KNN Algorithm

Introduction to Artificial Intelligence with Mathematics
Lecture Notes

Ganguk Hwang

Department of Mathematical Sciences
KAIST

K Nearest Neighbors Algorithm

The K-nearest neighbors (KNN) algorithm is a simple, easy-to-implement supervised machine learning algorithm that can be used to solve both classification and regression problems.

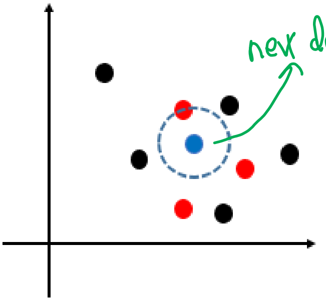
The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other.

info of neighbour point

Consider a new blue data and the objective is to classify it.

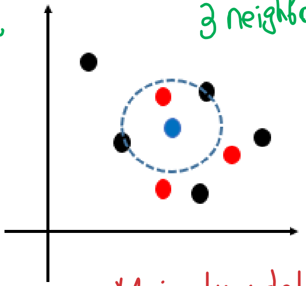
$K = 1$: classified as red

new data point



$K = 3$: classified as black

3 neighbour pts



Majority vote!

When used in regression, the value of a new data $\mathbf{z}$ is given by the average of the values of its K neighbors in $\mathcal{N}$.

↳ hyperparameter

Neighbours

Similarity can be measured in terms of *distance*. There are a number of different distance metrics as given below.

Distance metrics:

- Euclidean Distance

$$X = (x_1, x_2, \cdots, x_n), Y = (y_1, y_2, \cdots, y_n)$$
$$d_{(X,Y)} = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2}$$

- Manhattan Distance

$$d_{Manhattan(X,Y)} = \sum_{i=1}^{n} |x_i - y_i|$$

- Mahalanobis Distance: It takes into account the covariances of the data set. For two vectors $\mathbf{x}$ and $\mathbf{y}$

$$d_{Mahalanobis(\mathbf{x},\mathbf{y})} = \sqrt{(\mathbf{x}-\mathbf{y})^\top \Sigma^{-1}(\mathbf{x}-\mathbf{y})}$$

where Σ denotes the covariance matrix of the data set.

To understand Mahalanobis distance, consider two points $\mathbf{x} = (x_1, x_2)$ and $\mathbf{y} = (0,0)$ in $\mathbb{R}^2$. For $a_{12} = a_{21}$ and

$$\Sigma^{-1} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

Symmetric

if $d_{Mahalanobis(\mathbf{x},\mathbf{y})} = d$, then we have

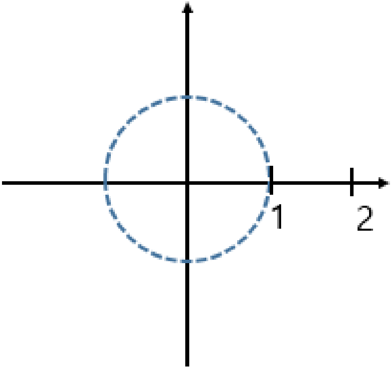
$$x_1^2 a_{11} + 2x_1x_2a_{12} + x_2^2 a_{22} = d^2$$

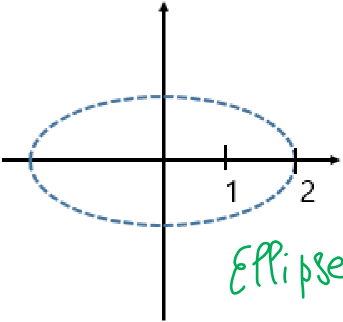
$$\Sigma^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$x_1^2 + x_2^2 = 1$$

