Chapter 9

Classification (Part 2)

Xike Xie

Slides are based on Prof. Ben Kao's work.

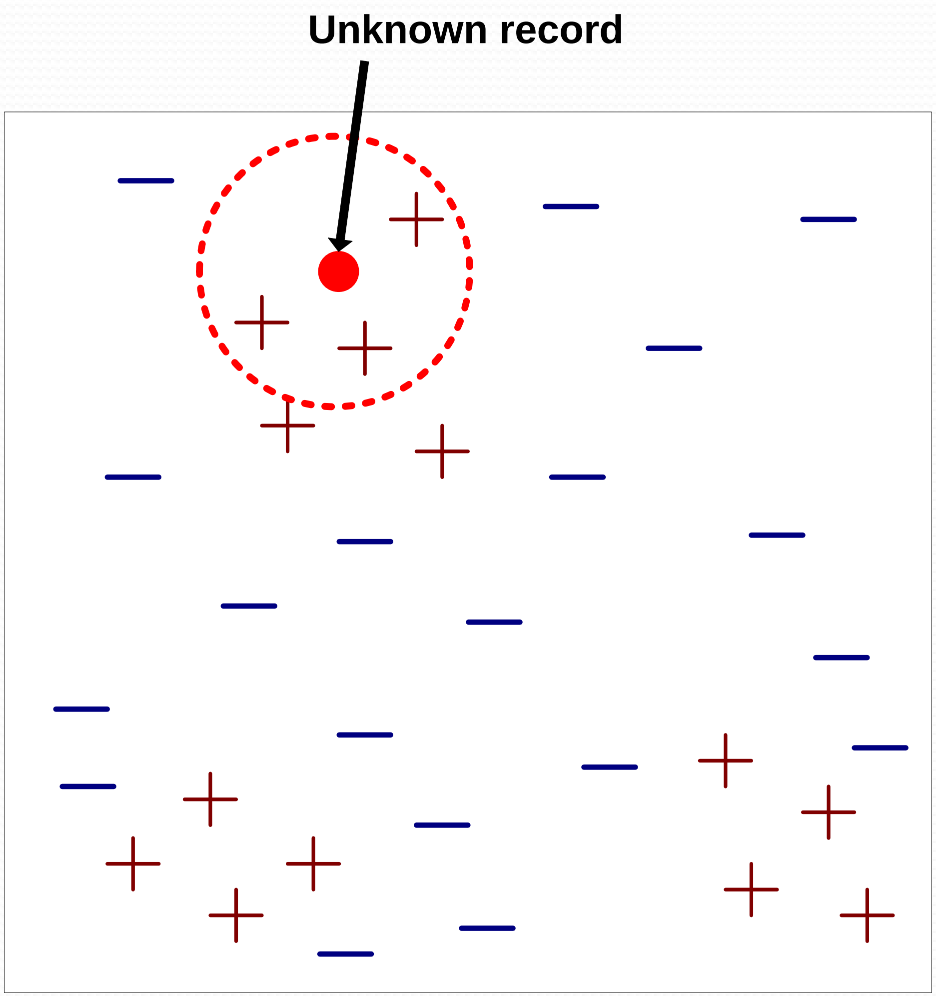
Overview

- Nearest-neighbor classifiers
- Bayesian classifiers
- Support vector machines
- Ensemble methods

Nearest Neighbor Classifiers

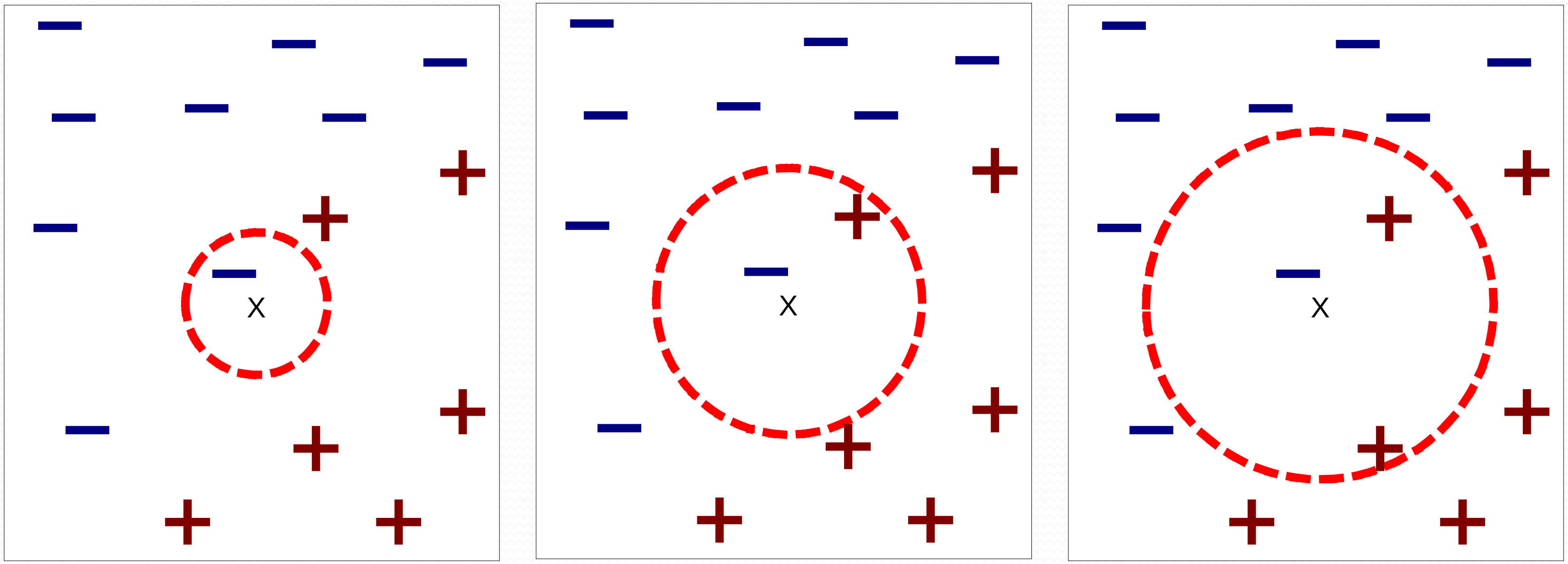
- Basic idea:
  - Given an unlabeled record Y, find the records in the training set that are most similar to Y (the nearest neighbors) to infer the label of Y.

