Machine Learning: finding patterns

Outline

- **Machine learning and Classification**
- Examples
- *Learning as Search
- Bias
- Weka

Finding patterns

- Goal: programs that detect patterns and regularities in the data
- Strong patterns ⇒ good predictions
  - Problem 1: most patterns are not interesting
  - Problem 2: patterns may be inexact (or spurious)
  - Problem 3: data may be garbled or missing

Machine learning techniques

- *Algorithms for acquiring structural descriptions from examples*
- Structural descriptions represent patterns explicitly
  - Can be used to predict outcome in new situation
  - Can be used to understand and explain how prediction is derived
    (*may be even more important*)
- Methods originate from artificial intelligence, statistics, and research on databases

Can machines really learn?

- Definitions of "learning" from dictionary:

  To get knowledge of by study, experience, or being taught } Difficult to measure

  To become aware by information or from observation
  To commit to memory
  To be informed of, ascertain; to receive instruction } Trivial for computers

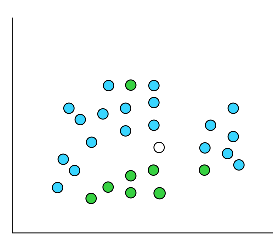
  - Operational definition:

    Things learn when they change their behavior in a way that makes them perform better in the future. } Does a slipper learn?

  - Does learning imply intention?

Classification

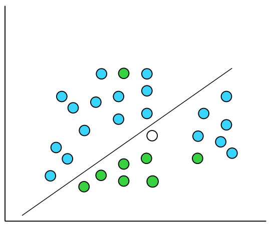
Learn a method for predicting the instance class from pre-labeled (classified) instances



Many approaches:
Regression,
Decision Trees,
Bayesian,
Neural Networks,
...

Given a set of points from classes ● ●
what is the class of new point ○?

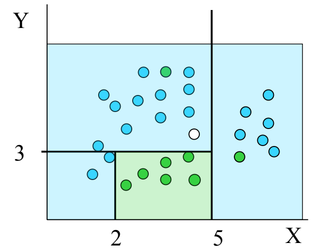
Classification: Linear Regression



- Linear Regression

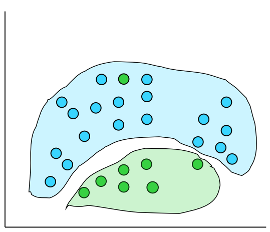
  $w_0 + w_1 x + w_2 y >= 0$
- Regression computes w_i from data to minimize squared error to 'fit' the data
- Not flexible enough

Classification: Decision Trees



if X > 5 then blue
else if Y > 3 then blue
else if X > 2 then green
else blue

Classification: Neural Nets



- Can select more complex regions
- Can be more accurate
- Also can overfit the data – find patterns in random noise

Outline

- Machine learning and Classification
- **Examples**
- *Learning as Search
- Bias
- Weka

The weather problem

Outlook	Temperature	Humidity	Windy	Play
sunny	85	85	false	no
sunny	80	90	true	no
overcast	83	86	false	yes
rainy	70	96	false	yes
rainy	68	80	false	yes
rainy	65	70	true	no
overcast	64	65	true	yes
sunny	72	95	false	no
sunny	69	70	false	yes
rainy	75	80	false	yes
sunny	75	70	true	yes
overcast	72	90	true	yes
overcast	81	75	false	yes
rainy	71	91	true	no

Given past data, Can you come up with the rules for Play/Not Play ?

What is the game?

The weather problem

- Conditions for playing golf

Outlook	Temperature	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	Normal	False	Yes
…	…	…	…	…

```
If outlook = sunny and humidity = high then play = no
If outlook = rainy and windy = true then play = no
If outlook = overcast then play = yes
If humidity = normal then play = yes
If none of the above then play = yes
```

witten&eibe

Weather data with mixed attributes

- Some attributes have numeric values

Outlook	Temperature	Humidity	Windy	Play
Sunny	85	85	False	No
Sunny	80	90	True	No
Overcast	83	86	False	Yes
Rainy	75	80	False	Yes
…	…	…	…	…

```
If outlook = sunny and humidity > 83 then play = no
If outlook = rainy and windy = true then play = no
If outlook = overcast then play = yes
If humidity < 85 then play = yes
If none of the above then play = yes
```

