ICS Summer Academy Session II
Topic 2: Nearest Neighbor Classifiers

Michael Shindler

A very early machine learning example

Recognizing flowers (by R. Fisher, 1936)

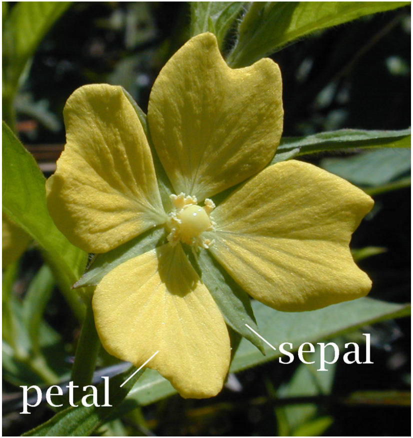
Types of Iris: setosa, versicolor, and virginica



Note: I (Michael) know almost nothing about flowers.
This is somehow not going to be a problem for this example!

Measuring the properties of the flowers

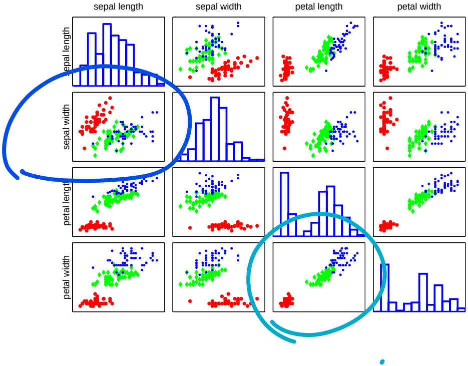
Features and attributes: the widths and lengths of sepal and petal



Pairwise scatter plots of 131 flower specimens

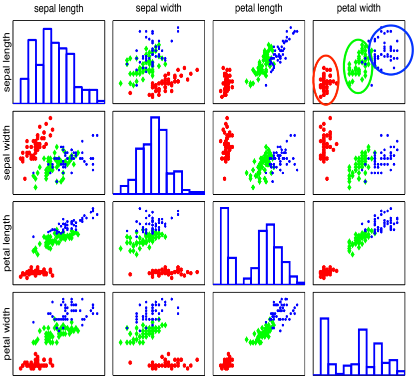
Visualization of data helps identify the right learning model to use

Each colored point is a flower specimen: setosa, versicolor, virginica



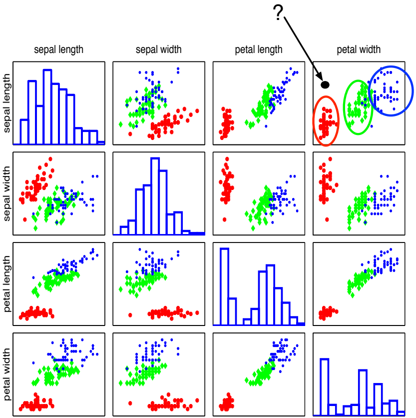
Different types seem well-clustered and separable

Using two features: petal width and sepal length



Labeling an unknown flower type

Closer to red cluster: so labeling it as setosa



What sort of problem is this solving?

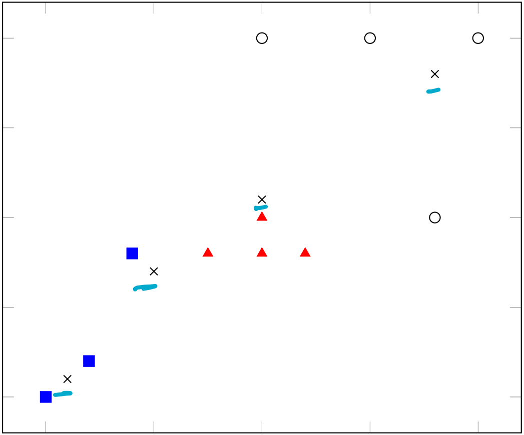
- This is a *classification* problem

- This is a *supervised learning* problem

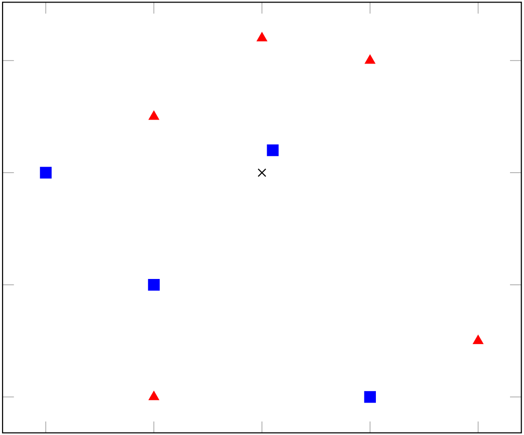
What sort of problem is this solving?