Nearest-Neighbor Classifiers



- Requires three things
  - The set of stored labeled records
  - Distance Metric to compute distance between records
  - The value of *k*, the number of nearest neighbors to retrieve
- To classify an unknown record:
  - Compute distance to other training records
  - Identify *k* nearest neighbors
  - Use class labels of nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)

Definition of Nearest Neighbor



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

K-nearest neighbors of a record x are data points that have the *k* smallest distances to x

Nearest Neighbor Classification

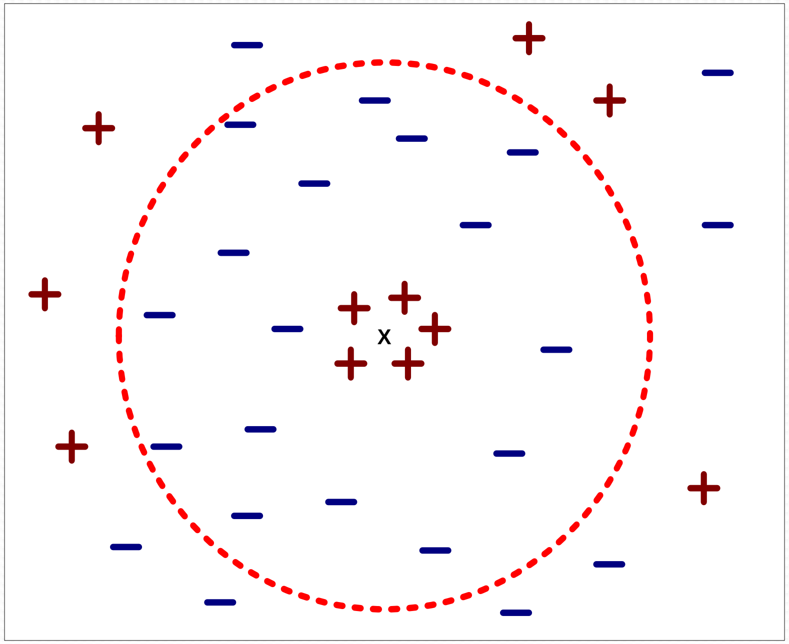
- Compute distance between two points:
  - Euclidean distance

$$d(p,q) = \sqrt{\sum_i (p_i - q_i)^2}$$

- Determine the class from nearest neighbor list
  - take the majority vote of class labels among the k-nearest neighbors
  - We can also weigh the votes according to neighbors' distances
    - weight factor, $w = 1/d^2$
- Attributes have to be normalized.

Nearest Neighbor Classification

- Choosing the value of k:
  - If *k* is too small, sensitive to noise points
  - If *k* is too large, neighborhood may include points from other classes



Nearest Neighbor Classification

- *k*-NN classifiers are *lazy learners*
  - they do not build model explicitly
  - Avoid expensive model-building
  - *K*-NN search could be expensive
  - *K*-NN search is typically assisted by indices.
- Distance-based so it performs poorly in high-dimensional spaces.
- Feature selection is important.
  - E.g., highly-correlated features shouldn't be all included in the distance function.

Bayesian Classifier

- Based on Bayes Theorem:
  - Given a hypothesis/class H and an observation X, denote $P(H|X)$ as the probability that the hypothesis H is true given X happens.
- Example:
  - H = an object O is an apple
  - X = an object O is *red* and *round*
  - $P(H|X)$ = prob. that an object O is an apple given that O is red and round

Bayes Theorem

- Note that we can consider
  - P(H) = probability that an arbitrary object is an apple
  - P(X) = probability that an arbitrary object is red and round
  - P($X|H$) = probability that an object O is red and round given that O is an apple
  - P($H|X$) = probability that an object O is an apple given that O is red and round

Bayes Theorem

- $P(H|X) = P(H,X) / P(X)$
- $P(X|H) = P(H,X) / P(H)$
- $P(H|X) = P(X|H) * P(H) / P(X)$

Applying Bayes theorem to classification

- given an unlabeled record *r*, we consider
  - $P(C_1)$ = probability that a record should be labeled class C_1
  - $P(X)$ = probability that a record has *r*'s attribute values
  - $P(X|C_1)$ = probability that a record has *r*'s attribute values given that the record is labeled C_1
  - $P(C_1|X)$ = probability that a record is labeled C_1 given that it has *r*'s attribute values

Applying Bayes theorem to classification

- suppose there are m class labels: $C_1, C_2, \ldots, C_m$
- we want to determine which class record *r* should belong
- method: compare $P(C_1|X)$, $P(C_2|X)$, …, $P(C_m|X)$ and pick the C_i with the largest probability

Applying Bayes theorem to classification

- Note that:
  - $P(C_1|X) = P(X|C_1) * P(C_1) / P(X)$
  - $P(C_2|X) = P(X|C_2) * P(C_2) / P(X)$
  - $P(C_1|X) > P(C_2|X) \Leftrightarrow$
    $P(X|C_1)P(C_1) > P(X|C_2)P(C_2)$
  - Then, the job is to pick the class C_i with the largest value of $P(X|C_i)P(C_i)$
  - To calculate $P(C_i)$ is easy. Given a training set *D*, we can estimate $P(C_i)$ by n_i/N, where
    - n_i = number of records in *D* of class C_i, and
    - N = total number of records in *D*

Naïve Bayesian Classification

- $P(X|C_i)$, however, is difficult to estimate
- Naïve Bayesian Classification assumes that the values of the attributes are conditionally independent of one another.
- That is,

$$\boxed{P(X \mid C_i) = \prod_{k=1}^{n} P(x_k \mid C_i)}$$

x_k = value of a record r for attribute k

Naïve Bayesian Classification

- If Attribute *k* is categorical (e.g., nominal, ordinal), then $P(x_k|C_i)$ can be estimated by

$$\boxed{n_{ik} / n_i}$$

where n_{ik} = number of records in the dataset that are of class C_i and whose values for attribute *k* is x_k

Record id	Age	Income	Student	Credit-rating	Own-computer
1	< 30	High	No	Bad	No
2	< 30	High	No	Good	No
3	30 .. 40	High	No	Bad	Yes
4	> 40	Medium	No	Bad	Yes
5	>40	Low	Yes	Bad	Yes
6	> 40	Low	Yes	Good	No
7	30 .. 40	Low	Yes	Good	Yes
8	< 30	Medium	No	Bad	No
9	< 30	Low	Yes	Bad	Yes
10	> 40	Medium	Yes	Bad	Yes
11	< 30	Medium	Yes	Good	Yes
12	30 .. 40	Medium	No	Good	Yes
13	30 .. 40	High	Yes	Bad	Yes
14	> 40	Medium	No	Good	No

Example

- Given a record *X*:

Age	*Income*	*Student*	*Credit-rating*
< 30	medium	yes	fair

is *X* a computer-owner or not?

