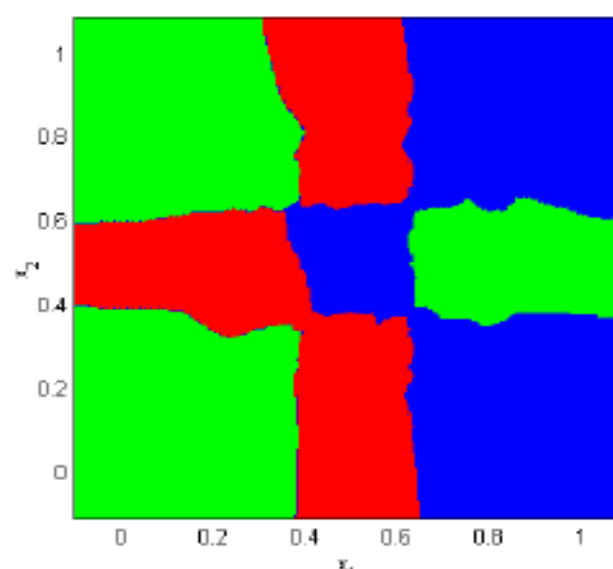
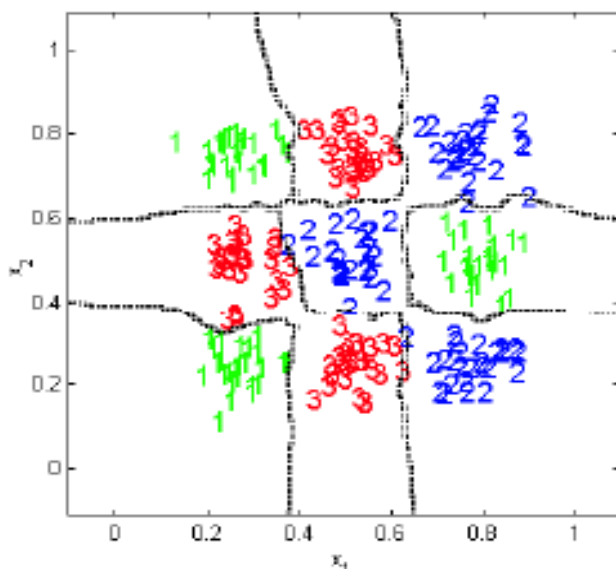
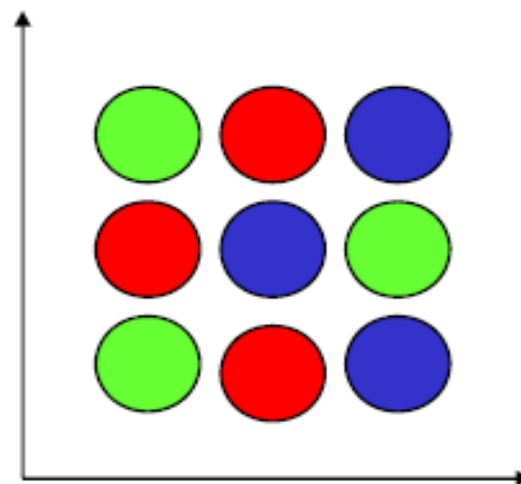