- This is a *classification* problem
- This is a *supervised learning* problem
- We are given *training data*
  - Feature vectors $\boldsymbol{x} \in \mathbb{R}^{\mathsf{D}}$
  - Each has one of C labels
  - Training Data: set of $(\boldsymbol{x}, y \in C)$ pairs
  - Goal: find $f(\cdot)$ such that $y = f(\boldsymbol{x})$
- We evaluate on test data / query points.

How do we classify a query point?



What about this point?



What is the algorithm?

hyper - parameter

Input:

- An integer, k
- A set of training examples, D
- A distance measure function, d

Algorithm:

for each test instance $z = (\boldsymbol{x'}, y')$: **do**
  Compute $d(\boldsymbol{x'}, \boldsymbol{x})$ between z and every example $(\boldsymbol{x}, y) \in D$
  Select $D_z \subseteq D$, the k closest training examples to z.
  $y' \leftarrow \arg\max_v \sum_{(\boldsymbol{x}_i, y_i) \in D_z} I(v = y_i)$
end for

What type of learning is this?

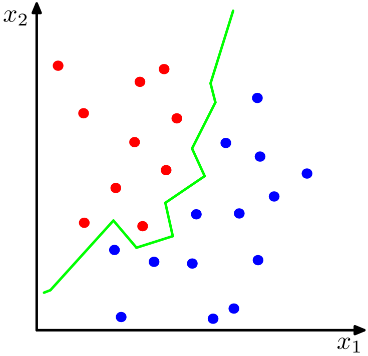
- This is not quite a *rote learner*
- This is a little more flexible than that.
- More of a *lazy learner*.

$$\lim_{x \to 8} \frac{1}{x-8} = \infty$$

$$\lim_{x \to 5} \frac{1}{x-5} = \infty$$

Decision boundary

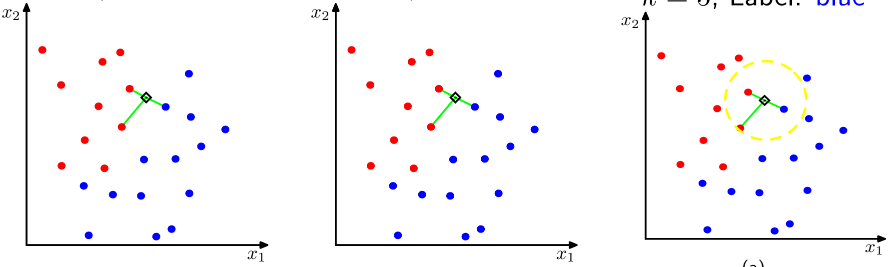
For every point in the space, we can determine its label using the NNC rule. This gives rise to a *decision boundary* that partitions the space into different regions.



(b)

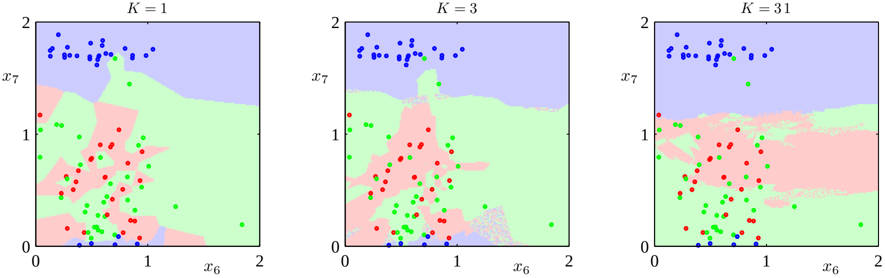
Example

$k = 1$, Label: red $\quad$ $k = 3$, Label: red $\quad$ $k = 5$, Label: blue



(a) (a) (a)

How to choose a good value of k?



