HALMSTAD UNIVERSITY

Learning Systems (DT8008)

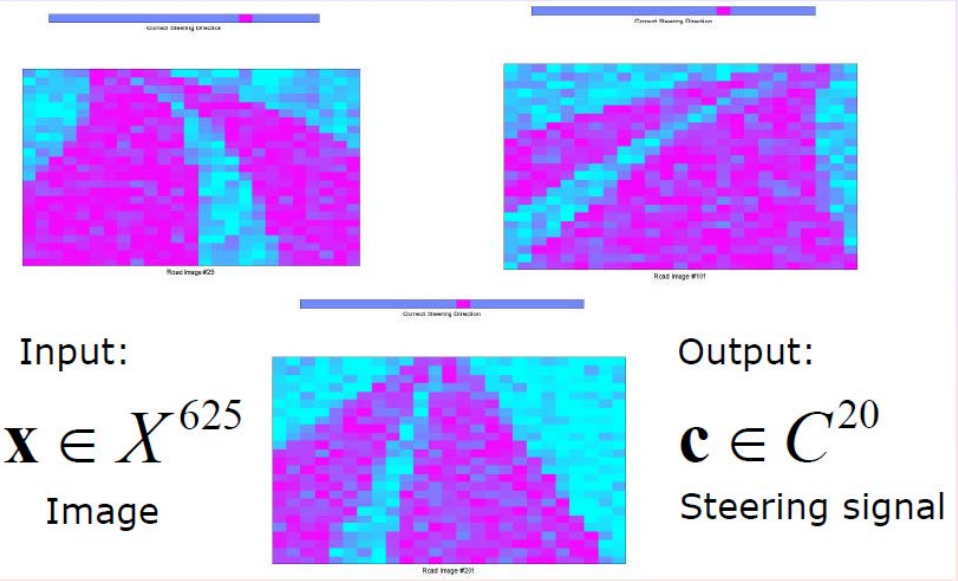
Introduction to Classification

Dr. Mohamed-Rafik Bouguelia
mohbou@hh.se

Halmstad University

Applications of classification (1)

ANN guided vehicle (1)



Applications of classification (2)

Classify the Lego pieces into red, blue, and yellow.

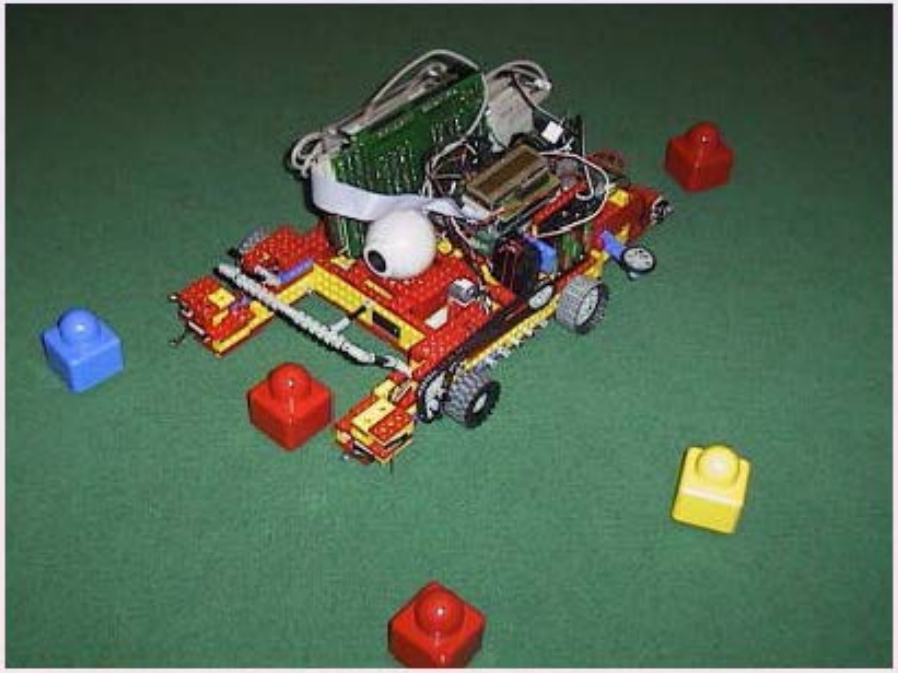
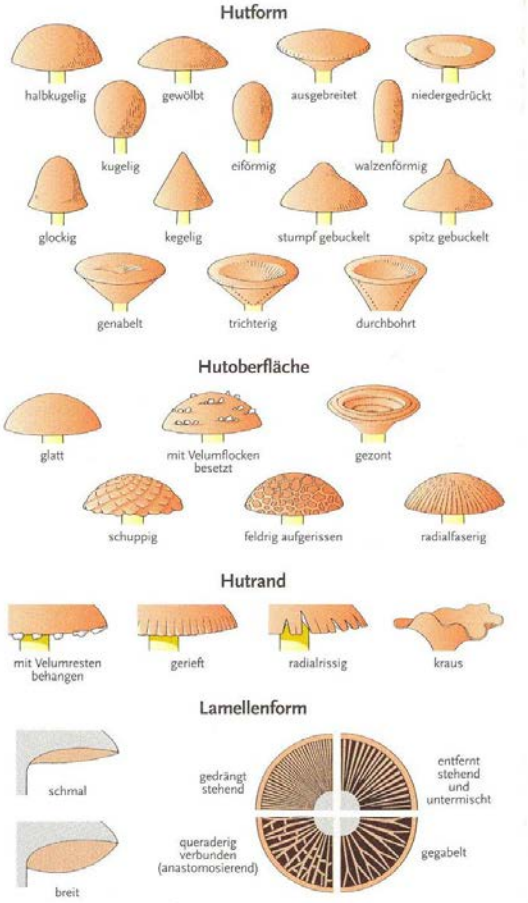
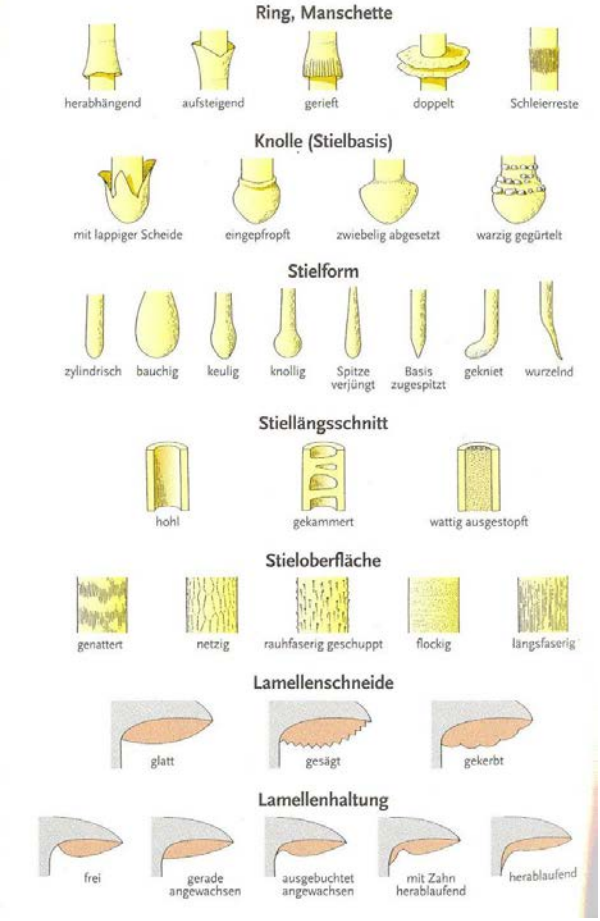


Figure: Robot and Lego pieces.

Applications of classification (3)



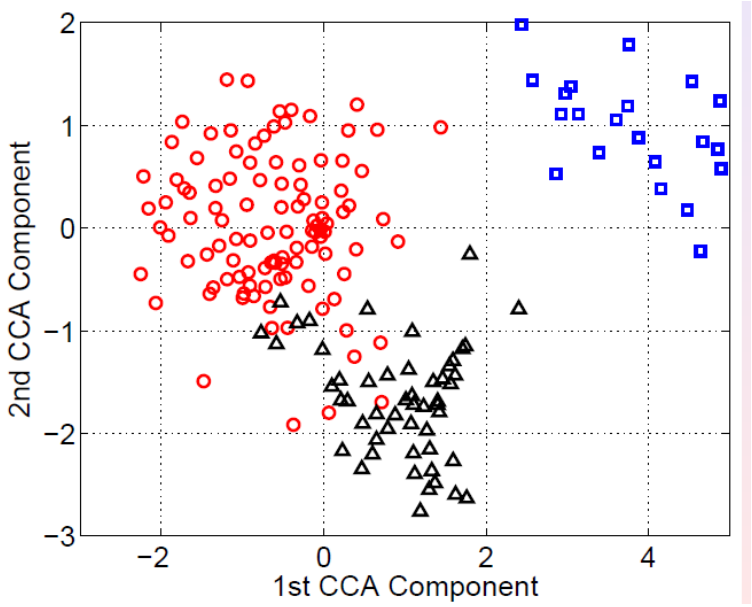


Edible or poisonous ?

Applications of classification (4)

- e.g. Laryngeal disease diagnostics

- Features / Attributes:
  - Age
  - Subjectively estimated illness duration (months)
  - Education (five grades)
  - Average duration of intensive speech use (hours/day)
  - Number of days of intensive speech use (days/week)
  - Smoking (Yes/No)
  - Smoked cigarettes/day
  - Smoking duration (years);
  - Subjective voice function assessment by the patient
  - Maximal tonality duration for "aaaaaa" (sec)
  - Functional voice index (F);
  - Emotional condition index (E);
  - Physical condition index (P);
  - Voice deficiency index



Example: Spam Filter

- Input: email
- Output: spam/ham
- Setup:
  - Get a large collection of example emails, each labeled “spam” or “ham”
  - Note: someone has to hand label all this data!
  - Want to learn to predict labels of new, future emails

- Features: The attributes used to make the ham / spam decision
  - Words: FREE!
  - Text Patterns: $dd, CAPS
  - Non-text: SenderInContacts
  - …

Dear Sir.

First, I must solicit your confidence in this transaction, this is by virture of its nature as being utterly confidencial and top secret. …

TO BE REMOVED FROM FUTURE MAILINGS, SIMPLY REPLY TO THIS MESSAGE AND PUT "REMOVE" IN THE SUBJECT.

99 MILLION EMAIL ADDRESSES FOR ONLY $99

Ok, Iknow this is blatantly OT but I'm beginning to go insane. Had an old Dell Dimension XPS sitting in the corner and decided to put it to use, I know it was working pre being stuck in the corner, but when I plugged it in, hit the power nothing happened.