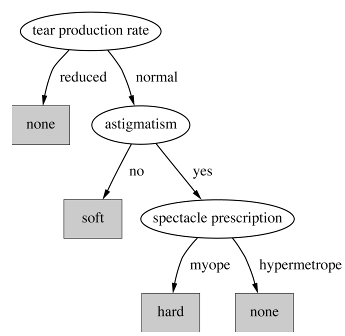

The contact lenses data

Age	Spectacle prescription	Astigmatism	Tear production rate	Recommended lenses
Young	Myope	No	Reduced	None
Young	Myope	No	Normal	Soft
Young	Myope	Yes	Reduced	None
Young	Myope	Yes	Normal	Hard
Young	Hypermetrope	No	Reduced	None
Young	Hypermetrope	No	Normal	Soft
Young	Hypermetrope	Yes	Reduced	None
Young	Hypermetrope	Yes	Normal	hard
Pre-presbyopic	Myope	No	Reduced	None
Pre-presbyopic	Myope	No	Normal	Soft
Pre-presbyopic	Myope	Yes	Reduced	None
Pre-presbyopic	Myope	Yes	Normal	Hard
Pre-presbyopic	Hypermetrope	No	Reduced	None
Pre-presbyopic	Hypermetrope	No	Normal	Soft
Pre-presbyopic	Hypermetrope	Yes	Reduced	None
Pre-presbyopic	Hypermetrope	Yes	Normal	None
Presbyopic	Myope	No	Reduced	None
Presbyopic	Myope	No	Normal	None
Presbyopic	Myope	Yes	Reduced	None
Presbyopic	Myope	Yes	Normal	Hard
Presbyopic	Hypermetrope	No	Reduced	None
Presbyopic	Hypermetrope	No	Normal	Soft
Presbyopic	Hypermetrope	Yes	Reduced	None
Presbyopic	Hypermetrope	Yes	Normal	None

A complete and correct rule set

```
If tear production rate = reduced then recommendation = none
If age = young and astigmatic = no
    and tear production rate = normal then recommendation = soft
If age = pre-presbyopic and astigmatic = no
    and tear production rate = normal then recommendation = soft
If age = presbyopic and spectacle prescription = myope
    and astigmatic = no  then recommendation = none
If spectacle prescription = hypermetrope and astigmatic = no
    and tear production rate = normal then recommendation = soft
If spectacle prescription = myope and astigmatic = yes
    and tear production rate = normal then recommendation = hard
If age young and astigmatic = yes
    and tear production rate = normal then recommendation = hard
If age = pre-presbyopic
    and spectacle prescription = hypermetrope
    and astigmatic = yes then recommendation = none
If age = presbyopic and spectacle prescription = hypermetrope
    and astigmatic = yes then recommendation = none
```


A decision tree for this problem



Classifying iris flowers

	Sepal length	Sepal width	Petal length	Petal width	Type
1	5.1	3.5	1.4	0.2	Iris setosa
2	4.9	3.0	1.4	0.2	Iris setosa
…					
51	7.0	3.2	4.7	1.4	Iris versicolor
52	6.4	3.2	4.5	1.5	Iris versicolor
…					
101	6.3	3.3	6.0	2.5	Iris virginica
102	5.8	2.7	5.1	1.9	Iris virginica
…					

```
If petal length < 2.45 then Iris setosa
If sepal width < 2.10 then Iris versicolor
...
```


Predicting CPU performance

- Example: 209 different computer configurations

	Cycle time (ns)	Main memory (Kb)		Cache (Kb)	Channels		Performance
	MYCT	MMIN	MMAX	CACH	CHMIN	CHMAX	PRP
1	125	256	6000	256	16	128	198
2	29	8000	32000	32	8	32	269
…							
208	480	512	8000	32	0	0	67
209	480	1000	4000	0	0	0	45

- Linear regression function

```
PRP =  -55.9 + 0.0489 MYCT + 0.0153 MMIN + 0.0056 MMAX
       + 0.6410 CACH - 0.2700 CHMIN + 1.480 CHMAX
```