When k increases, the decision boundary becomes smooth.

How to measure performance of a classifier?

- Accuracy : % test points correctly classified

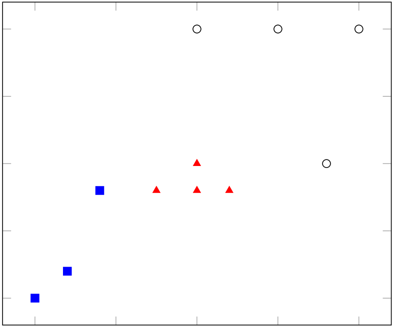
- Error rate: % test points *incorrectly* classified.

Classifier is not just training data!

- Which value of k to use?
- What distance measure?
- What voting system?

You need to tune the hyperparameters!

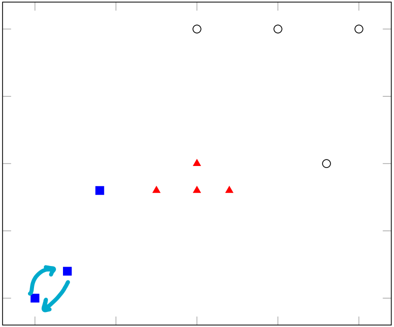
Stupid method: use training directly



Suppose $k = 1$

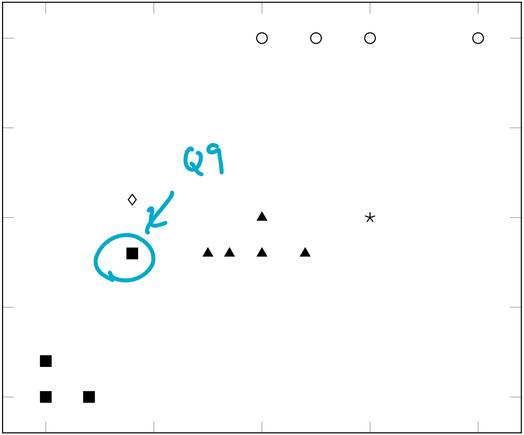
- Accuracy?
- Error rate?
- Would another value of k help?

Better: leave one out

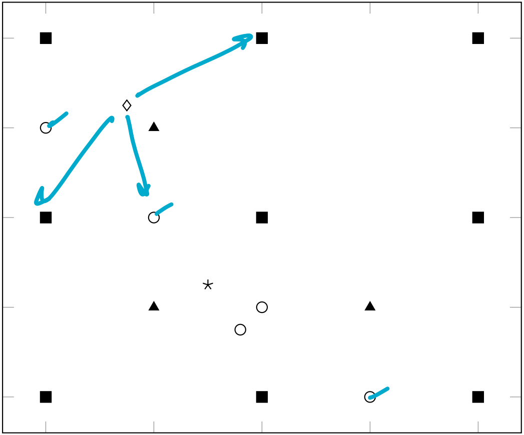


- For each point, treat as query & omit
- For $k = 1$, accuracy / error ?
- Would another value of k help?

Exercise Set 1



Exercise Set 2



Even better: validation dataset

- Training data already provided
- Test data, fully independent of training data
- Now what can I do?

Cross-validation

- We usually aren't given a fully independent dataset
- How can we create one?

Cross-validation

- We usually aren't given a fully independent dataset
- How can we create one?
  - Take, say, $1/5$ of the points.
  - Train on other $4/5$, test on the $1/5$ reserved.
  - Repeat for each fifth.

What is a model-free classifier?

- Much machine learning builds a *model* of the data.

- But classifying a test instance can be quite expensive.
- Uses local info to classify.

also: eager learner

What about missing values?

- Proximity usually needs all attributes.

Garbage in garbage out

What about interacting attributes?

- e.g., two attributes that *together* have more predictive power than alone

What about irrelevant attributes?

- Proximity from irrelevant?
- What about redundant?

What about scale issues?

- e.g., height in feet and weight in pounds?