k-Vizinhos mais Próximos
scc0201 — Trabalho 02

Professor: Moacir A. Ponti
PAE: Edesio Alcobaça e Leonardo Ribeiro

1 Resumo

Neste trabalho você deverá implementar o algoritmo de Aprendizado de Máquina (AM) k Vizinhos Mais Próximos (kNN *k-Nearest Neighbour*) para classificar diferentes espécies da flor Iris.

2 Background

O Aprendizado de Máquina (AM) é um campo de pesquisa fundamentado na Inteligência Artificial, na Matemática e na Estatística e que, utilizando conceitos dessas áreas, estuda e modela as diversas faces do processo de aprendizado. AM explora e estuda a construção de algoritmos inteligentes que aprendem extraindo padrões de dados.

Um algoritmo muito famoso é o kNN, que utiliza medidas de similaridade (e.g., distância euclidiana) para classificar um novo exemplo desconhecido. Neste trabalho você irá codificar o algoritmo kNN a fim de, dadas as características da sépala e da pétala da flor Iris, descobrir a que espécie ela pertence.

2.1 O Conjunto de Dados Iris

A Iris é um gênero de plantas com flor, muito apreciada pelas cores vivas e pela diversidade de espécies. O conjunto de dados na Tabela 1 descreve as características de três diferentes espécies da flor Iris. Em AM chamamos cada linha da tabela de **exemplo**, as colunas de **atributos** preditivos(Comprimento sépala, Largura sépala, Comprimento pétala, Largura pétala) e o atributo que desejamos prever de **classe** (Espécie). A Figura 1 exemplifica visualmente as diferentes espécies de Iris e como cada atributo foi coletado.

	Comprimento sépala(cm)	**Largura sépala(cm)**	**Comprimento pétala(cm)**	**Largura pétala(cm)**	**Espécie**
1	5.3	3.7	1.5	0.2	setosa
2	7.0	3.2	4.7	1.4	versicolor
3	6.3	3.3	6.0	2.5	virginica
4	5.0	3.3	1.4	0.2	setosa
5	6.4	3.2	4.5	1.5	versicolor
6	5.8	2.7	5.1	1.9	virginica

Tabela 1: Conjunto de dados Iris.

2.2 O Algoritmo kNN

O kNN é ilustrado na Figura 2. Cada ponto no espaço 2D representa um exemplo de um conjunto de dados e as cores as classes (vide Figura 2-a). O ponto em vermelho é um novo exemplo no qual

Figura 1: Exemplo das três diferentes espécies da flor Iris. Estão demarcadas com flechas como as medidas de comprimento da sépala (a), largura da sépala (b), comprimento da pétala (c) e largura da pétala (d) foram extraídas.

deseja-se **classificar**, isto é, saber a qual classe ele pertence (vide Figura 2-b). O kNN calcula a distância euclidiana do ponto vermelho para todos os outros pontos e seleciona os k-vizinhos mais próximos. A classe mais representativa dentre os k-vizinhos selecionados é atribuída como a classe do ponto vermelho. No caso de $k = 3$ será azul, contudo para $k = 5$ será laranja.