Slide from Dan Klein

Applications of classification (6)

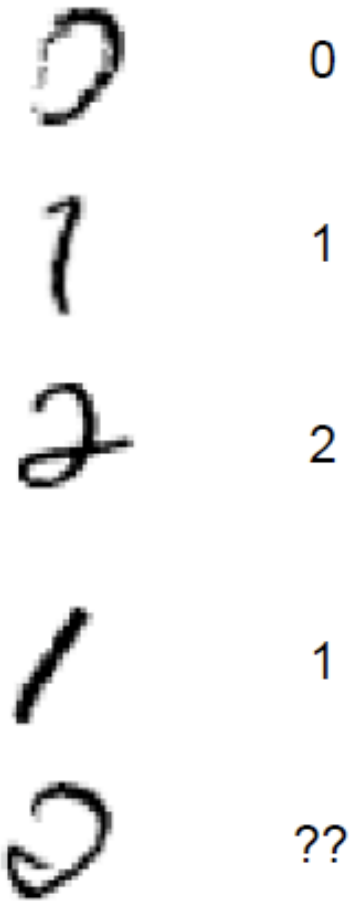
Training set (labels known)

Test set (labels unknown)

- Key challenge: *generalization* to unseen examples

Example: Digit Recognition

- Input: images / pixel grids
- Output: a digit 0-9



http://yann.lecun.com/exdb/mnist/index.html

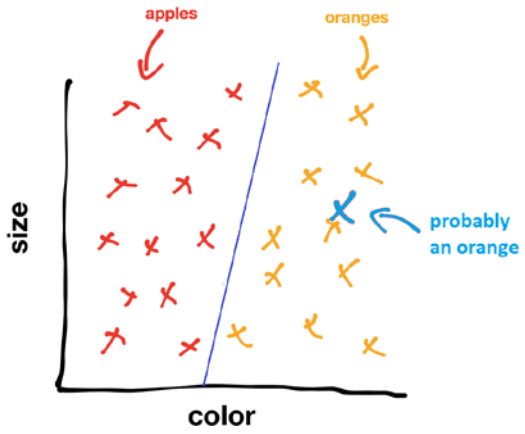
Slide from Dan Klein

Supervised learning for classification

- Learning decision boundaries. The task here is to find optimal borders between the different categories.

- Given a training data and their labels (target), the goal is to learn a classification model h (also called classifier), that assigns a data-point x into a class y.

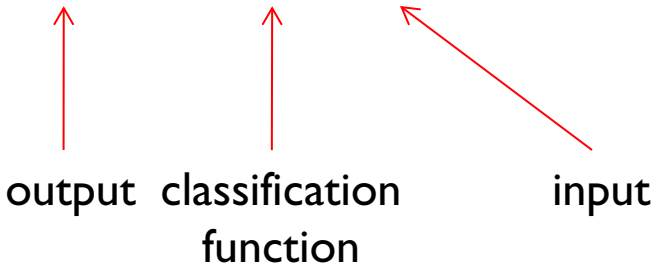
$$\text{Classification model} \quad h : \left| \begin{array}{ccc} \mathbb{R}^d & \longrightarrow & Y \\ x & \longmapsto & y = h(x) \end{array} \right.$$



$$X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \qquad \mathbf{c} = \begin{pmatrix} y_1 \\ y_2 \\ \cdots \\ \cdots \\ y_n \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix}$$

Basic classification function

$$y = h_w(\mathbf{x})$$



- **Learning:** given a *training set* of labeled examples $\{(\mathbf{x}_1,y_1), \ldots, (\mathbf{x}_N,y_N)\}$, estimate the parameters w of the prediction function h
- **Prediction:** apply h to a never before seen *test example* $\mathbf{x}$ and output the predicted value $y = f(\mathbf{x})$

Classification methods

- Nearest Neighbors
- Decision Trees / Random Forest …
- Support Vector Machines
- Artificial Neural Networks
- Bayesian methods
- Hidden Markov Models (HMMs)
- …

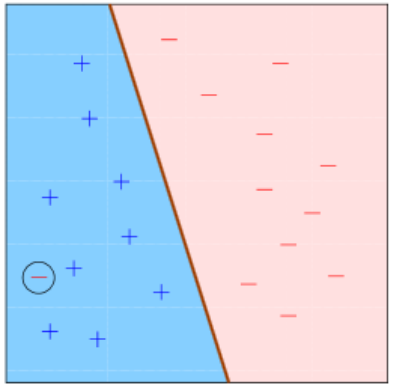
Overfitting

- Overfitting:
  - A classifier that performs well on the training examples, but poorly on new examples.
  - Training and testing on the same data will generally produce a good classifier (for this dataset) with high overfitting.

- To avoid overfitting:
  - Use separate training and testing data
  - Use cross-validation
  - Try using simple models first

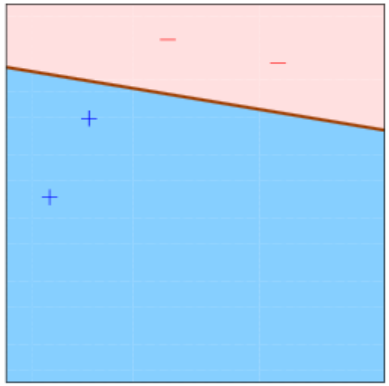
Good and Bad Classifiers

Good:

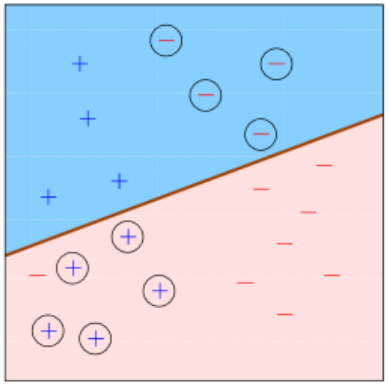


sufficient data
low training error
simple classifier

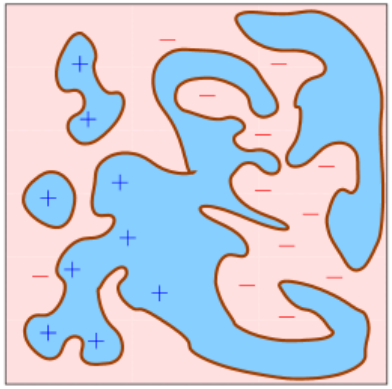
Bad:



insufficient data



training error
too high



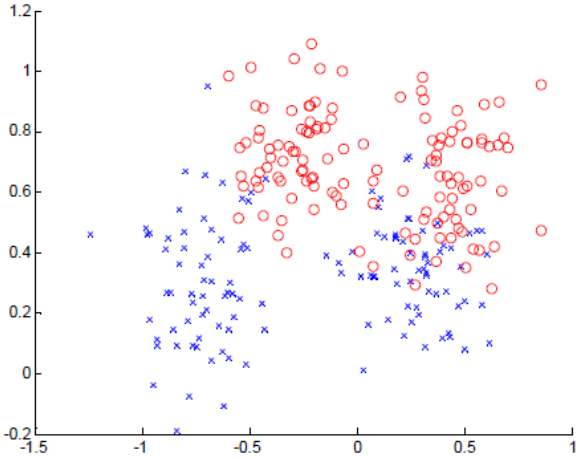
classifier
too complex

Evaluation - Classification

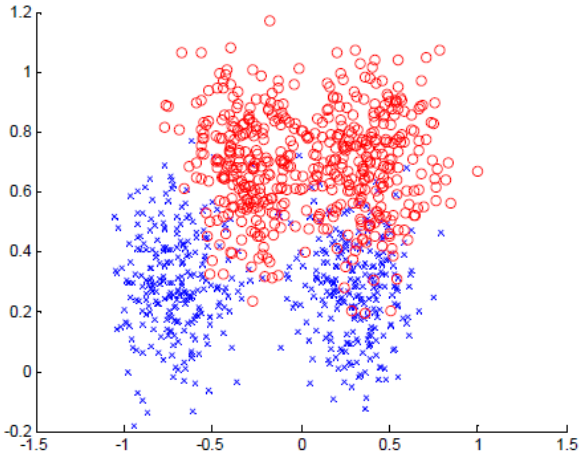
- Split the data into training / Testing sets
- Error rate

$$\frac{1}{N}\sum_{i=1}^{N} \text{loss}(\underbrace{\mathbf{y}_i \neq h(\mathbf{x}_i)}_{\text{loss function}})$$

loss(*cond*) = 1 if *cond* is True, 0 otherwise



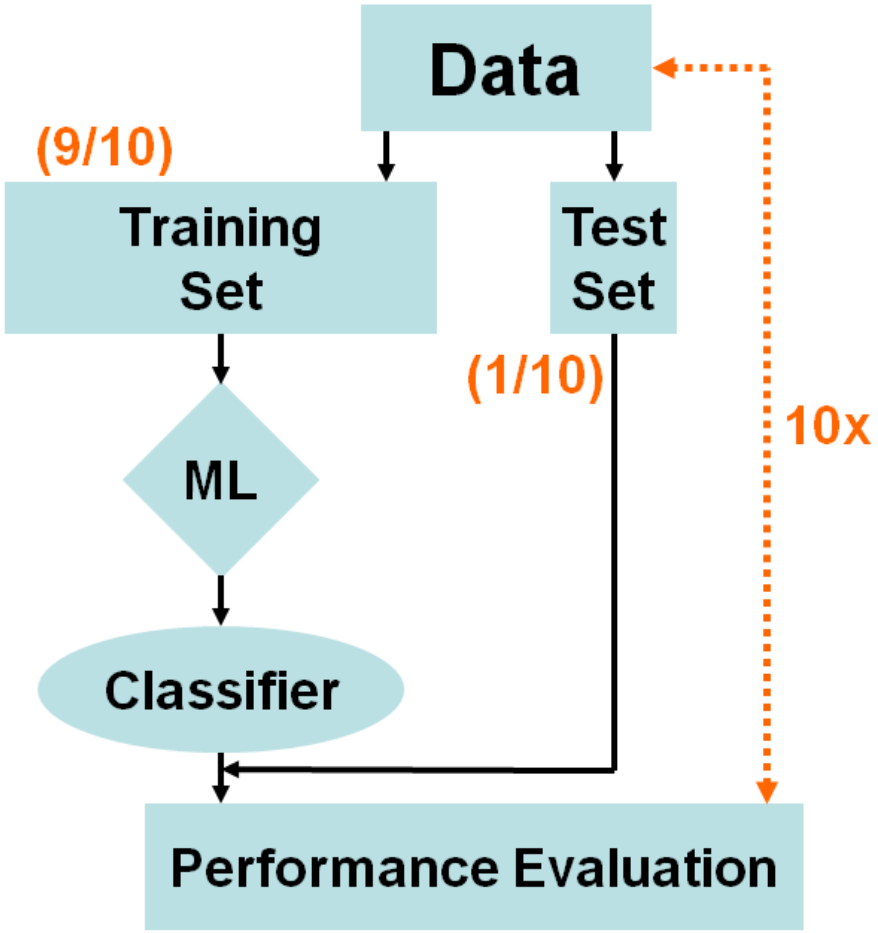
Training data



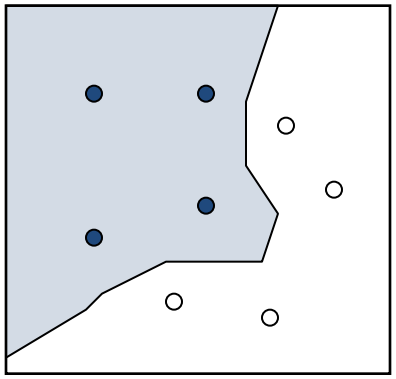
Testing data

Performance evaluation

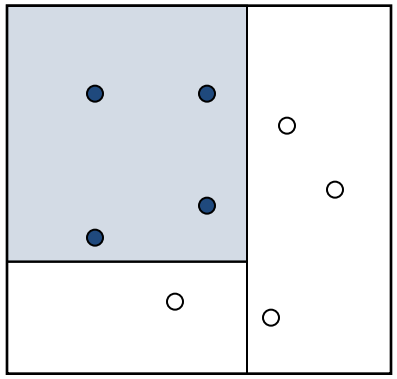
Cross-Validation
(10 fold)



