Classificação de Algarismos Manuscritos Utilizando Algorítimos de Classificação

Khalil Carsten do Nascimento
Departamento de Ciência da Computação
Universidade de Brasília
Brasília, Brasil
khalilcarsten@gmail.com

Renato Avelar Nobre
Departamento de Ciência da Computação
Universidade de Brasília
Brasília, Brasil
rekanobre@gmail.com

***Resumo*—Problemas de classificação são inerentes ao nosso dia a dia, onde muitas vezes é preciso distinguir informações em diferentes aspectos e categorias. Utilizou-se, neste trabalho, dois algoritmos de classificação para tratar o problema de reconhecimento de algarismos indo-arábicos, com o apoio de duas formas de tratamento de imagem, tratamento de Linhas Normalizadas e o Tratamento de Linhas Binárias. O primeiro algoritmo utilizado é o método *K-Vizinhos Mais Próximos*, uma abordagem não-paramétrica para classificações e regressões. Posteriormente foi realizado experimentos com o algoritmo de *Análise de Discriminantes Lineares*. Para abordar o problema foi desenvolvido um sistema em *Python*, que utilizou de apoio bibliotecas como *scikit-learn* e *matplotlib*, junto com o banco de imagens de algarismos MNIST que continha 60.000 imagens de treinamento e 10.000 de teste. O sistema foi testado com diversos conjuntos de experimentos, notou-se que os melhores resultados foram provenientes do algoritmo KNN com o valor de K = 3, e as imagens sem nenhum tratamento. Os tratamentos desenvolvidos para as imagens não se mostraram promissores, necessitando assim a busca de melhores métricas.**

***Index Terms*—KNN, LDA, algoritmos de classificação, aprendizagem de máquina, inteligência artificial, processamento de imagens,**

I. Introdução

Problemas de classificação são inerentes ao nosso dia a dia, onde muitas vezes é preciso distinguir informações em diferentes aspectos e categorias. Por exemplo, milhares de cartas são recebidas pelas agências de correio todos os meses, e a distinção entre os endereços das correspondências pode se tornar um trabalho massivo para os seres humanos, sendo assim necessário um método melhor de classificar grupos de cartas pelo CEP residencial. Visando este contexto, percebe-se uma limitação prévia, é preciso identificar e classificar dígitos numéricos manuscritos a priori. Para realizar tal tarefa de classificação é proposto uma abordagem na qual modelos de aprendizagem de máquina receberiam imagens tratadas de algarismos e classificaram de acordo com seu conhecimento prévio de cada número.

Utilizou-se, neste trabalho, dois algoritmos de classificação para tratar o problema de reconhecimento de algarismos indo-arábicos. O primeiro algoritmo utilizado é o método *K-Vizinhos Mais Próximos*, uma abordagem não-paramétrica para classificações e regressões. Posteriormente foi realizado experimentos com o algoritmo de *Análise de Discriminantes Lineares*, uma abordagem linear para problemas de classificação.

Escolheu-se para utilização o *Banco de Imagens MNIST para Dígitos Manuscritos*, o mesmo é descrito como útil e prático para pessoas que querem experimentar técnicas de aprendizado e métodos de reconhecimento de padrões em dados do mundo real, enquanto gastam esforços mínimos em pré-processamento e formatação [1].

Este relatório é organizado do seguinte modo. A Seção II desenvolve conhecimentos teóricos importantes para o entendimento dos algoritmos de aprendizagem e da proposta implementada; a Seção III explica como a plataforma foi desenvolvida e a estrutura do banco de imagens MNIST; a Seção IV descreve as metodologias de tratamento de imagens implementadas na plataforma; a Seção V mostra os resultados experimentais da bateria de testes realizadas; por ultimo, a Seção V finaliza o relatório, sumarizando os resultados obtidos e sugerindo trabalhos futuros.

II. Algoritmos de Classificação

Em problemas de classificação, onde há a necessidade de separar um certo objeto em diferentes classes, o sistema necessita criar modelos de classificação baseado em rótulos, para poder utilizá-lo para futuras predições.

