Machine learning, Statistics and Big Data

3-day short course

Participants: Australian Bureau of Statistics

Presenters: Hugh Anderson, Jacinta Holloway, Miles McBain, James McGree, Chris McCool, Kerrie Mengersen

Queensland University of Technology

October 2017

Program: Day 1

Time	Presentation
9.30am – 10.00am	Registration and Coffee
10.00am – 12.00pm	Welcome and overview of course 1. Overview of big data 2. Overview of stats & ML for big data: concepts, philosophy, terminology 3. Overview of computational frameworks: from divide & recombine to cloud computing 4. Case Study: grading images
12.00pm – 12.45pm	Lunch
12.45am – 2.45pm	1. Preparing your data 2. Overview of methods 3. Overview of algorithms
2.45pm – 3.00pm	Break
3.00pm – 4.30pm	Digging Deeper: Classification and Regression. 1. Generalised linear regression 2. Spatial and time series models 3. Tree-based approaches: CART, RF, BRT, bagging boosting 4. Support vector machines
4.30pm – 5.00pm	Extended Topics: Classification and Regression Semi-parametric regression, KNNs, Ensembles, XGBoost Discussion of cloud computing Concluding remarks: Day 1

Program: Day 2

Time	Presentation
9.30am – 10am	Coffee
10am – 12.00pm	Brief recap and discussion Digging Deeper: Clustering and Dimension Reduction. 1. kmeans 2. Mixture models 3. Feature extraction 4. PCA, FA and extensions 5. Page Rank
12.00pm – 12.45pm	Lunch
12.45am – 2.45pm	Digging Deeper: Neural networks 1. Overview of NNs 2. Convolutional and recurrent NNs 3. Deep learning
2.45pm – 3.00pm	Break
3.00pm – 4.30pm	Extended topics: NNs NNs for time series and 2D images
4.30pm – 5.00pm	Extended topics: NNs Deep Learning Systems Concluding remarks: Day 2

Program: Day 3

Time	Presentation
9.30am – 10.00am	Registration and Coffee
10.00am – 12.00am	Brief recap and discussion Case Study: Recommender systems. 1. Overview of recommender systems 2. Implementation 3. Use cases
12.00am – 12.45pm	Lunch
12.45pm – 2.45pm	The ABS context: Special Session. 1. Presentations from invited speakers 2. Discussion
2.45pm – 3.00pm	Break
3.00pm – 4.30pm	Extended Topics: 1. Overview of semi-supervised learning and ensembles of weak learners 2. Case study: return to classifying images
4.30pm – 5.00pm	Final issues Where to from here Concluding remarks: Day 3 Close

Day 1
Session 1

1. Overview of big data

2. Overview of stats & machine learning: concepts, philosophy, terminology

3. Overview of computational frameworks

4. Case Study: grading images

Definitions of big data

"volume, velocity, variety"

+ "veracity, value"

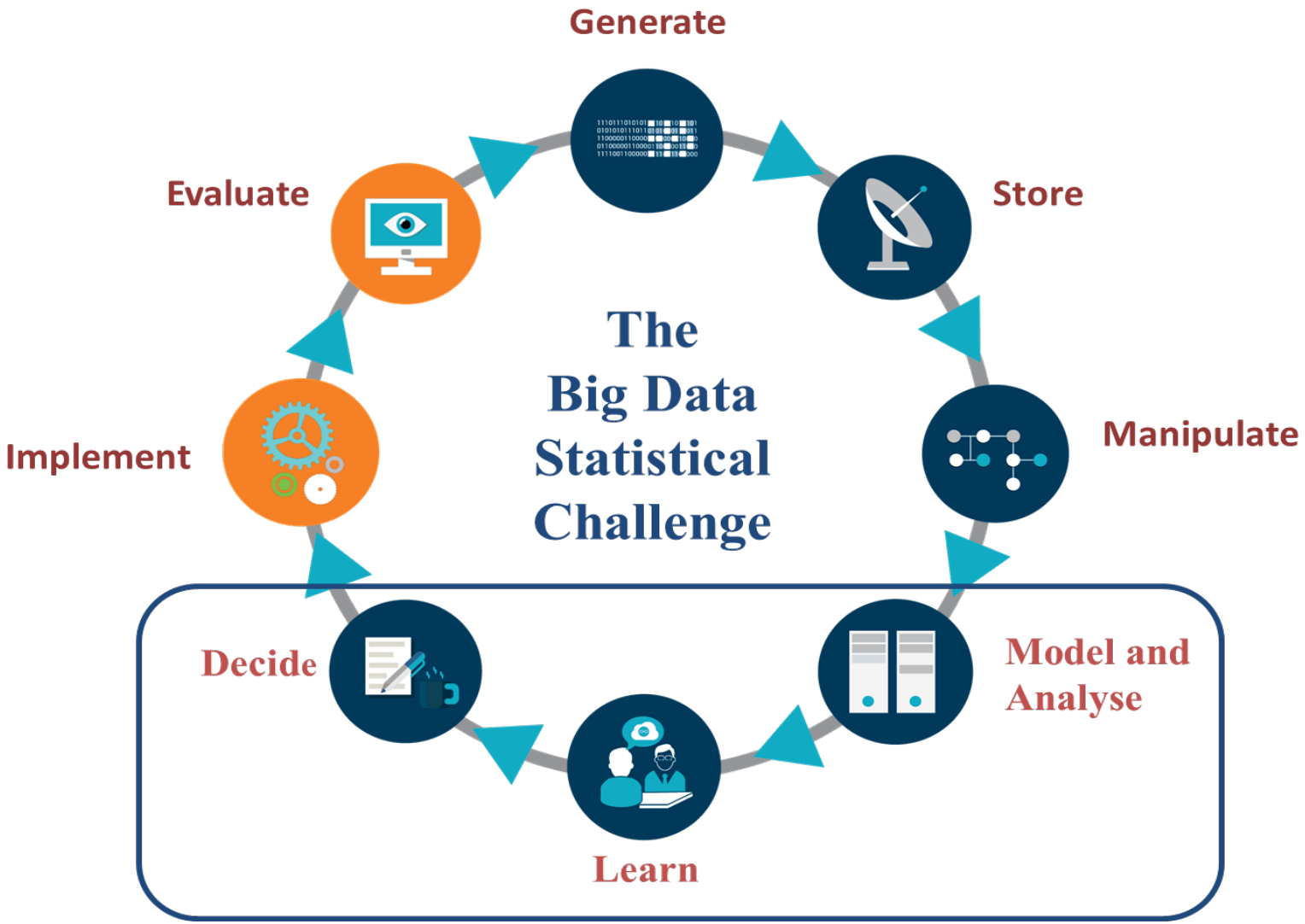
+ "complexity, …"