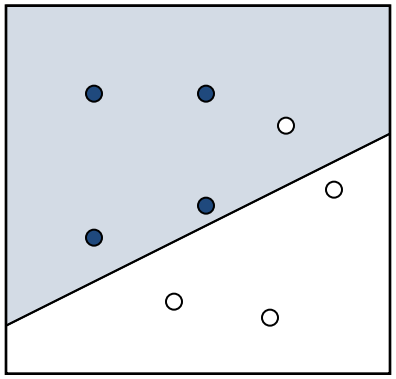
Classification with K-NN



Nearest Neighbor



Decision Tree



Linear Functions

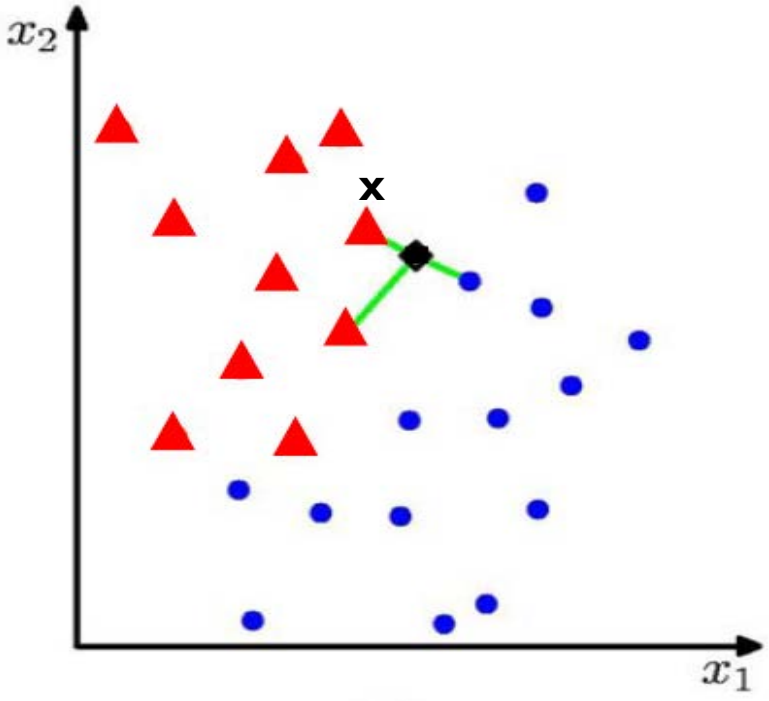
$$g(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$$

K Nearest Neighbors (KNN) - Classification

- Simple method that does not require learning (the model is just the labeled training dataset itself).
- For each test data-point **x**, to be classified, find the K nearest points in the training data.
- Classify the point **x**, according to the majority vote of their class labels

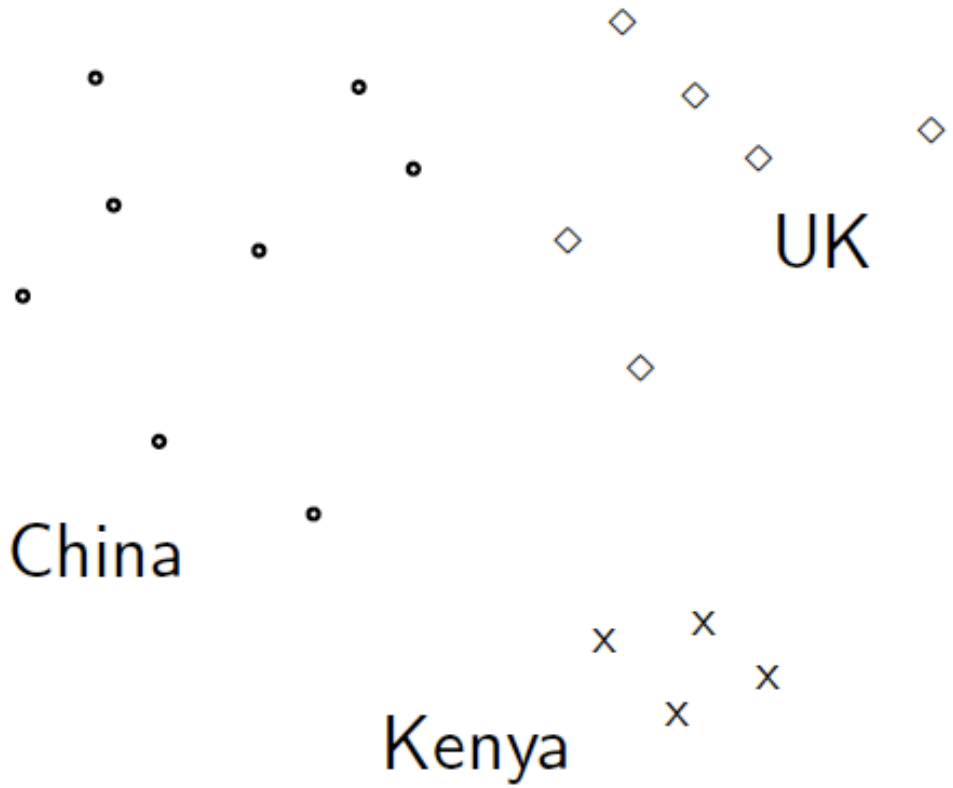
Example:

- K = 3
- 2 classes (red / blue)



Classification by Nearest Neighbor

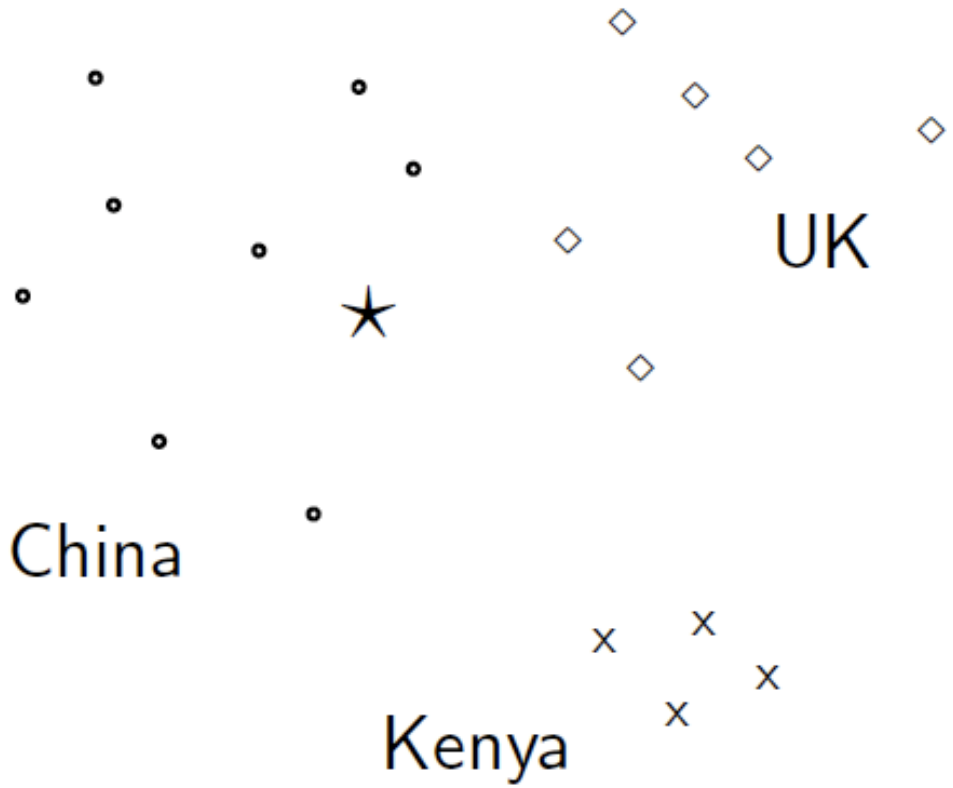
Classes in the vector space



Word vector document classification – here the vector space is illustrated as having 2 dimensions. How many dimensions would the data actually live in?

Classification by Nearest Neighbor

Classes in the vector space



Should the document ⋆ be assigned to *China*, *UK* or *Kenya*?

