Language Processing for Different Domains and Genres: Machine Learning Introduction

Caroline Sporleder, Ines Rehbein

Universität des Saarlandes

Wintersemester 2009/10

5.11.2009

Machine Learning Basics

Goal:

develop computer programs that automatically improve with experience by learning from representative input (and output) data

Motivation:

- for many problems, the best way of computing the correct output from the input is not known.
- manually determining input output rules by (informed) trial and error is time consuming and typically results in low coverage (but high precision)

Example:

predicting the plural form of a German noun

Example: German Plural Formation

Nine possibilities:

1. no ending, no umlaut: *das Zimmer - die Zimmer* (rooms)
2. no ending, but umlaut: *der Faden - die Fäden* (thread)
3. -e: *der Hund - die Hunde* (dogs)
4. -e plus umlaut: *der Stuhl - die Stühle* (chairs)
5. -er: *das Kind - die Kinder* (children)
6. -er plus umlaut: *das Lamm - die Lämmer* (lambs)
7. -n: *die Straße - die Straßen* (streets)
8. -en: *die Bank - die Banken* (banks)
9. -s: *das Trio - die Trios* (trios)

Hand-crafting Rules

Hypothesis 1: ending is determined by noun's grammatical gender

Hand-crafting Rules

Hypothesis 1: ending is determined by noun's grammatical gender

Rule 1:

- masculine $\Rightarrow$ -e
- neuter $\Rightarrow$ -er
- feminine $\Rightarrow$ -en

Hand-crafting Rules

Hypothesis 1: ending is determined by noun's grammatical gender

Rule 1:

- masculine $\Rightarrow$ -e
- neuter $\Rightarrow$ -er
- feminine $\Rightarrow$ -en

Applying Rule 1:

Hand-crafting Rules

Hypothesis 1: ending is determined by noun's grammatical gender

Rule 1:

- masculine $\Rightarrow$ -e
- neuter $\Rightarrow$ -er
- feminine $\Rightarrow$ -en

Applying Rule 1:

- *das Kind* (n)

Hand-crafting Rules

Hypothesis 1: ending is determined by noun's grammatical gender

Rule 1:

- masculine $\Rightarrow$ -e
- neuter $\Rightarrow$ -er
- feminine $\Rightarrow$ -en

Applying Rule 1:

- *das Kind* (n) $\Rightarrow$ *die Kinder*

Hand-crafting Rules

Hypothesis 1: ending is determined by noun's grammatical gender

Rule 1:

- masculine $\Rightarrow$ -e
- neuter $\Rightarrow$ -er
- feminine $\Rightarrow$ -en

Applying Rule 1:

- *das Kind* (n) $\Rightarrow$ *die Kinder*
- *der Hund* (m)

Hand-crafting Rules

Hypothesis 1: ending is determined by noun's grammatical gender

Rule 1:

- masculine $\Rightarrow$ -e
- neuter $\Rightarrow$ -er
- feminine $\Rightarrow$ -en

Applying Rule 1:

- *das Kind* (n) $\Rightarrow$ *die Kinder*
- *der Hund* (m) $\Rightarrow$ *die Hunde*

Hand-crafting Rules

Hypothesis 1: ending is determined by noun's grammatical gender

Rule 1:

- masculine $\Rightarrow$ -e
- neuter $\Rightarrow$ -er
- feminine $\Rightarrow$ -en

Applying Rule 1:

- *das Kind* (n) $\Rightarrow$ *die Kinder*
- *der Hund* (m) $\Rightarrow$ *die Hunde*
- *die Bank* (f)

Hand-crafting Rules

Hypothesis 1: ending is determined by noun's grammatical gender

Rule 1:

- masculine $\Rightarrow$ -e
- neuter $\Rightarrow$ -er
- feminine $\Rightarrow$ -en

Applying Rule 1:

- *das Kind* (n) $\Rightarrow$ *die Kinder*
- *der Hund* (m) $\Rightarrow$ *die Hunde*
- *die Bank* (f) $\Rightarrow$ *die Banken*

Hand-crafting Rules

Hypothesis 1: ending is determined by noun's grammatical gender

Rule 1:

- masculine $\Rightarrow$ -e
- neuter $\Rightarrow$ -er
- feminine $\Rightarrow$ -en

Applying Rule 1:

- *das Kind* (n) $\Rightarrow$ *die Kinder*
- *der Hund* (m) $\Rightarrow$ *die Hunde*
- *die Bank* (f) $\Rightarrow$ *die Banken*
- *das Zimmer* (n)

Hand-crafting Rules

Hypothesis 1: ending is determined by noun's grammatical gender

Rule 1:

- masculine $\Rightarrow$ -e
- neuter $\Rightarrow$ -er
- feminine $\Rightarrow$ -en

Applying Rule 1:

- *das Kind* (n) $\Rightarrow$ *die Kinder*
- *der Hund* (m) $\Rightarrow$ *die Hunde*
- *die Bank* (f) $\Rightarrow$ *die Banken*
- *das Zimmer* (n) $\Rightarrow$ *die Zimmerer*

Hand-crafting Rules

Hypothesis 1: ending is determined by noun's grammatical gender

Rule 1:

- masculine ⇒ -e
- neuter ⇒ -er
- feminine ⇒ -en

Applying Rule 1:

- *das Kind* (n) ⇒ *die Kinder*
- *der Hund* (m) ⇒ *die Hunde*
- *die Bank* (f) ⇒ *die Banken*
- *das Zimmer* (n) ⇒ **die Zimmerer (die Zimmer)*

Hand-crafting Rules (2)

Hypothesis 2: morpho-phonological form also influences ending

Hand-crafting Rules (2)

Hypothesis 2: morpho-phonological form also influences ending

Rule 2: don't add ending if noun already ends in -e, -en, or -er

Hand-crafting Rules (2)

Hypothesis 2: morpho-phonological form also influences ending

Rule 2: don't add ending if noun already ends in -e, -en, or -er

Applying Rules 1 and 2:

Hand-crafting Rules (2)

Hypothesis 2: morpho-phonological form also influences ending

Rule 2: don't add ending if noun already ends in -e, -en, or -er

Applying Rules 1 and 2:

- *das Zimmer* (n)

Hand-crafting Rules (2)

Hypothesis 2: morpho-phonological form also influences ending

Rule 2: don't add ending if noun already ends in -e, -en, or -er

Applying Rules 1 and 2:

- *das Zimmer* (n) $\Rightarrow$ die Zimmer

Hand-crafting Rules (2)

Hypothesis 2: morpho-phonological form also influences ending

Rule 2: don't add ending if noun already ends in -e, -en, or -er

Applying Rules 1 and 2:

- *das Zimmer* (n) $\Rightarrow$ die Zimmer
- *die Ampel* (f)

Hand-crafting Rules (2)

Hypothesis 2: morpho-phonological form also influences ending

Rule 2: don't add ending if noun already ends in -e, -en, or -er

Applying Rules 1 and 2:

- *das Zimmer* (n) $\Rightarrow$ die Zimmer
- *die Ampel* (f) $\Rightarrow$ *die Ampelen*

Hand-crafting Rules (2)

Hypothesis 2: morpho-phonological form also influences ending

Rule 2: don't add ending if noun already ends in -e, -en, or -er

Applying Rules 1 and 2:

- *das Zimmer* (n) $\Rightarrow$ die Zimmer
- *die Ampel* (f) $\Rightarrow$ **die Ampelen (die Ampeln)*

Hand-crafting Rules (3)

Rule 3: -e and -en become ø and -n if the last syllable of the singular contains a schwa.

Hand-crafting Rules (3)

Rule 3: -e and -en become ø and -n if the last syllable of the singular contains a schwa.

Applying Rules 1, 2 and 3:

Hand-crafting Rules (3)

Rule 3: -e and -en become ø and -n if the last syllable of the singular contains a schwa.

Applying Rules 1, 2 and 3:

- *die Ampel* (f)

Hand-crafting Rules (3)

Rule 3: -e and -en become ø and -n if the last syllable of the singular contains a schwa.

Applying Rules 1, 2 and 3:

- *die Ampel* (f) $\Rightarrow$ *die Ampeln*

Hand-crafting Rules (3)

Rule 3: -e and -en become ø and -n if the last syllable of the singular contains a schwa.

Applying Rules 1, 2 and 3:

- *die Ampel* (f) $\Rightarrow$ *die Ampeln*
- *der Nachbar* (m)

Hand-crafting Rules (3)

Rule 3: -e and -en become ø and -n if the last syllable of the singular contains a schwa.