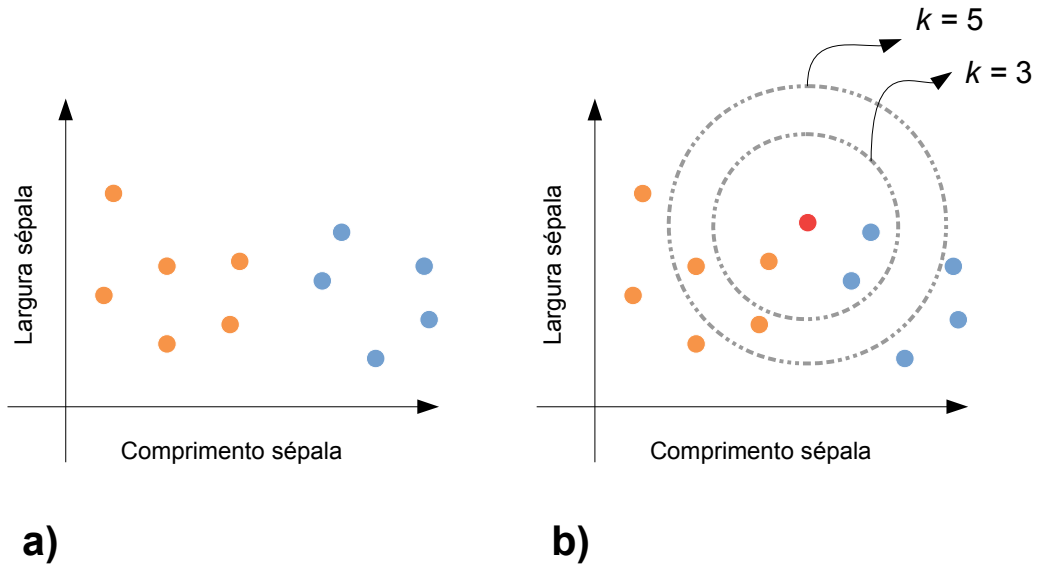

Figura 2: Ilustração do algoritmo kNN. Em *a)* temos o conjunto de dados representado por dois atributos. Em *b)* temos um novo exemplo em vermelho que será classificado pelo kNN. No caso de $k = 3$ a classe será azul e no caso de $k = 5$ laranja.

O kNN pode ser então descrito como segue:

Algorithm 1: kNN

Entrada: Conjunto de dados - D, novo exemplo - ex, número de vizinhos - k
Saída: Classe para ex
Calcular a distância de ex para cada exemplo em D;
Determinar o conjunto Q dos $k's$ exemplos em D mais próximos de ex;
Verificar a classe que é mais representativa em Q;
Retornar essa classe como a classe de ex;

A princípio, podemos usar qualquer distância. Nesse trabalho utilizaremos a distância Euclidiana entre dois vetores **a** e **b**, ambos com m elementos, dada por:

$$dE(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_{i=1}^{m}(a_i - b_i)^2},$$

note que os elementos dos vetores são os atributos, no caso da base de dados Iris, largura e comprimento da pétala e sépala, com $m = 4$.

2.3 Exemplo de execução do kNN

Seja a Tabela 1 o conjunto de dados D, o qual iremos chamar de **conjunto de treino**, pois o kNN utilizará estes dados para calcular a distância relativa a um novo exemplo a ser classificado. E seja o conjunto de dados descritos na Tabela 2, três novos exemplos que queremos classificar, o qual iremos chamar de **conjunto de teste**, pois vamos usar o kNN para descobrir as classes referentes a estes exemplos.

	Comprimento sépala(cm)	**Largura sépala(cm)**	**Comprimento pétala(cm)**	**Largura pétala(cm)**	**Espécie**
1	5.0	3.0	1.5	0.5	**?**
2	6.0	3.0	2.0	0.3	**?**
3	6.1	3.4	6.1	2.4	**?**

Tabela 2: Exemplos que deseja-se identificar a classe.

Calculando a distância Euclidiana do primeiro exemplo do conjunto de teste para todos os exemplos no conjunto de treino, temos os seguintes valores:

$$(1)\ \sqrt{(5.3-5.0)^2+(3.7-3.0)^2+(1.5-1.5)^2+(0.2-0.5)^2} = 0.67$$
$$(2)\ \sqrt{(7.0-5.0)^2+(3.2-3.0)^2+(4.7-1.5)^2+(1.4-0.5)^2} = 15.59$$
$$(3)\ \sqrt{(6.3-5.0)^2+(3.3-3.0)^2+(6.0-1.5)^2+(2.5-0.5)^2} = 26.03$$
$$(4)\ \sqrt{(5.0-5.0)^2+(3.3-3.0)^2+(1.4-1.5)^2+(0.2-0.5)^2} = 0.19$$
$$(5)\ \sqrt{(6.4-5.0)^2+(3.2-3.0)^2+(4.5-1.5)^2+(1.5-0.5)^2} = 12.00$$
$$(6)\ \sqrt{(5.8-5.0)^2+(2.7-3.0)^2+(5.1-1.5)^2+(1.9-0.5)^2} = 15.65$$

Então, para $k = 3$, as linhas 4, 1 e 5 indicam os vizinhos mais próximos e a classe `setosa` deve ser atribuída a este exemplo. Note que para $k = 5$ haverá um empate, isto é, 2 vizinhos da classe `setosa` e 2 vizinho da classe `versicolor`. Existem diversas formas de tratar este problema, uma delas é atribuir a classe do vizinho com menor distância, que neste caso é a `setosa` (0.19).