Classification by Nearest Neighbor

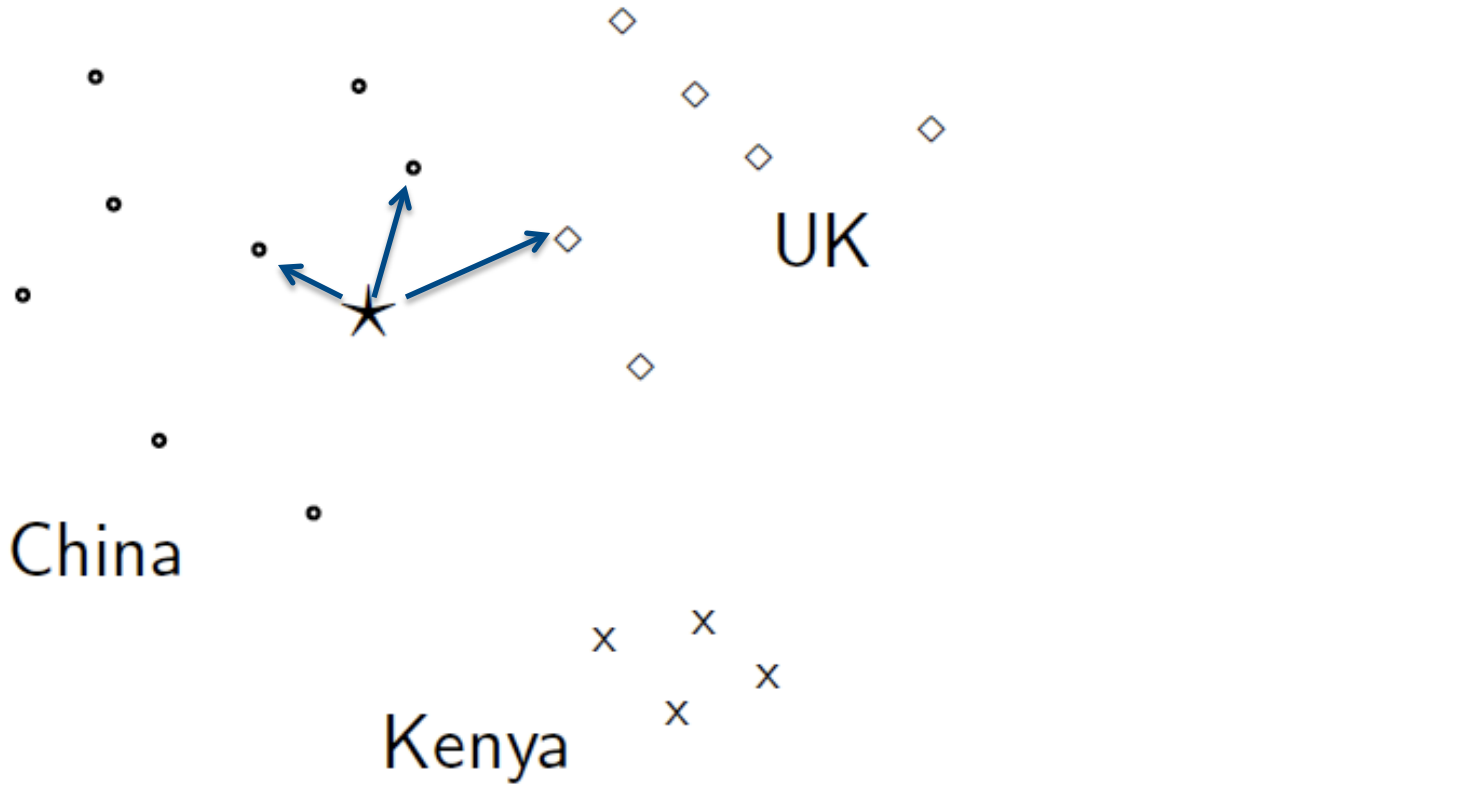
Classes in the vector space

Should the document ⋆ be assigned to *China*, *UK* or *Kenya*?

Classify the test document as the class of the document “nearest” to the query document (use vector similarity to find most similar doc)

Classification by kNN

Classes in the vector space



Should the document ⋆ be assigned to *China*, *UK* or *Kenya*?

Classify the test document as the majority class of the k documents "nearest" to the query document

Classification by kNN

Classes in the vector space

Should the document ⋆ be assigned to *China*, *UK* or *Kenya*?

What are the features? What's the training data? Testing data? Parameters?

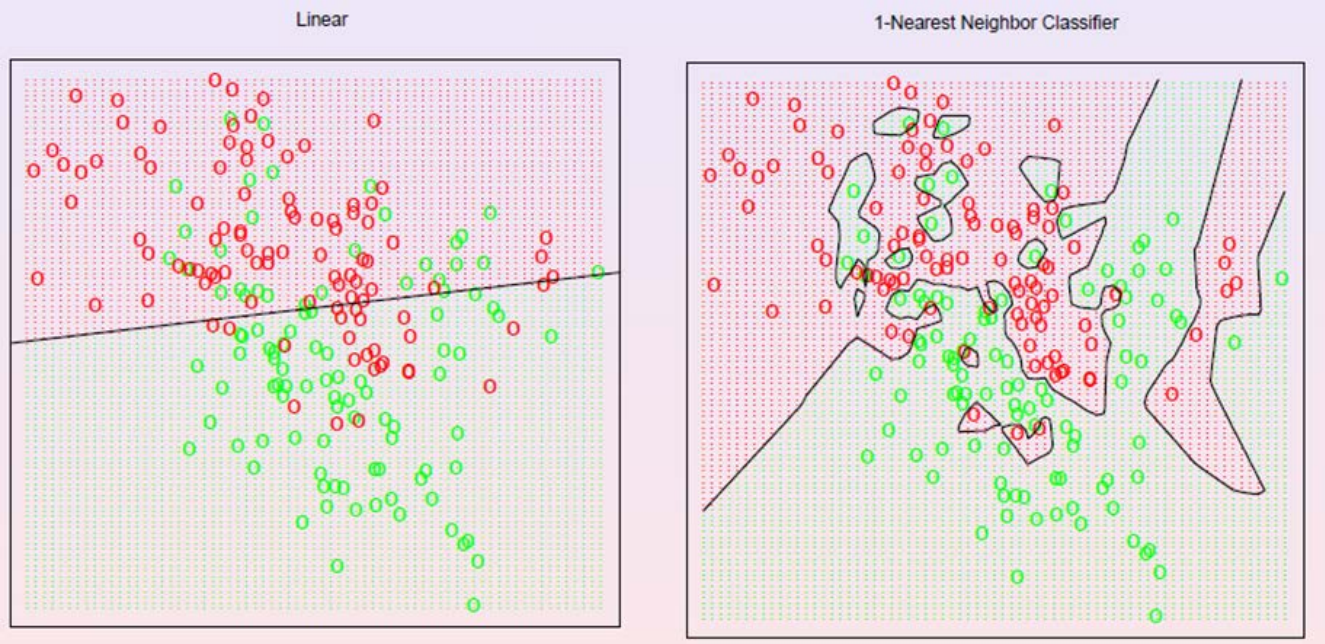


Figure: Decision boundaries of the linear and 1NN classifiers.

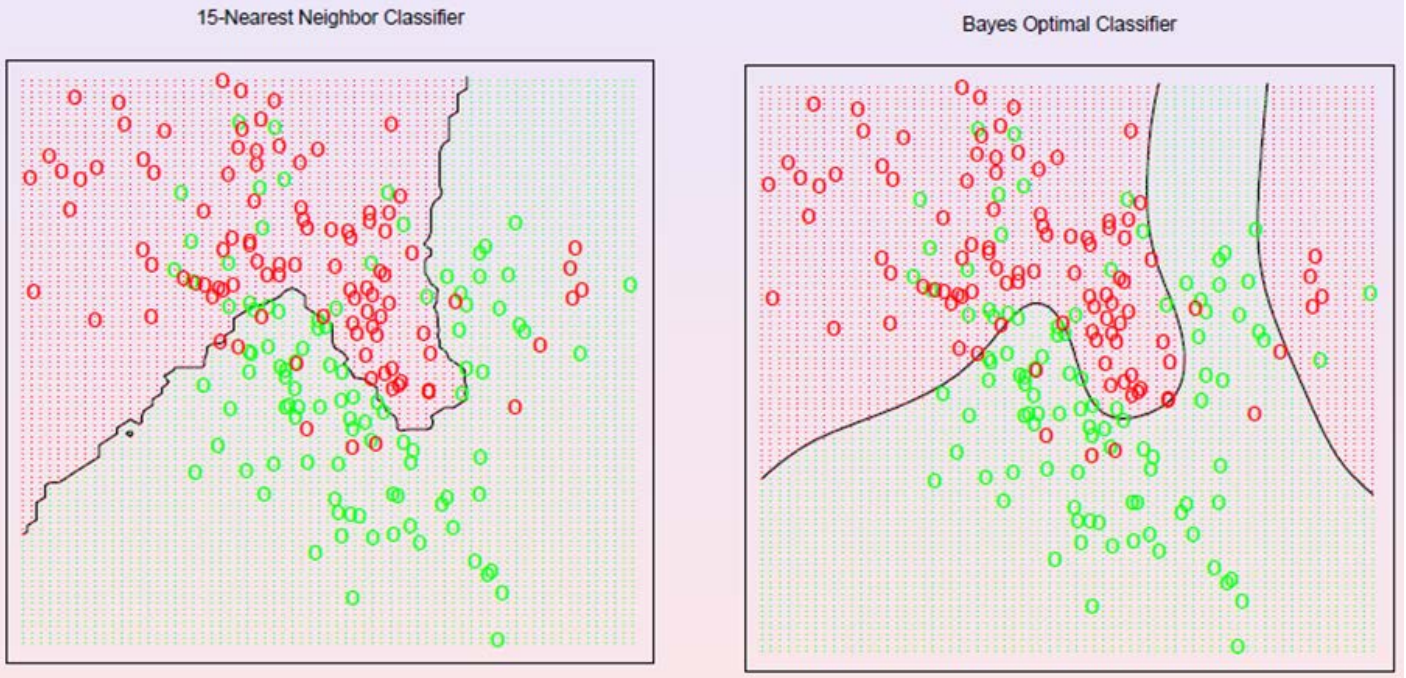
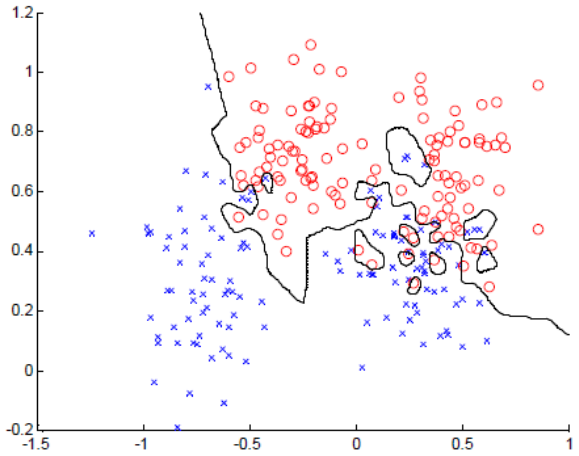


Figure: Decision boundaries of the 15NN and Bayes classifiers.

