UM MÉTODO PARA CLASSIFICAÇÃO DE DADOS BASEADO NOS K-VIZINHOS MAIS PRÓXIMOS PARA O RECONHECIMENTO DE CARACTERES

Gustavo Silva Semaan[1,2]
[1] Inst. do Noroeste Fluminense de Educação Superior - Universidade Federal Fluminense
[2] Escola de Engenharia Industrial e Metalúrgica de Volta Redonda – UFF
[1] Av. João Jasbick, s/nº - Aeroporto - Santo Antônio de Pádua -RJ
gustavosemaan@id.uff.br

Ênio de Oliveira Nascimento, Lubert Henrique, Débora Alvernaz Corrêa
Faculdade Metodista Granbery
R. Sampaio, 300 - Granbery, Juiz de Fora - MG
enio.eon@gmail.com, lubert-h@hotmail.com, deboradac@gmail.com

José André de Moura Brito
Escola Nacional de Ciências (ENCE) - Instituto Brasileiro de Geografia e Estatística (IBGE)
Rua André Cavalcanti, 106, Bairro de Fatima – Centro – Rio de Janeiro- RJ
jambrito@gmail.com

RESUMO

Os avanços constantes na Tecnologia da Informação proporcionaram um aumento significativo no volume de dados. Nas corporações é impraticável relacionar uma melhora de qualidade e/ou aumento na produção de bens e serviços sem investimentos em Tecnologia da Informação. Fonte de intensa pesquisa, o reconhecimento de padrões ainda apresenta vários problemas, seja no reconhecimento de caracteres manuscritos, caracteres impressos ou imagens. As dificuldades mais evidentes para análises estão relacionadas ao grande volume de dados associados às abordagens tradicionais de extração de informação ou à natureza complexa de dados em conjuntos de bases menores. Diante desse cenário exemplificador podemos observar o clássico algoritmo de classificação k-NN. Uma nova abordagem para o algoritmo k-NN é proposta com o objetivo de tornar a classificação de dados para reconhecimento de caracteres mais eficaz. Esta abordagem pode trazer benefícios tanto para sociedade no uso em aplicações tecnológicas, como para a comunidade científica.

Palavra-chave: Mineração de Dados, Aprendizagem de Máquina, Reconhecimento de Padrões, Classificação.

ABSTRACT

The constant advances in Information Technology have provided a significant increase in the data volume. In corporations, it is impossible to relate an improve in quality and/or increase in production of assets and services without investments in computer systems. Intense research sources, pattern recognition still present several issues, whether in recognition of handwritten characters, printed characters or images. The clearest difficulties for analysis are related to the large data volume associated with traditional approaches of

information extracting or the complex nature of smaller sets of base data. In front of this exemplified scenario, one can observe the classic kNN algorithm of classification. The present work aims a proposal for a new approach to the kNN algorithm. These goals make the classification data more robustness for character recognition. This approach can bring benefits, both to society and the scientific community, in the use of technological applications.

Keywords: Data Mining, Machine Learning, Pattern Recognition, Classification.

Como Citar:
SEMAAN, Gustavo Silva et al. Um método para classificação de dados baseado nos k-vizinhos mais próximos para o reconhecimento de caracteres. *In*: SIMPÓSIO DE PESQUISA OPERACIONAL E LOGÍSTICA DA MARINHA, 19., 2019, Rio de Janeiro, RJ. **Anais** […]. Rio de Janeiro: Centro de Análises de Sistemas Navais, 2019.

1. INTRODUÇÃO

Os avanços da tecnologia da informação vêm gerando, atualmente, um ganho significativo no volume de dados produzidos por sistemas computacionais. A análise dessas bases de dados tem fornecido informações relevantes para a comunidade científica, assim como para o mercado corporativo. Além disso, tem colaborado de maneira relevante na tomada de decisões administrativas, a fim de possibilitar a criação de novas estratégias de mercado.

Atender à crescente demanda por extração, interpretação e/ou relacionamentos de informações, não consiste somente em modelos de recuperação de informações implícitas, mas, também, em uma tarefa que exige rigor analítico e consistência de informação. Com isso, torna-se necessária a utilização do processo de descoberta de conhecimento em bases de dados (KDD, do inglês *Knowledge Discovery in Database*) [3].

O processo de descoberta de conhecimento em bases de dados foi definido por [2] como "*processo não trivial, formado por várias etapas, interativo e iterativo, para identificação de padrões válidos, novos, potencialmente úteis e compreensíveis, em grandes conjuntos de dados*". Trata-se de um processo que pode utilizar conceitos e técnicas de diversas disciplinas como aprendizado de máquina, banco de dados, estatística e inteligência artificial.

O KDD é considerado um processo interativo por necessitar de especialistas em sua condução, e iterativo por eventualmente ser necessário retornar ou avançar etapas. Basicamente, as etapas operacionais do processo de KDD são organizadas em três grandes grupos, a saber: pré-processamento, mineração de dados e pós-processamento.