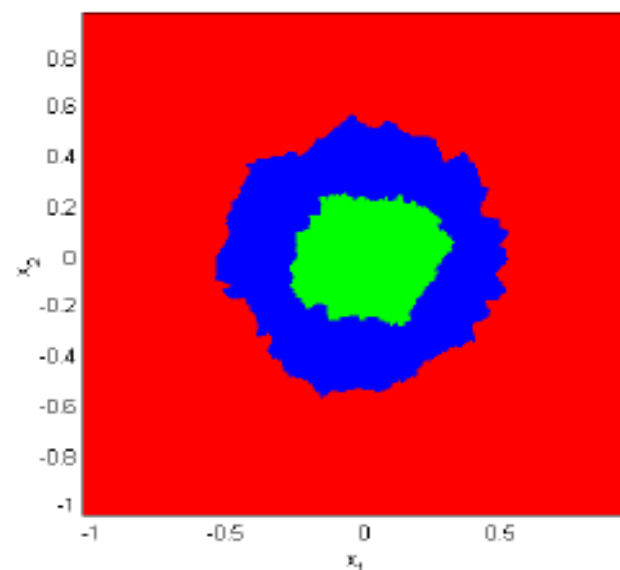
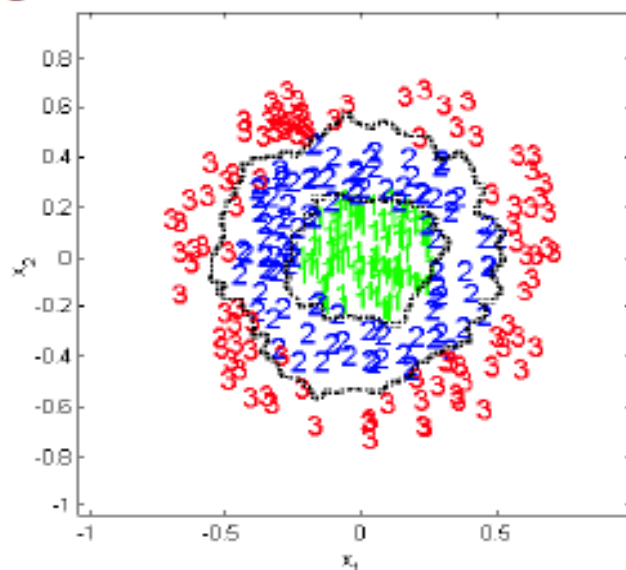
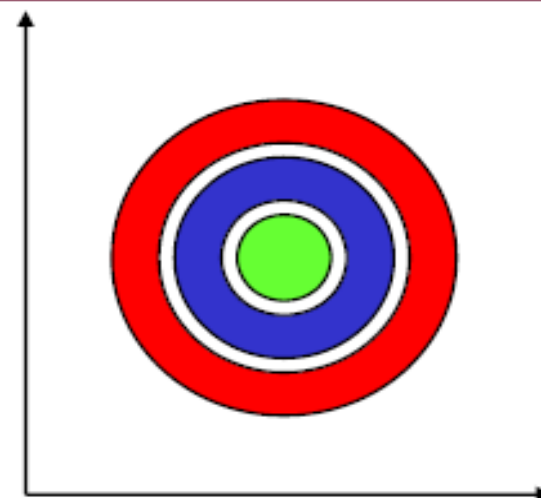
*k*NN in action: example 1

- We have generated data for a 2-dimensional 3-class problem, where the class-conditional densities are multi-modal, and non-linearly separable, as illustrated in the figure
- We used the kNN rule with
 - $k = 5$
 - The Euclidean distance as a metric
- The resulting decision boundaries and decision regions are shown below



*k*NN in action: example 2

- We have generated data for a 2-dimensional 3-class problem, where the class-conditional densities are unimodal, and are distributed in rings around a common mean. These classes are also non-linearly separable, as illustrated in the figure
- We used the kNN rule with
 - $k = 5$
 - The Euclidean distance as a metric
- The resulting decision boundaries and decision regions are shown below



Characteristics of the kNN classifier

■ Advantages

- Analytically tractable
- Simple implementation
- Nearly optimal in the large sample limit ($N \rightarrow \infty$)
 - $P[\text{error}]_{\text{Bayes}} < P[\text{error}]_{1\text{NN}} < 2P[\text{error}]_{\text{Bayes}}$
- Uses local information, which can yield highly adaptive behavior
- Lends itself very easily to parallel implementations

■ Disadvantages

- Large storage requirements
- Computationally intensive recall
- Highly susceptible to the curse of dimensionality

