ARCHER Training Courses

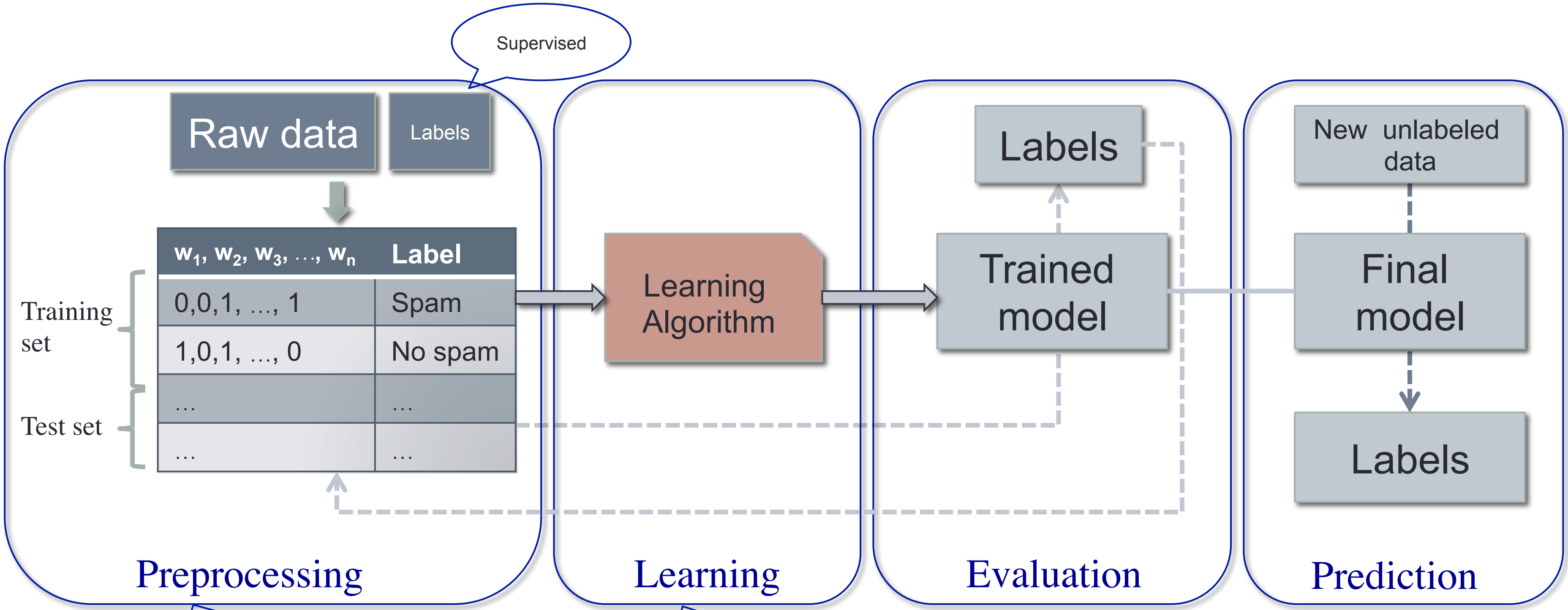
Supervised Learning - Feature Extraction, Feature Selection, Decision Trees, Random Forests

Reusing this material

Machine Learning

- **Motivation:** to solve a problem on a computer we need an algorithm to transform an input to an output. For example, we could devise an algorithm for sorting a set of numbers. However, for some tasks we **do not have an algorithm**.
- **Example**: to tell spam emails from legitimate emails.
  - Input: email document
  - Output: spam/not spam
- **What we lack in knowledge, we make up for in data.**
  - We can easily store thousands of example messages (*training set*) which we know to be spam or not spam, and we want the machine to automatically extract the algorithm (learn) for this classification task.
- **Types of Machine Learning:**
  - **Supervised Learning** where there is an input X, and an output Y and the task is to learn the mapping from the input to the output.
    - **Classification**: when Y is a categorical variable (e.g. spam/not spam)
    - **Regression**: when Y is a continuous variable.
  - **Unsupervised Learning:** there is only an input X. The aim is to find regularities/structure in the input space. One method is called *clustering*, where the aim is to find clusters or groupings of input.

