Aprendizagem de Máquina

k-Vizinhos mais Próximos

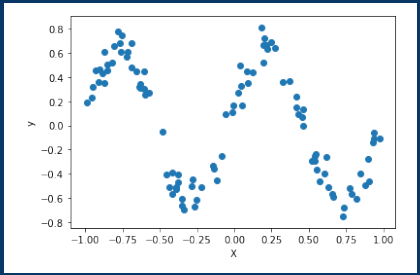
Telmo de Menezes e Silva Filho
tmfilho@gmail.com/telmo@de.ufpb.br
www.de.ufpb.br

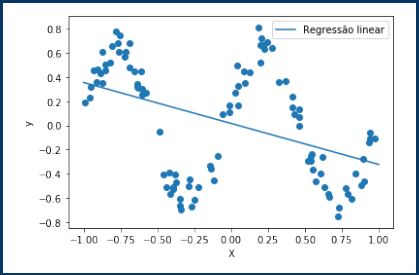
UFPB Departamento de ESTATÍSTICA

Sumário

Introdução

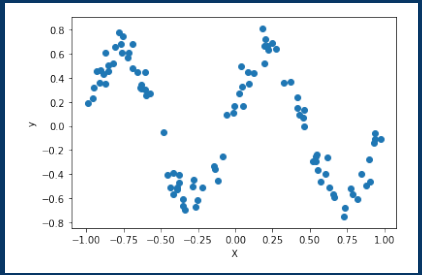
- Nesta aula, daremos início ao estudo de modelos não-paramétricos
  - Modelos que não assumem que os dados seguem formas pré-definidas
- Nem sempre é possível modelar os dados por meio de funções lineares

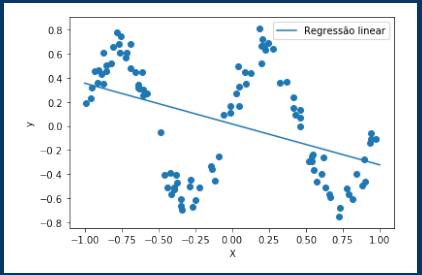




Introdução

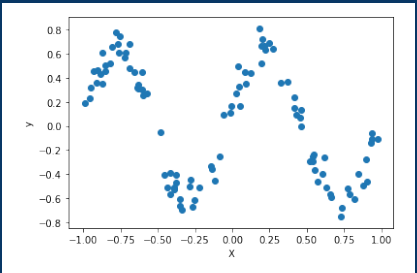
- Nesse caso temos Y como uma função senoidal de X
- Já que conseguimos visualizar isso, poderíamos tentar assumir essa relação entre as variáveis e tentar encontrar seus parâmetros

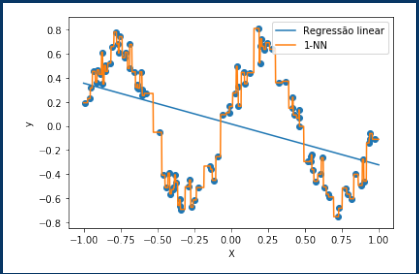




Introdução

- Porém, podemos tentar resolver esse problema de forma mais simples
- Para qualquer função, é razoável assumir que valores próximos de X devem assumir valores próximos de Y
- Essa é a motivação do método dos k-vizinhos mais próximos
  - Um dos algoritmos mais antigos e mais intuitivos da AM





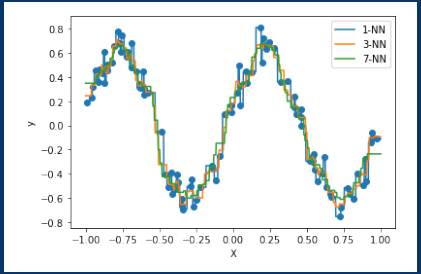
k-Vizinhos mais Próximos e Regressão

k-Vizinhos mais Próximos e Regressão

- Métodos de vizinhos mais próximos (kNN) usam os k dados de treinamento mais próximos ao vetor $\mathbf{x}$ para decidir sua resposta $\hat{y}$
- Para tarefas de regressão, podemos escrever:

$$\hat{y} = \frac{1}{k} \sum_{i \in N_k(\mathbf{x})} y_i$$

- Onde $N_k(\mathbf{x})$ é o conjunto dos k índices dos pontos que definem a vizinhança de $\mathbf{x}$

