10. K-Nearest Neighbors

K-Nearest Neighbor (KNN) Algorithms

KNN Algorithms: In a Nutshell

K-Nearest Neighbor algorithms are some of the simplest Machine Learning algorithms.

KNN algorithms leverage the entire training dataset to make predictions for any new test data point.

In contrast, other learning algorithms seen so far leverage instead the training set to extract a compact hypothesis that can then be used to make new predictions for new test data points.

KNN predictions -- which are based on the idea that “things that look alike must be alike” – are carried out by inspecting the labels of the closest neighbor data points to the test data point.

KNN algorithms can be used both for classification and regression tasks.

K-Nearest Neighbor (KNN) Algorithms

KNN Rule

It is assumed there is a metric $\rho : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ that returns a "distance" between two instances (points) of the instance space $\mathcal{X}$.

For example, the Euclidean distance between two points $\boldsymbol{x}$ and $\boldsymbol{x}'$ in *d*-dimensional Euclidean space $\mathcal{X} = \mathbb{R}^d$ is given by:

$$\rho(\boldsymbol{x}, \boldsymbol{x}') = \|\boldsymbol{x} - \boldsymbol{x}'\|_2^2 = \sqrt{\sum_{k=1}^{d} (x_k - x_k')^2}$$

It is also assumed one has access to a training set $\mathcal{S} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$. Consisting of *m* data point pairs $(\boldsymbol{x}_i, y_i)$ where $\boldsymbol{x}_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$.

A KNN rule $h_{\mathcal{S}}: \mathcal{X} \to \mathcal{Y}$ is given with respect to some function $\varphi : (\mathcal{X} \times \mathcal{Y})^k \to \mathcal{Y}$, as follows:

$$h_{\mathcal{S}}(\boldsymbol{x}) = \varphi((\boldsymbol{x}_{\pi_1(\boldsymbol{x})}, y_{\pi_1(\boldsymbol{x})}), \ldots, (\boldsymbol{x}_{\pi_k(\boldsymbol{x})}, y_{\pi_k(\boldsymbol{x})}))$$

where $\pi_i(\boldsymbol{x})$ be a permutation of $\{1,2,..,m\}$ such that

$$\rho\left(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{\pi_i}(\boldsymbol{x})}\right) \leq \rho\left(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{\pi_{i+1}}(\boldsymbol{x})}\right), \qquad \text{for } i < m$$

KNN rules can be applied both to classification tasks – where the target (label) is discrete – or regression tasks – where the target (label) is continuous.

K-Nearest Neighbor (KNN) Algorithms

KNN for Classification

The k-NN algorithm for binary classification

input: a training sample $S = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$

output: for every point $\boldsymbol{x} \in \mathcal{X}$

return the majority label among $\{y_{\pi_i(\boldsymbol{x})} : i \leq k\}$

For example, for $k = 1$, the 1-NN rule returns $h_S(\boldsymbol{x}) = y_{\pi_1(\boldsymbol{x})}$

K-Nearest Neighbor (KNN) Algorithms

KNN for Regression

The k-NN algorithm for regression

input: a training sample $S = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$

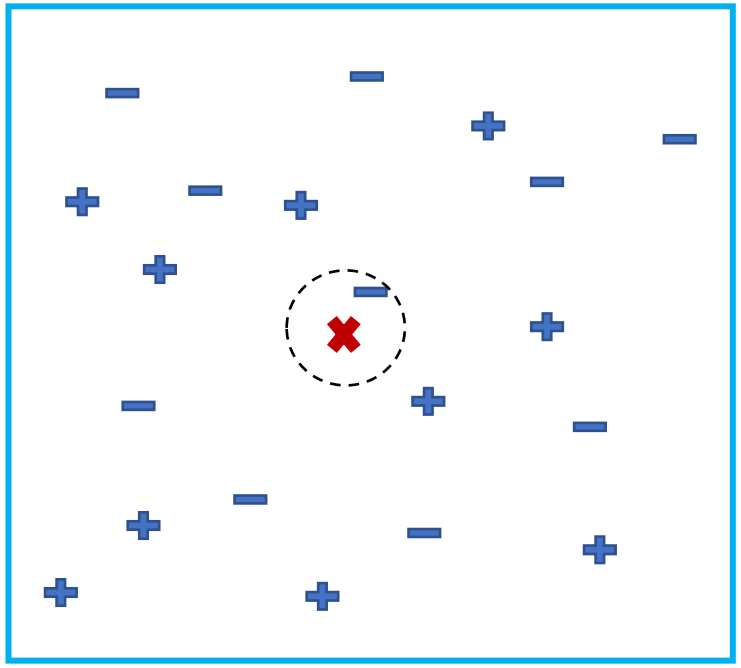
output: for every point $\boldsymbol{x} \in \mathcal{X}$

return average of the NN labels $h_S(\boldsymbol{x}) = {}^1/_k \sum_{i=1}^{k} y_{\pi_i(\boldsymbol{x})}$