"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

The Big Data Wheel

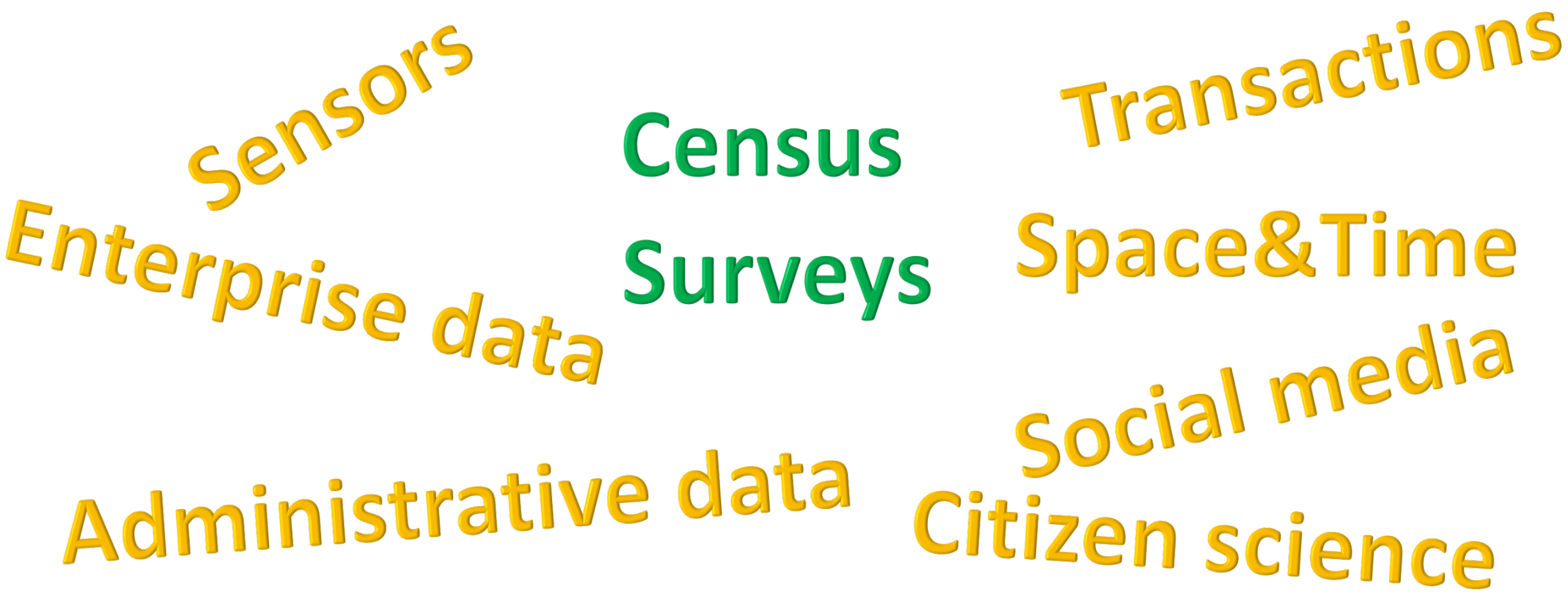


Sources of big data

Census

Surveys

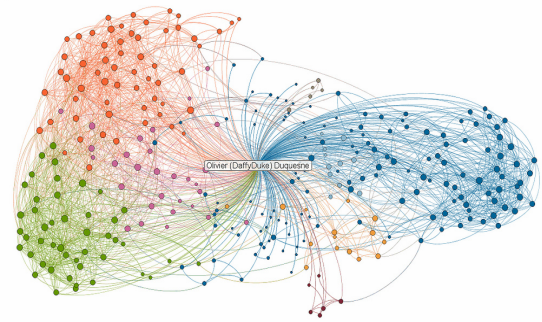
Sources of big data

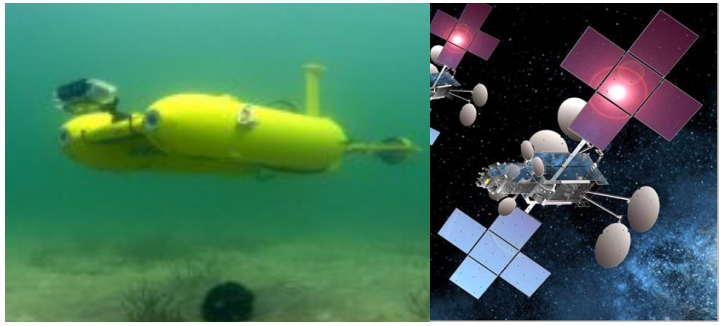


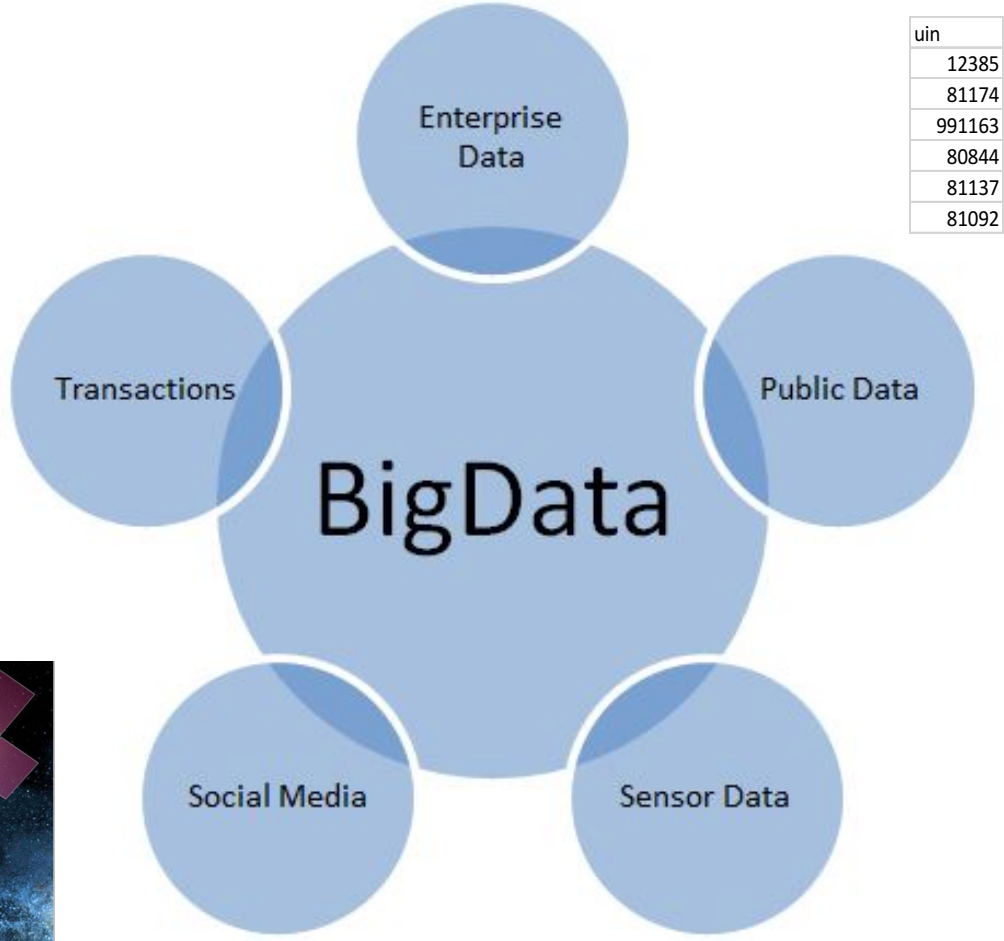
Data are increasingly gathered by cheap and numerous information-sensing mobile devices, aerial (remote sensing), software logs, cameras, microphones, radio-frequency identification (RFID) readers and wireless sensor networks.
https://en.wikipedia.org/wiki/Big_data