Problemas reais de diversos domínios podem ser analisados e estudados para que conhecimentos sejam adquiridos através de aplicações de técnicas de mineração de dados. Nesse sentido, pode-se citar: Processamento de Imagens, Bioinformática, Mineração de Texto e Análise de Dados para previsão de Abalos Sísmicos [8]. Dentre as tarefas de mineração de dados existentes na literatura, a Classificação consiste em associar objetos a uma classe, tendo por base uma coleção de classes predefinidas [3].

O presente trabalho propõe a pesquisa e o estudo relacionados à Classificação e considera o problema de Identificação de Caracteres Manuscritos como um estudo de caso. O artigo está estruturado da seguinte maneira: a seção 2 descreve uma revisão da literatura em relação ao problema de reconhecimento de caracteres manuscritos; a seção 3 apresenta o procedimento de classificação de dados; a seção 4 introduz o assunto de pré-processamento e a sua importância; a seção 5, após a etapa de pré-processamento, relata o método de

classificação proposto que considera o clássico algoritmo k-NN e conceitos de processamento de imagens no pré-processamento; a seção 6 apresenta os experimentos computacionais e comparativos com algoritmos da literatura; a seção 7 trata das considerações finais e trabalhos futuros.

2. RECONHECIMENTO DE CARACTERES MANUSCRITOS E SUAS ABORDAGENS

Fonte de intensa pesquisa, a aplicação de técnicas que possibilitem o reconhecimento de caracteres ainda apresenta limitações, seja ou no reconhecimento de caracteres manuscritos ou até mesmo impressos. O desafio hoje é produzir um algoritmo que possa reduzir, consideravelmente, o número de erros no reconhecimento de caracteres e/ou aumentar a eficácia em relação às abordagens existentes, embora já existam aplicações reais de relativo sucesso, como a identificação de placas de automóveis. Embora o tema não seja novidade, com propostas alternativas de solução já há alguns anos como em [1][4][5][11] [12], o método utilizado no presente trabalho pode inspirar adaptações para aplicações em diversas áreas.

Em [5] é proposta uma ferramenta que se baseia na modelagem do comportamento da sequência de pixels, onde são observadas as transições entre os níveis de cinza dos pixels adjacentes, gerando uma matriz binária com m x n posições. As técnicas e algoritmos utilizados para o reconhecimento de caracteres servem de base para outros trabalhos relacionados ao reconhecimento de texto em imagens. Por exemplo, [7] utiliza, respectivamente, metodologias para digitalização; restauração; segmentação; extração de atributos; classificação e decisão, com o objetivo de identificar textos em imagens, separá-los e realizar seu reconhecimento.

Segundo [4], uma das etapas do processo de reconhecimento de caracteres é a classificação. A técnica utilizada classifica os dados de forma supervisionada a fim de obter regras que permitam predizer, com precisão, a classe de cada amostra. Portanto, utilizam-se medições realizadas em caracteres, para identificação de classes desconhecidas. Já em [1] foi apresentada uma técnica que utiliza extração de regras de associação para o reconhecimento de caracteres numéricos manuscritos.

3. A CLASSIFICAÇÃO DE DADOS E SEUS ALGORITMOS

Conforme [2], a mineração de dados é uma das etapas do KDD que consiste em aplicar algoritmos para análise e descoberta de padrões ou modelos de dados. Dentro da etapa de mineração de dados existem diversas tarefas, como: a extração de regras de associação, o agrupamento de dados (do inglês *clustering*) e a classificação de dados.

A classificação de dados consiste em um processo que mapeia um objeto (exemplar ou tupla) em uma ou classe (categoria) predefinida com base em um conjunto de objetos previamente classificados [8].

Em síntese, um modelo de classificação pode ser definido como a tarefa de aprendizado de uma função que mapeie um conjunto de objetos em classes (rótulos) pré-determinadas. Este modelo é construído através de uma técnica de classificação, ou seja, uma abordagem sistemática a partir de um conjunto de dados de entrada [6].

Para a execução dessas tarefas existem diversas técnicas ou algoritmos que podem ser considerados, destacando-se, dentre eles: redes neurais, classificadores baseados em regras e algoritmos que atuam em árvores de decisão. Já a avaliação de um modelo de classificação é baseada nas contagens de registros de testes previstos, as quais apresentam, pelo modelo, respostas corretas e incorretas. Estas contagens são tabuladas em uma tabela conhecida como matriz de confusão [8].

A Tabela 1 apresenta a matriz confusão de uma classificação binária em que os resultados obtidos satisfatoriamente pelo algoritmo (acertos) são apresentados na diagonal principal (em destaque). Além disso, embora a matriz confusão forneça informações importantes, apresentar tais resultados como um único índice pode ser mais conveniente. Nesse sentido, podem ser consideradas como métricas de desempenho tanto a Acurácia (Precisão) (Equação 1) quanto a Taxa de erro (Equação 2).**Tabela 1**: Matriz confusão para um problema de 2 classes (A e B)

		Classe Prevista	
		A	B
Classe Real	A	***AA***	AB
	B	AB	***BB***

$$Precisão = \frac{\sum nº\ de\ previsões\ de\ sucesso}{nº\ de\ previões} \quad (1)$$

$$Taxa\ de\ erro = \frac{\sum nº\ de\ previsões\ de\ erradas}{nº\ de\ previões} \quad (2)$$

Neste contexto, o presente trabalho propõe um método de classificação que considera o clássico algoritmo k-NN e conceitos de processamento de imagens no pré-processamento. A seção 4 apresenta o pré-processamento realizado e o algoritmo de classificação.