KNN - Classification

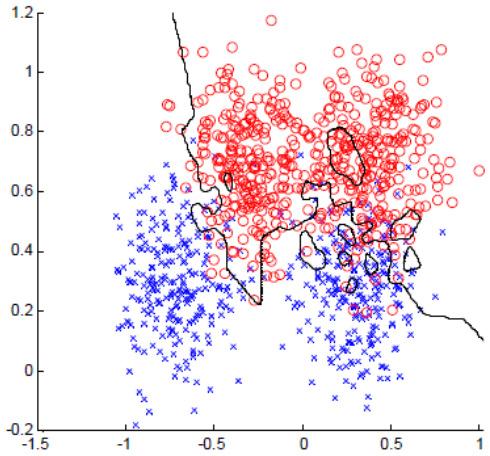
K = 1

Training data



error = 0.0

Testing data

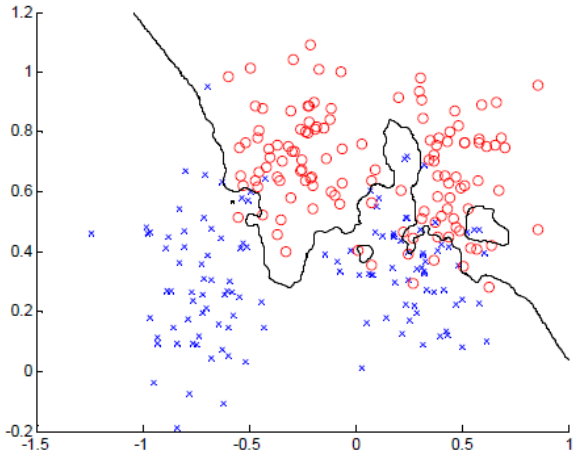


error = 0.15

KNN - Classification

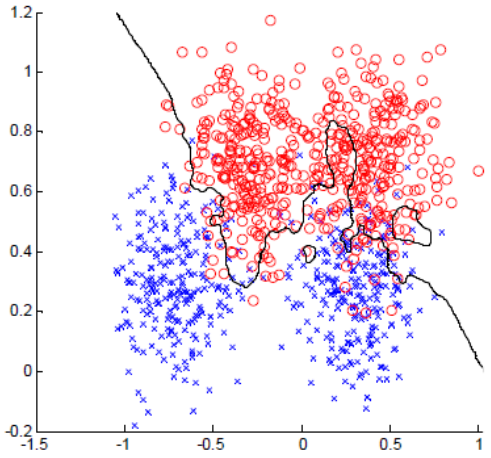
K = 3

Training data



error = 0.0760

Testing data

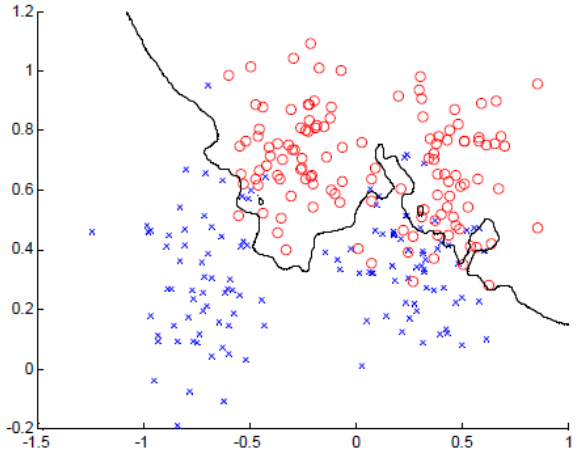


error = 0.1340

KNN - Classification

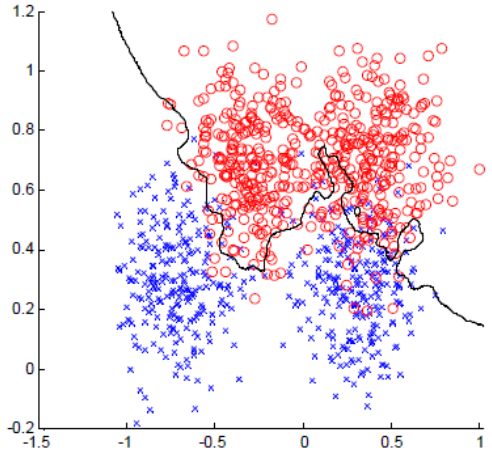
K = 7

Training data



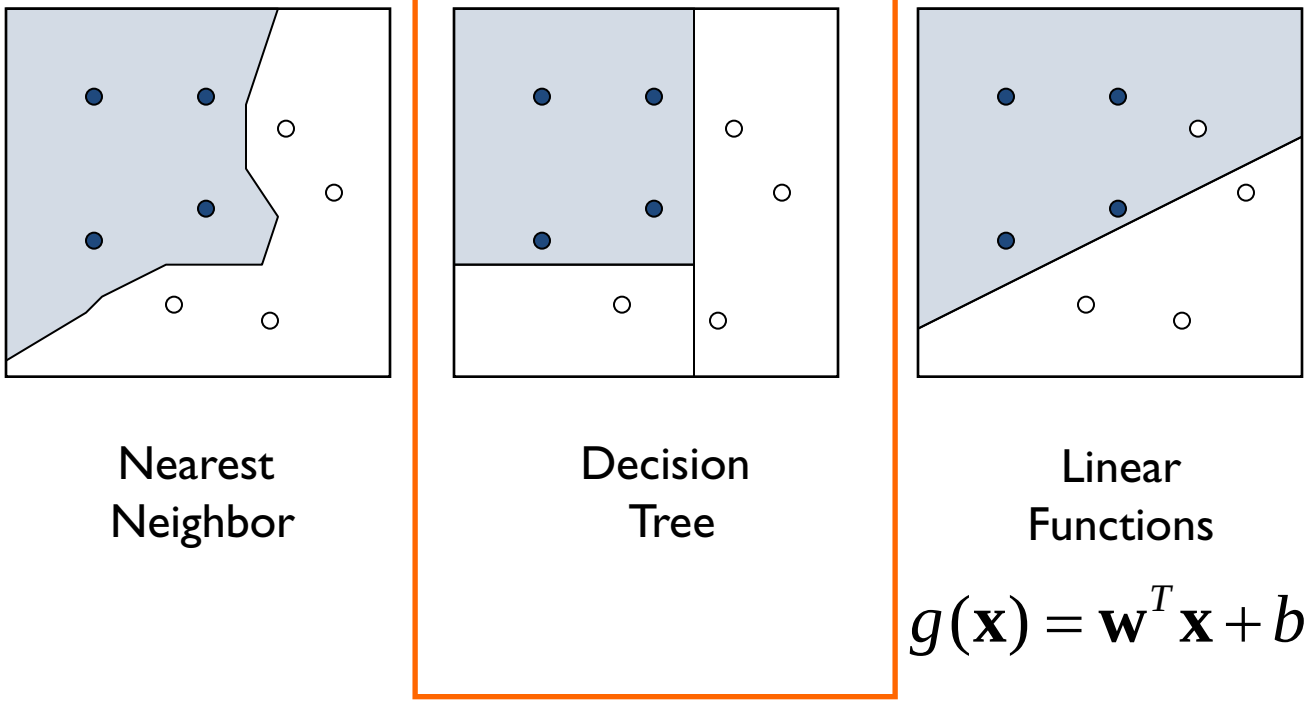
error = 0.1320

Testing data



error = 0.1110

Classification with Decision Trees



Nearest Neighbor

Decision Tree

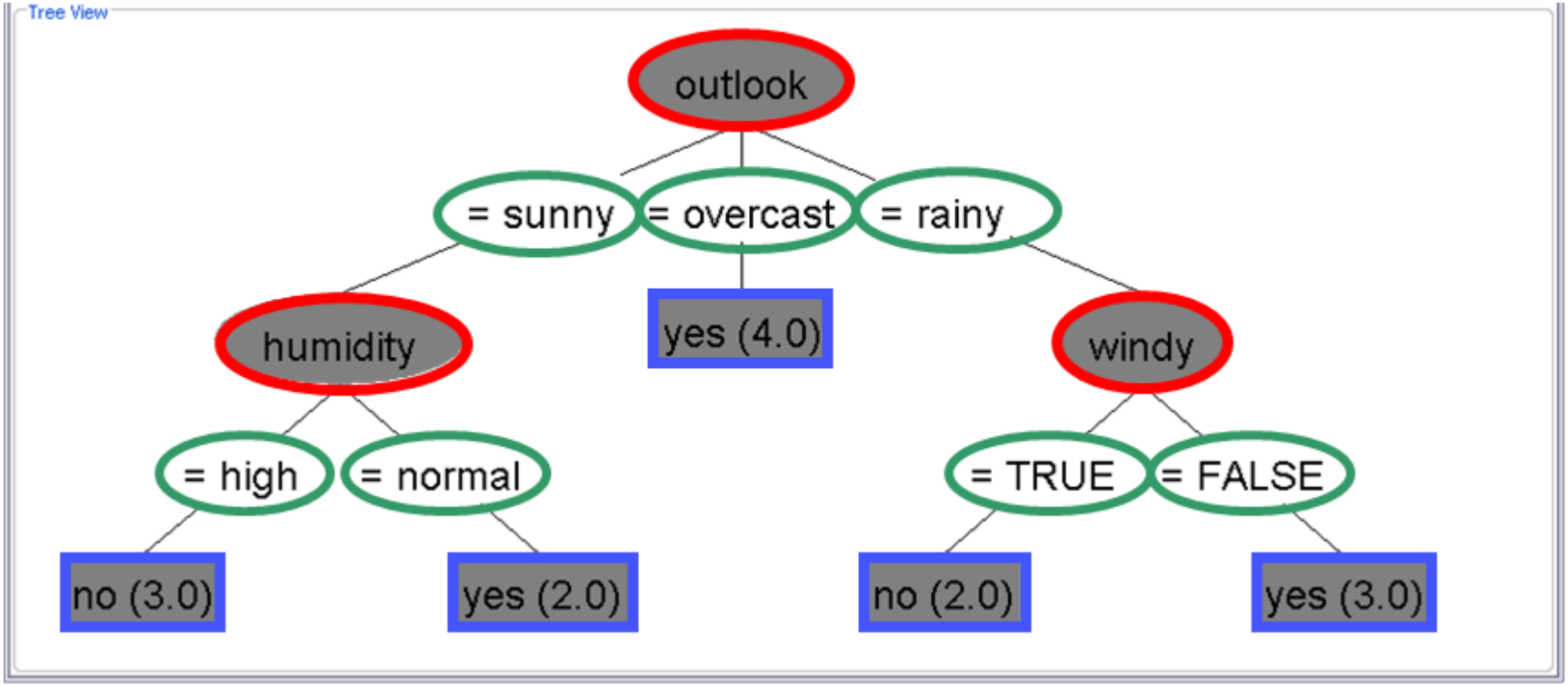
Linear Functions

$$g(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$$

Decision Tree

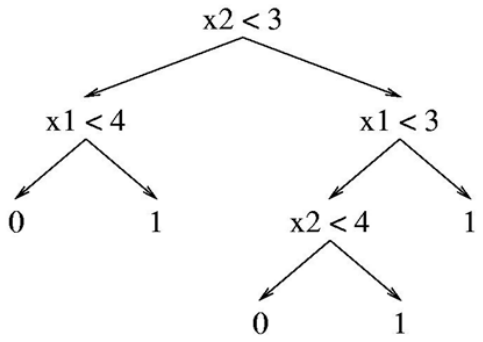
Attributes / Features
Attribute Values
Classes

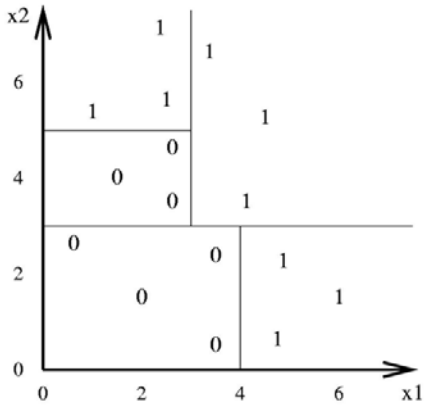
- Example: Shall we play golf today ?