Na inteligência artificial, aprendizagem de máquina indutiva é o processo de aprendizado proveniente de um conjunto de regras de instâncias, ou de uma forma mais geral, é o processo de criar um classificador para generalizar instancias não conhecidas. [1]. Sendo assim a tarefa do aprendizado supervisionado é usar características X para aprender uma função de classificação ótima, ou seja, uma função que consegue classificar uma nova instancia com o menor erro possível. As funções de classificação podem ser não-paramétricas ou lineares que utilizam cálculo de distancias para estimar suas instancias. É o caso dos algoritmos K-Vizinhos mais Próximos, e de Análise de Discriminantes Lineares respectivamente.

A. Algoritmo de K-Vizinhos Mais Próximos

O algoritmo de K-vizinhos mais próximos, podendo ser referenciado como KNN (do inglês *K-nearest neighbors*), trata-se de um método de classificação não-paramétrico. Utilizando

[1] Disponível em: http://yann.lecun.com/exdb/mnist/

vetores de n dimensões, como entrada de treinamento, o algoritmo calcula a distância entre todos os dados os colocando em uma lista. Após o ordenamento dessa lista, a classe com número de indivíduos mais próximo de K é selecionado como classe de predição.

Na Figura 1 existem duas classes, triângulos e círculos. O símbolo de interrogação representa o elemento a ser classificado. Com o K igual a três o círculo preto representa os 3 indivíduos mas próximos e, devido a maior quantidade, o elemento é classificado como um triângulo.

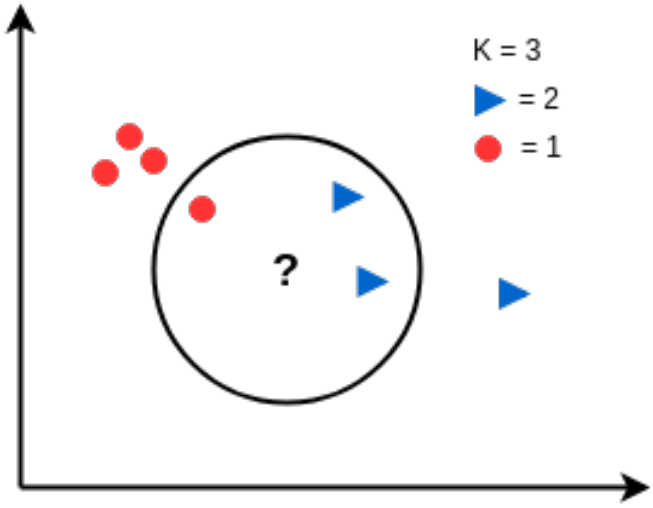

Figura 1. Representação gráfica do K-Vizinhos Mais Próximos

B. Algoritmo de Análise de Discriminantes Lineares

O algoritmo de análise de discriminantes lineares, podendo ser referido como LDA (do inglês, Linear Discriminant Analysis), trata-se de um método de classificação estatístico baseado em definição de parâmetros. Basicamente o algoritmo determina uma equação linear (1) afim de dividir os dados em classes Figura 2. Isso implica em algumas premissas sobre os dados utilizados no algoritmo. Os dados têm que seguir uma distribuição Gaussiana e a variância dos atributos deve ser próxima.

O LDA utiliza da regra de Bayes para classificação, ou seja, entre as K classes ele seleciona a que possui a maior probabilidade para o atributo.

$$\boldsymbol{D} = v_1 X_1 + v_2 X_2 + \ldots + v_i X_i + a \qquad (1)$$

III. Plataforma de Classificação e Banco de Imagens MNIST

Para realizar a plataforma de classificação utilizou-se um conjunto de ferramentas e técnicas.