4. PRÉ-PROCESSAMENTO

O pré-processamento, que consiste em uma etapa de extrema relevância no processo do KDD, é aplicado com o objetivo de preparar os dados de maneira apropriada para a mineração. O tratamento dos dados é realizado com o uso de diversas técnicas e, dentre elas, podem-se destacar: amostragem; redução de dimensionalidade; seleção de subconjuntos de recursos; e transformação de atributos [8].

Em especial, no presente estudo, essa etapa é de grande importância devido à grande diversidade de formas em que os caracteres manuscritos podem ser encontrados. Uma das abordagens do reconhecimento de um caractere manuscrito é através da digitalização da imagem em análise. Nesse sentido, o trabalho considera também conteúdos de outras áreas como processamento de imagens e morfologia matemática. Dentre as técnicas mais utilizadas na literatura, o presente trabalho considera: (i) *Limpeza de ruídos*: eliminar interferências na imagem, como eventuais “manchas” ou “sujeiras”; (ii) Centralização do caractere: mover o caractere para o centro da imagem; (iii) *ajuste da imagem*: ampliar ou reduzir o caractere para ocupar uma área considerada ideal; (iv) *Matriz binária*: além de converter a imagem para monocromática (preto e/ ou branco), deve-se gerar uma matriz de binários. (v) *cortes*: deve-se identificar se a célula será considerada 1 (preto) ou 0 (branca) com base na quantidade de pixels em preto, conforme um percentual de corte submetido.

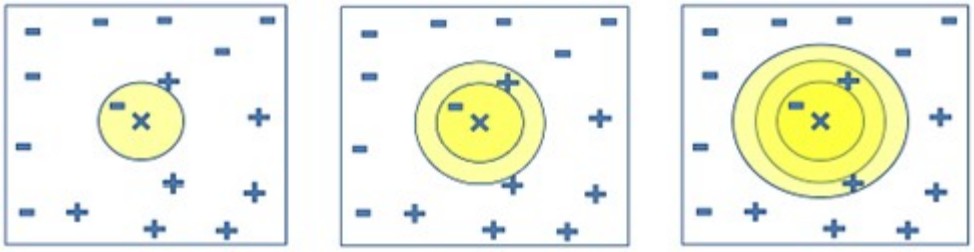
5. MÉTODO PROPOSTO

O algoritmo k-NN atua na classificação de objetos desconhecidos, isto é, objetos ainda sem classe, baseando-se na comparação com objetos similares que foram previamente classificados. Para cada objeto ainda não classificado (desconhecido), o algoritmo verifica as classes dos k objetos mais próximos (mais semelhantes ou menos diferentes) e aloca esse objeto a uma das classes. Para a análise de similaridade entre dois objetos podem ser

utilizadas várias métricas presentes na literatura, como a Distância de Manhattan ou a Distância Euclidiana.

A Figura 1 ilustra a execução do algoritmo k-NN, considerando que cada objeto desconhecido está ilustrado como um '×' em uma mesma instância (amostra, base de dados ou conjunto) considerando os valores de k={1, 2, 3}.

Seguem as análises para os três casos supracitados: (i) para k = 1 (o vizinho mais próximo) o objeto desconhecido está mais próximo do objeto que pertence à classe "-", então deve ser alocado a essa classe (Figura 1(a)); (ii) para k = 2 os k objetos (dois objetos) menos distantes do objeto desconhecido pertencem à classe "-" e à classe "+". Nesse caso ocorreu um empate e um critério que pode ser considerado é a eliminação do objeto mais distante entre os k objetos mais próximos. Nesse caso, o objeto a ser classificado deve ser alocado na classe "-" (Figura 1(b)); (iii) para k = 3 os k objetos (três objetos) menos distantes do objeto desconhecido pertencem às classes '-' e '+' (uma e duas ocorrências, respectivamente). Assim o algoritmo deve alocar o objeto a ser classificado na classe '+' (Figura 1(c)).



(a) Objeto mais próximo ao objeto (x) pertence à classe (-).

(b) Segundo objeto mais próximo pertence à classe (+).

(c) Terceiro objeto mais próximo pertence a classe (+).

Figura 1: vizinhos mais próximos adaptado de [8]

O pseudocódigo do Algoritmo 1 abaixo traz, por etapas, a execução do k-NN. O algoritmo calcula a distância para cada objeto z ainda não classificado e todos os objetos da classe de treinamento. Em seguida, as distâncias de z aos seus k-vizinhos mais próximos são ordenadas, e z recebe o rotulo (é alocado à classe) de maior frequência dentre os k objetos mais próximos.