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

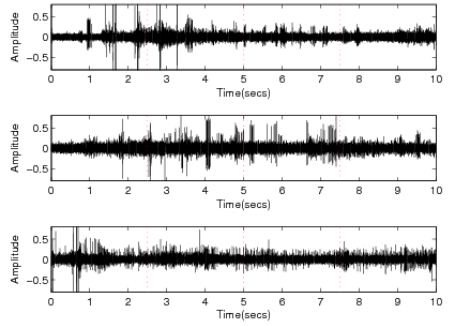
Sources of big data







uin	dob	date	tagz	age	sex	seg	status	type	code
12385	########	########	0	21.79603	female	Tech servi	employee	Initial	Amber
81174	########	########	0	21.71116	female	Plant oper	contractor	Periodic	Green
991163	########	########	0	21.27584	male	Developm	contractor	Initial	Green
80844	########	########	0	27.37577	female	Tech servi	employee	Initial	Green
81137	########	########	0	19.66598	male	Tech servi	employee	Initial	Green
81092	########	########	0	24.37509	female	Plant oper	contractor	Initial	Green



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/

ACEMS-QUT MOOC

https://www.futurelearn.com/courses/big-data-decisions

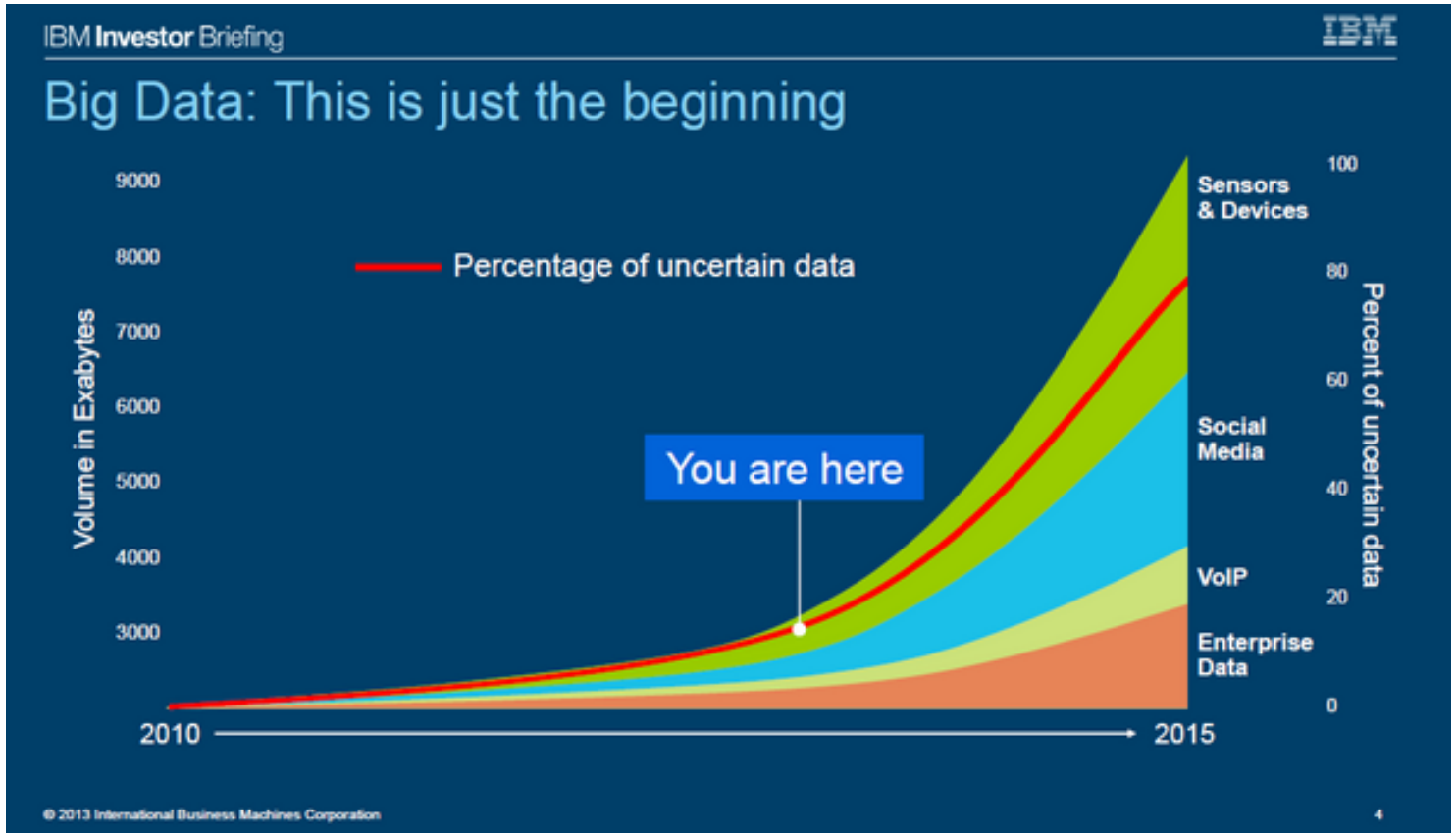
“Data management” challenges

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

“Dirty data” challenges

How to cope with:

- Size
- Quality
- Diversity
- Uncertainty



For interest

Big data Rap

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

Big data and Pokemon

https://pixelastic.github.io/pokemonorbigdata/

For interest

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

For interest

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

Day 1
Session 1

1. Overview of big data

2. Overview of stats & machine learning: concepts, philosophy, terminology

3. Overview of computational frameworks

4. Case Study: grading images

What *is* machine learning?

Machine learning is "concerned with the question of how to construct computer programs that automatically improve with experience."

The main artefacts of machine learning research are algorithms which facilitate this automatic improvement from experience.

Machine learning is interdisciplinary in nature, and employs techniques from the fields of computer science, statistics, and artificial intelligence, among others.

https://www.kdnuggets.com/2016/05/machine-learning-key-terms-explained.html

What *is* statistics?