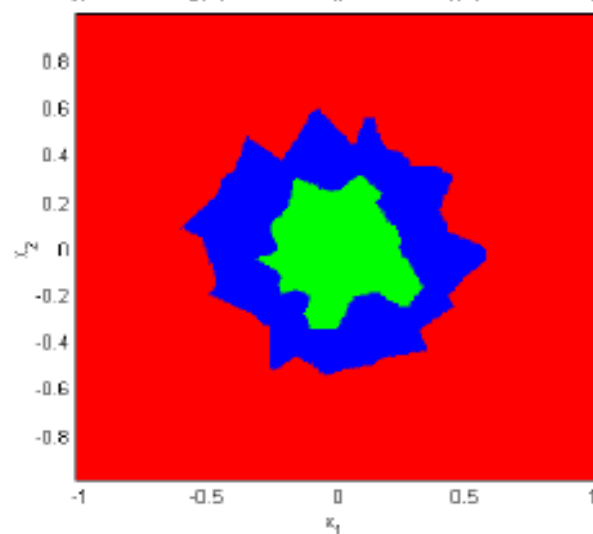
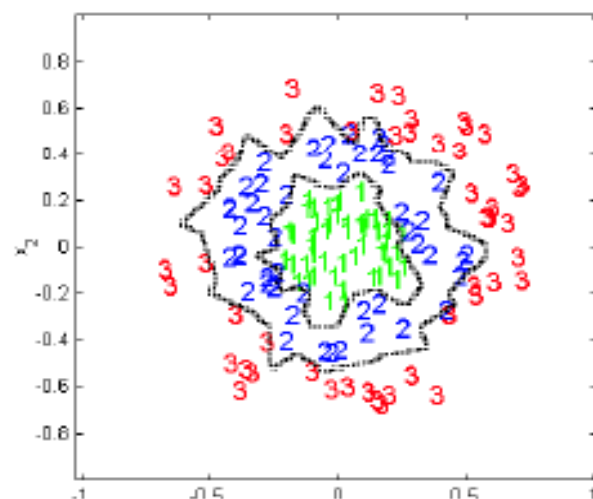
■ 1NN versus kNN

- The use of large values of k has two main advantages
 - Yields smoother decision regions
 - Provides probabilistic information
 - The ratio of examples for each class gives information about the ambiguity of the decision
- However, too large a value of k is detrimental
 - It destroys the locality of the estimation since farther examples are taken into account
 - In addition, it increases the computational burden

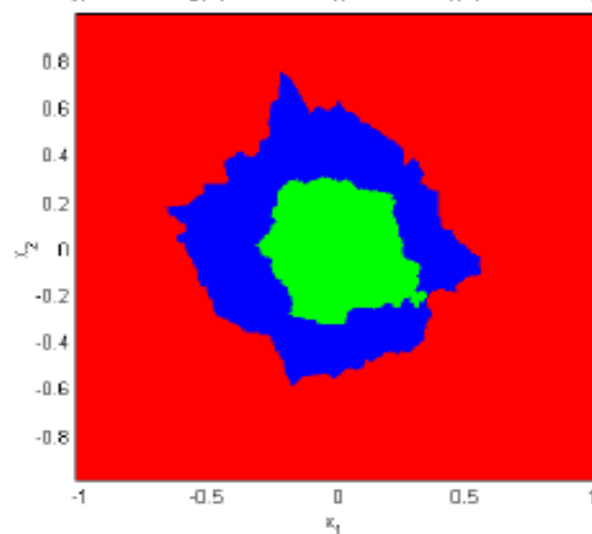
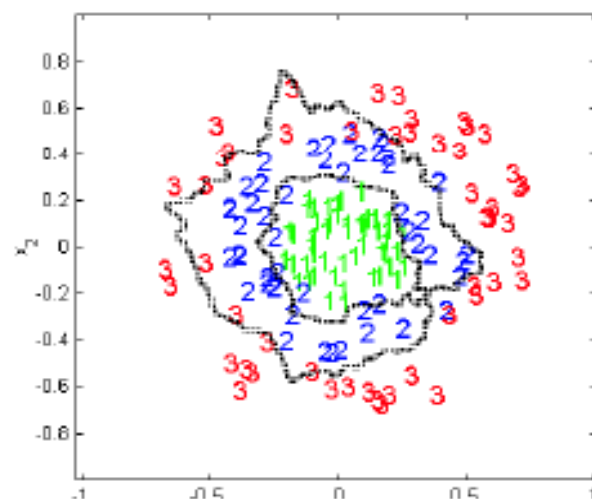


