Projeto 2 - Classificadores

Lukas Lorenz de Andrade
Departamento de Ciências da Computação
UnB
Matrícula: 16/0135109

Gustavo Costa
Departame de Ciência da Computação
UnB
Matrícula: 14/0142568

***Resumo*—Ao tratar-se de algorítimos de Inteligência Artificial, a ferramenta Matlab, ou sua versão *opensource* Octave, fornecem uma base para a implementação de soluções de diversos problemas da atualidade. Neste trabalho, por meio dessas ferramentas, será desenvolvido um programa capaz de classificar os pixels das 10 imagens de plantação usadas como *dataset*. As análises comparativas de classificação serão feitas usando os métodos LDA (*Linear Discriminant Analysis*), QDA (*Quadratic Discriminant Analysis*), Baysiano e Knn. Já as regiões serão divididas em R1) Folhas verdes; R2) Solo descoberto; R3) Folhas ressecadas, amareladas e frutos amarelos e vermelhos expostos na foto; R4) Sombras e/ou regiões indeterminadas. Assim, este trabalho exemplifica a versatilidade dessa ferramenta aplicada aos métodos de classificação de Inteligência Artificial.**

***Index Terms*—Inteligência Artificial, Classificadores, Imagens de Plantações de Tomate, MatLab R2017a.**

I. Introdução

O aprendizado humano se baseia na associação de títulos aos novos conceitos incorporados, ou seja, é por meio de uma aprendizagem supervisionada por reforço. Além disso, a classificação das abstrações, seja por exclusão (diferenciando o aprendido do resto), seja por identificação das partes, é a forma mais comum de aprendizado durante a formação do indivíduo.

Ademais, dada a atual necessidade de automatização de processos e da precisão dos sistemas em detrimento do erro humano, os métodos de Inteligência Artificial tem-se mostrado uma solução viável para diferentes problemas enfrentados atualmente. Sendo assim, os métodos de classificação supervisionado podem ser usados em vários contextos em que os parâmetros são conhecidos, ou seja, pode-se dividir os dados em classes.

Para tanto, a fim de se evitar erros humanos no processo de classificação de frutos, com o interesse de identificar doenças, por exemplo, pode-se usar diversos classificadores para tal finalidade, como LDA (*Linear Discriminant Analysis*), QDA (*Quadratic Discriminant Analysis*), Baysiano e Knn (usados neste trabalho), dada as características dos rôtulos tratados. Assim, a escolha do classificador traduz-se na qualidade dos resultados obtidos, devido a vários fatores, como dispersão dos dados, correlação entre variáveis, entre outros.

Dessa forma, o classificador Knn se baseia na distância do ponto a qual se quer prever dos seus "k"vizinhos conehcidos, ou seja, dado um ponto não conhecido, classifica-se este pela classe dos seus vizinhos mais próximos. Neste trabalho, será usado a distância euclidiana como métrica de distância para o problema de classificação das regiões R1 a R4 com base nos pixels extraídos.

$$d(x, y) = \sqrt{(xi - xf)^2 + (yi - yf)^2} \tag{1}$$

Além disso, os métodos LDA e QDA são feitos utilizando-se uma a projeção de características do espaço vetorial (n dimensional) em subespaços (de ordem k menor que n-1) sem perder o fator discriminante, conforme explicado no artigo *Linear Discriminant Analysis*. O algorítimo aplicado ao LDA pode ser explicado sucintamente no tópico *Summarizing the LDA approach in 5 steps* do mesmo artigo. Ambos são derivados de modelos probabilísticos clássicos, como a regra de Bayes da probabilidade condicional. Contudo, o QDA se baseia na distribuição multivariada Gaussiana , conforme no artigo *Linear and Quadratic Discriminant Analysis*.

Por fim, tem-se o classificador baysiano, cuja análise de probabilidade se baseia na interseção de duas distribuições Gaussianas em torno das suas respectivas médias. Tem-se como uma das distribuições é a de probabilidade a posteriori e a probabilidade do dado analisado, como exemplificado no artigo *Naive Bayes*.

