Introduction to Machine Learning

Chapter 3: K-Nearest Neighbors

Bernd Bischl, Christoph Molnar, Daniel Schalk, Fabian Scheipl

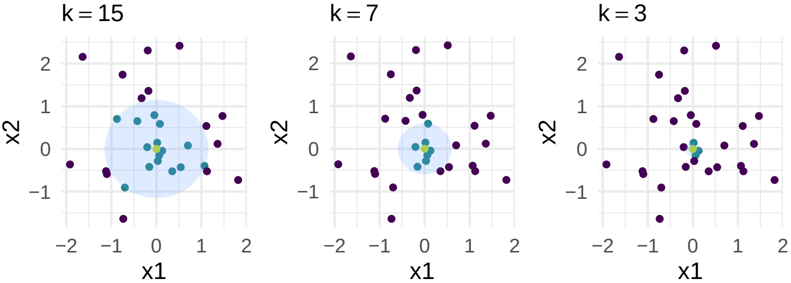
Department of Statistics – LMU Munich

NEAREST NEIGHBORS: INTUITION

- Say we know locations of cities in 2 different countries.
- Say we know which city is in which country.
- Say we don’t know where the countries’ border is.
- For a given location, we want to figure out which country it belongs to.
- Nearest neighbor rule: every location belongs to the same country as the closest city.
- K-nearest neighbor rule: vote over the *k* closest cities (smoother)

K-NEAREST-NEIGHBORS

- **k-NN** can be used for regression and classification
- It generates predictions $\hat{y}$ for a given x by comparing the k observations that are closest to x
- "Closeness" requires a distance or similarity measure (usually: Euclidean).
- The set containing the k closest points $x^{(i)}$ to x in the training sample is called the **k-neighborhood** $N_k(x)$ of x.



K-NEAREST-NEIGHBORS

How to calculate distances?

- Most popular distance measure for numerical features: **Euclidean distance**
- Imagine two data points $x = (x_1, ..., x_p)$ and $\tilde{x} = (\tilde{x}_1, ..., \tilde{x}_p)$ with p features $\in \mathbb{R}$
- The Euclidean distance:

$$d_{Euclidean}(x, \tilde{x}) = \sqrt{\sum_{j=1}^{p} (x_j - \tilde{x}_j)^2}$$

K-NEAREST-NEIGHBORS

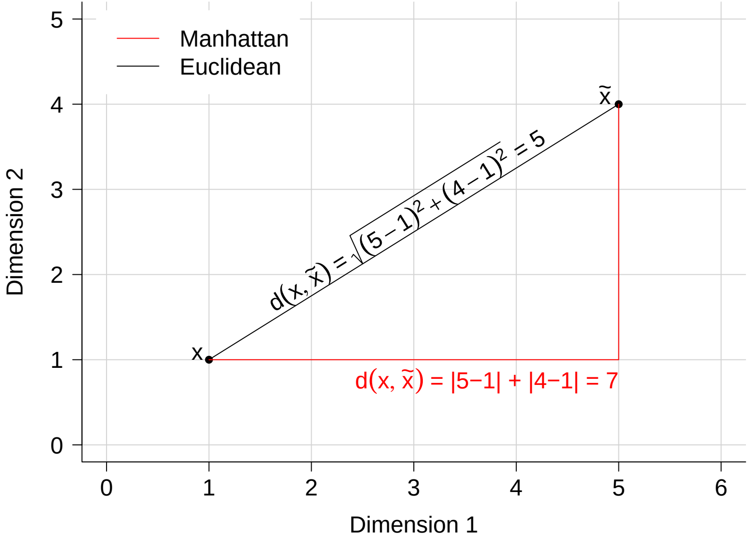
- Example:
  - Three data points with two metric features each: $a = (1, 3)$, $b = (4, 5)$ and $c = (7, 8)$
  - Which is the nearest neighbor of b in terms of the Euclidean distance?
  - $d(b, a) = \sqrt{(4-1)^2 + (5-3)^2} = 3.61$
  - $d(b, c) = \sqrt{(4-7)^2 + (5-8)^2} = 4.24$
  - $\Rightarrow$ a is the nearest neighbor for b.
- Alternative distance measures are:
  - Manhattan distance

$$d_{manhattan}(x, \tilde{x}) = \sum_{j=1}^{p} |x_j - \tilde{x}_j|$$

  - Mahalanobis distance (takes covariances in $\mathcal{X}$ into account)