Supervised Machine Learning Classification

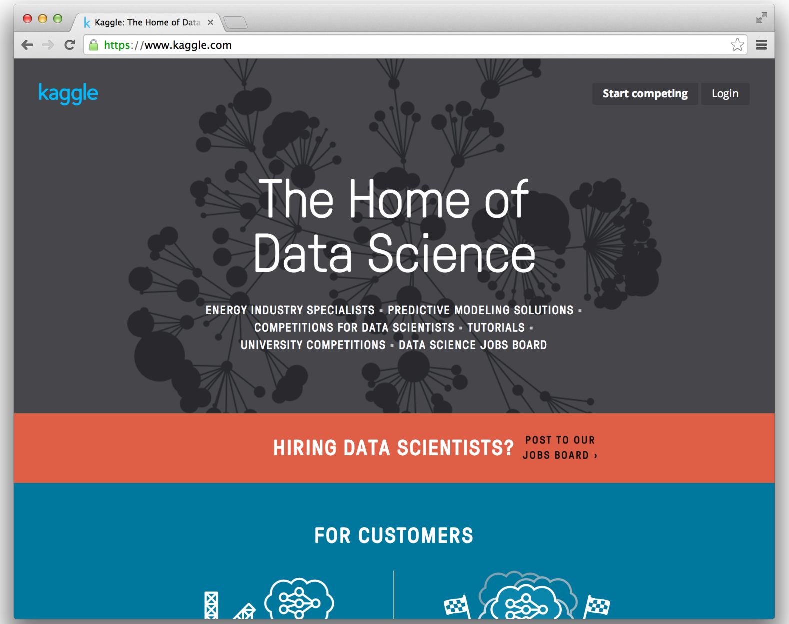


Data cleaning
Feature extraction
Dimensionality reduction
Sampling

Model Selection
Cross-validation
Cost function / performance metrics
Hyperparameters optimisation

Feature extraction and selection

- “*Feature extraction and selection are the most important but underrated steps in machine learning. Better features are better than better algorithms.*”, Will Cukierski, Kaggle


Kaggle: The Home of Data
https://www.kaggle.com
kaggle
Start competing
Login
The Home of Data Science
ENERGY INDUSTRY SPECIALISTS • PREDICTIVE MODELING SOLUTIONS • COMPETITIONS FOR DATA SCIENTISTS • TUTORIALS • UNIVERSITY COMPETITIONS • DATA SCIENCE JOBS BOARD
HIRING DATA SCIENTISTS?
POST TO OUR JOBS BOARD ›
FOR CUSTOMERS

Example mobile phone data

Outgoing Calls
Date-time
Duration
Number
Country

Customer
Address
Age
Email
Twitter account

Handsets
Date given
Manufacturer
Model

Payments
Date-time
Channel
Amount

Incoming Calls
Date-time
Duration
Number
Country

Contracts
Contracts
Start
End
Duration
Amount

Support events
Date-time
Type {tech, complaint}
Duration

Feature extraction

- Use a domain expert if you have one
  - If you don't have one – find one!
- Features that may be useful to predict churn
  - Average calls per week
  - Average data usage per week
  - Number of calls made abroad in last month
  - Usage of data abroad in last month
  - Number of foreign trips in last 6 months
  - Time left on contract
  - Number of friends (detected through calls and texts) recently churned?
  - Number of positive/negative tweets in last month
  - Number of positive/negative phone related tweets in last month
  - Phone related tweets of friends?

Feature extraction information buckets

- Relevant and useful but impossible to get:
  - Salary, family status
  - Can some of your data be useful proxies for this?
  - Can external data help here?
- Relevant and useful, possible to log, did log.
  - Feature selection will help discover if it is useful.
- Relevant and useful, possible to log, didn’t log.
  - Think in advance if you want about what may be useful to log.
- Not relevant or useful, but didn’t know that and logged it.
  - Features selection with find this out.
- Not relevant or useful, can’t capture it.
  - Don’t worry ‘bout a thing!

