eurac research

Image Classification Algorithms

Alexander Jacob
Eurac Research

Overview

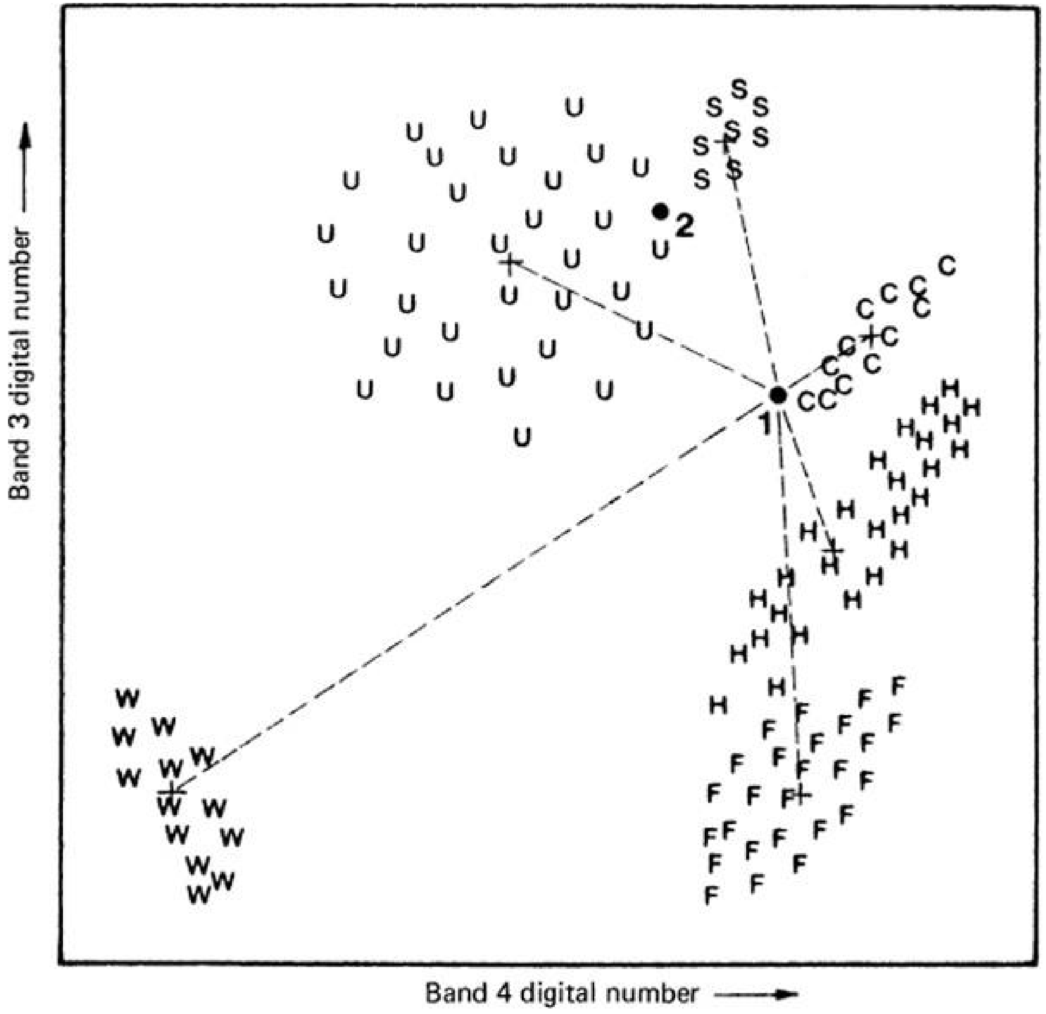
- General Classifier Characteristics
- Classic Classifiers
- Expert systems
- Machine Learning

Classifier Characteristics

- Supervised or Unsupervised
  - Maximum likelihood, ISO-data, K-means
- Parametric or Non-parametric
  - Maximum likelihood, cluster, nearest neighbor
- Hard or Soft (fuzzy) classification set
- Pixel-based or Object-based