Algoritmo 1: Algoritmo k-NN

```
Seja D conjunto de objetos, z o objeto a classificar e k vizinhos prox.
Para cada objeto d_i de D faça
  Calcule distancia(d_i,z)
Fim para
Ordenar distâncias obtidas
Conjunto Dz recebe os k vizinhos mais próximos de z
Exemplo z recebe o rótulo de maior frequência em Dz
```

Fonte: adaptado de (TAN: STEINBACH: KUMAR, 2009)

Acrescentando etapas de *binarização* da imagem através de diferentes limiares de uma matriz de inteiros intermediária, denominada corte de pixels, e diferentes grupos de atributos ao algoritmo k-NN tradicional, o método proposto tem como objetivo tornar mais robusta a identificação de caracteres manuscritos com base no clássico algoritmo k-NN da literatura.

Antes de iniciar a identificação da imagem pelo método proposto, algumas etapas de pré-processamento devem ser realizadas. Como uma primeira etapa é necessário converter a imagem do formato RGB (Vermelho, Verde e Azul, do inglês "*Red, Green and Blue*") para o formato monocromático. Na segunda etapa um filtro para eliminação de ruídos foi aplicado. A terceira etapa foi atuar na centralização de cada caractere no espaço. Por fim,

a quarta etapa consiste no ajuste da imagem para a dimensão 300x300 pixels.

Após ter a etapa de pré-processamento efetuada, o método proposto é executado e deve ocorrer a identificação da imagem do caractere submetido.

A primeira etapa do método proposto consiste na extração de atributos com base na matriz binária B com 32 linhas e 32 colunas (32x32). Para isso a imagem submetida serve como origem para preencher uma matriz intermediária de inteiros I com 32 linhas e 32 colunas (32x32). Cada célula dessa matriz possui a quantidade x de pixels pretos existente em sua área correspondente na imagem original (Figura 2).

Com o objetivo de gerar diferentes matrizes binárias, valores de corte são submetidos. Cada valor de corte indica o percentual mínimo (valor) que cada célula na matriz I deve possuir para que a célula correspondente na matriz B seja 1. Nesse trabalho foram utilizados os valores 25%, 50%, 75% e 90%, conforme ilustra a Figura 3. Destaca-se que esses valores foram identificados em experimentos preliminares.

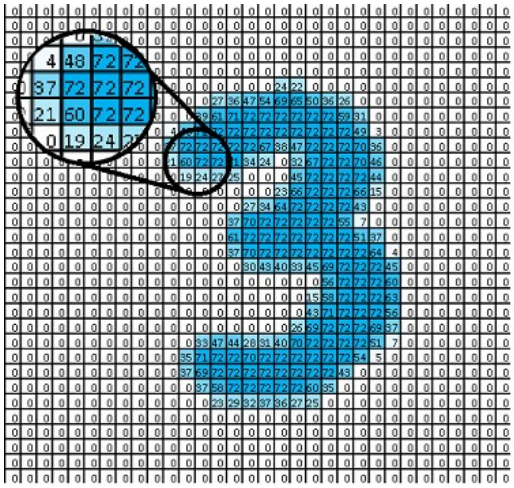


Figura 2 - representação de uma imagem binarizada.

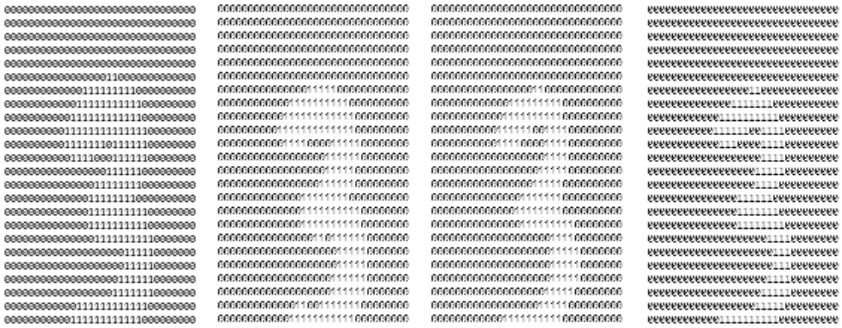


Figura 3 - Representação da matriz B obtida pela imagem do caractere 3 e cortes de pixel 25%, 50% 75% e 90% respectivamente.

Com base em cada matriz B obtida, conforme o valor de corte, os atributos devem ser extraídos. A Figura 4 apresenta os 112 atributos utilizados no presente trabalho. Para facilitar o entendimento da Figura 4, seguem exemplos de como realizar a extração de atributos da matriz binária: (i) Atributos de 1 a 32: quantidades de bits “1” da matriz binária em suas respectivas colunas; (ii) Atributos 33 a 64: quantidades de bits “1” da matriz binária em suas respectivas linhas. Nesse caso o atributo 33 corresponde à linha 1, e o atributo a corresponde à linha (a + 32); (iii) Atributo 65: soma dos atributos 1 e 2; (iv) Atributo 82: soma dos atributos 35 e 36; (v) Atributo 102: soma dos atributos 75 e 76. Destaca-se que o atributo 75, por exemplo, consiste na soma dos atributos 21 e 22.