Large number of features

- In late 1990s few domains explored more than 40 features
- Now it is not uncommon for the number of features to be very large:
  - Gene expression:
    - Microarray data says what genes are expressed in a sample (e.g. cancer biopsy)
    - 6,000 to 60,000 genes (features)
    - 100 patients in each category (cancer/non-cancer)
  - Text classification:
    - Bag of words
    - 15,000 effective words (features)
    - x50,000 to 800,000 documents

Feature selection for supervised learning

- Feature selection: choosing features to include in a model
- Why do feature selection?
  - Reduce measurement and storage requirements
  - Reduce time for both training and model utilisation
  - Facilitate visualisation and improve understanding
  - Defy the curse of dimensionality and improve performance
- Not looking to rank individual features but instead find *useful* subsets for building good predictors
  - Relevant features that are highly correlated with features already in a subset would therefore be excluded.
- Ranking may be useful for other tasks:
  - e.g. targeting drug research

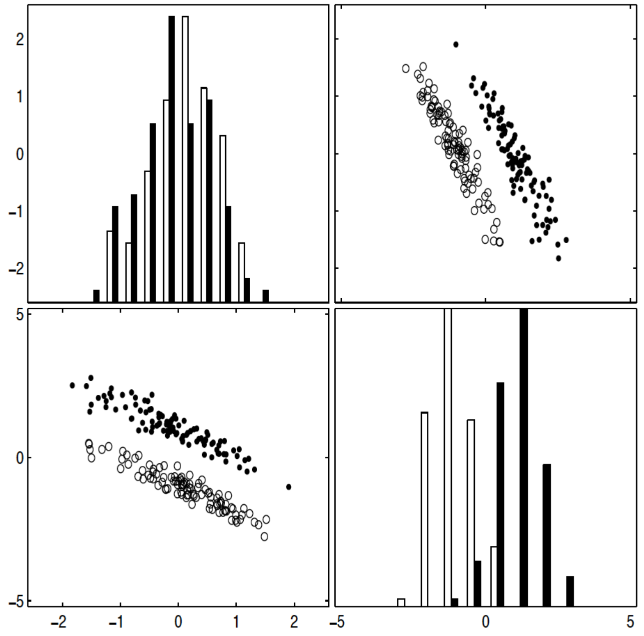
Choosing features

- In a 2003 paper *Introduction to Variable and Feature Selection*, Guyon and Elisseeff classify three approaches to feature selection:
  - Filters
    - Select subsets of variables as pre-processing step before learning.
  - Wrappers
    - Use learning system as black box to score subsets of variables as to their predictive power.
  - Embedded methods
    - Feature selection is an embedded part of the learning algorithm.

Filters

- Rank features according to a metric or statistic
  - This is a proxy measure for the performance of a model and is designed to fast to compute while still being useful
- If we look at all features independently
  - Could miss important interactions
  - Could have redundant features
    - But unless they are absolutely correlated then likely to still be useful

- Example methods:
  - Mutual information
  - Correlation with target variable
  - Significance tests
    - e.g. run linear regression with one feature and look at R-squared or p-values.

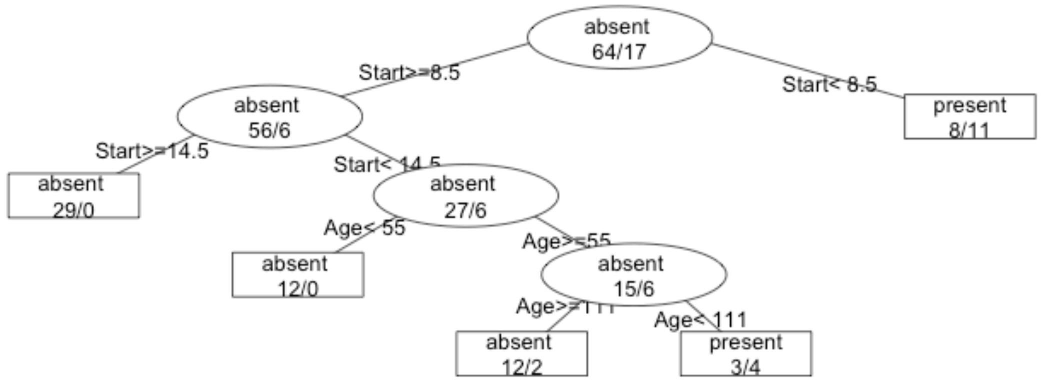


Wrappers

- Use learning system as black box to score subsets of variables as to their predictive power.
- Selection algorithms
  - Forward selection
    - Add features one at a time choosing the one that gives best improvement.
    - Stop when selection criterion fails to improve.
  - Backward elimination
    - Remove features one at a time choosing the one that gives best improvement
    - Stop when selection criterion fails to improve.
  - Combined approach
    - Add a few, remove the worst, repeat.
    - {A} {A,B} {A,B,C} {A,B,C,D} {A,C,D} {A,C,D,E}
- Need to divide training set into two