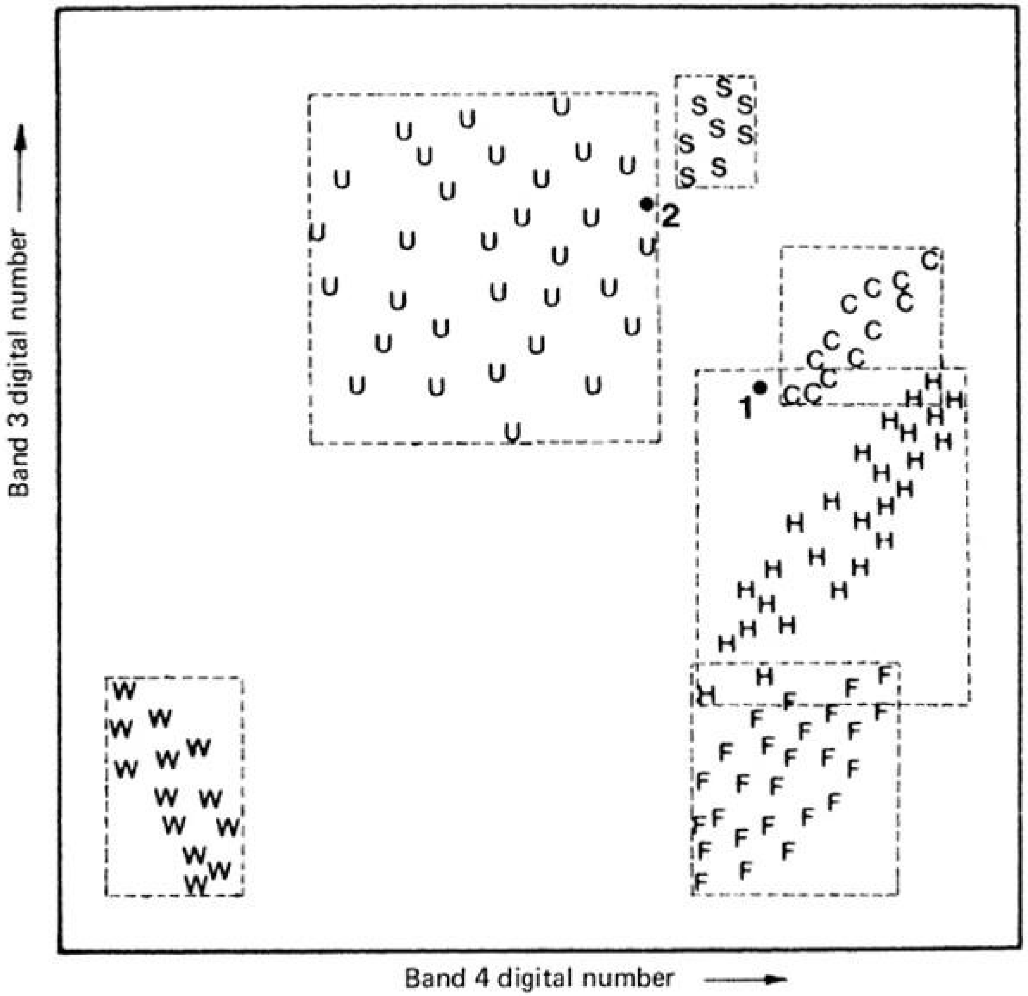
Classic Classifiers

- Each class is represented by its mean value (a vector in multispectral image)
- A new pixel is classified by computing its distances from the classes' means
- The pixels is assigned to the class with minimum distance
- Ignoring the variance: pixel 2 is assigned to class "S" instead of "U"



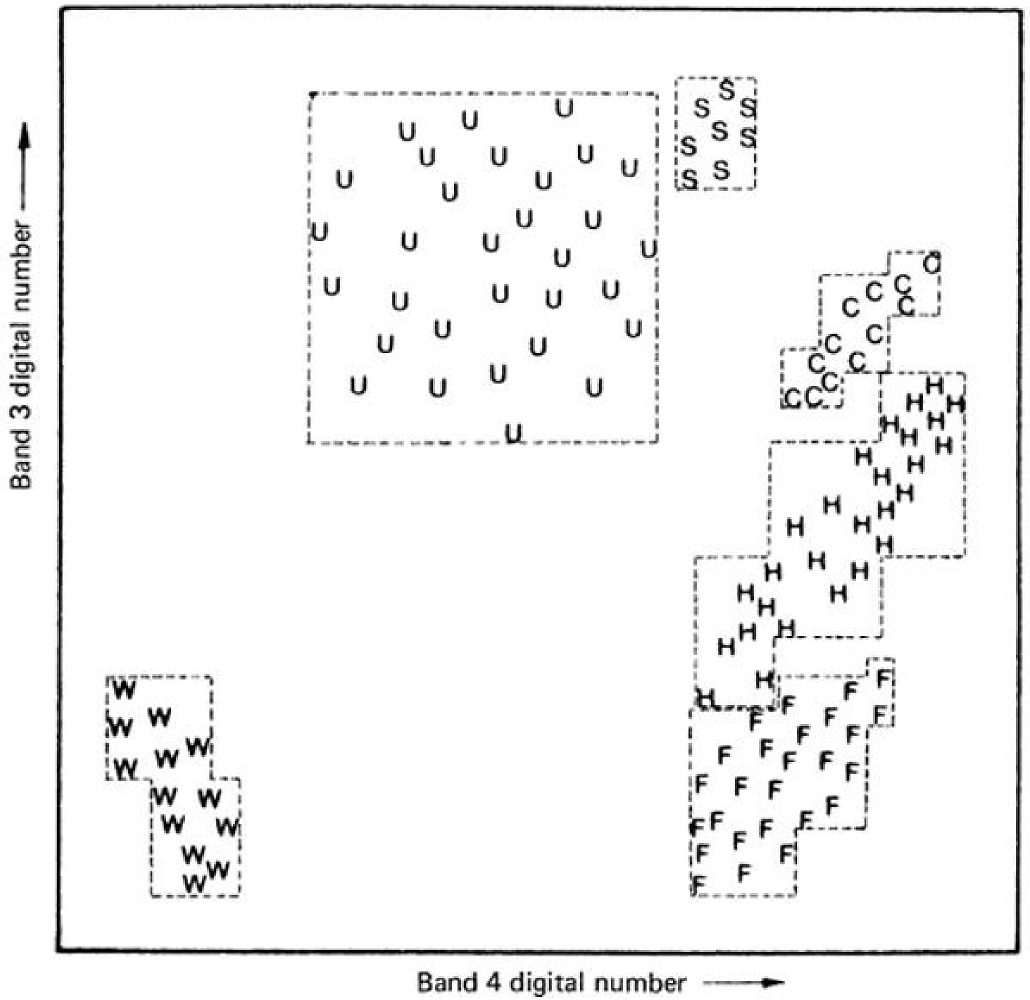
Classic Classifiers

- New pixels (1) and (2) are classified according to how they fit within the min and max values of the training areas -> parallelepiped classifier
- Pixel “1” will be assigned the class “H” and not “C”