Os atributos apresentados foram organizados em quatro grupos: (i) Grupo 1: todos os 112 atributos; (ii) Grupo 2: 64 atributos, que referem-se às linhas e colunas da matriz 32x32; (iii) Grupo 3: atributos 65 até os 96 (32 atributos); (iv) Grupo 4: atributos 97 a 112 (16 atributos). Para os grupos 3 e 4 de atributos, a matriz principal (32 x 32) é reduzida para matrizes secundárias contendo 16x16 e 8x8 células respectivamente, aumentando a quantidade de células que são somadas quando comparadas à matriz principal.

Com o objetivo de tornar o método mais robusto, uma base de dados com os resultados coletados é formada. Nesse sentido são armazenados os k objetos mais próximos aplicados a cada matriz B para cada grupo de instâncias. O método retorna, dentre todos os resultados, à classe com maior ocorrência dentre os k objetos mais semelhantes.

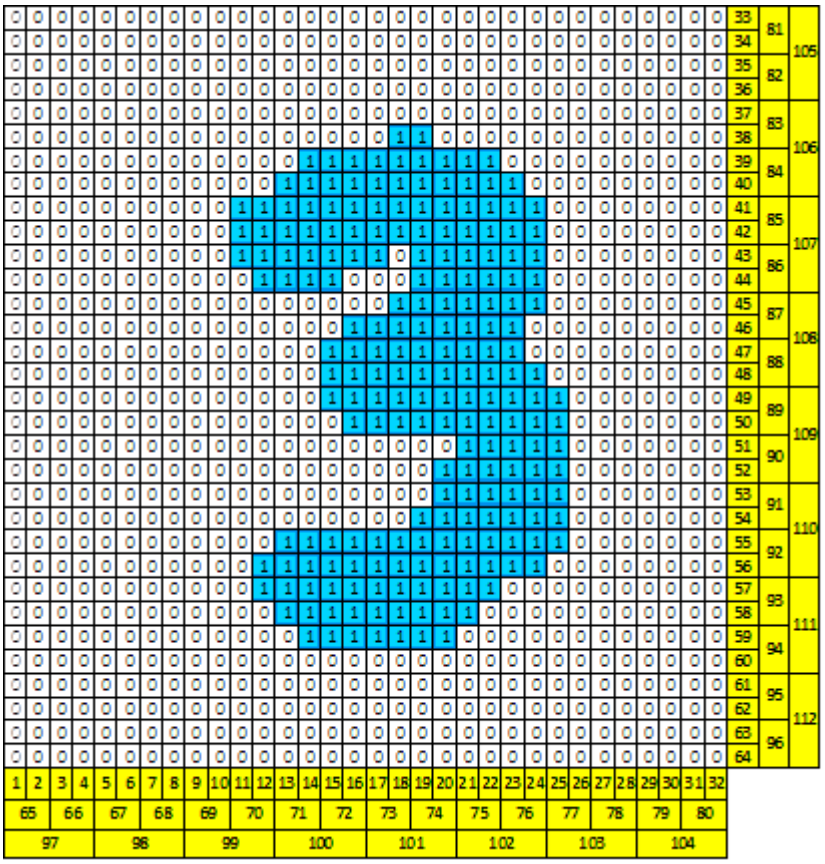


Figura 4 – detalhamento do método de extração de atributos

6. RESULTADOS COMPUTACIONAIS

O algoritmo proposto no presente trabalho foi implementado em linguagem JAVA. Para a execução dos experimentos computacionais foi utilizado um computador dotado de um processador Intel Celeron 1,5GHz, com 2 GB de memória RAM e Sistema Operacional Windows 10 Pro. A avaliação dos resultados foi realizada em duas etapas: (i) Validação das bases utilizando o *k-fold cross-validation*, que atua na verificação da precisão das bases através de sua acurácia; (ii) Experimentos realizados com o método proposto e com algoritmos da ferramenta Weka.

6.1. MÉTODO DE VALIDAÇÃO CRUZADA

Uma das formas de validar a generalização de um modelo para uma base de dados é a validação cruzada. Esta abordagem consiste em selecionar, a partir da base, várias amostras aleatórias simples de igual tamanho, as quais serão destinadas a conjuntos de treino e de

teste. O comparativo é feito através da aplicação da expressão que permite calcular o percentual de erro fornecido, segundo [8].

O método *k-fold cross-validation* consiste em estabelecer segmentos (amostras) de mesmo tamanho removidos da base. Além disso, para cada iteração, essas amostras são utilizadas como base de treinamento ou de teste. A Figura 5 apresenta o ciclo de iterações do *k-fold cross-validation*. O *Leave-One-Out* é um caso especial onde o *k-fold* possui tamanho 1(um único registro). Esse modelo, apesar de apresentar um grande custo computacional, permite a utilização do uso de toda base necessária para treinamento, e é um modelo com iterações mutuamente exclusivas, cobrindo o conjunto inteiro de dados [8].