Embedded Selection: Decision Trees

- Embedded selection:
  - Learning algorithms explicitly selects features
    - E.g. decision tree (classification) or regression tree (prediction)



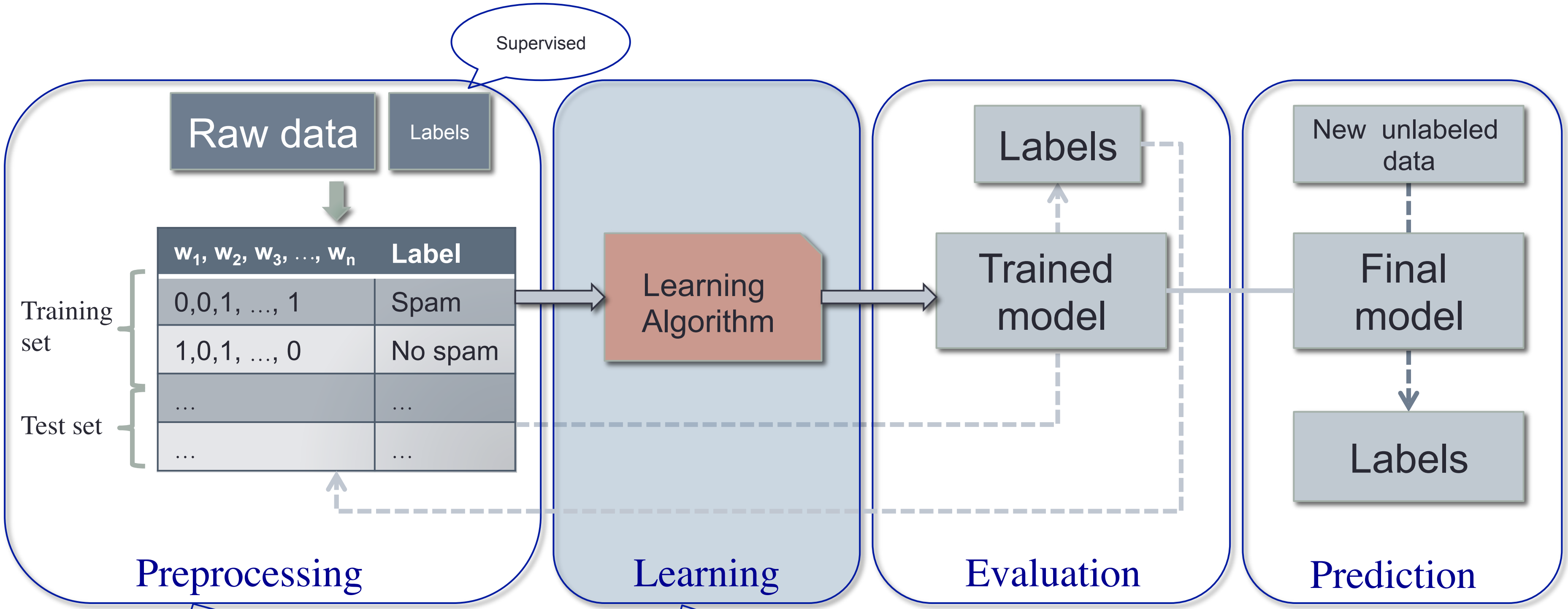
Kyphosis dataset of children who have had corrective back surgery.

Data:

- Kyphosis absent/present
- Age (months)
- Number of vertebrae
- Start vertebra

- Decision trees can be very useful to extract knowledge that humans can understand and act upon.

