# AMSI SUMMER SCHOOL 2017
# Modelling and Analysis of Big Data

Lecturer: Kerrie Mengersen, Professor of Statistics
Tutor: Miles McBain, Consultant Statistician

In collaboration with colleagues from BRAG and ACEMS
Acknowledgement: some content was written for QUT FL MOOC

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Introductions

Bayesian Research and Applications Group

Faculty of Science and Engineering

Institute for Future Environments

Queensland University of Technology

Centre of Excellence in Mathematical and Statistical Frontiers:
Big Data, Big Models, New Insights

Australia

QUT

ACEMS

AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR
MATHEMATICAL AND STATISTICAL FRONTIERS


# Australian Research Council Centre of Excellence

# Mathematical & Statistical Frontiers:

# Big Data, Big Models, New Insights

7 year horizon

6 Universities

7 Partner Organisations

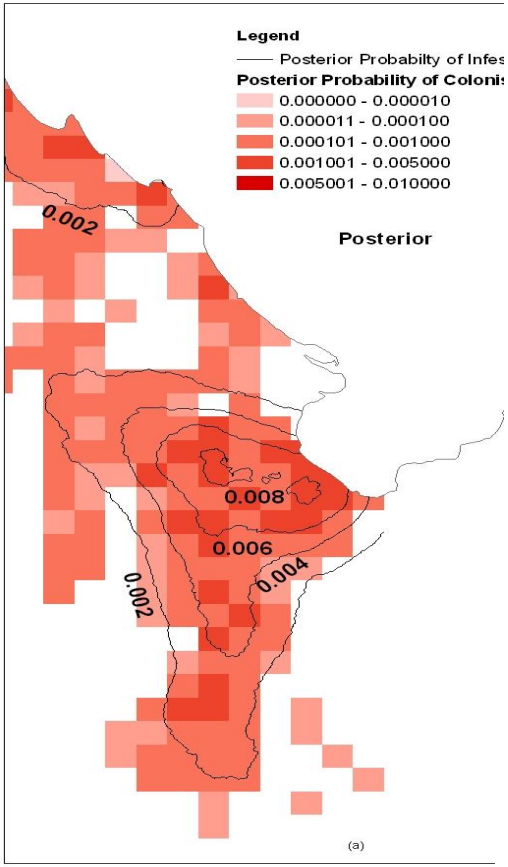
50 Professorial Investigators, >50 Researchers&students

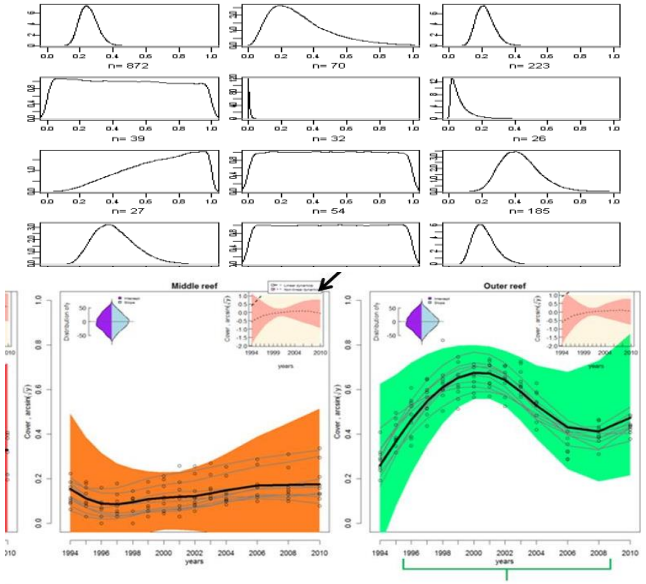
QUT

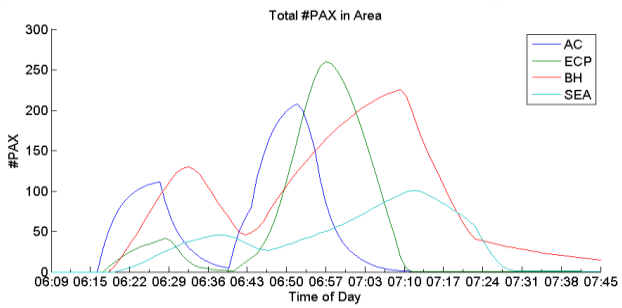
ACEMS

AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Our Interests at QUT: Modelling and Analysis



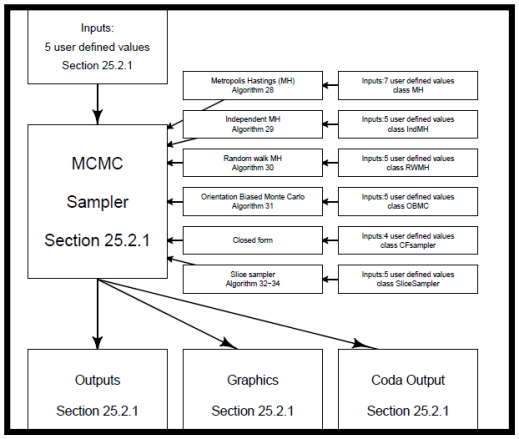



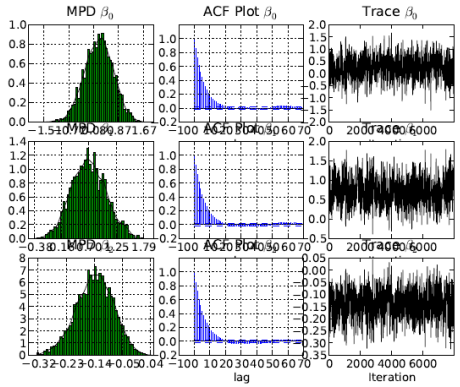

- Combining data sources
- Modelling with uncertainty
- Using prior information
- Probabilistic prediction
- Risk stratification
- Complex systems models

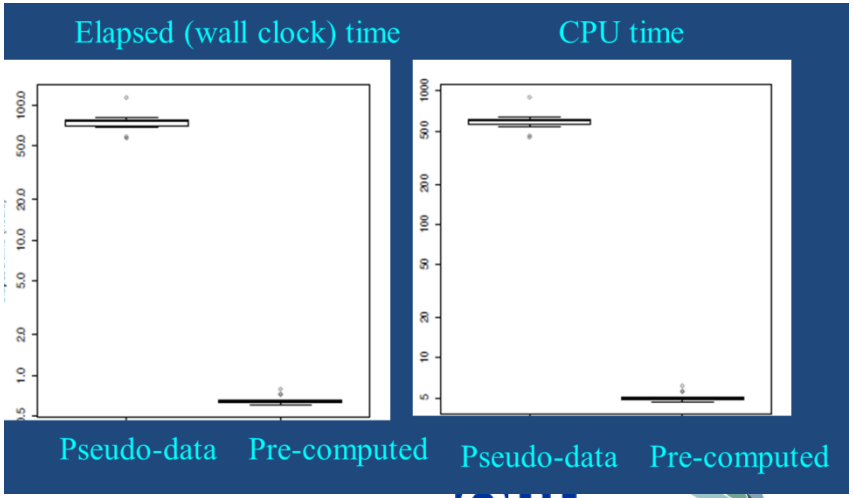
QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Our Interests at QUT: Fast computation



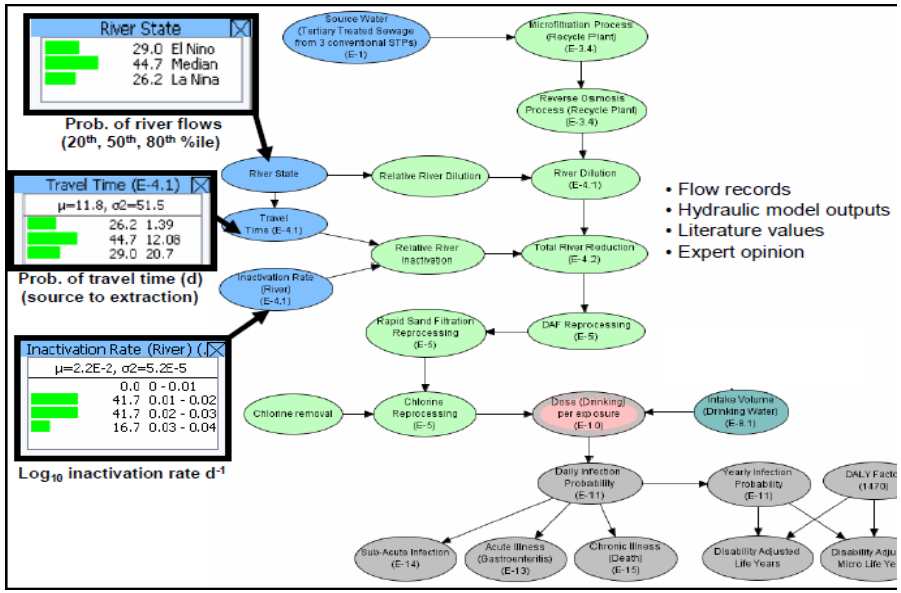




- Algorithms
- Approximations
- Ensembles
- Big data environments
- Platforms
- HPC

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR
MATHEMATICAL AND STATISTICAL FRONTIERS

# Our Interests at QUT: Systems approaches




| Indicator | Category | Farm | Factory | Market | Rating |
|---|---|---|---|---|---|
| Economic | Commodity prices | 0.8 | 0.6 | 0.6 | 0.7 |
| | Legal and administrative environment | 0.0 | 0.1 | 0.8 | 0.3 |
| | Access to capital and labour | 0.6 | 0.8 | 0.8 | 0.7 |
| | Profitability | 0.3 | 0.8 | 0.7 | 0.6 |
| | Workforce capability | 0.1 | 0.8 | 0.6 | 0.5 |
| | Economic sustainability rating | 0.4 | 0.6 | 0.7 | 0.6 |
| Social | Lifestyle and community | 0.0 | 0.5 | 0.1 | 0.3 |
| | Health and well being | 0.6 | 0.9 | 0.8 | 0.7 |
| | Value and contribution | 0.1 | 0.6 | | 0.6 |
| | Product, safety and production | 0.8 | 0.0 | 0.0 | 0.1 |
| | Social relevance | 0.6 | 0.8 | 0.8 | 0.7 |
| | Social sustainability rating | 0.4 | 0.6 | 0.3 | 0.5 |
| Environment | Energy, effluent and water | 0.6 | 0.2 | 0.2 | 0.3 |
| | Materials, suppliers and transport | 0.2 | 0.8 | | 0.6 |
| | Products and services | 0.8 | 0.8 | 0.2 | 0.6 |
| | Biodiversity | 0.2 | 0.8 | 0.6 | 0.6 |
| | Compliance | 0.2 | 0.0 | 0.0 | 0.1 |
| | Environment sustainability rating | 0.4 | 0.5 | 0.2 | 0.4 |
| Dairy Industry sustainability rating | | 0.4 | 0.6 | 0.4 | 0.5 |

- *G*raphical, probabilistic modelling of complex processes
- Multiple perspectives and stakeholders
- Varied information sources

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Examples of our BD interests

Ben Fitzpatrick video

Amy Cook video

Matthew Sutton video

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR
MATHEMATICAL AND STATISTICAL FRONTIERS

# Plan

1. Let's talk about "Big Data"
2. Methods for modelling and analysis of big data
3. Digging deeper: (1) classification
4. Digging deeper: (2) regression
5. Digging deeper: (3) clustering
6. Digging deeper: (4) dimension reduction
7. Case study: recommender systems
8. From the learning to the doing: tips and tricks

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Plan

1. Let's talk about "Big Data"
2. Methods for modelling and analysis of big data
3. Digging deeper: (1) classification
4. Digging deeper: (2) regression
5. Digging deeper: (3) clustering
6. Digging deeper: (4) dimension reduction
7. Case study: recommender systems
8. From the learning to the doing: tips and tricks

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Definitions of big data

**"volume, velocity, variety"**

**+ "veracity, value"**

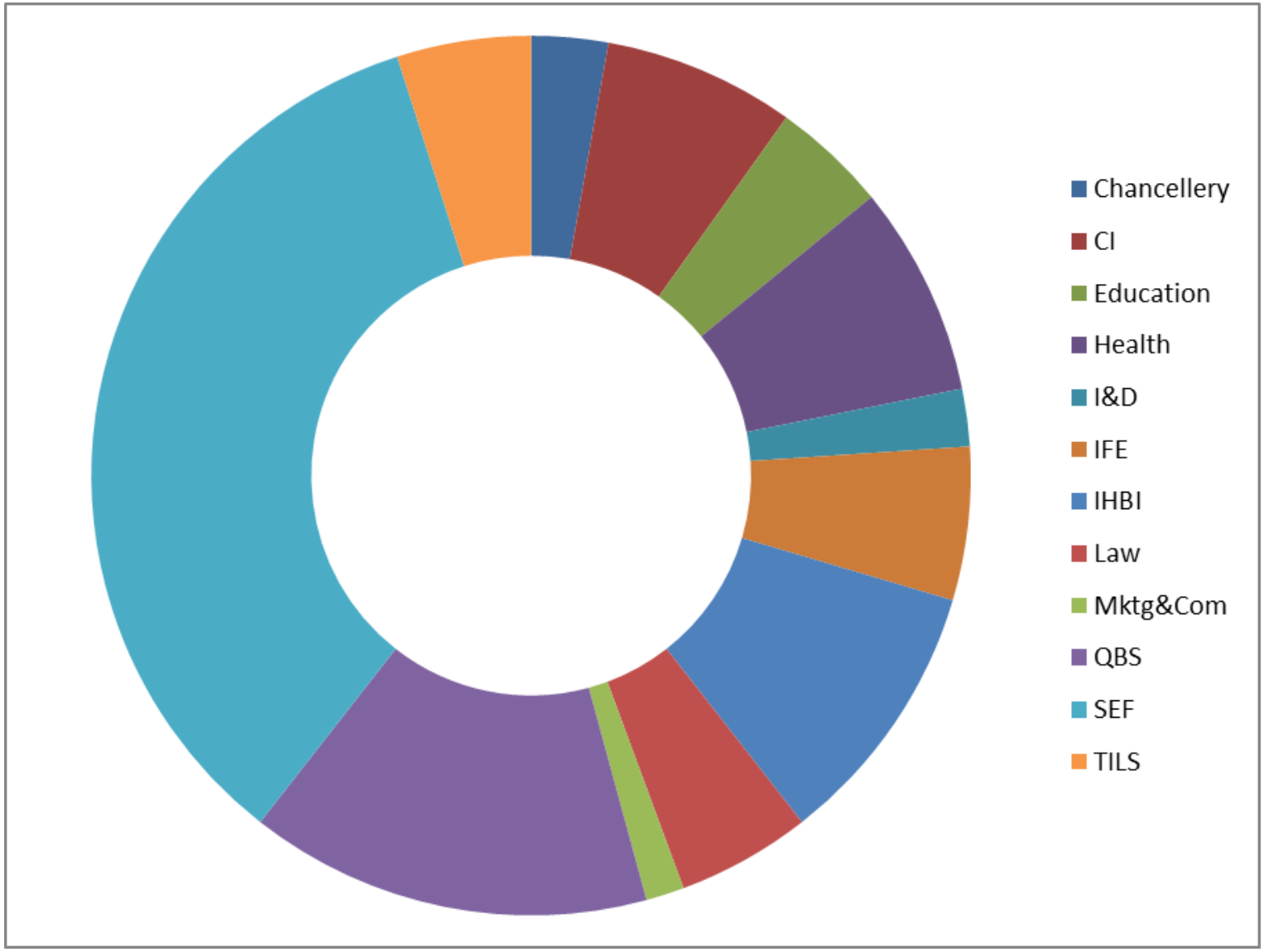
**+ "complexity, ..."**

**"inconveniently large"**

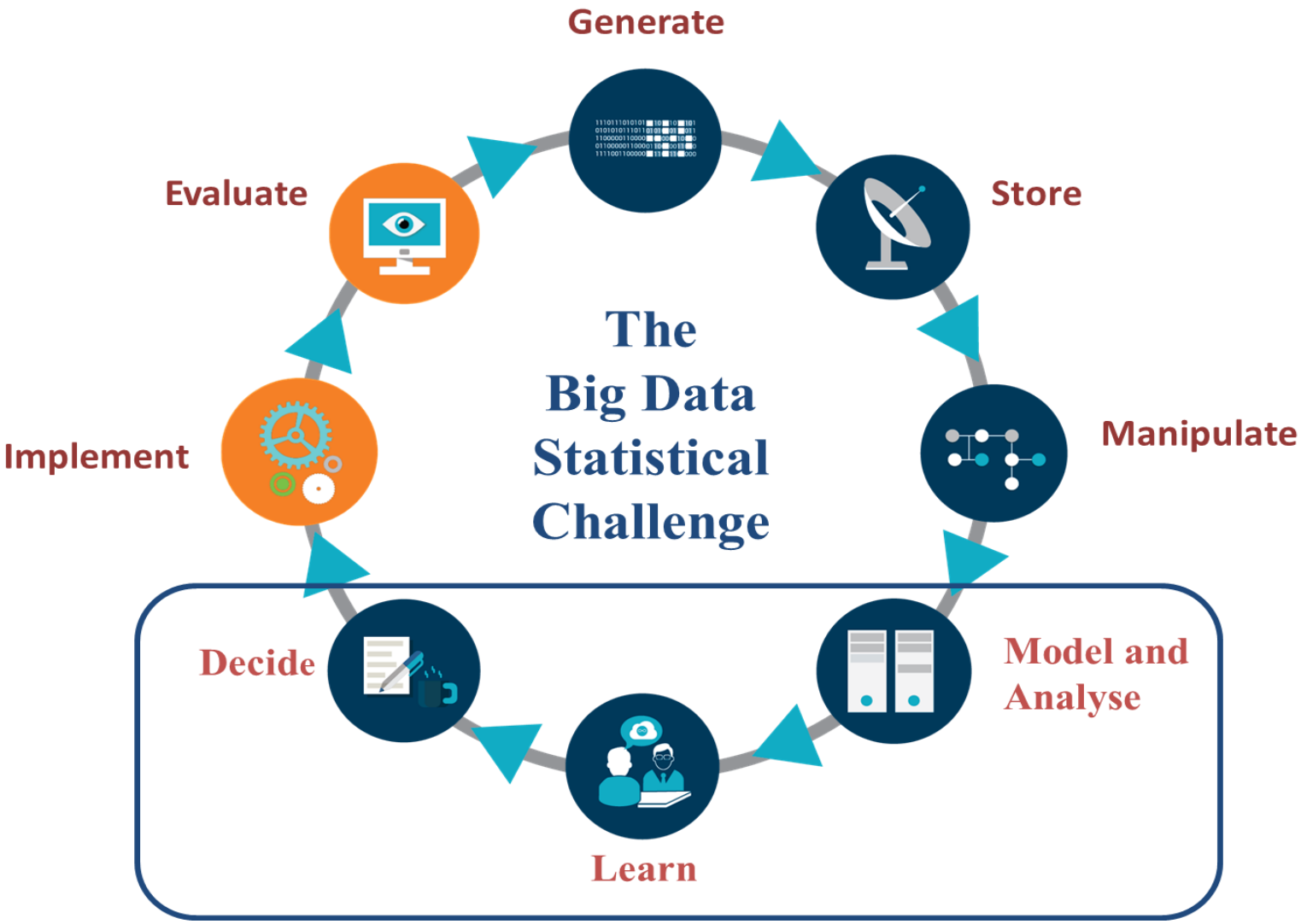
# Big data definitions

- **Volume**: data at rest
  - The amount of data
  - Data explosion problem
- **Variety**: data in many forms
  - Different types (structured, unstructured)
  - Data sources (internal, external, public)
  - Data resolutions
- **Velocity**: data in motion
  - Speed of data generation
  - Speed of data handling
- **Veracity**: data in doubt
  - The varying levels of noise and processing errors
- **Value**: cost of data

# “Big Data” at QUT

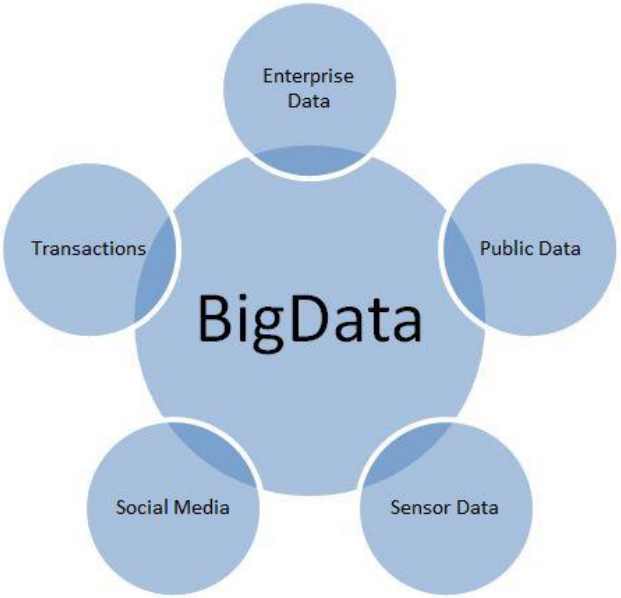


# The Big Data Wheel



# Sources of big data

- Sensors
- Transactions
- Enterprise data
- Social media
- Citizen science
- Spatial data




QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Sources of big data

Data are increasingly gathered by cheap and numerous information-sensing mobile devices, aerial (remote sensing), software logs, cameras, microphones, radio-frequency identification (RFID) readers and wireless sensor networks.