Logo, este trabalho consiste na comparação dos quatro métodos supracitados no contexto de classificação das regiões pelo valor do pixel correspondente, o qual será explicitado nas sessões II, III e IV. Na sessão II, será apresentada a metodologia usada para adquirir o resultado final de cada etapa, mostrado na sessão III, junto com a análise dos dados obtidos. Por fim, na sessão IV será apresentada as considerações finais dados os parâmetros assumidos, assim como os resultados obtidos.

A. Objetivos

O objetivo desse trabalho se resume na classificação nas regiões **R1)** Folhas verdes; **R2)** Solo descoberto; **R3)** Folhas ressecadas, amareladas e frutos amarelos e vermelhos expostos na foto; **R4)** Sombras e/ou regiões indeterminadas, das 10 imagens dadas para treinamentos, assim como a contraposição dos desempenhos dos classificadores neste contexto. Entretando, os objetivos específicos do trabalho se resumem a:

(1) Separação do *dataset* manualmente com base nas regiões R1 a R4;

(2) Treinamento dos modelos apresentados com base na reordenação randômica do *dataset*;

Universidade de Brasília

(3) Comparação do desempenhos dos classificadores com base nas matrizes de confusão geradas e acurácia de cada modelo apresentado.

II. Metodologia

O projeto foi desenvolvido usando a plataforma MatLab R2017a. Assim, a metodologia consiste na elaboração de dois programas, um para a seleção de pixels e associação das respectivas regiões manualmente e outro para a classificação com os métodos citados na sessão anterior.

A. Programa Desenvolvido

No primeiro programa (*dataset.m*), abre-se a figura **photo-9-orig.jpg** devido maior variedade de exemplos das classes R1 a R4. Assim, acessa-se os valores RGB para gravação no arquivo *dataset.xlsx*. Além disso, tal programa abre as outras 9 figuras e seleciona o valor RGB de cada pixel destas.

Figura 1. Figura 9 usada como exemplo para o treinamento dos modelos onde se indentifica claramente as 4 classes: **R1)** Folhas verdes; **R2)** Solo descoberto; **R3)** Folhas ressecadas, amareladas e frutos amarelos e vermelhos expostos na foto; **R4)** Sombras e/ou regiões indeterminadas.

No segundo programa, importa-se os valores do *dataset.xlsx* e treina-se os modelos conforme explicitado no tutorial Mathworks usando as variáveis $(X\_training, Y\_traning)$, separando os 100 pixels em 70 para treino e 30 para teste. Após o treinamento dos 4 classificadores, importou-se dos valores RGB das outras 9 imagens e fez-se a classificação dos pixels, analisando-se os erros associados a cada classe.

III. Análise do projeto

Primeiramente, para o Método Knn, com K=5, a predição apresentou uma acurácia de 5,71% com erro prevalecendo exclusivamente na região 1. Foi-se avaliado o método para diferentes K's, tendo-se obtido o menor resultado com o K explicitado acima. Para essas condições, o número de pixels avaliados erroneamente com o método Knn seria de 5.974 pixels. Para este método foi-se utilizada a função $fitcknn$ do MatLab para adicionar os dados de treinamento, que foram 70 pixels selecionados manualmente, a função $crossval$ que faz a crosvalidação dos dados de treino, e a função $predict$ que efetivamente utiliza o método para prever os dados teste.