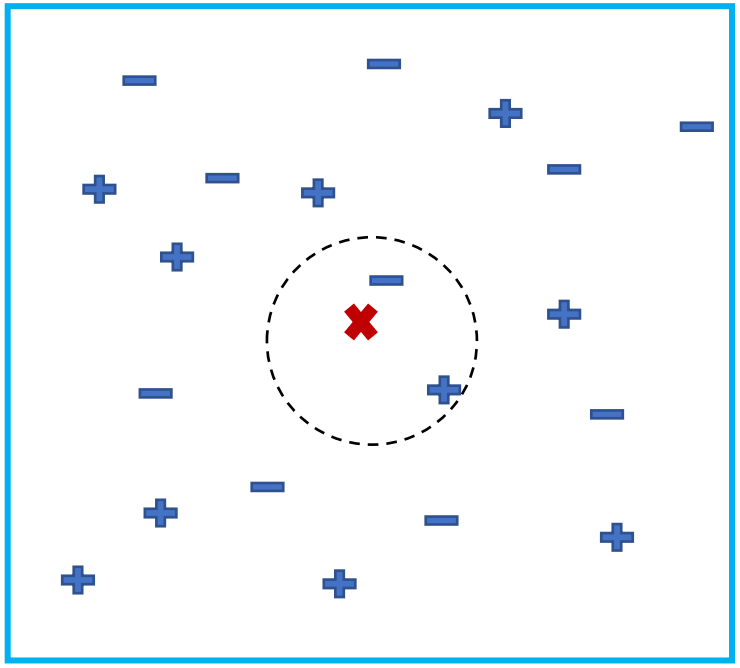
Other possible outputs can be a weighted average of the NN labels.