Decision Tree - Classification

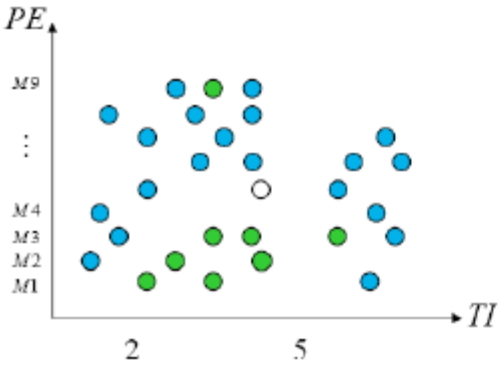
- At each nodes
  - A question is asked about data
  - One child node per possible answer
- Leaf nodes
  - Class label (i.e. decision to take)

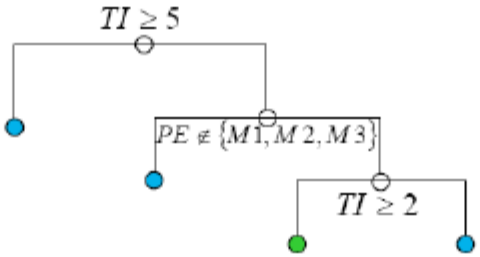


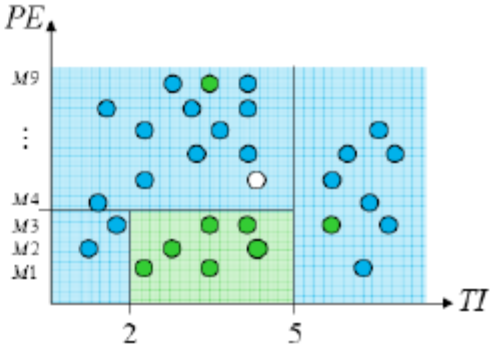


- Building the Tree:
  - For each node, find the feature F + threshold value T
  - ... that split the samples assigned to the node into 2 subsets
  - ... so as to maximize the label purity within these subsets.

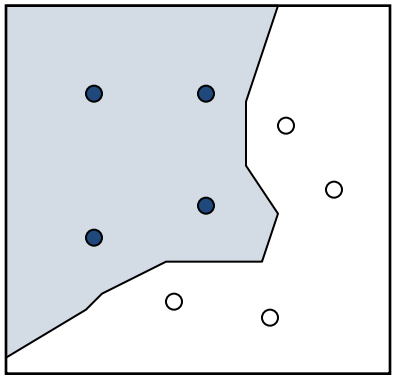
Simple, practical and easy to interpret.
Given a set of instances (with a set of features), a tree is constructed with internal nodes as the features and the leaves as the classes.



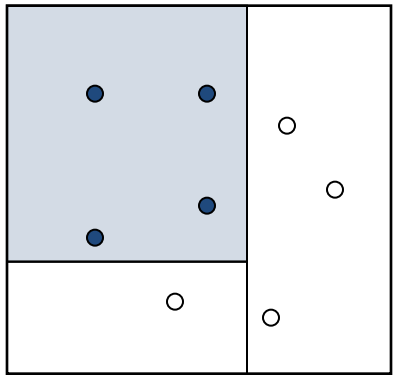




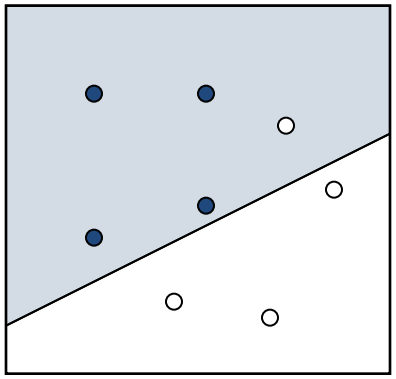
Classification with a Linear classifier



Nearest Neighbor



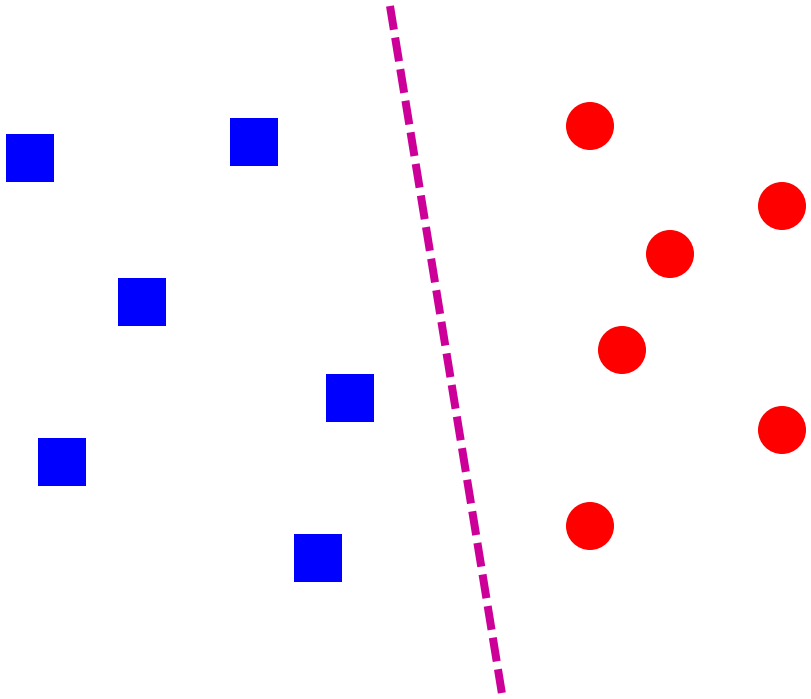
Decision Tree



Linear Functions

$$g(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$$

Linear classifier



- Find a *linear function* to separate the classes

$$f(\mathbf{x}) = \text{sgn}(w_1 x_1 + w_2 x_2 + \ldots + w_D x_D) = \text{sgn}(\mathbf{w} \cdot \mathbf{x})$$

Slide credit: Svetlana Lazebnik

Linear Discriminant Function

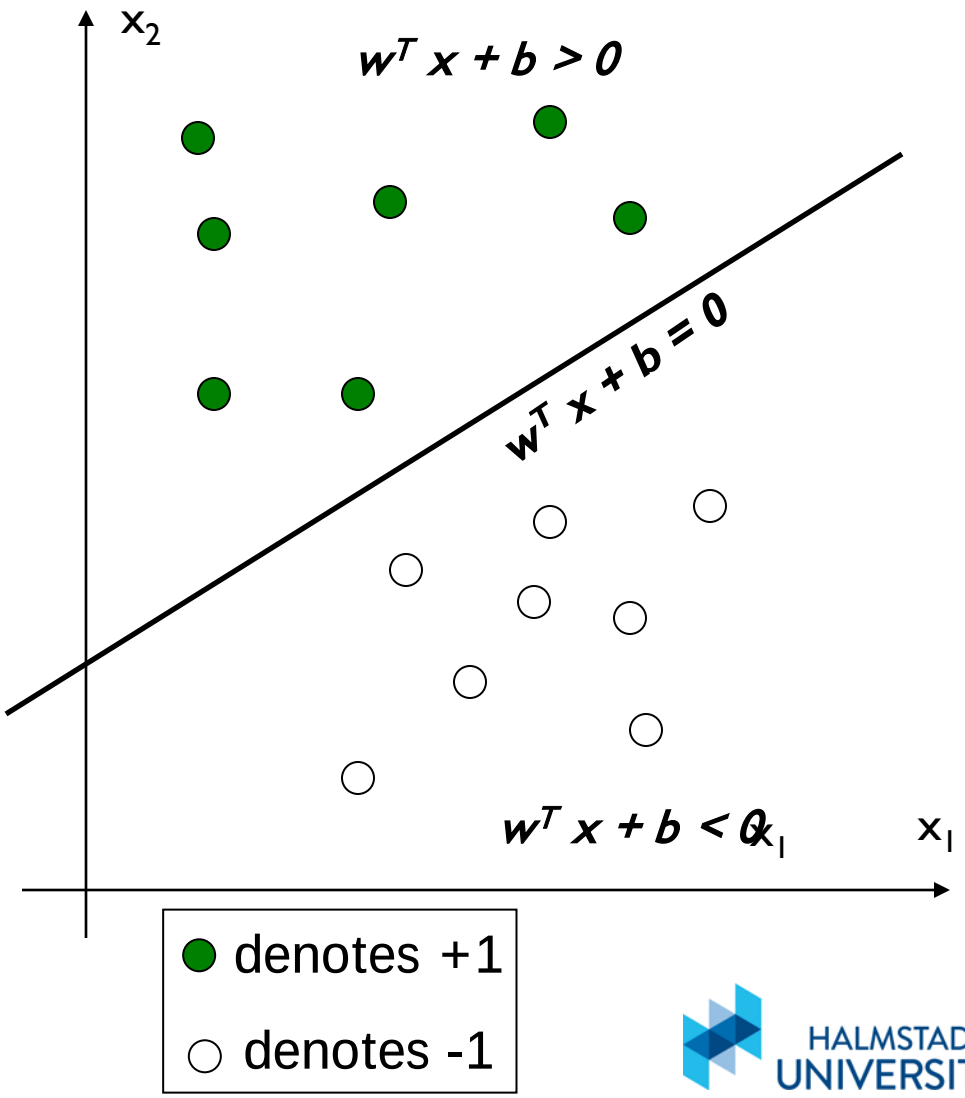
- g(x) is a linear function:

$$g(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$$

- A hyper-plane in the feature space

Find some linear function (hyper plane) that best divides the training samples.

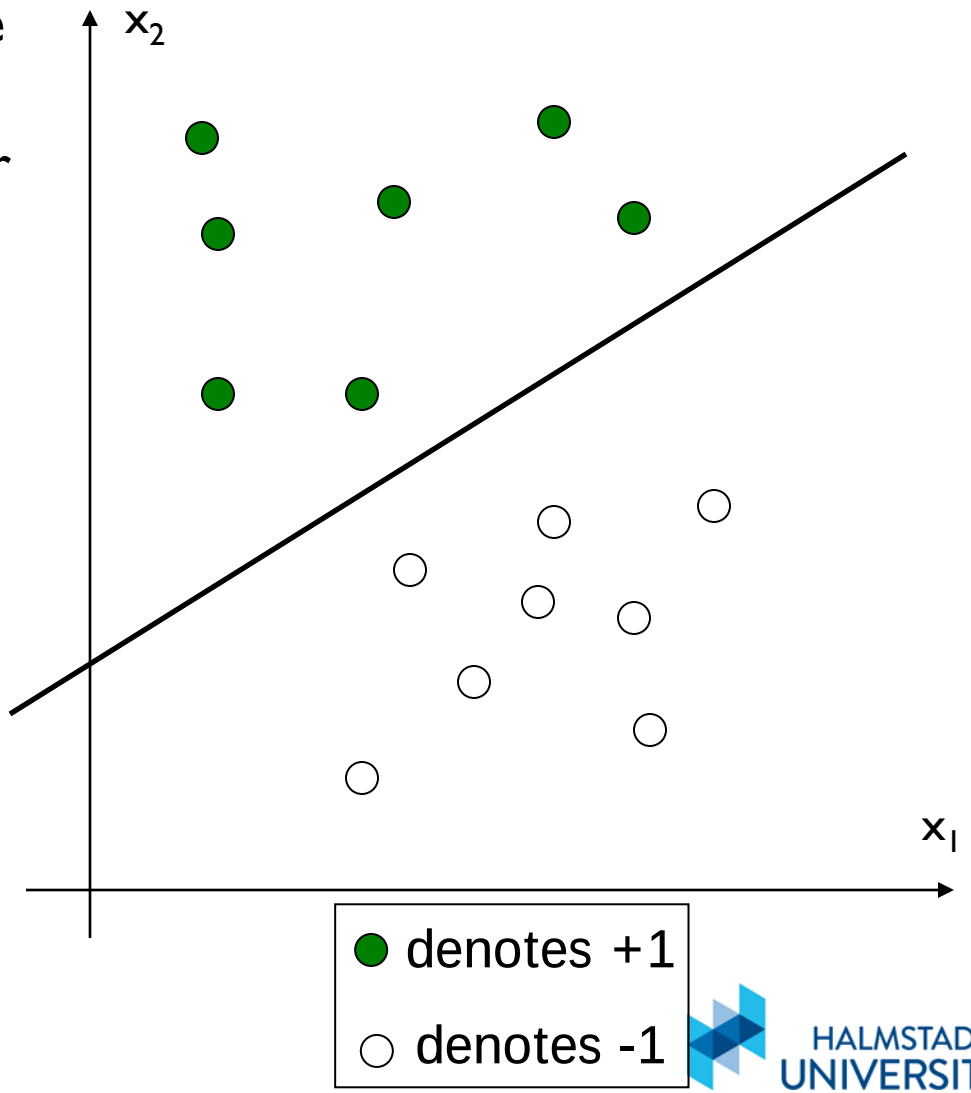
The goal is to compute the weights vector **w** that best divide the training samples; for example visualized here as the green class and the white class.