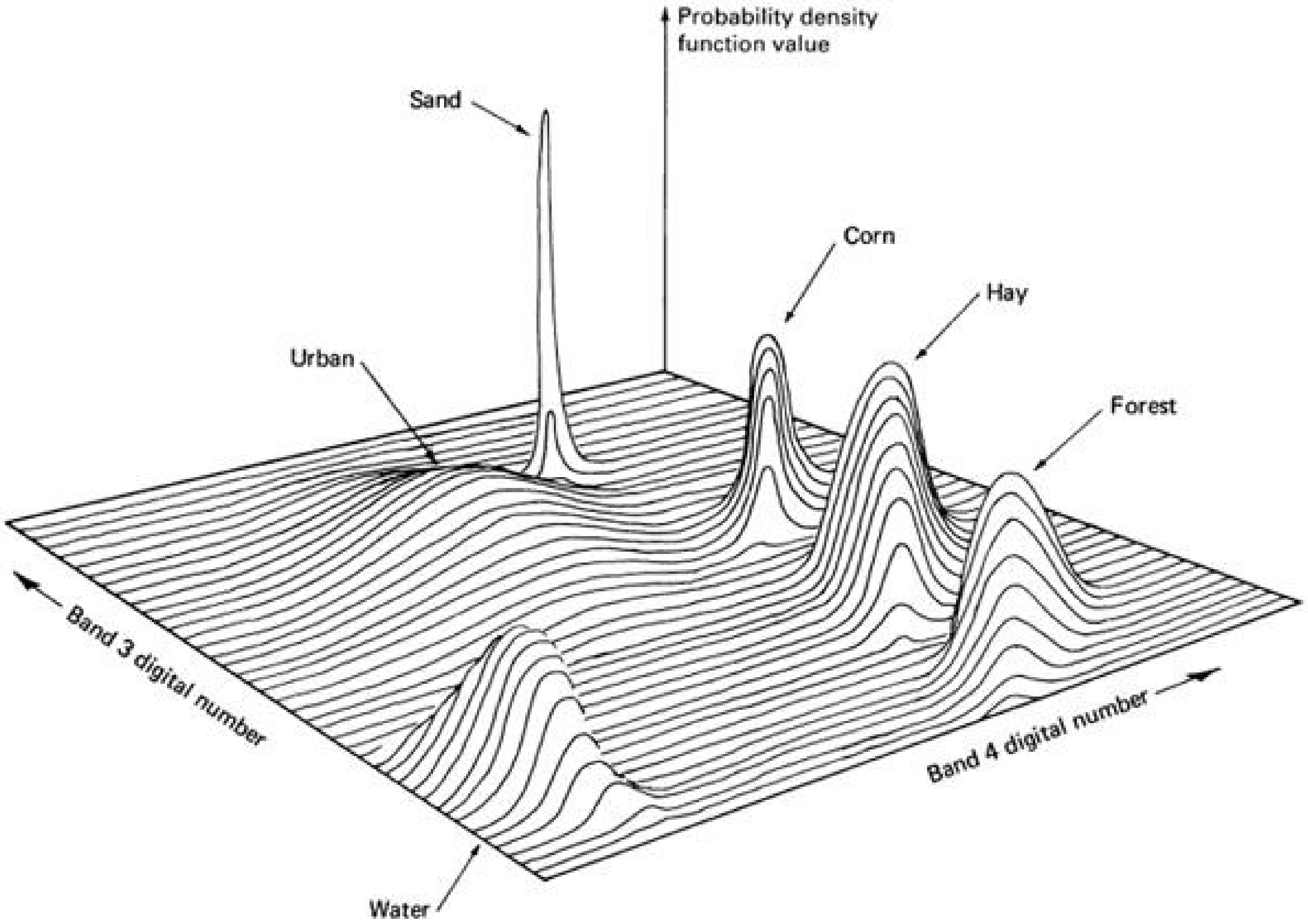


Classic Classifiers

- High covariance is often the rule
- The resulting overlap problems with parallelepiped classification can be somewhat solved by modifying the single rectangle into a series of rectangles for each class.