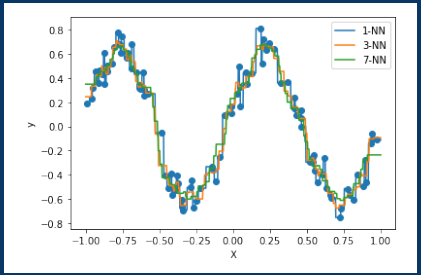


k-Vizinhos mais Próximos e Regressão

- Métodos de vizinhos mais próximos (kNN) usam os k dados de treinamento mais próximos ao vetor $\mathbf{x}$ para decidir sua resposta $\hat{y}$
- Para tarefas de regressão, podemos escrever:

$$\hat{y} = \frac{1}{k} \sum_{i \in N_k(\mathbf{x})} y_i$$

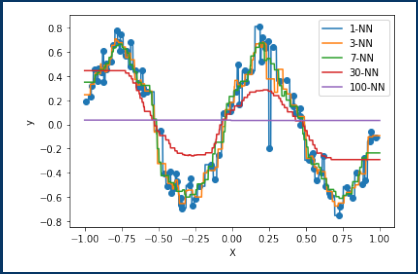
- Em outras palavras, predizemos a **média** dos valores de y dos vizinhos de $\mathbf{x}$



k-Vizinhos mais Próximos e Regressão

- O valor de *k* é bastante importante
- Valores pequenos podem resultar em predições mas **precisas**, porém mais sensíveis a **ruídos** e **anomalias**
- Por outro lado, valores maiores são mais robustos a esses problemas nos dados de treinamento, mas podem perder informação

- Normalmente, otimizamos o valor de *k* para cada conjunto de dados
- A fase de treinamento consiste apenas em armazenar os dados observados



Mas como Determinamos a Vizinhança?

- Para dados contínuos, comumente usa-se a distância Euclidiana (ou distância L_2:

$$d(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{\sum_{j=1}^{p} (x_{1j} - x_{2j})^2}$$

- Outras distâncias:
  - Distância Manhattan, City block ou L_1:

$$d(\mathbf{x}_1, \mathbf{x}_2) = \sum_{j=1}^{p} |x_{1j} - x_{2j}|$$

  - Cosseno (similaridade):

$$cos(\mathbf{x}_1, \mathbf{x}_2) = \frac{\mathbf{x}_1 \cdot \mathbf{x}_2}{||\mathbf{x}_1|| ||\mathbf{x}_2||}$$

Mas como Determinamos a Vizinhança?

- Pode ser muito útil normalizar os dados antes de aplicar o kNN
- Devido às fórmulas das distâncias, variáveis com escalas muito maiores do que outras terão mais peso no seu cálculo
  - Principalmente no caso da distância Euclidiana

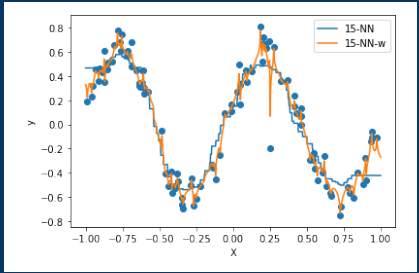
k-Vizinhos mais Próximos Ponderados

- Como vimos, valores maiores de k podem suavizar os valores preditos
- No entanto, essa suavização pode ser “demais”
- Uma alternativa para perder menos informação é dar **pesos** para os vizinhos de acordo com suas distâncias

$$\hat{y} = \sum_{i \in N_k(\mathbf{x})} w_i y_i$$

- Onde

$$w_i = \frac{\frac{1}{d(\mathbf{x}, \mathbf{x}_i) + \epsilon}}{\sum_{N_k(\mathbf{x})} \frac{1}{d(\mathbf{x}, \mathbf{x}_l) + \epsilon}}$$



