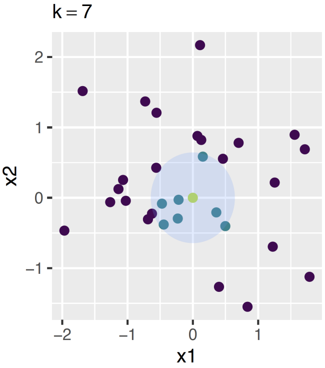
# Introduction to Machine Learning

## *k*-Nearest Neighbors

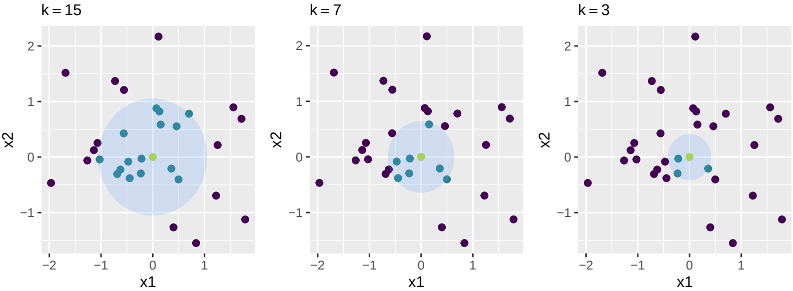


### Learning goals

- Understand the basic idea of k-NN
- Know different distance measures for different scales of feature variables
- Understand that k-NN has no optimization step

# $K$-NEAREST-NEIGHBORS

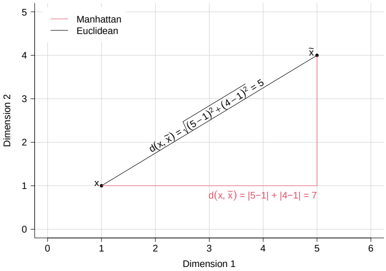
- $k$-**NN** can be used for regression and classification
- It generates predictions $\hat{y}$ for a given $\mathbf{x}$ by comparing the $k$ observations that are closest to $\mathbf{x}$
- "Closeness" requires a distance or similarity measure (usually: Euclidean).
- The set containing the $k$ closest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



# DISTANCE MEASURES

## How to calculate distances?

- Most popular distance measure for numerical features: **Euclidean distance**
- For two data points $\mathbf{x}$ and $\tilde{\mathbf{x}}$ with $p$ features $\in \mathbb{R}$
  - the Euclidean distance is $d_{Euclidean}\left(\mathbf{x}, \tilde{\mathbf{x}}\right) = \sqrt{\sum_{j=1}^{p} (x_j - \tilde{x}_j)^2}.$
  - the Manhattan distance is $d_{manhattan}(\mathbf{x}, \tilde{\mathbf{x}}) = \sum_{j=1}^{p} |x_j - \tilde{x}_j|.$



# PREDICTION

For each point to predict:

- Compute $k$-nearest neighbours in training data $N_k(\mathbf{x})$
- Average output $y$ of these $k$ neighbors
- For regression:

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

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

with neighbors weighted according to their distance to $\mathbf{x}$: $w_i = \frac{1}{d(\mathbf{x}^{(i)}, \mathbf{x})}$

# PREDICTION

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

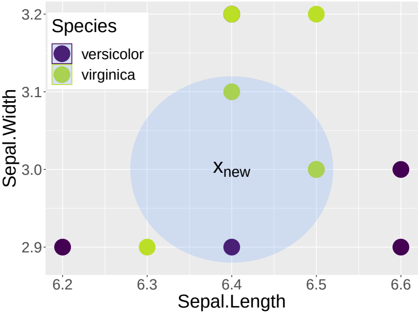
$$\hat{h}(\mathbf{x}) = \underset{\ell \in \{1,...,g\}}{\arg\max} \sum_{i:\mathbf{x}^{(i)} \in N_k(\mathbf{x})} \mathbb{I}(y^{(i)} = \ell)$$