Applying Rules 1, 2 and 3:

- *die Ampel* (f) ⇒ *die Ampeln*
- *der Nachbar* (m) ⇒ *die Nachbare*

Hand-crafting Rules (3)

Rule 3: -e and -en become ø and -n if the last syllable of the singular contains a schwa.

Applying Rules 1, 2 and 3:

- *die Ampel* (f) ⇒ *die Ampeln*
- *der Nachbar* (m) ⇒ **die Nachbare (die Nachbarn)*

Machine Learning German Plural Formation

Learn from input-output pairs:

- Zimmer, Zimmer
- Faden, Fäden
- Hund, Hunde
- Stuhl, Stühle
- Kind, Kinder
- Lamm, Lämmer
- Straße, Straßen
- Bank, Banken
- Trio, Trios
- Ampel, Ampeln
- Nachbar, Nachbarn
- Maus, Mäuse

$\Rightarrow$ Input is typically represented as a feature vector.

Machine Learning German Plural Formation (2)

What information needs to be represented for the task to be learnable (i.e., which features need to be modelled)?

Machine Learning German Plural Formation (2)

What information needs to be represented for the task to be learnable (i.e., which features need to be modelled)?

1. Zimmer: <n>, Zimmer
2. Faden: <m>, Fäden
3. Hund: <m>, Hunde
4. Stuhl: < m>, Stühle
5. Kind: <n>, Kinder
6. Lamm: <n>, Lämmer
7. Straße: <f>, Straßen
8. Bank: <f>, Banken
9. Trio: <n>, Trios
10. Ampel: <f>, Ampeln
11. Nachbar: <m>, Nachbarn
12. Maus: <f>, Mäuse

Machine Learning German Plural Formation (2)

What information needs to be represented for the task to be learnable (i.e., which features need to be modelled)?

1. Zimmer: <n, er>, Zimmer
2. Faden: <m, en>, Fäden
3. Hund: <m, ø>, Hunde
4. Stuhl: <m, ø>, Stühle
5. Kind: <n, ø>, Kinder
6. Lamm: <n, ø>, Lämmer
7. Straße: <f, e>, Straßen
8. Bank: <f, ø>, Banken
9. Trio: <n, ø>, Trios
10. Ampel: <f, ø>, Ampeln
11. Nachbar: <m, ø>, Nachbarn
12. Maus: <f, ø>, Mäuse

Machine Learning German Plural Formation (2)

What information needs to be represented for the task to be learnable (i.e., which features need to be modelled)?

1. Zimmer: <n, er, a-schwa>, Zimmer
2. Faden: <m, en, e-schwa>, Fäden
3. Hund: <m, ø, no schwa>, Hunde
4. Stuhl: <m, ø, no schwa>, Stühle
5. Kind: <n, ø, no schwa>, Kinder
6. Lamm: <n, ø, no schwa>, Lämmer
7. Straße: <f, e, e-schwa>, Straßen
8. Bank: <f, ø, no schwa>, Banken
9. Trio: <n, ø, no schwa>, Trios
10. Ampel: <f, ø, e-schwa>, Ampeln
11. Nachbar: < m, ø, no schwa>, Nachbarn
12. Maus: <f, ø, no schwa>, Mäuse

Basic Terminology

instance: one input-output pair, where the input is a feature vector
Hund, m, ø, no schwa, Hunde

label or class to be predicted: the output value
Hunde
label can be nominal, numeric, binary etc.

features: types of information encoded in the input
singular form, gender, ending of singular, schwa information

feature values: the values the features assume (for a given instance)
Hund, m, ø, no schwa
values can be nominal, numeric, binary etc.

Basic Terminology (2)

training set: set of manually labelled instances from which target function (mapping from input to output) is learnt

test set: set of instances to which the trained machine learner is applied, test set labels are used to compute performance of classifier (labels are not known to the classifier)

development set: set of instances used to choose the best parameter settings (typically those that optimise performance on the development set)

Basic Types of Machine Learning

Supervised Machine Learning:

system learns target function (mapping from input to output) from a labelled training set (decision tree learners, Naive Bayes, k-NN etc.)

Unsupervised Machine Learning:

no labelled training set, system searches for best model to account for unlabelled test set (clustering etc.)

Semi-Supervised Machine Learning:

combination of supervised and unsupervised; training set consists of a small labelled seed set and a large unlabelled set

Machine Learning Spaces

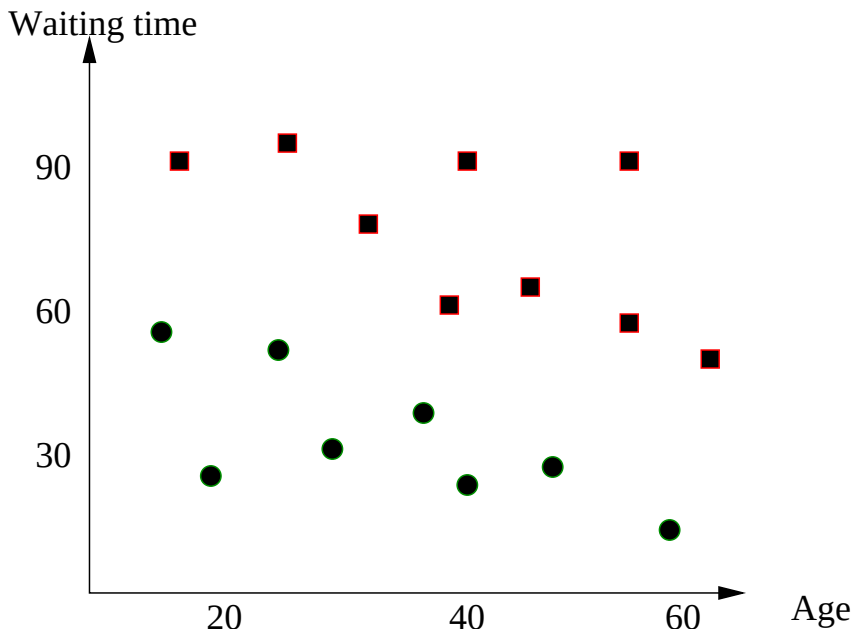
Instances are points (vectors) in n-dimensional space, where n is the number of features

Example: predict customer satisfaction (*happy* vs. *not happy*) of the clients of a call centre from (i) their age and (ii) the number of minutes they were kept in the waiting loop.

Machine Learning Spaces

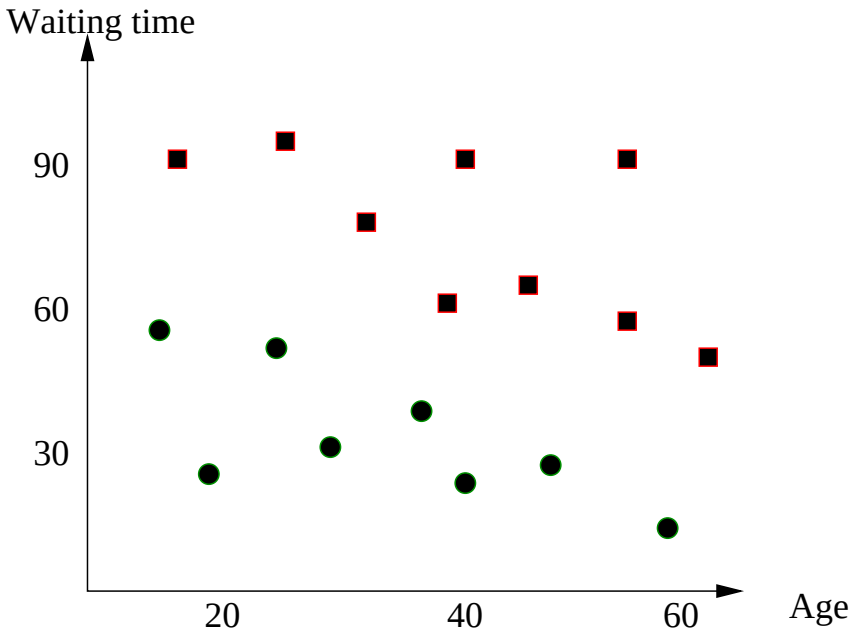
Instances are points (vectors) in n-dimensional space, where *n* is the number of features

Example: predict customer satisfaction (*happy* vs. *not happy*) of the clients of a call centre from (i) their age and (ii) the number of minutes they were kept in the waiting loop.