k-Vizinhos mais Próximos e Classificação

k-Vizinhos mais Próximos e Classificação

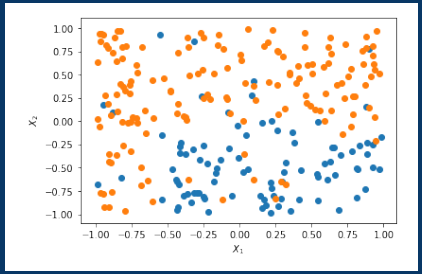
- O kNN é facilmente aplicado também à tarefa de classificação
- Aqui, a ideia é que indivíduos devem pertencer à mesma classe que seus vizinhos
- Assim, podemos predizer

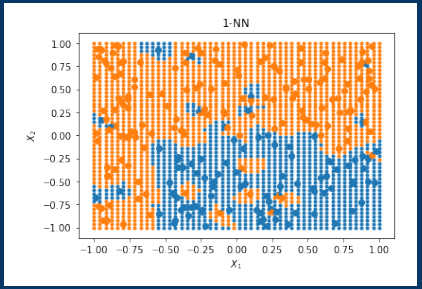
$$\hat{y} = \arg \max_{c \in \{1,\dots K\}} \sum_{i \in N_k(\mathbf{x})} \mathbb{1}(y_i = c)$$

  - Onde $\mathbb{1}(y_i = c) = 1$ se i pertencer à classe c e $\mathbb{1}(y_i = c) = 0$, caso contrário
- Em outras palavras, atribuímos **x** à **classe mais frequente** entre seus vizinhos

k-Vizinhos mais Próximos e Classificação

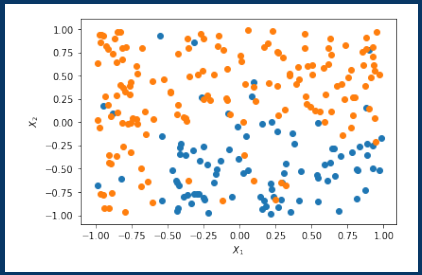
- Naturalmente, a atribuição das classes forma uma fronteira de decisão que não precisa ser linear
- Assim como na tarefa de regressão, o valor de k é muito importante
- Com $k = 1$, podemos chegar a modelar possíveis ruídos nos dados

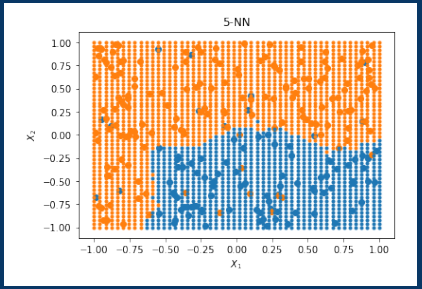




k-Vizinhos mais Próximos e Classificação

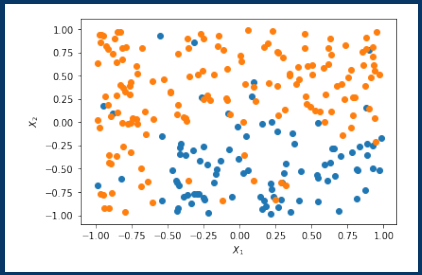
- Naturalmente, a atribuição das classes forma uma fronteira de decisão que não precisa ser linear
- Assim como na tarefa de regressão, o valor de k é muito importante
- Aqui, com $k = 5$ evitamos os ruídos

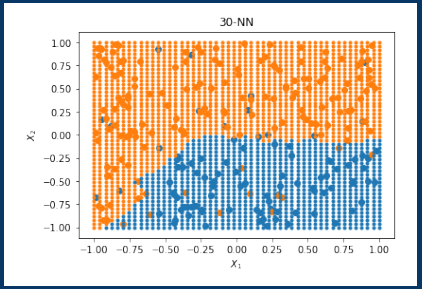




k-Vizinhos mais Próximos e Classificação

- Naturalmente, a atribuição das classes forma uma fronteira de decisão que não precisa ser linear
- Assim como na tarefa de regressão, o valor de k é muito importante
- À medida que k aumenta, a fronteira de decisão torna-se mais suave