K-NEAREST-NEIGHBORS

Comparison between Euclidean and Manhattan distance measures



K-NEAREST-NEIGHBORS

Categorical variables, missing data and mixed space:

The Gower distance $d_{gower}(x, \tilde{x})$ is a weighted mean of $d_{gower}(x_j, \tilde{x}_j)$:

$$d_{gower}(x, \tilde{x}) = \frac{\sum_{j=1}^{p} \delta_{x_j, \tilde{x}_j} \cdot d_{gower}(x_j, \tilde{x}_j)}{\sum_{j=1}^{p} \delta_{x_j, \tilde{x}_j}}.$$

- $\delta_{x_j, \tilde{x}_j}$ is 0 or 1. It becomes 0 when the j-th variable is ***missing*** in at least one of the observations (x or $\tilde{x}$), or when the variable is asymmetric binary (where “1” is more important/distinctive than “0”, e. g. “1” means “color-blind”) and both values are zero. Otherwise it is 1.
- $d_{gower}(x_j, \tilde{x}_j)$, the j-th variable contribution to the total distance, is a distance between the values of x_j and $\tilde{x}_j$. For nominal variables the distance is 0 if both values are equal and 1 otherwise. The contribution of other variables is the absolute difference of both values, divided by the total range of that variable.

K-NEAREST-NEIGHBORS

Example of Gower Distance with data on sex and income:

index	sex	salary
1	m	2340
2	w	2100
3	NA	2680

$$\boxed{d_{gower}(x, \tilde{x}) = \frac{\sum_{j=1}^{p} \delta_{x_j,\tilde{x}_j} \cdot d_{gower}(x_j, \tilde{x}_j)}{\sum_{j=1}^{p} \delta_{x_j,\tilde{x}_j}}}$$

$$d_{gower}(x^{(1)}, x^{(2)}) = \frac{1 \cdot 1 + 1 \cdot \frac{|2340-2100|}{|2680-2100|}}{1+1} = \frac{1+\frac{240}{580}}{2} = \frac{1+0.414}{2} = 0.707$$

$$d_{gower}(x^{(1)}, x^{(3)}) = \frac{0 \cdot 1 + 1 \cdot \frac{|2340-2680|}{|2680-2100|}}{0+1} = \frac{0+\frac{340}{580}}{1} = \frac{0+0.586}{1} = 0.586$$

$$d_{gower}(x^{(2)}, x^{(3)}) = \frac{0 \cdot 1 + 1 \cdot \frac{|2100-2680|}{|2680-2100|}}{0+1} = \frac{0+\frac{580}{580}}{1} = \frac{0+1.000}{1} = 1$$

K-NEAREST-NEIGHBORS

Weights:

Weights can be used to address two problems in distance calculation:

- **Standardization:** Two features may have values with a different scale. Many distance formulas (not Gower) would place a higher importance on a feature with higher values leading to an imbalance. Assigning a higher weight for the lower valued feature can combat this effect.
- **Importance:** Sometimes one feature has a higher importance (e. g. more recent measurement). Assigning weights according to the importance of the feature can align the distance measure with known feature importance.

For example:

$$d_{Euclidean}^{weighted}(x, \tilde{x}) = \sqrt{\sum_{j=1}^{p} w_j (x_j - \tilde{x}_j)^2}$$

K-NEAREST-NEIGHBORS

Predictions:

- For regression:

$$\hat{y} = \frac{1}{k} \sum_{i: x^{(i)} \in N_k(x)} y^{(i)}$$

- For classification in g groups, a majority vote is used:

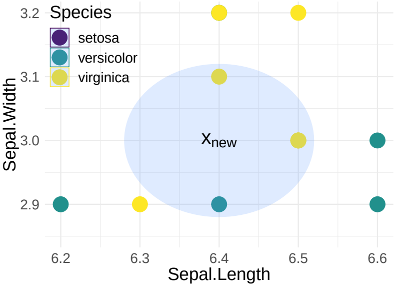
$$\hat{y} = \underset{\ell \in \{1,..,g\}}{\arg\max} \sum_{i: x^{(i)} \in N_k(x)} \mathbb{I}(y^{(i)} = \ell)$$