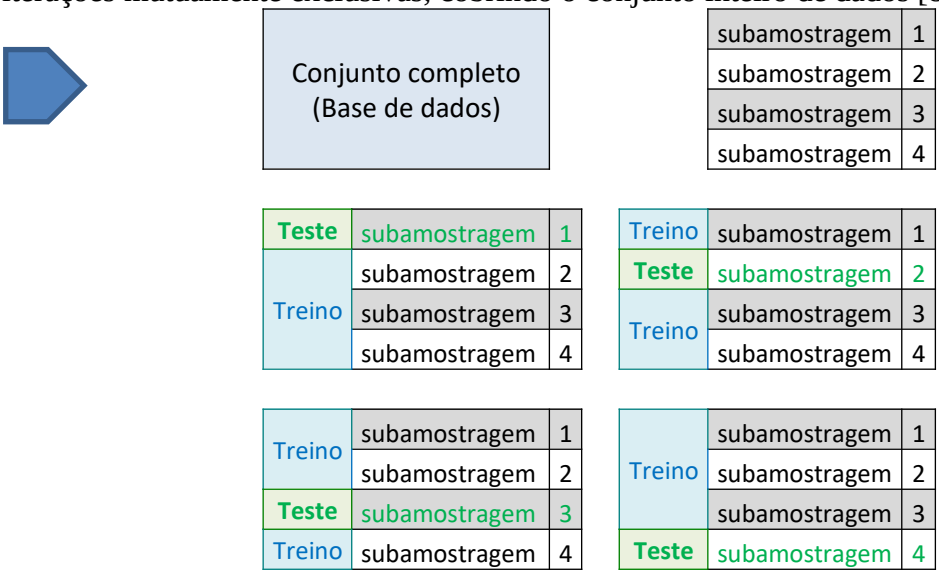
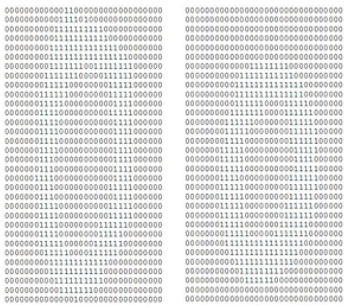


Figura 5 – Exemplo do modelo K-Fold

6.2. MÉTODO DE VALIDAÇÃO CRUZADA

Dentre essas bases consideradas, quatro foram obtidas do UCI [9] e as demais foram propostas pelos autores. As bases construídas pelos autores foram geradas a partir do processo de escrita de números (em ordem de 0 a 9) em folhas de papel A4, divididas em matrizes 10X8. As folhas foram digitalizadas e os caracteres foram recortados manualmente através de ferramentas de edição de imagem. Em seguida, esses caracteres foram submetidos ao pré-processamento relatado na seção 3. A Figura 6 apresenta o caractere 0 (zero) capturado da base do UCI e o caractere 0 (zero) produzido pelos autores. A Figura 7 apresenta imagens dos números das bases propostas pelos autores.



(UCI) (autores)

Figura 6 – caractere 0 obtido pelo UCI e produzido pelo artigo nesta ordem

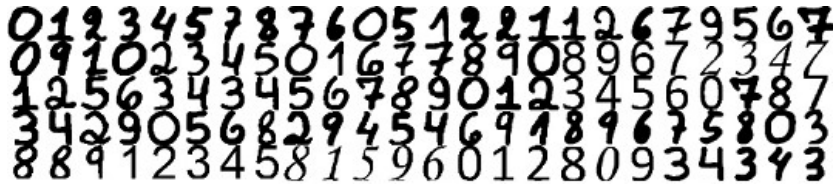


Figura 7 – base de 120 caracteres manuscritos propostos.

6.3. FERRAMENTA PARA MODELO COMPARATIVO

Dentro da complexidade inerente ao processo do KDD são adotadas ferramentas no intuito de executar tarefas operacionais e de controle. As ferramentas operacionais auxiliam no equacionamento de adversidades como, por exemplo, a necessidade de lidar com volume de dados e heterogeneidade, dificuldades de integração e comparação de algoritmos específicos. A ferramenta Weka, que possui implementações de diversos dos algoritmos mais conhecidos para Mineração de Dados da literatura, foi considerada nos experimentos computacionais.

A Weka corresponde a uma ferramenta com código aberto, customizável e expansível. Foi desenvolvida em JAVA e possui entradas de arquivos diretos em ARFF, CSV e C45. A extensão ARFF trata-se de um arquivo no formato ASCII para definir atributos e valores. O software possui interfaces de entrada implementadas que permitem atender diversas aplicações [10].

6.4. VALIDAÇÃO DAS BASES DE TREINAMENTO E TESTE

Para a avaliação das bases de dados, o presente trabalho utilizou a abordagem *k-fold cross-validation*. Diante das bases binárias do repositório UCI, testes foram realizados utilizando o método proposto e os algoritmos de classificação k-NN e Redes Neurais, disponíveis na ferramenta Weka.

Os resultados do algoritmo k-NN da ferramenta Weka foram processados através do classificador denominado IBK, e foi usada a distância euclidiana. Para o algoritmo de Redes Neurais foram utilizadas as configurações *MultilayerPerceptron* e a distância euclidiana.

O experimento para a validação das bases de dados fornece uma estimativa sobre a precisão do modelo utilizado. A Tabela 2 apresenta os resultados obtidos com o uso do método proposto do algoritmo k-NN e de Redes Neurais. No caso do método proposto e do k-NN da ferramenta Weka, a quantidade k de objetos mais próximos foi 3 (três).

