Introduction to Machine Learning

Classification: K-Nearest Neighbors

compstat-lmu.github.io/lecture_i2ml

K-NEAREST-NEIGHBORS

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:

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

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

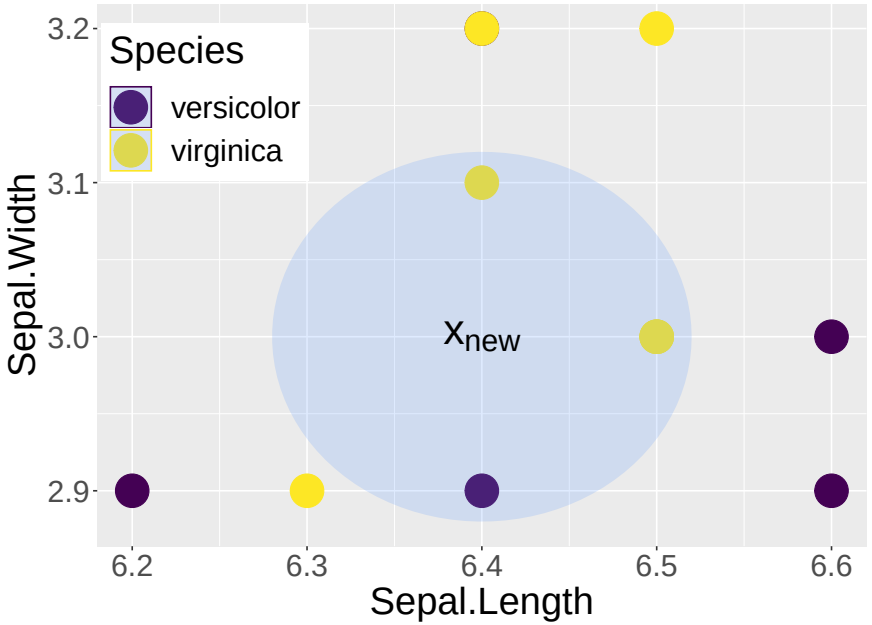
$$\hat{h}(\mathbf{x}) = \underset{\ell \in \{1,\ldots,g\}}{\arg\max} \sum_{i:\mathbf{x}^{(i)} \in N_k(x)} \mathbb{I}(y^{(i)} = \ell)$$

  And posterior probabilities can be estimated with:

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

K-NEAREST-NEIGHBORS

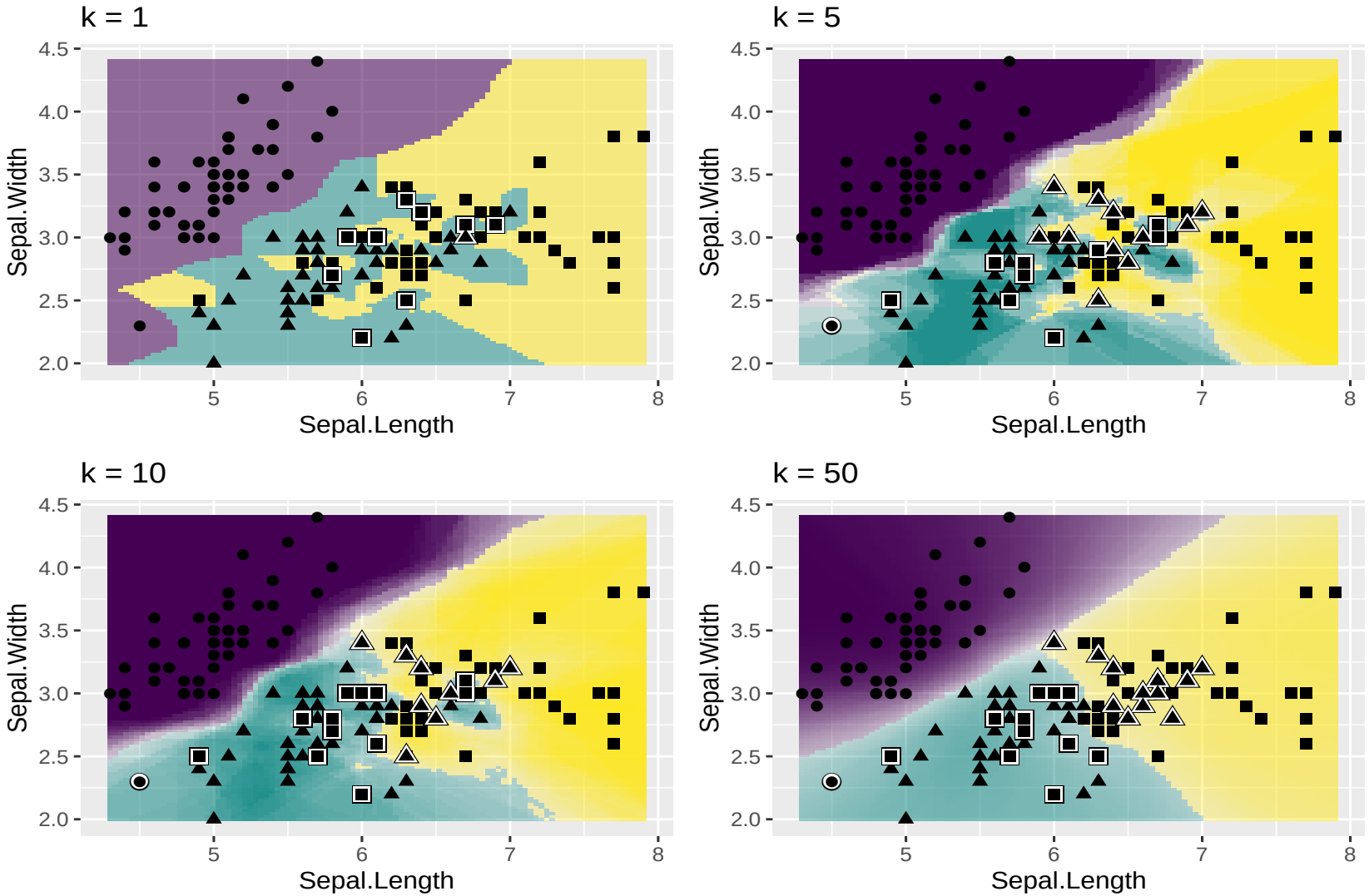
Example with subset of iris data (k = 3):



	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 AS NON-PARAMETRIC MODEL

- 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 number of parameters are growing 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