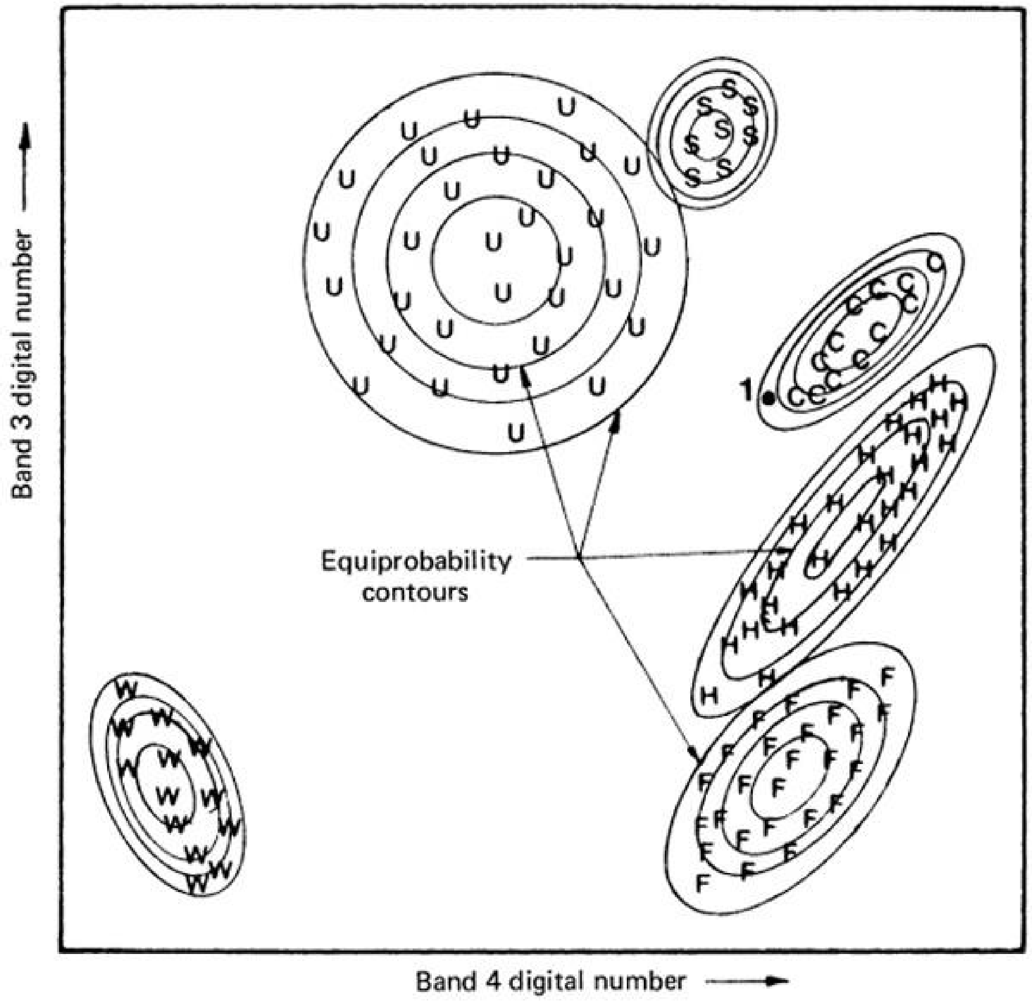


Classic Classifiers



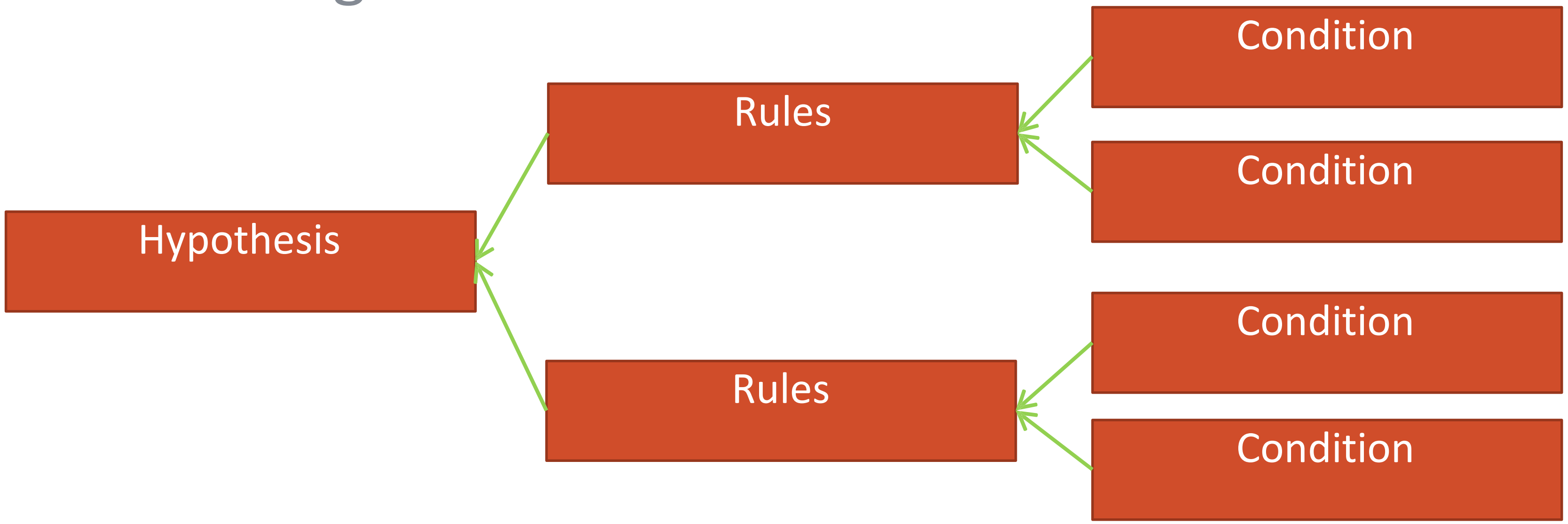
Classic Classifiers

- The probability density functions are used to classify an unidentified pixel by computing the probability of the pixel value belonging to each class, and then assign the pixel to the most likely class.
- Pixel "1" will be assigned to "C"