Supervised Machine Learning Classification



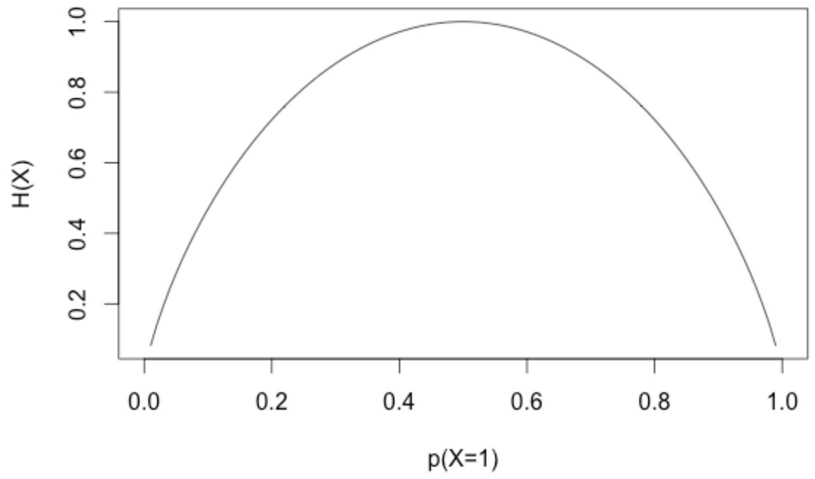
Data cleaning
Feature extraction
Dimensionality reduction
Sampling

Decision Trees:
- Embedded Feature selection
- Maximise information gain using entropy to decide the splits

Entropy

- Entropy – measure of disorder or uncertainty.
- For binary event X we define entropy as:

$$H(X) = -p(X=1)\log_2(p(X=1)) - p(X=0)\log_2(p(X=0))$$



Information gain for a feature

- Define information gain for feature *a* as the amount of entropy we lose by adding that feature

$$IG(X,a) = H(X) - H(X|a)$$

- Define *specific conditional entropy* as:

$$H(X|a=a_0) = -p(X=1|a=a_0)\log_2(p(X=1|a=a_0)) - p(X=0|a=a_0)\log_2(p(X=0|a=a_0))$$

- Combine these to get *conditional entropy*:

$$H(X|a) = \sum_{a_i} p(a=a_i)\cdot H(X|a=a_i)$$

Information gain worked example

Hat	Glasses	Gender
T	F	M
F	F	M
F	F	M
F	T	M
T	F	F
T	F	F
T	T	F
F	T	F

Random variable: gender

$$H(X) = -p(X = M)\log_2(p(X = M)) - p(X = F)\log_2(p(X = F)) = 1$$

$$\begin{aligned} H(X|\text{hat} = T) = -&p(X = M|\text{hat} = T)\log_2(p(X = M|\text{hat} = T)) - \\ &p(X = F|\text{hat} = T)\log_2(p(X = F|\text{hat} = T)) \\ &= -0.25 \cdot \log_2(0.25) - 0.75 \cdot \log_2(0.75) = 0.8112781 \end{aligned}$$

$$H(X|\text{hat} = F) = 0.8112781$$

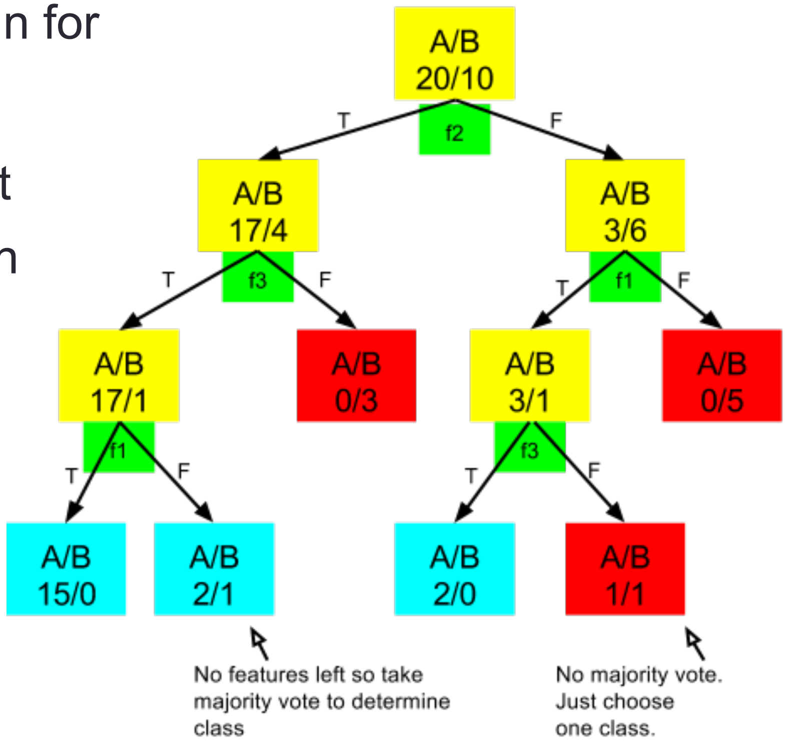
$$\begin{aligned} H(X|\text{hat}) &= p(\text{hat} = T)H(X|\text{hat} = T) + p(\text{hat} = F)H(X|\text{hat} = F) \\ &= 0.5 \cdot 0.8112781 + 0.5 \cdot 0.8112781 = 0.8112781 \end{aligned}$$

$$IG(X, \text{hat}) = H(X) - H(X|\text{hat}) = 1 - 0.8112781 = 0.1887219$$

$$IG(X, \text{glasses}) = H(X) - H(X|\text{glasses}) = 1 - 0.9508458 = 0.0491542$$

