22/10/2020

Primeiro Trabalho de Programação I

Prof. Flávio Miguel Varejão

I. Descrição do Problema

Classificação de dados é o problema mais comum na área de aprendizado de máquina. Esse problema consiste em prever a classe de um objeto dentre um conjunto pré-determinado de possíveis classes com base nas características deste objeto.

Formalmente, dado um conjunto de dados X com N objetos $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, sendo que cada ponto $\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{id}]^t$ possui d características e uma classe C_i, pertencente ao conjunto de classes $\{C_1, \ldots, C_K\}$, deseja-se criar um classificador $H(\mathbf{x}_j)$ capaz de prever a classe C_j do objeto $\mathbf{x}_j$.

A idéia deste trabalho é implementar duas técnicas simples de construção automática de classificadores e avaliar seu desempenho em bases de dados de classificação. Para avaliar o desempenho do classificador, o conjunto de dados X é dividido em dois subconjuntos: um de treinamento Tr e outro de teste Ts. O conjunto de treinamento Tr é usado para definir o classificador $H(\mathbf{x}_j)$ e, para isso, são utilizados os objetos e suas classes. O conjunto de teste Ts é usado para avaliar o desempenho do classificador $H(\mathbf{x}_j)$ criado no treinamento. Neste caso, os objetos são fornecidos para o classificador sem a informação de sua classe. O classificador prevê a classe e um avaliador de desempenho compara a classe predita com a classe real do objeto para avaliar se o classificador acertou ou não.

A primeira técnica utilizada no trabalho será o tradicional algoritmo do vizinho mais próximo. Dado o objeto $\mathbf{x}_j$ que se deseja classificar, o algoritmo busca no conjunto de treinamento o exemplo $\mathbf{x}_i$ mais semelhante a $\mathbf{x}_j$ e atribui a $\mathbf{x}_j$ a classe de $\mathbf{x}_i$. Neste algoritmo será usada a distância Euclideana $|| x_i - x_j ||$ como métrica de distância. Ela é calculada pela expressão:

$$|| x_i - x_j || = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \ldots + (x_{id} - x_{jd})^2}$$

A segunda técnica utilizada será o algoritmo do classificador por centróides. Usa-se todos os exemplos de uma classe do conjunto de treinamento para determinar o centróide desta classe. Determina-se o centróide de cada uma das classes. Dado o objeto $\mathbf{x}_j$ que se deseja classificar, o algoritmo busca no conjunto de centróides aquele mais próximo do exemplo $\mathbf{x}_j$ e atribui a ele a classe do centróide mais próximo. Também aqui será usada a distância Euclideana.

O centróide $\mu_j = [\mu_{j1}, \mu_{j2}, \ldots, \mu_{jd}]^t$ é o ponto representativo da classe C_j e é calculado como o centro de massa do grupo de exemplos da classe C_j e no conjunto de treinamento Tr:

$$\mu_\mathbf{j} = \frac{1}{n_j} \sum_{\mathbf{x_i} \in C_j} \mathbf{x_i}$$

onde n_j é o total de objetos pertencentes a classe C_j no conjunto de treinamento.

A base de dados será lida de um arquivo no formato csv (comma separated values) na qual cada linha representa um objeto da base. O nome do arquivo será lido da entrada padrão. Todas as características do objeto são representadas por números ponto flutuante. A classe do objeto é uma string e corresponde ao último dado da linha. Para a divisão dos objetos nos conjuntos de treinamento e teste, será lido da entrada padrão um valor que indica o percentual do total de exemplos usado para teste. A escolha dos objetos para compor o conjunto de teste será feita de maneira aleatória a partir de uma semente também lida da entrada padrão durante a execução do programa.

Os resultados da avaliação dos classificadores será apresentado em termos de acurácia e da matriz de confusão (ver https://pt.wikipedia.org/wiki/Matriz_de_confusão). A acurácia corresponde ao percentual de acertos obtidos pelo classificador, isto é, o resultado da multiplicação de 100 pela divisão do número de exemplos de teste preditos corretamente pelo total de exemplos de teste. A acurácia dos dois classificadores será apresentada na saída padrão sempre com duas casas decimais com arredondamento na segunda casa. A matriz de confusão será apresentada em um arquivo texto de saída cujo nome será lido da entrada padrão.