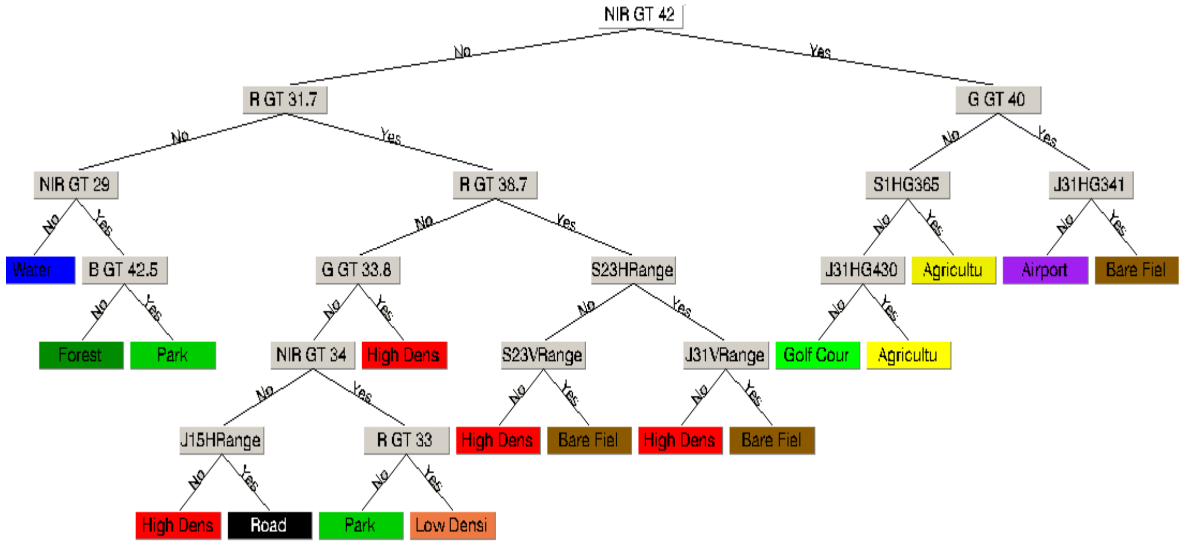
Expert Systems

- Expert knowledge approach
- Hierarchical organized



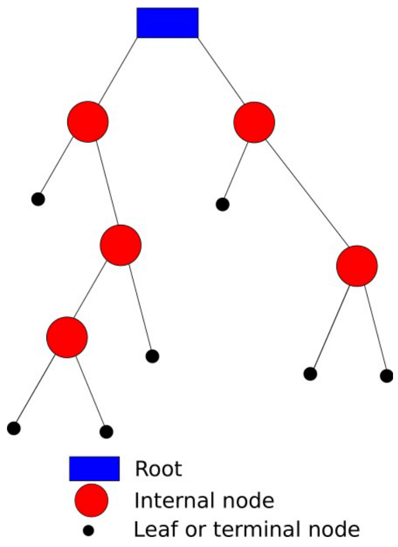
Decision Trees

- Obvious connection between input and outcome
- Nonparametric and non distribution dependent
- Robust with regard to outliers in training data
- Classification is fast once rules are developed
- Does not use all data



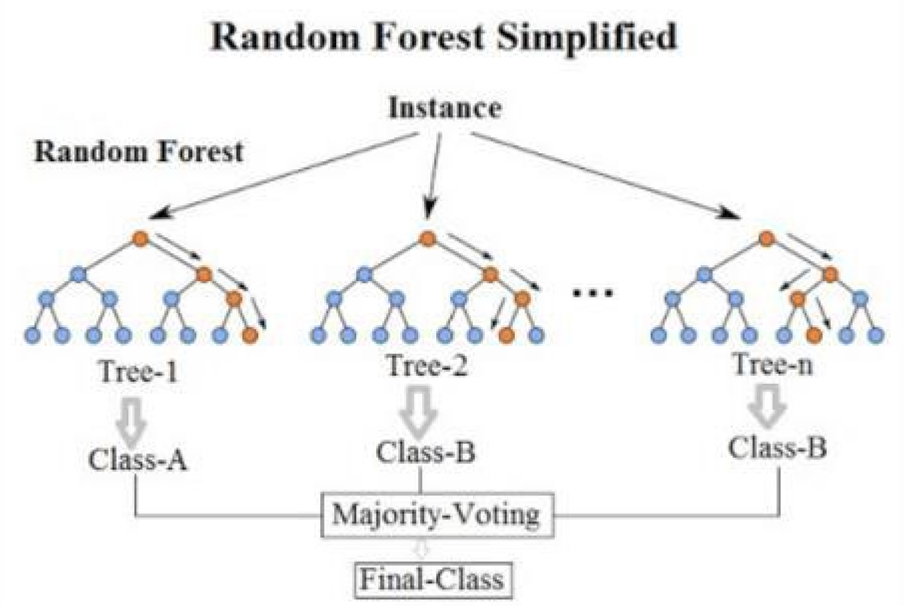
Decision Trees

- Rules can be created by expert or machine learning
- Tree generation:
  - Decide split variables and split value
  - Termination criterion
  - Class assignment of leaf nodes
- C5.0 is a good choice for machine learned rule set.
  - Recursive divide and conquer strategy



Random Forest

- Create a lot of random decision trees based on random subset of variables for splitting each node.
- Use them to predict your data
- Make a vote of all trees
- Main parameters:
  - Number of variables used
  - Number of trees created



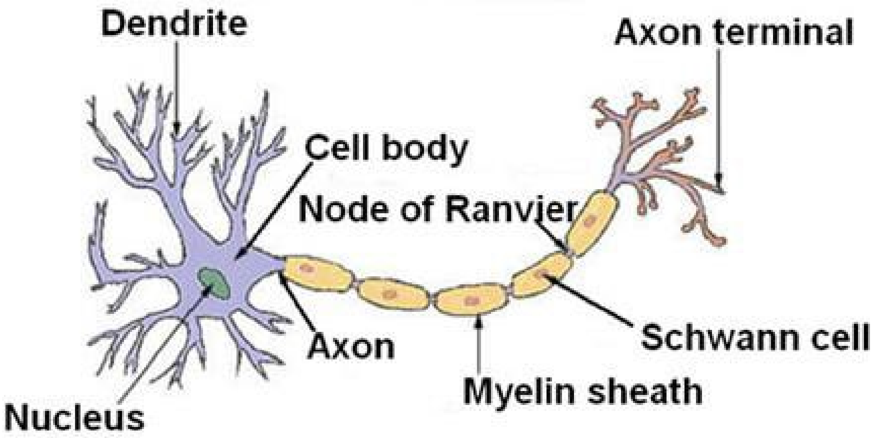
Random Forest

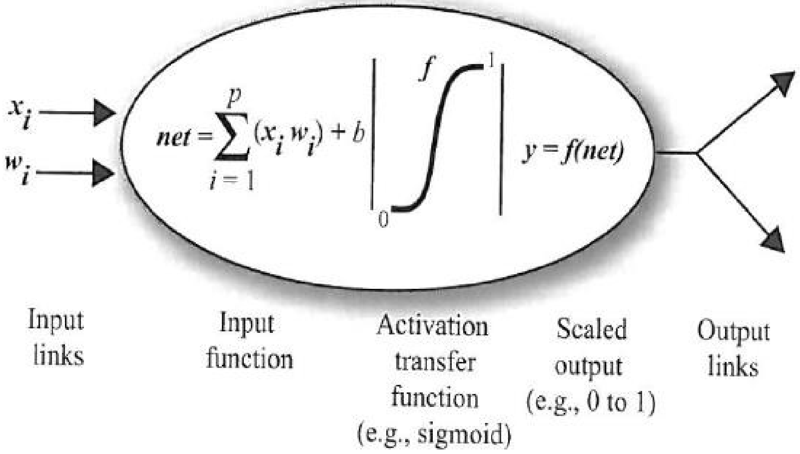
- Multiple decision tree ensemble classifier
- + computationally less demanding
- + data handling
- + estimates accuracy
- + outlier robust
- + parameters relatively easy to select