https://en.wikipedia.org/wiki/Big_data

# Sources of big data

## Every minute of every day in 2014:

>204 million email messages

> 2 million Google search queries

48 hours of new youtube videos

>$280,000 spent on e-commerce

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR
MATHEMATICAL AND STATISTICAL FRONTIERS

# Digging into data

## Characteristics of:

1. Long / Wide data
2. Image data
3. Real time data
4. Streaming data
5. Internet data
6. Social media data
7. Internet of Things

# Digging into data

Streaming data

http://politicaldatascience.blogspot.com.au/2015/12/rtutorial-using-r-to-harvest-twitter.html

Image data

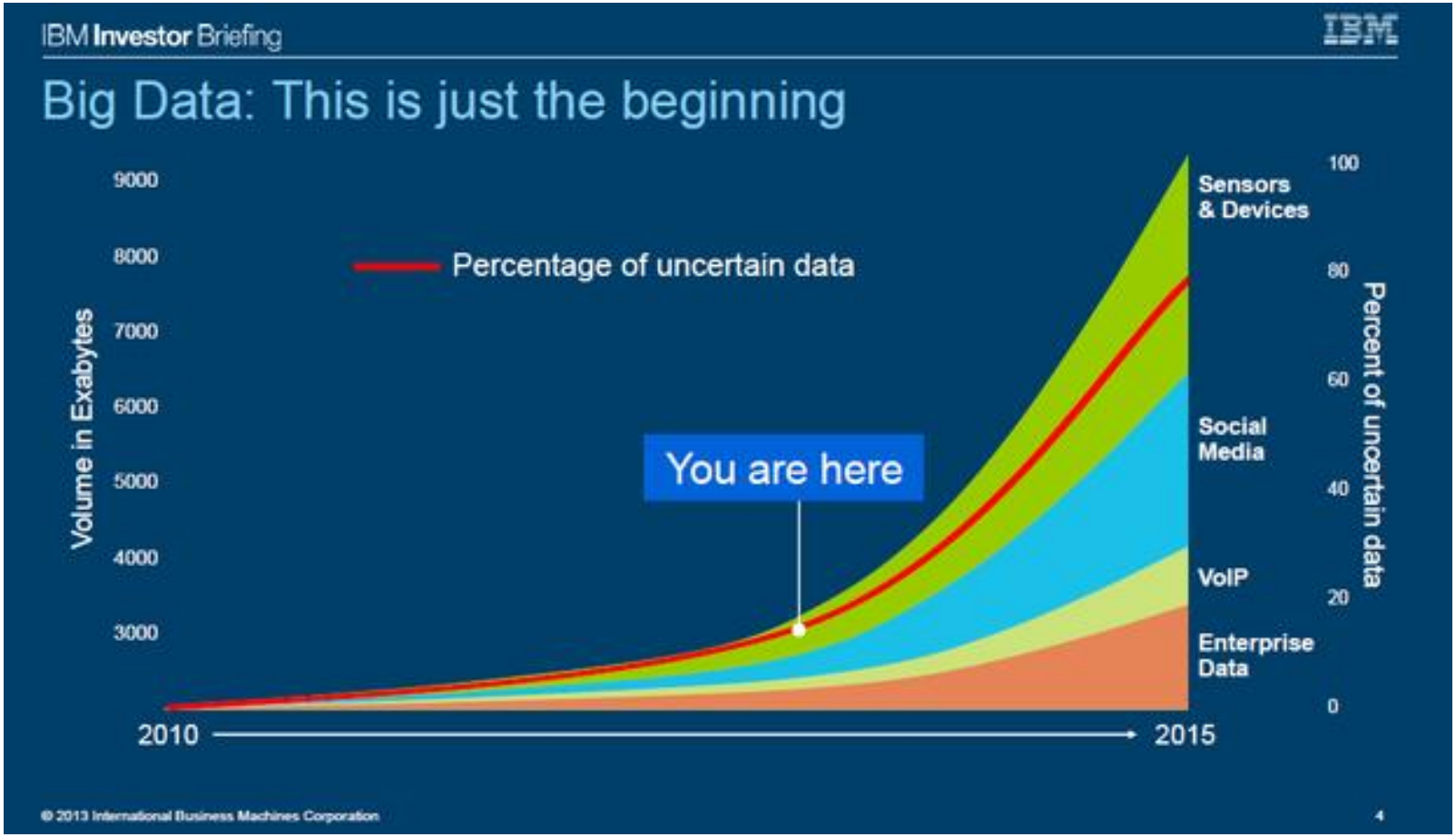
http://neondataskills.org/R/

# Dealing with dirty data

## How to cope with:

- Size
- Quality
- Diversity
- Uncertainty

# Big data = big uncertainty (IBM)




QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# From dirty to tidy data

## Miles: personal experiences

## Journal of Statistical Science: Hadley Wickham

- What is “tidy data”?
- How can we create tidy data?
- What issues arise in creating tidy data?

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# More broadly: challenges in big data manipulation

- Storage
- Curation (data quality)
- Searching and querying
- Transfer and sharing
- Updating
- Information privacy
- Ownership and use

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Discussion

## Big data Rap

- https://www.youtube.com/watch?v=PI7SLOovO5c

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Discussion

## Big Data Literature

- This website presents some of the most informative and well-known papers in the Big Data field:
  - http://bigdata-madesimple.com/research-papers-that-changed-the-world-of-big-data/

- There are also papers that give general overviews of BD, eg:
  - The promise and peril of Big Data. http://www.aspeninstitute.org/sites/default/files/content/docs/pubs/The_Promise_and_Peril_of_Big_Data.pdf

# Discussion

## BD doing cool things

BD for catching tiger poachers
http://www.huffingtonpost.com/2015/01/12/india-wild-tigers-big-data_n_6458386.html

BD wrote a movie script
http://arstechnica.com/the-multiverse/2016/06/an-ai-wrote-this-movie-and-its-strangely-moving/

- The movie https://www.youtube.com/watch?v=LY7x2Ihqjmc

BD wrote a song http://www.theverge.com/2016/9/26/13055938/ai-pop-song-daddys-car-sony

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Discussion

Big data and Pokemon
https://pixelastic.github.io/pokemonorbigdata/

Dirty data and tidy data

Wickham (2014):
https://www.jstatsoft.org/article/view/v059i10

# More on using R

If you haven't done any coding in R, you might like to TryR. This is a good interactive tutorial made available by CodeSchool which doesn't require any downloads.

https://www.codeschool.com/courses/try-r

Here's a document about setting up R in the http://clostrimas.com/r/rstudio-cloud-1/

# Practical Session

- Summary of prac

# Plan

1. Let's talk about "Big Data"
2. Methods for modelling and analysis of big data
3. Digging deeper: (1) classification
4. Digging deeper: (2) regression
5. Digging deeper: (3) clustering
6. Digging deeper: (4) dimension reduction
7. Case study: recommender systems
8. From the learning to the doing: tips and tricks

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# BD modelling platforms

1. Empirical

   a statistical relationship is established between the spectral bands or frequencies used and the variable measured (field-based) without there necessarily being a causal relationship.

2. Semi-empirical

   combine knowledge about the process with statistical models. This knowledge can be about the process itself or about the variables that are included in the process.

3. Physics-based

   physics based inversion models include all required variables assessed in one spectral inversion simultaneously. This method provides physics-based consistency in results, and is most suitable for automation across large areas, provided the inversion model is properly parameterized for the desired variables.

4. Object-based

   Methods Incorporating Spatial Dependencies : field-level aggregation/object oriented approaches, contextual-based approaches, serial or parallel multiple classifiers

5. Machine-learning approaches

   also called artificial intelligence methods, use algorithmic models to analyse data. These methods treat the way in which the data were generated or relationships between the variables as unknown.

# BD analysis platforms

- Apache Mahout
- Apache Spark and MMLib
- TensorFlow
- Tableau
- Domo
- H2O
- H2O extensions: Sparkling Water, Steam

**Source**: CSIRO Big Data Workshop 2014

# Analytics Software (SelectHub)

- Oracle, IBM InfoSphere, SAP HANA, SAS High Performance Analytics, HPE Vertica, Microsoft Cloud Platform, Google Analytics and Google Big Query
- SPSS, SAS, Statistica, Revolution R Enterprise
- Inetsoft Style, Targit, Alteryx, MicroStrategy, BIT, Board, Domo, WebFOCUS, Jaspersoft, Alteryx, Arcplan Enterprise, MicroStrategy Analytics, Logi Analytics, Panorama Necto 16 Advanced Analytics, GoodData, TIBCO Spotfire, SiSense Prism, Teradata Aster Database, WaLa!, 1010 Big Data Discovery, InfiniDB database, MoData Smart Data Discovery Platform, Birst, SlamData, QlikView, Tableau, Domo, H2O

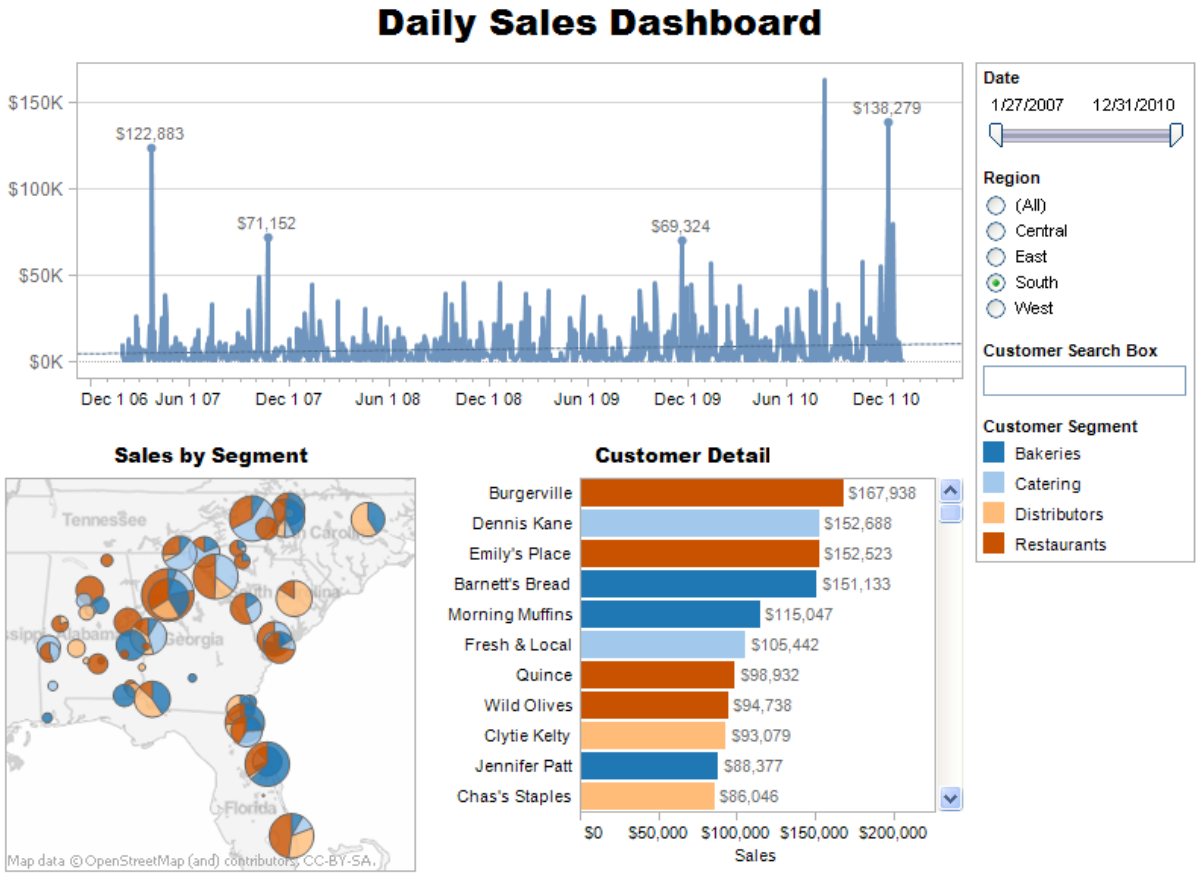
QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Analytics Platforms

## Tableau

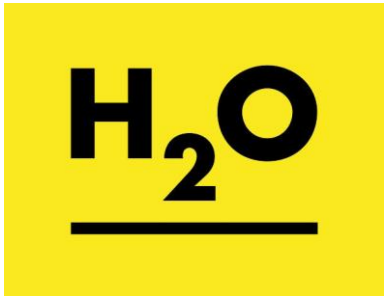
- Hadoop
- Cloudera Impala
- Cassandra
- HortonWorks
- Karmasphere
- Google Analytics
- Google BigQuery
- Teradata
- Amazon RedShift
- SAP HANA
- MySQL, PostgreSQL
- Many other data source




QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# $H_2O$



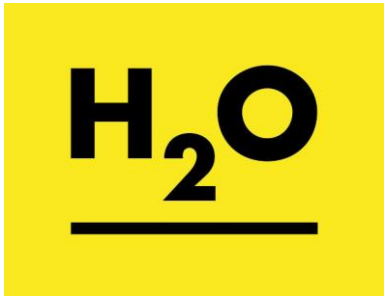

*"H2O is the world's fastest in-memory platform for machine learning and predictive analytics on big data."*

- Open source big data platform for machine learning and predictive analytics
- Web-based user interface and familiar programming environments like R, Java, Scala, Python, JSON and tools like Excel or Tableau, e.g. develop R code on a laptop and easily scale to a cluster
- Algorithms make use of of Hadoop and map reduce … or Spark
- The data structure looks like a data frame
- Buy in from the prominent people in stats community (Tibshirani, Hastie, Chambers … )

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# What $H_2O$ does




- Regression
  - GLMnet (Gaussian, Binomial, Poisson, Gamma)
  - Bayesian Regression
  - Multinomial Regression
- Classification
  - Distributed Random Forest
  - Gradient Boosting Machine
  - Naïve Bayes
  - Distributed Trees
- Neural Networks
  - Multi-Layer Perceptron
  - Auto-encoder
  - Deep Learning
  - Restricted Boltzmann Machines
- Clustering
  - K-Means, K-Nearest Neighbours
  - Singular Value Decomposition
  - Dimensionality Reduction
  - Local Sensitivity Hashing
- Time Series
  - ARIMA, ARMA Modelling
  - Forecasting
- Solvers and Optimisation
  - Ordinary Least-Squares Solver
  - Stochastic Gradient Descent
  - MCMC
  - Generalised ADMM Solver

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

Spark + $H_2O$

SPARKLING WATER

# Example

Dataset of patients with varying degrees of prostate cancer.

CAPSULE (binary variable indicating tumor penetration of prostatic capsule), AGE (in years), RACE (1 = white, 2 = black), PSA (prostatic-specific antigen value), and GLEASON (total gleason score, indicating how aggressive the cancer is).

| ID | CAPSULE | AGE | RACE | DPROS | DCAPS | PSA | VOL | GLEASON |
|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 65 | 1 | 2 | 1 | 1.4 | 0.0 | 6 |
| 2 | 0 | 72 | 1 | 3 | 2 | 6.7 | 0.0 | 7 |
| 3 | 0 | 70 | 1 | 1 | 2 | 4.9 | 0.0 | 6 |
| 4 | 0 | 76 | 2 | 2 | 1 | 51.2 | 20.0 | 7 |
| 5 | 0 | 69 | 1 | 1 | 1 | 12.3 | 55.9 | 6 |
| 6 | 1 | 71 | 1 | 3 | 2 | 3.3 | 0.0 | 8 |

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# H2O in practice: logistic regression

```
h2o.glm(y = "CAPSULE", x = c("AGE","RACE","PSA","GLEASON"), data =
prostate.data, family = "binomial", nfolds = 10, alpha = 0.5)
```

https://www.r-bloggers.com/diving-into-h2o/
http://blog.h2o.ai/2013/08/run-h2o-from-within-r/

Coefficients:

| AGE | RACE | PSA | GLEASON | Intercept |
|---|---|---|---|---|
| -0.02119 | -0.46410 | 0.02804 | 1.07613 | -5.86616 |

Degrees of Freedom: 379 Total (i.e. Null); 374 Residual

Null Deviance: 512.3

Residual Deviance: 416.3

AIC: 426.3

The log-odds (and by extension, probability) of prostate capsular penetration increase with PSA and gleason score, as expected, but decrease slightly with age. A patient who is black is significantly less likely to exhibit capsular involvement than one who is white, although it is unknown whether this is a direct effect, or whether race is capturing some other characteristic excluded from the regression.

MSE:  0.2027036
RMSE:  0.4502261
LogLoss:  0.5914634
Mean Per-Class Error:  0.3826121
AUC:  0.717601
Gini:  0.435202
R^2:  0.1572256

Null Deviance:  512.2888
Residual Deviance:  449.5122
AIC:  459.5122

Confusion Matrix for F1-optimal threshold:

| | 0 | 1 | Error | Rate |
|---|---|---|---|---|
| 0 | 80 | 147 | 0.647577 | =147/227 |
| 1 | 18 | 135 | 0.117647 | =18/153 |
| Totals | 98 | 282 | 0.434211 | =165/380 |

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# H2O in practice: k-means clustering

```
> prostate.km = h2o.kmeans(data = prostate.data, centers = 5,
cols = c("AGE","RACE","GLEASON","CAPSULE","PSA"))
> print(prostate.km)
```

K-means clustering with 5 clusters of sizes 278, 4, 23, 69, 6

Cluster means:

| | AGE | RACE | GLEASON | CAPSULE | PSA |
|---|---|---|---|---|---|
| 1 | 66.14947 | 1.071174 | 6.124555 | 0.3060498 | 7.107402 |
| 2 | 65.75000 | 1.250000 | 8.000000 | 1.0000000 | 131.175000 |
| 3 | 66.09091 | 1.227273 | 7.136364 | 0.7272727 | 55.213636 |
| 4 | 65.44776 | 1.089552 | 7.014925 | 0.6119403 | 23.876119 |
| 5 | 67.50000 | 1.166667 | 7.666667 | 1.0000000 | 86.500000 |

From our results, we see that 278 patients – the large majority – are in category 1, with age close to 66 years and only about 30% exhibiting capsular penetration. The PSA and gleason score of this cluster are by far the lowest. In contrast, category 2 is the smallest cluster, with only 4 patients, but they all show capsular penetration and, as expected, far higher gleason scores and PSA. Clearly, k-means was correct in categorizing these patients into separate groups.

# Choosing Software Packages

- Questions to ask:
  - What do you want to do?
  - How do you want to do it? In-house, cloud, etc
  - What are you allowed to do?
  - What will you want to do in the future?

Eg: Tableau is a leader in visualisation

QlikView is a leader in visualisation and (summary) analytics
https://www.qlik.com/us/

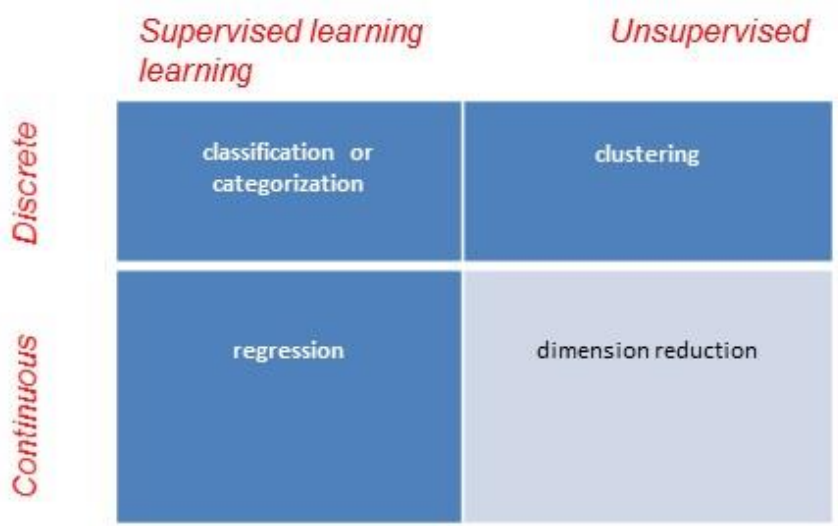
H2O is free and links to R.

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Stats & ML approaches

- Four common analytic problems: **classification, regression, clustering** and **dimension reduction.**
- Broadly categorised by whether the output is **continuous** or **discrete** (classes) and whether it is **supervised** (includes desired outputs) or **unsupervised** (doesn't include desired outputs).




QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Discrete output & supervised learning: Classification

- The overall aim is to accurately allocate objects to a discrete (usually small) set of known classes, or groups.
- This allocation is based on a set of input variables (also called explanatory variables, factors, predictor variables, independent variables, covariates or attributes).
- An example is the development of a diagnostic test, which declares a person to be of class ‘diseased’ or ‘healthy’, based on a set of clinical variables (blood measurements, medical observations, etc). Another example is prediction of ‘rain’ or ‘no rain’ based on a set of meteorological variables.

# Discrete output & supervised learning: Classification

- Classification approaches are typically ‘trained’ or developed using a ‘training set’, in which both the input variables and the class labels (which indicate the class to which each object belongs) are known. We then use the analytic approach to classify other objects in a ‘test set’, for which we have the input variables but not the class labels.

- Eg, develop diagnostic test based on patients whom we know are either ‘diseased’ or ‘healthy’, then apply the test to other patients to predict their disease status.