Example

- 2 classes:
  - C_1 = O.C. = yes; $P(C_1) = 9/14$
  - C_2 = O.C. = no; $P(C_2) = 5/14$
- 4 attributes:
  - $P(\text{Age} < 30 \mid C_1) = 2/9$
  - $P(\text{Income} = \text{medium} \mid C_1) = 4/9$
  - $P(\text{Student} = \text{yes} \mid C_1) = 6/9$
  - $P(\text{C.R} = \text{fair} \mid C_1) = 6/9$
- Hence,
  - $P(X|C_1) = (2/9)(4/9)(6/9)(6/9) = 0.044$

Example

- Similarly, we have:
  - $P(X|C_2) = 0.019$
- Therefore,
  - $P(X|C_1)P(C_1) = (9/14) * 0.044 = 0.028$
  - $P(X|C_2)P(C_2) = (5/14) * 0.019 = 0.007$
- X is classified as C_1, or X is a computer-owner

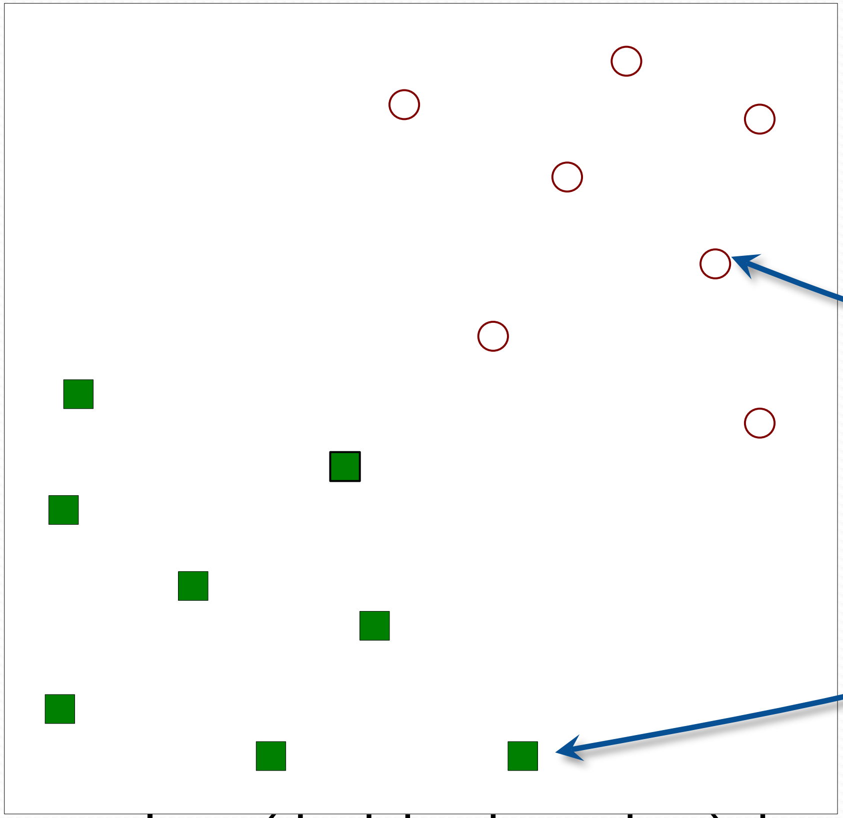
What about numerical attributes?

- For numerical attributes:
  - Discretize the range into bins
    - replace by ordinal attribute
    - result sensitive to discretization
  - Probability density estimation:
    - Assume attribute follows a normal distribution
    - Use data to estimate parameters of distribution (e.g., mean and standard deviation)
    - Once probability distribution is known, can use it to estimate the conditional *probability density* $P'(x_k|C_i)$
    - Compare classes based on their probability densities.

Naïve Bayes (Summary)

- Robust to isolated noise points
- Robust to noisy attributes that are uncorrelated to class
- Independence assumption may not hold for some attributes
  - Use other techniques such as Bayesian Belief Networks (BBN) to capture attributes correlation

