COMP9414: Artificial Intelligence

Lecture 6a: Learning

Wayne Wobcke

e-mail:w.wobcke@unsw.edu.au

This Lecture

- Machine Learning
  - Methodological Issues
- Supervised Learning
  - Decision Tree Learning
- Text Classification
  - Bayesian Classification
- Data Science

Types of Learning

- Supervised Learning
  - Agent is presented with examples of inputs and their target outputs, and must learn a function from inputs to outputs that agrees with the training examples and generalizes to new examples
- Reinforcement Learning
  - Agent is not presented with target outputs for each input, but is periodically given a reward, and must learn to maximize (expected) rewards over time
- Unsupervised Learning
  - Agent is only presented with a series of inputs, and must find useful patterns in these inputs

Supervised Learning

- Given a training set and a test set, each consisting of a set of items for each item in the training set, a set of features and a target output
- Learner must learn a model that can predict the target output for any given item (characterized by its set of features)
- Learner is given the input features and target output for each item in the training set
  - Items may be presented all at once (batch) or in sequence (online)
  - Items may be presented at random or in time order (stream)
  - Learner **cannot** use the test set **at all** in defining the model
- Model is evaluated by its performance on predicting the output for each item in the test set

Methods vs Models

- Various learning methods can be used to generate models
  - Decision Trees
  - Support Vector Machines
  - Neural Networks/Deep Learning

基准

- Evaluate methods by evaluating models on a variety of datasets
  - Problem with availability of standard benchmark datasets
  - Models depend on problem formulation and on parameters
  - End users may only care about a model, not a general method
  - Most machine learning research evaluates methods, not models

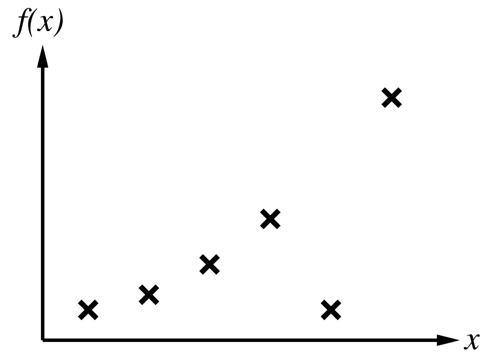
Supervised Learning – Methodology

- Feature “engineering” – select relevant features
- Choose representation of input features and outputs
- Preprocessing method to extract features from raw data
- Choose learning method(s) to evaluate
- Choose training regime (including parameters)
- Evaluation
  - Choose realistic baseline for comparison
  - Choose type of internal validation, e.g. cross-validation
  - Sanity check results with human expertise, other benchmarks

明智

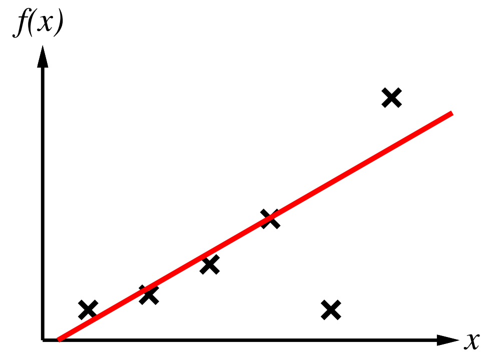
Curve Fitting

Which curve gives the “best fit” to this data?



Curve Fitting

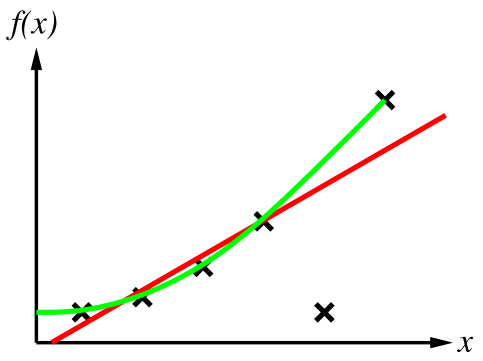
Which curve gives the “best fit” to this data?



Straight line?

Curve Fitting

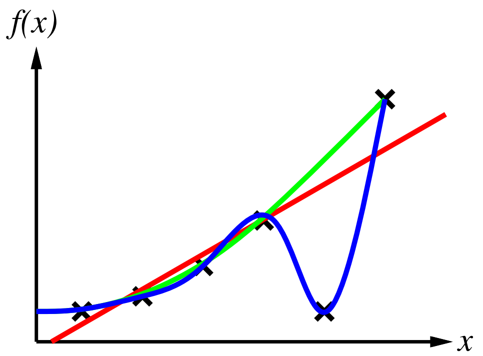
Which curve gives the “best fit” to this data?



Parabola?