# Classification: common approaches

**Decision trees (CART)**

- Split the objects in the dataset into two groups, then split each of these groups into subgroups, and so on, based on the input data.
- The result is a tree-like structure, with branches depicting the splits and the final (terminal) nodes depicting the satisfactorily homogeneous groups.
- These nodes can then then be used to predict a new data point by following the branches tree to the final group.
- There are many types of decision trees, from CART (classification and regression trees), in which only one tree is constructed, to boosted regression trees and random forests, in which an ensemble of trees is developed and combined to give overall better predictions.

# Classification: common approaches

**Bayesian networks (BN)**

- Develop a set of relationships between the input variables (attributes) and the output variable (response) using the conditional dependence of each attribute on each other attribute.
- The result is like a ‘mind map’ showing attributes as nodes and dependencies between the attributes as directed arrows that link the nodes.
- This allows us to encode prior knowledge or assumptions about our data into our model, by specifying the values for some of the nodes.
- The network can tell us about the relative importance of variables in influencing the response, and what happens to the response if we change certain input variables, given all of the other attributes that are influencing it.

# Classification: common approaches

- **K-nearest neighbour (KNN)** – Predict the class to which an object belongs by using the majority vote of its K nearest neighbouring data points.
- **Support Vector Machines (SVM)** – Find a linear combination of the input variables that separates the classes as well as maximise the distance of every point away from this line.
- **Logistic and multinomial regression** – Develop a weighted combination of the input variables to optimally differentiate between the classes.

# Discrete output & unsupervised learning: Clustering

- The overall aim is to combine objects into groups or classes based on a set of input variables.
- Unlike classification, we don't know the classes, even in the 'training set'. Therefore we need to work out a measure of similarity between the objects and a way of grouping them according to these similarities.
- We might specify the number of groups (clusters), or also make unknown and estimate the number of groups as part of the analysis. The analysis can be used to make decisions about the objects that were clustered, or to predict cluster membership for new objects.

# Discrete output & unsupervised learning: Clustering (2)

- An example is the allocation of customers into groups based on their responses to a satisfaction survey. We can then inspect these groupings and the customer traits that describe the groups and differentiate between them. This can be used to manage the existing customers or predict satisfaction of new customers.

# Clustering: common approaches

- **K-means** - This algorithm assumes the data is drawn from K different clusters and assigns each unlabelled points to the closest group centre which are recalculated until no changes occur.
- **Agglomerative clustering** - Start with each point as its own cluster and iteratively merge the closest clusters.
- **Mixture models** – Allocate the objects to groups (and determine the number of groups if necessary) based on an underlying model that describes the groups. For example, the groups might be considered to be normally distributed and the mixture model is then seen as a weighted combination of these distributions, where the weights reflect the proportion of objects that are classified to that group. Instead of allocating an object to only one group, it is allocated probabilistically to groups, so there can be 20% chance of belonging to one group and 80% chance of belonging to the other group.

# Continuous output & supervised learning: Regression

- Regression methods are similar to classification methods, but the output variable (response) is continuous instead of categorical (a set of classes).
- The aim is to accurately and precisely estimate or predict the response, given a set of input variables (explanatory variables, attributes).
- The regression model is 'trained' using a set of objects for which the response is known. The analyst might be interested in the estimated values for the objects in the training set, predicting responses for new objects, identifying which input variables are most important in making good predictions, or inspecting the relationships between these variables.

# Continuous output & supervised learning: Regression (2)

- An example of regression is accurately predicting the result of a diagnostic test given a patient's medical data, but this time the test is a continuous score, rather than simply 'healthy' or 'diseased' as in the classification problem discussed above.
- Another example is predicting the amount of rainfall (e.g., in mm), instead of simply whether it will rain or not.

# Regression: common approaches

- **Linear methods** – Construct a weighted linear combination of the input variables (or functions of these variables) that provide the best predictions of the output variable. The aim is to estimate the values of the weights, or coefficients, that provide the most accurate and precise predictions.
- **Regression Trees** – These trees are constructed using the input variables, but here the aim is to minimise the difference in responses within the groups and maximise the difference in average response between the groups. The tree will then predict the expected response for a new object, based on following the branches of the tree to the final (terminal) node and calculating the average value of the responses in that group of objects. The uncertainty of the prediction can also be estimated by the variance of the responses in this group. There are many different kinds of regression trees, including Boosted Regression Trees (BRT) and Random Forests (RF).

# Regression: common approaches

- **Gradient Boosted Machines** – Add various regression trees together where each next tree is trained on the errors of the previous trees added together. Use the sum of these trees to predict the value of new data points.
- **Neural networks (NN)** – Train one or more layers of non-linear functions which map the input variables to the output variable. These layers of functions emulate the neural networks in our brains, and how we learn. The network is 'trained' on a set of data with both input data and the response. This network can then be used for estimation of the objects in the training dataset, or prediction of the responses of new objects in a test dataset (which has input variables but no output variables) by feeding in the input variables, flowing them through the network, and obtaining the predicted outputs at the end.

# Continuous output & unsupervised learning: Dimension reduction

- The aim is to construct an output variable (or set of variables) based on a set of input variables, where this output variable is unknown. The output variable(s) should be continuous (otherwise it would be a clustering problem) and the new output variables should maximise the information in the data.
- Dimension reduction is a common tool in Big Data Analytics, since it can be used to create a small set of output variables that can effectively carry all of the information in a very large set of input variables. The analyst can then inspect these new variables to see which of the original variables are most important in explaining the variation in the data. The new variables can also be used as inputs to regression, clustering and classification problems.

# Continuous output & unsupervised learning: Dimension reduction (2)

Another common use of dimension reduction is in the creation of indices. E.g., an economic index, a weather index, a health index, psychological index.

These single numbers are actually made up of combinations of many variables. They are sometimes ‘named’ based on the dominant variable in the index.

e.g., a psychological index might be constructed from the questions in a psychological assessment, and if the most important variables in this index are related to work ethic, the index might be called the Work Ethic Index. A person who completes the assessment and gets a high score on this index might then be labelled as a ‘Hard Worker’, for example.

# Dimension reduction: common approaches

- **Principal Component Analysis (PCA)** – Converts the input variables into a set of uncorrelated output variables by projecting them onto a new set of axes. The number of principal components will be the same as the number of input variables, but hopefully only a few of these will have substantial weight attached to them, so this reduced number can be used as surrogates for the original data. The advantage is that they are uncorrelated so they individually contribute substantial information.
- **Factor Analysis (FA)** – Like Principal Component Analysis, but the axes are 'rotated' so that the new output variables (indices) are dominated by only one or two of the original variables. This means that they are more 'interpretable' (and hence can be 'named' if wished) but this comes at the cost that they are no longer uncorrelated.

# Passive or active learning

- In Machine Learning, passive learning involves algorithms used to infer from a given, static data set. The algorithm does not interact with its environment or choose data points.
- This is closely related to statistical methods of estimation, prediction, dimension reduction and clustering, when these approaches utilise a static dataset. Also optimal experimental and sampling design, when the design is specified at the beginning of the data collection phase of the study and does not change as the data are obtained.
- Active learning can be thought of as agents which are able to choose or influence which data points are chosen or made available. This problem of choosing which data points to evaluate for learning is a major part of machine learning, creating subfields such as reinforcement learning. It is also related to adaptive experimental design in statistics.

# Active Learning

- A recommendation system chooses a product for a customer based on features of that person and is rewarded if the customer accepts the recommendation.

- 'Google Now' infers patterns and asks questions based on these. It receives feedback based on your responses.

- A reinforcement learning agent is embedded in an environment which it can influence through some pre-defined actions. It then receives a feedback for this action in the form of a reward and state/observation signal which influences what actions it chooses in the future. This action observation reward cycle takes place each time step.

# Active Learning (cont)

- This setup is very general and can express many possible environments. In order for an agent to undertake a given task we need only choose the rewards appropriate to that task. This choice of utility along with environment together define the task at hand. Although this choice may be clear in many cases, it is non-trivial in general. For example, in creating general artificial intelligence the question of what kind of utility would define human intelligence often arises. Some argue that our utility would need to be immensely complex and subjective to account for our hugely varied values and emotional range. Others argue that a simple utility could emergently give rise to all these features and differences simply due to our varied experiences.

- In order to extract the maximum reward from its environment an agent must make good predictions about future observations as well as plan its actions according to those predictions. Reinforcement learning can be broken down into these two parts conceptually.

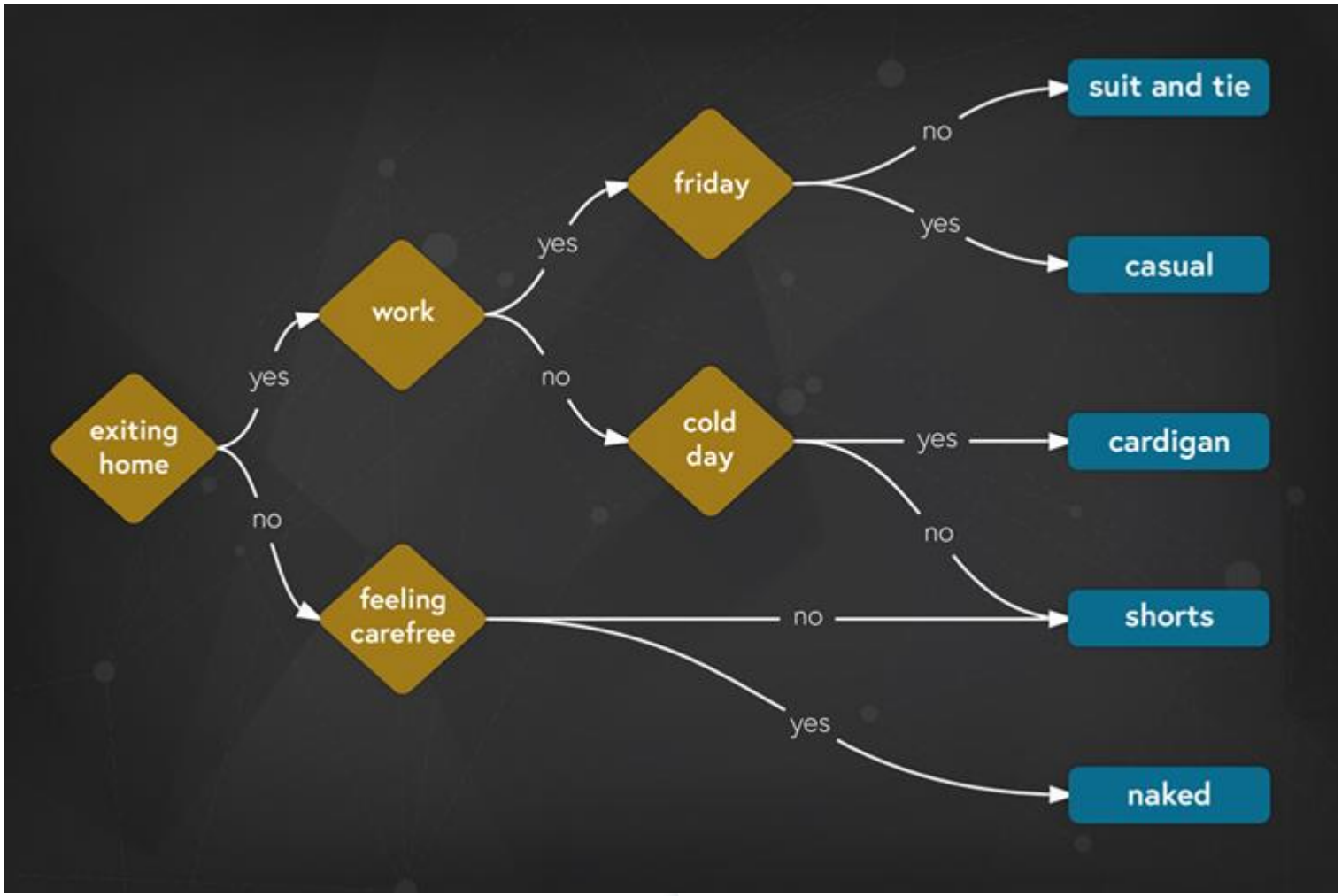
# Quick question

Think about a problem, system or process.

Think about a data collection process and analysis that would be classified as:

- passive learning
- active learning.

# “What will I wear?”




QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Big Data Analytics: top 10 algorithms

- Xindong Wu · Vipin Kumar · J. Ross Quinlan · Joydeep Ghosh · Qiang Yang · Hiroshi Motoda · Geoffrey J. McLachlan · Angus Ng · Bing Liu · Philip S. Yu · Zhi-Hua Zhou · Michael Steinbach · David J. Hand · Dan Steinberg (2008) Top 10 algorithms in data mining. Knowl Inf Syst 14:1–37. DOI 10.1007/s10115-007-0114-2.
- http://www.cs.uvm.edu/~icdm/algorithms/10Algorithms-08.pdf

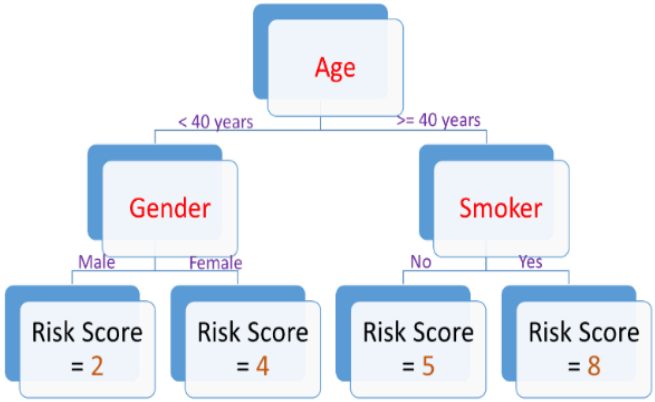
- These techniques are used for classification, clustering, statistical learning, association analysis and link mining.

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# CART, C4.5, See5, C5.0

- Classification and regression algorithms: predict the class to which a case belongs, or the expected value (and variance) of a group, based on a set of attributes.
- Two main steps: (i) grow a tree using a divide-and-conquer algorithm, which recursively splits the set of cases into two subgroups based on a splitting rule (find the attribute, and a threshold value of that attribute that minimizes diversity within the subgroups or maximizes information gain); (ii) prune the initial tree to avoid overfitting, based on the classification error rate.
- See5/C5.0 is a commercial system that is more scalable for big data (by multi-threading, enabling use of multiple cores), improves the predictive accuracy of C4.5 by including boosting (which constructs an ensemble of classifiers and votes on the final classification) and unordered rulesets, and allows for different data types.
- Other tree-based methods include boosted regression trees and random forests.




QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Boosted regression tree

- A BRT is a combination of single regression trees.
- The trees are fitted in a sequential manner, such that each tree is fitted to the residual of the trees that preceded it.
- In this sense, the BRT is similar to an additive model in regression.

An excellent article on Boosted Regression Trees by J. Elith, J.R. Leathwick and T. Hastie is available at:

http://onlinelibrary.wiley.com/doi/10.1111/j.1365-2656.2008.01390.x/pdf

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Random forests

- A RF is also a collection of trees (hence the name ‘forest’!).
- Each tree is built from a sample of the data. The output of a RF is the model of the classes (for classification) or the mean prediction (for regression) of the individual trees.

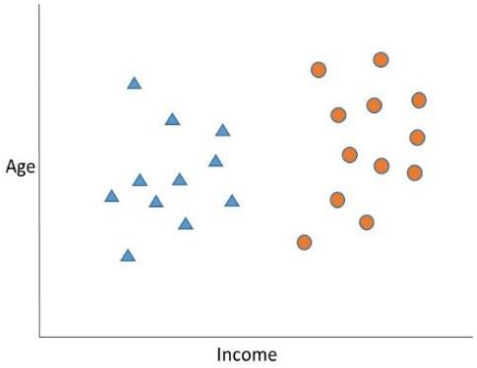
- There are different approaches to constructing the collection of trees in a RF. One popular approach builds each tree from a random sample of the objects (rows of the dataset); the sample is taken with replacement and is the same size as the original dataset. This approach is called ‘bagging’.
- Another approach builds trees based on a random sample of the predictor variables (features, i.e. the columns of the dataset). This is called ‘feature bagging’.
- Another approach takes a random sample of both the features and the objects (the rows and the columns).
- There are also ‘extremely randomised trees’ in which the variable used to construct the initial split of the tree is also randomly selected.

- All of these approaches aim to ‘mix up’ the data and provide opportunities to find an improved predictive model. This is very important for big data, with many explanatory variables that can be highly correlated and interact in nonlinear ways with the target variable.

# k-means

- Clustering algorithm: aims to partition a given dataset into a user specified number of clusters, k. If the desired k is not known in advance, one will typically run k-means with different values of k, and then select a k based on some criterion.
- Like CART and C4.5, the algorithm uses a set of attributes from a sample of cases (individuals, patients, etc) and predicts the class, or cluster, to which a case belongs. However, unlike CART and C4.5, the clusters are typically unknown.
- The algorithm starts by picking k points as the initial k cluster means or "centroids". These points can be chosen at random, or using a small subset of the data, or perturbing the global mean. Then the algorithm iterates between assigning each data point to its closest centroid (e.g., using Euclidean distance), and estimating the cluster means: re-assign, re-estimate, etc until there is no further change.
- Other algorithms such as agglomeration or hierarchical clustering can be included to allow for more complex cluster shapes.
- With an appropriate assignment rules, the k-means algorithm is scalable to big data. Variations such as kd-trees are also available for big data.

https://www.r-bloggers.com/k-means-clustering-in-r/



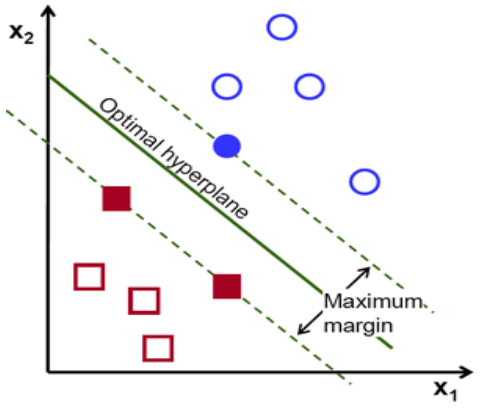

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Support vector machine (SVM)

- Classification algorithm for two classes. The algorithm is 'trained' on a sample of cases with both explanatory variables (attributes) and responses (categories). The algorithm finds a linear function of the attributes (a hyperplane in geometrical space) that best separates the two classes (i.e., maximizes the margin hyperplane, or distance, between them). A new case is then classified using the linear function.
- In addition to being relatively robust, efficient and accurate, SVMs have a theoretical base, need only a small number of cases for training, and can be scaled to big data. This scalability is achieved by breaking the problem into a series of smaller problems, each with a small number of selected variables, and iterating until all the decomposed optimization problems are solved successfully. Another very fast extension is core-vector machines, which aim to find "balls of cases", or core sets.

(figure from OpenCV)




QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Apriori

- Pattern finding algorithm that aims to find frequent sets of variables or items (itemsets) from a dataset and derive association rules.
- The algorithm first creates an itemset of size 1 comprising the most frequent items. Amongst this itemset, it then finds the most frequent pairs of items (itemsets of size 2), then amongst this new set it finds frequent itemsets of size 3, and so on.
- Although the standard Apriori algorithm is appropriate for big data, some extensions are even more efficient. Examples include hash-based techniques (put the itemsets into buckets based on a hash tag and remove buckets with small numbers of itemsets), partitioning (subset the data and analyse each subset separately) , sampling (analyse a sample of the data), using vertical data format (associate cases, instead of variables, with itemsets), and frequent pattern growth (FP-growth).
- FP-growth is one of the fastest extensions: it creates a tree-like pattern of frequent items (a FP-tree), uses this to create a set of databases each associated with one frequent itemset, and analyses each database separately. This drastically reduces the number of database scans.

# Expectation-Maximisation (EM)

- Clusters can be considered as a mixture of distributions. The EM (Expectation-Maximisation) algorithm is a fast, mathematically sound algorithm for estimating the means, variances and relative weights of the clusters, assuming that they are normally distributed (i.e., a finite normal mixture).
- The algorithm proceeds by iteratively calculating the probability of observations belonging to clusters, then estimating the mixture parameters (means, variances, weights), until a pre-specified level of stability is reached.
- The model can deal with known or unknown numbers of clusters.

# PageRank

- Aims to rank web pages based on their hyperlinks (i.e., links between the pages).
- This algorithm underpins the search engine Google, and variations of the algorithm are now used for every online search engine.
- A hyperlink from page x to page y is defined as a vote, by page x, for page y. Votes casted by pages that are themselves “important” weigh more heavily and help to make other pages more “important”. This is exactly the idea of rank prestige in social networks.
- The computation of PageRank values of the Web pages can be done using the power iteration method, which produces a principal eigenvector with an eigenvalue of 1. The iteration ends when the PageRank values do not change much (e.g, the sum of the absolute values of the residuals are less than a specified threshold).

# AdaBoost

- Classification algorithm based on an ensemble learning method (i.e., it uses multiple algorithms, or learners).
- It is fast, has a solid theoretical basis and good predictive ability, and is simple to program.
- The algorithm is trained on a sample of cases with a set of attributes X (the instance space) and class labels Y (representing two categories). AdaBoost applies a base classification algorithm (a learner), then increases the weights of misclassified cases and applies the algorithm again, and so on.
- This idea of “boosting” to improve the predictive capability is now very prevalent in many statistical and machine learning algorithms.
- Algorithms such as AdaBoost.M1 and AdaBoost.MH have been developed for problems with more than two classes, and algorithms for regression problems (continuous responses) are also available.