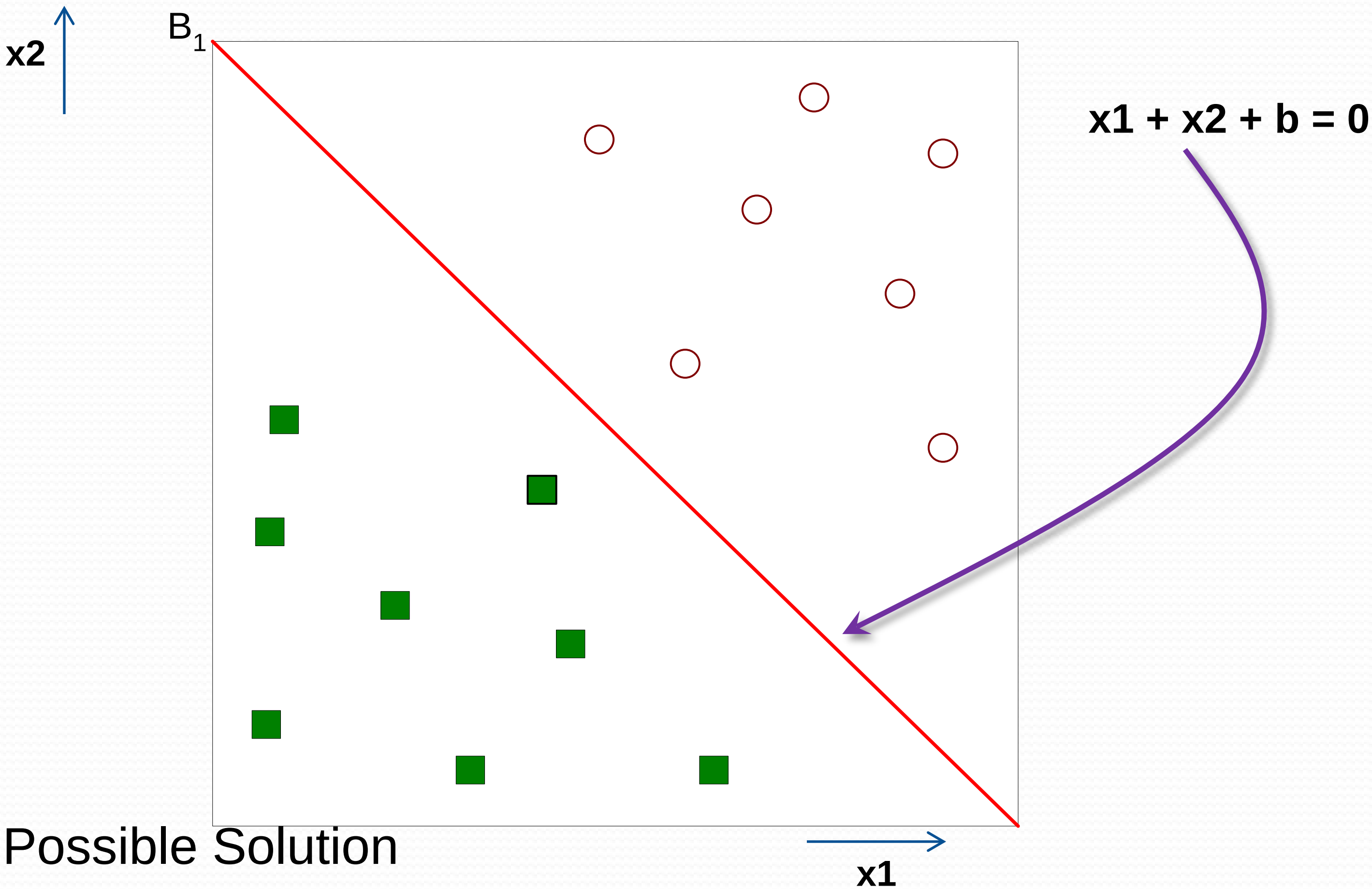
Support Vector Machines



Data of 2 classes

- Find a linear hyperplane (decision boundary) that will separate the data

Support Vector Machines

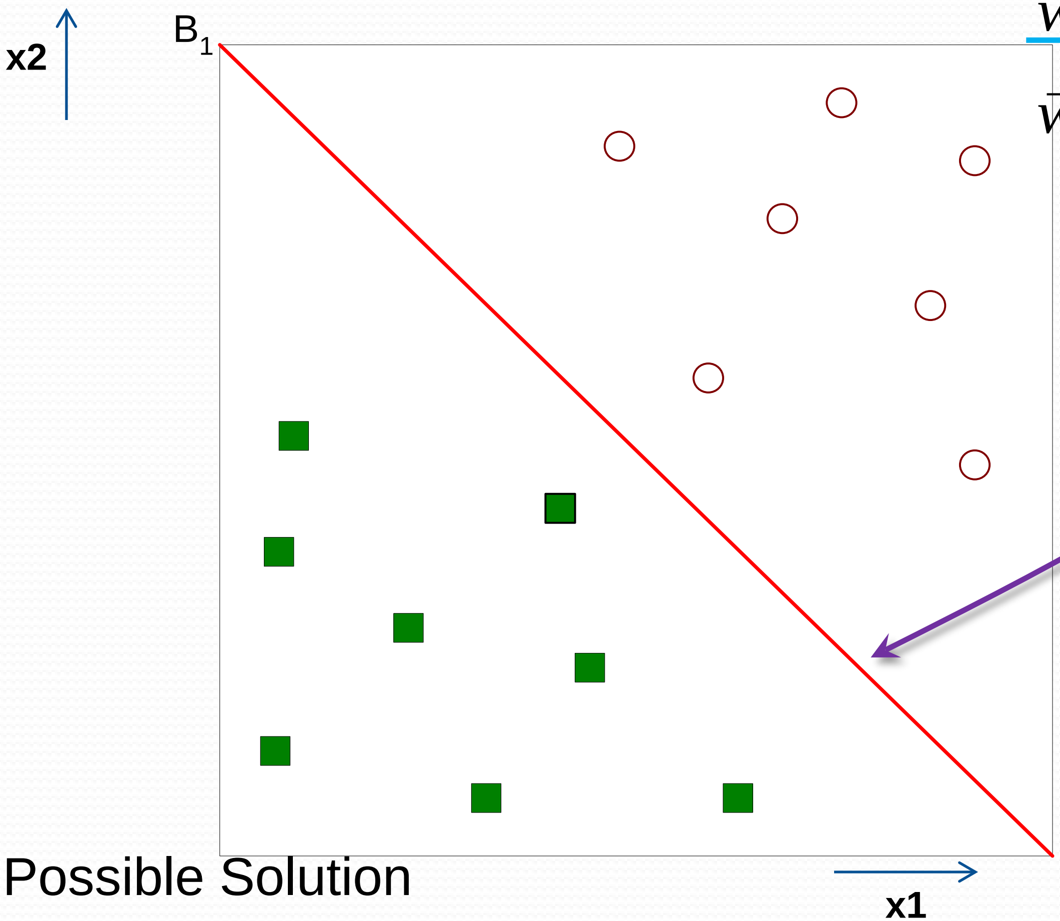


- One Possible Solution

Support Vector Machines

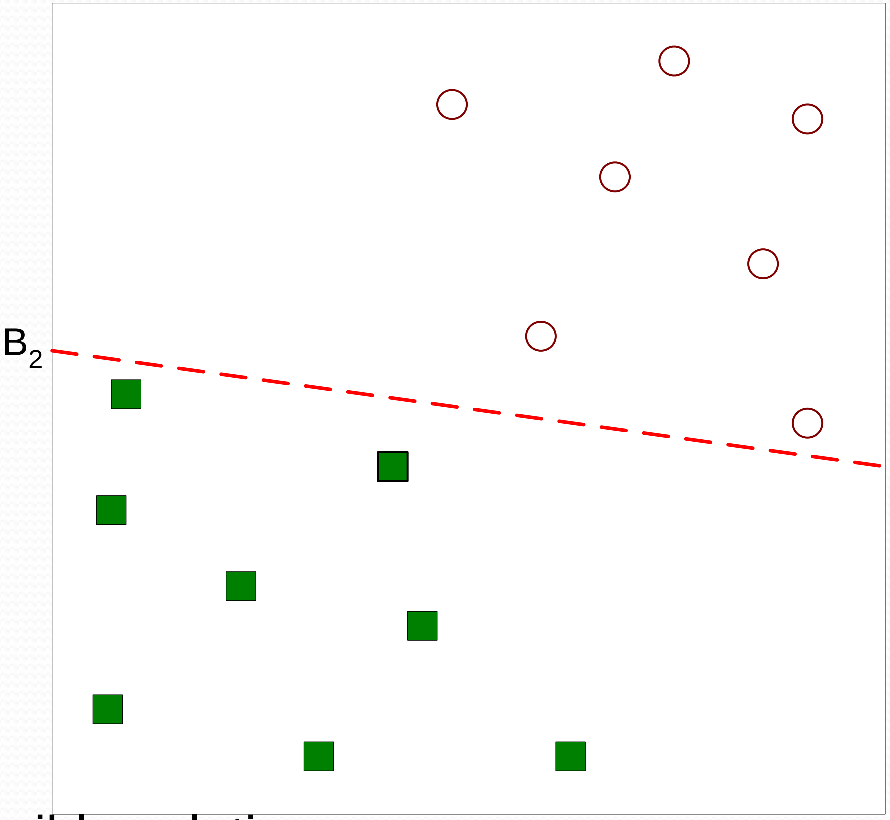