And posterior probabilities can be estimated with:

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

# PREDICTION

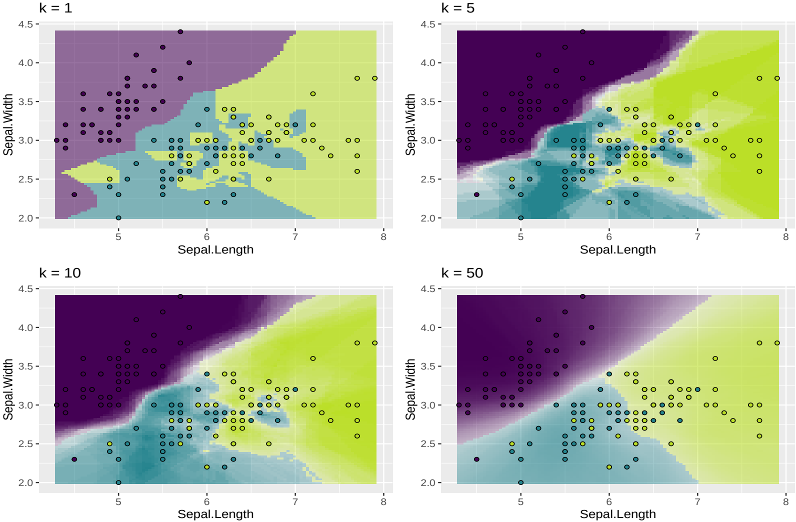
Example with subset of iris data ($k = 3$):



| | SL | SW | Species | dist |
|---|---|---|---|---|
| 52 | 6.4 | 3.2 | versicolor | 0.200 |
| 59 | 6.6 | 2.9 | versicolor | 0.224 |
| 75 | 6.4 | 2.9 | versicolor | 0.100 |
| 76 | 6.6 | 3.0 | versicolor | 0.200 |
| 98 | 6.2 | 2.9 | versicolor | 0.224 |
| 104 | 6.3 | 2.9 | virginica | 0.141 |
| 105 | 6.5 | 3.0 | virginica | 0.100 |
| 111 | 6.5 | 3.2 | virginica | 0.224 |
| 116 | 6.4 | 3.2 | virginica | 0.200 |
| 117 | 6.5 | 3.0 | virginica | 0.100 |
| 138 | 6.4 | 3.1 | virginica | 0.100 |
| 148 | 6.5 | 3.0 | virginica | 0.100 |

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

# *K*-NN: FROM SMALL TO LARGE *K*



Complex, local model vs smoother, more global model

# *K*-NN SUMMARY

- *k*-NN is a lazy classifier, it has no real training step, it simply stores the complete data - which are needed during prediction.
- Hence, its parameters are the training data, there is no real compression of information.
- As the number of parameters grows with the number of training points, we call *k*-NN a non-parametric model
- Hence, *k*-NN is not based on any distributional or strong functional assumption, and can, in theory, model data situations of arbitrary complexity.
- *k*-NN has no optimization 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 $k = 1$. boundary becomes.
- Accuracy of *k*-NN can be severely degraded by the presence of noisy or irrelevant features, or when the feature scales are not consistent with their importance.

# DISTANCE MEASURES - EXTENSIONS

**Categorical variables, missing data and mixed space:**

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

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

- $\delta_{x_j,\tilde{x}_j}$ is 0 or 1. It becomes 0 when the $j$-th variable is ***missing*** in at least one of the observations ($\mathbf{x}$ or $\tilde{\mathbf{x}}$), 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.

# DISTANCE MEASURES - EXTENSIONS

- $d_{gower}(x_j, \tilde{x}_j)$, 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.

# DISTANCE MEASURES - EXTENSIONS

Example of Gower distance with data on sex and income:

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

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

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

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

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

# DISTANCE MEASURES - EXTENSIONS

**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 to 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:

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