Supervised Machine Learning (1)

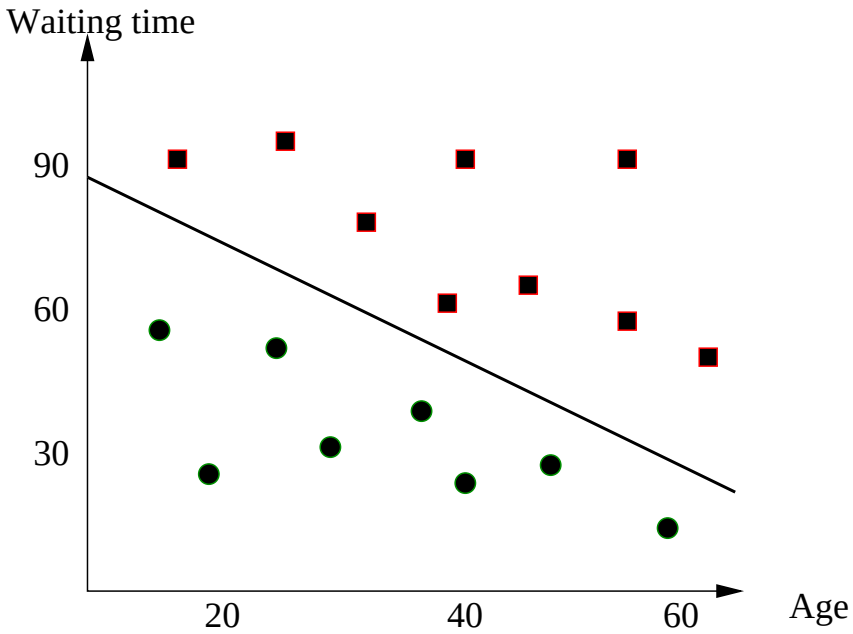
Classifier uses training set to try to find the correct decision boundary between the classes.



Training set

Supervised Machine Learning (1)

Classifier uses training set to try to find the correct decision boundary between the classes.



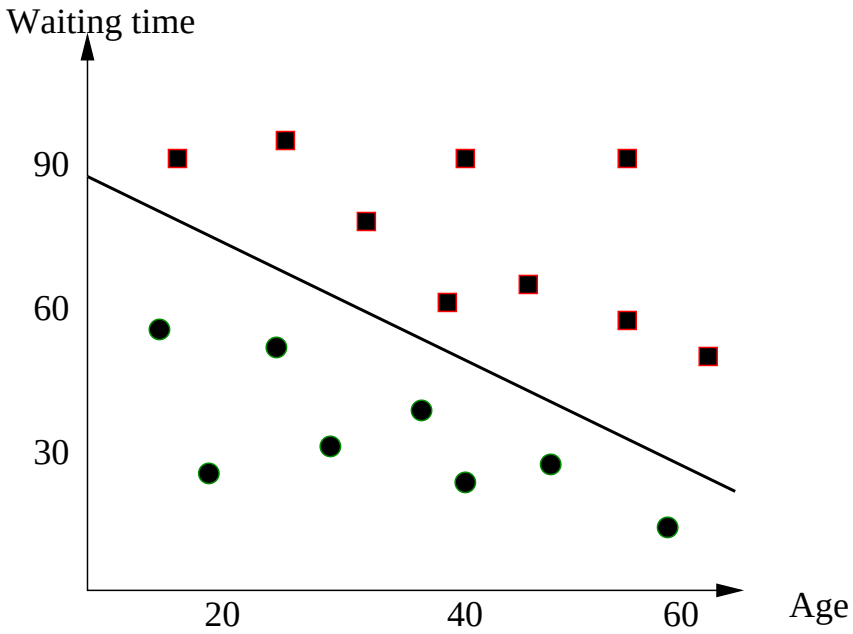
Training set

Separability

Whether and how two sets can be separated depends on

- the machine learning algorithm (linear or non-linear)
- the dimensionality of the space (the number of features) and the suitability of the features

Data sets which can be separated by a line (2-d), plane (3-d), or hyperplane (higher dimensional space) are called linearly separable.



Generalisation (1)

All machine learners generalise (opposite rote learning)

If they didn't generalise, they wouldn't be able to

Generalisation (1)

All machine learners generalise (opposite rote learning)

If they didn't generalise, they wouldn't be able to

- label unseen instances
- ignore outliers (e.g., mislabelled instances in the training set)
  $\Rightarrow$ overfitting

Generalisation (1)

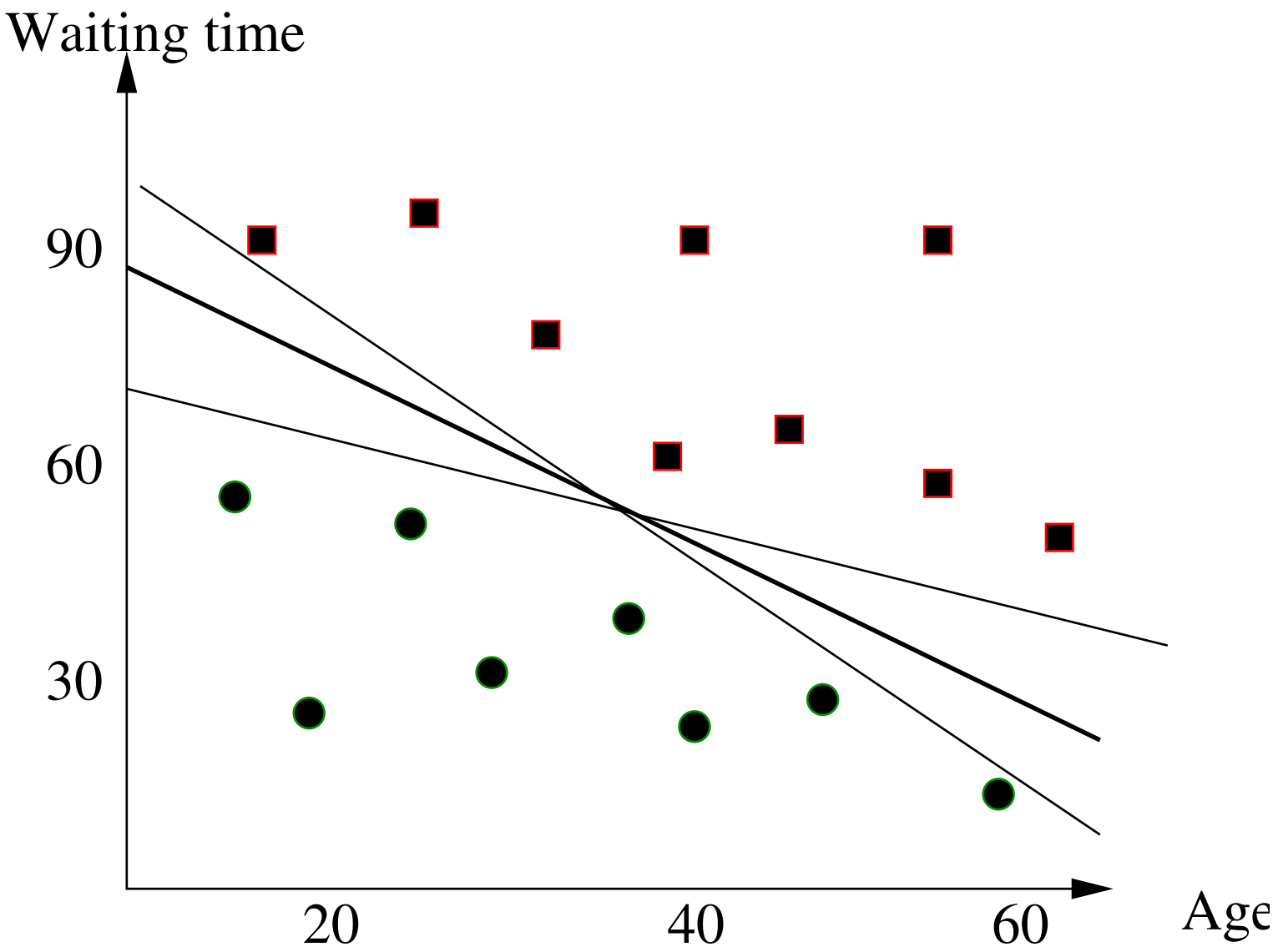
All machine learners generalise (opposite rote learning)