Linear Discriminant Function

- How would you classify these points using a linear discriminant function in order to minimize the error rate?

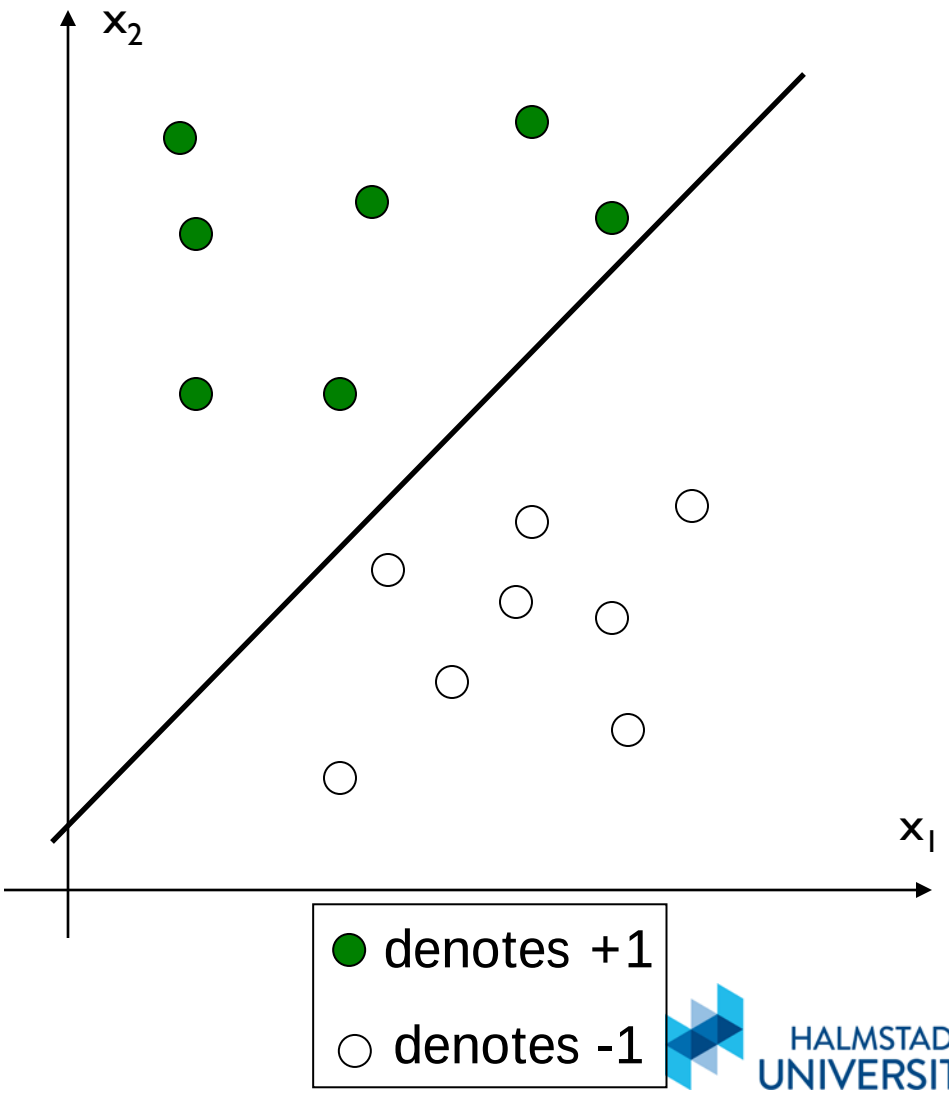
- Infinite number of answers!



Linear Discriminant Function

- How would you classify these points using a linear discriminant function in order to minimize the error rate?

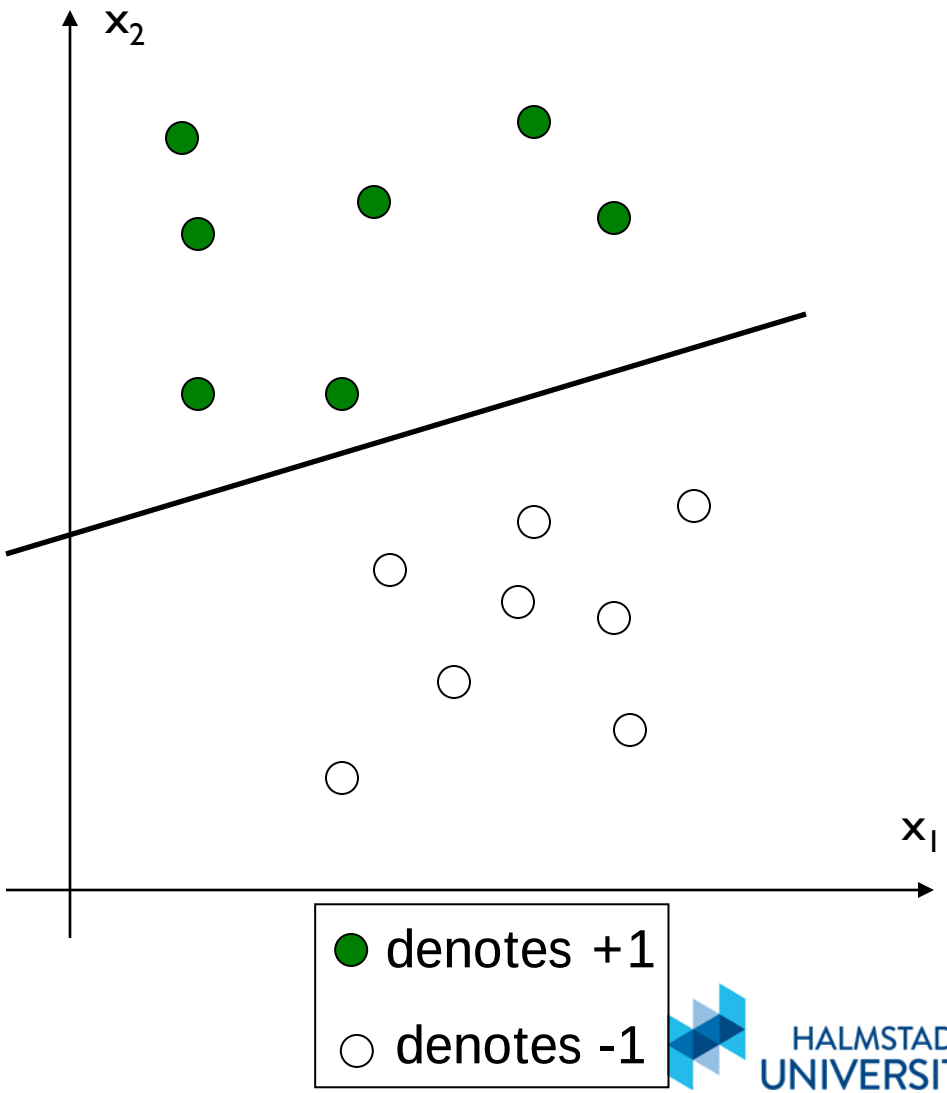
- Infinite number of answers!



Linear Discriminant Function

- How would you classify these points using a linear discriminant function in order to minimize the error rate?

- Infinite number of answers!

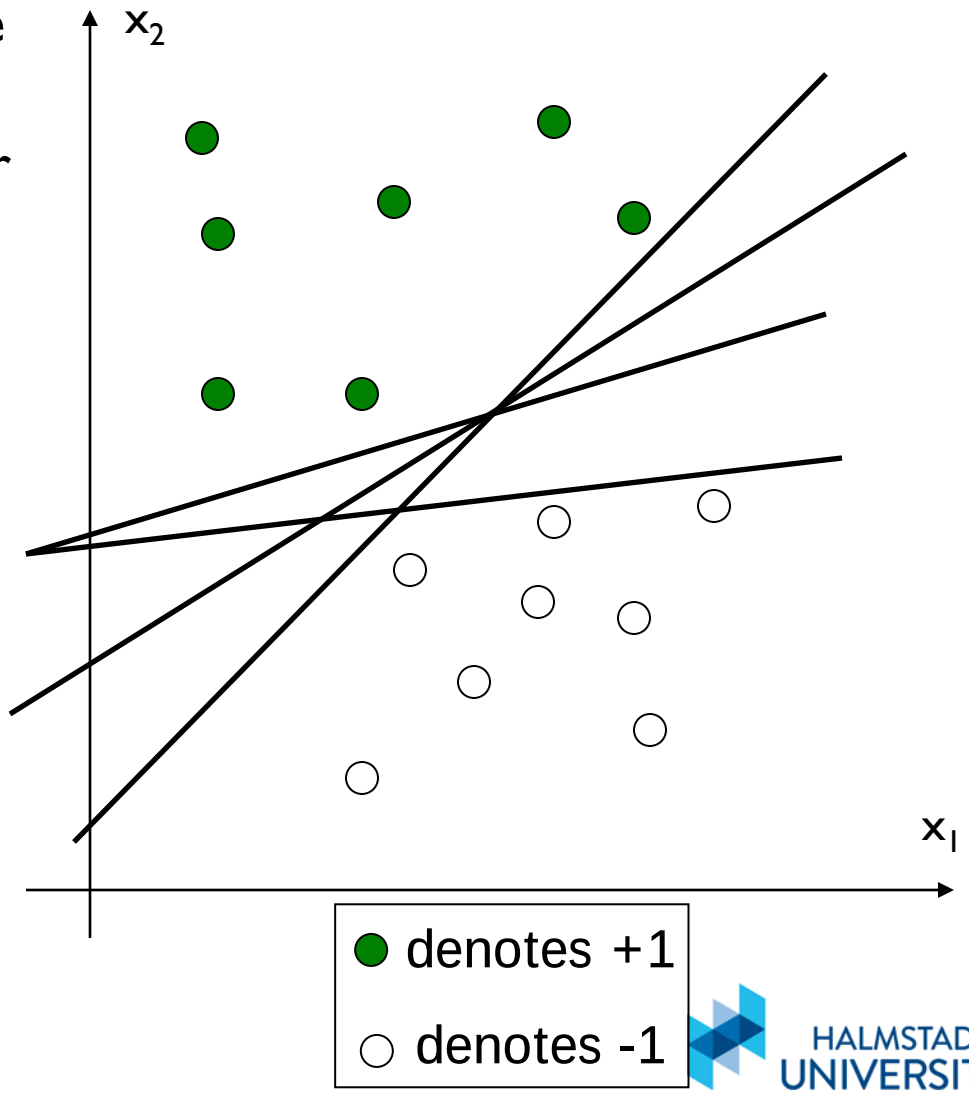


Linear Discriminant Function

- How would you classify these points using a linear discriminant function in order to minimize the error rate?