Decision Tree algorithm

- At root node determine the information gain for each feature
- Split the node according to the feature that gives the maximum information gain (given starting entropy for that node)



- Repeat process again for each new node until:
  - All entries in the node are the same class, or
  - All features are used: take a majority vote

- Final tree to often pruned to avoid **over-fitting**

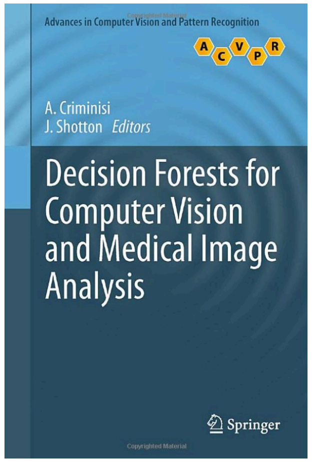
The model learns the noise of the training data and is unable to generalise to unseen data

Bagging

- Ensemble algorithms
  - Produce multiple models and then when classifying (or predicting) combine the results of all the models in some way
- Bagging is a popular ensemble algorithm
  - Bagging = Bootstrap AGGregation
  - Bootstrap sampling – sampling with replacement
  - Take bootstrap samples, train multiple models, aggregate results from multiple models
- Aggregation of results:
  - Classification: choose most common class
  - Prediction: average the results
- No need to prune as the sampling and multiple trees prevents over-fitting

Random Forest

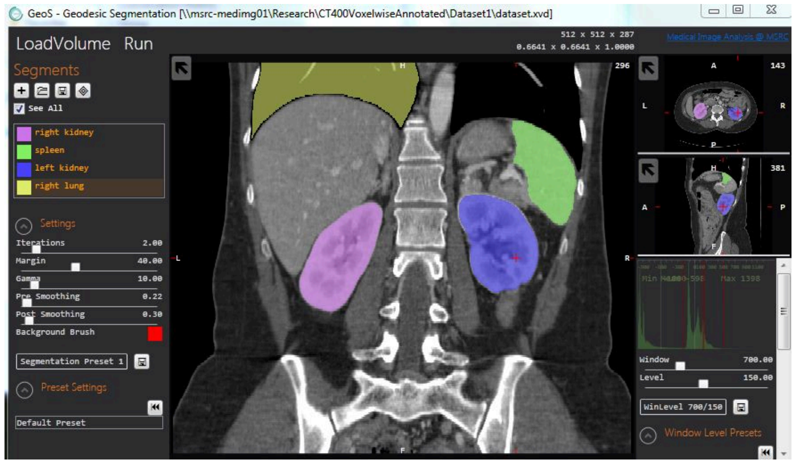
- Bagging of decision trees plus one change
- Change to algorithm to select from only a random sample of features at *each node*
- Typically *SQRT(n)* where *n* is the number of features
- Notoriously difficult to understand
- Very popular in medical image analysis and segmentation


Advances in Computer Vision and Pattern Recognition

A. Criminisi
J. Shotton *Editors*

Decision Forests for Computer Vision and Medical Image Analysis

Springer


GeoS - Geodesic Segmentation [\\msrc-medimg01\Research\CT400VoxelwiseAnnotated\Dataset1\dataset.xvd]

LoadVolume Run

512 x 512 x 287
0.6641 x 0.6641 x 1.0000

Medical Image Analysis @ MSRC

Segments

✓ See All

🟪	right kidney
🟩	spleen
🟦	left kidney
🟨	right lung

Settings

```
Iterations           2.00
Margin              40.00
Gamma               10.00
Pre Smoothing        0.22
Post Smoothing       0.30
Background Brush     🟥
```

