

# Otimizando o classificador k-NN a partir de Algoritmos Genéticos

Jorge L. F. Rossi<sup>1</sup>

<sup>1</sup>Departamento de Ciência da Computação  
Universidade Tecnológica Federal do Paraná (UTFPR) – Campo Mourão, PR – Brasil

jorgerossi@alunos.utfpr.edu.br

**Resumo.** *Este artigo apresenta uma abordagem otimizada do algoritmo de classificação k-NN, utilizando-se Algoritmos Genéticos. O principal objetivo é encontrar o conjunto de características que resulta na maior taxa de acerto.*

## 1. k-NN

O k-NN (k-nearest neighbors) é um algoritmo de aprendizado de máquina supervisionado bastante utilizado. Sua implementação é uma das mais simples entre os algoritmos de aprendizado e pode ser utilizado tanto para classificação quanto regressão. O algoritmo assume que indivíduos similares estão próximos entre si, portanto calcula a distância do indivíduo a ser classificado em relação à todos os indivíduos de classes conhecidas. Após isso, é realizada uma votação entre os  $k$  elementos mais próximos para decidir a qual classe o indivíduo provavelmente pertence.

## 2. Algoritmos Genéticos

Algoritmos Genéticos são Algoritmos Evolucionários inspirados na Teoria da Evolução das Espécies, de Charles Darwin. São utilizados para encontrar soluções melhores para problemas de otimização e busca, por meio de operadores inspirados na biologia, como mutação, cruzamento e seleção.

### 2.1. Inicialização

A população (conjunto de indivíduos utilizados no Algoritmo Genético) é inicializada de forma aleatória, contendo uma grande variedade de soluções possíveis.

### 2.2. Seleção

Os melhores indivíduos são selecionados com base em uma função de *fitness*, a qual indica o quão bem tal indivíduo pode resolver o problema alvo. Pode ser utilizado um sistema de roleta, onde todos os indivíduos da população tem chances de serem selecionados, porém os com maiores valores de *fitness* tem mais chances, ou um sistema elitista, onde sempre os melhores indivíduos são selecionados para prosseguirem para a próxima geração.

### **2.3. Cruzamento**

Nessa etapa, os indivíduos selecionados anteriormente são combinados para gerarem uma nova população. Existem diversas abordagens para gerar novos indivíduos, como selecionar dois “pais” aleatórios entre a população e combinar o “material genético” de ambos. Pode ser utilizado metade da sequência genética de um pai e metade do outro, ou escolher um ponto de separação aleatório.

### **2.4. Mutação**

Para introduzir variedade nos indivíduos, pode ser aplicado o operador de mutação, onde cada bit do material genético de cada indivíduo tem uma pequena probabilidade de ser modificado. Após realizada a mutação, o algoritmo reinicia, até que uma solução suficiente seja encontrada, ou após um número definido de iterações.

## **3. Abordagem otimizada: Algoritmos Genéticos + k-NN**

A abordagem utilizada neste trabalho faz o uso de Algoritmos Genéticos para buscar o conjunto de características ótimo para ser utilizado no k-NN. O programa foi escrito utilizando a linguagem de programação Go.

### **3.1. Descrição do problema**

O problema proposto consiste em reconhecer dígitos de 0 a 9 a partir de características extraídas de imagens. Foram fornecidos um conjunto de treino e de teste, contendo 1000 indivíduos cada, com um total de 132 características.

### **3.2. Representação do indivíduo**

Um indivíduo é composto por uma sequência de 132 0s e 1s, onde valores 0s indicam que a característica correspondente será ignorada no k-NN e 1s indicam que a característica correspondente será utilizada no k-NN.

### **3.3. Inicialização**

Para compor a população inicial, foram gerados 100 indivíduos aleatórios, contendo uma sequência de 0s e 1s.

### **3.4. Seleção**

A função de *fitness* utilizada foi o k-NN, com o parâmetro  $k$  igual a 1. A função retorna a porcentagem de acertos do algoritmo em relação ao conjunto de testes, contendo 1000 indivíduos. Dessa forma, os conjuntos de características que resultarem em uma melhor acurácia de classificação serão escolhidos para prosseguirem para a próxima população. Foi utilizado um sistema de seleção elitista, onde apenas a melhor metade da população é escolhida. Tendo em vista que não existe dependência na execução do k-NN para cada indivíduo, foram utilizadas threads para reduzir o tempo total de cada iteração.

### 3.5. Cruzamento

Nessa etapa, são escolhidos dois “pais” aleatoriamente, os quais são combinados para gerar 2 filhos, contendo parte do material genético de ambos. O ponto de separação é escolhido aleatoriamente.

### 3.6. Mutação

Para manter a diversidade da população, foi utilizado o operador de mutação. Cada bit de cada indivíduo tem 0.01% de chance de ser trocado, ou seja, se for 0 se torna 1 e vice-versa.

## 4. Resultados

O algoritmo proposto, em conjunto com os parâmetros descritos, foi executado um total de 100 iterações. Abaixo estão descritos os valores obtidos após executar o k-NN isoladamente, além dos resultados obtidos a partir do uso do algoritmo proposto. Os testes foram executados em um computador com as seguintes especificações:

Processador: Intel i7 8750H - 6 núcleos - 12 threads

Memória: 8 GB DDR4

GPU: Nvidia GTX 1050 Ti

Armazenamento: Samsung EVO 970 Plus - 250 GB

#### 4.1. k-NN

<i>k</i>	1
Número de classes	132
Número de indivíduos no conjunto de treino	1000
Número de indivíduos no conjunto de teste	1000
Porcentagem de acertos	89.9%
Tempo de execução (segundos)	4.82

#### 4.2. Algoritmo Genético + k-NN

<i>k</i>	1
Número de classes máximo	132
Número de indivíduos no conjunto de treino	1000

<b>Número de indivíduos no conjunto de teste</b>	1000
<b>Número de indivíduos na população</b>	100
<b>Taxa de mutação</b>	0.01%
<b>Iterações</b>	100
<b>Porcentagem de acertos máxima</b>	96.1%
<b>Tempo de execução (segundos)</b>	3046.89

## 5. Conclusão

Analisando os resultados é possível concluir que houve uma otimização da acurácia do k-NN ao utilizar o Algoritmo Genético. Após 100 iterações, o Algoritmo Genético encontrou um conjunto de características que permite classificar os dígitos com uma porcentagem 6.2% maior do que o k-NN convencional. Existe a possibilidade de otimizar ainda mais os resultados, alterando os parâmetros de execução ou executando mais iterações.

## Referencias

Wikipedia contributors. (2019, 27 de maio). K-nearest neighbors algorithm. Em Wikipedia, The Free Encyclopedia. Acessado as 22:19, 12 de Junho de 2019, em [https://en.wikipedia.org/w/index.php?title=K-nearest\\_neighbors\\_algorithm&oldid=899016278](https://en.wikipedia.org/w/index.php?title=K-nearest_neighbors_algorithm&oldid=899016278)

Wikipedia contributors. (2019, 5 de junho). Genetic algorithm. Em Wikipedia, The Free Encyclopedia. Acessado as 22:21, 12 de Junho de 2019, em [https://en.wikipedia.org/w/index.php?title=Genetic\\_algorithm&oldid=900431220](https://en.wikipedia.org/w/index.php?title=Genetic_algorithm&oldid=900431220)