k-Nearest Neighbors

1. Visão geral

Uma das estratégias mais simples para abordar os problemas de classificação e regressão está associada ao método dos k vizinhos mais próximos (kNN, do inglês *k-nearest neighbors*).

Este método é do tipo não-paramétrico, uma vez que não há um modelo a ser ajustado, tampouco se faz qualquer suposição a respeito dos dados.

Em linhas gerais, o kNN requer o armazenamento de todos os padrões de treinamento $\mathbf{x}(i) \in \mathbb{R}^{K\times 1}$, juntamente com as respectivas respostas desejadas $y(i), i = 0, \ldots, N-1$. Então, para um novo dado de entrada $\mathbf{x}'$, a saída gerada pelo kNN depende das saídas associadas aos k padrões de treinamento que estão mais

próximos à entrada $\mathbf{x}'$ no espaço de atributos. Por exemplo, pode-se tomar a média simples das saídas dos vizinhos mais próximos:

$$\hat{y}(\mathbf{x}') = \frac{1}{k} \sum_{\mathbf{x}(i) \in \mathcal{N}_k(\mathbf{x}')} y(i), \tag{1}$$

onde $\mathcal{N}_k(\mathbf{x}')$ denota a vizinhança de $\mathbf{x}'$, formada pelos padrões de treinamento $\mathbf{x}(i)$ que correspondem aos k vizinhos mais próximos a $\mathbf{x}'$.

Sendo assim, o uso do kNN envolve a definição de:

- Uma **métrica de distância** a ser calculada no espaço dos atributos a fim de determinar os vizinhos mais próximos;
- Um valor para o parâmetro k, i.e., **a escolha do número de vizinhos** que são levados em consideração na geração da saída.

Uma vez que k é um hiperparâmetro deste método, podemos utilizar uma abordagem de validação cruzada baseada em q pastas* para identificar o melhor valor de k.

Por conta destas características, o kNN é visto como um método de aprendizado competitivo, uma vez que os elementos do modelo – que são os próprios padrões de treinamento – competem entre si pelo direito de influenciar a saída do sistema quando a medida de similaridade (distância) é calculada para cada novo dado de entrada.

Além disso, o kNN explora a ideia de *lazy learning*, uma vez que o algoritmo não constrói um modelo até o instante em que uma predição é necessária. Isto traz o benefício de incluir apenas os dados relevantes para a análise do novo padrão de

* Para evitar confusões com o parâmetro k do kNN, denotamos por q o número de pastas utilizadas na técnica de validação cruzada.

entrada, sendo, por este motivo, um modelo do tipo localizado. Por outro lado, o kNN tem como desvantagem o fato de que todos os dados de treinamento precisam ser armazenados e consultados para se identificar os vizinhos mais próximos.

1.1. Métricas de distância

Distância de Minkowski de ordem p:

$$d(\mathbf{x};\mathbf{y}) = \left(\sum_{i=1}^{K} |x_i - y_i|^p\right)^{1/p}$$

Casos particulares:

Para $p = 1$, temos a distância de Manhattan: $d(\mathbf{x};\mathbf{y}) = \sum_{i=1}^{K} |x_i - y_i|$

Para $p = 2$, temos a distância Euclidiana: $d(\mathbf{x};\mathbf{y}) = \sqrt{\sum_{i=1}^{K} |x_i - y_i|^2}$