$$\vec{w}\cdot\vec{x}+b=0$$

$$\vec{w}=(1,1);\ \vec{x}=(x1,x2)^T$$



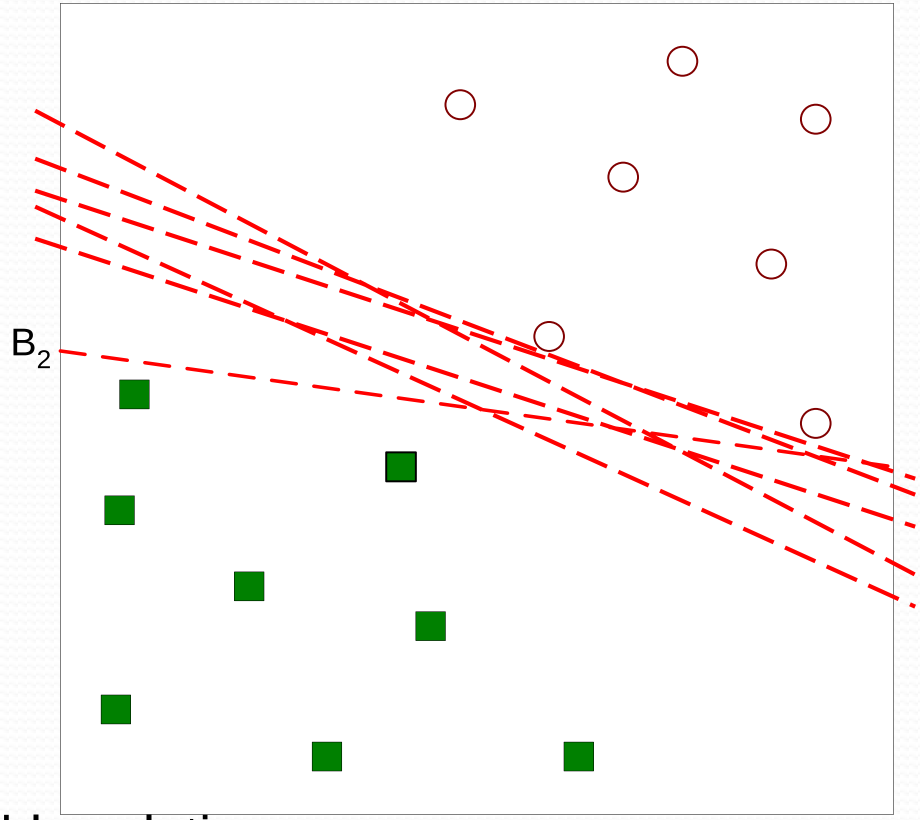
- One Possible Solution

Support Vector Machines



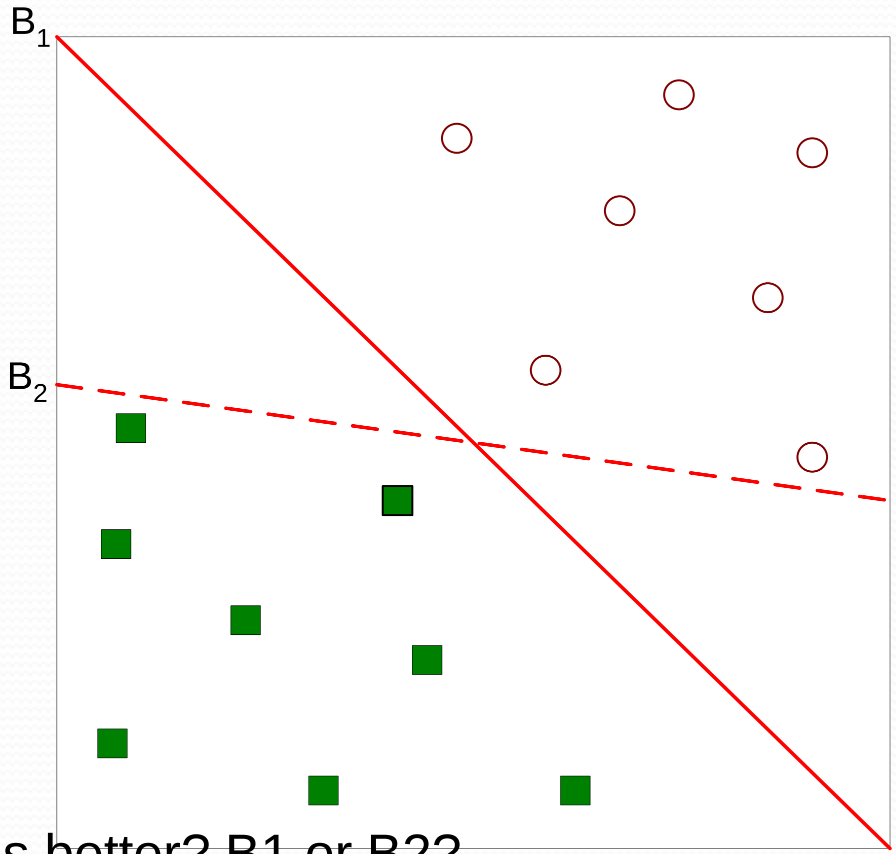
- Another possible solution

Support Vector Machines



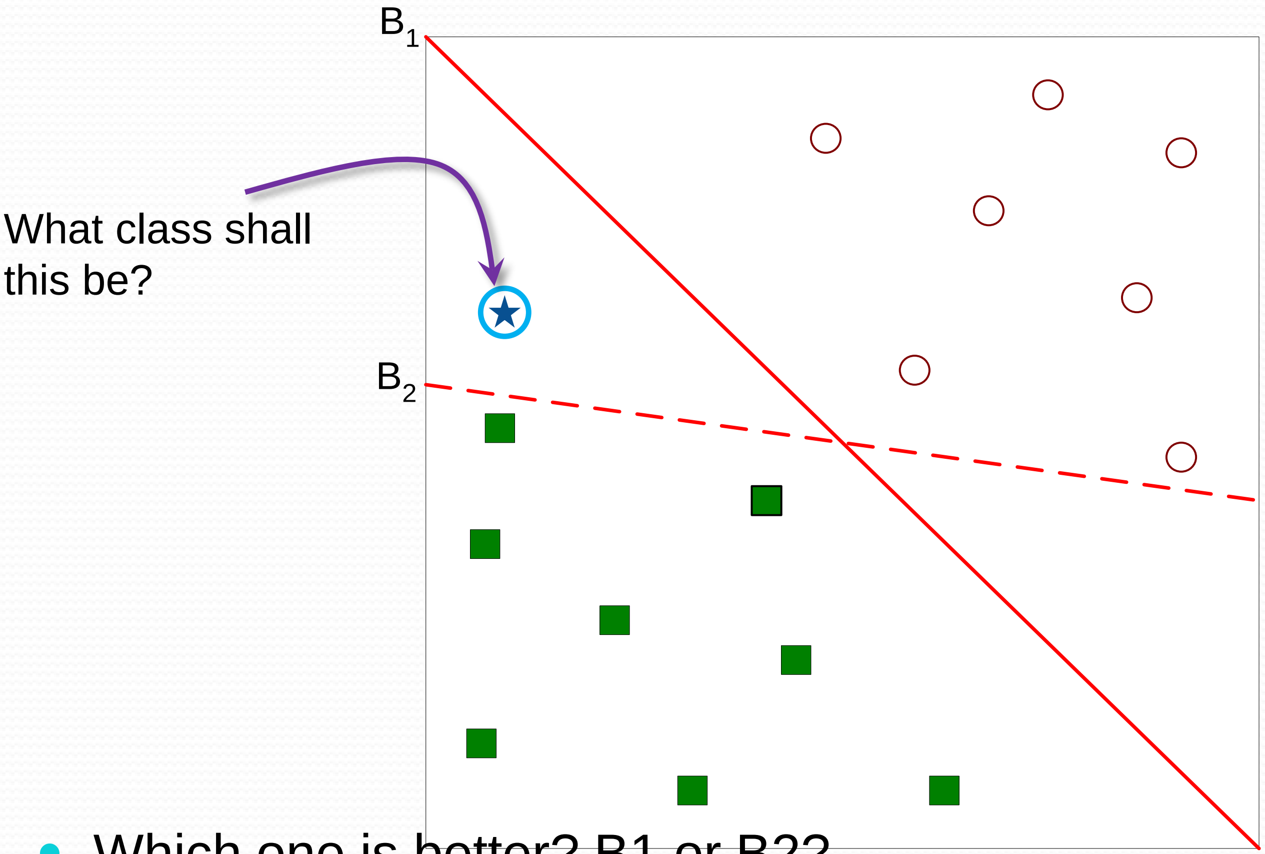