# K nearest neighbours (kNN)

- Supervised classification algorithm
- Starts with a training dataset in which each object (e.g., person, record, item) has a set of input variables and a class label (e.g., 1,2,…) that indicates the class, or group, to which the object belongs.
- A test dataset has input variables but no class labels, and we wish to assign the objects to the classes.
- For each object in the test dataset, the algorithm finds a group of k objects in the training set that are closest to that object (i.e. its k-nearest neighbours). It then assigns the test object a class label based on the labels of these neighbours.
- A common rule is to assign a label based on majority vote (i.e., the most common class amongst the neighbours) but other rules based on distance to the neighbours are also used.
- The algorithm thus relies on the user specification of the test dataset, the distance metric for choosing neighbours, and the value of k.
- KNN is sometimes described as a “lazy learner”, since there is no real underlying model as for decision trees, SVM, etc.
- KNN is argued to be particularly well suited for multi-modal classes as well as applications in which an object can have many class labels.

# Naïve Bayes

- Supervised classification algorithm: developed using a training set of input variables and known class labels, and then applied to a test set with input variables but no class labels.
- The algorithm constructs a score based on the input data and uses this to assign the objects in the test set to classes
- For example, with two classes, objects with scores less than a certain threshold are allocated to one class and those with scores above the threshold are allocated to the other class.
- The score is computed based on the ratio of the probabilities of belonging to the different classes based on the data and on the prior. If nothing is known beforehand about the allocations and the training set is a random sample, the prior probability of belonging to a class can be estimated by the proportion of objects of that class in the training set.
- The naïve Bayes algorithm is very robust and easy to construct and compute, so it can be easily applied to huge datasets, and it is easy to interpret. It is very popular in many fields, including text classification and spam filtering.
- Many extensions have been developed, but even the basic algorithm works well.
- The logistic regression model can also be seen as a type of naïve Bayes classifier.

# Discussion

Some package resources

https://github.com/qinwf/awesome-R

https://github.com/josephmisiti/awesome-machine-learning

# Discussion

## ML projects with R code

In order of difficulty to implement:

Analyse the text in Gone girl
http://danielphadley.com/Gone-Girl-Prediction/

Predict the winners of the 2015 Rugby world cup
https://rwcpredictor.herokuapp.com/how-it-works

# Discussion

## General surveys of Big Data analytics

- Big Data analytics: a survey.
  http://www.journalofbigdata.com/content/pdf/s40537-015-0030-3.pdf

- Top 10 algorithms
  http://www.cs.uvm.edu/~icdm/algorithms/10Algorithms-08.pdf

# Discussion

## Overviews of the use of Big Data in particular fields:

- Big Data Analysis Using Modern Statistical & ML Methods in Medicine. http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4076480/pdf/inj-18-50.pdf2.

- Statistical analysis & modelling of Internet VoIP traffic for network engineering. http://www.stat.purdue.edu/~xbw/research/VoIP.EJS2010.pdf

# Practical Session

- Summary of prac

# Plan

1. Let's talk about "Big Data"
2. Methods for modelling and analysis of big data
3. Digging deeper: (1) classification
4. Digging deeper: (2) regression
5. Digging deeper: (3) clustering
6. Digging deeper: (4) dimension reduction
7. Case study: recommender systems
8. From the learning to the doing: tips and tricks

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

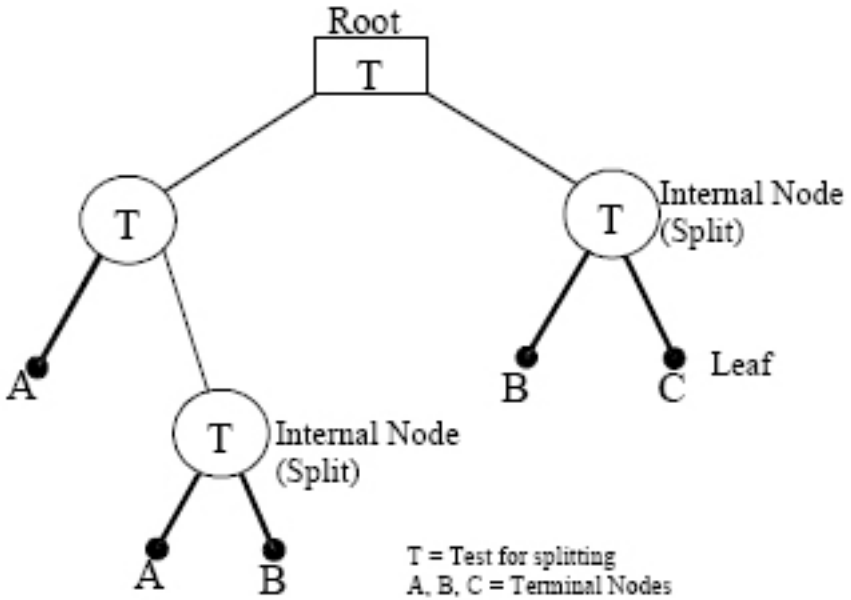
# Classification

## Common approaches:

- Decision trees: CART
- Bayesian networks
- K nearest neighbours (KNN)
- Support vector machines (SVM)
- Logistic and multinomial regression

# CART

https://www.bu.edu/sph/files/2014/05/MorganCART.pdf

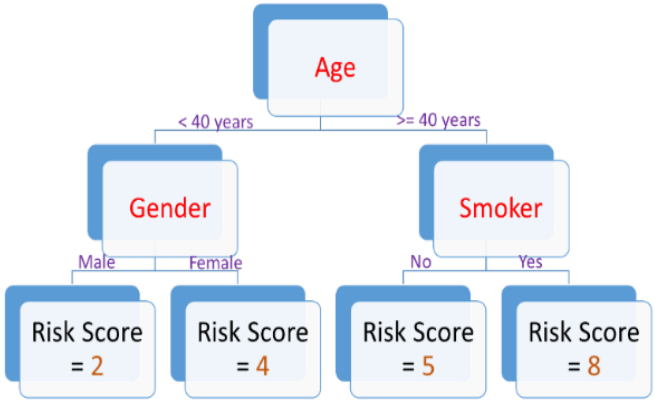



QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR
MATHEMATICAL AND STATISTICAL FRONTIERS

# CART, C4.5, See5, C5.0

- Classification and regression algorithms: predict the class to which a case belongs, or the expected value (and variance) of a group, based on a set of attributes.
- Two main steps: (i) grow a tree using a divide-and-conquer algorithm, which recursively splits the set of cases into two subgroups based on a splitting rule (find the attribute, and a threshold value of that attribute that minimizes diversity within the subgroups or maximizes information gain); (ii) prune the initial tree to avoid overfitting, based on the classification error rate.
- See5/C5.0 is a commercial system that is more scalable for big data (by multi-threading, enabling use of multiple cores), improves the predictive accuracy of C4.5 by including boosting (which constructs an ensemble of classifiers and votes on the final classification) and unordered rulesets, and allows for different data types.
- Other tree-based methods include boosted regression trees and random forests.




QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# CART

https://www.bu.edu/sph/files/2014/05/MorganCART.pdf

Recalling we want to find a function $d(x)$ to map our domain $X$ to our response variable $Y$ we need to assume the existence of a sample of $n$ observations $\mathcal{L} = \{(x_1, y_1), ..., (x_n, y_n)\}$. As in standard in regression equations, our criterion for choosing $d(x)$ will be the mean squared prediction error $E\{d(x) - E(y|x)\}^2$, or expected misclassification cost in the case of the classification tree. For each leaf-node $l$ and $c$ training samples in the regression tree, then, our model is just $\hat{y} = \frac{1}{c}\sum_{c=1}^{c} y_1$, "the sample mean of the response variable in that cell[12]" which creates a piecewise constant model.

Breiman JH, L. Olshen, RA Friedman, and Charles J. Stone. "Classification and Regression Trees." Wadsworth International Group (1984).

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# CART

https://www.bu.edu/sph/files/2014/05/MorganCART.pdf

In the case of the classification tree with leaf node $l$, training sample $c$ and $p(c|l)$, the probability that an observation $l$ belongs to class $c$, the **Gini index node impurity criterion**[3] $(1 - \sum_{c=1}^{C} p^2(c|l))$ defines the node splits, where each split maximizes the decrease in impurity. Whether using classification or regression, reducing error – either in classification or prediction – is the principal driving statistical mantra behind CART.

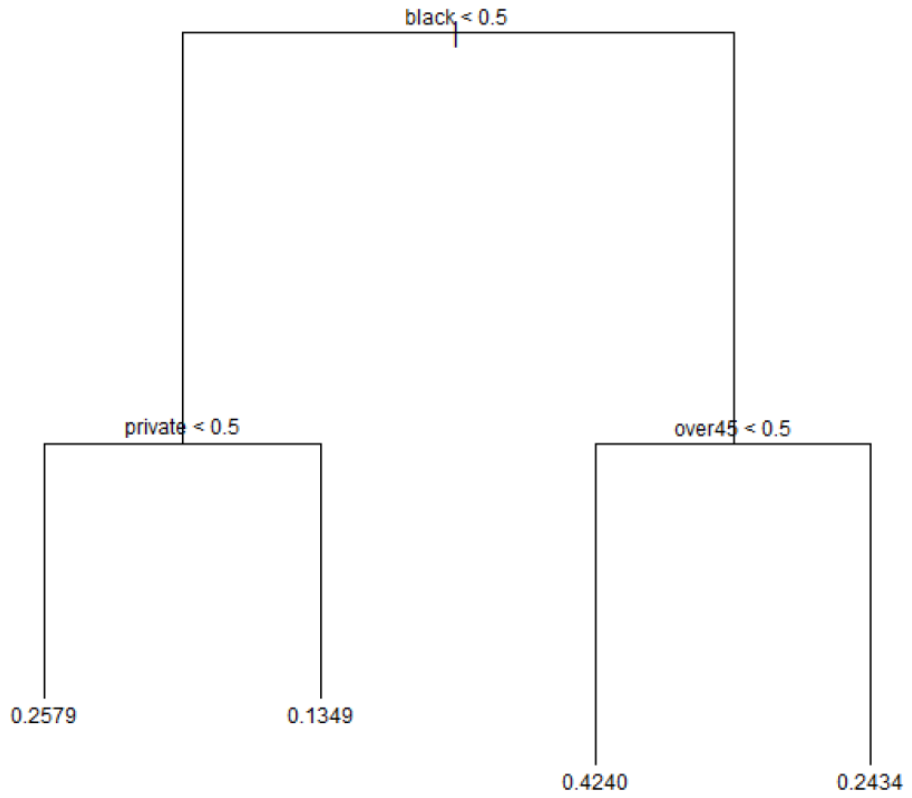
Breiman JH, L. Olshen, RA Friedman, and Charles J. Stone. "Classification and Regression Trees." Wadsworth International Group (1984).

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# CART

https://www.bu.edu/sph/files/2014/05/MorganCART.pdf




Whether or not a physician is a family practitioner at Boston University Hospital:

Main variables:
Race
Private/public
Age

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# “Let’s talk about kevin CART”

https://en.wikipedia.org/wiki/Random_forest

- Tree learning "come[s] closest to meeting the requirements for serving as an off-the-shelf procedure for data mining", say Hastie *et al.*, because it is invariant under scaling and various other transformations of feature values, is robust to inclusion of irrelevant features, and produces inspectable models. However, they are seldom accurate.
- In particular, trees that are grown very deep tend to learn highly irregular patterns: they overfit their training sets, i.e. have low bias, but very high variance.

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Extensions: Bagging

https://en.wikipedia.org/wiki/Bagging

Given a training set $X = x_1, ..., x_n$ with responses $Y = y_1, ..., y_n$, bagging repeatedly ($B$ times) selects a random sample with replacement of the training set and fits trees to these samples:

- For $b = 1, ..., B$:
- Sample, with replacement, $n$ training examples from $X$, $Y$; call these $X_b$, $Y_b$.
- Train a decision or regression tree $f_b$ on $X_b$, $Y_b$.

After training, predictions for unseen samples $x'$ can be made by averaging the predictions from all the individual regression trees on $x'$ or by taking the majority vote in the case of decision trees.

# Extensions: Bagging

https://en.wikipedia.org/wiki/Bagging

https://en.wikipedia.org/wiki/Random_forest

Given a training set $X = x_1, ..., x_n$ with responses $Y = y_1, ..., y_n$, bagging repeatedly ($B$ times) selects a random sample with replacement of the training set and fits trees to these samples:

- For $b = 1, ..., B$:
- Sample, with replacement, $n$ training examples from $X$, $Y$; call these $X_b$, $Y_b$.
- Train a decision or regression tree $f_b$ on $X_b$, $Y_b$.

After training, predictions for unseen samples $x'$ can be made by averaging the predictions from all the individual regression trees on $x'$ or by taking the majority vote in the case of decision trees.

Note: bagging can also be used for other models

# Extensions: Random forests

- https://en.wikipedia.org/wiki/Random_forest

- Random forests are a way of averaging multiple deep decision trees, trained on different parts of the same training set, with the goal of reducing the variance. This comes at the expense of a small increase in the bias and some loss of interpretability, but generally greatly boosts the performance of the final model.
- Random forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

# Random forests

- https://en.wikipedia.org/wiki/Random_forest

- Random forests differ in only one way from the original bagging algorithm for trees: they use a modified tree learning algorithm that selects, at each candidate split in the learning process, a random subset of the features.
- This process is sometimes called "feature bagging".
- The reason for doing this is the correlation of the trees in an ordinary bootstrap sample: if one or a few features are very strong predictors for the response variable (target output), these features will be selected in many of the *B* trees, causing them to become correlated.
- Typically, for a classification problem with *p* features, √*p* (rounded down) features are used in each split. For regression problems the inventors recommend *p/3* (rounded down) with a minimum node size of 5 as the default.

# Example: Costing consultations

- With Amy Cook *et al.*

- Predict number of hours for a consulting job
- Based on information extracted from 2000 consultancies
- Compare random forests, SVM, neural networks

- Used ‘cforest’ in Party R package

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Quote Assist

**Discipline:** Civil

**Primary Job Type:** Building Structures

**Bulk of Work by:** Director

**Secondary Job Type:** Please start typing

**Billing Type:** Hourly Rate

**Client Industry:** Please start typing

**Team Size:** 1 – 6 (1 2 3 4 5 6)

**Approximate Fee:** 20,000 – 250,000 (500 25,500 50,500 100,500 150,500 200,500 250,000)

Calculate

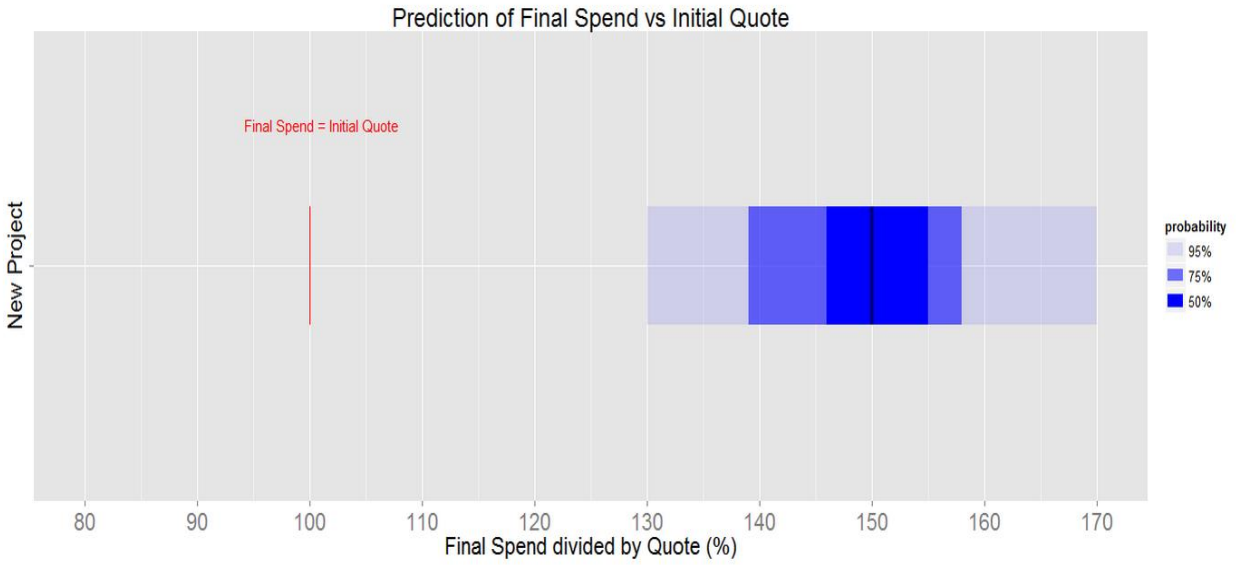
**Client:** Please start typing

**Client size:** individual

**Client sector:** Government

**% Hours by Professional:** 0 – 50 – 100 (0 10 20 30 40 50 60 70 80 90 100)

Calculate




QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Related applications

- Ravi Kumar and Ravi: found the most widely used machine learning models in business problems were neural networks, although decision trees, SVM's, and linear models were also popular.
- Saradhi and Palshikar's Employee Churn study predicted internal employee churn using an SVM model, naïve bayes, and neural networks.

P. Ravi Kumar and V. Ravi, "Bankruptcy prediction in banks and firms via statistical and intelligent techniques – a review," European Journal of Operational Research, vol. 180, pp. 1-28, July 2007.

V.V. Saradhi and G.K. Palshikar, "Employee churn prediction," Expert Systems with Applications, vol. 38, pp. 1999-2006, March 2011

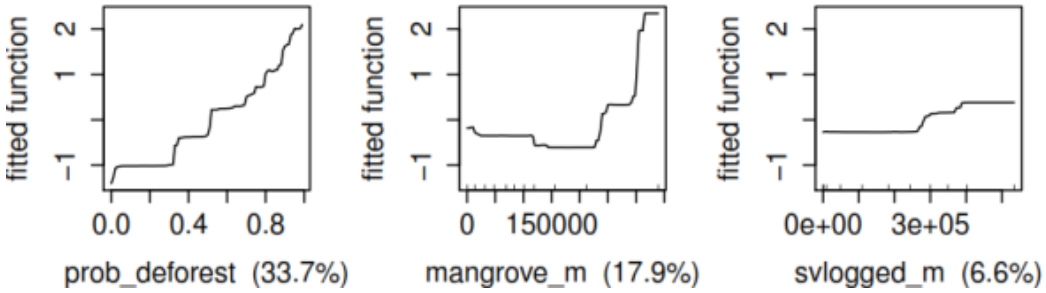
# Extensions: Boosting

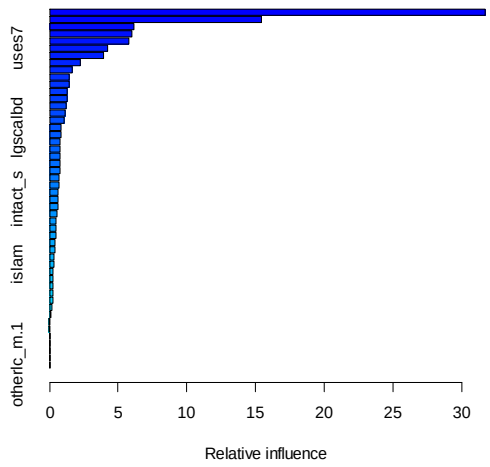
- https://en.wikipedia.org/wiki/Gradient_boosting
- https://en.wikipedia.org/wiki/AdaBoost

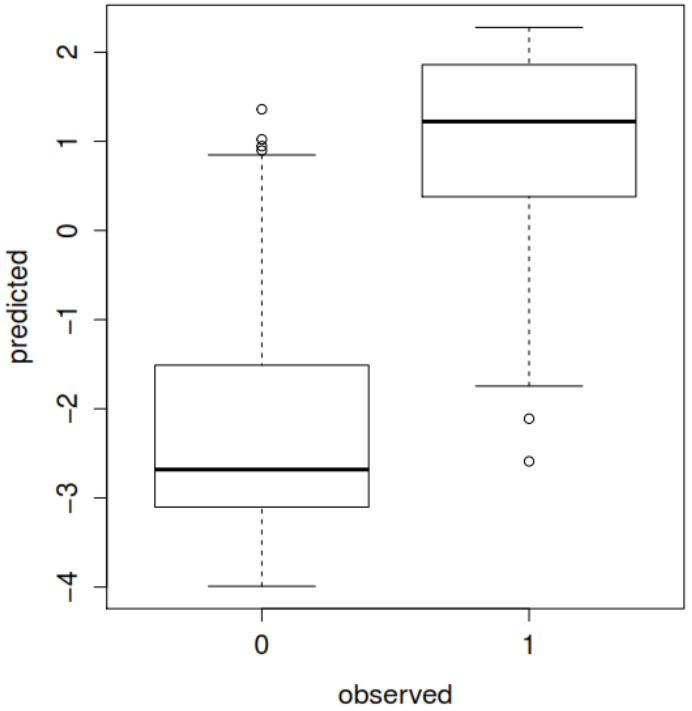
# Example: orangutans!