K-Nearest Neighbor (KNN) Algorithms

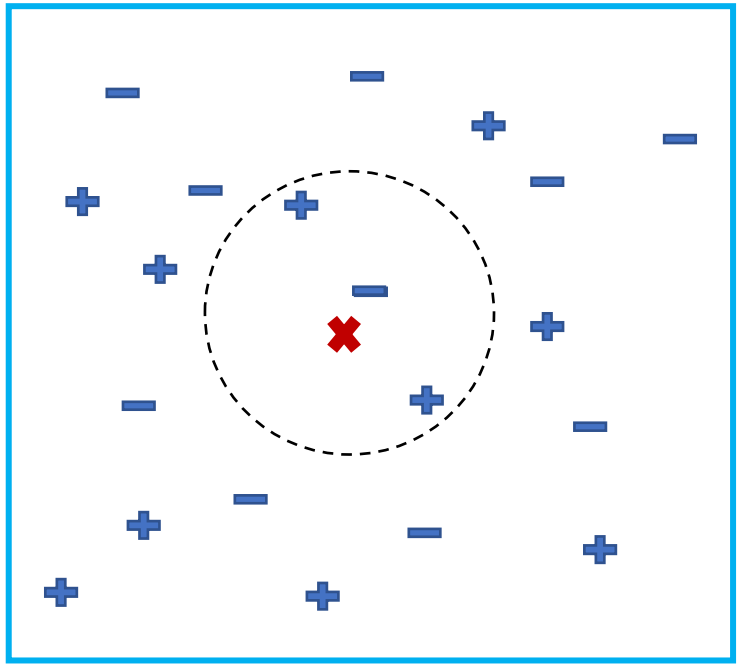
KNN Algorithms



$k = 1$



$k = 2$

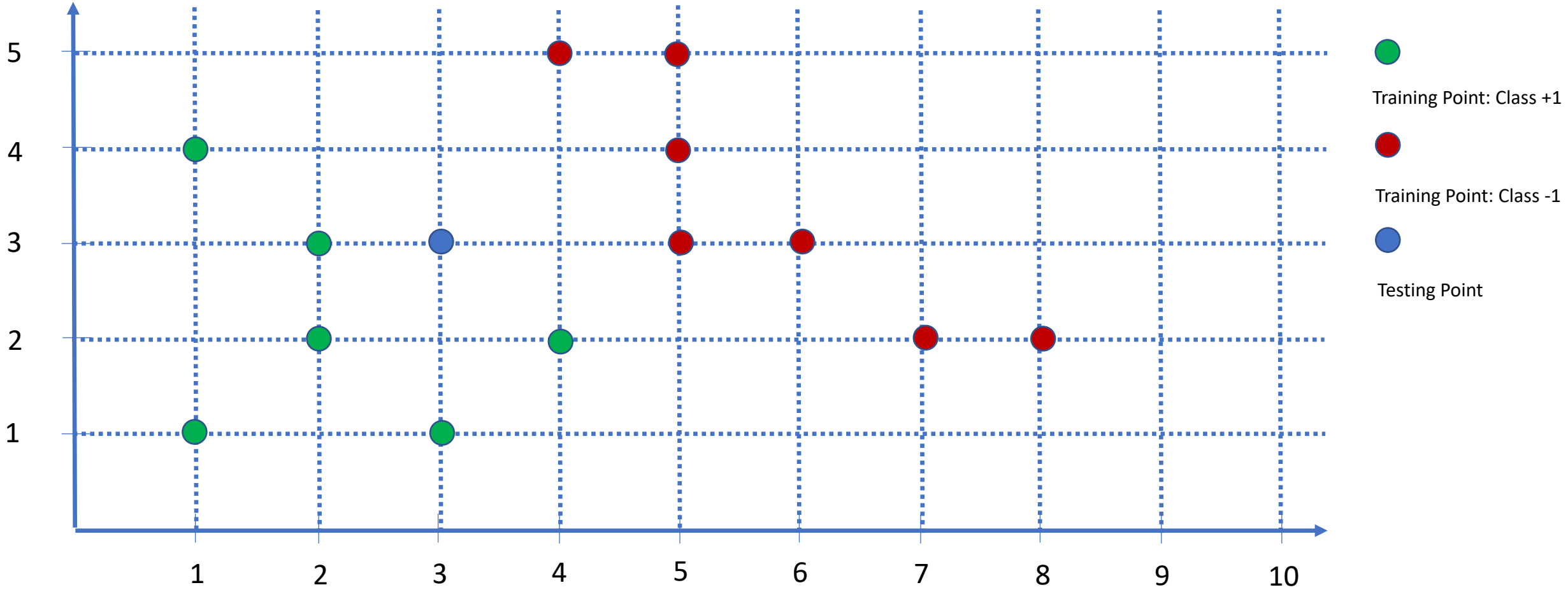


$k = 3$

Adapted from MIT Open Courseware

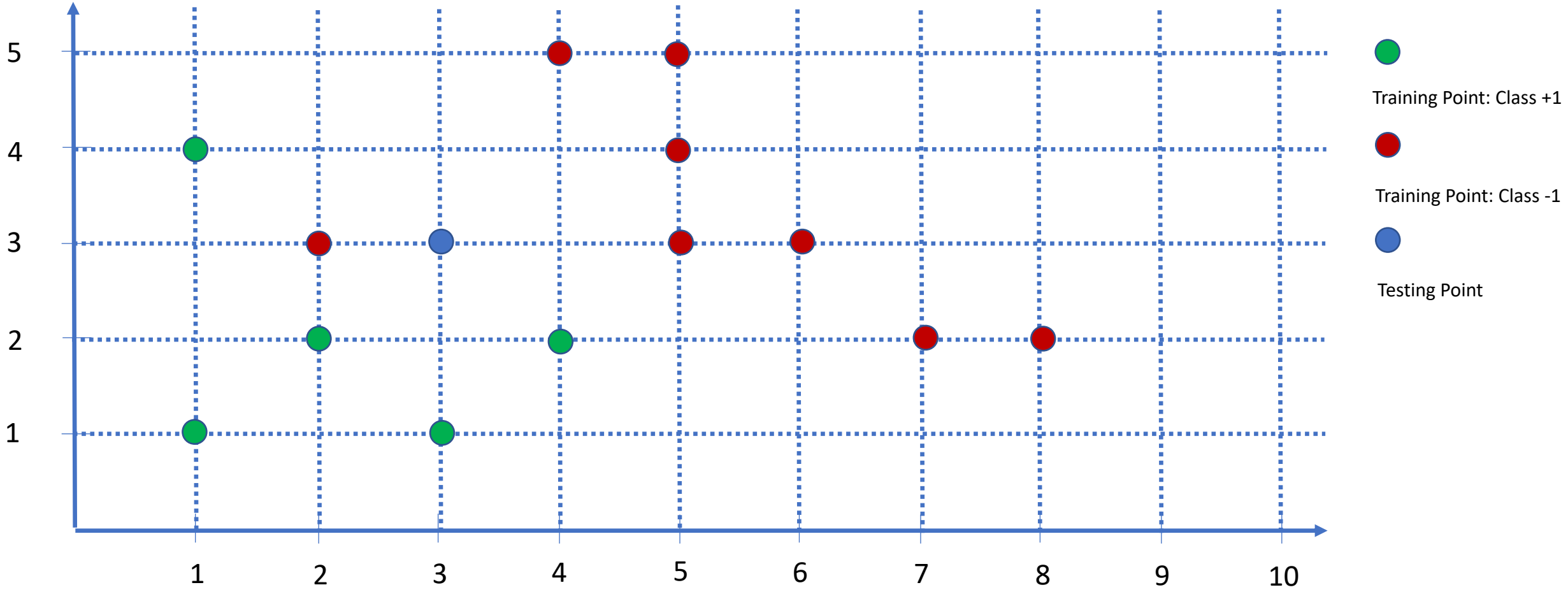
Example 1

How does KNN classify the test data point? What are KNN decision regions?