Já para $k = 3$, o segundo exemplo do conjunto de treino deve ser classificado como `setosa` e o terceiro como `virginica`. Assim, as linhas 1, 2 e 3 do conjunto de treino para $k = 3$ devem ser classificadas pelo kNN como `setosa`, `setosa` e `virginica` respectivamente.

Suponha que algum especialista em botânica tenha olhado as três amostras do conjunto de teste e classificando-as em `setosa`, `versicolor` e `virginica`. Repare que o segundo exemplo foi classificado errado segundo o especialista. Uma forma de mensurar a taxa de acerto do kNN é dividir o número de acertos do conjunto de teste, isto é, aqueles que condizem com o que o especialista espera, pelo número de exemplos no conjunto de teste. Desta maneira, para este exemplo a taxa de acerto é $\frac{2}{3} \approx 0.6667$.

3 Especificação

Implemente um programa chamado `knn`, usando a linguagem `C`, que use o kNN para classificar diferentes espécies da flor Iris. Seu programa receberá 3 entradas via teclado, o nome do arquivo de treino, o nome do arquivo de teste e o número de vizinhos k. O formato de uso será o seguinte:

```
./knn
train.csv test.csv 3\n
```

Ou seja, o programa recebe o nome do arquivo CSV com os exemplos de treinamento, do arquivo com os exemplos de teste, e o parâmetro k do algoritmo.

O número de vizinhos k deve estar entre 1 e o número máximo de exemplos no conjunto de treinamento, caso contrário a seguinte mensagem deve ser exibida:

```
k is invalid\n
```

Você deve executar o kNN utilizando o conjunto de treino para calcular as distâncias e o de teste para a classificação. Os arquivos de treino e teste estarão no formato `CSV`, onde cada elemento estará separado por virgula (`,`) e strings por aspas (`" "`). Os conjuntos de treino na Tabelas 1 e teste na Tabela 2, são apresentados no formato `CSV` abaixo. Note que o conjunto de teste está com a classificação feita pelo especialista.

`[train.csv]`

```
"Sepal.Length","Sepal.Width","Petal.Length","Petal.Width","Species"\n
5.3,3.7,1.5,0.2,"setosa"\n
7.0,3.2,4.7,1.4,"versicolor"\n
6.3,3.3,6.0,2.5,"virginica"\n
5.0,3.3,1.4,0.2,"setosa"\n
6.4,3.2,4.5,1.5,"versicolor"\n
5.8,2.7,5.1,1.9,"virginica"\n
```

`[test.csv]`

```
"Sepal.Length","Sepal.Width","Petal.Length","Petal.Width","Species"\n
5.0,3.0,1.5,0.5,"setosa"\n
6.0,3.0,2.0,0.3,"versicolor"\n
6.1,3.4,6.1,2.4,"virginica"\n
```

Você deve imprimir na tela a classe encontrada pelo kNN seguida pela classe verdadeira (a que o especialista classificou) para cada exemplo do conjunto de teste. No final, imprima a taxa de acerto do kNN. Para o exemplo da subseção 2.3 com $k = 3$ a seguinte saída é esperada:

```
setosa setosa\n
setosa versicolor\n
virginica virginica\n
0.6667\n
```

Em caso de empate no kNN, retorne a classe do vizinho mais próximo. Use 4 casas decimais para imprimir a taxa de acerto. Se alocação dinâmica for utilizada, seu programa não pode apresentar vazamento de memória.

3.1 Entrega e Avaliação

O trabalho será avaliado levando em consideração:

1. Realização dos objetivos / lógica utilizada
2. Uso de comentários e estrutura no código (e.g. indentação, legibilidade, modularização)
3. Resultado da plataforma run.codes
4. Eficiência da implementação

ATENÇÃO:

- O projeto deverá ser entregue apenas pelo (`http://run.codes`) no formato de **código fonte**, ou seja apenas o código C.
- O prazo está no sistema run.codes
- Em caso de projetos **copiados** de colegas ou da Internet, todos os envolvidos recebem nota zero. Inclui no plágio a cópia com pequenas modificações, cópia de apenas uma parte ou função. Portanto programe seu próprio trabalho.