Statistics is a branch of mathematics dealing with the collection, analysis, interpretation, presentation, and organization of data.

We begin with a statistical population or a statistical model process to be studied, and make inferences based on a sample of data.

Statistics deals with all aspects of data including the planning of data collection in terms of the design of surveys and experiments.

Two main statistical methods are used in data analysis: descriptive statistics and inferential statistics.

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

Machine Learning vs Statistical Modelling

Machine Learning is *an algorithm that can learn from data without relying on rules based programming.*

Statistical Modelling is *the formalization of relationships between variables in the form of mathematical equations.*

Machine Learning emphasises *prediction*

Statistical Modelling emphasises *inference*

Machine Learning focuses on *large datasets*

Statistical Modelling focuses on *small datasets*

https://www.analyticsvidhya.com/blog/2015/07/difference-machine-learning-statistical-modeling/

https://medium.com/data-science-analytics/statistical-learning-vs-machine-learning-f9682fdc339f

Different world views: terminology

Stats: models and algorithms

row – single entity/observation
column – single data type, describes a property of the entity

dependent variable = f(independent variables)

$Y = f(X)$

SML: output = f(input)

https://machinelearningmastery.com/data-terminology-in-machine-learning/

Different world views: terminology

CS: learned representation and process for learning

row = attributes – describes an entity or observation about an entity

row = instance – a single example or single instance of data observed or generated by the problem domain

column = features – describes a property of the entity

features can be extracted from raw data to make an observation

output attribute = program(input attributes)

output = program(input features)

prediction = program(instance)

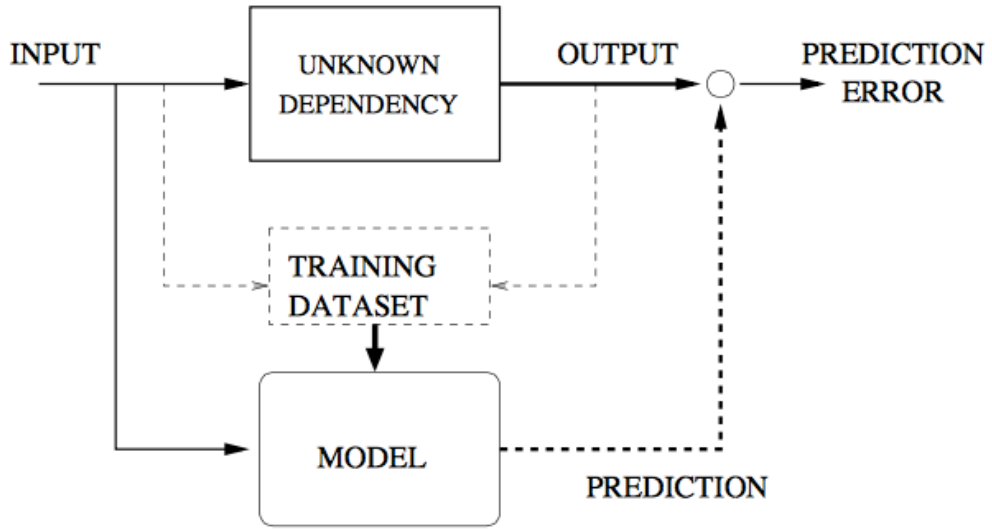
https://machinelearningmastery.com/data-terminology-in-machine-learning/

Different world views: terminology

Statistics	Computer Science	Meaning
estimation	learning	using data to estimate an unknown quantity
classification	supervised learning	predicting a discrete Y from $X \in \mathcal{X}$
clustering	unsupervised learning	putting data into groups
data	training sample	$(X_1, Y_1), \ldots, (X_n, Y_n)$
covariates	features	the X_i's
classifier	hypothesis	a map from covariates to outcomes
hypothesis	—	subset of a parameter space Θ
confidence interval	—	interval that contains unknown quantity with a prescribed frequency
directed acyclic graph	Bayes net	multivariate distribution with specified conditional independence relations
Bayesian inference	Bayesian inference	statistical methods for using data to update subjective beliefs
frequentist inference	—	statistical methods for producing point estimates and confidence intervals with guarantees on frequency behavior
large deviation bounds	PAC learning	uniform bounds on probability of errors

https://machinelearningmastery.com/data-terminology-in-machine-learning/

Case Study: Supervised Learning



From now on we consider the prediction problem as a problem of **supervised learning** problem, where we have to infer from historical data the possibly nonlinear dependance between the input (past embedding vector) and the output (future value).

Statistical machine learning is the discipline concerned with this problem.

Botempi, G. 2013. Machine learning strategies for time series prediction. http://www.ulb.ac.be/di/map/gbonte/ftp/time_ser.pdf

- A typical way of representing the unknown input/output relation is the **regression plus noise form**

$$\mathbf{y} = f(x) + \mathbf{w}$$

where $f(\cdot)$ is a deterministic function and the term $\mathbf{w}$ represents the noise or random error. It is typically assumed that $\mathbf{w}$ is independent of $\mathbf{x}$ and $E[\mathbf{w}] = 0$.

- Suppose that we have available a **training set** $\{\langle x_i, y_i \rangle : i = 1, \ldots, N\}$, where $x_i = (x_{i1}, \ldots, x_{in})$ and y_i, generated according to the previous model.
- The goal of a learning procedure is to estimate a model $\hat{f}(x)$ which is able to give a good approximation of the unknown function $f(x)$.
- But how to choose $\hat{f}$, if we do not know the probability distribution underlying the data and we have only a limited training set?

Botempi, G. 2013. Machine learning strategies for time series prediction. http://www.ulb.ac.be/di/map/gbonte/ftp/time_ser.pdf

- How to estimate the quality of a model? Is the training error a good measure of the quality?
- The goal of learning is to find a model which is able to **generalize**, i.e. able to return good predictions for input values independent of the training set.
- In a nonlinear setting, it is possible to find models with such a complicate structure that they have null training errors. Are these models good?
- Typically NOT. Since doing very well on the training set could mean doing badly on new data.
- This is the phenomenon of **overfitting**.
- Using the same data for training a model and assessing it is typically a wrong procedure, since this returns an over optimistic assessment of the model generalization capability.

Botempi, G. 2013. Machine learning strategies for time series prediction. http://www.ulb.ac.be/di/map/gbonte/ftp/time_ser.pdf

- A fundamental result of estimation theory shows that the mean-squared-error, i.e. a measure of the generalization quality of an estimator can be decomposed into three terms:

$$\text{MISE} = \sigma^2_{\mathbf{w}} + \text{squared bias} + \text{variance}$$

  where the intrinsic noise term reflects the target alone, the bias reflects the target's relation with the learning algorithm and the variance term reflects the learning algorithm alone.
- This result is purely theoretical since these quantities cannot be measured on the basis of a finite amount of data.
- However, this result provides insight about what makes accurate a learning process.