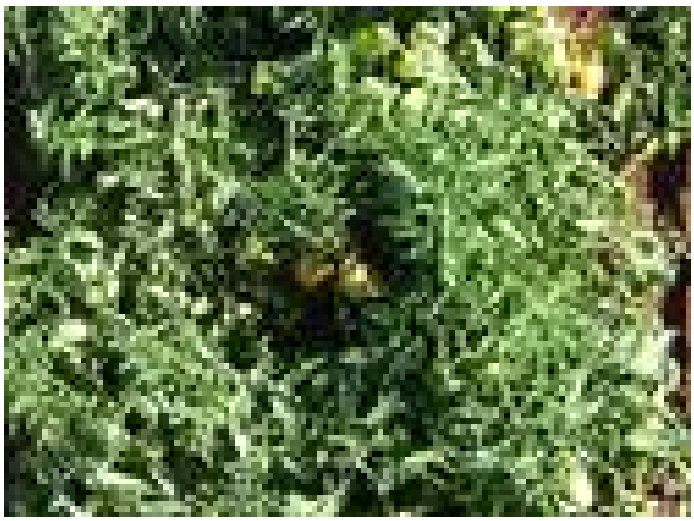

Figura 2. A figura 1 das imagens disponibilizadas exemplifica um modelo de *dataset* ruim para o treinamento dos modelos usados em vista que a variedade das classes R1 a R4 é baixa.

Tabela I
Matriz de confusão do método Knn

	R1	R2	R3	R4
R1	11	0	2	0
R2	0	6	0	0
R3	0	0	6	0
R4	0	0	0	5

Em seguida foi-se avaliado o método de Bayes, onde foi obtida uma acurácia de 17,14%, o que é maior se comparada com o Método Knn. Avaliando-se as 9 imagens do projeto com o erro observado, é de se esperar que o erro nas regiões 1, 2, 3 e 4 (R1, R2, R3 e R4, respectivamente), baseado na matriz de confusão, seja de: $E_{R1} = 46,15\% = 48.288 pixels, E_{R2} = E_{R3} = 16,66\% = 17.437 pixels E_{R4} = 20\% = 20.925 pixels$.

Tabela II
Matriz de confusão do método Baysiano

	R1	R2	R3	R4
R1	7	0	6	0
R2	1	5	0	0
R3	1	0	5	0
R4	0	1	0	4

Para o método LDA obteve-se, para os dados totais $(X\_training + X\_Teste)$ um erro de 3,00%, menor que o Knn.

Para o último método, QDA, obteve-se um erro de 0,00%, o que é avaliado como um erro na execução das funções referentes ao mesmo. Ao se realizar inspeções no script principal, não se obteve soluções para o resultado erroneo, e a parte do script utilizado no projeto foi retirada do item 1 da bibliografia.

Tabela III
MATRIZ DE CONFUSÃO DO MÉTODO LDA

	R1	R2	R3	R4
R1	25	0	0	0
R2	0	25	0	0
R3	0	0	25	0
R4	0	3	0	22

Tabela IV
MATRIZ DE CONFUSÃO DO MÉTODO QDA

	R1	R2	R3	R4
R1	25	0	0	0
R2	0	25	0	0
R3	0	0	25	0
R4	0	0	0	25

IV. CONCLUSÃO

A partir dos dados coletados na sessão anterior, junto as matrizes de confusão e os erros calculados, pode-se considerar o LDA o melhor classificador para os 100 pixels escolhidos manualmente, apresentando um erro relativo de 3% de classificação erronea. Enquanto que o Knn se deu em 4.29% (para K = 5) e o de Bayes em torno de 20%. Quanto a classificação em todas as imagens, o Knn se mostrou como o melhor classificador, dado o erro relativo de classificação menor. Quanto ao erro relativo observado por cada região na matriz de confusão gerada, observou-se que o classificador Knn obteve a maior concentração de erro por região, sendo apresentada somente na região 1, enquanto que nos outros métodos o erro era variado.

Além disso, os parâmetros adotados são coerentes com os apresentados no tutorial [1], onde se tem os métodos aplicados ao *Iris dataset*. Um distinção feita, além da aplicação em questão, é a reprodutibilidade gerada nesse experimento ao se randomizar os dados antes do treinamento dos modelos. Entretanto, um fator a se considerar nos resultados obtidos é a interferência humana quanto a seleção dos dados, podendo ter interferido diretamente na relação dos classificadores, ou seja, o sistema ser viciado (*overfitting*).

REFERÊNCIAS

[1] https://www.mathworks.com/help/stats/examples/classification.html, Acessado em 15 de Abril de 2018.