Pattern Recognition

Lecture 11. Generative Methods II: non-Parametric methods: KNN

Dr. Shanshan ZHAO

shanshan.zhao@xjtlu.edu.cn

School of AI and Advanced Computing

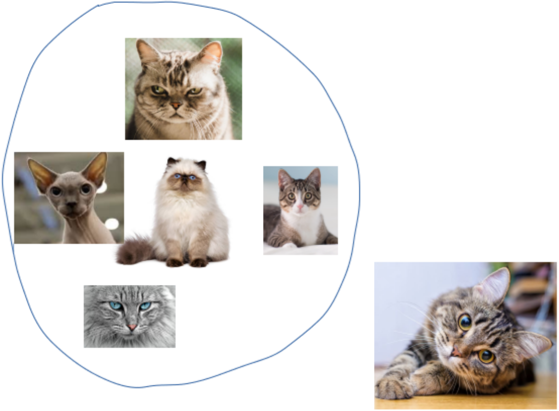
Xi'an Jiaotong-Liverpool University

Academic Year 2021-2022

Table of Contents

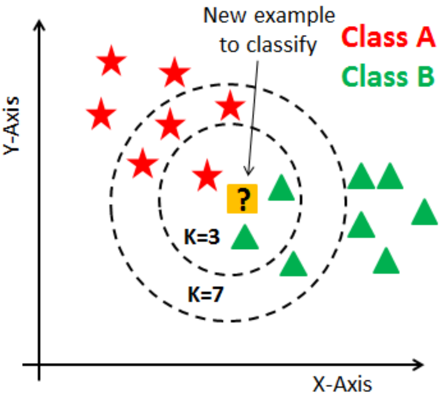
K-Nearest Neighbor Classification problem

Intuition



K-Nearest Neighbor Classification problem

Basic idea



K-Nearest Neighbor Density Estimation

Theory in [1]

$$p_n(x) = \frac{k_n/n}{V_n} \tag{1}$$

where p_n is the approximation to p(x), which is the density function we want to estimate. n is the number of samples.
How to interpret it?

K-Nearest Neighbor Density Estimation

Let $x_1, x_2, ..., x_n$ be our random samples. Assume each observation has d different variables; namely, $x_i \in \mathcal{R}^d$. For each given point x, we first rank every observation based on its distance to x. Let $R_k(x)$ denotes the distance from x to its k-th nearest neighbor point.

For a given point x, the KNN density estimator estimates the density by

$$
\begin{aligned}
p_n(x) &= \frac{k}{n} \frac{1}{V_d \cdot R_k^d(x)} &(2)\\
&= \frac{k}{n} \cdot \frac{1}{\text{Volume of a } d\text{-dimensional ball with radius being } R_k(x)} &(3)
\end{aligned}
$$

Where $V_d = \frac{\pi^d 2}{\Gamma(\frac{d}{2}+1)}$ is the volume of a unit d-dimensional ball and $\Gamma(x)$ is the Gamma function.

Here are the results when $d = 1, 2$, and 3

- $d = 1$, $V_1 = 2 : p_n(x) = \frac{k}{n} \frac{1}{2R_k(x)}$.
- $d = 2$, $V_2 = \pi : p_n(x) = \frac{k}{n} \frac{1}{\pi R_k^2(x)}$.
- $d = 3$, $V_3 = \frac{4}{3}\pi : p_n(x) = \frac{k}{n} \frac{3}{4\pi R_k^3(x)}$.

K-Nearest Neighbor Density Estimation

Example

We consider a simple example in $d = 1$. Assume our data is $X = \{1, 2, 6, 11, 13, 14, 20, 33\}$.

(i) What is the KNN density estimator at $x = 5$ with $k = 2$?

Solution: First, we calculate $R_2(5)$. The distance from $x = 5$ to each data point in X is $\{4, 3, 1, 6, 8, 9, 15, 28\}$. Thus, $R_2(5) = 3$ and

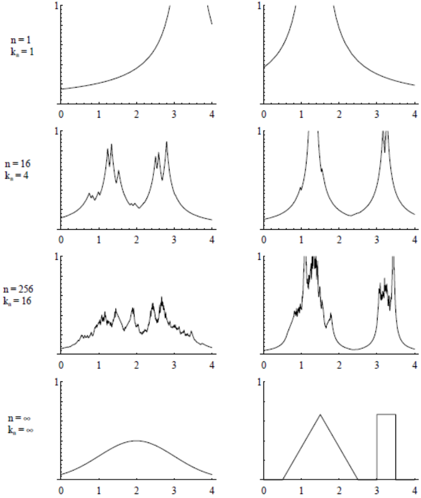
$$p(5) = \frac{2}{8}\frac{1}{2 \cdot R_2(5)} = \frac{1}{24} \tag{4}$$

(ii) What will the density estimator be when we choose $k = 5$?

In this case, $R_5(5) = 8$, so

$$p(5) = \frac{5}{8}\frac{1}{2 \cdot R_5(5)} = \frac{5}{128} \tag{5}$$

K-Nearest Neighbor Density Estimation



Several k-nearest-neighbor estimates of two unidimensional densities: a Gaussian and a bimodal distribution. Notice how the finite n estimates can be quite “spiky.”[1]

k-NN Posterior Estimation for Classification

- We can directly apply the k-NN methods to estimate the posterior probabilities $P(\omega_i|x)$ from a set of n labeled samples.
- Place a window of volume V around x and capture k samples, with k_i turning out to be of label ω_i.
- The estimate for the joint probability is thus

$$p_n(x, \omega_i) = \frac{k_i}{nV} \tag{6}$$

- A reasonable estimate for the posterior is thus

$$P_n(\omega_i|x) = \frac{p_n(x, \omega_i)}{\sum_i^N p_n(x, \omega_i)} = \frac{k_i}{k} \tag{7}$$

- The Posterior probability for ω_i is simply the fraction of samples within the window that are labeled ω_i. This is a simple as well as an intuitive result.

Pros and Cons

Pros	Cons
▪ Extremely easy to implement ▪ It is a lazy algorithm and therefore requires no training prior to making predictions. This makes KNN much faster ▪ There are only two parameters required to implement KNN, i.e., the value of k and the distance function (e.g., Euclidean or Manhattan etc.)	▪ KNN doesn't work well with high dimensional data because it becomes difficult to calculate the distance. ▪ KNN has high prediction cost for large datasets. ▪ KNN doesn't work well with categorical features since it is difficult to find the distance.

Codes

Learn how to do classification with KNN.

KNN.ipynb

Think: How to choose K?

Just as the bandwidth in the KDE, it is a very difficult problem in practice. However, according to theoretical analysis, we a rough idea about how k should be changing with respect to the sample size n. (check the material provided on LMO if you are interested)

Reference I

[1] Richard O Duda, Peter E Hart, et al. **Pattern Classification**. 2nd ed. Wiley New York, 2000.

Thank You !

$\mathcal{Q}$ & $\mathcal{A}$