kNN versus 1NN

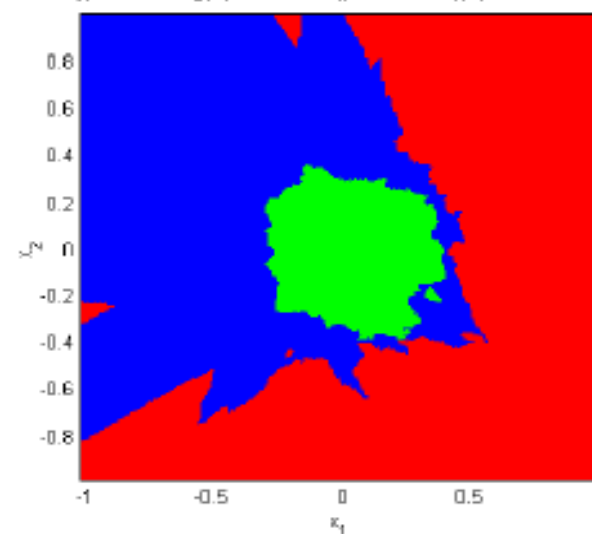
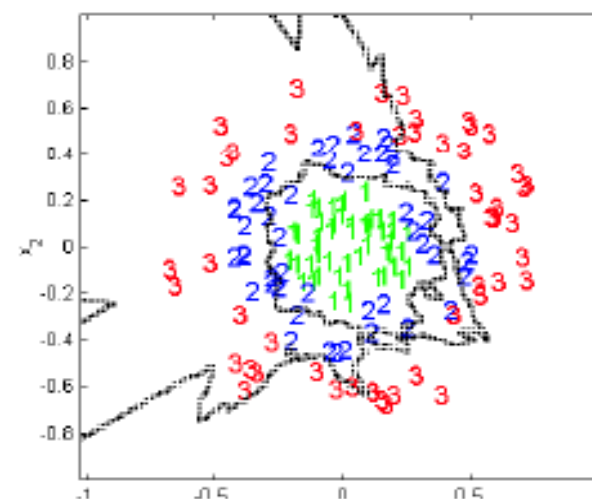
1-NN



5-NN



20-NN



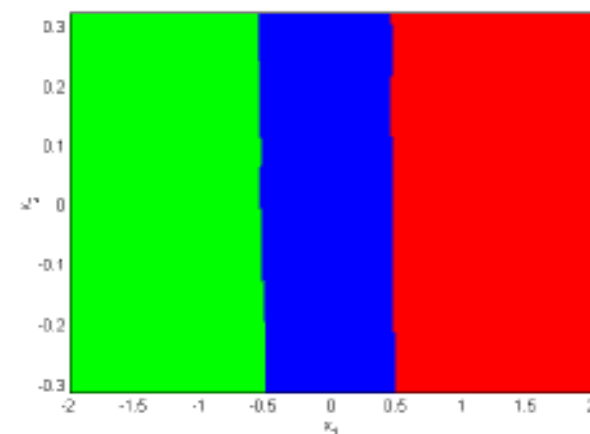
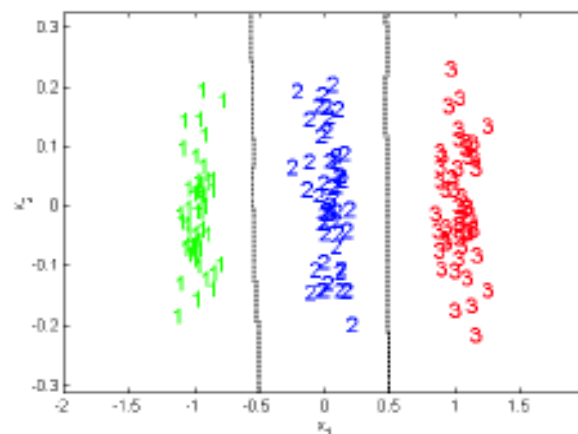
kNN and the problem of feature weighting

- The basic kNN rule computes the similarity measure based on the Euclidean distance

- This metric makes the kNN rule very sensitive to noisy features
- As an example, we have created a data set with three classes and two dimensions
 - The first axis contains all the discriminatory information. In fact, class separability is excellent
 - The second axis is white noise and, thus, does not contain classification information

- In the first example, both axes are scaled properly

- The kNN ($k=5$) finds decision boundaries fairly close to the optimal



- In the second example, the magnitude of the second axis has been increased two order of magnitudes (see axes tick marks)

- The kNN is biased by the large values of the second axis and its performance is very poor