Segmentation Preset 1

Preset Settings

Default Preset

```
Window             700.00
Level              150.00
```

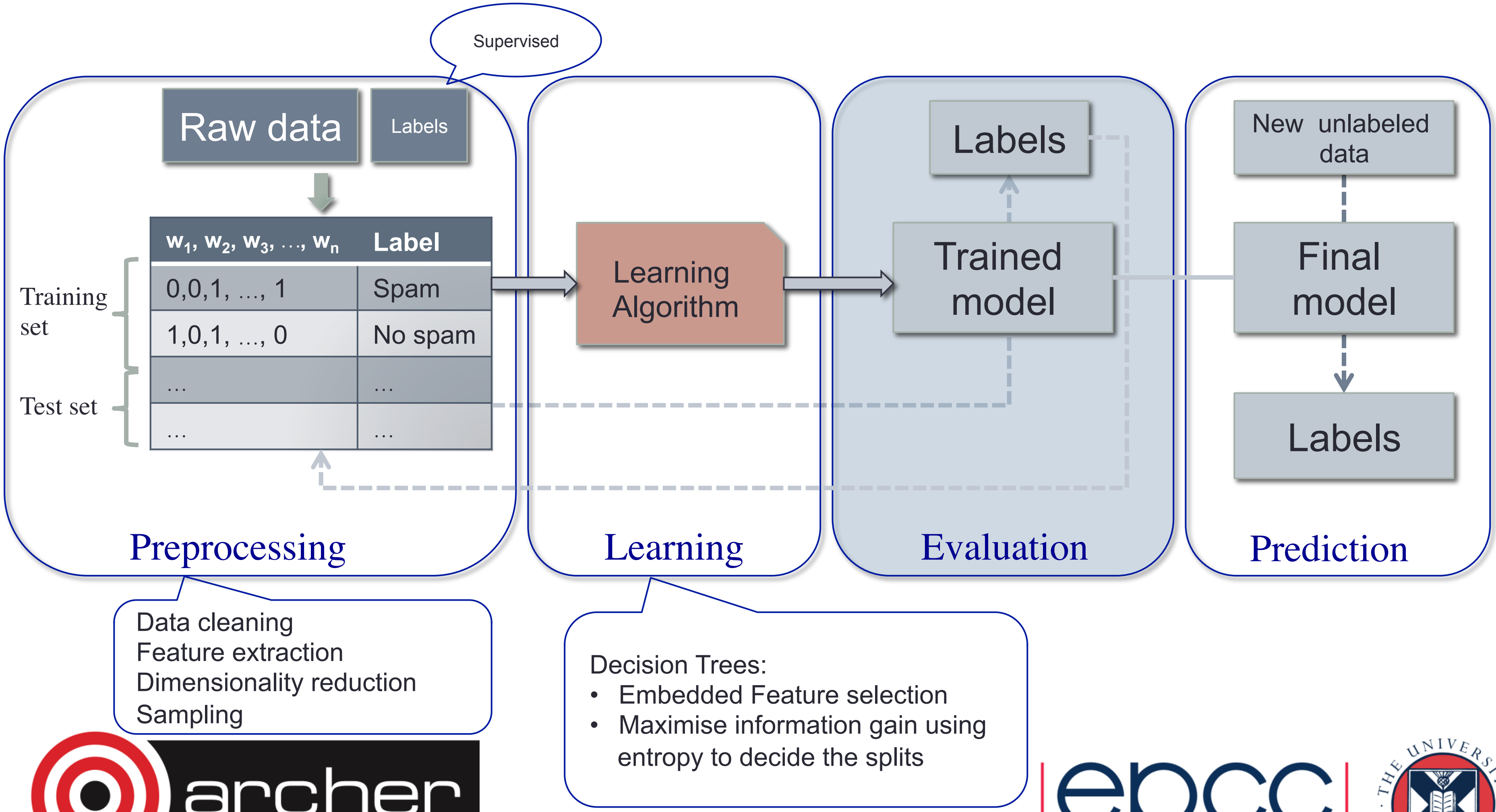
WinLevel 700/150

Window Level Presets

HPC Implications

- For a decision forest can adopt a task parallel approach where the trees are generated in parallel.
- Efficiencies can be made by using a tree-reduction pattern to combine results, *O(log N)*, rather than a gather-to-master approach, *O(N)*.
- Parallel algorithms exist for building single decision trees
  - These expect many sample and few features (e.g. social science)
  - Microarray data is typically many features and few samples
- Applying the model can also be parallelised
  - Each tree processed in parallel
  - Multiple cases processed in parallel
- GPUs can be used to apply model to images or 3D volumes