Botempi, G. 2013. Machine learning strategies for time series prediction. http://www.ulb.ac.be/di/map/gbonte/ftp/time_ser.pdf

- The first term is the variance of y around its true mean $f(x)$ and cannot be avoided no matter how well we estimate $f(x)$, unless $\sigma^2_{\mathbf{w}} = 0$.
- The bias measures the difference in x between the average of the outputs of the hypothesis functions $\hat{f}$ over the set of possible D_N and the regression function value $f(x)$
- The variance reflects the variability of the guessed $\hat{f}(x, \boldsymbol{\alpha}_N)$ as one varies over training sets of fixed dimension N. This quantity measures how sensitive the algorithm is to changes in the data set, regardless to the target.

Botempi, G. 2013. Machine learning strategies for time series prediction. http://www.ulb.ac.be/di/map/gbonte/ftp/time_ser.pdf

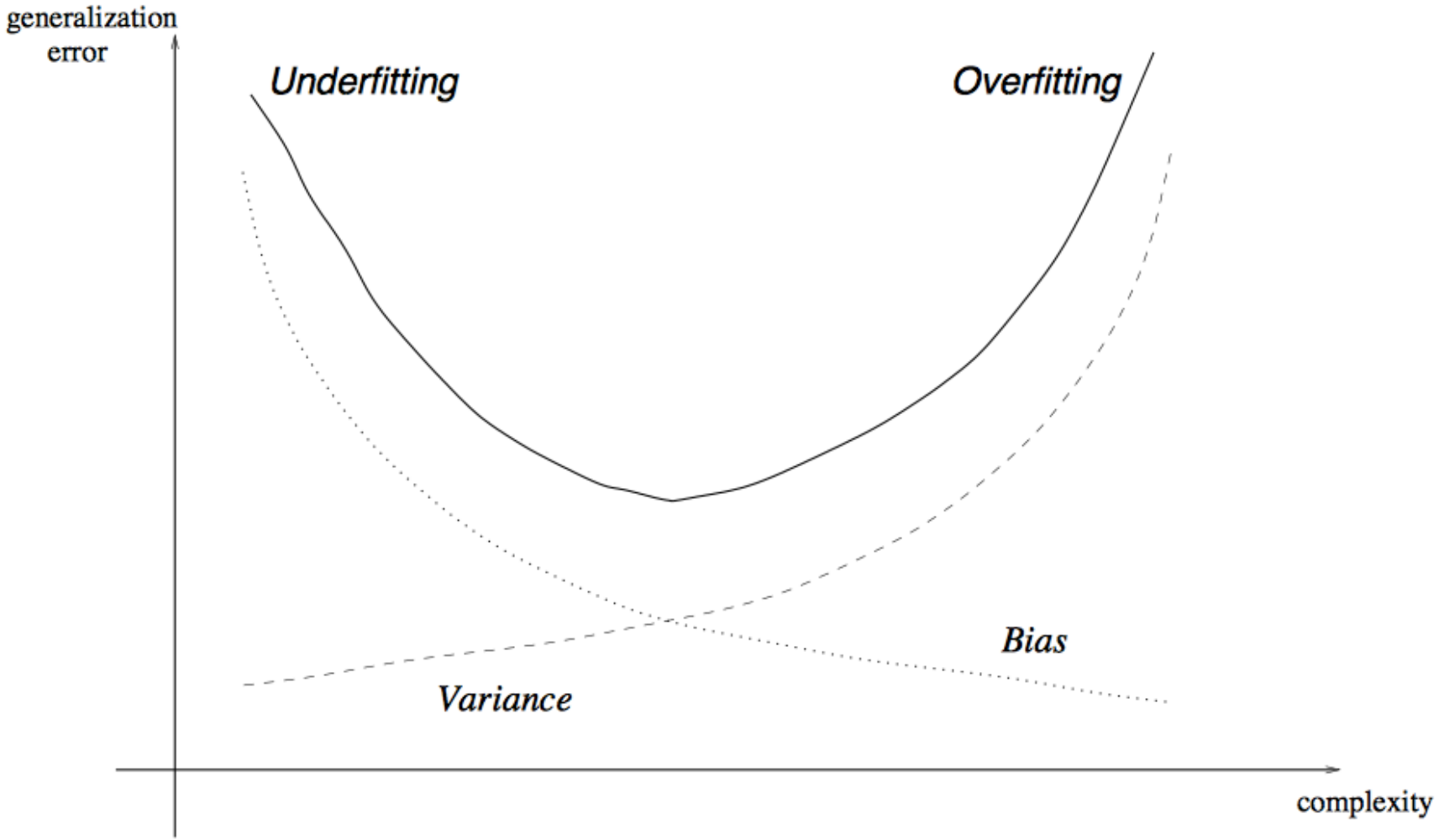
- The designer of a learning machine has not access to the term MISE but can only estimate it on the basis of the training set. Hence, the bias/variance decomposition is relevant in practical learning since it provides a useful hint about the features to control in order to make the error MISE small.
- The bias term measures the lack of representational power of the class of hypotheses. To reduce the bias term we should consider complex hypotheses which can approximate a large number of input/output mappings.
- The variance term warns us against an excessive complexity of the approximator. This means that a class of too powerful hypotheses runs the risk of being excessively sensitive to the noise affecting the training set; therefore, our class could contain the target but it could be practically impossible to find it out on the basis of the available dataset.

Botempi, G. 2013. Machine learning strategies for time series prediction. http://www.ulb.ac.be/di/map/gbonte/ftp/time_ser.pdf

- In other terms, it is commonly said that an hypothesis with large bias but low variance *underfits* the data while an hypothesis with low bias but large variance *overfits* the data.
- In both cases, the hypothesis gives a poor representation of the target and a reasonable trade-off needs to be found.
- The task of the model designer is to search for the optimal trade-off between the variance and the bias term, on the basis of the available training set.

Botempi, G. 2013. Machine learning strategies for time series prediction. http://www.ulb.ac.be/di/map/gbonte/ftp/time_ser.pdf

Bias/variance trade-off



Botempi, G. 2013. Machine learning strategies for time series prediction. http://www.ulb.ac.be/di/map/gbonte/ftp/time_ser.pdf

Glossary of ML terms

Accuracy (error rate) the rate of correct (incorrect) predictions made by the model over a data set
Association learning techniques that find conjunctive implication rules of the form
``X and Y implies A and B'' (associations) that satisfy given criteria.
Attribute (field, variable, feature) a quantity describing an instance;
an attribute has a domain defined by the attribute type, which denotes the values that can be taken by an attribute
Categorical
Continuous (quantitative)
Classifier a mapping from unlabeled instances to (discrete) classes
Confusion matrix
Coverage the proportion of a data set for which a classifier makes a prediction
Cost (utility/loss/payoff)
Cross-validation
Data cleaning/cleansing
Data mining
Data set
Dimension
Error rate

http://robotics.stanford.edu/~ronnyk/glossary.html

Glossary (cont.)

