Machine Learning I

Chapter 09 - Instance-based Method: KNN Classification

Prof. Dr. Sandra Eisenreich

December 21 2023

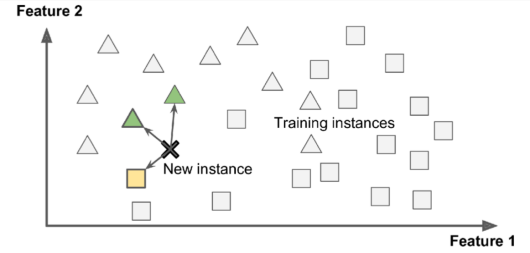
Hochschule Landshut

Overview

- task: **classification or regression**
- data: labeles, i.e. **supervised** learning; Note: should not be too many and not too high-dimensional
- model: **instance-based**
- complexity: simple model, but high memory demand and computational complexity grows fast with number of training instances.
- focus: **explainable**

When to use: not too many training instances, not many features (e.g. ¡10).

K Nearest Neighbors



K nearest neighbor (KNN) models work as follows: Keep all training data in memory and compare a new instance to the K closest instances (the K nearest neighbors). The predict the following value:

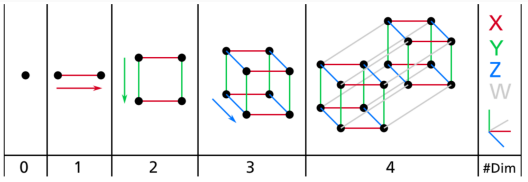
- KNN classification: probability vector for all classes given by:

$$p_c = \frac{\text{number of class c instances among the K nearest neighbors}}{K}.$$

- KNN regression: $\hat{y}$= mean of the labels of the K nearest neighbors.

Curse of dimensionality

In high dimensional spaces distances between nearest and farthest points from query points become almost equal. Therefore, nearest neighbor calculations cannot discriminate candidate points; KNN doesn't work well in high dimensions, i.e. with many features.



KNN Classification with Scikit-Learn

```
from sklearn.neighbors import KNeighborsClassifier


knn_clf = KNeighborsClassifier()
knn_clf.fit(X_train, y_train)
```

Then `kneighbors(X_new, n_neighbors=k)` returns the distances and the indices of the k nearest neighbors in the training set, (two matrices with k columns).

For regression, use: `KNeighborsRegressor`.