Curve Fitting

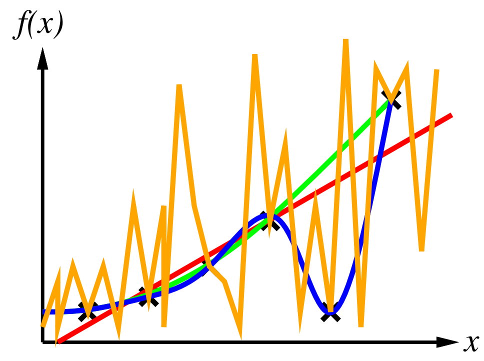
Which curve gives the “best fit” to this data?



4th order polynomial?

Curve Fitting

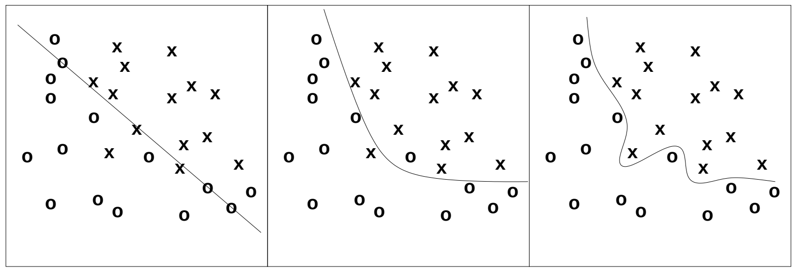
Which curve gives the “best fit” to this data?



Something else?

Ockham’s Razor

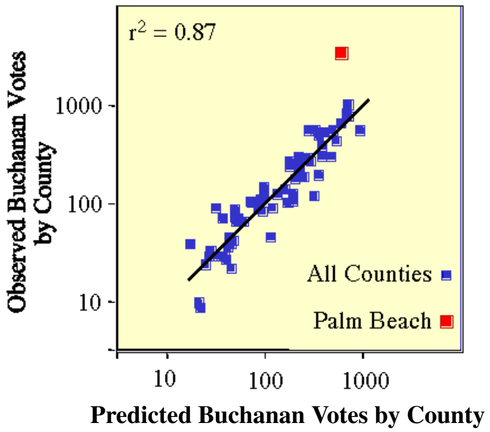
“The most likely hypothesis is the simplest one consistent with the data.”



inadequate good compromise overfitting

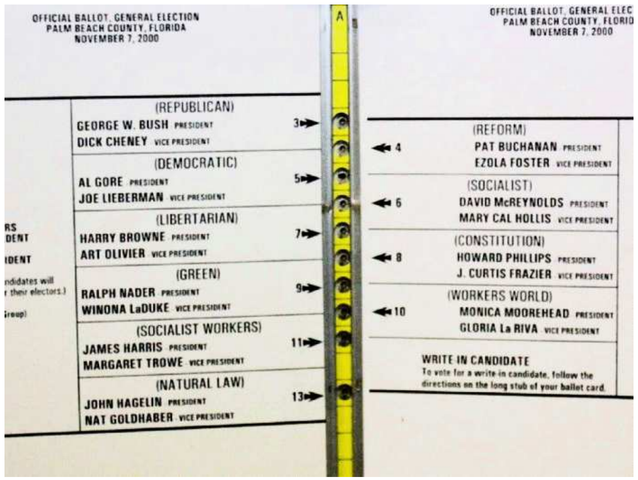
Since there can be noise in the measurements, in practice need to make a tradeoff between simplicity of the hypothesis and how well it fits the data

Outliers



When is it OK to remove outliers?

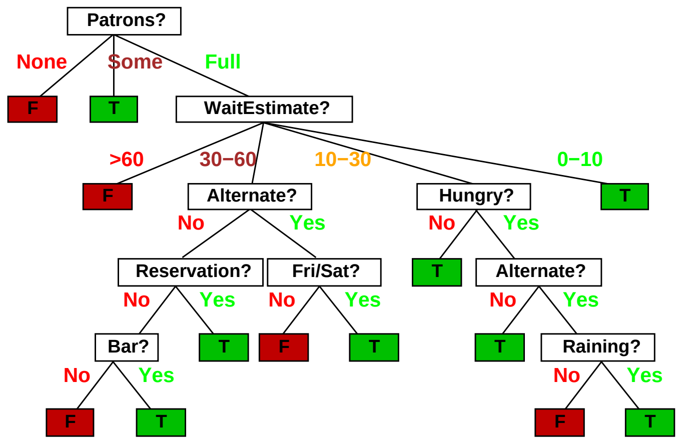
Butterfly Ballot



Restaurant Training Data

	Alt	Bar	F/S	Hun	Pat	Price	Rain	Res	Type	Est	Wait?
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

