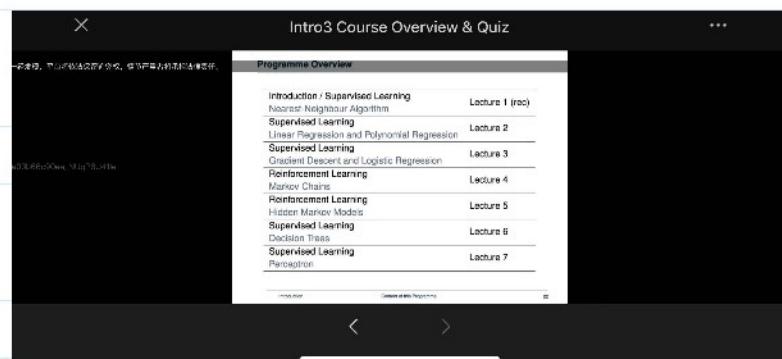


PS:

- ① Pre-Learning
- ② Inference TextBook:
"Machine Learning"
- ③ Code:
"Python Language"
- ④ 先导课



K-Nearest Neighbour (KNN)

(I) Introduction.

KNN basically 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.

Algorithm I: (KNN method)

Input: Training Data Set. ①

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

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

$$I = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$$

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

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

A sample vector x

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.

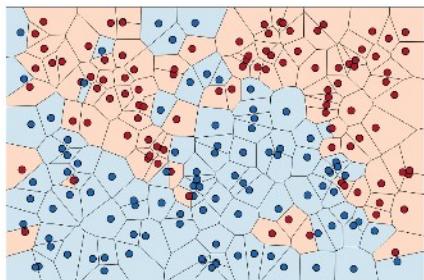
(2) At the range $N_k(x)$, determine the class of x based on the classification decision rule. (分类决策规则)

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

in which, Function I is need for introduction. When $y_i = c_j, I=1$

III) KNN Model

- Cell (单元) \rightarrow 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.

\rightarrow Measurement of Distance (距离度量)

设特征空间 X 是 n 维实数向量空间 \mathbb{R}^n , $x_i, x_j \in X$,
 $x_i = \{x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(n)}\}^T$, $x_j = \{x_j^{(1)}, x_j^{(2)}, \dots, x_j^{(n)}\}^T$, x_i 和 x_j 为 L_p 空间

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

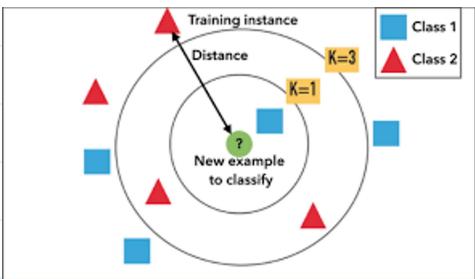
① $p=1$ 时, 称为曼哈顿距离 (Manhattan Distance)

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 < Euclidean Distance >

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



③ 当 $P=\infty$ 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 的最近邻点。

Answer:

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

而由

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

$P=1$ 时， $L_p(x_1, x_3) = 6$ $P=2$ 时， $L_p(x_1, x_3) = 4\sqrt{2}$ $P=3$ 时， $L_p = 3.78 < 4$

故在 $P \geq 3$ 时， x_1 的最近邻点为 x_3 而 $P \leq 2$ 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 separated by several more subspace. and makes the model more complex, and it's more easier to be overfitting. 训练时候，命中率高

$K=N$ 时，无论输入什么，都将简单地预测他训练实例中最多分类。

→ 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

$$f: R^n \rightarrow \{c_1, c_2, \dots, c_k\}$$

Thus, the misclassification probability is

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

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

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

要使误分类少，就要使 $()$

(IV). Realization < Kd Tree >

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

→ Construction of Kd Tree

◦ Description: Kd Tree is a binary tree. Constructing a kd tree is equivalent to continuously separate K-dimension space by a hyperplane which is vertical to frame axis. Every nodes of a kd Tree corresponds to a K-dimensions hyperrectangle area.

◦ 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的矩形实例。
- ② 通过递归方法，不断对空间剖分，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 $T = \{x_1, x_2, \dots, x_n\}$. In which
 $x_i = \{x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(k)}\}^T$, $i \in 1, 2, \dots, n$

Output: Kd Tree.

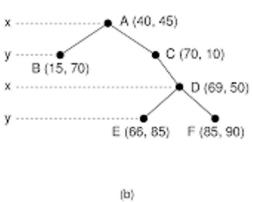
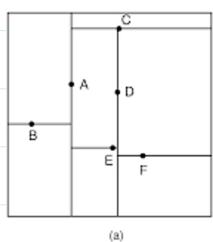
- (1) 开始：构造根结点，对应于包含 T 的 k 维空间的矩形区域
- (2) 选择 $x^{(l)}$ 为坐标轴，在 T 中所有实例 $x^{(l)}$ 的中位数为切分点。
- (3) 重复：对深度为 l 的结点，选择 $x^{(l)}$ 为切分的坐标轴，
 $b = j \bmod k + 1$ 为 $x^{(l)}$ 的中位数为切分点。
- (4) 停止：直到子区域内没有实例。

Ex:

给定一个数据集 $T = \{(15, 70), (40, 45), (70, 10), (69, 58), (66, 88)\}$.

Q1:

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



方法：选 $x^{(1)}$ 作为中位数

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

$$= \frac{345}{6} = 57.5 \quad \text{故选 } 40.$$

$(40, 45)$

再取右半部，由于左半部只有一个，故不讨论。

$$\text{Median}_x^{(2)} = \frac{10 + 50 + 85 + 90}{6} = \frac{235}{6} = 37.5 \quad \text{故选 } 10. \rightarrow (70, 10)$$

* 左子结点对应于坐标 $20^{(1)}$ 小于切分点的区域，在上结点，反之。