A. Especificações do Sistema

A estrutura do sistema foi desenvolvida para evitar acoplamento entre os algoritmos de aprendizagem, os algoritmos de tratamento de imagens, e os algoritmos de suporte. O sistema

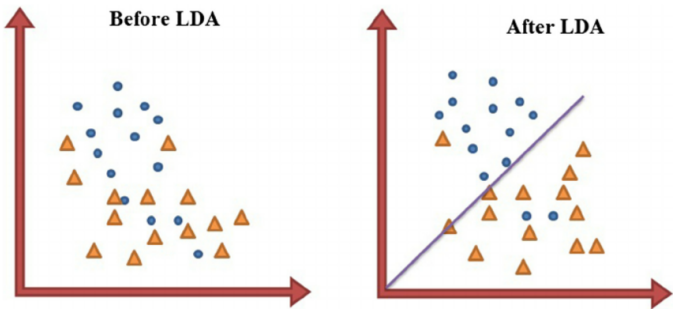

Figura 2. Gráficos de uma classificação por LDA

adaptador é responsável por conciliar o ambiente para respeitar as interfaces requeridas pelo usuário ao realizar entradas pela linha de comando.

Todo o sistema foi construído utilizando a linguagem de programação *Python 3* que é a linguagem usada em milhares de aplicações de negócio no mundo, incluindo diversos sistemas de larga escala [2]. Para o auxilio no desenvolvimento das funções de classificação, foi utilizada uma biblioteca de aprendizagem de máquina denominada *scikit-learn*. A biblioteca, acessível para todos, é um conjunto de métodos simples e eficientes para mineração e analise de dados [3]. Também foi utilizada a biblioteca *matplotlib* para criar gráficos e a matriz de confusão. [4]

B. Banco de Imagens MNIST

O banco de imagens MNIST de algarismos manuscritos, utilizado neste projeto, possui um conjunto de 60.000 imagens de treino e 10.000 imagens de teste. Para todas as imagens, os dígitos foram normalizados por tamanho e centralizados em uma imagem de tamanho fixo.

As imagens resultantes contêm níveis de cinza como resultado da técnica de *anti-aliasing* usada pelo algoritmo de normalização. As imagens centraram-se em uma tamanho de 28x28, calculando o centro de massa dos pixeis e traduzindo a imagem para posicionar este ponto no centro do campo. Desta forma, garantimos que os números estão propriamente centralizados, e que não haverá erros de classificação em questão do posicionamento do número

Junto com o banco de imagens é providenciado também um banco de seus respectivos rótulos para serem utilizados nos sistemas de classificação.

IV. Proposta e Desenvolvimento

Visto os algoritmos citados e o banco de imagens MNIST propomos uma experimentação de três entradas diferentes visando alcançar níveis altos de precisão dos agentes e otimização do tempo de predição dos algarismos manuscritos.

Como primeira e mais básica estrutura das entrada temos os dados brutos sendo apresentados como vetores de 784 elementos com valores definidos entre 0 e 255 (2). Porém

[2]Disponível em: https://www.python.org
[3]Disponível em: http://scikit-learn.org
[4]Disponível em: https://matplotlib.org

vetores muito grandes resultam em complexos cálculos de distância, este presente no algoritmo KNN.

Visando uma melhor performasse ou possivelmente um aumento da precisão propomos dois tratamentos desses dados. Ambos possuem o propósito de reduzir o custo de predição reduzindo o tamanho dos vetores de avaliação sem afetar o padrão dos dados.

$$v = (v_1, v_2, ..., v_n), \quad \text{onde } n = 784 \text{ e } v_n \in [0, 255] \tag{2}$$

A. Implementação do Tratamento de Linhas Binárias

Primeiramente definimos um vetor b (3) este com o mesmo número de elementos de v. Em seguida para cada elemento de b atribuímos o valor da função definida em (4). Assim temos um vetor de valores binários b. Definindo v' como o vetor final do tratamento e abstraindo b para uma matriz de 28 linhas por 28 colunas temos (5). Assim v' será um vetor em cada elemento representa a soma dos 28 elementos de cada coluna da matriz advinda de b.

$$b = (b_1, b_2, ..., b_n), \quad \text{onde n = 784} \tag{3}$$

$$b_n = f(v_n) = \begin{cases} 1, & \text{se } x > 0. \\ 0, & \text{se não.} \end{cases}, \quad \text{onde } b_n \in b \text{ e } v_n \in v \tag{4}$$

$$v' = \left(\sum_{n=(1+(28*0))}^{(28*1)} b_n, \sum_{n=(2+(28*1))}^{(28*2)} b_n, \ldots, \sum_{n=(i+(28*j))}^{(28*j)} b_n \right), \quad \text{onde } i = 1, \ldots, 28 \text{ e } j = 0, \ldots, 27 \tag{5}$$