With Nicola Abram *et al.*

Understand conflict associated with oil palm

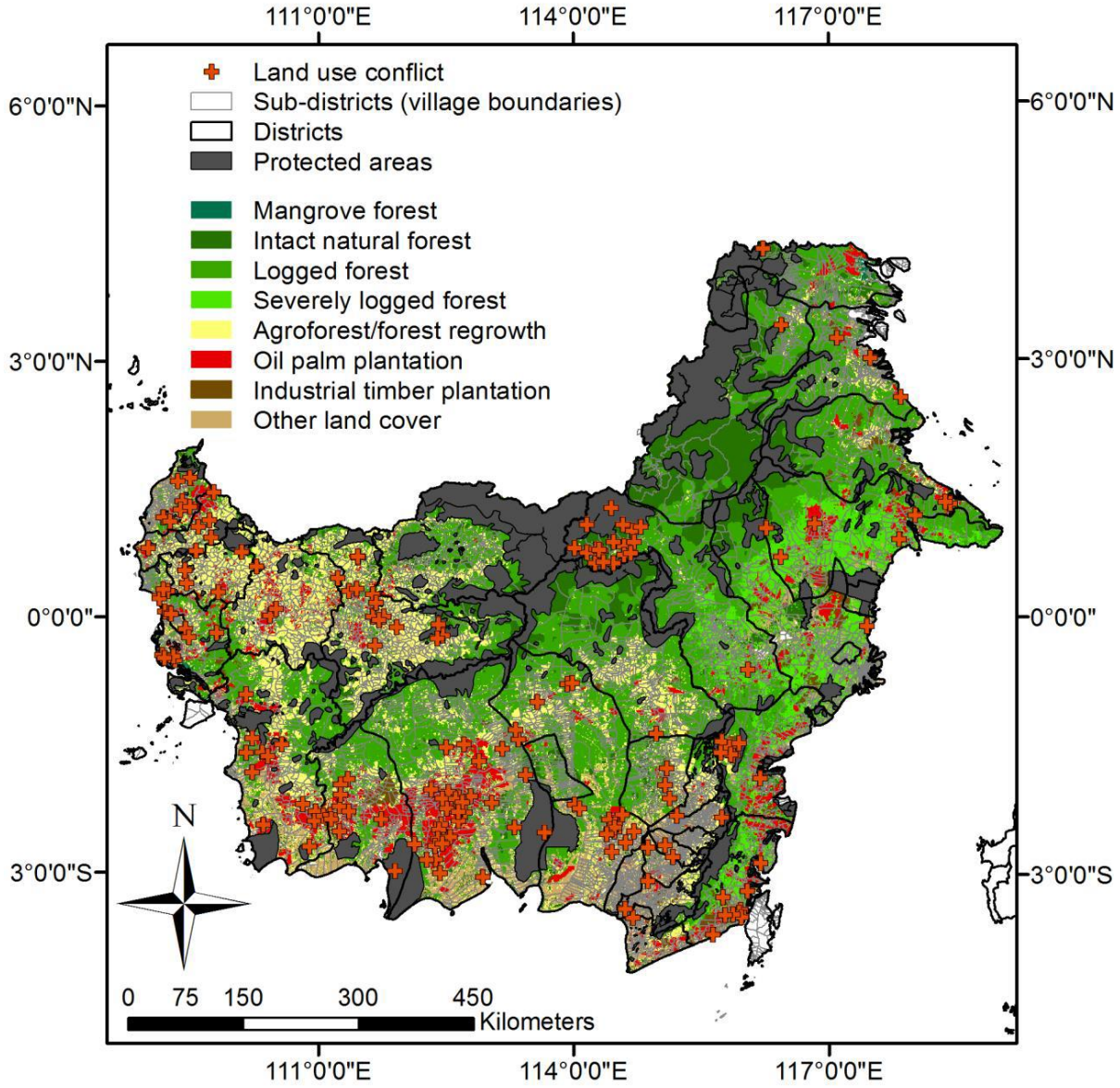


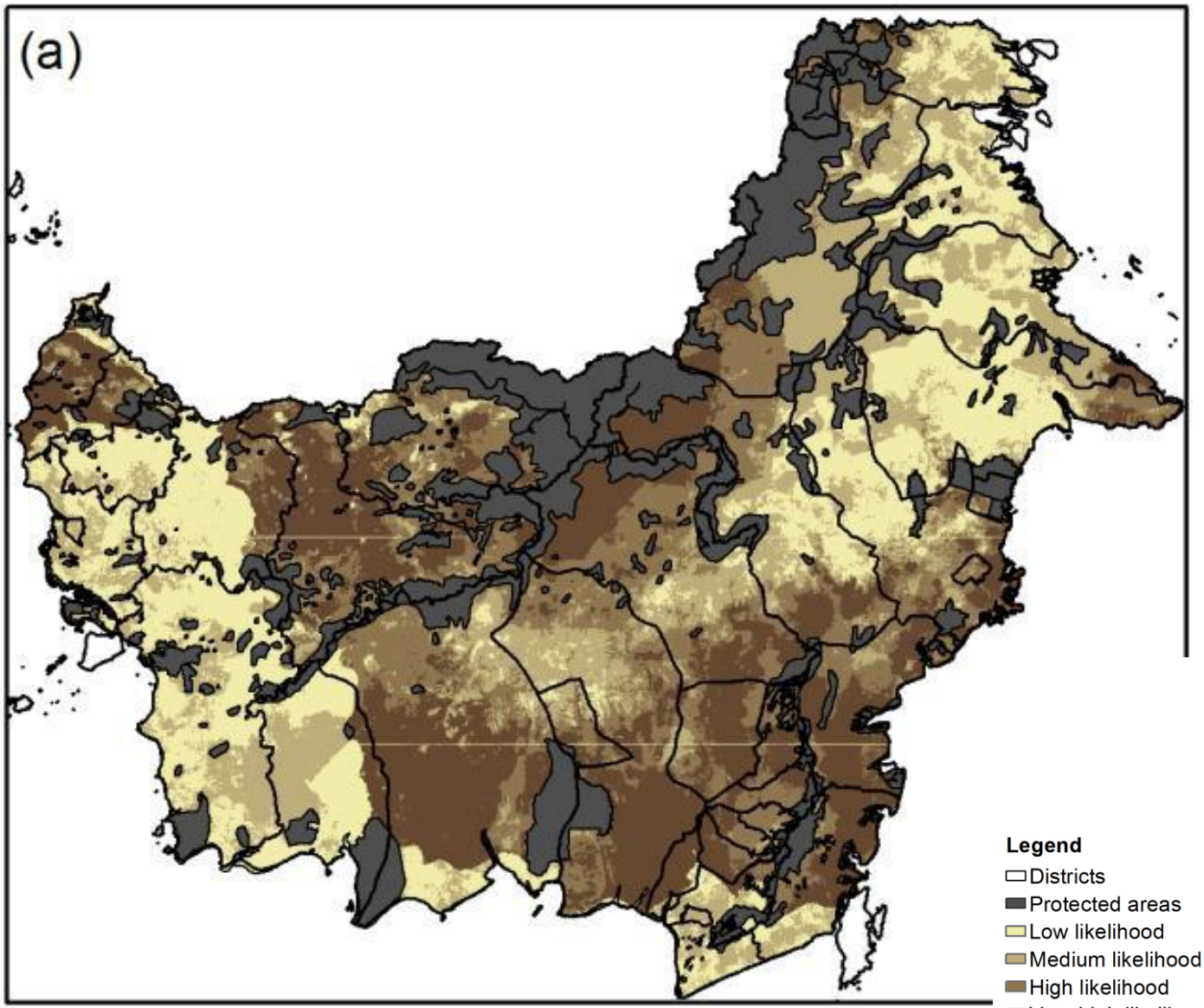

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS




QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR
MATHEMATICAL AND STATISTICAL FRONTIERS




QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR
MATHEMATICAL AND STATISTICAL FRONTIERS

# Example: the Titanic

- The sinking of the RMS Titanic in 1912 resulted in the loss of over 1500 lives at sea.
- Let's use a simple decision tree in R to predict the survival of an unknown passenger.
- The available variables are:
  - passenger class
  - survival
  - gender
  - age
  - number of siblings or spouses aboard
  - number of parents or children aboard
- Our passenger is:
  - female
  - 3rd class passenger
  - one spouse aboard, no siblings
  - 28 years old
  - no parents or children aboard

Wikipedia

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# R commands

```
# install packages
> install.packages("rpart")
> install.packages("rpart.plot")
> install.packages(“gbm”)
> library(rpart)
> library(rpart.plot)
> library(gbm)

# read in data
> data(ptitanic)  # dataset is already in rpart.plot
> ?ptitanic
> head(ptitanic)
```

# R commands

```
# run classification and regression tree (CART)
> titanic.rpart <- rpart(survived ~ pclass + sex + age + sibsp + parch, data = ptitanic)
> prp(titanic.rpart)

# run logistic regression
> titanic.glm <- glm(survived ~ pclass + sex + age + sibsp + parch, family=binomial, data = ptitanic)
> predict.glm(titanic.glm, newdata=data.frame(pclass=factor("3rd"), sex=factor("female"), age=28, sibsp=0, parch=0), type="response")

# run boosted regression tree (BRT)
> surv=as.double(survived)-1
> titanic.brt <- gbm(survived ~ pclass + sex + age + sibsp + parch, data = ptitanic)
> summary(titanic.brt)
> predict.gbm(gbm.tree,newdata=data.frame(pclass=factor("3rd"), sex=factor("female"), age=28, sibsp=0, parch=0), n.trees=100, type="response")
```

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

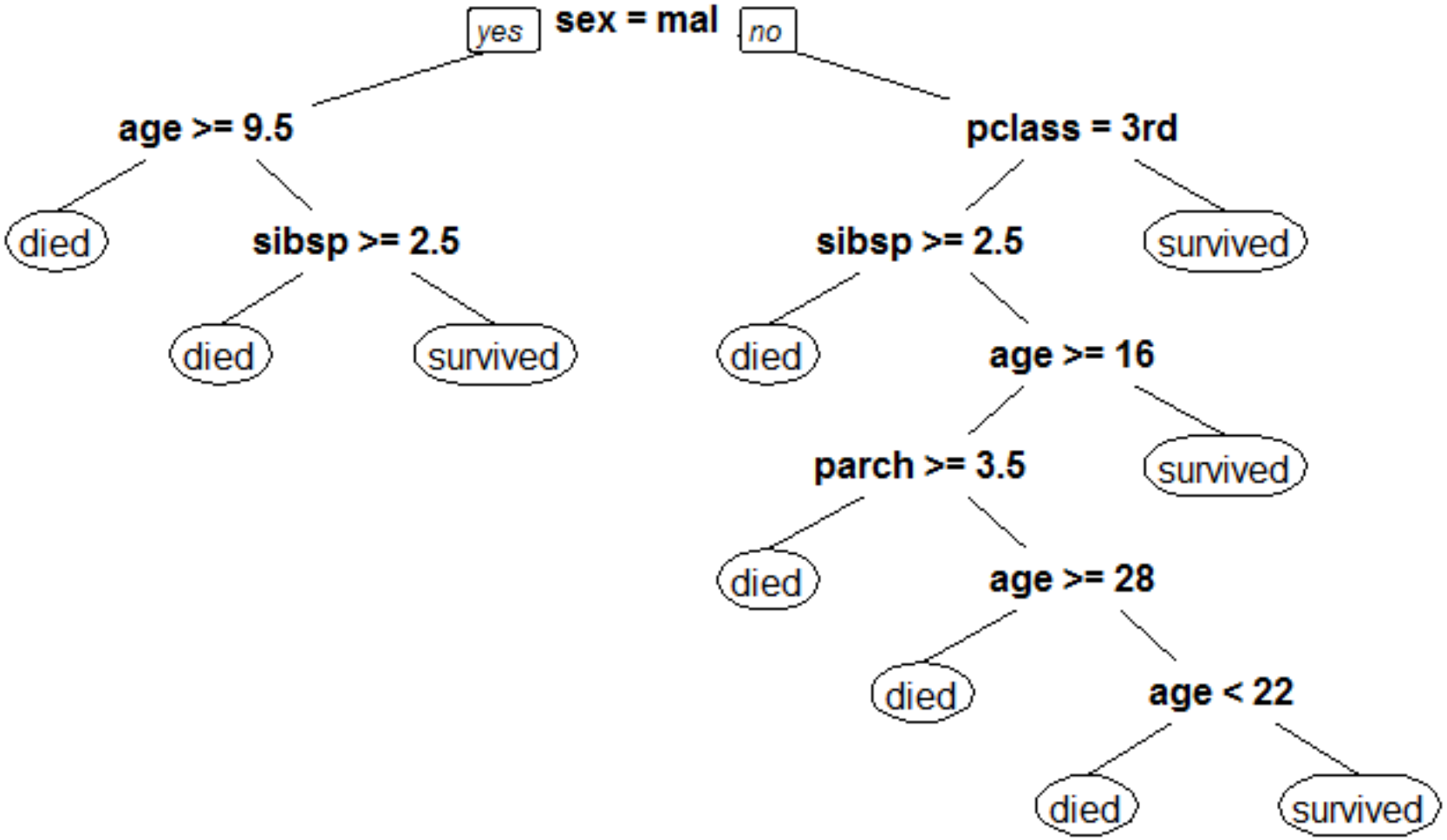
> head(ptitanic)

| | pclass | survived | sex | age | sibsp | parch |
|---|---|---|---|---|---|---|
| 1 | 1st | survived | female | 29.0000 | 0 | 0 |
| 2 | 1st | survived | male | 0.9167 | 1 | 2 |
| 3 | 1st | died | female | 2.0000 | 1 | 2 |
| 4 | 1st | died | male | 30.0000 | 1 | 2 |
| 5 | 1st | died | female | 25.0000 | 1 | 2 |
| 6 | 1st | survived | male | 48.0000 | 0 | 0 |

Our unknown passenger:

| | | | | | | |
|---|---|---|---|---|---|---|
| | 3rd | ? | female | 28.0000 | 0 | 0 |

# CART model


yes
sex = mal
no
age >= 9.5
pclass = 3rd
died
sibsp >= 2.5
sibsp >= 2.5
survived
died
survived
died
age >= 16
parch >= 3.5
survived
died
age >= 28
died
age < 22
died
survived

## Logistic regression model

Call: glm(formula = survived ~ pclass + sex + age + sibsp + parch,
family = binomial, data = ptitanic)

Coefficients:

| (Intercept) | pclass2nd | pclass3rd | sexmale | age | sibsp | parch |
|---|---|---|---|---|---|---|
| 3.90679 | -1.36676 | -2.35202 | -2.55686 | -0.03949 | -0.35291 | 0.07436 |

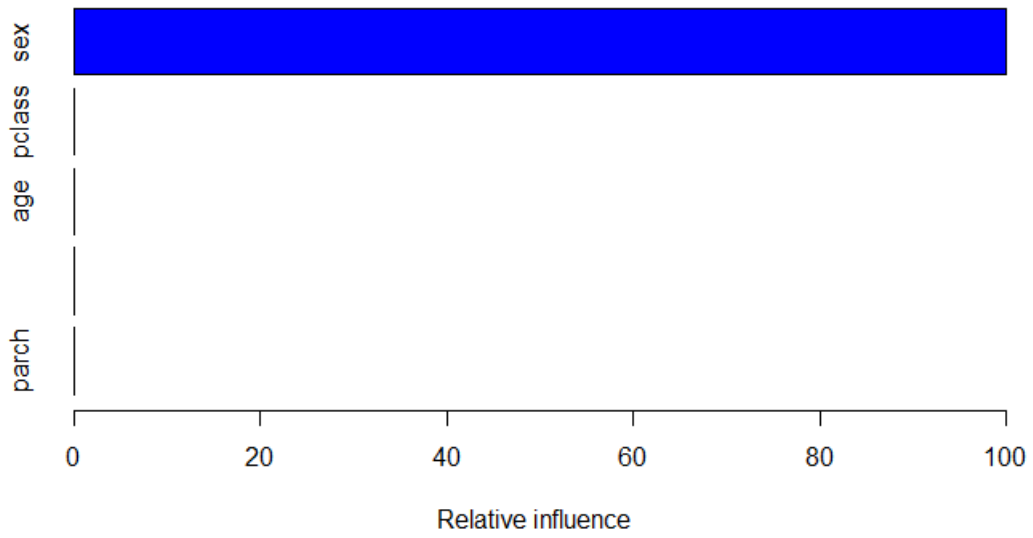
Degrees of Freedom: 1045 Total (i.e. Null); 1039 Residual
(263 observations deleted due to missingness)
Null Deviance: 1415
Residual Deviance: 970.1 AIC: 984.1

> predict….

1
0.6104219

# BRT model

| var | rel.inf |
|---|---|
| sex | 100 |
| pclass | 0 |
| age | 0 |
| sibsp | 0 |
| parch | 0 |




> predict.gbm(gbm.tree,newdata=data.frame(pclass=factor("3rd"), sex=factor("female"), age=28, sibsp=0, parch=0), n.trees=100, type="response")

0.4149037

# Bayesian networks

Tutorial by David Heckerman

https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr-95-06.pdf

A Bayesian network for a set of variables $\mathbf{X} = \{X_1, \ldots, X_n\}$ consists of (1) a network structure $S$ that encodes a set of conditional independence assertions about variables in $\mathbf{X}$, and (2) a set $P$ of local probability distributions associated with each variable. Together, these components define the joint probability distribution for $\mathbf{X}$. The network structure $S$ is a directed acyclic graph. The nodes in $S$ are in one-to-one correspondence with the variables $\mathbf{X}$. We use $X_i$ to denote both the variable and its corresponding node, and $\mathbf{Pa}_i$ to denote the parents of node $X_i$ in $S$ as well as the variables corresponding to those parents.

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Bayesian networks

Tutorial by David Heckerman

https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr-95-06.pdf

The

*lack* of possible arcs in $S$ encode conditional independencies. In particular, given structure $S$, the joint probability distribution for $\mathbf{X}$ is given by

$$p(\mathbf{x}) = \prod_{i=1}^{n} p(x_i|\mathbf{pa}_i) \tag{16}$$

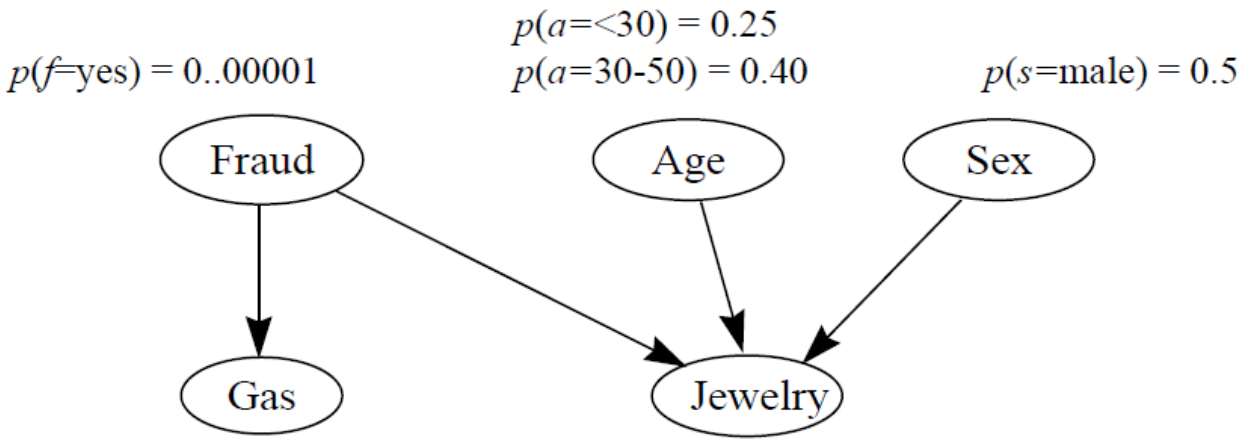
The local probability distributions $P$ are the distributions corresponding to the terms in the product of Equation 16. Consequently, the pair $(S, P)$ encodes the joint distribution $p(\mathbf{x})$.