If they didn't generalise, they wouldn't be able to

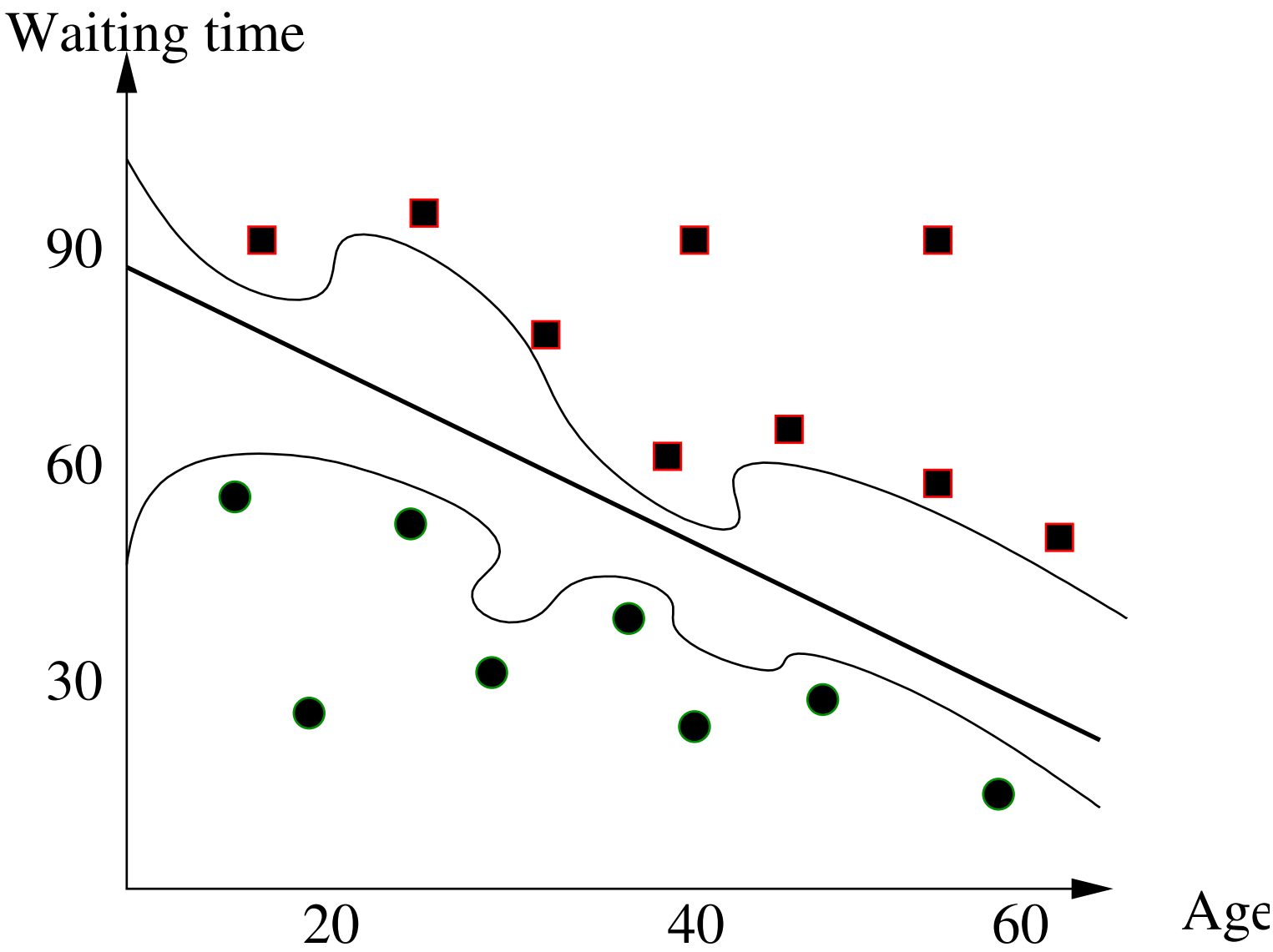
- label unseen instances
- ignore outliers (e.g., mislabelled instances in the training set)
  $\Rightarrow$ overfitting

However: different machine learners generalise in different ways (inductive bias)

Generalisation (2)



Generalisation (2)

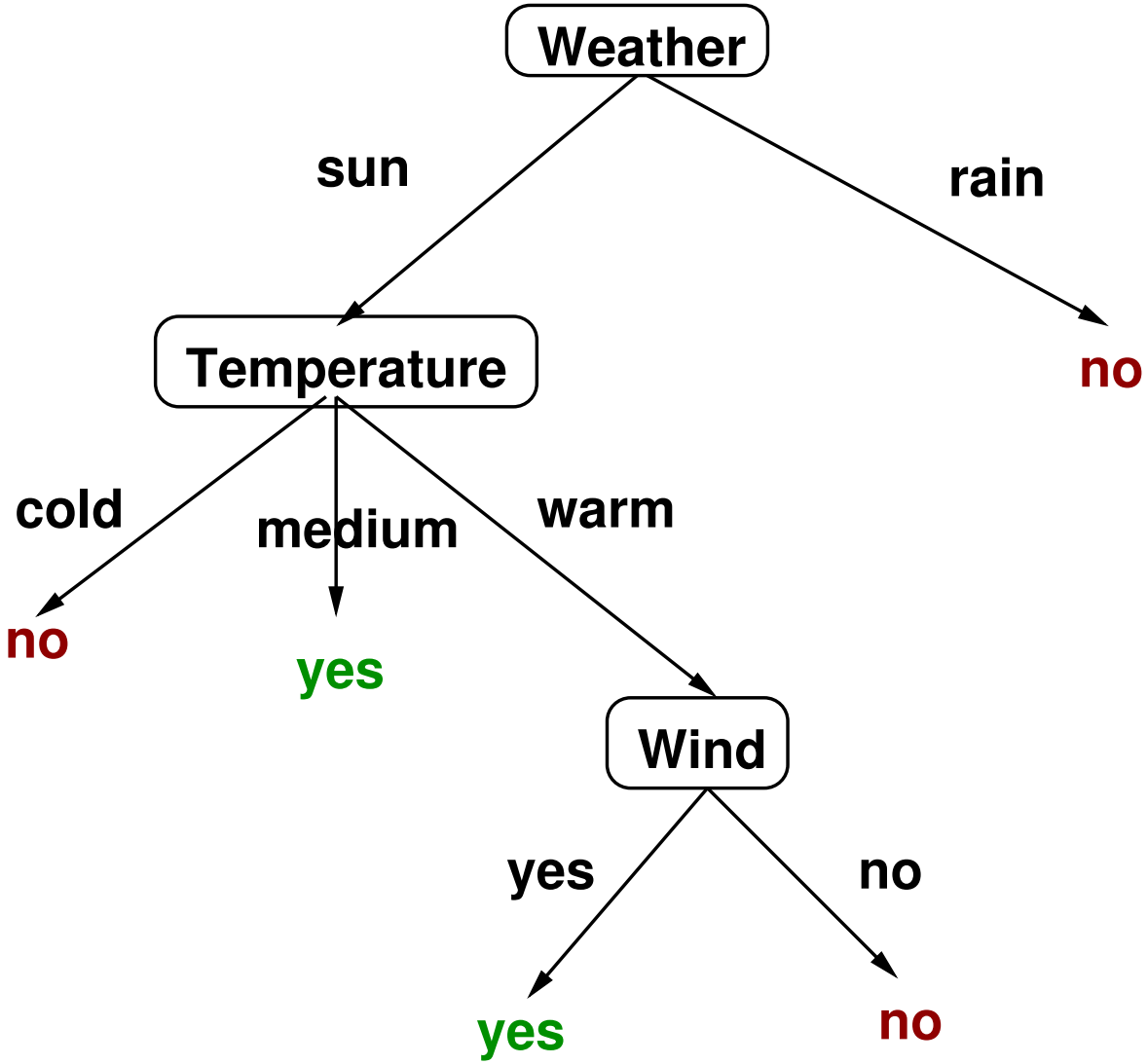


Example: Decision Tree Learning (1)

