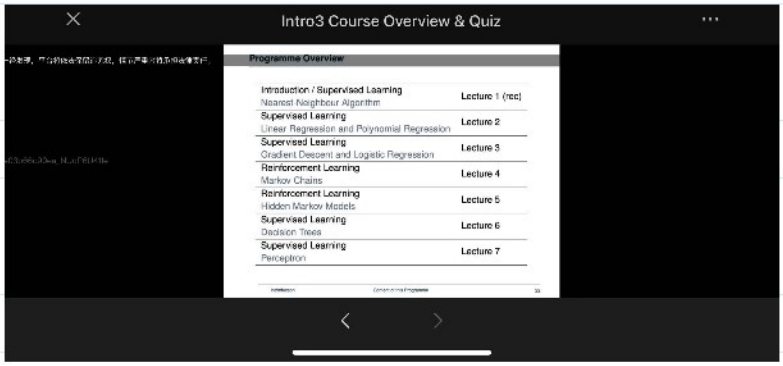
Pre-Learning: KNN

2021年1月26日 18:50



K-Nearest Neighbour (KNN)

(I) Introduction.

KNN basicly is a method of classification and regression. Its inputs are the feature vectors of samples. and the point mapped to the feature space., its outputs are the classes of samples.

[features, points.] ⟶ [classes]. → Discrete : Classification
↘ Continuous : Regression.

KNN actually has utilized training data sets to classify the space of feature vectors. which is its "Model".

Three Basic Points of KNN:

① The selection of the value K <K 值的选取>
② The measurement of the distance <距离的度量>
③ The decision rule of classification <分类决策规则>

(II) KNN Algorithm

Simple and Intuitive:

① Suppose a training data set.
② For a new input, To Find K nearest samples.
③ Classify the input to the class. which is the most of K nearest samples belong to.

Algorith 1: (KNN method)

Input: Training Data Set①

$$T = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$$

in which, $x_i \in X \subseteq R^n$ is the feature vector of samples.

$T = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$

in which, $x_i \in \mathcal{X} \subseteq R^n$ is the feature vector of samples.

$y_i \in \mathcal{Y} = \{c_1, c_2, \cdots, c_K\}$ is the class of samples. $i = 1, 2, \cdots N$

A sample vector x ②

Output: The class which the vector x belong to.

Mention: (※)

(1) According to the supposed distance value (距离度量), Find K points which are nearest to the vector x in the Training Data Set And the neighbour value range of x is referred to as $N_k(x)$ which are covered the K points.

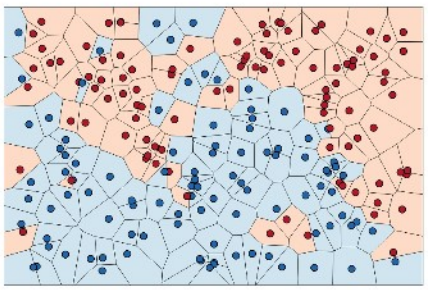
(2) At the range $N_k(x)$, determine the class of x based on the classification decision rule. (分类决策规则)

$$y = \arg\max_{c_j} \sum_{x_i \in N_k(x)} I(y_i = c_j) \qquad i = 1, 2, \cdots, N. \quad j = 1, 2, \cdots, K$$

in which, Function I is used for introduction. when $y_i = c_j$, $I = 1$

(三) KNN Model

- Cell (单元) → For each sample x_i, there is a group of points which is close to x_i than other points.



Every cell of points are determined Nearest Neighbour Method has labelled all of the points in that unit classes.

→ Measurement of Distance (距离度量)

设特征空间 $\mathcal{X}$ 是 n 维实数向量空间 R^n, $x_i, x_j \in \mathcal{X}$,
$x_i = \{x_i^{(1)}, x_i^{(2)}, \cdots, x_i^{(n)}\}^T$, $x_j = \{x_j^{(1)}, x_j^{(2)}, \cdots, x_j^{(n)}\}^T$, x_i 和 x_j 的 L_p 定义为

$$L_p(x_i, x_j) = \left(\sum_{l=1}^{n} |x_i^{(l)} - x_j^{(l)}|^p\right)^{\frac{1}{p}} \quad \text{in which} \quad p \geq 1.$$

① $p = 1$ 时, 称为曼哈顿距离 (Manhattan Distance)

① p=1 时，称为曼哈顿距离 < Manhattan Distance >

$$L_1(x_i, x_j) = \sum_{l=1}^{n} |x_i^{(l)} - x_j^{(l)}|$$

② P=2 时，称为欧式距离 < Euclidian Distance >

$$L_2(x_i, x_j) = \left(\sum_{l=1}^{n} |x_i^{(l)} - x_j^{(l)}|^2\right)^{\frac{1}{2}}$$

③ 当 p=∞ 时，它是各个坐标距离的最大值。

$$L_\infty(x_i, x_j) = \max_l |x_i^{(l)} - x_j^{(l)}|$$

Ex: 已知二维空间的三个点 $x_1=(1,1)^T$，$x_2=(5,1)^T$，$x_3=(4,4)^T$

试求在 p 取不同时，L_p 距离下 x_1 的最近临点。

Answer:

x_1 与 x_2 的第一维值相同，故 $L_p(x_1, x_2) \equiv 4$.