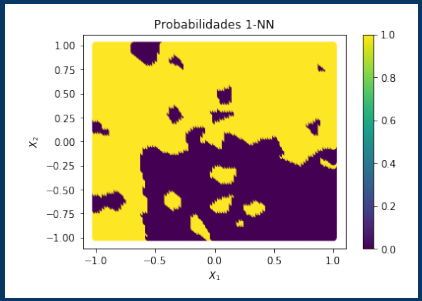
Estimando Probabilidades de Classe

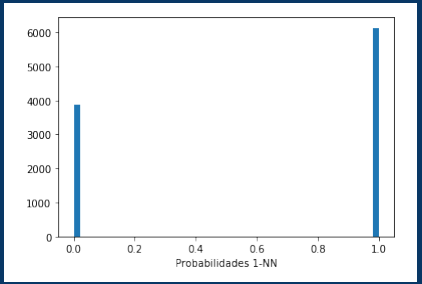
- Apesar de não assumir modelo probabilístico algum para os dados, o kNN pode ser usado para estimar $P(Y|X)$ localmente
- Para isso, ao invés de retornar a classe mais frequente entre os vizinhos do vetor **x**, retornamos as suas frequências

$$\hat{P}(Y = c|X = \mathbf{x}) = \frac{\sum_{i \in N_k(\mathbf{x})} \mathbb{1}(y_i = c)}{\sum_{m=1}^{K} \sum_{i \in N_k(\mathbf{x})} \mathbb{1}(y_i = m)}$$

Estimando Probabilidades de Classe

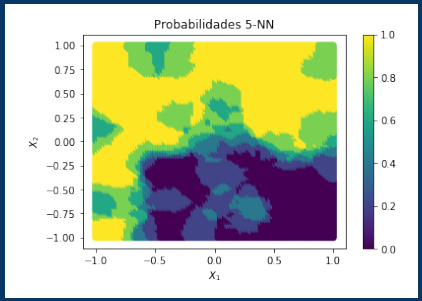
- Como sempre, valores maiores de k podem oferecer estimativas mais confiáveis, mas perdem informação local
- Nos gráficos abaixo, temos estimativas para $P(Y = 1|X = \mathbf{x})$

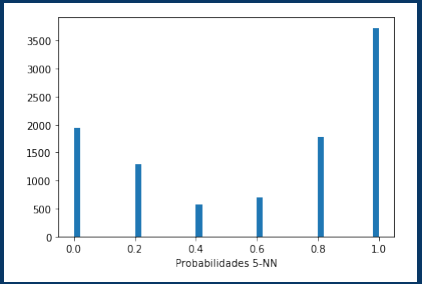




Estimando Probabilidades de Classe

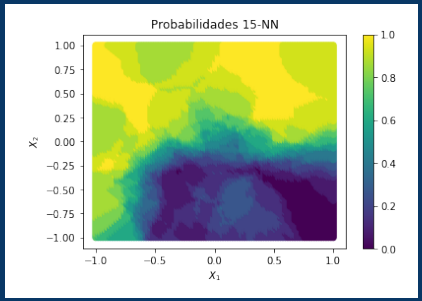
- Como sempre, valores maiores de k podem oferecer estimativas mais confiáveis, mas perdem informação local
- Nos gráficos abaixo, temos estimativas para $P(Y = 1|X = \mathbf{x})$

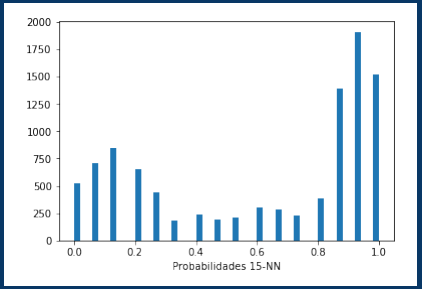




Estimando Probabilidades de Classe

- Como sempre, valores maiores de k podem oferecer estimativas mais confiáveis, mas perdem informação local
- Nos gráficos abaixo, temos estimativas para $P(Y = 1|X = \mathbf{x})$