Example 2

How does KNN classify the test data point? What are KNN decision regions?



K-Nearest Neighbor (KNN) Algorithms

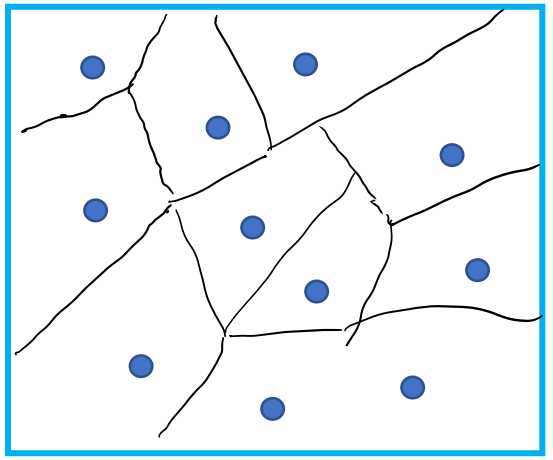
How to choose k?

A small value of k results in a large number of regions but may lead to: (1) high-variance (2) over-fitting (hence sensitive to noise and outliers)

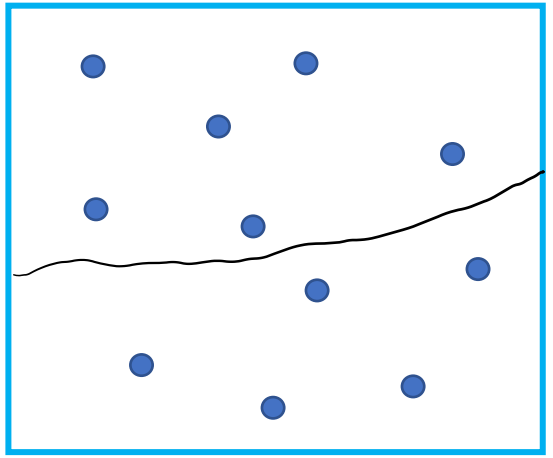
A large value of k results in small number of larger regions but may lead to: (1) high-bias (2) under-fitting (hence less sensitive to noise and outliers)

An optimal value of k can be chosen using as cross validation. Rule of thumb: $k < \sqrt{n}$

$k = 1$



$k = 5$



K-Nearest Neighbor (KNN) Algorithms

Lazy Methods

Lazy methods – such as KNN – do not extract a compact model from the training dataset. Lazy methods instead process the entire training dataset to label each individual test data point every time.

Lazy methods can therefore be computationally intensive.

Eager Methods

Eager methods on the other hand extract a compact model from the training set that can be used to label each individual test data point *a posteriori*.

Such a compact model can include: (1) distribution parameters (in Bayesian estimators and classifiers) and (2) a graph structure and associated weights (in neural networks regressors and classifiers.)

Eager methods – in contrast to lazy methods – also discard the training data once the compact model has been extracted from the training dataset.

K-Nearest Neighbor (KNN) Algorithms

NN Performance

It can be shown that – in binary classification problems with 0-1 loss under a Euclidean metric such that $\mathcal{X} \in [0,1]^d$ and $\mathcal{Y} \in \{0,1\}$ – the 1-NN rule converges to twice the (optimal) Bayes error as the sample size *m* approaches infinity.

It can also be shown that – in similar settings – the *k*-NN rule converges to $1 + \sqrt{8}/_k$ times the (optimal) Bayes error as the sample size *m* approaches infinity.

However, it is also required that the sample size scales exponentially with the ambient dimension – this is known as curse of dimensionality.

This curse of dimensionality is typically alleviated by performing in practice NN classification/regression in conjunction with dimensionality reduction.

K-Nearest Neighbor (KNN) Algorithms

Characteristics: Summary

Advantages:

It incorporates local information (naturally forming complex decision boundaries)

It exhibits simple implementation, lending itself to parallel computation

It is asymptotically consistent

Disadvantages:

It does not incorporate global information

It can be computationally complex, exhibiting large memory requirements