Para o método proposto foi obtida a precisão mínima de 83% para 16 atributos. A média das acurácias superior a 90% foi considerada aceitável, mesmo possuindo bases cuja precisão está inferior aos algoritmos da literatura. Na segunda etapa de avaliação dos resultados foram utilizadas 10 bases distintas, das quais quatro foram obtidas do repositório do UCI (*Irvine Machine Learning Repository*) [9]. As demais bases foram propostas pelos autores e correspondem a números escritos por diferentes pessoas.

Tabela 2 – resultado de validação das bases de dados

Base de dados (instâncias)	Acurácia (%)					
	Método proposto k = 3				k-NN Weka k = 3	Redes Neurais Weka
	112 atributos	64 atributos	32 atributos	16 atributos	112 atributos	112 atributos
optdigits-orig.tra	97,8	98,0	98,0	97,8	97,3	99,6
optdigits-orig.cv	96,8	97,1	97,1	87,4	96,6	99.0
optdigits-orig.wdep	97,5	97,9	98,0	96,8	96.5	99,5
optdigits-orig.windep	98,9	98,9	98,9	98,8	98,4	99,6
Artificial-numeros	100,0	100,0	100,0	100,0	80,0	100,0

Real-numeros1	93,3	93,3	96,7	83,3	93,3	100,0
Real-numeros2	90,0	90,0	90,0	90,0	90,0	100,0
Real-numeros3	100,0	100,0	100,0	96,7	100,0	100,0
Real-numeros1,2e3	97,8	97,8	97,8	97,8	95,5	100,0
Artificial-Real1,2e3	97,5	97,5	97,5	97,5	93,3	98,3

6.5. RESULTADOS DO MÉTODO PROPOSTO

O próximo experimento consistiu em executar o método proposto e os algoritmos do Weka no grupo de atributos 1 para k=3. O método proposto se destacou, obtendo os melhores resultados em oito das dez instâncias. Em relação ao k-NN, o método proposto produziu os melhores resultados em nove das dez bases, havendo um empate em uma das bases (Tabela 3). A Figura 8 ilustra a matriz confusão obtida com os resultados do Grupo 1 do método proposto na base *optdigits-orig.tra* com o número dos k objetos mais próximos igual a três. A diagonal principal (destacada em cinza) apresenta os acertos na classificação em que o número total de instâncias de cada classe é 10.

Tabela 3 – Grupo 1 de resultados

	Acurácia (%)			
Base de dados	Método proposto k = 3	k-NN Weka k = 3	Redes Neurais Weka	Melhor Acurácia
optdigits-orig.tra	77,8	64,4	71,1	***Método proposto***
optdigits-orig.cv	81,1	67,8	64,4	***Método proposto***
optdigits-orig.wdep	78,9	70,0	73,3	***Método Proposto***
optdigits-orig.windep	73,3	71,1	67,5	***Método Proposto***
Artificial-números	64,4	41,0	27,8	***Método Proposto***
Real-numeros1	64,4	63,3	67,8	*Redes Neurais*
Real-numeros2	72,2	66,7	76,7	*Redes Neurais*
Real-numeros3	68,8	68,9	68,9	***Empate***
Real-numeros1,2e3	100,0	95,5	100,0	***Empate***
Artificial-Real1,2,3	100,0	95,5	100,0	***Empate***

O algoritmo de Redes Neurais, por ser um método estocástico (base em critérios de parada), possui maior custo uma vez que necessita de maior tempo de processamento. Já o método proposto, por utilizar em sua estrutura pelo menos quatro execuções de sua implementação do k-NN (uma para cada valor de corte), também demandou mais tempo que o k-NN disponível no Weka. Em caso de bases de dados com número de instâncias grandes como a *optdigits-orig.tra*, que possui 1834 objetos, o algoritmo de Redes Neurais do Weka demandou um alto tempo de processamento. No caso das bases que possuem poucos objetos, como a Real-numeros1, o tempo médio de processamento do método proposto foi superior ao algoritmo de Redes Neurais do Weka. A Tabela 4 apresenta o tempo médio de processamento por instância.

Tabela 4: Tempo médio de processamento dos testes

	Tempo médio de processamento (em segundos)		
Base de dados	Método proposto	k-NN Weka	Redes Neurais Weka
optdigits-orig.tra	39,1	1,2	602,8
optdigits-orig.cv	38,4	0,9	298,9
optdigits-orig.wdep	38,2	0,8	267,1
optdigits-orig.windep	39,0	1,1	270,1
Artificial-números	33,7	0,5	8,1

Real-numeros1	33,6	0,4	8,2
Real-numeros2	32,9	0,4	9,3
Real-numeros3	32,4	0,4	9,3
Real-numeros1,2e3	32,5	0,4	28,0
Artificial-Real1,2,3	32,6	0,4	33,3