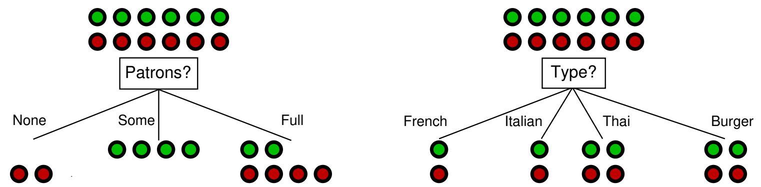
Decision Tree



Generalization

- Provided the training set is not inconsistent, attributes can be split in any order to produce a tree that correctly classifies all examples in the training set
- However, what is needed is a tree likely to generalize – correctly classify the (unseen) examples in the test set
- In view of Ockham's Razor, a simpler hypothesis is preferred – "simpler" = smaller tree
- How to choose attributes in order to produce a small tree?

Choosing an Attribute to Split



Patrons is a "more informative" attribute than Type, because it splits the examples more nearly into sets that are "all positive" or "all negative"

This notion of "informativeness" can be quantified using the mathematical concept of "entropy"

A parsimonious tree can be built by minimizing the entropy at each step
过于节俭的、质量差的

Entropy

- Entropy is a measure of "randomness" (lack of uniformity)
  - Related to prior distribution of some random variable
  - Higher entropy means more randomness
  - "Information" (about distribution) reduces entropy
- Idea: Split based on information gain
  - Loss of entropy based on "communicating" value of attribute
  - Related to Shannon's information theory
  - Measure information gain in bits

Definition. If the prior probabilities of n attribute values are $p_1, \cdots, p_n$, then the entropy of the distribution is

$$H(\langle p_1, \cdots, p_n \rangle) = \sum_{i=1}^{n} -p_i \log_2 p_i$$

Entropy and Huffmann Coding

Entropy is the number of bits per symbol achieved by a (block) Huffman Coding scheme

Example 1: $H(\langle 0.5, 0.5 \rangle) = 1$ bit

To encode in 0s and 1s, a long message composed of the two letters `A` and `B` which occur with equal frequency, assign `A=0` and `B=1`

One bit (binary digit) is needed to encode each letter

Entropy and Huffmann Coding

Example 2: $H(\langle 0.5, 0.25, 0.25\rangle) = 1.5$ bits

To encode a message consisting of the letters `A`, `B` and `C`, where `B` and `C` occur equally often but `A` occurs twice as often as the other two letters, assign `A=0`, `B=10`, `C=11`

The average number of bits needed to encode each letter is 1.5

If the letters occur in some other proportion, they need to be "blocked" in some order to encode them efficiently

The average number of bits required by the most efficient coding scheme is given by the entropy

$$H(\langle p_1, \cdots, p_n\rangle) = \sum_{i=1}^{n} -p_i \log_2 p_i$$

Entropy

Suppose there are p positive and n negative examples at a node –
$\rightarrow H(\langle p/(p+n), n/(p+n)\rangle)$ bits needed to classify a new example
– for the 12 restaurant examples, $p = n = 6$, so need 1 bit

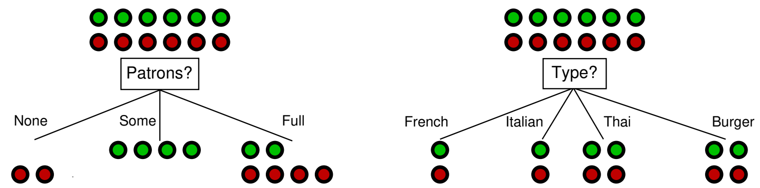
An attribute splits the examples E into subsets E_i, each of which needs less information to complete the classification (reduces entropy)

Let E_i have p_i positive and n_i negative examples –
$\rightarrow H(\langle p_i/(p_i+n_i), n_i/(p_i+n_i)\rangle)$ bits needed to classify a new example
$\rightarrow$ expected number of bits per example over all branches is

$$\sum_i \frac{p_i+n_i}{p+n} H(\langle \frac{p_i}{p_i+n_i}, \frac{n_i}{p_i+n_i}\rangle)$$

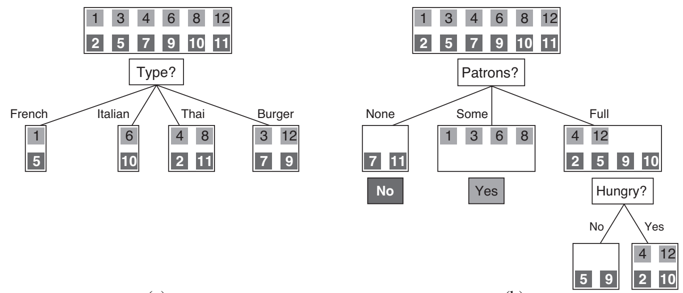
For Patrons, this is 0.459 bits, for Type this is (still) 1 bit $\therefore$ split on Patrons