Example see Instance
Feature see Attribute
Feature vector (record, tuple)
Field see Attribute
i.i.d. sample
Inducer / induction algorithm
Instance (example, case, record)
Knowledge discovery
Loss
Machine learning
Missing value
Model (A structure and corresponding interpretation that summarizes or partially summarizes a set of data, for description or prediction)
Model deployment
OLAP (MOLAP, ROLAP) (On-Line Analytical Processing)

http://robotics.stanford.edu/~ronnyk/glossary.html

Glossary (cont.)

Record (see Feature vector)
Regressor (A mapping from unlabeled instances to a value within a predefined metric space)
Resubstitution accuracy (error/loss)
Schema (a description of a data set's attributes and their properties)
Sensitivity (see Confusion matrix)
Specificity (see Confusion matrix)
Supervised learning (Techniques used to learn the relationship between independent attributes and a designated dependent attribute (the label)).
Tuple (see Feature vector)
Unsupervised learning (Learning techniques that group instances without a pre-specified dependent attribute.
Utility (see Cost)

http://robotics.stanford.edu/~ronnyk/glossary.html

A final say…

Glossary	
Machine learning	**Statistics**
network, graphs	model
weights	parameters
learning	fitting
generalization	test set performance
supervised learning	regression/classification
unsupervised learning	density estimation, clustering
large grant = $1,000,000	large grant = $50,000
nice place to have a meeting: Snowbird, Utah, French Alps	nice place to have a meeting: Las Vegas in August

Rob Tibshirani

Day 1
Session 1

1. Overview of big data

2. Overview of stats & machine learning:
   concepts, philosophy, terminology

3. Overview of computational frameworks

4. Case Study: grading images

Big Data platforms

- Bespoke Platforms:
  - Storage/Manipulation/Analysis
  - Hadoop, Apache Mahout, Apache Spark and MMLib, TensorFlow
  - Alteryx

- Enterprise Platforms:
  - Google, Amazon, Microsoft

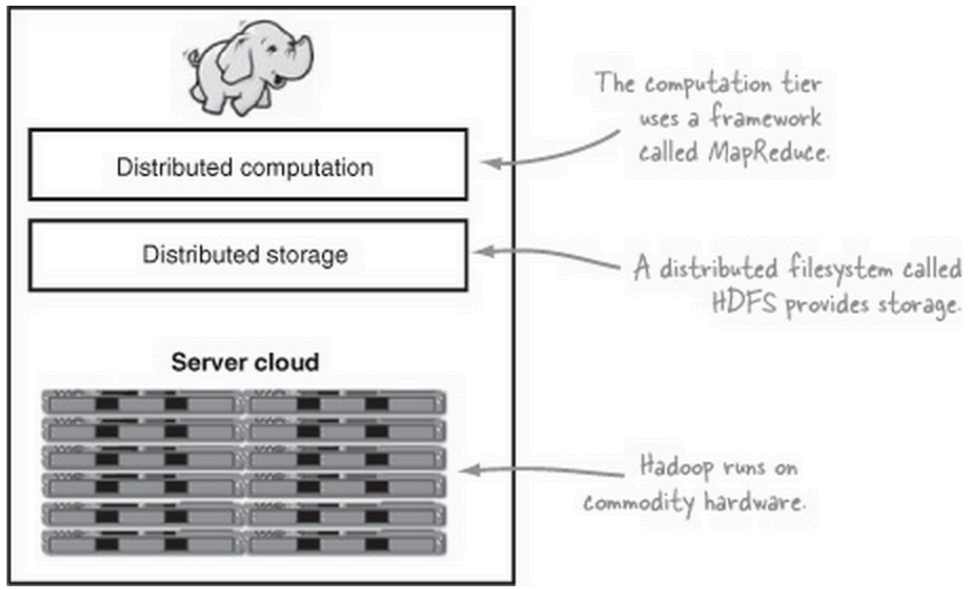
Source: CSIRO Big Data Workshop 2014

Hadoop

- The **Apache Hadoop** is a framework that allows for distributed processing of large data sets across cluster of computers using simple programming models.
- Designed to scale up from single servers to thousands of machines, each offering local computation and storage.
- Available under http://hadoop.apache.org

Hadoop

- Hadoop components:
  - **Hadoop Distributed File System** (HDFS)
  - **MapReduce**
- Hadoop handles any data type
  - Structured
  - Unstructured
  - Schema
  - No schema
  - High volume
  - Low volume



How Hadoop Works

HDFS

- A single large file is split into blocks, and the blocks are distributed among the nodes.
- Blocks in HDFS are large – typically 128MB in size.
- Files in HDFS are processed by MapReduce.
- Results stored back in HDFS.
- Original data file is not modified.

MapReduce

- A method for distributing a task across multiple nodes (parallel data processing)
  - Automatic parallelisation and distribution
  - Each node processes data stored on that node
  - Status and monitoring tools