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Bayesian networks

https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr-95-06.pdf




$p(g=yes|f=yes) = 0.2$
$p(g=yes|f=no) = 0.01$

$p(j=yes|f=yes,a=*,s=*) = 0.05$
$p(j=yes|f=no,a=<30,s=male) = 0..0001$
$p(j=yes|f=no,a=30\text{-}50,s=male) = 0.0004$
$p(j=yes|f=no,a=>50,s=male) = 0.0002$
$p(j=yes|f=no,a=<30,s=female) = 0..0005$
$p(j=yes|f=no,a=30\text{-}50,s=female) = 0.002$
$p(j=yes|f=no,a=>50,s=female) = 0.001$

Detecting credit card fraud

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Bayesian networks

Green and Mortera, BNs for criminology

http://projecteuclid.org/download/pdfview_1/euclid.aoas/1245676193

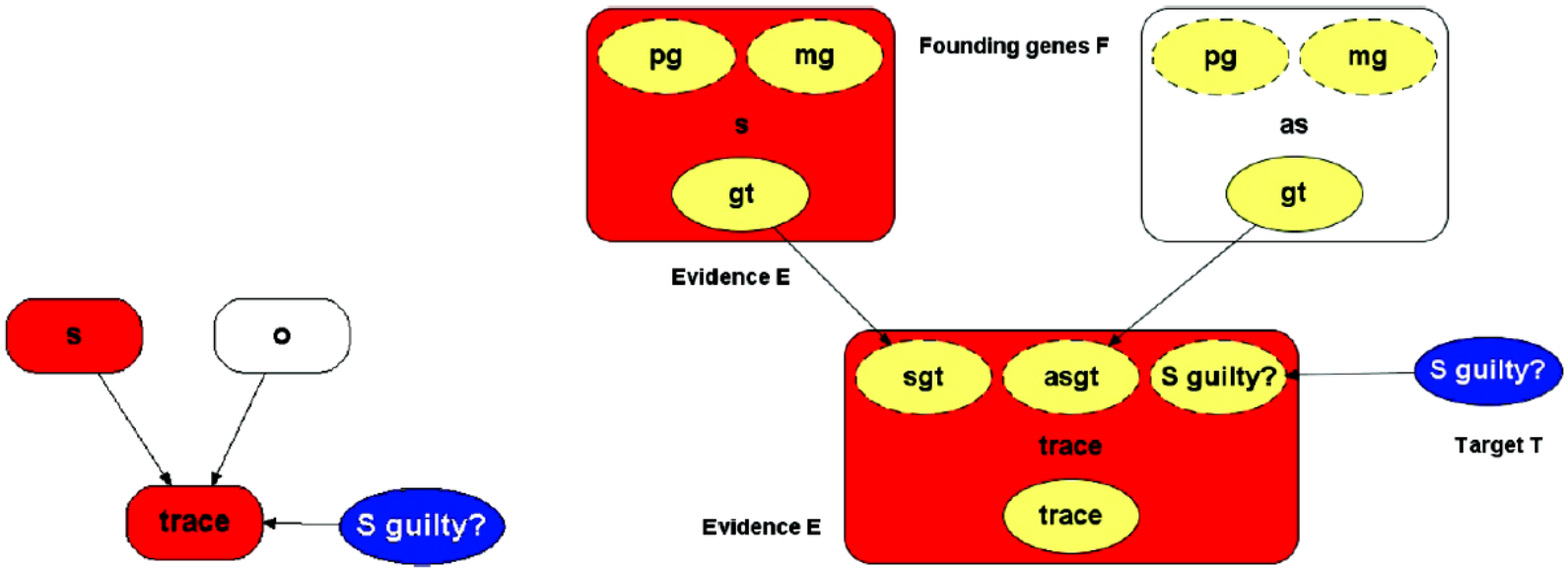



FIG. 1. *(Left) Block-level network for criminal identification, showing actors* s *and* as*, crime trace and target. (Right) Expanded network revealing variables within each block node.*

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# “How stressed is your airport?”


With
17 airports &
airlines

“Airports of
the future”

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR
MATHEMATICAL AND STATISTICAL FRONTIERS


QUT

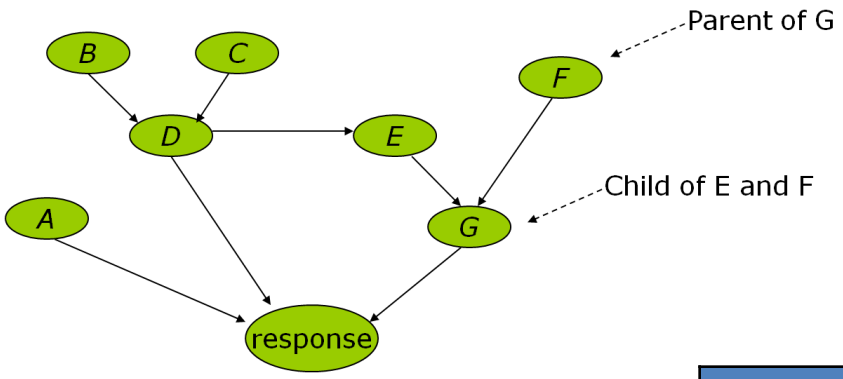
ACEMS

AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR
MATHEMATICAL AND STATISTICAL FRONTIERS

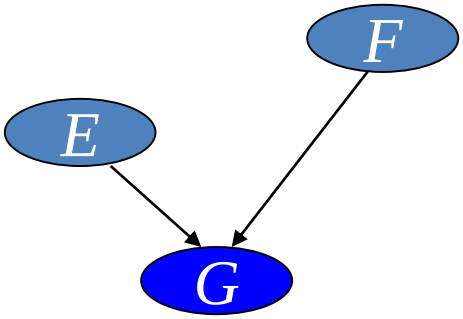

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR
MATHEMATICAL AND STATISTICAL FRONTIERS

# Bayesian Networks




| F (traffic) | |
|---|---|
| *low* | 0.7 |
| *medium* | 0.2 |
| *high* | 0.1 |




| | | G (baggage time) | |
|---|---|---|---|
| *E (weekend)* | *F (traffic)* | *normal* | *high* |
| *yes* | *low* | 0.4 | 0.6 |
| | *medium* | 0.2 | 0.8 |
| | *high* | 0.1 | 0.9 |
| *no* | *low* | 0.5 | 0.5 |
| | *medium* | 0.6 | 0.4 |
| | *high* | 0.4 | 0.6 |

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

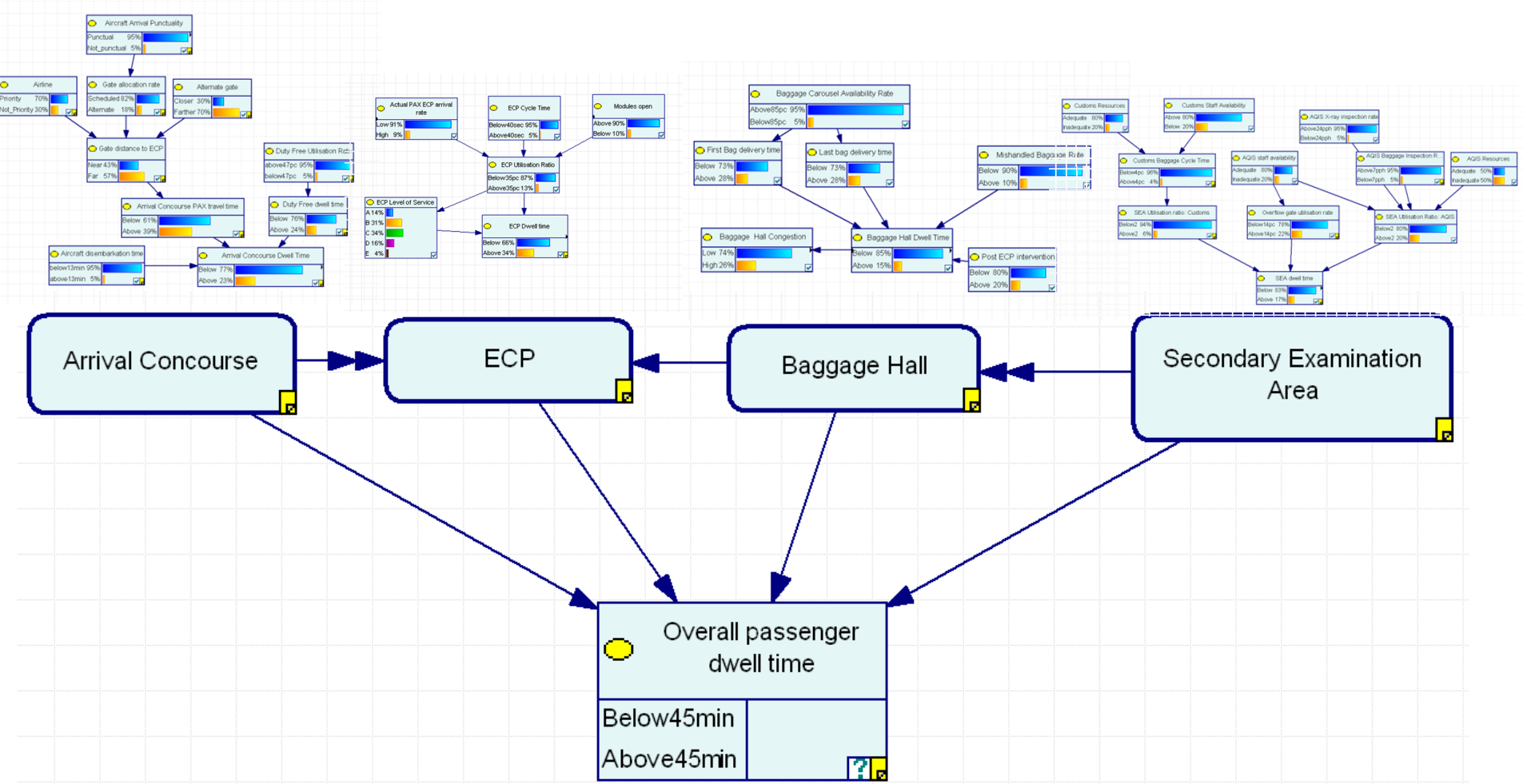
# Inbound passenger process: available data

- Recorded data (e.g. Flights: origin, time of day, carrier)
- CCTV intelligent surveillance
- Expert knowledge


| 2.1.1 Arrival Concourse Dwell Time - Recruitment to Joining entry control (3) | 2.1.2 Entry Control Point Dwell Time - Join entry control (3) to Exit control point (4) | 2.1.3 Baggage Reclaim Hall Dwell Time - Exit control point (4) to Join secondary queue (6) |
|---|---|---|
| 5 | 2 | 11 |
| 4 | 5 | 19 |
| 6 | | |
| 8 | 1 | 17 |
| 11 | 4 | 12 |
| 8 | 4 | 15 |
| 6 | 5 | |
| 4 | 1 | 10 |
| 6 | | |

| Date | Chocks-on Time | Gate | Total PAX |
|---|---|---|---|
| 30/9/2012 | 06:10:00 | 77 | 309 |
| 30/9/2012 | 06:33:00 | 80 | 4 |
| 30/9/2012 | 06:33:00 | 86 | 349 |
| 30/9/2012 | 06:37:00 | 79 | 301 |

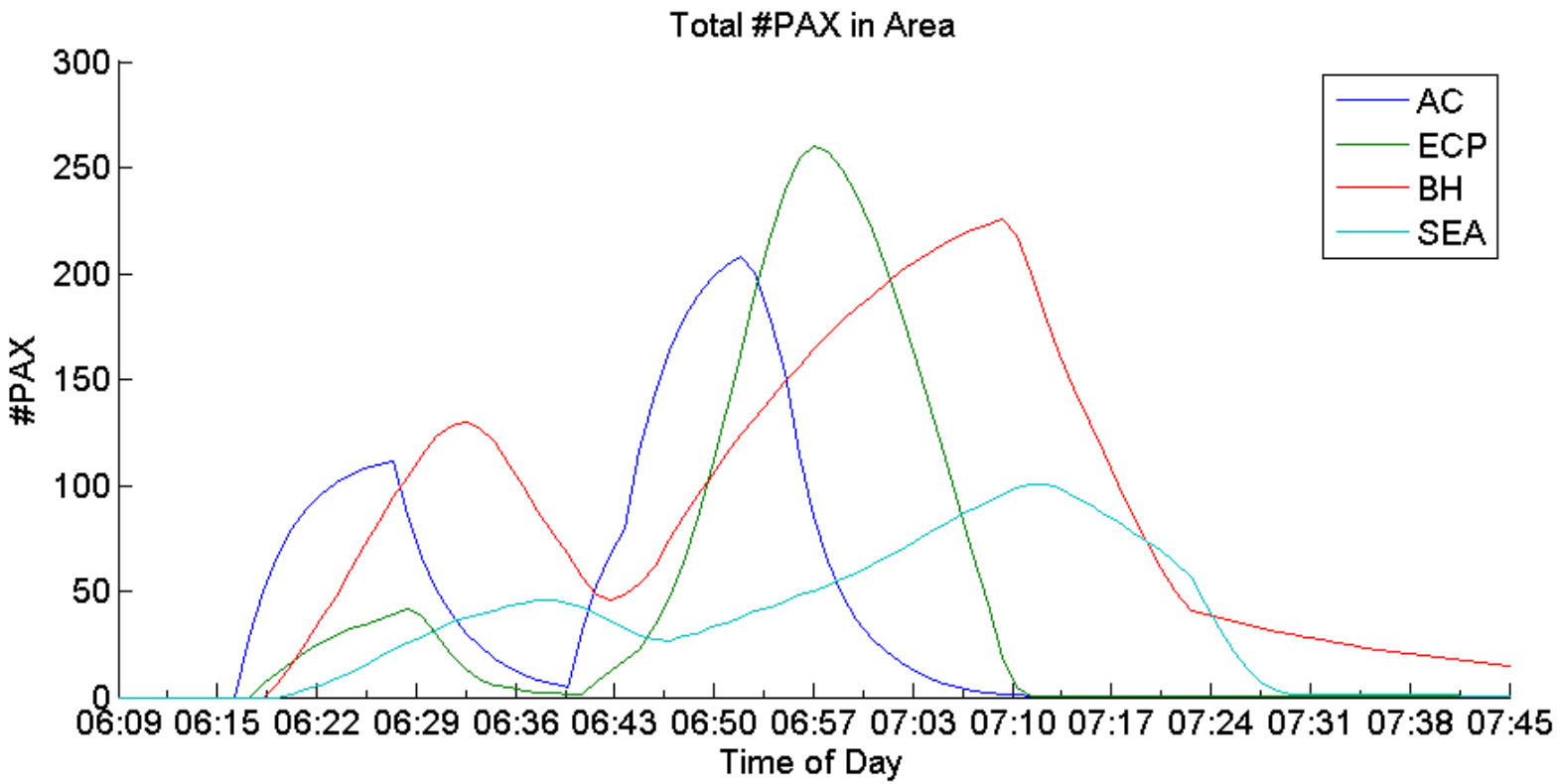
# Airports Systems Bayesian Network: Inbound Passengers




QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

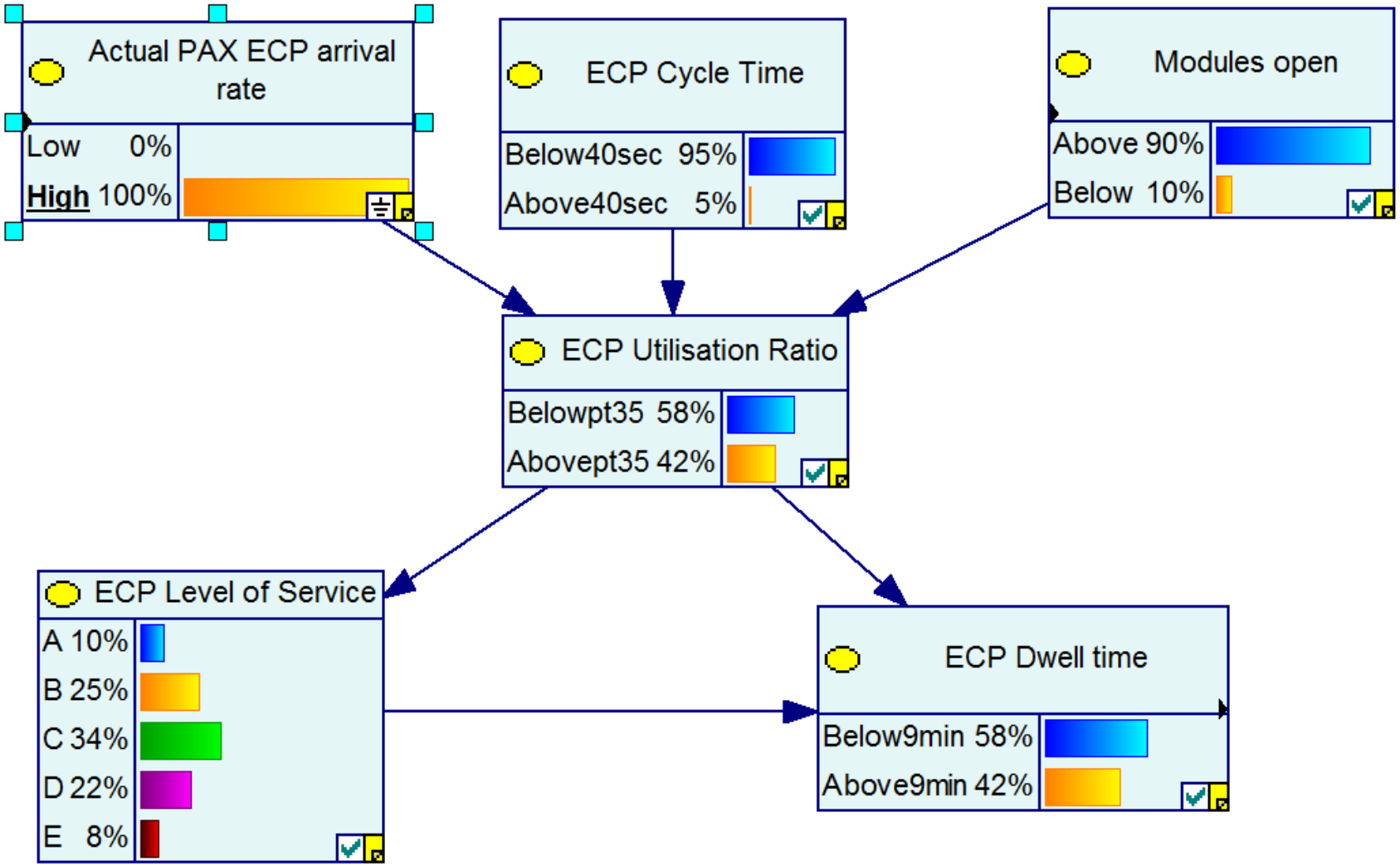
# Prediction (dwell time in each area)