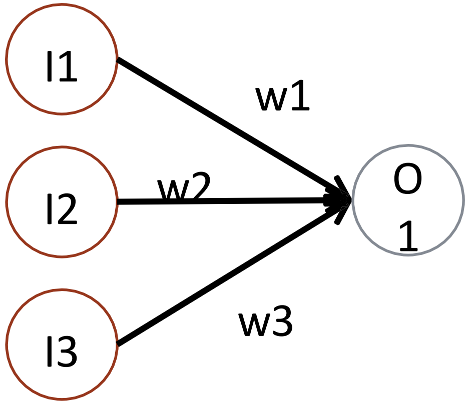
Neural Network

- Origins: Algorithms that try to mimic the brain.
- Was very widely used in 80s and early 90s; popularity diminished in late 90s.
- Recent resurgence: State-of-the-art technique for many applications

Neural Network



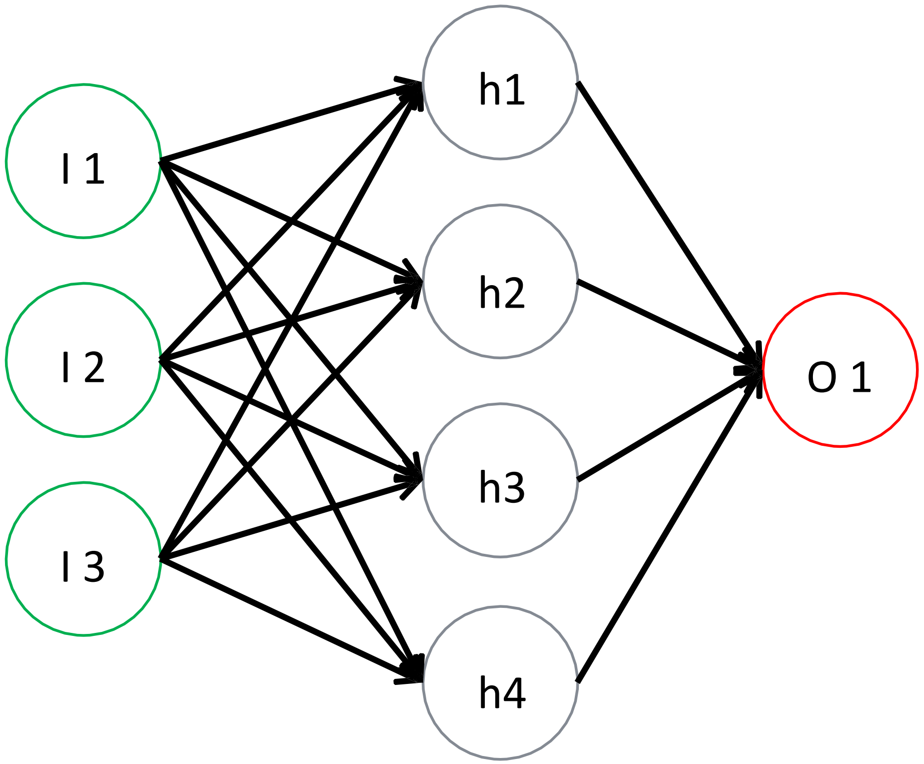


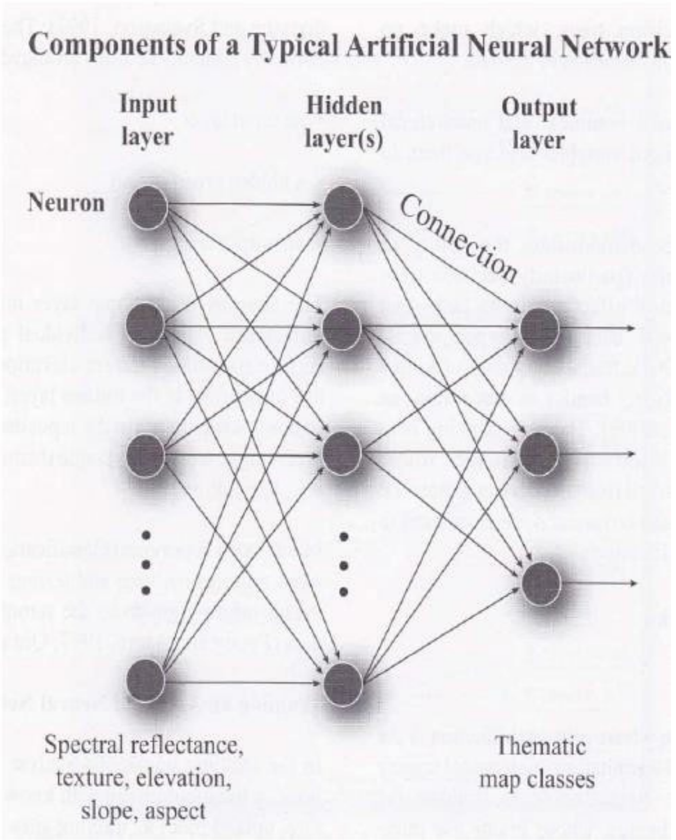


$$I = \begin{bmatrix} I1 \\ I2 \\ I3 \end{bmatrix} \qquad w = \begin{bmatrix} w1 \\ w2 \\ w3 \end{bmatrix}$$

$$O_w(I) = \frac{1}{1 + e^{-w^T I}}$$

Neural Network





Neural Network

- Training is based on error correction rule
- Initial weights and threshold are set to random
- Each learning iteration consist of a forward pass and a backward pass.
- Measure the error to known training samples, cost function.
- Minimize the error in a least-square sense