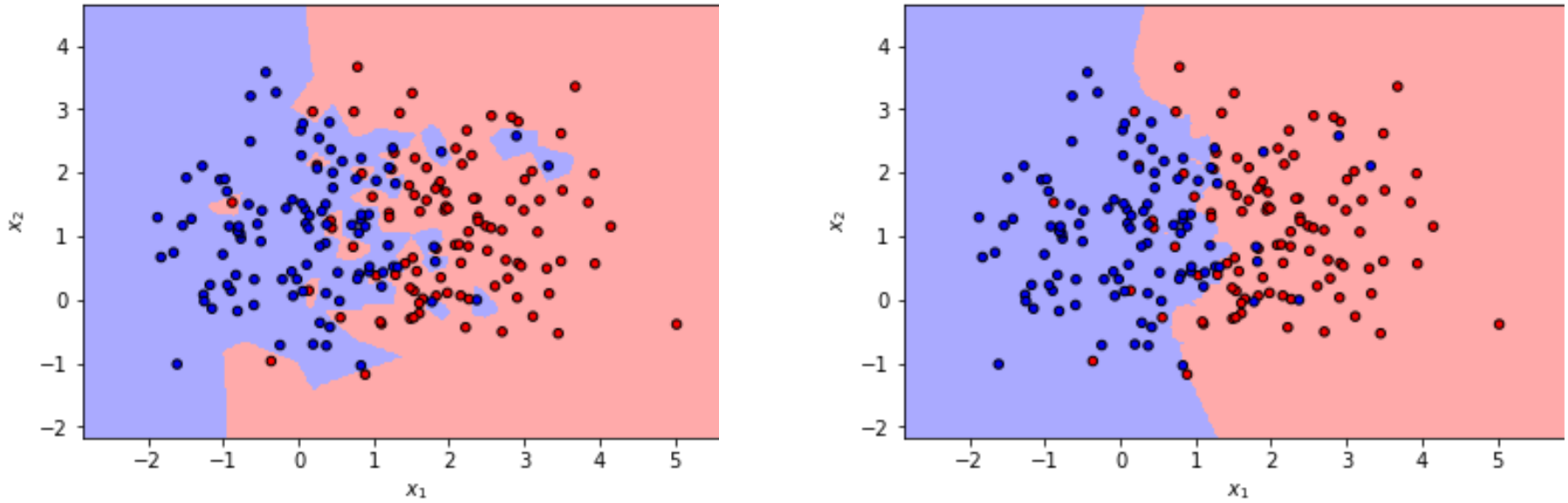
2. kNN para classificação

No âmbito do problema de classificação, a saída em (1) gerada pelo kNN equivale a tomar o voto majoritário dos k vizinhos mais próximos. Ou seja, um novo padrão $\mathbf{x}'$ é classificado como sendo pertencente à classe que contiver o maior número de vizinhos de $\mathbf{x}'$.

Observações:

- É possível também atribuir pesos diferentes à contribuição de cada vizinho à decisão final. Uma alternativa usual é definir os pesos como sendo inversamente proporcionais às distâncias dos vizinhos ao padrão de entrada $\mathbf{x}'$.
- Interessantemente, em (COVER & HART, 1967), foi demonstrado que a taxa de erro assintótica do classificador 1-NN nunca ultrapassa o dobro da taxa de erro mínima, dada pelo classificador bayesiano, ou MAP.

Exemplo: classificação binária



(a) $k = 1$ (b) $k = 5$

Figura 1 – Distribuição dos dados de treinamento e fronteira de decisão associada ao kNN para diferentes valores de k. À medida que k aumenta, a fronteira tende a ficar mais suave e menos regiões isoladas são criadas para cada classe.

3. kNN para regressão

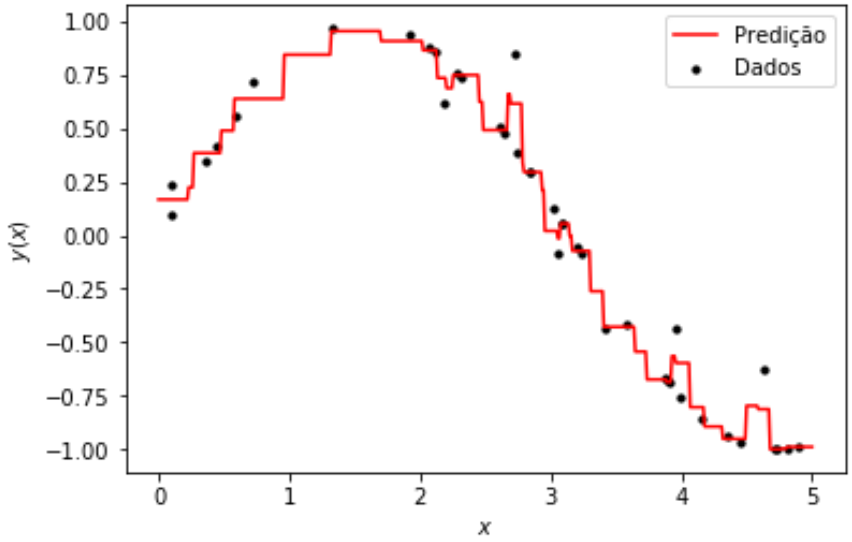
Seja $\mathcal{N}_k(\mathbf{x}')$ o conjunto formado pelos k padrões de treinamento $\mathbf{x} \in \mathbb{R}^{K\times 1}$ mais próximos ao dado de entrada $\mathbf{x}'$. As saídas associadas a estes padrões de treinamento são denotadas por $y_j(\mathbf{x} \in \mathcal{N}_k(\mathbf{x}')), j = 1, \ldots, k$.