- Other possible solutions

Support Vector Machines



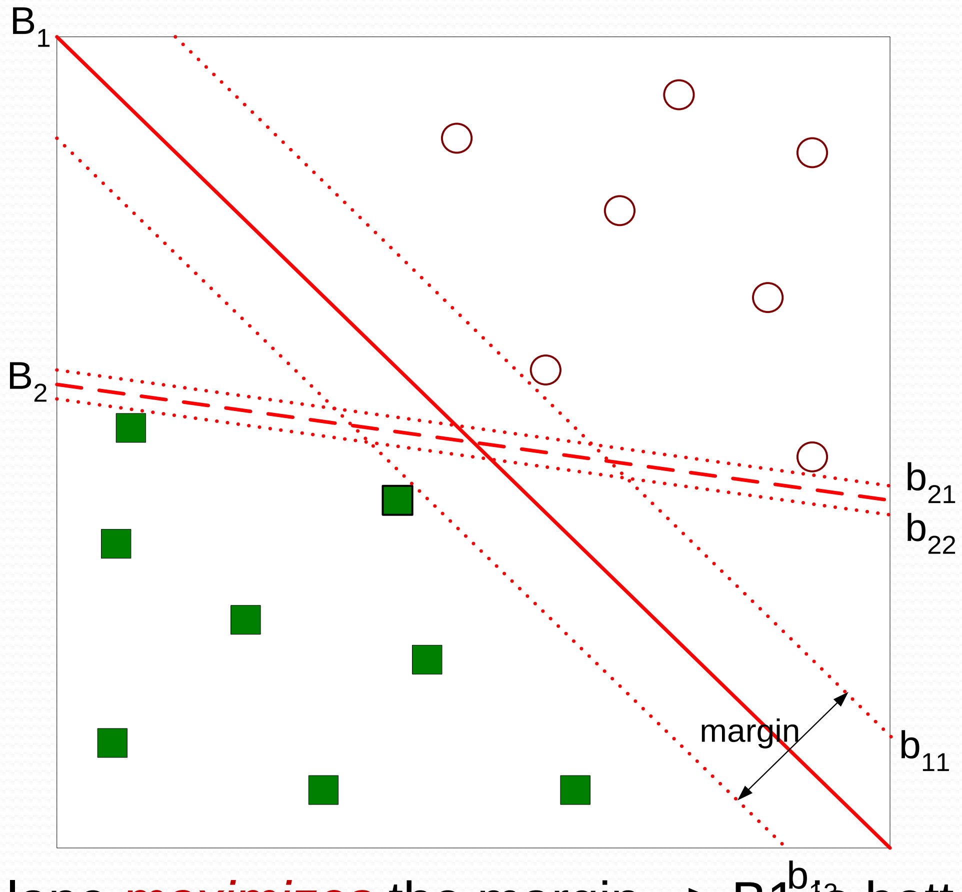
- Which one is better? B1 or B2?
- How do you define better?

Support Vector Machines



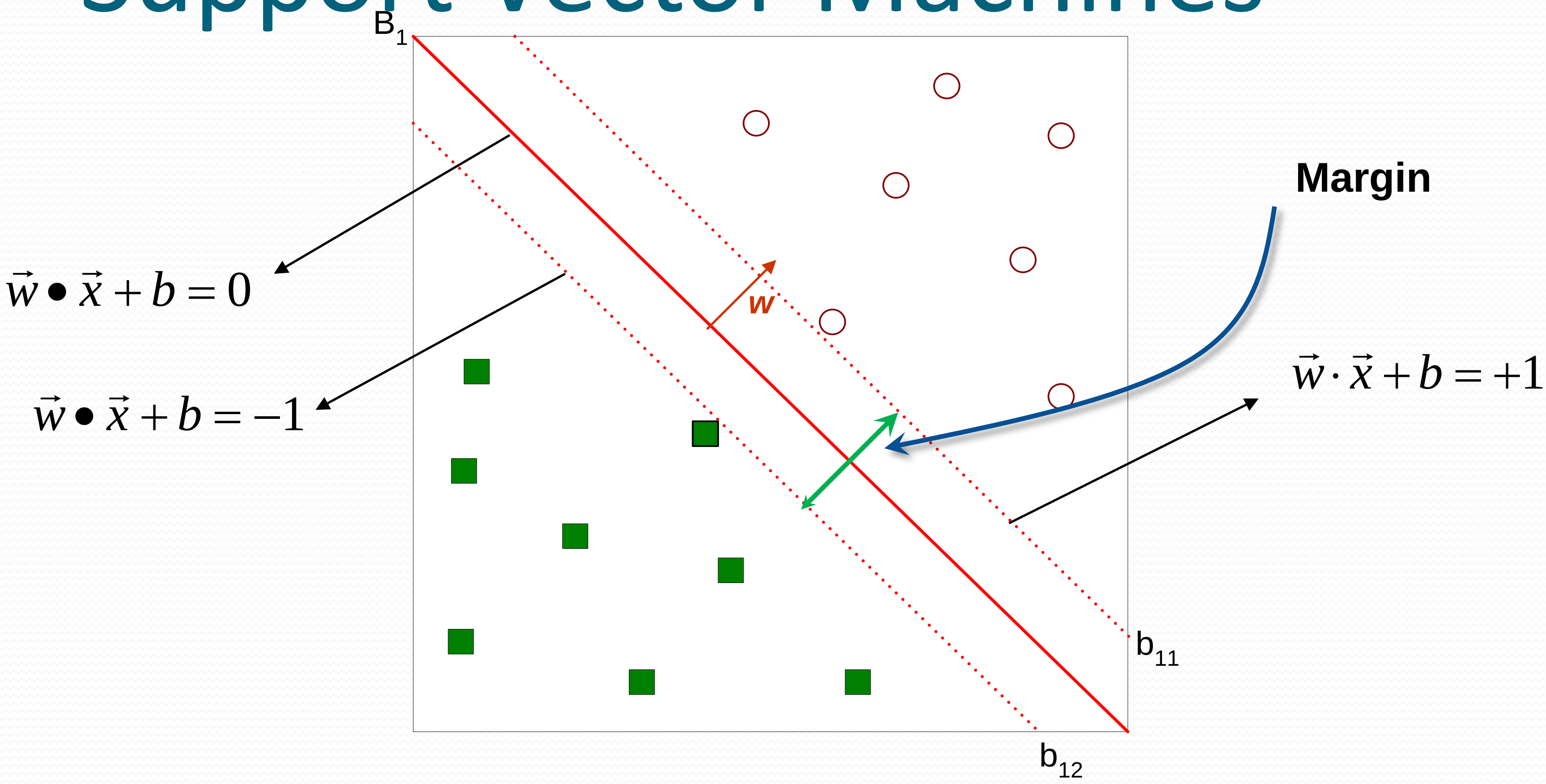
- Which one is better? B1 or B2?
- How do you define better?