		Classe Encontrada									
	Classes	0	1	2	3	4	5	6	7	8	9
Classe Verdadeira	0	9	0	0	0	0	0	0	0	0	0
	1	0	6	3	0	0	0	0	0	0	0
	2	1	0	5	0	0	0	0	0	3	0
	3	0	0	0	9	0	0	0	0	0	0
	4	1	0	0	0	4	0	0	0	0	4
	5	0	0	0	1	0	6	0	0	0	2
	6	0	1	0	0	0	0	8	0	0	0
	7	0	0	0	0	0	0	0	8	0	1
	8	1	0	0	1	0	0	0	0	7	0
	9	0	0	0	0	0	0	0	1	0	8

Figura 8: matriz confusão referente ao grupo 1 de resultados, base optdigits-orig.tra para k = 3

7. CONSIDERAÇÕES FINAIS

O presente trabalho trouxe a proposta de um método para classificação de caracteres, considerando conceitos de vizinhança mais próxima. Foram realizadas atividades do processo de KDD, em que: (i) no pré-processamento foi realizada limpeza, transformação e organização de imagens e matrizes com dados de entrada; (ii) na Mineração de dados ocorreu execução do método proposto e de algoritmos do Weka para obtenção dos resultados; (iii) por fim, no pós-processamento, houve a apresentação dos resultados comparativos com algoritmos da literatura bem como apresentação de uma matriz confusão.

A partir dos resultados, constatou-se que o presente trabalho apresentou um ganho considerável em acurácia, especialmente em caracteres que se diferenciam do padrão comum. Naturalmente, o método proposto consome maior tempo de processamento do que o k-NN devido utilização deste método várias vezes, sendo mais rápido que o algoritmo de Redes Neurais para testes com instâncias de mesmo tamanho. Assim, com base nos resultados obtidos, o método proposto pode se constituir como uma alternativa para a resolução do problema abordado.

Para a realização de novas pesquisas seguem como sugestões para trabalhos futuros: (i) Considerar, de maneira aleatória, subconjuntos de atributos. Dessa maneira o método pode se tornar heurístico, buscando a maximização da acurácia por meio da seleção de subconjuntos de atributos; (ii) Pesquisar sobre maneiras de calibrar automaticamente o valor de corte para a geração de matrizes binárias; (iii) Usar outras instâncias maiores tanto em quantidade de caracteres quanto em classes.

8. AGRADECIMENTOS

Os autores agradecem o apoio da PROPPI (Pró-Reitoria de Pesquisa, Pós-Graduação e Inovação) da UFF e da FAPERJ (G. E-26/010.101237/2018), que financiaram esta pesquisa.

9. REFERÊNCIAS BIBLIOGRÁFICAS

[1] CARVALHO, J V; SAMPAIO, M C; MONGIOVI, G. Utilização de Técnicas de “Data Mining” para o Reconhecimento de Caracteres Manuscritos. 1999. XIV Simpósio Brasileiro de Banco de Dados, 1999, Florianópolis. Anais…Florianópolis, 1999.

[2] FAYYAD, U. M.; PIATETSKY–SHAPIRO, G.; SMYTH, P. Knowledge Discovery and Data Mining: Towards a Unifying Framework, In Proceeding of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, august, 1996.

[3] GOLDSCHMIDT, R.; PASSOS, E. Data Mining: Um Guia Prático. Rio de Janeiro: Elsevier. 2005. 271p.

[4] MIRANDA, R. A. R. et al. Algoritmo para o Reconhecimento de Caracteres Manuscritos, 2013.

[5] SILVA, F. A. et al. Reconhecimento de Caracteres Baseado em Regras de Transição entre Pixels Vizinhos, 2011.

[6] SILVA, L. A. Mineração de Dados: Uma abordagem introdutória e ilustrada. São Paulo, SP: Editora Mackenzie, 2015. 154p.

[7] SILVA, L. S. Distinção Automática de Texto Impresso e Manuscrito em uma Imagem de Documento. 2009. 100f. Dissertação (Mestrado em Computação) - Universidade Federal Fluminense, Niterói. 2009.

[8] TAN, P.; STEINBACH, M.; KUMAR, V. Introdução ao data mining, Mineração de Dados. Rio de Janeiro, Ciência Moderna, 2009.

[9] Site *ics.uci.edu*, Optical Recognition of Handwritten Digits DataSet .University of California, 01/07/1998. Disponível em <https://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits> Acessado em 02/05/2019.

[10] Site *cs.waikato.ac.nz*, The online appendix on the WEKA workbench for the fourth edition of "Data Mining: Practical Machine Learning Tools and Techniques" by I.H. Witten, Eibe Frank, Mark A. Hall, and Chris J. Pal., Disponível em <http://www.cs.waikato.ac.nz/ml/weka/Witten_et_al_2016_appendix.pdf>Acessado em 02/05/2019.

[11] Z. Sun, L. Jin, Z. Xie, Z. Feng and S. Zhang, "Convolutional Multi-directional Recurrent Network for Offline Handwritten Text Recognition," *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Shenzhen, 2016.

[12] C. T. Nguyen and M. Nakagawa, "Finite State Machine Based Decoding of Handwritten Text Using Recurrent Neural Networks," *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Shenzhen, 2016.