B. Implementação do Tratamento de Linhas Normalizadas

O vetor v definido em (2) possui elementos com valores de 0 a 255. Tendo isso em vista criamos um vetor l onde o valor atribuído a l_n é gerado pela função (7) recebendo cada v_n e dividindo-o por 255, o máximo valor possível. Em seguida, da mesma maneira que o tratamento anterior, abstraímos o vetor para uma matriz de 28 linhas por 28 colunas e somamos os elementos de cada linha atribuindo a um l'_n mostrado em (8). Agora obtemos w (9) como a soma de todos os valores de l'. Ao final geramos um vetor v' que cada v'_n corresponde a l'_n divido por w.

$$l = (l_1, l_2, ..., l_n), \quad \text{onde n = 784} \tag{6}$$

$$l_n = f(v_n) = \frac{v_n}{255}, \quad \text{onde } l_n \in l \text{ e } v_n \in v \tag{7}$$

$$l' = \left(\sum_{n=(i+(28*0))}^{(28*1)} l_n, \sum_{n=(i+(28*1))}^{(28*2)} l_n, \ldots, \sum_{n=(i+(28*j))}^{(28*j)} l_n \right), \quad \text{onde } i = 1, \ldots, 28 \text{ e } j = 0, \ldots, 27 \tag{8}$$

$$w = \sum_{n=1}^{28} l_n, \quad \text{onde } n = 28 \tag{9}$$

$$v'_n = f(l'_n) = \frac{l'n}{w}, \quad \text{onde } l'_n \in l' \text{ e } v'_n \in v' \tag{10}$$

V. Resultados Experimentais

Com base nos problemas citados na Seção IV, e com os métodos implementados no sistema descritos na Seção III foram elaboradas as seguintes perguntas:

- É possível classificar o conjunto de dados com uma margem de erro menor que 5% usando o Algoritmo KNN? E com o algoritmo LDA
- Qual o melhor valor de K encontrado para utilizar no classificador KNN?
- Dentro das diversas formas de tratamento de dados implementados, qual forma trara o melhor resultado para o Algoritmo KNN? E para o LDA?

Com o propósito de responder as perguntas apresentadas, foram desenvolvidos diversos experimentos. Para cada forma de tratamento de imagem, incluindo a que não possui tratamento, foram realizados experimentos em ambos o KNN e LDA. No entanto, o KNN foi realizado três vezes para cada tratamento, com o propósito de encontrar o valor de K mais adequado para as classificações.

A. Classificação Utilizando o Algoritmo KNN

O KNN foi o que obteve os melhores resultados. Foi testado utilizando 3 valores de K, sendo 3, 5 e 10. Além disso também foi testado utilizando os dois tratamentos citados anteriormente. Como visto na Tabela I a precisão do KNN utilizando os dados sem tratamento foi acima de 95%, ou seja, fora gerado um modelo com menos de 5% de erro. De maneira mais detalhada a melhor pontuação de precisão foi com KNN sendo K igual a 3.

Em seguida podemos analisar as matrizes de cofusão dos melhores resultados (Figura 3 4 5). Nota-se que o algarismo de número 1 obteve 100% de acerto nas três matrizes, o que se explica pela singularidade de escrita, nenhum outro número possui somente um traço vertical individual, o que torna as matrizes das imagens mais fácil de se identificar.

Em compensação os outros dois métodos, linhas binarias e linhas normais não obtiveram a mesma eficiência Tabela I.

K	Sem Tratamento	Linhas Normais	Linhas Binárias
3	97.05%	72.34%	81.66%
5	96.88%	73%	82.02%
10	96.65%	72.51%	81.83%

Tabela I
PONTUAÇÃO DE PRECISÃO DOS MODELOS DE KNN GERADOS

Podemos supor que isso se deve a minimização das distâncias entre os vetores depois de tratados. Antes do tratamento os vetores maiores possibilitavam um cálculo mais detalhado da distância, ao reduzir os vetores para 28 elementos e reduzirmos a amplitude dos valores reduzimos também a distância entre eles, gerando uma quantidade maior de intersecções entre as classes.