Support Vector Machines

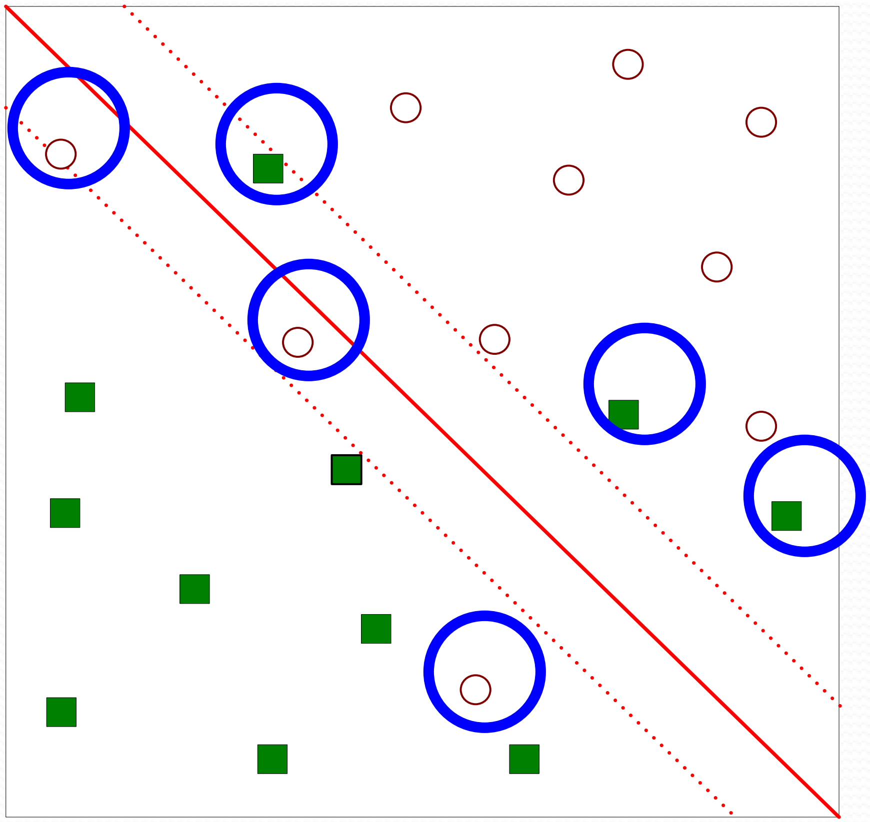


- Find hyperplane *maximizes* the margin => B1 is better than B2

Support Vector Machines

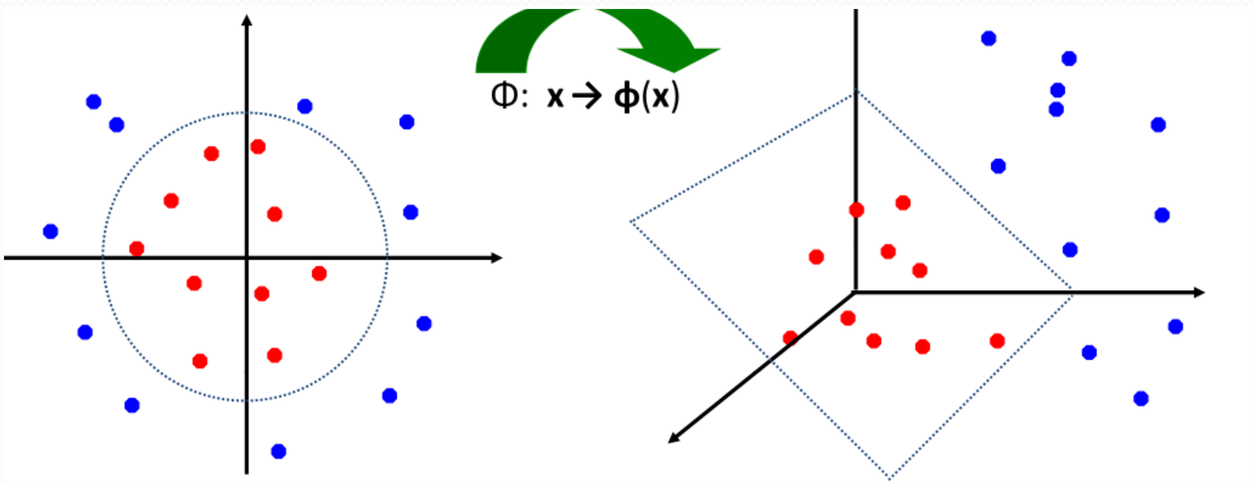


What if the problem is not linearly separable?



No straight line can separate the examples into their classes

What if the problem is not linearly separable?