And posterior probabilities can be estimated with:

$$\hat{\pi}_\ell(x) = \frac{1}{k} \sum_{i: x^{(i)} \in N_k(x)} \mathbb{I}(y^{(i)} = \ell)$$

K-NEAREST-NEIGHBORS

Example with iris data excerpt (k = 3):

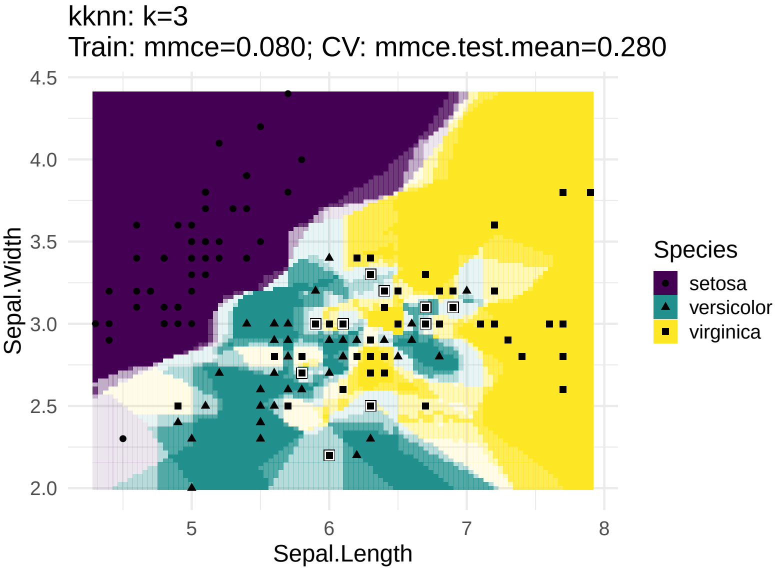


Sepal.Length	Sepal.Width	Species
6.4	3.2	versicolor
6.6	2.9	versicolor
6.4	2.9	versicolor
6.6	3.0	versicolor
6.2	2.9	versicolor
6.3	2.9	virginica
6.5	3.0	virginica
6.5	3.2	virginica
6.4	3.2	virginica
6.4	3.1	virginica

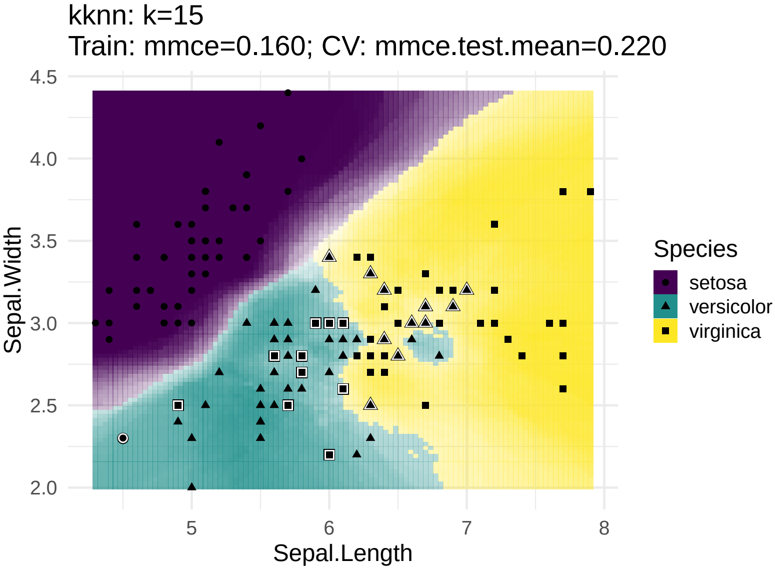
$$\hat{\pi}_{setosa}(x_{new}) = \frac{0}{3} = 0\%$$
$$\hat{\pi}_{versicolor}(x_{new}) = \frac{1}{3} = 33\%$$
$$\hat{\pi}_{virginica}(x_{new}) = \frac{2}{3} = 67\%$$