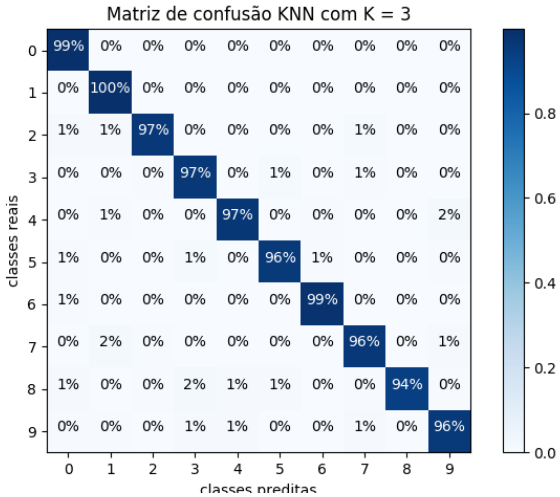

Figura 3. Matriz de cofusão relativa ao KNN com K igual a 3 e dados sem tratamento

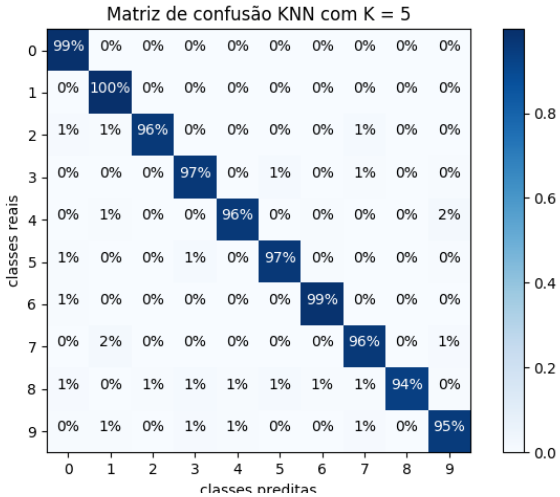

Figura 4. Matriz de cofusão relativa ao KNN com K igual a 5 e dados sem tratamento

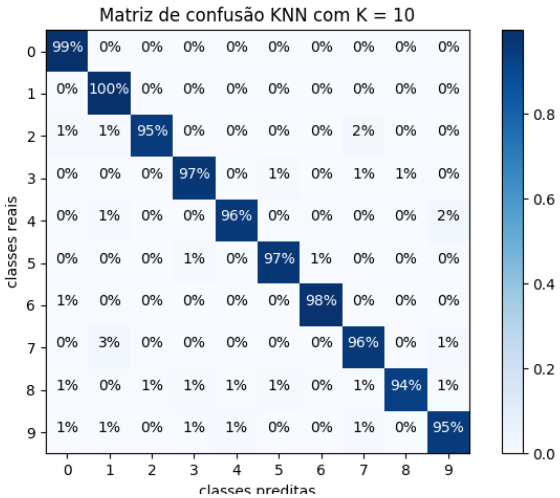

Figura 5. Matriz de cofusão relativa ao KNN com K igual a 10 e dados sem tratamento

Sem Tratamento	Linhas Normais	Linhas Binárias
87.3%	24.91%	66.69%

Tabela II
PONTUAÇÃO DE PRECISÃO DOS MODELOS DE LDA GERADOS

B. Classificação Utilizando o Algoritmo LDA

Os resultados dos experimentos do LDA estão representados na tabela II onde o modelo recebendo dados sem tratamento se saiu consideravelmente melhor que os outros dois métodos. Analisando a matriz de cofusão Figura 6 notamos que novamente o algarismo 1 obteve o melhor resultado por motivos já explicado anteriormente. Porém o algarismo 2 com 79% de acerto sendo o pior resultado se deve a semelhança da parte superior do algarismo, onde se assemelha muito com a parte superior do número 3 e o número 8.

Em relação aos outros tratamentos a precisão caiu de maneira considerável para linhas binárias e se tornaram insatisfatórias para as linhas normais. Como LDA é um algorítimo paramétrico de acordo com a entrada dos dados os pesos das variáveis se ajustam. Devido a isso os tratamentos reduzem muito o valores contidos nos vetores, o que também reduz o quanto os dados afetam nos parâmetros da equação gerada pelo LDA. Através da Tabela 7 é nítido que as predições ficaram concentradas no algarismo de número 1.