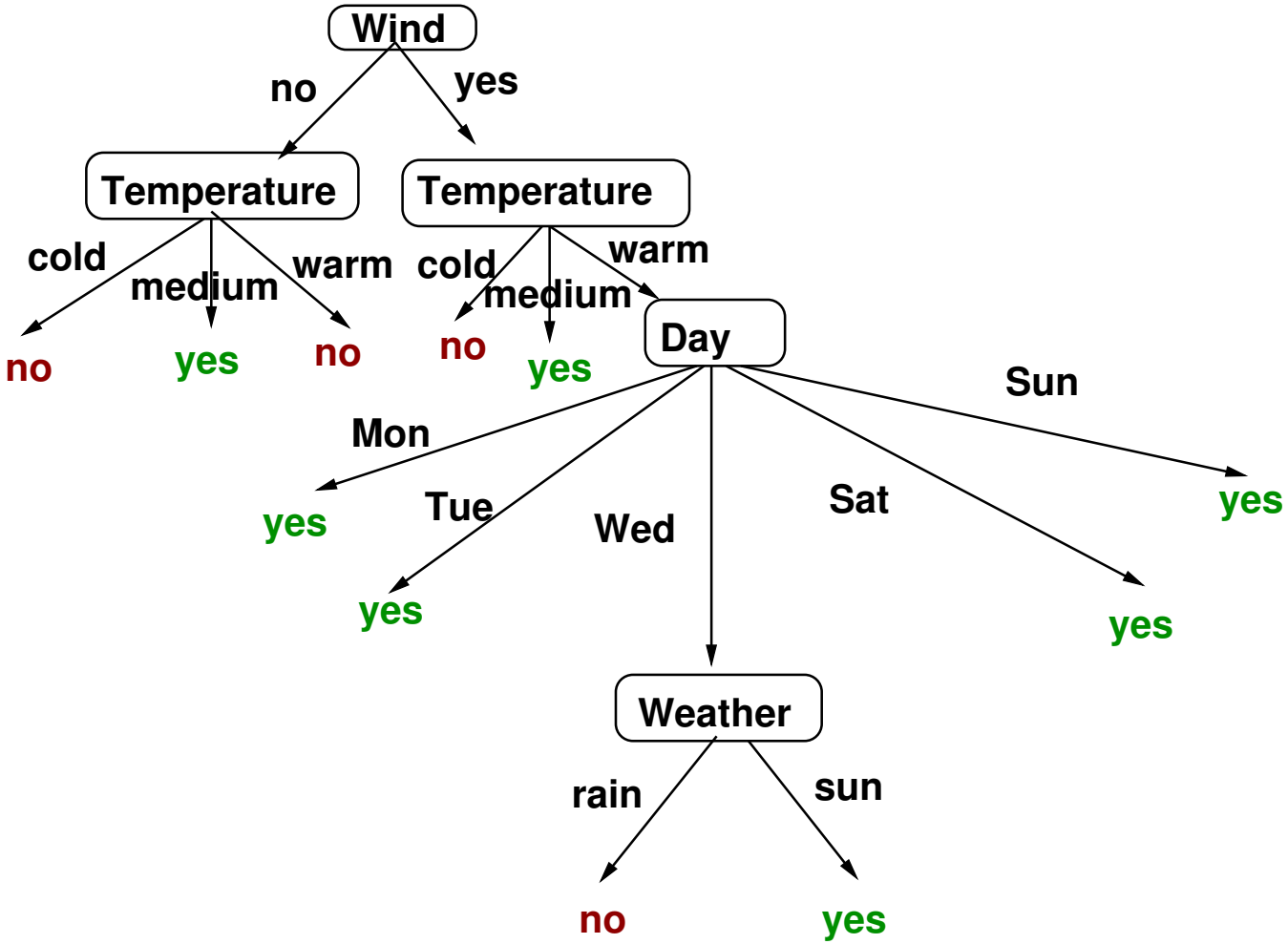
Training Data for "Play Tennis" Task

day	weather	temperature	wind	play_tennis?
Tues	sun	warm	no	no
Sun	rain	cold	no	no
Mon	sun	medium	no	yes
Wed	rain	warm	yes	no
Sat	sun	warm	yes	yes
Wed	sun	warm	yes	yes
Mon	sun	warm	yes	yes
Sun	sun	warm	yes	yes

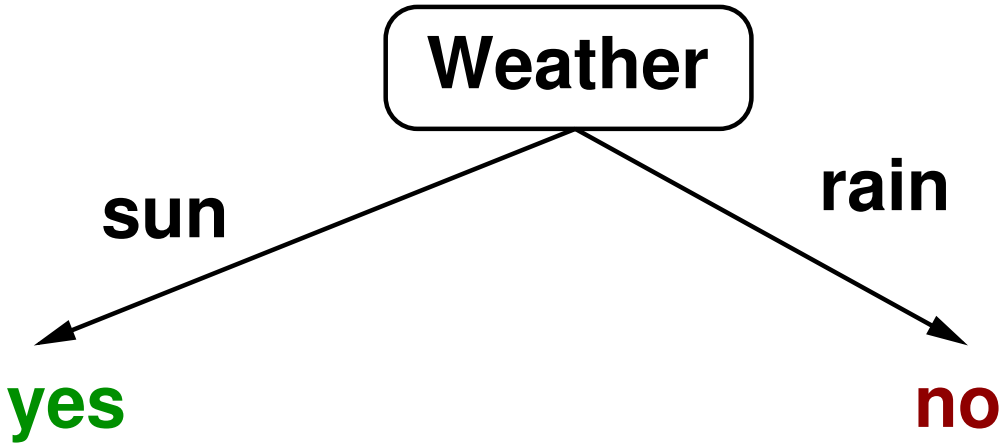
Example: Alternative Decision Trees (2)



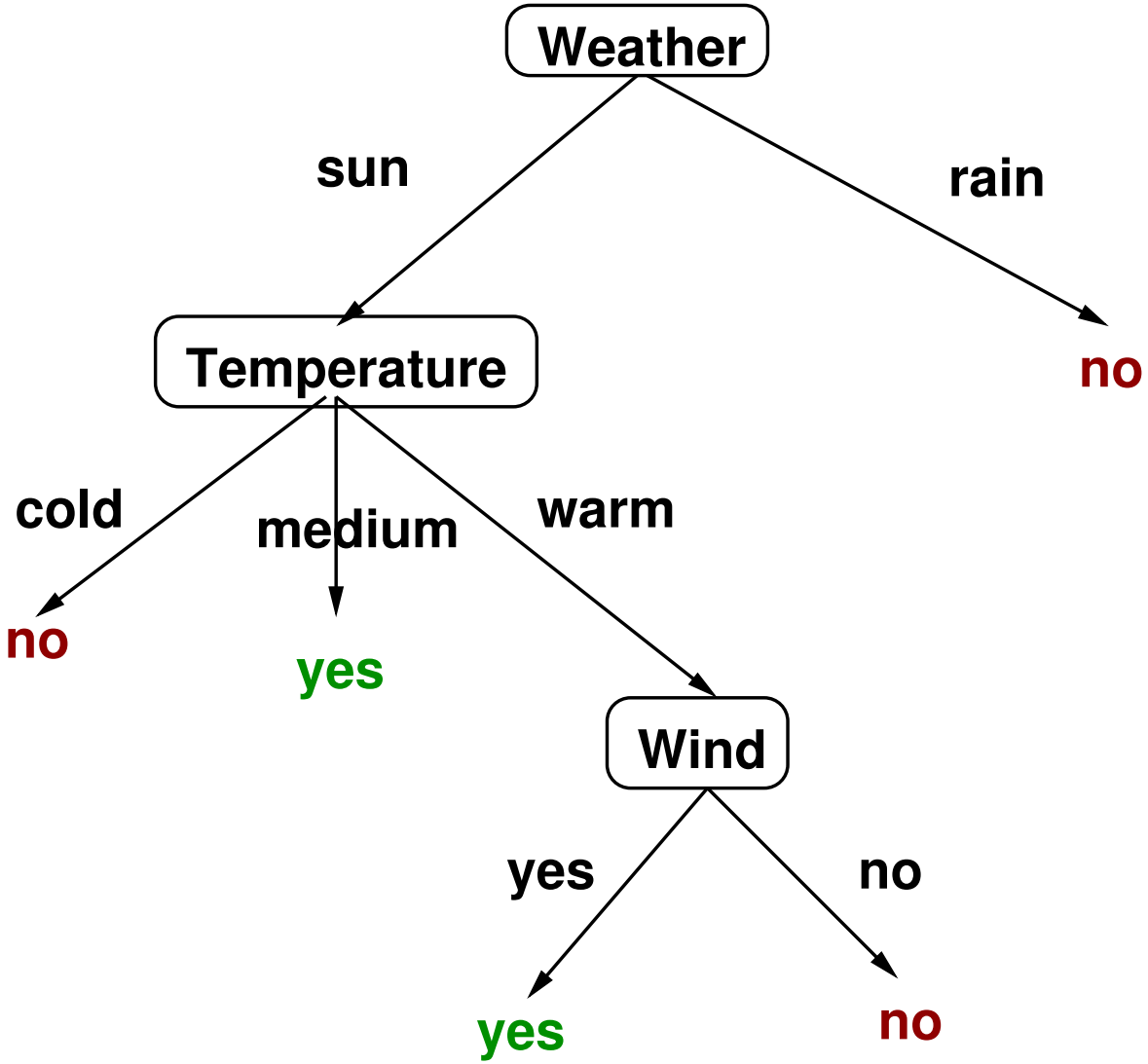
Example: Alternative Decision Trees (2)