$$\Sigma^{-1} = \begin{bmatrix} \frac{1}{4} & 0 \\ 0 & 1 \end{bmatrix}$$

$$\frac{1}{4}x_1^2 + x_2^2 = 1$$



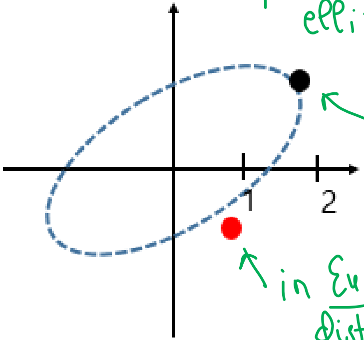


Ellipse

$$\Sigma^{-1} = \begin{bmatrix} \frac{1}{2} & -\sqrt{2} \\ -\sqrt{2} & 2 \end{bmatrix}$$

$$\frac{1}{2}x_1^2 - 2\sqrt{2}x_1x_2 + 2x_2^2 = 1$$

Rotated ellipse



Consider two points and determine which one is closer to $(0, 0)$.

Mahalanobis distance much near

in Euclidean distance, much near

- Pearson Correlation Distance

For $\mathbf{x} = (x_1, x_2, \cdots, x_n)$ and $\mathbf{y} = (y_1, y_2, \cdots, y_n)$

$$d_{Corr(\mathbf{x},\mathbf{y})} = 1 - \rho_{\mathbf{xy}} \in [0, 2]$$

correlation coefficient

where $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$, and

$$-1 \leq \rho_{\mathbf{xy}} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} \leq 1$$

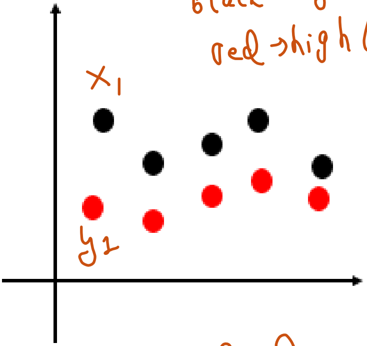
negatively strongly $\rightsquigarrow d \simeq 2$

positively $\rightsquigarrow d \simeq 0$

$\rho_{\mathbf{xy}}$ is close to 1, and hence $d_{Corr(\mathbf{x},\mathbf{y})}$ is small.

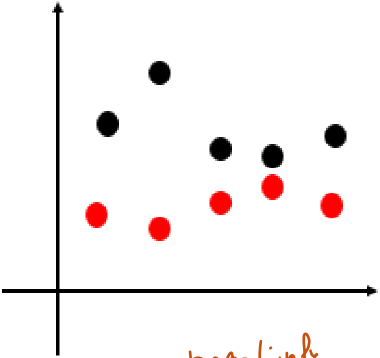
black → high (small)

red → high (small)



positively correlated

$\rho_{\mathbf{xy}}$ is close to -1, and hence $d_{Corr(\mathbf{x},\mathbf{y})}$ is large. ≈ 2



negatively correlated

- Spearman's Rank Correlation

$$\rho_{\mathbf{xy}}^{(r)} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}(r_j - r_i)(s_j - s_i)}{\sum_{i=1}^{n}\sum_{j=1}^{n}(r_j - r_i)^2}$$

where r_i and s_i denote the ranks of x_i and y_i in $\mathbf{x}$ and $\mathbf{y}$, respectively.

x_1 - largest $\Rightarrow r_1 = 1$

x_2 - smallest $\Rightarrow r_2 = n$

We see that

$$\rho_{\mathbf{xy}}^{(r)} = 1 - \frac{6}{n(n^2-1)}\sum_{i=1}^{n}(r_i - s_i)^2$$

$$
\begin{aligned}
&\sum_{i=1}^{n}\sum_{j=1}^{n}(r_j - r_i)(s_j - s_i) \\
&\quad = \sum_{i=1}^{n}\sum_{j=1}^{n} r_i s_i + \sum_{i=1}^{n}\sum_{j=1}^{n} r_j s_j - \sum_{i=1}^{n}\sum_{j=1}^{n} r_i s_j - \sum_{i=1}^{n}\sum_{j=1}^{n} r_j s_i \\
&\quad = 2n\sum_{i=1}^{n} r_i s_i - 2\sum_{i=1}^{n} r_i \sum_{j=1}^{n} s_j \rightarrow \underline{\text{ranks}} \\
&\quad = 2n\sum_{i=1}^{n} r_i s_i - 2\left(\frac{1}{2}n(n+1)\right)^2 \\
&\quad = 2n\sum_{i=1}^{n} r_i s_i - \frac{1}{2}n^2(n+1)^2
\end{aligned}
$$

