

MAC460 Aprendizagem Computacional - Tarefa 2

Fernando Omar Aluani (NUSP: 6797226)

September 7, 2012

1 Execução e Especificação do Programa

Para executar o programa, basta rodar

`./tarefa2.py <treino> <teste>`

Onde:

- *<treino>* é o caminho para o arquivo que contém o conjunto de dados que será usado para treinamento.
- *<teste>* é o caminho para o arquivo que contém o conjunto de dados que será usado para testar os classificadores.

O programa aceita conjuntos de dados de qualquer tamanho, mas espera que ele sigam o mesmo padrão do *gerador.py*: "valorObservado classe". Ele também supõe que as 3 distribuições estatísticas possíveis são uniforme contínua (*uniform*), normal (*norm*) e exponencial (*expon*), que são as que o *gerador.py* usa.

Com os dados de treinamento, o programa estima os parâmetros das 3 distribuições estatísticas para as 3 classes nos 27 casos possíveis de combinações. Com os dados de testes, o programa testa cada um dos 27 classificadores e marca se ele acertou ou errou a classificação, e então imprime a "pontuação" de cada classificador nos testes e qual deles foi o que se saiu melhor. Quando há mais de um classificador empatado com a maior pontuação, o programa escolhe um desses aleatoriamente como o se saiu melhor.

2 Resultado dos Testes

CARALHO

Eis os resultados dos testes da execução do programa para um conjunto de amostras de treinamento e outro conjunto para testes gerados pelo *gerador.py*, lembrando que a pontuação é $(acertos)/(totaldetestes)$, e consequentemente $(totaldetestes - acertos) = (erros)$:

```

Classifier(uniform/uniform/uniform) score: 47/60
Classifier(uniform/uniform/  norm) score: 48/60
Classifier(uniform/uniform/  expon) score: 29/60
Classifier(uniform/  norm/uniform) score: 46/60
Classifier(uniform/  norm/  norm) score: 47/60
Classifier(uniform/  norm/  expon) score: 27/60
Classifier(uniform/  expon/uniform) score: 39/60
Classifier(uniform/  expon/  norm) score: 40/60
Classifier(uniform/  expon/  expon) score: 19/60
Classifier(  norm/uniform/uniform) score: 48/60
Classifier(  norm/uniform/  norm) score: 48/60
Classifier(  norm/uniform/  expon) score: 35/60
Classifier(  norm/  norm/uniform) score: 47/60
Classifier(  norm/  norm/  norm) score: 47/60
Classifier(  norm/  norm/  expon) score: 35/60
Classifier(  norm/  expon/uniform) score: 39/60
Classifier(  norm/  expon/  norm) score: 39/60
Classifier(  norm/  expon/  expon) score: 34/60
Classifier(  expon/uniform/uniform) score: 47/60
Classifier(  expon/uniform/  norm) score: 48/60
Classifier(  expon/uniform/  expon) score: 47/60
Classifier(  expon/  norm/uniform) score: 46/60
Classifier(  expon/  norm/  norm) score: 47/60
Classifier(  expon/  norm/  expon) score: 47/60
Classifier(  expon/  expon/uniform) score: 39/60
Classifier(  expon/  expon/  norm) score: 40/60
Classifier(  expon/  expon/  expon) score: 46/60

```

Como é possível notar, o classificador "certo" (aquele que tem a mesma distribuição das classes geradas pelo *gerador.py*, expon/uniform/norm) fez 48 acertos, sendo o classificador que menos errou junto com outros 3 classificadores que também só erraram 12 testes.