Example: Alternative Decision Trees (2)



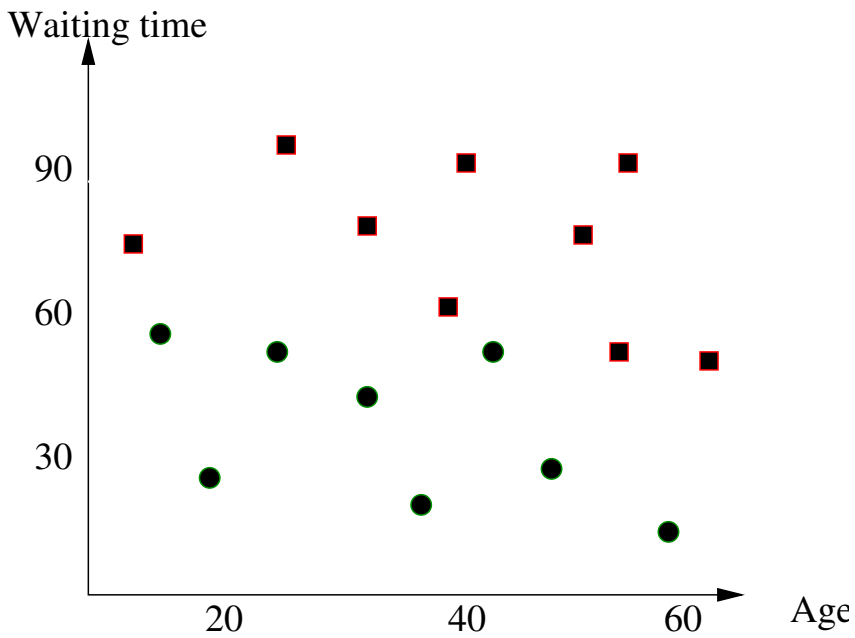
Example: Alternative Decision Trees (2)



Inductive bias: the simplest tree that fits the data (Occam's razor)

Applying the Classifier (1)

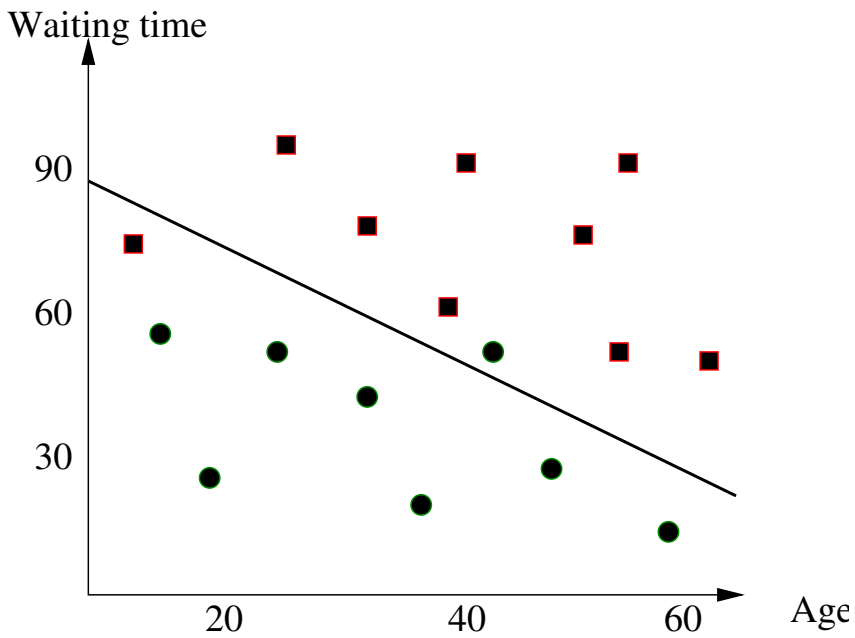
Assumption: the decision boundary learnt from the training set is also a good decision boundary for any test set because the test set is drawn from the same distribution as the training set (i.e., the two sets are not fundamentally different)



Test set

Applying the Classifier (1)

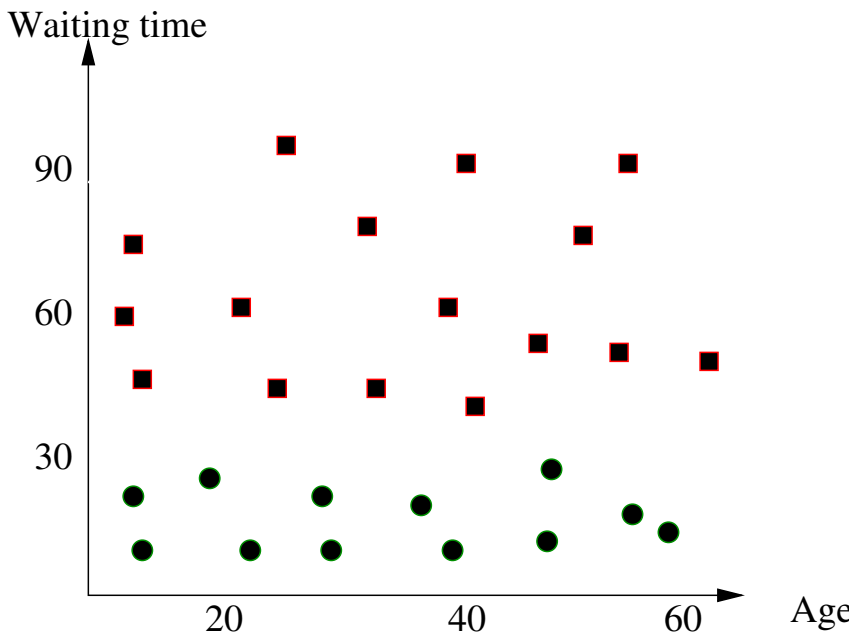
Assumption: the decision boundary learnt from the training set is also a good decision boundary for any test set because the test set is drawn from the same distribution as the training set (i.e., the two sets are not fundamentally different)



Test set

Applying the Classifier (2)

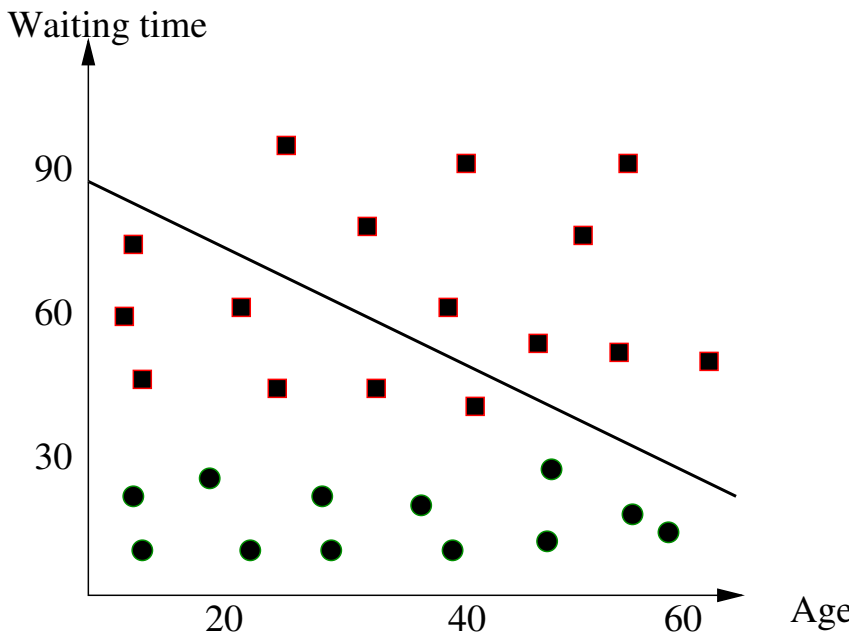
What happens if the test data are differently distributed?



Test set

Applying the Classifier (2)

What happens if the test data are differently distributed?



Test set

Semi-Supervised Machine Learning (1)

Co-Training

- two classifiers, representing different views of the data
- train both on a small labelled seed set
- apply both to large unlabelled set
- classifier A trains classifier B and vice versa

Example: Named Entity Recognition

Peter saw a documentary on **United Cakes Ltd**.

It was said they were planning to buy shares in **Henderson**.

Semi-Supervised Machine Learning (1)

Co-Training

- two classifiers, representing different views of the data
- train both on a small labelled seed set
- apply both to large unlabelled set
- classifier A trains classifier B and vice versa

Example: Named Entity Recognition

Peter saw a documentary on **United Cakes Ltd**.