Em regressão, a saída do kNN para um novo dado de entrada $\mathbf{x}'$ pode ser escrita de uma forma geral como:

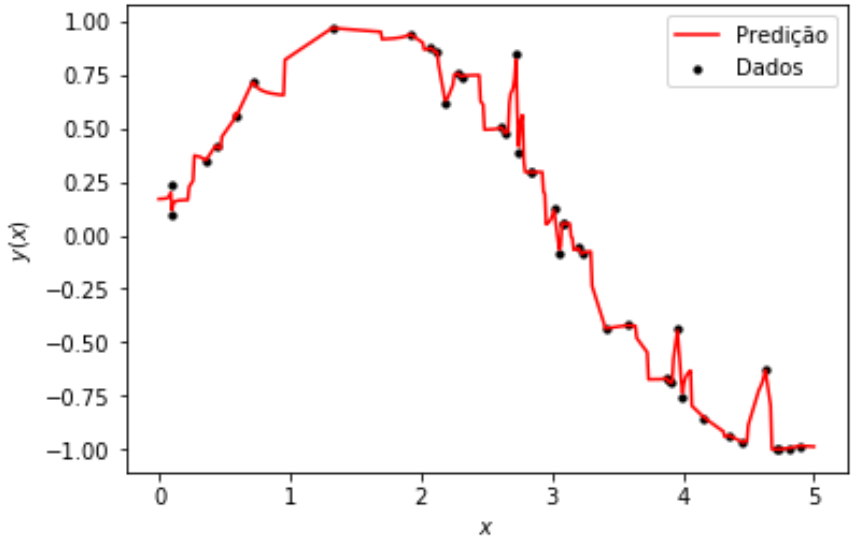
$$\hat{y}(\mathbf{x}') = \frac{\sum_{j=1}^{k} w_j y_j(\mathbf{x} \in \mathcal{N}_k(\mathbf{x}'))}{\sum w_j}, \tag{2}$$

onde $w_j, j = 1, \ldots, k$ representa o peso associado ao j-ésimo vizinho de $\mathbf{x}'$.

Exemplo:



(a) $k = 2$, pesos uniformes



(b) $k = 2$, pesos inversamente proporcionais à distância

Figura 2 – Dados de treinamento e mapeamento gerado pelo kNN.

4. Referências bibliográficas

ALPAYDIN, E. **Introduction to Machine Learning**. MIT Press. 3rd edition. 2014.

ALTMAN, N. S. An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression, *The American Statistician*, vol. 46, pp. 175-185, 1992.

BISHOP, C. M. **Pattern Recognition and Machine Learning**. Springer. 2006.

COVER, T. M., HART, P. Nearest Neighbor Pattern Classification. *IEEE Transactions on Information Theory*, IT-11, pp. 21-27, 1967.

DUDA, R. O., HART, P. E., STORK, D. G. **Pattern Classification**. John Wiley & Sons. 2nd edition, 2001.

HASTIE, T., TIBSHIRANI, R., FRIEDMAN, J. **The Elements of Statistical Learning: Data Mining, Inference and Prediction**. Springer. 2nd edition, 2009.