Soybean classification

	Attribute	Number of values	Sample value
Environment	Time of occurrence	7	July
	Precipitation	3	Above normal
…			
Seed	Condition	2	Normal
	Mold growth	2	Absent
…			
Fruit	Condition of fruit pods	4	Normal
	Fruit spots	5	?
Leaves	Condition	2	Abnormal
	Leaf spot size	3	?
…			
Stem	Condition	2	Abnormal
	Stem lodging	2	Yes
…			
Roots	Condition	3	Normal
Diagnosis		19	Diaporthe stem canker

The role of domain knowledge

```
If leaf condition is normal
   and stem condition is abnormal
   and stem cankers is below soil line
   and canker lesion color is brown
then
   diagnosis is rhizoctonia root rot
```

```
If leaf malformation is absent
   and stem condition is abnormal
   and stem cankers is below soil line
   and canker lesion color is brown
then
   diagnosis is rhizoctonia root rot
```

But in this domain, "leaf condition is normal" implies
"leaf malformation is absent"!

Outline

- Machine learning and Classification
- Examples
- ***Learning as Search**
- Bias
- Weka

Learning as search

- Inductive learning: find a concept description that fits the data
- Example: rule sets as description language
  - Enormous, but finite, search space
- Simple solution:
  - enumerate the concept space
  - eliminate descriptions that do not fit examples
  - surviving descriptions contain target concept

Enumerating the concept space

- Search space for weather problem
  - 4 x 4 x 3 x 3 x 2 = 288 possible combinations
  - With 14 rules ⇒ 2.7×10^{34} possible rule sets
- Solution: candidate-elimination algorithm
- Other practical problems:
  - More than one description may survive
  - No description may survive
    - Language is unable to describe target concept
    - *or* data contains noise

The version space

- Space of consistent concept descriptions
- Completely determined by two sets
  - *L*: most specific descriptions that cover all positive examples and no negative ones
  - *G*: most general descriptions that do not cover any negative examples and all positive ones
- Only *L* and *G* need be maintained and updated
- But: still computationally very expensive
- And: does not solve other practical problems

*Version space example

- Given: red or green cows or chicken

L={} *G*={<*, *>}
<green,cow>: positive
L={<green, cow>} *G*={<*, *>}
<red,chicken>: negative
L={<green, cow>}
G={<green,*>,<*,cow>}
<green, chicken>: positive
L={<green, *>} *G*={<green, *>}

*Candidate-elimination algorithm

```
Initialize L and G
For each example e:
  If e is positive:
   Delete all elements from G that do not cover e
   For each element r in L that does not cover e:
     Replace r by all of its most specific generalizations
       that    1. cover e and
         2. are more specific than some element in G
   Remove elements from L that
     are more general than some other element in L
  If e is negative:
   Delete all elements from L that cover e
   For each element r in G that covers e:
     Replace r by all of its most general specializations
       that    1. do not cover e and
         2. are more general than some element in L
   Remove elements from G that
     are more specific than some other element in G
```


Outline

- Machine learning and Classification
- Examples
- *Learning as Search
- **Bias**
- Weka

Bias

- Important decisions in learning systems:
  - Concept description language
  - Order in which the space is searched
  - Way that overfitting to the particular training data is avoided
- These form the "bias" of the search:
  - Language bias
  - Search bias
  - Overfitting-avoidance bias

Language bias

- Important question:
  - is language universal
    or does it restrict what can be learned?
- Universal language can express arbitrary subsets of examples
- If language includes logical *or* ("disjunction"), it is universal
- Example: rule sets
- Domain knowledge can be used to exclude some concept descriptions *a priori* from the search

Search bias

- Search heuristic
  - "Greedy" search: performing the best single step
  - "Beam search": keeping several alternatives
  - …
- Direction of search
  - *General-to-specific*
    - E.g. specializing a rule by adding conditions
  - *Specific-to-general*
    - E.g. generalizing an individual instance into a rule

Overfitting-avoidance bias

- Can be seen as a form of search bias
- Modified evaluation criterion
  - E.g. balancing simplicity and number of errors
- Modified search strategy
  - E.g. pruning (simplifying a description)
    - Pre-pruning: stops at a simple description before search proceeds to an overly complex one
    - Post-pruning: generates a complex description first and simplifies it afterwards

Weka