- Infinite number of answers!

- Which one is the best?

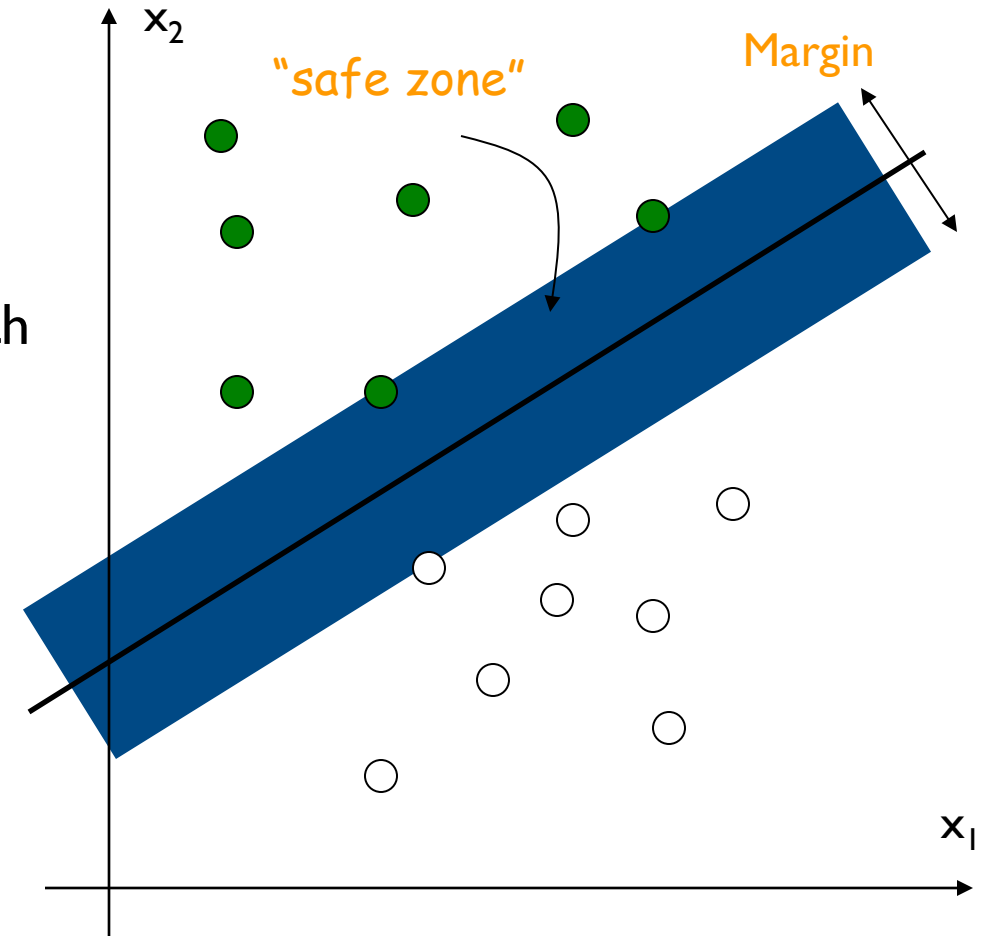


Large Margin Linear Classifier

- The linear discriminant function (classifier) with the maximum margin is the best

- Margin is defined as the width that the boundary could be increased by before hitting a data point

- Why it is the best?
  - strong generalization ability



Linear SVM

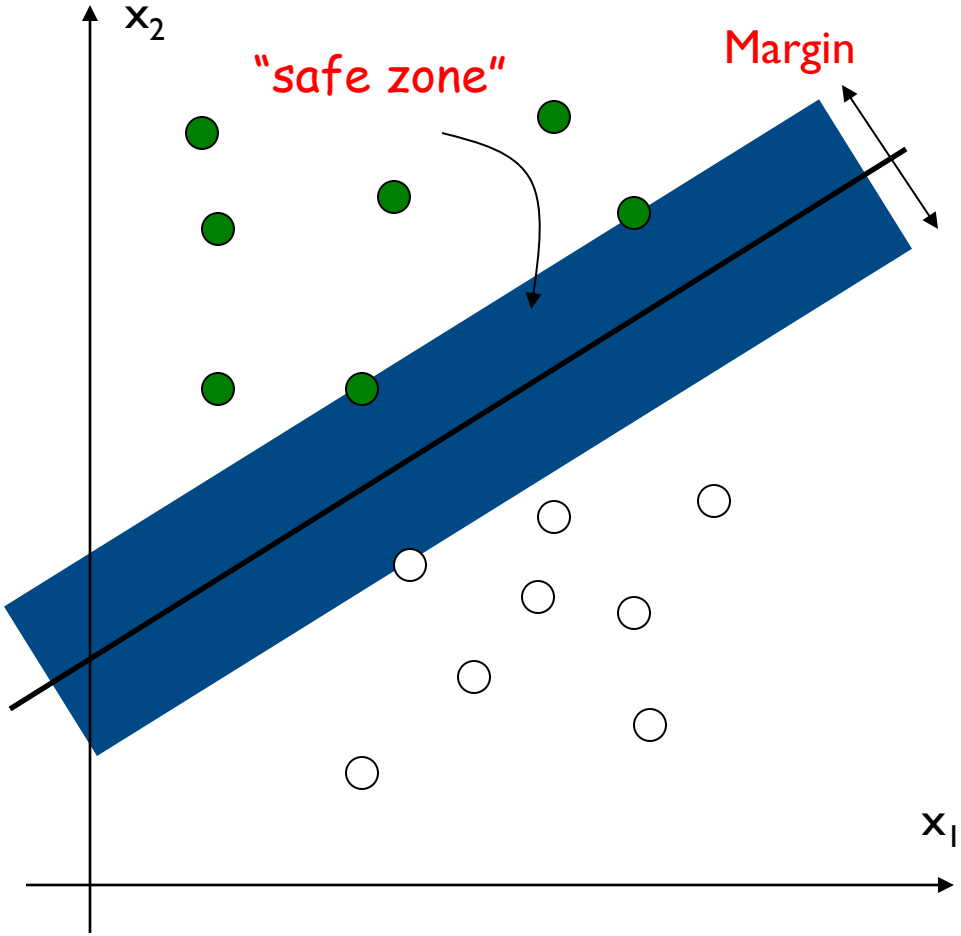
Large Margin Linear Classifier

The idea of a large margin linear classifier is to choose the function with the maximum margin.

The margin is defined as the width that the boundary could be increased by before hitting a data point.

This means that you should put the hyperplane as far as possible from the closest training examples.

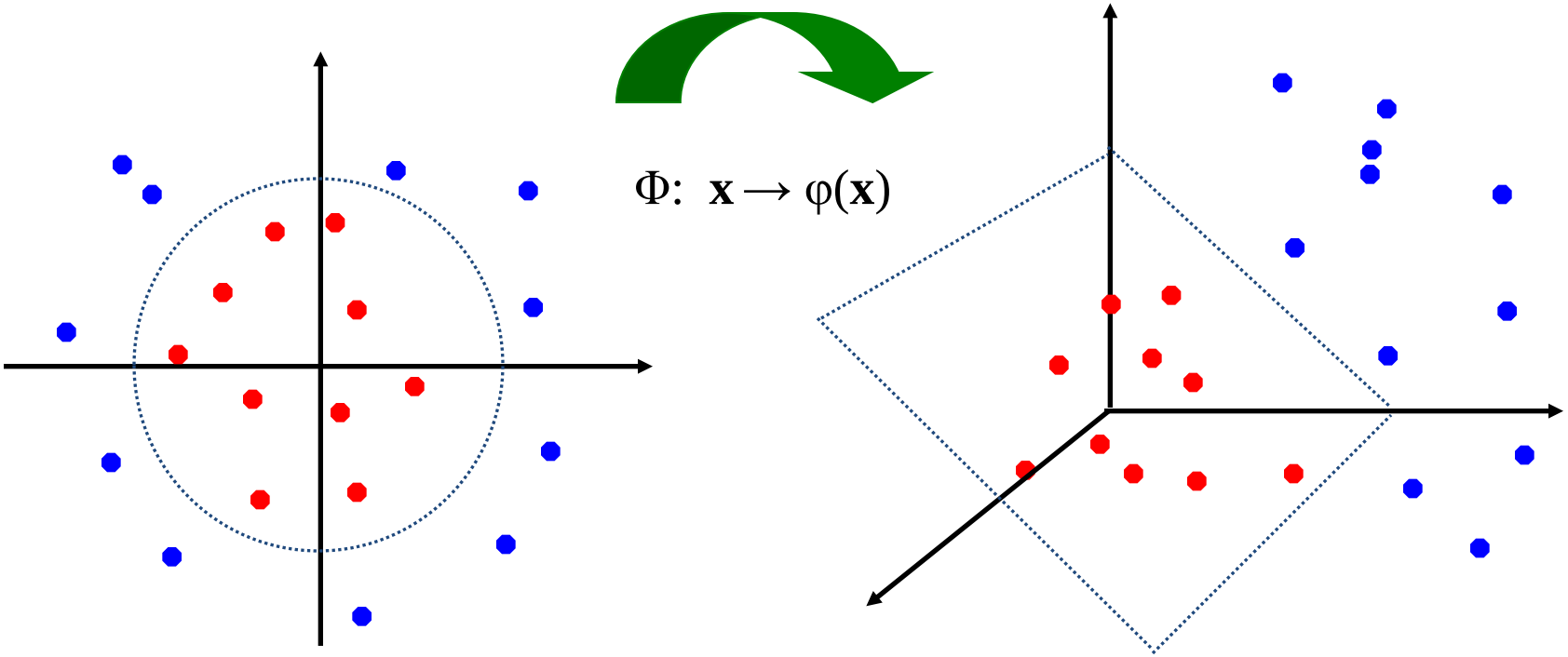
Why is this the best? Because it's the furthest from any of the samples we've seen as possible. So, it will have strong generalization ability – less overfitting to your training samples.



Linear SVM

Non-linear SVMs: Feature Space

- General idea: the original input space can be mapped to some higher-dimensional feature space where the training set is separable:



In the original input space the samples are not linearly separable. So we map the input space into some higher dimensional space where the training set is linearly separable.