Information Gain



$$\begin{aligned}
\text{For Patrons, Entropy} &= \frac{1}{6}(0) + \frac{1}{3}(0) + \frac{1}{2}[-\frac{1}{3}\log(\frac{1}{3}) - \frac{2}{3}\log(\frac{2}{3})] \\
&= 0 + 0 + \frac{1}{2}[\frac{1}{3}(1.585) + \frac{2}{3}(0.585)] = 0.459 \\
\text{For Type, Entropy} &= \frac{1}{6}(1) + \frac{1}{6}(1) + \frac{1}{3}(1) + \frac{1}{3}(1) = 1
\end{aligned}$$

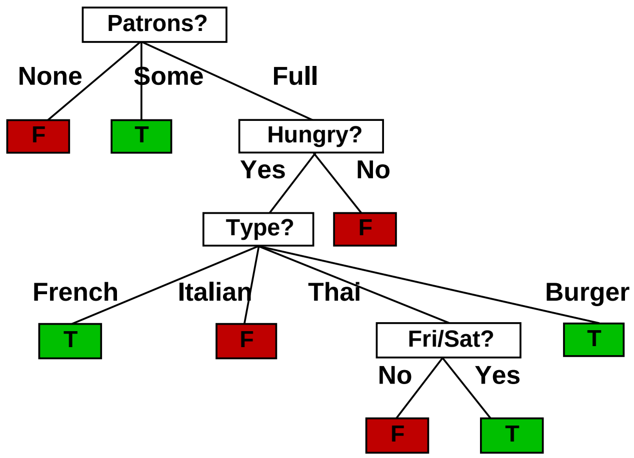
Choosing Next Attribute



(a) (b)

After splitting on Patrons, split on Hungry

Induced Decision Tree



Laplace Error and Pruning

Following Ockham's Razor, prune branches that do not provide much benefit in classifying the items (aids generalization, avoids overfitting)

For a leaf node, all items will assigned the majority class at that node. Estimate error rate on the (unseen) test items using the Laplace error

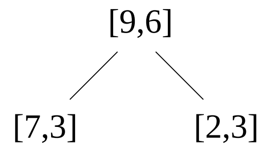
$$E = 1 - \frac{n+1}{N+k}$$

N = total number of (training) items at the node
n = number of (training) items in the majority class
k = number of classes

If the average Laplace error of the children exceeds that of the parent node, prune off the children

Minimal Error Pruning

Should the children of this node be pruned or not?



Left child has class frequencies [7,3]

$$E = 1 - \frac{n+1}{N+k} = 1 - \frac{7+1}{10+2} = 0.333$$

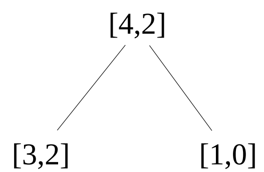
Right child has $E = 0.429$
Parent node has $E = 0.412$
Average for Left and Right child is

$$E = \frac{10}{15}(0.333) + \frac{5}{15}(0.429) = 0.365$$

Since $0.365 < 0.412$, children should **not** be pruned

Minimal Error Pruning

Should the children of this node be pruned or not?



Left child has class frequencies [3,2]

$$E = 1 - \frac{n+1}{N+k} = 1 - \frac{3+1}{5+2} = 0.429$$

Right child has $E = 0.333$
Parent node has $E = 0.375$
Average for Left and Right child is

$$E = \frac{5}{6}(0.429) + \frac{1}{6}(0.333) = 0.413$$

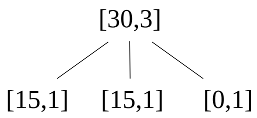
Since $0.413 > 0.375$, children should be pruned

Minimal Error Pruning

Should the children of this node be pruned or not?

Left and Middle child have class frequencies [15,1]

$$E = 1 - \frac{n+1}{N+k} = 1 - \frac{15+1}{16+2} = 0.111$$



Right child has $E = 0.333$
Parent node has $E = \frac{4}{35} = 0.114$
Average for Left, Middle and Right child is

$$E = \frac{16}{33}(0.111) + \frac{16}{33}(0.111) + \frac{1}{33}(0.333) = 0.118$$

Since $0.118 > 0.114$, children should be pruned

Summary

- Supervised Learning
  - Training set and test set
  - Predict target value based on input features
- Ockham's Razor
  - Tradeoff between simplicity and accuracy
- Decision Trees
  - Generalize by building a smaller tree (using entropy)
  - Prune nodes based on Laplace error
  - Other ways to prune Decision Trees