II. Especificação do Sistema

Funcionalidades a serem implementadas:

*1.*Leitura do nome do arquivo de entrada, do nome do arquivo de saída, do percentual de exemplos de teste e da semente aleatória da entrada padrão.
*2.*Leitura da base de dados do arquivo csv de entrada.
*3.*Separação dos conjuntos de treinamento e teste.
*4.*Cálculo dos centróides das classes.
*5.*Classificação dos exemplos de teste usando o classificador vizinho mais próximo.
*6.*Apresentação na saída padrão da acurácia do classificador vizinho mais próximo.
*7.*Gravação da matriz de confusão do classificador vizinho mais próximo no arquivo de saída.
*8.*Classificação dos exemplos de teste usando o classificador por centróides.
*9.*Apresentação na saída padrão da acurácia do classificador por centróides.
*10.*Gravação da matriz de confusão do classificador por centróides no arquivo de saída.

Formato dos Dados do Sistema:

```
percentual de teste:          inteiro positivo
semente:                      inteiro positivo
características dos objetos:  ponto flutuante
classe:                       string
```

Fórmula para determinar número de exemplos de teste:

```
nteste = truncamento((percentual/100)*nTotal)

onde
      percentual é o valor no intervalo (0,100) lido da entrada padrão
      nTotal é o número total de exemplos da base
```

```
      truncamento é uma função que pega a parte inteira de um decimal
      nteste é o número de exemplos do conjunto de teste
```

Funções de Haskell para escolha randomizada de exemplos de teste:

```
import System.Random
mkStdGen semente
randomRs (1, ntotal) gerador

onde
      semente é um valor inteiro lido da entrada padrão
      ntotal é o número total de exemplos da base
      randomRIO gera um número inteiro aleatório no intervalo (1,
ntotal) e retorna como um IO(Int)

Observe que randomRIO será chamado sucessivamente até que tenham sido
escolhidos os nteste exemplos do conjunto de teste.
```

Os exemplos seguintes são apenas ilustrativos dos formatos de entrada e saída e não existe correspondência entre os seus dados.

Exemplo de formato de arquivo de entrada:

```
7,5.4,6.32,9,classe1
17,32.3,5,9.99,classe2
33,54,5.6,65.8,classe2
77.7,33.4,98,7.56,classe1
8.9,5.8,6,9,classe1
```

Exemplo de formato de arquivo de saida:

```
vizinho mais próximo:
72,28
15,35

centroides
80,20
30,20
```

Exemplo de formato de interação do programa com o usuário:

```
Forneca o nome do arquivo de entrada: base.csv
Forneca o nome do arquivo de saida: confusao.txt
Forneca o percentual de exemplos de teste: 30
Forneca o valor da semente para geracao randomizada: 42
Acuracia(vizinho): 71.33%
Acuracia(centroide): 66.67%
```

III. Requisitos da implementação

- Modularize seu código adequadamente.
- Crie códigos claros e organizados. Utilize um estilo de programação consistente, Comente seu código.

- Os arquivos do programa devem ser lidos e gerados na mesma pasta onde se encontram os arquivos fonte do seu programa.

IV. Condições de Entrega

O trabalho deve ser feito individualmente e submetido por e-mail até as 23:59 horas da data limite especificada na Atividade Primeiro Trabalho Computacional na nossa sala de aula virtual. O trabalho deve ser submetido em um arquivo zip com o nome PG_1_TRABALHO_1_NomedoAluno_SobrenomedoAluno.zip. O arquivo principal (o que contém o main do trabalho) obrigatoriamente deve estar com o nome "main". Note que a data limite já leva em conta um dia adicional de tolerância para o caso de problemas de submissão via rede. Isso significa que o aluno deve submeter seu trabalho até no máximo um dia antes da data limite. Se o aluno resolver submeter o trabalho na data limite, estará fazendo isso assumindo o risco do trabalho ser cadastrado no sistema após o prazo. Em caso de recebimento do trabalho após a data limite, o trabalho não será avaliado e a nota será ZERO. Aluno que receber zero por este motivo e vier pedir para o professor considerar o trabalho estará cometendo um ato de DESRESPEITO ao professor e estará sujeito a perda adicional de pontos na média.

V. Data de Entrega: 15/11/2020

A correção/revisão dos trabalhos será marcada posteriormente.

VI. Avaliação

Os trabalhos terão nota zero se:

A data de entrega for fora do prazo estabelecido;

O trabalho não compilar;

O trabalho não gerar o arquivo com o resultado e formato esperado;

For detectada a ocorrência de plágio.

Observação importante

Caso haja algum erro neste documento, serão publicadas novas versões e divulgadas erratas em sala de aula. É responsabilidade do aluno manter-se informado, freqüentando as aulas ou acompanhando as novidades na página da disciplina na Internet.