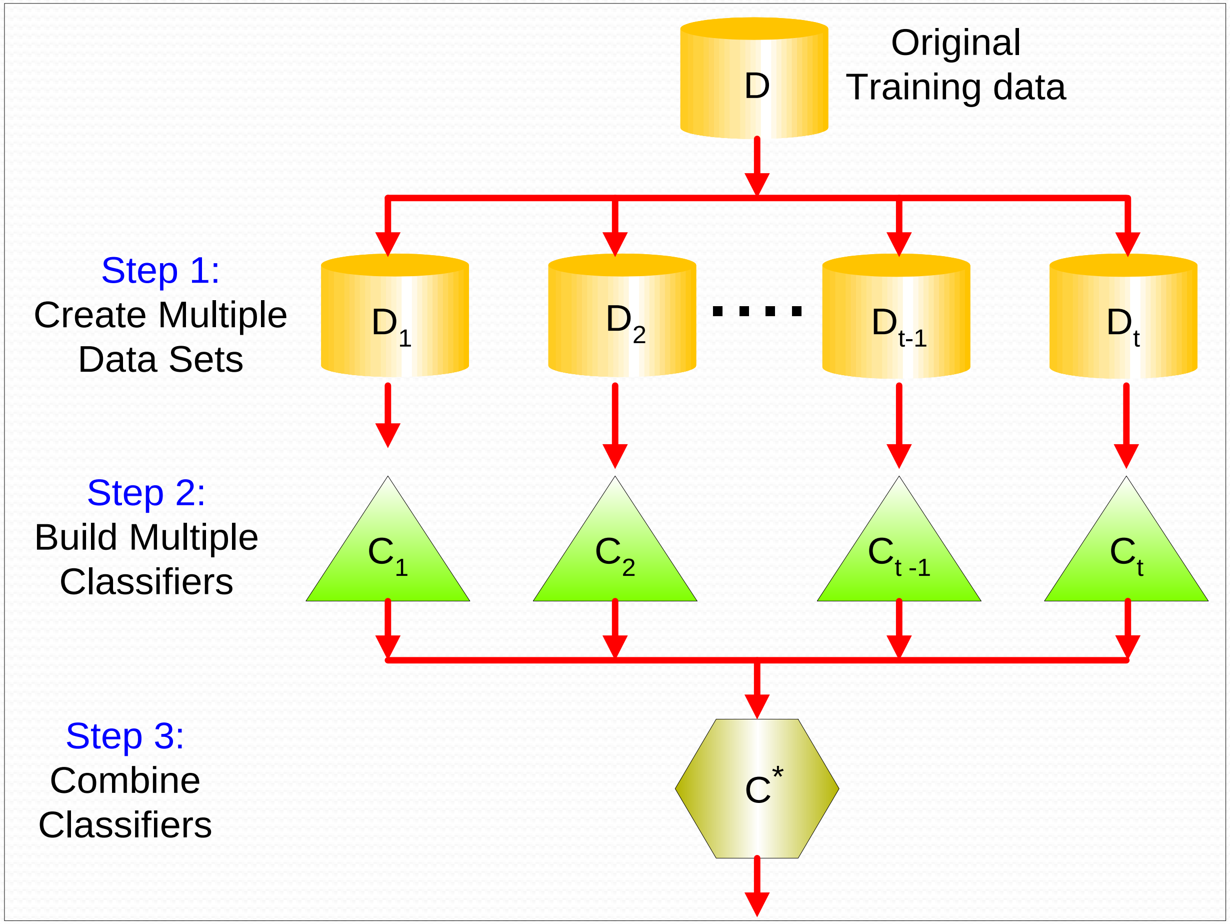


- $\mathbf{x} = [x_1, x_2]^t$
- $\phi([x_1, x_2]^t) = [1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1 x_2]^t$

Ensemble Methods

- Construct a set of classifiers from the training data
- Predict class label of previously unseen records by aggregating predictions made by multiple classifiers
  - voting

General Idea



Why does it work?

- Suppose there are 25 base classifiers
  - Each classifier has error rate, ε = 0.35
  - Assume classifiers are *independent*
  - Probability that the ensemble classifier makes a wrong prediction:

$$\sum_{i=13}^{25} \binom{25}{i} \varepsilon^{i} (1-\varepsilon)^{25-i} = 0.06$$

Random Forest

Random Forest:

- Each classifier in the ensemble is a *decision tree* classifier and is generated using a random selection of attributes at each node to determine the split
- During classification, each tree votes and the most popular class is returned

Two Methods to construct Random Forest:

- Forest-**RI** (***random input** selection*): Randomly select, at each node, F attributes as candidates for the split at the node**.**
- Forest-**RC** (***random linear combinations***)*:* Creates new attributes (or features) that are a linear combination of the existing attributes (reduces the correlation between individual classifiers)

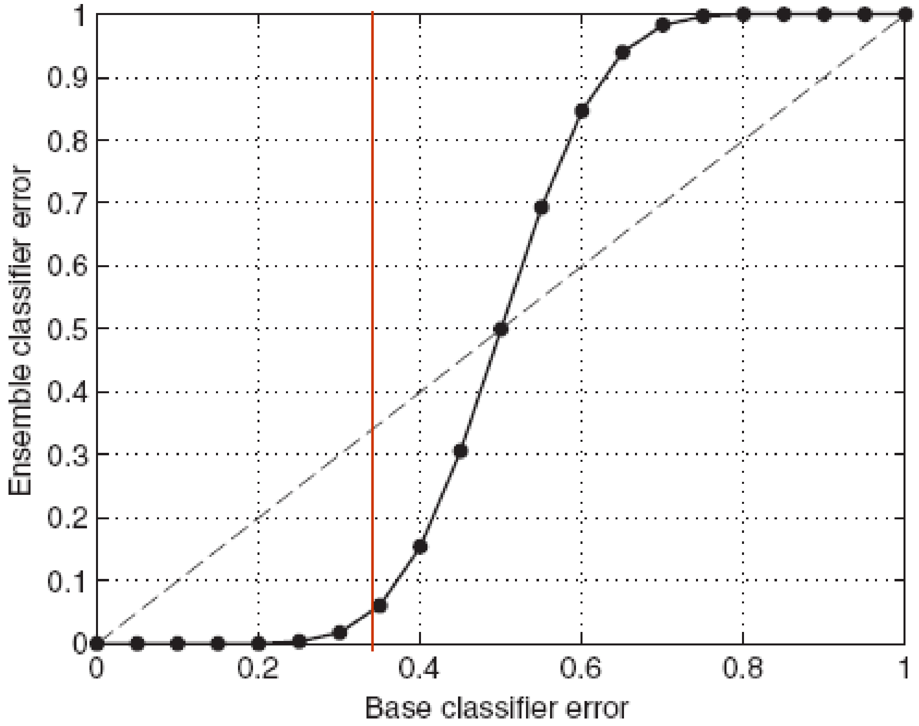


Figure 5.30. Comparison between errors of base classifiers and errors of the ensemble classifier.

Examples of Ensemble Methods

- How to generate an ensemble of classifiers?
  - use different training sets
  - use different attribute sets for input
  - use different partitions of class labels
  - use different learning algorithms