It was said they were planning to buy shares in **Henderson**.

Semi-Supervised Machine Learning (1)

Co-Training

- two classifiers, representing different views of the data
- train both on a small labelled seed set
- apply both to large unlabelled set
- classifier A trains classifier B and vice versa

Example: Named Entity Recognition

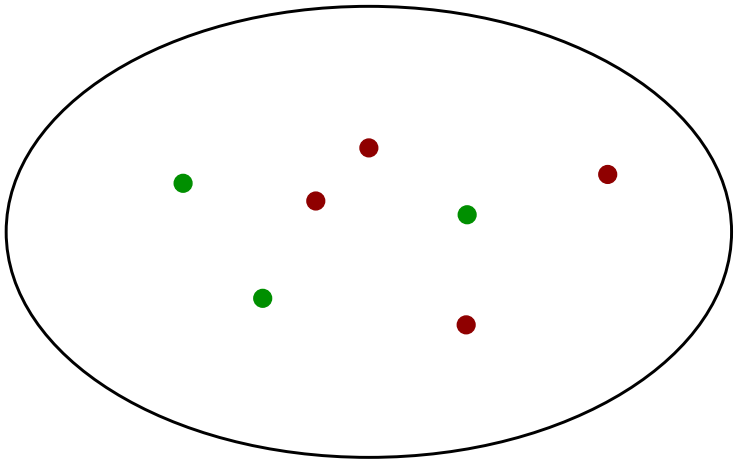
Peter saw a documentary on **United Cakes Ltd**.

It was said they were planning to buy shares in **Henderson**.