Neural Network

Advantages

+ Non-parametric, can use any numerical input
+ Handles multi-sensor, high-dimensional data well
+ Generalize well
+ Noise tolerant

Disadvantages

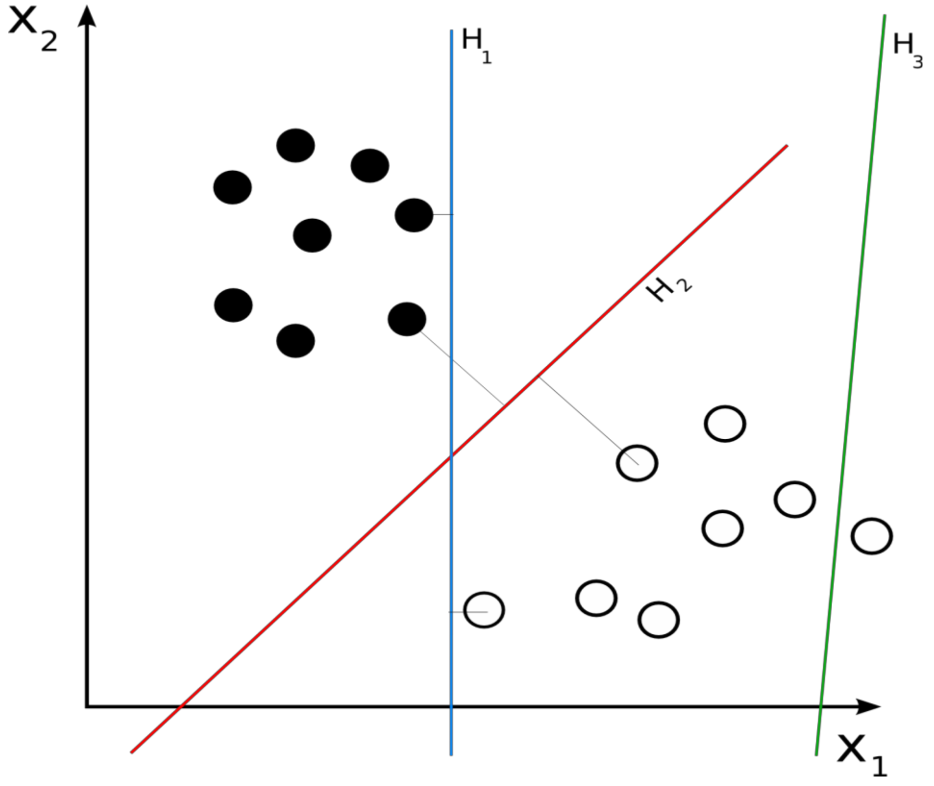
- Number of hidden layers
- Number of neurons in hidden layers
- Hard to interpret
- Longer training times

Support Vector Machine

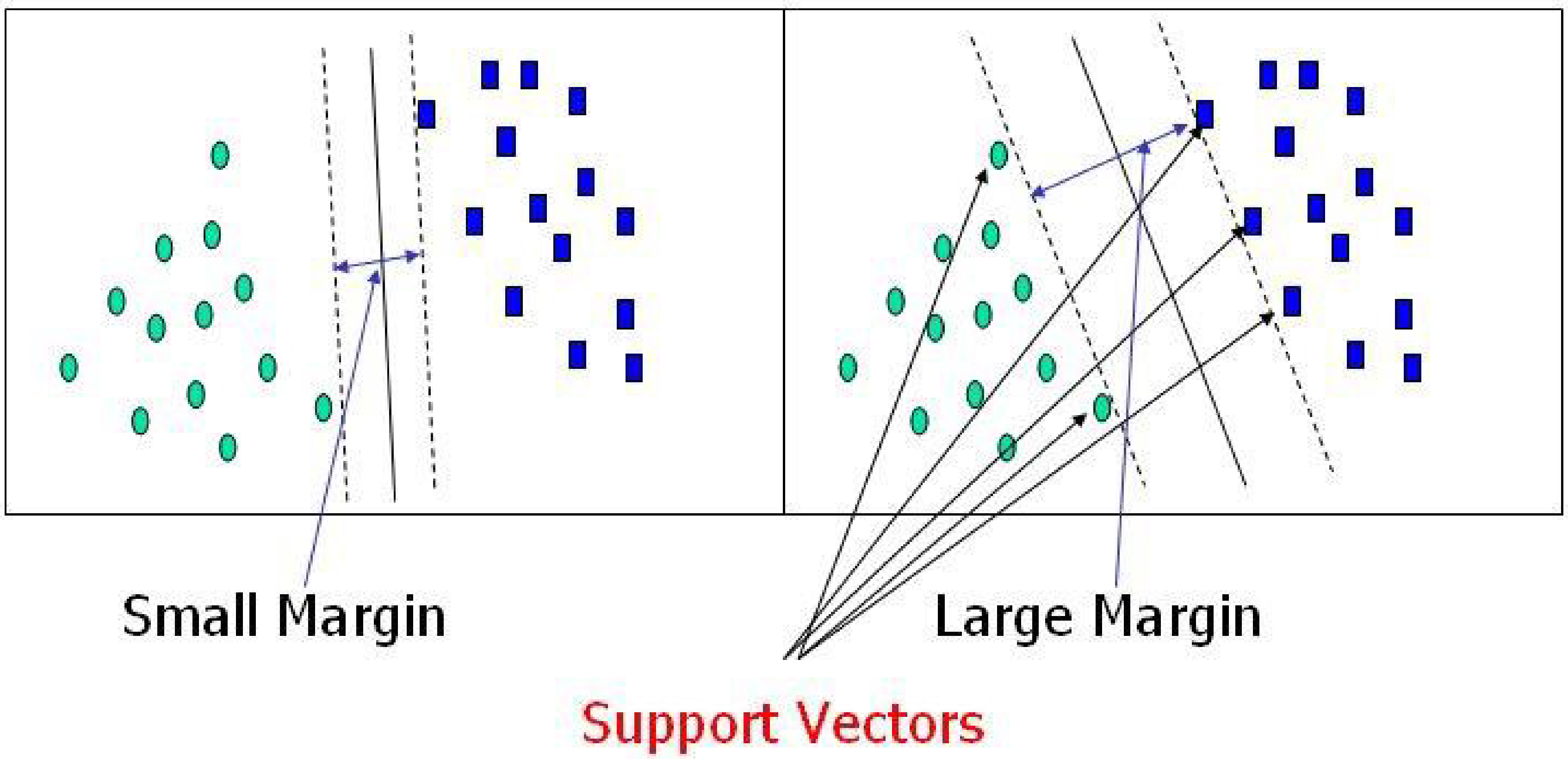
- Maps input space to higher dimensional feature space
- Separates two classes in feature space by a hyper plane
- Has no assumption on data distribution
- Performs well even with little training data
- Comparatively difficult to understand/implement