Big data computing

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

Big data computing

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

BD analysis platforms

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

Source: CSIRO Big Data Workshop 2014

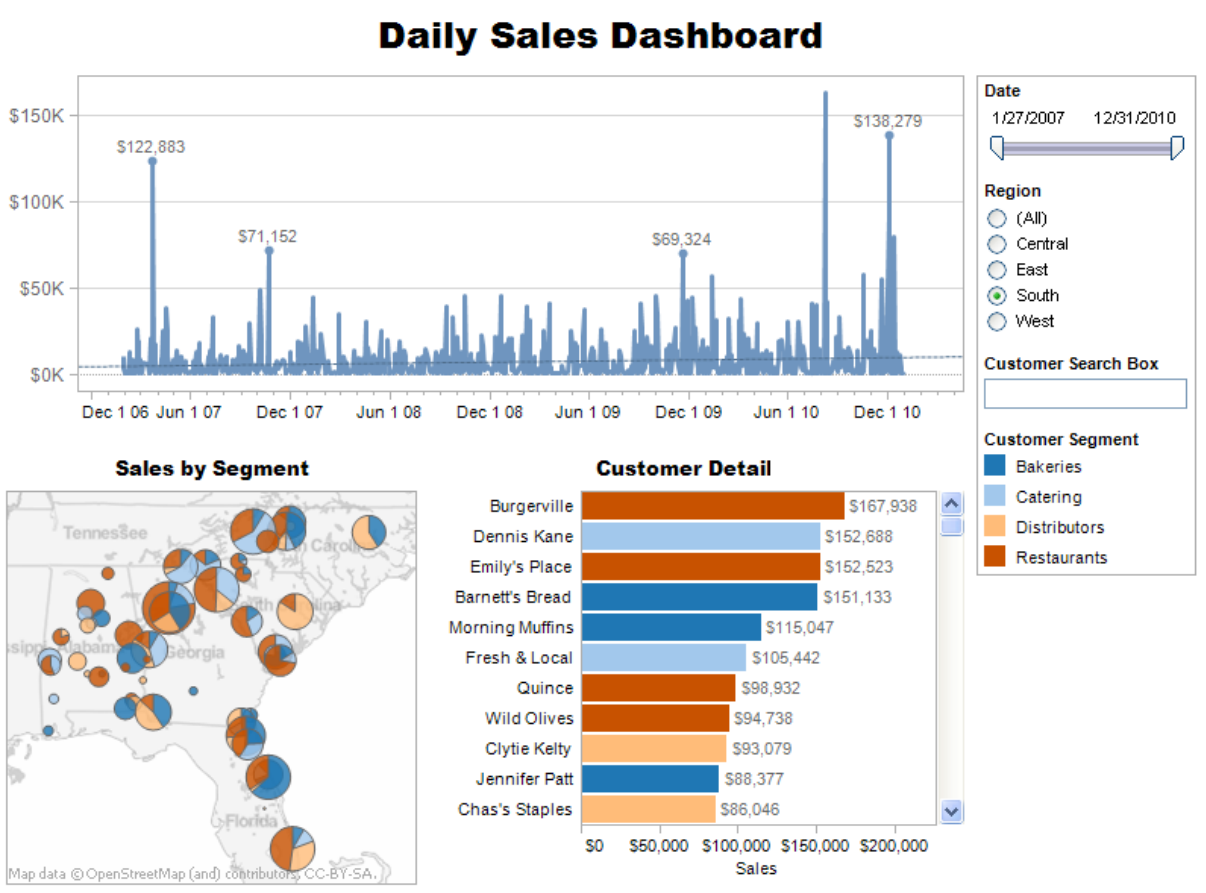
Analytics Software (SelectHub)

- 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

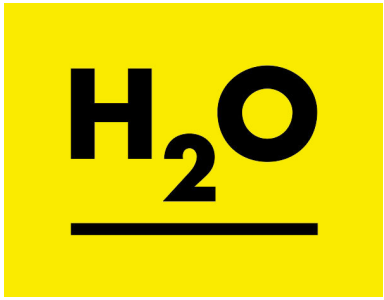
Analytics Platforms

Tableau

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



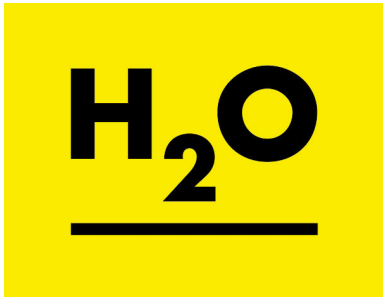
H_2O



"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 …)

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

Spark + H_2O

SPARKLING WATER

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/

Big data analysis in the cloud

Over to Miles…

Informative reference site:

https://cloud.google.com/blog/big-data/

Day 1
Session 1

1. Overview of big data

2. Overview of stats & machine learning: concepts, philosophy, terminology

3. Overview of computational frameworks

4. Case Study: grading images

Case study: Grading images

Over to Miles…

Day 1
Session 2

1. Preparing your data

2. Overview of methods

3. Overview of algorithms

Dirty data and tidy data

Over to Miles…

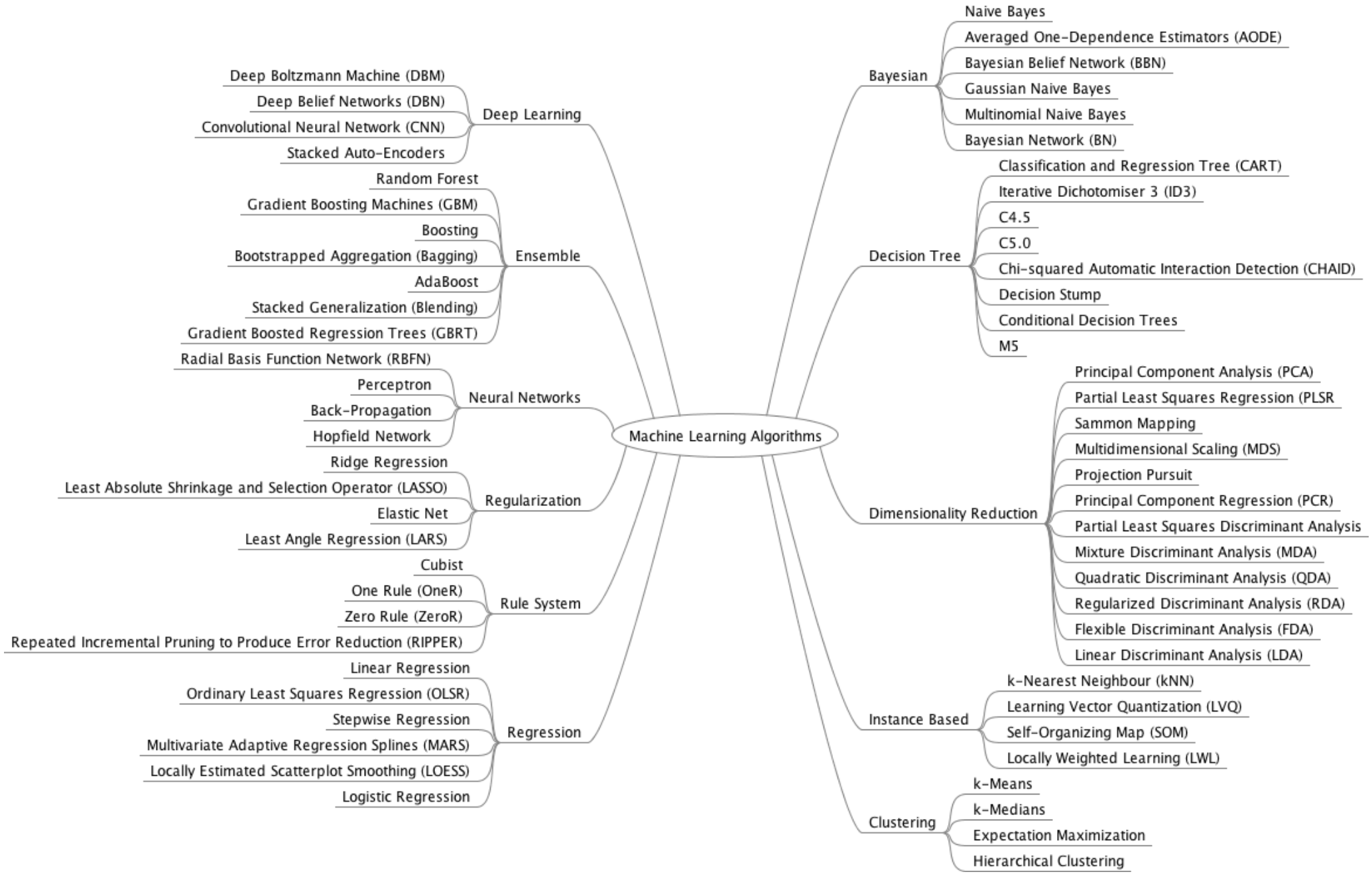
Wickham (2014):
https://www.jstatsoft.org/article/view/v059i10

Day 1
Session 2

1. Preparing your data

2. Overview of methods

3. Overview of algorithms



https://s3.amazonaws.com/MLMastery/MachineLearningAlgorithms.png?__s=17oupmsuthrzaaubgepa

Stats & ML approaches

- Four common analytic problems: **classification, regression, clustering** and **dimension reduction.**

- Broadly categorised by whether the problem is **supervised** or **unsupervised**.