Supervised Machine Learning Classification



Data cleaning
Feature extraction
Dimensionality reduction
Sampling

Decision Trees:
- Embedded Feature selection
- Maximise information gain using entropy to decide the splits

Evaluation metrics

- Accuracy: fraction of correctly classified cases.
- For binary outcomes, we define TP, FP, FN, and TN as follows:

		Gold Standard	
		True	False
Test Outcome	True	True Positive (TP)	False Positive (FP)
	False	False Negative (FN)	True Negative (TN)

Binary Classification Statistics

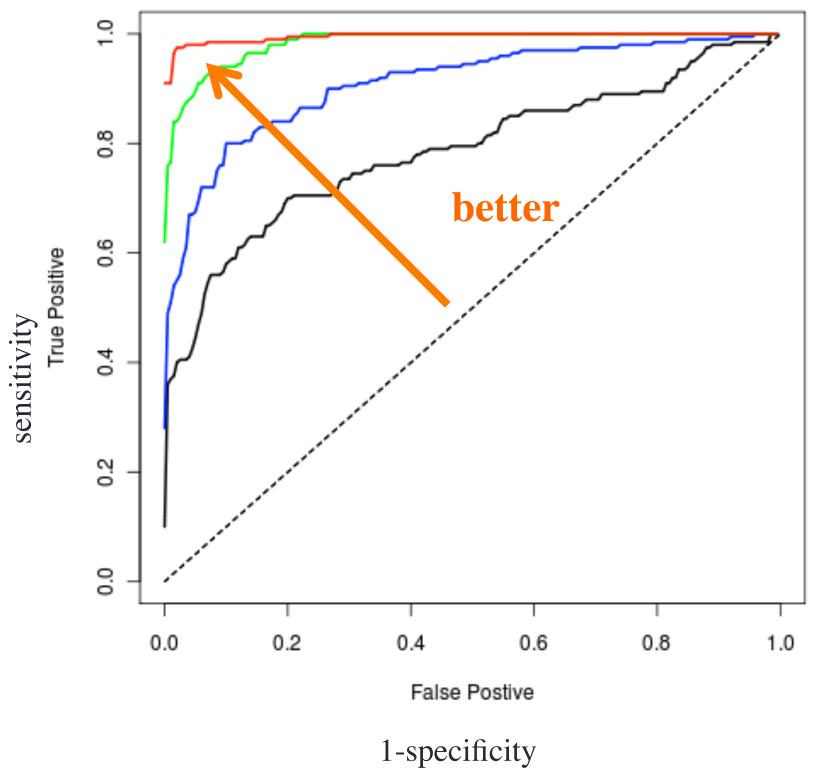
- **Accuracy**
  - (TP+TN)/(P+N) = (17+55)/100 = 0.72
- **Sensitivity** (**recall**, true positive rate): % of Positives predicted as being positive
  - TP/P = TP/(TP+FN) = 17/20 = 0.85
- **Specificity** (true negative rate): % of Negatives predicted as being negative
  - TN/N = TN/(FP+TN) = 55/80 = 0.69
- **Precision** (positive predictive value): % of predicted positives that are True Positives.
  - TP/(TP+FP) = 17/(17+25) =17/42 = 0.40
- **Negative predictive value**
  - TN/(TN+FN) = 55/58 = 0.94
- **F-score**: harmonic mean of precision and recall
  - (2 x precision x recall)/(precision + recall) = 0.548

		Gold Standard	
		True	False
Test	True	TP	FP
	False	FN	TN

	True	**False**	**Total**
True	17	25	42
False	3	55	58
Total	20	80	100

ROC curve

- Receiver operating characteristic (ROC) curve
  - Sensitivity against (1-specificity)
  - (1-specifity) = 1 - TN/(FP+TN) = False Positive rate = FP/(FP+TN)
- Point on curve dependent on tunable parameter in algorithm
  - On Random Forest, the tunable parameter is the proportion of the trees necessary to predict 1 (positive).
- If the ROC curve is close to the line y=x then the classification algorithm is as good as tossing a coin.
- The area under the curve (AUC) provides an evaluation measure of the classifier:
  - a AUC close to 1 implies a good classifier


sensitivity
True Positive
better
False Postive
1-specificity