Estimando Probabilidades de Classe

- No caso mais extremo, se usarmos $k = N$, as probabilidades estimadas para todos os dados serão sempre iguais às probabilidades a priori das classes, ou seja

$$\hat{P}(Y = 1|X = \mathbf{x}) = \pi_1$$
$$\hat{P}(Y = 0|X = \mathbf{x}) = \pi_0$$

k-Vizinhos Ponderados

- Assim como na regressão, podemos modificar o kNN para dar pesos aos vizinhos de acordo com suas distâncias para **x**
- Na tarefa de classificação, isso é particularmente útil quando lidamos com problemas desbalanceados
  - Ou seja, as classes tem quantidades de dados de treinamento (muito) diferentes

$$\hat{y} = \arg \max_{c \in \{1,\dots K\}} \sum_{i \in N_k(\mathbf{x})} \mathbb{1}(y_i = c) w_i$$

- Relembrando:

$$w_i = \frac{\frac{1}{d(\mathbf{x},\mathbf{x}_i)+\epsilon}}{\sum_{N_k(\mathbf{x})} \frac{1}{d(\mathbf{x},\mathbf{x}_l)+\epsilon}}$$

Exemplo:

i	$d(\mathbf{x}, \mathbf{x}_i)$	w_i	y_i
1	0.1	$\frac{1/0.1}{1/0.1+1+1/3} = 0.88$	1
2	1	$\frac{1}{1/0.1+1+1/3} = 0.09$	0
3	3	$\frac{1/3}{1/0.1+1+1/3} = 0.03$	0

k-Vizinhos Ponderados

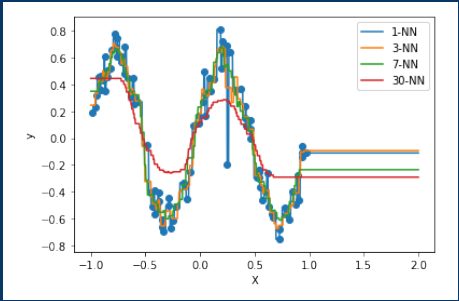
- O kNN ponderado também pode ser usado para estimar probabilidades localmente:

$$\hat{P}(Y = c|X = \mathbf{x}) = \frac{\sum_{i \in N_k(\mathbf{x})} \mathbb{1}(y_i = c) w_i}{\sum_{m=1}^{K} \sum_{i \in N_k(\mathbf{x})} \mathbb{1}(y_i = m) w_i}$$

Limitações e Aplicações

Limitações

- O modelo do kNN é totalmente dependente dos dados observados
- Para novos dados fora dos limites dos dados de treinamento, as predições do kNN deixam de ser úteis, particularmente na tarefa de regressão
- O cálculo de distâncias pode não ser ideal com grandes números de dimensões (tudo fica muito distante – maldição da dimensionalidade) e pode ser custoso para grandes conjuntos de treinamento



Aplicações

- O kNN pode ser muito útil em cenários em que o conjunto inicial de dados é pequeno, mas está em constante construção
  - Pode ser difícil ajustar outros modelos
  - Um exemplo é *Active Learning*
- Ele também costuma ser usado em explorações iniciais de soluções
- Por fim, o kNN pode ser aplicado à **detecção de outliers**, anomalias e instâncias fora de distribuição
  - Podemos escolher um limiar, de forma que se o vizinho mais próximo de **x** estiver mais distante que o limiar, consideramos **x** um outlier

Sugestões de Atividades

- Tente implementar o kNN ponderado para classificação
- Tente implementar a detecção de outliers usando kNN. Como você faria para determinar o limiar?

FIM

Aprendizagem de Máquina

k-Vizinhos mais Próximos

Telmo de Menezes e Silva Filho
tmfilho@gmail.com/telmo@de.ufpb.br
www.de.ufpb.br

UFPB Departamento de ESTATÍSTICA