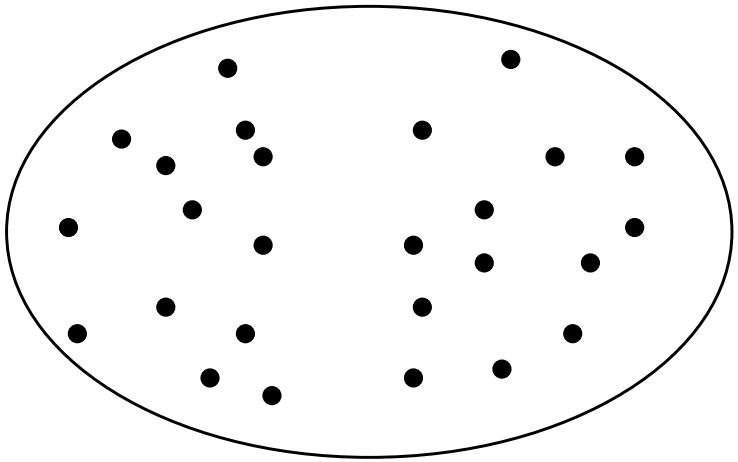
Co-Training: Example

Classifier A

Classifier B

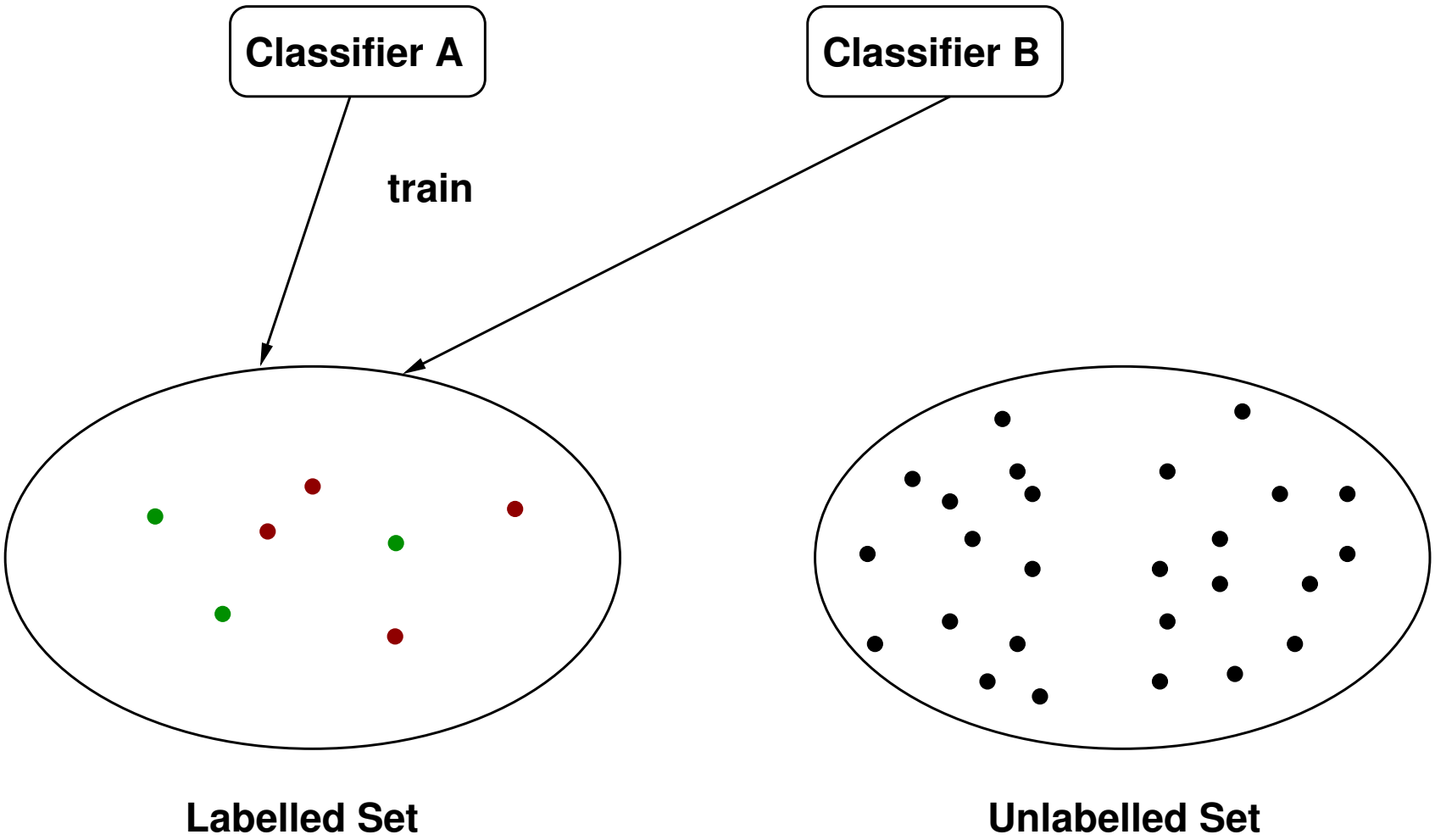


Labelled Set

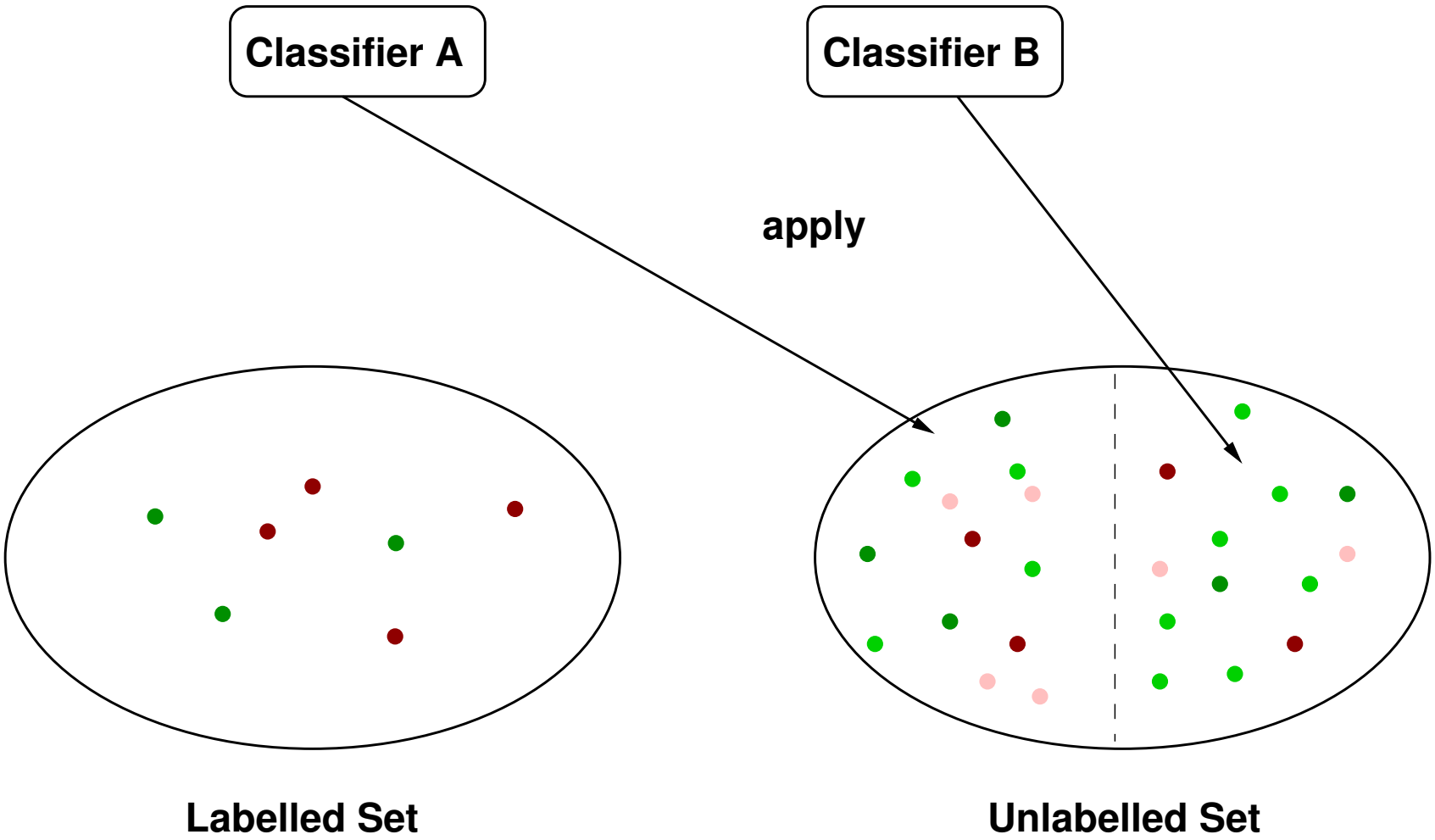


Unlabelled Set

Co-Training: Example



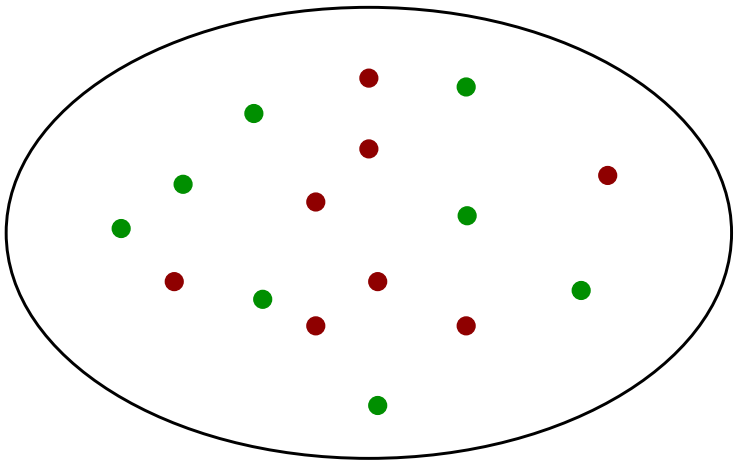
Co-Training: Example



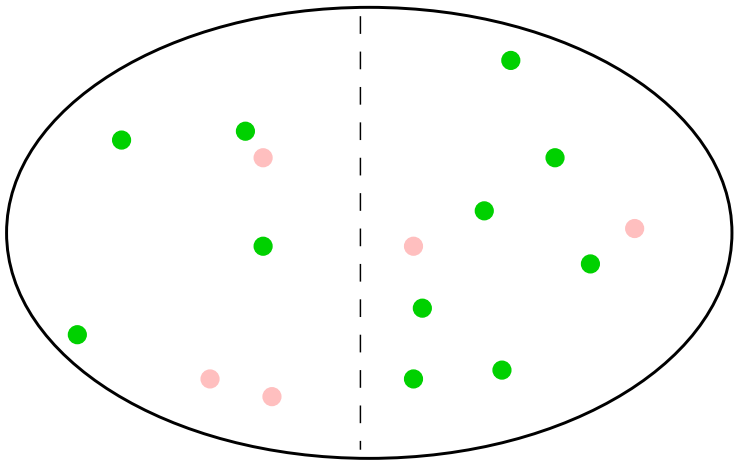
Co-Training: Example

Classifier A **Classifier B**

select new training examples

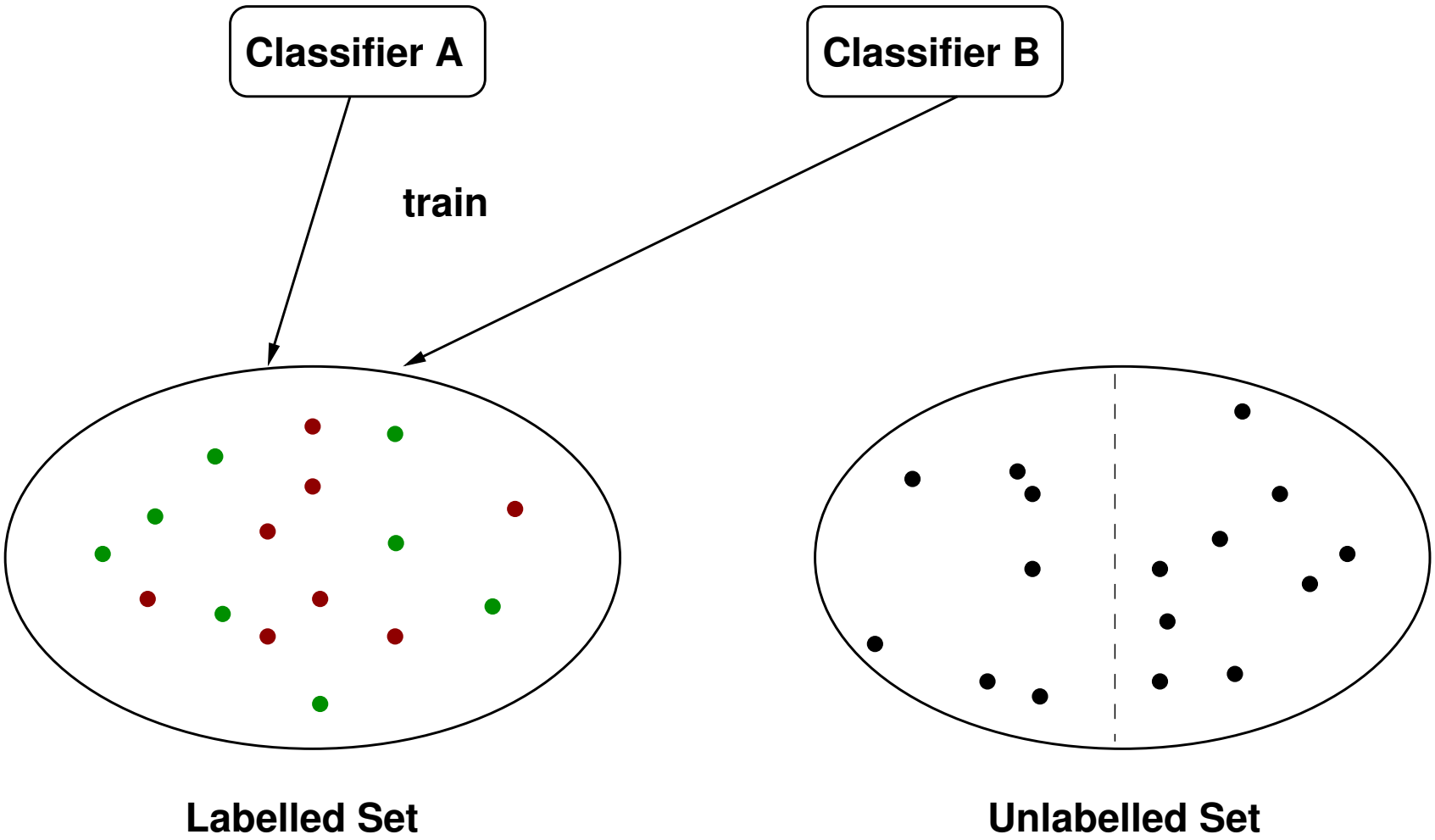


Labelled Set



Unlabelled Set

Co-Training: Example



Semi-Supervised Machine Learning (2)

Self-Training

- similar to co-training but only one classifier
- not clear whether it works