From $\sum_{i=1}^{n}(r_i - s_i)^2 = 2\sum r_i^2 - 2\sum r_i s_i$, we have

$$\sum_{i=1}^{n}\sum_{j=1}^{n}(r_j - r_i)(s_j - s_i) = 2n\sum_{i=1}^{n} r_i^2 - \frac{1}{2}n^2(n+1)^2 - n\sum_{i=1}^{n}(r_i - s_i)^2$$
$$= \frac{1}{6}n^2(n^2-1) - n\sum_{i=1}^{n}(r_i - s_i)^2.$$

numerator

Further,

$$\sum_{i=1}^{n}\sum_{j=1}^{n}(r_j - r_i)^2 = 2n\sum_{i=1}^{n} r_i^2 - 2\sum_{i=1}^{n}\sum_{j=1}^{n} r_i r_j$$
$$= 2n\sum_{i=1}^{n} r_i^2 - 2\left(\sum_{i=1}^{n} r_i\right)^2 = \frac{1}{6}n^2(n^2-1)$$

denominator

and thus, by substituting into the original formula these results we get

$$\rho_{\mathbf{xy}}^{(r)} = 1 - \frac{6\sum_{i=1}^{n}(r_i - s_i)^2}{n(n^2-1)}.$$

How to choose K?

- For a validation set we perform the algorithm several times with different values of K.
- We select the value K that minimizes the misclassification error.

The KNN algorithm has to compute all distances between a new data and neighbor data, which results in high computational cost. There exist some algorithms to reduce the computational cost such as

- Locality Sensitive Hashing
- Network based Indexer
- Optimized product quantization

Example. KNN for binary classification

We generate data by using `make_circles` function. It makes two circles whose centers are the same. We use K nearest neighbors algorithm for binary classification.

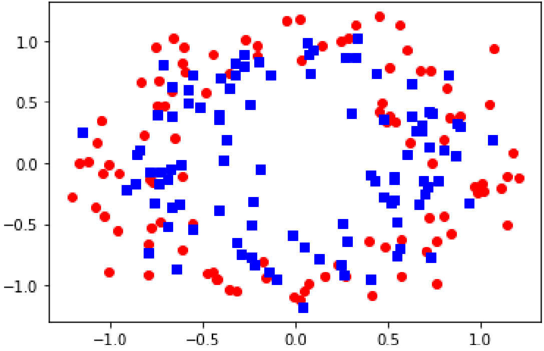


Figure: Randomly generated data $(N = 200)$

Decision boundary by KNN with $K = 3$ is as follows. This result is changed when we change K.

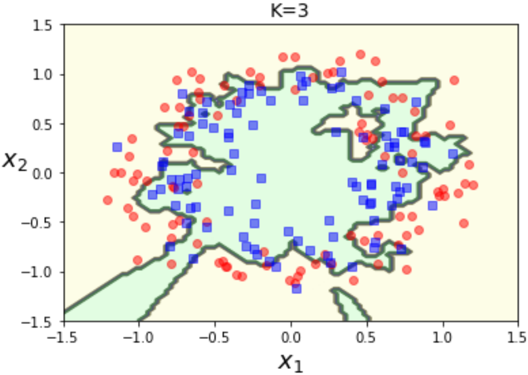


Figure: Decision boundary by KNN

Comparison with different K

Different boundaries
for several K values