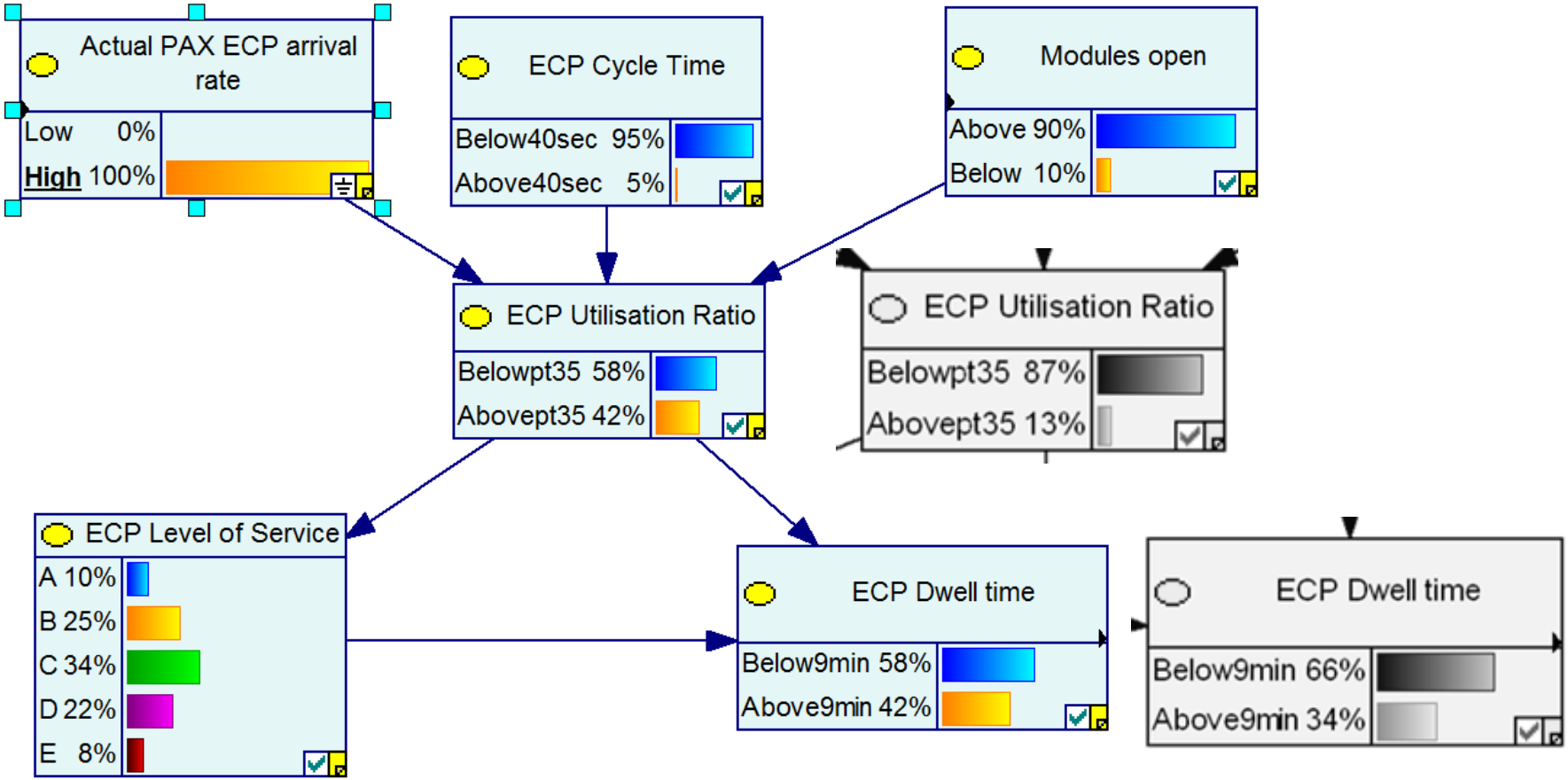

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

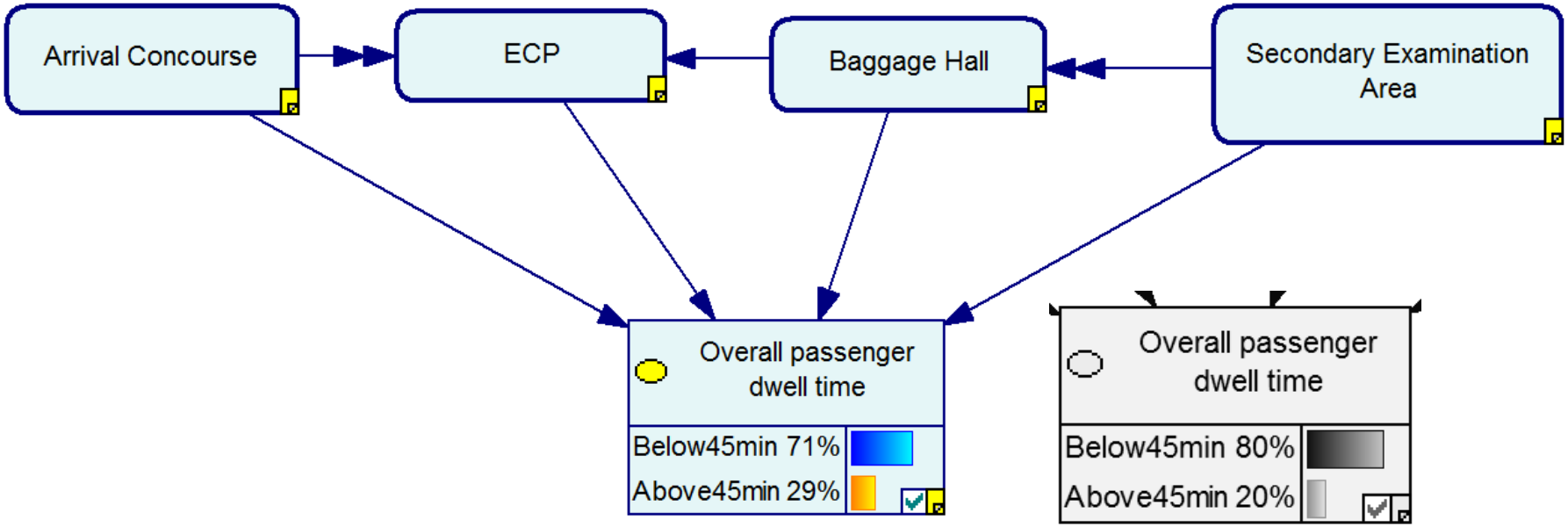
# “What if” scenario testing



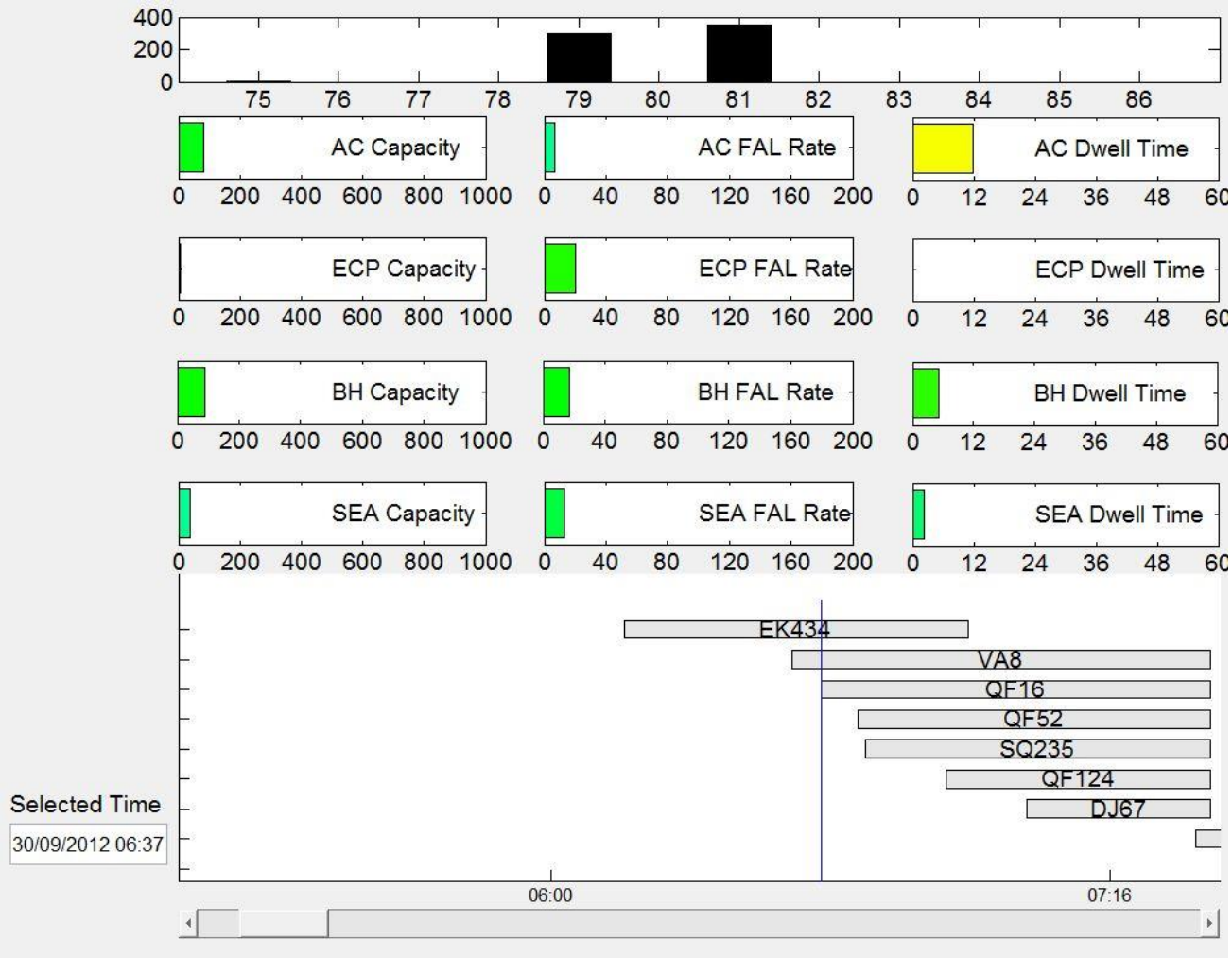
# “What if” scenario testing



# “What if” scenario testing



# Dashboard – *how stressed is your airport?*



# Other applications of BNS

- Ecological windows for low impact dredging
  - with Paul Wu *et al.*
- Reduction of peak demand for energy
  - with James Lewis, Laurie Buys *et al.*
- Sustainability of the dairy industry
  - with Laurie Buys *et al.*
- Survival of cheetahs in Southern Africa
  - with Sandra Johnson *et al.*
- Spread of infection in a hospital
  - with Tony Morton, TPCH Hospital, *et al.*

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Logistic and multinomial regression

Binary response variable *y,* independent variables X

$$y \sim \text{Binomial}(n,p)$$

$$\text{logit}(p) = X\beta + e$$

# Discussion

## Boosted regression trees

- Tutorial by Elith and Leathwick (2017)

https://cran.r-project.org/web/packages/dismo/vignettes/brt.pdf

## Bayesian networks

- Probabilistic networks and R code (Grappa) by Peter Green

https://people.maths.bris.ac.uk/~mapjg/Grappa/

https://people.maths.bris.ac.uk/~mapjg/Grappa/Handout.pdf

## Logistic regression

- https://en.wikipedia.org/wiki/Logistic_regression

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Discussion

## Other references

http://www.r2d3.us/visual-intro-to-machine-learning-part-1/

This youtube playlist, which is a companion to *An Introduction to Statistical Learning with Applications in R* by Trevor Hastie and Rob Tibshirani.These guys have another highly regarded book: The Elements of Statistical Learning

# Classification: Prac

- Description of prac

# Plan

1. Let's talk about "Big Data"
2. Methods for modelling and analysis of big data
3. Digging deeper: (1) classification
4. Digging deeper: (2) regression
5. Digging deeper: (3) clustering
6. Digging deeper: (4) dimension reduction
7. Case study: recommender systems
8. From the learning to the doing: tips and tricks

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Regression

## Common approaches:

- Linear models
- Decision trees

  CART, Boosted regression trees, random forests
- Gradient boosted models
- Neural networks

# Regression

Given a dataset of $n$ observations, a linear regression model assumes that the relationship between the dependent variable $y_i$ and the $p$-vector of regressors $\boldsymbol{x}_i$ is linear.

$$\mathbf{y} = \boldsymbol{X}^{\mathrm{T}}\boldsymbol{\beta} + \varepsilon$$

Extensions for big data analysis:

- Principal component analysis
- Least angle regression
- Multivariate adaptive regression splines (MARS)
- Support vector machines
- "Online" regression

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# (Online) real-time regression

## Linear models

- Segment data and/or segment analysis
- Develop iterative equations
- Bayesian approach
- Keep a small sample of the data in memory and change the calculation function (Peng)
- http://www.slideshare.net/HesenPeng/linear-regression-on-1-terabytes-of-data-some-crazy-observations-and-actions-jsm

# Segmenting data

## Divide and recombine

Analyse subsets of the data in parallel by different processors, via either multi-core CPU or massively parallel GPU, and combine results

- *Pros*: very effective
- *Cons*: (i) can't alleviate bottlenecks related to memory or disk; (ii) requires sophisticated programming; (iii) different communication costs, so need different algorithms

https://en.wikipedia.org/wiki/Divide_and_conquer_algorithm

# Segmenting analyses

## Consensus Monte Carlo

Run a separate Monte Carlo algorithm on each machine, then average individual Monte Carlo draws across machines

- *Pros:* embarrassingly parallel, so can be run on virtually any system for parallel computing, multi-core systems, or networks of workstations
- *Cons:* need rules for combining posterior samples; still requires careful programming

Scott et al. (2013)

http://www.rob-mcculloch.org/some_papers_and_talks/papers/working/consensus-mc.pdf

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# DeltaRho

R platform for big data analysis

- Builds on Hadoop and MapReduce
- Uses divide and recombine algorithm
- R package: datadr

```
# Install base DeltaRho packages from CRAN
install.packages("datadr")

install.packages("trelliscope")
```

http://deltarho.org/

“Try it out”

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Updating regression equations: Gradient descent

https://en.wikipedia.org/wiki/Gradient_descent

- first-order iterative optimization algorithm
- To find a local minimum of a function using gradient descent, one takes steps proportional to the negative of the gradient (or of the approximate gradient) of the function at the current point.
- Batch and stochastic versions

Ruder: overview of gradient descent methods
https://arxiv.org/abs/1609.04747

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Gradient descent

Pitt: course notes on gradient descent methods

https://people.cs.pitt.edu/~milos/courses/cs2750-Spring03/lectures/class6.pdf

Stochastic gradient descent for streaming data

http://www.subsubroutine.com/sub-subroutine/2014/10/19/real-time-learning-in-data-streams-using-stochastic-gradient-descent

# Semiparametric regression

- http://realtime-semiparametric-regression.net/Examples/index.html?ex=SydneyRealEstate

  The logarithm of weekly rent is assumed to follow a normal distribution with variance $\sigma_\epsilon^2$ and mean

  $$E[\log(\{\text{weekly rent}\}_{ij})] = \beta_0 + U_i + \beta_1 \text{house}_{ij} + f_2(\{\text{number of bedrooms}\}_{ij}) + f_3(\{\text{number of bathrooms}\}_{ij}) + f_4(\{\text{number of car spaces}\}_{ij}) + f_5(\text{longitude}_{ij}, \text{latitude}_{ij}),$$

  where

  - $\{\text{weekly rent}\}_{ij}$ is the weekly rental amount in Australian dollars of the jth property for the ith real estate agency
  - $U_1, \ldots, U_{992} | \sigma_U^2 \sim .N(0, \sigma_{U2})$ are random intercepts for the rental agency
  - $\text{house}_{ij}$ is an indicator for the property being a house or apartment
  - $\text{longitude}_{ij}$ and $\text{latitude}_{ij}$ convey the geographic location of the property
  - $f_2, f_3, f_4, f_5$ are unknown functions estimated using penalized spline methodology
- http://realtime-semiparametric-regression.net/assets/pdf/LutsBroderickWandPaper.pdf
- http://realtime-semiparametric-regression.net/Examples/index.html?ex=StockData

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Time series regression

Gruber and West (2016): forecasting and scalable multivariate volatility analysis

https://arxiv.org/pdf/1606.08291v1.pdf

standard univariate DLM combines a normal linear observation equation,

$$y_t = \mathbf{F}_t \theta_t + \nu_t, \tag{1}$$

with a conditionally normal, multivariate linear system equation to govern the state evolutions of $\theta_t$ from time $t$ to $t+1$,

$$\theta_{t+1} = \mathbf{G}_{t+1} \theta_t + \omega_{t+1}. \tag{2}$$

Recently introduced simultaneous graphical dynamic linear models (SGDLMs: Gruber and West 2016) address scalability. These involve: (i) a sets of *decoupled* univariate dynamic linear models (DLMs) for individual series, allowing a range of time-varying parameter models and univariate volatilities, and for which standard theory and resulting efficient forward filtering/forecasting algorithms apply; (ii) exploitation of a simultaneous equations formulation with sparse graphical modeling ideas that *recouple* the series and define rich yet sparse representations of multivariate stochastic volatility; and (iii) variational Bayes methods combined with importance sampling to coherently integrate/couple the series for forecasting and decisions. Parallel, GPU-based implementation enables on-line analysis of increasingly high-dimensional time series.

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

Gruber and West (2016): daily log returns, 400 stocks

https://arxiv.org/pdf/1606.08291v1.pdf

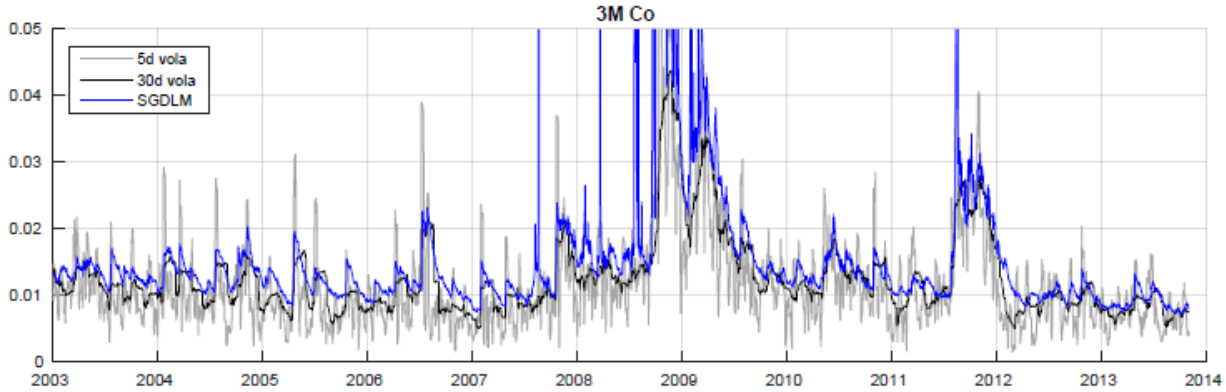



Figure 1: Volatility of stock returns of company 3M ($j = 245$): observed 5-day and 30-day tracking volatilities (gray and black, respectively) together with the predicted volatility (blue) under model M1 of Table 1.

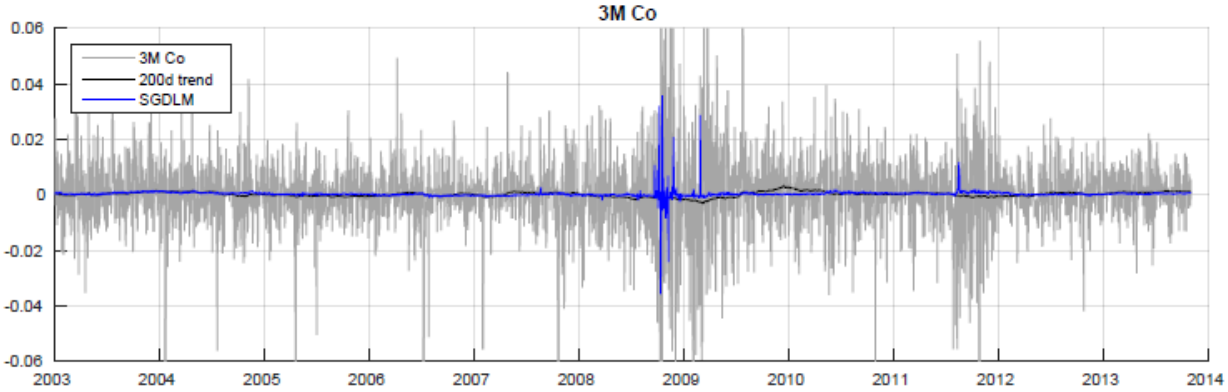



Figure 2: Daily log-returns (gray) of company 3M, together with the observed 200-day trend line (black) and the predicted trend (blue) from model M1 of Table 1.

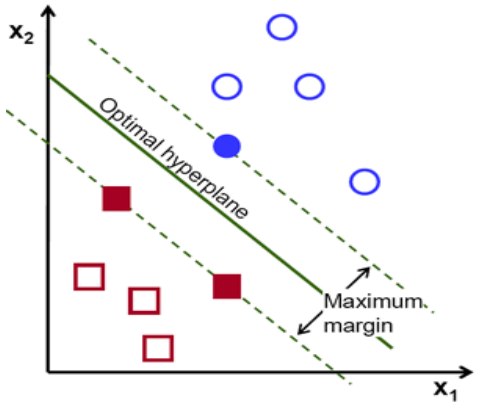
QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Support vector machine (SVM)

- Classification algorithm for two classes. The algorithm is 'trained' on a sample of cases with both explanatory variables (attributes) and responses (categories). The algorithm finds a linear function of the attributes (a hyperplane in geometrical space) that best separates the two classes (i.e., maximizes the margin hyperplane, or distance, between them). A new case is then classified using the linear function.
- In addition to being relatively robust, efficient and accurate, SVMs have a theoretical base, need only a small number of cases for training, and can be scaled to big data. This scalability is achieved by breaking the problem into a series of smaller problems, each with a small number of selected variables, and iterating until all the decomposed optimization problems are solved successfully. Another very fast extension is core-vector machines, which aim to find "balls of cases", or core sets.

(figure from OpenCV)




QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Bayesian regression

$$p(\theta|Y) = p(Y|\theta)\, p(\theta) / p(Y)$$

$p(Y|\theta)$

*What do we know about y given values of $\theta$ ?*

$p(\theta)$ is the *prior* for $\theta$.

*What do we know about $\theta$ independently of the data?*

$p(Y) = \Sigma\, p(\theta)\, p(y|\theta)$ or $\int p(\theta)\, p(y|\theta)\, d\theta$

i.e., the probability of the data for all values of $\theta$.

*(constant – calculate analytically or numerically)*

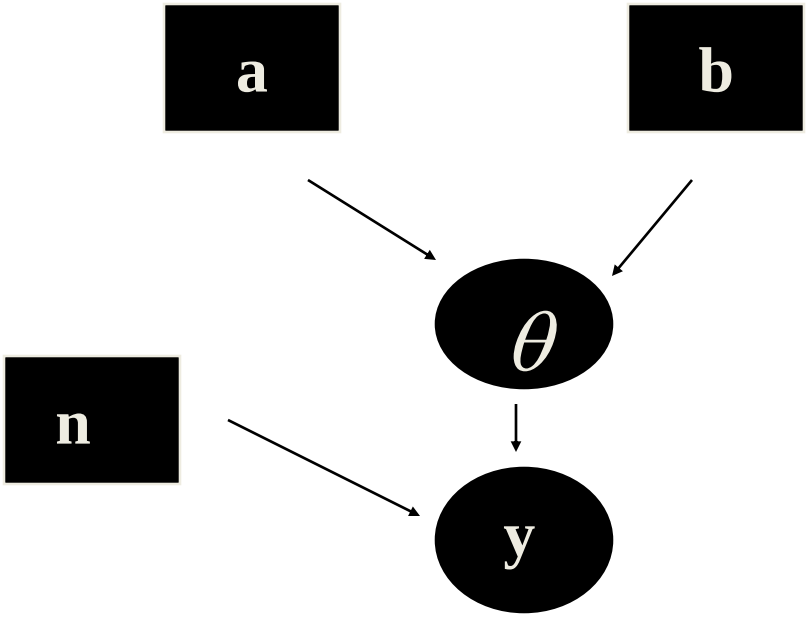
QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Example: Binomial model

*y ~ Binomial ($\theta$, n)*

*$\theta$ ~ Beta (a,b)*




QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Posterior

$$p(\theta \mid y) \propto \textit{likelihood} \times \textit{prior}$$

$$= \theta^{y} (1-\theta)^{n-y} \times \theta^{a-1} (1-\theta)^{b-1}$$

$$= \theta^{y+a-1} (1-\theta)^{n-y+b-1}$$

**This is another beta distribution**

$$\theta \mid y \sim \textit{Beta}(y+a, n-y+b)$$

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Dynamic Updating

- If we obtain more data, we do not have to redo all of the analysis: our posterior from the first analysis simply becomes our prior for this next analysis.

- Binomial example:

Stage 0. Prior $p(\theta) \sim$ Beta(1,1); ie $E(\theta)=0.5$.