Prediction: highest posterior probability/majority vote: *virginica*

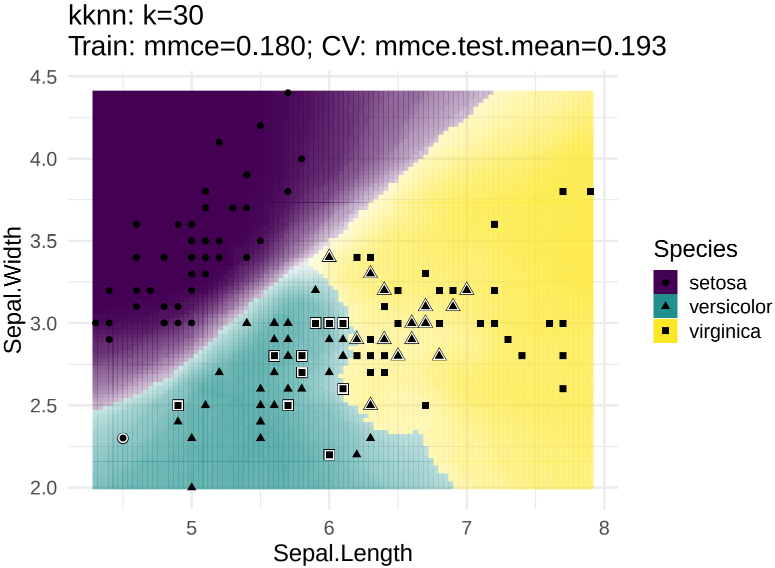
K-NEAREST-NEIGHBORS



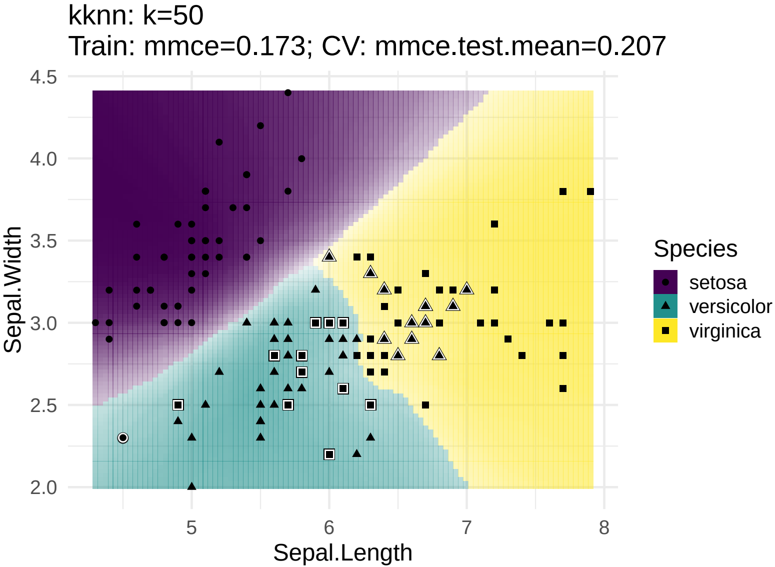
K-NEAREST-NEIGHBORS



K-NEAREST-NEIGHBORS



K-NEAREST-NEIGHBORS



K-NEAREST-NEIGHBORS

- k-NN has no training-step and is a very local model.
- We cannot simply use least-squares loss on the training data for picking k, because we would always pick $k = 1$.
- k-NN makes no assumptions about the underlying data distribution.
- The smaller k, the less stable, less smooth and more “wiggly” the decision boundary becomes.
- Accuracy of k-NN can be severely degraded by the presence of noisy or irrelevant features, or if the feature scales are not consistent with their importance.
- In binary classification, we might choose an odd k to avoid ties.
- For $\hat{y}$, we might inversely weigh neighbors with their distance to x, e.g., $w_i = 1/d(x^{(i)}, x)$

K-NEAREST-NEIGHBORS

Representation: Training data $\mathcal{D}$.
Hyperparameters: distance measure $d(\cdot, \cdot)$ on $\mathcal{X}$; size of neighborhod k.

Evaluation: Any loss function for regression or classification.

Optimization: Not applicable/necessary. (But: clever look-up methods & data structures to avoid computing all n distances for generating predictions.)