Mineração de Dados
Trabalho Prático 1

Prof. Wagner Meira Jr.

Data de entrega: 1º de novembro de 2012

---

1 Descrição do Problema

O governo brasileiro, cansado de tantas greves, realizou um censo de seus habitantes para tentar traçar importantes relações a partir do perfil dos trabalhadores. Depois de entender a classe trabalhadora melhor, o governo almeja conseguir traçar um plano de atividades específico à esse setor para evitar futuras greves. Entretanto, como o orçamento está apertado, eles resolveram economizar e contrataram você, um dedicado estudante com habilidades em Mineração de Dados para fazer o serviço.

O governo desenvolveu um questionário com diversas perguntas direcionadas à profissão, como salário, horário de trabalho, escolaridade, cargo, etc., e algumas pessoais como idade, sexo, estado civil, etc. Mas, após o levantamento da pesquisa, o governo se deparou com uma enorme quantidade de dados que, na forma em que estão, são inúteis para tentar identificar qualquer relação importante sobre os trabalhadores.

É aí que você entra na história. O governo te responsabilizou de extrair toda e qualquer informação relevante do censo obtido, e forneceu a você a base de dados levantada durante o período de entrevistas.

Sua tarefa é, portanto, minerar padrões e regras frequentes do último censo brasileiro destinado aos trabalhadores.

2 Avaliação

Você deve implementar o algoritmo **Apriori** de forma a extrair informações importantes da base de dados fornecida.

Um problema muito comum que ocorre quando mineramos padrões frequentes é o de ignorar itens infrequentes, porém interessantes. Você deve propor e implementar uma modificação no seu algoritmo de forma a tratar esse problema, que é conhecido como **Rare Item Problem**.

Analise as soluções geradas (mostre regras interessantes, varie o suporte e a confiança, etc).

3 Formato

Os arquivos necessários estão disponíveis em [1]. A pasta contém o arquivo com as informações sobre o censo a serem mineradas.

3.1 Formato de Entrada

O programa receberá um arquivo de entrada. Cada linha desse arquivo representa uma instância, na forma:

```
<atributo>=<valor> <atributo>=<valor> ...
```

Um exemplo do arquivo poderia ser:

```
age=middle-aged education=Bachelors race=White sex=Male country=United-States salary<=50K
age=senior education=Bachelors race=White sex=Male country=United-States salary<=50K
age=middle-aged education=HS-grad race=White sex=Male country=United-States salary<=50K
age=senior education=11th race=Black sex=Male country=United-States salary<=50K
age=young education=Bachelors race=Black sex=Female country=Cuba salary<=50K
age=middle-aged education=Masters race=White sex=Female country=United-States salary<=50K
age=middle-aged education=9th race=Black sex=Female country=Jamaica salary<=50K
age=senior education=HS-grad race=White sex=Male country=United-States salary>50K
age=middle-aged education=Masters race=White sex=Female country=United-States salary>50K
age=middle-aged education=Bachelors race=White sex=Male country=United-States salary>50K
```

O programa deve ainda receber como entrada, obrigatoriamente: o parâmetro s, que corresponde ao suporte mínimo a ser utilizado, e também o parâmetro c, que é a confiança mínima das regras a serem geradas. A execução deve seguir o formato:

```
$> comando -i <arquivo entrada> -s <suporte mínimo> -c <confiança mínima>
```

Você pode incluir outros parâmetros, caso sinta necessidade. Entretanto, eles devem ser opcionais, logo, escolha um valor *default* para que, apenas o com o comando acima, seu programa seja executado corretamente. Não se esqueça de especificar os novos parâmetros no arquivo README.txt.

3.2 Formato de Saída

A saída deve conter os itemsets frequentes obtidos, um por linha, ordenados inversamente pelos suportes, separados em grupos correspondentes ao tamanho dos itemsets. Depois dos itemsets frequentes, deve vir a impressão das regras obtidas, ordenadas inversamente pela confiança de cada uma. Em caso de empate, itens e regras devem ser ordenados alfabeticamente. Tanto o suporte quanto a confiança devem conter 3 casas decimais.

A saída deve ser feita conforme o modelo abaixo:

```
Itemsets of size 1
<item> <suporte>
...
<item> <suporte>

⋮

Itemsets of size <X>
<X itens separados por ‘,’> <suporte>

RULES
<um ou mais itens separados por ‘,’> -> <um ou mais itens separados por ‘,’> <confiança>
```

Para os exemplos de entrada acima, usando $s = 0.7$ e $c = 0.8$, a saída deveria ser:

```
Itemsets of size 1
country=United-States 0.800
race=White 0.700
salary<=50K 0.700

Itemsets of size 2
country=United-States,race=White 0.700

RULES
race=White -> country=United-States 1.000
country=United-States -> race=White 0.875
```

4 O que entregar

Você deverá entregar o código com a implementação, juntamente com um arquivo README.txt contendo, resumidamente, os comandos necessários para a execução do seu programa. O comando contido neste arquivo será o utilizado durante a correção e, se ele falhar, serão desconsiderados os pontos da parte de implementação. Logo, **confira bem** se os comandos estão certos.

Você poderá implementar o código na linguagem de sua escolha, mas **não** pode utilizar nenhuma biblioteca já implementada, nem ferramenta de mineração de dados existente para tal.

As implementações devem ser testadas em uma das máquinas de graduação do DCC de livre acesso via acesso remoto. Alguns exemplos de máquinas:

```
cipo.grad.dcc.ufmg.br
claro.grad.dcc.ufmg.br
```

Você deve ainda submeter a documentação do trabalho, em formato *pdf*, com no máximo 10 páginas. A documentação deve abordar, pelo menos, os seguintes pontos:

- Uma breve descrição do algoritmo implementado.
- A ordem de complexidade dele.
- Uma análise de como o algoritmo se comporta quando variamos os parâmetros e o tamanho da entrada.
- Proposta e implementação de uma solução para o *Rare Item Problem*.
- Análise da base de dados fornecida.
- Discussão sobre a qualidade da solução.
- A máquina na qual seu tp foi testado.

Crie uma pasta no formato {*seu login*}*_tp1.tar.gz*, contendo apenas o código fonte, o arquivo README.txt e o *pdf* da documentação. Não inclua nenhum executável. Submeta o *.tar.gz* no moodle, e entregue a documentação impressa na secretaria.

5 Referências

1. `www.dcc.ufmg.br/~sara/mineracao/tp1_data.tar.gz`