoose, opossum, oryx, platypus, polecat, pony, porpoise, puma, pussycat, raccoon, reindeer, seal, sealion, squirrel, vampire, vole, wallaby,wolf. 2 - chicken, crow, dove, duck, flamingo, gull, hawk, kiwi, lark, ostrich, parakeet, penguin, pheasant, rhea, skimmer, skua, sparrow, swan, vulture, wren. 3 - pitviper, seasnake, slowworm, tortoise, tuatara. 4 - bass, carp, catfish, chub, dogfish, haddock, herring, pike, piranha, seahorse, sole, stingray, tuna. 5 - frog, frog, newt, toad 6 - flea, gnat, honeybee, housefly, ladybird, moth, termite, wasp. 7 - clam, crab, crayfish, lobster, octopus, scorpion, seawasp, slug, starfish, worm.

Tabela 1. Atributos da Base de dados do Zoológico.

```
# -*- coding: utf-8 -*-

from itertools import combinations
import numpy as np
import pandas as pd
import random

def apriori(data, support, minlen, confianca):
    ts = pd.get_dummies(data.unstack().dropna()).groupby(level=1).sum()
    collen, rowlen = ts.shape
    pattern = []
    iterations = 0
    for cnum in range(minlen, rowlen+1):
        for cols in combinations(ts, cnum):
            patsup = ts[list(cols)].all(axis=1).sum()
            a = patsup
            patsup = float(patsup)/collen
            aux = list(cols)
            del aux[-1]
            confiance = 0
            b = ts[aux].all(axis=1).sum()
            if b != 0:
                confiance = float(a)/b
            pattern.append([",".join(cols), patsup*100, confiance*100])
            iterations += 1
    sdf = pd.DataFrame(pattern, columns=["Padrao", "Suporte", "Confianca"])
    results = sdf[sdf.Suporte >= support]
    results = results[results.Confianca >= confianca]
    print (results)
    print ('Iterações: ' + str(iterations))

def legsFunction(legs, n, s):
            if legs == n:
                        return s
            return np.nan

zoo = pd.read_csv('zooOriginal.csv', sep=',', header=None)
zoo.columns =
['name','hair','feathers','eggs','milk','airborne','aquatic','predator','toothed','backbone','breathes','venomou
s','fins','legs','tail','domestic','catsize','type']
zoo = zoo.drop(['name'], axis=1)
zoo = zoo.drop(['type'], axis=1)
legs = zoo['legs']
zoo = zoo.drop(['legs'], axis=1)
for i in list(zoo.columns.values):
            zoo[i] = zoo[i].replace(1, i)
            zoo[i] = zoo[i].replace(0, np.nan)
zoo['No Legs'] = legs.apply(lambda x : legsFunction(x, 0, 'No Legs'))
zoo['2 Legs'] = legs.apply(lambda x : legsFunction(x, 2, '2 Legs'))
zoo['4 Legs'] = legs.apply(lambda x : legsFunction(x, 4, '4 Legs'))
zoo['5 Legs'] = legs.apply(lambda x : legsFunction(x, 5, '5 Legs'))
```

```
zoo['6 Legs'] = legs.apply(lambda x : legsFunction(x, 6, '6 Legs'))
zoo['8 Legs'] = legs.apply(lambda x : legsFunction(x, 8, '8 Legs'))
#print (zoo)
apriori(zoo, 30, 4, 97)
```

Código Fonte. Implementação em Python