Stage 1. Observe y=22 'presences' from 29 sites.
Likelihood: $p(y|\theta) \sim$ Bin(n=29,$\theta$);
Posterior: $p(\theta|y) \sim$ Beta(23,8); ie $E(\theta|y) = 0.74$

Stage 2: Observe 5 more 'presences' from 10 sites.
Likelihood: $p(y|\theta) \sim$ Bin(n=10,$\theta$);
Prior $p(\theta) \sim$ Beta(23,8);
Posterior $p(\theta|y) \sim$ Beta(28,13); ie $E(\theta|y) = 0.68$.

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Normal linear regression

'Usual' simple linear regression model:

$$y_i = a + b\,x_i + e_i$$
$$e_i \sim N(0, \sigma^2)$$

Equivalent model:

$$y_i \sim N(\mu_i, \sigma^2)$$
$$\mu_i = a + b\,x_i$$

Priors: a~Normal, b~Normal, $\sigma$~Uniform

# Naïve Bayes

- Supervised classification algorithm: developed using a training set of input variables and known class labels, and then applied to a test set with input variables but no class labels.
- The algorithm constructs a score based on the input data and uses this to assign the objects in the test set to classes
- For example, with two classes, objects with scores less than a certain threshold are allocated to one class and those with scores above the threshold are allocated to the other class.
- The score is computed based on the ratio of the probabilities of belonging to the different classes based on the data and on the prior. If nothing is known beforehand about the allocations and the training set is a random sample, the prior probability of belonging to a class can be estimated by the proportion of objects of that class in the training set.
- The naïve Bayes algorithm is very robust and easy to construct and compute, so it can be easily applied to huge datasets, and it is easy to interpret. It is very popular in many fields, including text classification and spam filtering.
- Many extensions have been developed, but even the basic algorithm works well.
- The logistic regression model can also be seen as a type of naïve Bayes classifier.

# Neural networks

Tutorial on neural networks

ai-junkie.com

http://www.ai-junkie.com/ann/evolved/nnt2.html

Tutorial on deep learning

http://deeplearning.net/tutorial/

Recurrent neural networks (and other deep learning methods, eg recurrent glms)

http://blog.shakirm.com/ml-series/a-statistical-view-of-deep-learning/

Deep learning applications and challenges (Journal of Big Data)

https://journalofbigdata.springeropen.com/articles/10.1186/s40537-014-0007-7

Crazy things to do with deep learning

http://www.kdnuggets.com/2015/11/crazy-deep-learning-topological-data-analysis.html

QUT

ACEMS

AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Neural Networks

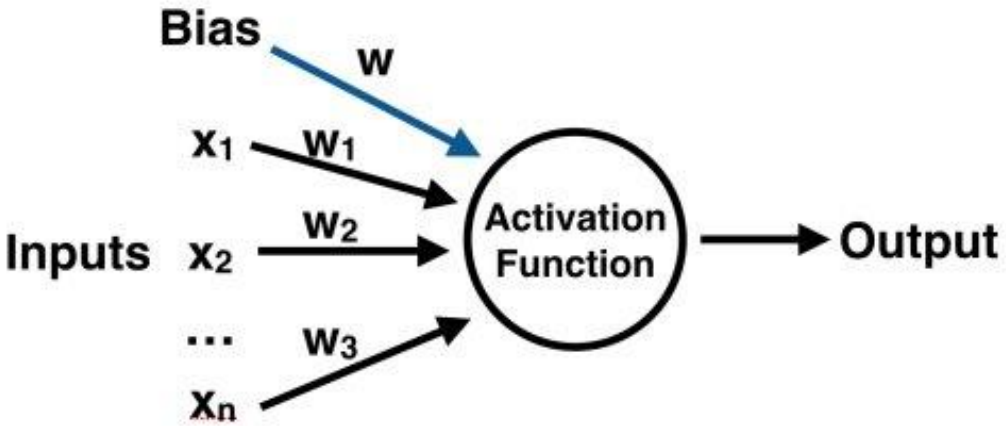
http://www.kdnuggets.com/2016/08/begineers-guide-neural-networks-r.html

- Neural Networks are a machine learning framework that attempts to mimic the learning pattern of natural biological neural networks.
- Biological neural networks have interconnected neurons with dendrites that receive inputs, then based on these inputs they produce an output signal through an axon to another neuron.
- We try to mimic this process through the use of Artificial Neural Networks (ANN), which we will just refer to as neural networks from now on.

# Neural Networks

http://www.kdnuggets.com/2016/08/begineers-guide-neural-networks-r.html

- The process of creating a neural network begins with the most basic form, a single perceptron.
- A perceptron has one or more inputs, a bias, an activation function, and a single output. The perceptron receives inputs, multiplies them by some weight, and then passes them into an activation function to produce an output. There are many possible activation functions to choose from, such as the logistic function, a trigonometric function, a step function etc. We also make sure to add a bias to the perceptron, this avoids issues where all inputs could be equal to zero (meaning no multiplicative weight would have an effect).



# Neural Networks

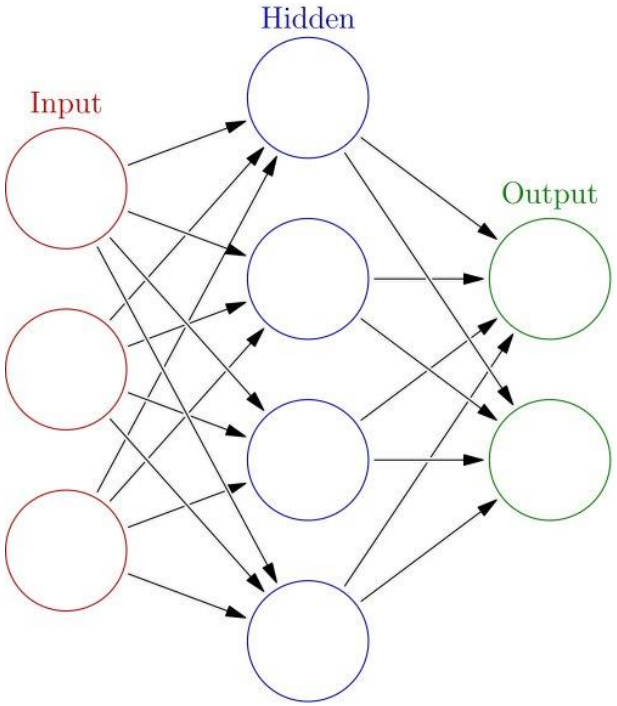
http://www.kdnuggets.com/2016/08/begineers-guide-neural-networks-r.html

Once we have the output we can compare it to a known label and adjust the weights accordingly (the weights usually start off with random initialization values). We keep repeating this process until we have reached a maximum number of allowed iterations, or an acceptable error rate.

To create a neural network, we simply begin to add layers of perceptrons together, creating a multi-layer perceptron model of a neural network. You'll have an input layer which directly takes in your feature inputs and an output layer which will create the resulting outputs. Any layers in between are known as hidden layers because they don't directly "see" the feature inputs or outputs.

# Neural Networks

http://www.kdnuggets.com/2016/08/begineers-guide-neural-networks-r.html




Run the example R code!

QUT

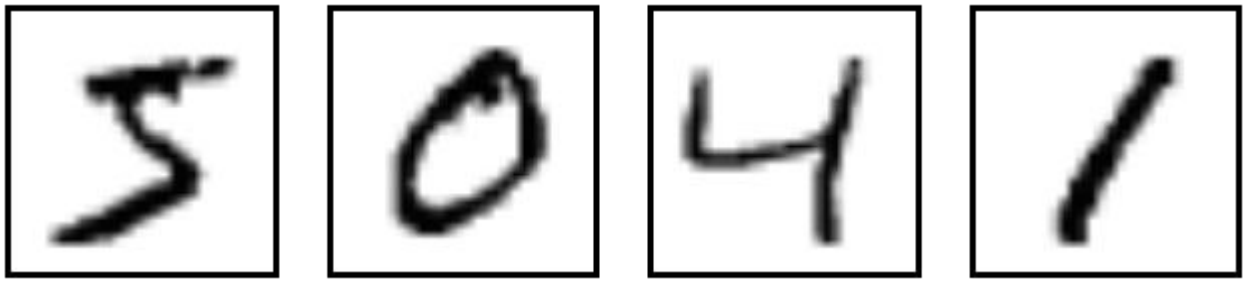
ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR
MATHEMATICAL AND STATISTICAL FRONTIERS

# Example

https://rstudio.github.io/tensorflow/tutorial_mnist_beginners.html#the_mnist_data

MNIST is a simple computer vision dataset.
It consists of images of handwritten digits like these:

It also includes labels for each image, telling us which digit it is.



Every MNIST data point has two parts: an image of a handwritten digit and a corresponding label. We'll call the images "x" and the labels "y". Both the training set and test set contain images and their corresponding labels

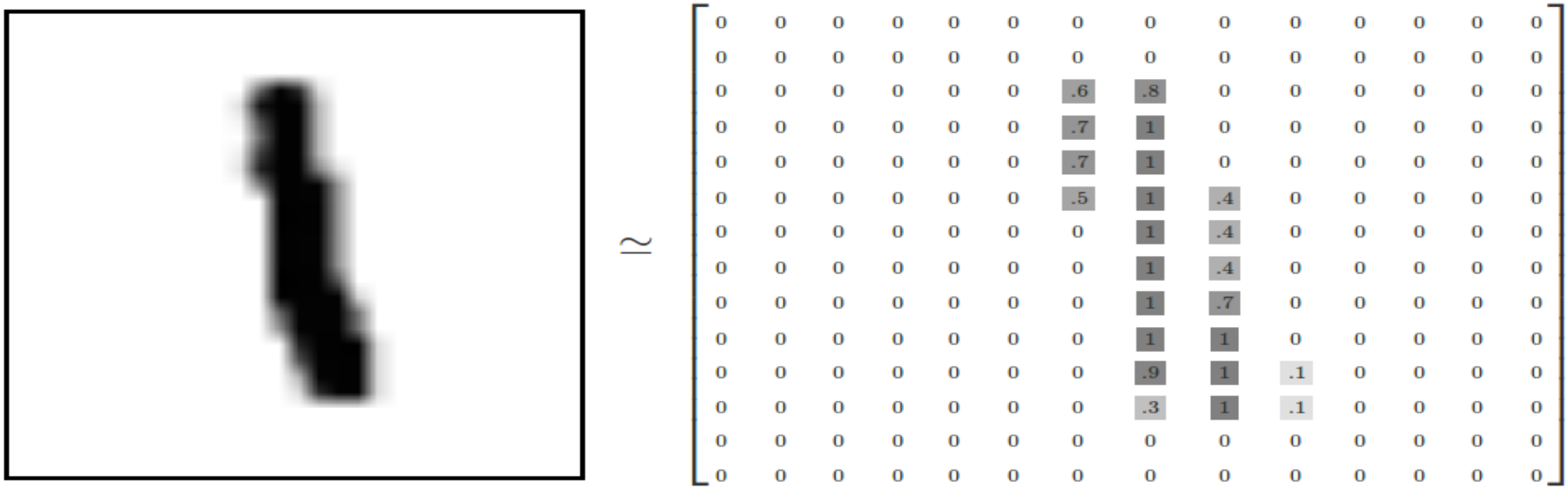
QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Example

https://rstudio.github.io/tensorflow/tutorial_mnist_beginners.html#the_mnist_data

Each image is 28 pixels by 28 pixels. We can interpret this as a big array of numbers:




We can flatten this array into a vector of 28x28 = 784 numbers. It doesn't matter how we flatten the array, as long as we're consistent between images. From this perspective, the MNIST images are just a bunch of points in a 784-dimensional vector space.

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Example

https://rstudio.github.io/tensorflow/tutorial_mnist_beginners.html#the_mnist_data

The result is that `mnist$train$images` is a tensor (an n-dimensional array) with shape `(55000L, 784L)`.

The first dimension is an index into the list of images and the second dimension is the index for each pixel in each image.

Each entry in the tensor is a pixel intensity between 0 and 1, for a particular pixel in a particular image.

Each image in MNIST has a corresponding label, a number between 0 and 9 representing the digit drawn in the image.

Analyse use TensorFlow (see above link)

QUT

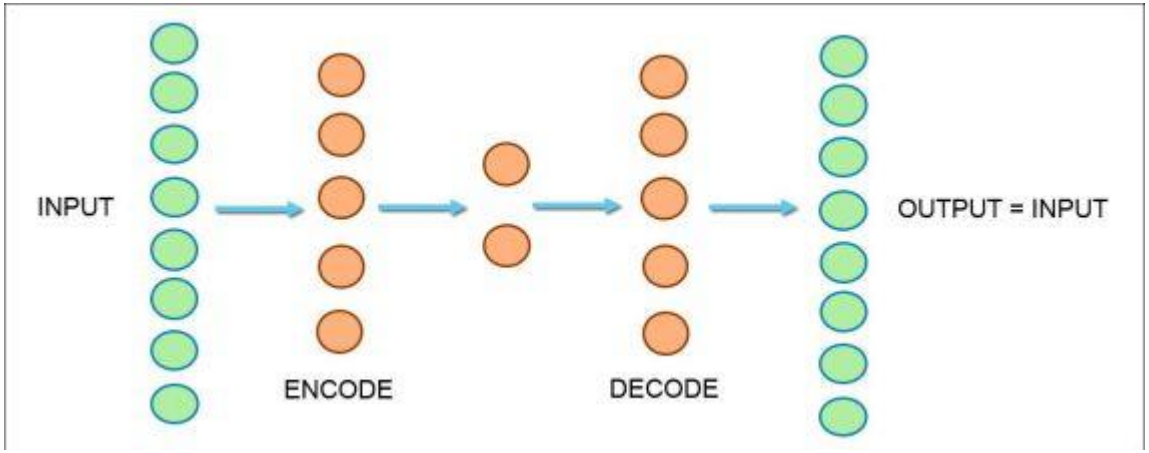
ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Example

https://www.r-bloggers.com/a-little-h2o-deeplearning-experiment-on-the-mnist-data-set/

R code for reading in images

Analyse using an autoencoder




QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Example

https://www.r-bloggers.com/a-little-h2o-deeplearning-experiment-on-the-mnist-data-set/

NN_model = h2o.deeplearning( x = 2:785, training_frame = MDIG,
hidden = c(400, 200, 2, 200, 400 ), epochs = 600,
activation = "Tanh", autoencoder = TRUE)

So there is one input layer with 784 neurons,
a second layer with 400 neurons,
a third layer with 200,
the middle layer with 2 neurons, etc.

The middle layer is a 2-dimensional representation of a 784 dimensional digit.
The 42.000 2-dimensional representations of the digits are just points that we can plot.

# Example

https://www.r-bloggers.com/a-little-h2o-deeplearning-experiment-on-the-mnist-data-set/

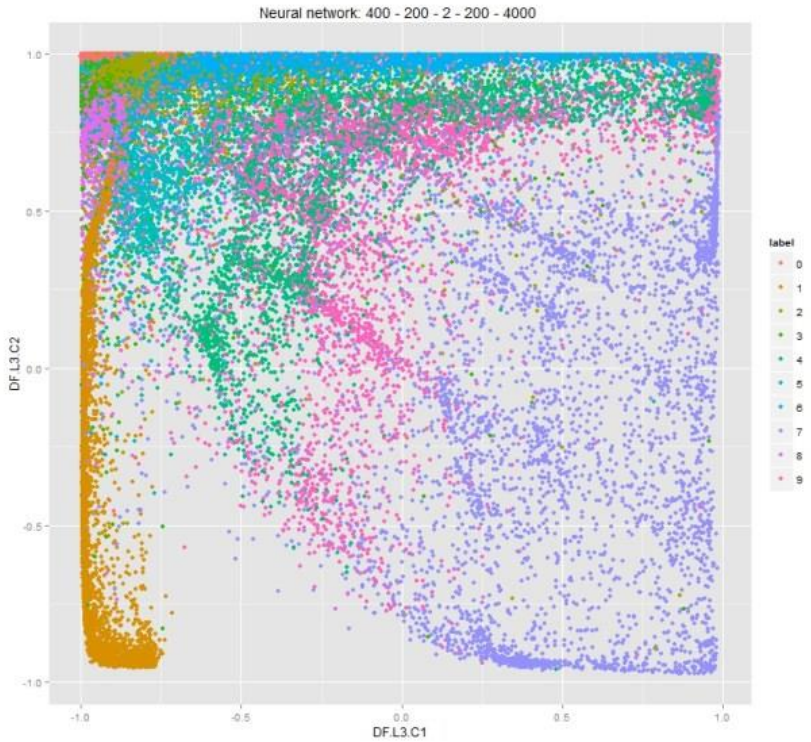
To extract the data from the middle layer we need to use the function h20.deepfeatures.

```
train_supervised_features2 = h2o.deepfeatures(NN_model, MDIG,
layer=3)

plotdata2 = as.data.frame(train_supervised_features2)
plotdata2$label = as.character(as.vector(MDIG[,1]))
qplot(DF.L3.C1, DF.L3.C2, data = plotdata2, color = label,
   main = &amp;quot;Neural network: 400 - 200 - 2 - 200 - 4000 &amp;quot;)
```

# Example

https://www.r-bloggers.com/a-little-h2o-deeplearning-experiment-on-the-mnist-data-set/




We can see the ‘1’ digits clearly on the left-hand side, while the ‘7’ digits are more on the right-hand side, and the pink ‘8’ digits are more in the center.

QUT

ACEMS
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

# Discussion

## Neural networks for Big Data

See how google's Neural Nets process images
https://research.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html

There's a Python notebook that allows you to edit the code and upload your own images

https://github.com/google/deepdream

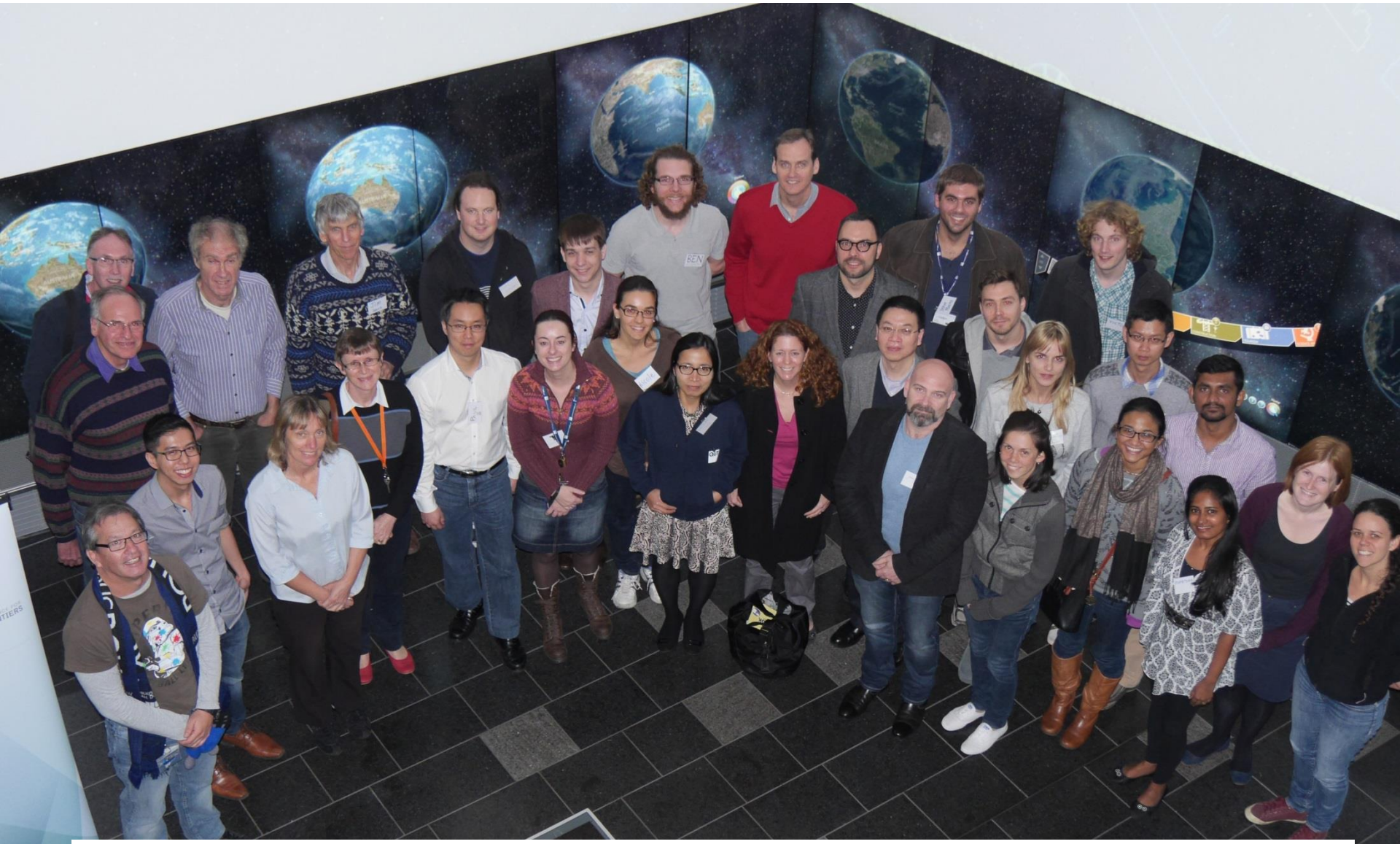
# Discussion

"Deep learning"

https://www.analyticsvidhya.com/blog/2015/11/free-resources-beginners-deep-learning-neural-network/

# Prac: Regression

- Description of prac



With acknowledgements to our awesome QUT & ACEMS teams and collaborators!