Support Vector Machine

- Performs classification by constructing an *N*-dimensional hyperplane that optimally separates the data into two categories.
- H3 doesn't separate the 2 classes.
- H1 (blue) does, with a small margin and
- H2 (red) with the maximum margin.



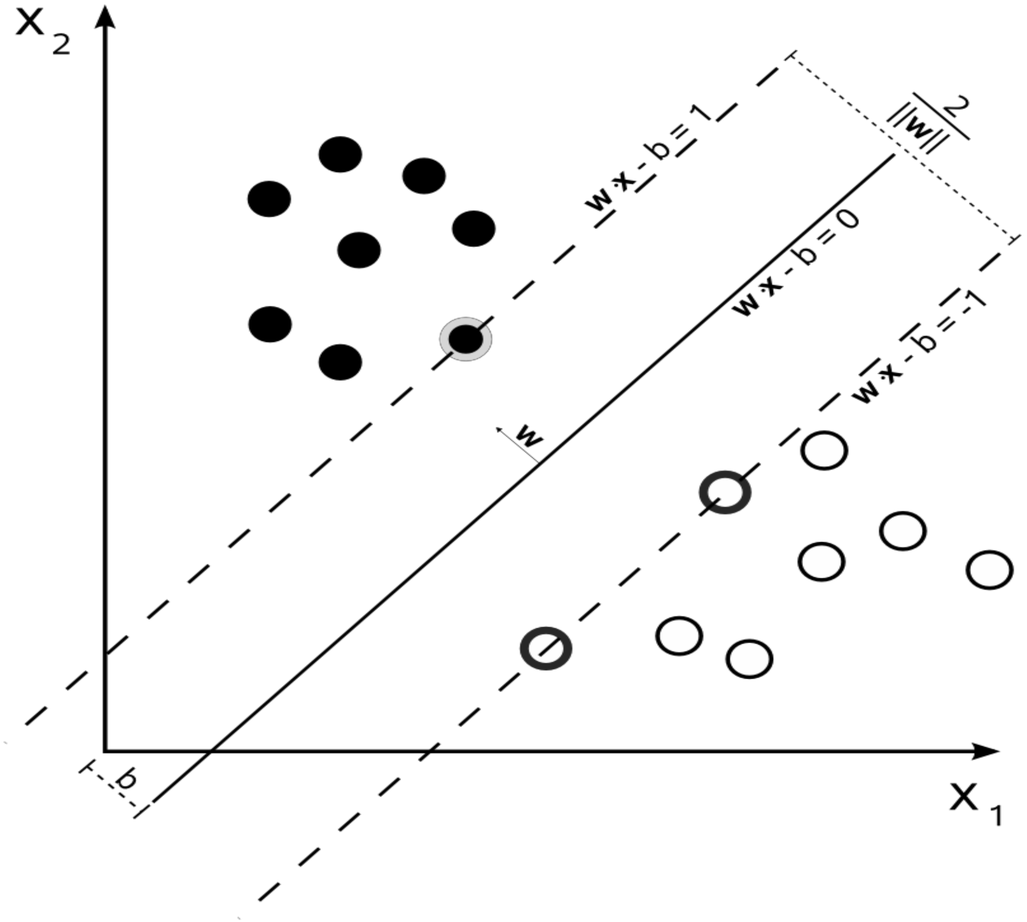
Support Vector Machine



An SVM analysis finds the hyperplane that is oriented so that the margin between the support vectors is maximized.

Support Vector Machine

- Maximum margin
- However, the real case
  - more than two predictor variables,
  - separating the points with non-linear curves,
  - handling the cases where clusters cannot be completely separated, and
  - handling classifications with more than two classes.



Support Vector Machine

- More than 2 classes
  - (1) “one against many” where each class is split out and all of the other classes are merged; and,
  - (2) “one against one” where $k(k\text{-}1)/2$ models are constructed where k is the number of classes.

Support Vector Machine

- Not straight line, flat plane or an N-dimensional hyperplane, instead of using non-linear curve, using a *kernel function* to map the data into a different space where a hyperplane can be used to do the separation. E.g.
  - Linear, polynomial
  - **Radial basis function:** exp(-gamma*|u-v|^2) (recommended)

Summary

- General Classifier Characteristics
- Classic Classifiers -> statistical, cluster distances
- Expert systems (hierarchical, decision trees)
- Machine Learning
  - Random Forest
  - Neural Networks
  - Support Vector Machine