UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
INF01048 INTELIGÊNCIA ARTIFICIAL 2015/1

Tutor: Edigleison F. Carvalho
Email: edigleison.carvalho@inf.ufrgs.br

Classificador NB para Dígitos Manuscritos

1. Objetivos

O objetivo deste trabalho prático é gerar e avaliar um modelo Naive Bayes para classificação de imagens de dígitos manuscritos. O conjunto de dados será disponibilizado na página da disciplina.

2. Descrição

O conjunto de dados a ser utilizado neste trabalho refere-se ao conjunto USPS, o qual contém 9298 imagens de tamanho 16x16 em tons de cinza digitalizadas de dígitos manuscritos (de 0 a 9) retirados de envelopes do serviço postal americano (*U.S. Postal Service*). As imagens estão na forma de vetores com 256 posições, uma para cada pixel da imagem, e cada linha do arquivo (.csv) contém um exemplo de dígito. O desenvolvimento de classificadores eficientes para dígitos é importante para automatizar o serviço de separação dos envelopes para os seus devidos destinos (através do código postal). Exemplos de imagens do USPS são apresentados na Figura 1.

Figura 1: amostras do conjunto de dados USPS.

3. Metodologia

Um subconjunto dos dados será atribuído a cada aluno. Este deverá implementar o algoritmo Naive Bayes para estes dados usando obrigatoriamente a densidade de probabilidade condicional gaussiana, isto é, a verossimilhança (*likelihood*) deve ser calculada usando a função de densidade de probabilidade da distribuição normal (gaussiana). Isso é necessário para estes dados, pois os valores dos pixels são reais e não discretos. Os cálculos devem ser realizados utilizando logaritmos para evitar erros numéricos como visto em sala de aula. E a classificação de uma dada imagem de teste deve ser feita por **MAP.** Após a implementação do classificador, o aluno deve avaliar seu modelo através de validação cruzada (10-*fold*) e para cada iteração da validação cruzada deve-se calcular as seguintes métricas para cada classe/dígito: precisão, *recall*, medida-F. Além disso, deve-se calcular a acurácia do modelo nos dados de teste e a tabela de confusão para cada iteração. Após a validação cruzada deve-se calcular o valor médio de todas métricas

obtidas durante a validação cruzada e então preencher uma tabela de resultados semelhante a tabela abaixo:

Classe	Precisão média	Recall médio	Medida-F média
0			
1			
2			
3			
4			
5			
6			
7			
8			
9			

Dica: Primeiro calcule a tabela de confusão na iteração da validação cruzada e a partir dos dados da tabela extraia todas as métricas solicitadas. Considere que as linhas da matriz de confusão representam as verdadeiras hipóteses/classes e as colunas representam as hipóteses/classes preditas. Note que a precisão P_i da classe *i* num problema multiclasse (com mais de 2 classes) é dado por:

$$P_i = \frac{M_{ii}}{\sum_j M_{ji}}$$

onde M_{ij}é o elemento na posição (i,j) da matriz de confusão *M*. E o recall R_i da classe *i* é dado por:

$$R_i = \frac{M_{ii}}{\sum_j M_{ij}}$$

A matriz de confusão média é obtida calculando a média de cada elemento da matriz utilizando os 10 valores obtidos para cada elemento nas iterações da validação cruzada.

O aluno pode implementar o algoritmo na sua linguagem preferida, porém não pode utilizar implementações prontas de bibliotecas ou *toolboxes*. É recomendado o uso da linguagem Python pela facilidade de manipulação de vetores multidimensionais com a biblioteca numpy.

4. Relatório

Deve ser entregue, juntamente com o código-fonte e executável, um relatório de no máximo 2 páginas contendo uma descrição das ferramentas utilizadas, bem como os resultados obtidos (como descrito na seção 3 – metodologia). **Atenção**: caso necessário, o código-fonte pode vir acompanhado de um arquivo Readme.txt explicando como executar o código.