VI. CONCLUSÃO

Este trabalho implementa dois algoritmos de classificação e dois tratamentos de imagem, para analisar o impacto na aprendizagem de classificação de algarismos manuscritos. Os algoritmos implementados forma o K-Vizinhos Mais Próximos e o Algoritmo de Análise de Discriminantes Lineares. As imagens foram passadas para o algoritmo utilizando três formas diferentes, sem realizar nenhum tratamento, o Tratamento de Linhas Binárias, e o Tratamento de Linhas Normalizadas.

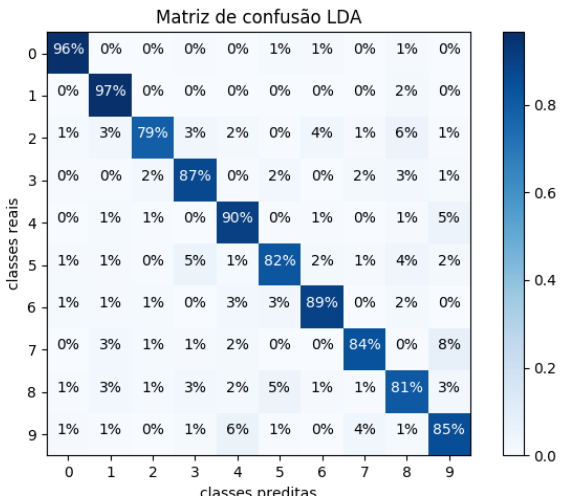

Figura 6. Matriz de cofusão relativa ao LDA com dados sem tratamento

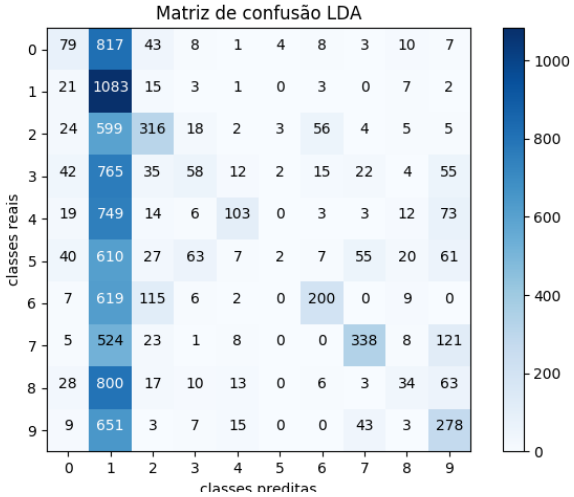

Figura 7. Matriz de cofusão relativa ao LDA com dados tratados utilizando linhas normais

As implementações foram testadas por diversos experimentos. Seus resultados foram analisados:

- O KNN com dados sem tratamento foi capaz de chegar a uma precisão de 97% , ou seja, erro abaixo de 5%
- O melhor valor para K entre os três experimentados foi 3, alcançando cerca de 0.20% de precisão acima dos demais.
- Tanto para o KNN e LDA a melhor forma de entrada dos dados, dentre os testados, são os dados brutos, ou seja, sem tratamento.

Nota-se que dentre todos os resultados, o que melhor classificou os dígitos foi proveniente da imagem sem nenhum tratamento, atingindo uma acurácia de 97,05%. É interessantes para trabalhos futuros buscar novas métricas de tratamento das imagens que possuam uma taxa de acerto tão precisa quanto da imagem sem tratamento, e considerando também, que a métrica diminua o tempo de processamento significativamente. Após uma boa métrica ser descoberta, pode-se também realizar um sistema de classificação em tempo real com dados lidos de uma câmera, para aproximar à realidade o sistema de leitura de CPF das cartas.

Referências

[1] MLA Kotsiantis, Sotiris B., I. Zaharakis, and P. Pintelas. “Supervised machine learning: A review of classification techniques.” Emerging artificial intelligence applications in computer engineering 160 (2007): 3-24.