所由

$$L_p(x_i, x_j) = \left(\sum_{l=1}^{n} |x_i^{(l)} - x_j^{(l)}|^p\right)^{\frac{1}{p}}$$ 可得

p=1 时，$L_1(x_1, x_3) = 6$　p=2 时，$L_2(x_1, x_3) = 4.24$　p>3 时，$L_p = 3.78 < 4$

故在 p>3 时，x_1 的最近临点的 x_3 所 p≤2 时 是 x_2.

→ The Selection of K value < K 值的选择 >

The standard of selecting K value:

If K is higher ⇑, the feature space will be seperated by several more subspace. and makes the model more complex, and it's more easier to be overfitting. 训练时误、命中率高

K=N 时，无论输入什么，都将简单地预测为在训练实例中最多的类。

→ Classification Decision Rule

The classification rule of KNN always is Majority voting rule and it is about that the class of input is determined by the most of classes by K nearest value of it.

Explain:

If the classification loss function is [0, 1] and it is

$$f: R^n \to \{c_1, c_2, \cdots, c_K\}$$

Thus, the misclassification probability is

$$P(Y \neq f(X)) = 1 - P(Y = f(X))$$

Suppose a sample $x \in \mathcal{X}$, the nearest point consist $N_k(x)$. If the most of classes in this set is C_j, the misclassification:

$$\frac{1}{k}\sum_{x_i \in N_k(x)} I(y_i \neq C_j) = 1 - \frac{1}{k}\sum_{x_i \in N_k(x)} I(y_i = C_j)$$

要使误分类小，就要使( )大

(IV). Realization <Kd Tree>

Linear Scan is the simplest method. But it hasn't a excellent peformance at big data.

→ Construction of Kd Tree

- Description: Kd Tree is a binary tree①. Constructing a kd tree is equivlant to continuosly seperate K-dimension space by a hyperplane which is vertical to frame Axis②. Every nodes of a kd Tree corresponds to a K-dimensions hyperrectangle area③.

- Method:

① 构造根结点，使根结点对应于K维空间中包含所有实例点的矩形实例。
② 通过递归方法，不断对空间划分，生成子结点。
③ 直到子区域内没有实例为止。

- Algorithm: <构造平衡 Kd 树>

Input: k-d space Data set $T = \{x_1, x_2, \cdots, x_n\}$, in which $x_i = \{x_i^{(1)}, x_i^{(2)}, \cdots, x_i^{(k)}\}^T$, $i \in 1, 2, \cdots, N$

Output: Kd Tree.

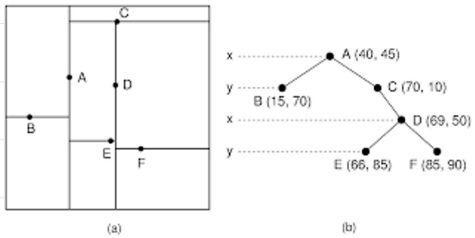
(1) 开始：构造根结点，对应于包含T的K维空间的矩形区域
(求) 选择 $x^{(1)}$ 为坐标轴，以T中所有实例 $x^{(1)}$ 坐标的中位数为切分点。
(2) 重复：对深度为 j 的结点，选择 $x^{(l)}$ 为切分的坐标轴，$l = j(\mathrm{mod}\ k) + 1$ 以 $x^{(l)}$ 的中位数为切分点。
(3) 停止：直到子区域不再有实例。

Ex:

给定一个数据集 $T = \{(15, 70), (40, 45), (70, 10), (69, 50), (66, 85)$

Ex:

给定一个数据集 $T=\{(15,70),(40,45),(70,10),(69,50),(66,85),(85,90)\}$.



先验：选 $x^{(1)}$ 的中位数.

$Median\_x^{(1)}=\frac{15+40+70+69+66+85}{6}$

$=\frac{345}{6}=57.5$ 故选 40. $\Rightarrow$ (40, 45)

再取右节点，由于右节点只有一个，故不用讨论，

$Median\_x^{(1)}=\frac{10+50+85+90}{6}=\frac{235}{6}=37.5$ 故选 10. $\Rightarrow$ (70, 10)

注：右子结点对应于采样 $x^{(1)}$ 小于切分点的区域，右子结点反之。