Classification: supervised learning with a categorical response

- 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).

- 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.

Classification: common approaches

- **Decision trees (CART)**
- **Bayesian networks**
- **K-nearest neighbour (KNN)**
- **Support Vector Machines (SVM)**
- **Logistic and multinomial regression**

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.

Bayesian networks - 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. 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.

Clustering: unsupervised learning

- 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.

Clustering: common approaches

- **K-means**
- **Agglomerative clustering**
- **Mixture models**

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.

Regression: supervised learning with a continuous or discrete response

- 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.

Regression: common approaches

- **Linear methods**
- **Regression trees**
- **Gradient boosted machines**
- **Neural networks**

Regression: common approaches

- **Linear methods** – Construct a weighted linear combination of the input variables (or functions of these variables) that provides 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.

Dimension reduction: unsupervised learning

- 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 new output variables should maximise the information in the data.

- Dimension reduction can be used to create a small set of output variables that can effectively carry most 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. Another common use is as a method to create ‘indices’, e.g. weather index, psychological index.

Dimension reduction: common approaches

- **Principal components analysis**
- **Factor analysis**

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.

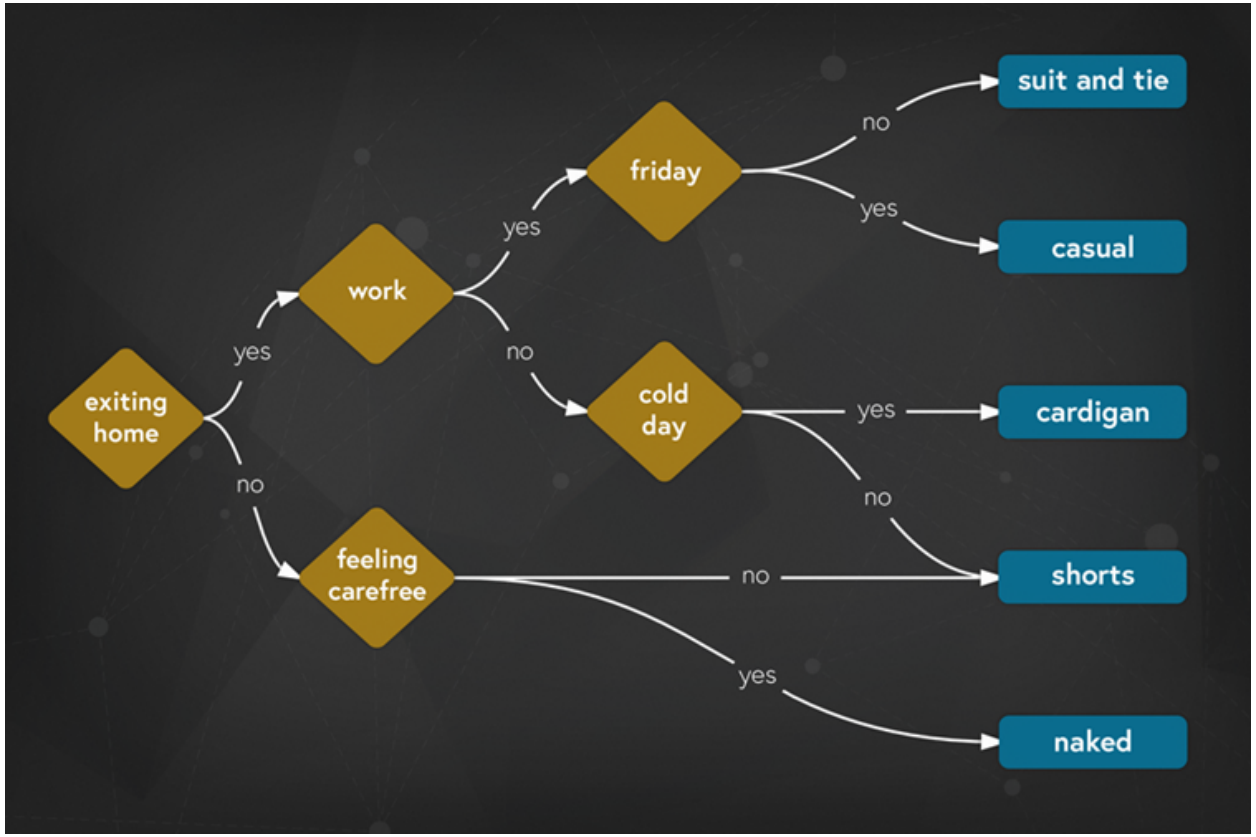
- A reinforcement learning agent is embedded in an environment which it can influence through some pre-defined actions. It then receives a feedback for this action in the form of a reward and state/observation signal which influences what actions it chooses in the future. This action observation reward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

For interest:
“What will I wear?”



Day 1
Session 2

1. Preparing your data

2. Overview of methods

3. Overview of algorithms

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

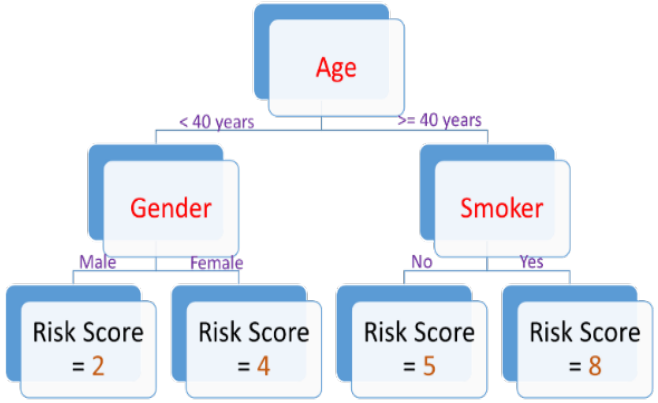
- These techniques are used for classification, clustering, statistical learning, association analysis and link mining.

Top 10 algorithms

- CART, C4.5, See5, C5.0
- Boosted Regression Tree
- Random Forests
- K-Means
- Support Vector Machine
- A-Priori
- Expectation-Maximisation (EM)
- Page Rank
- ADA Boost
- K Nearest Neighbours (KNN)
- Naïve Bayes

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.



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

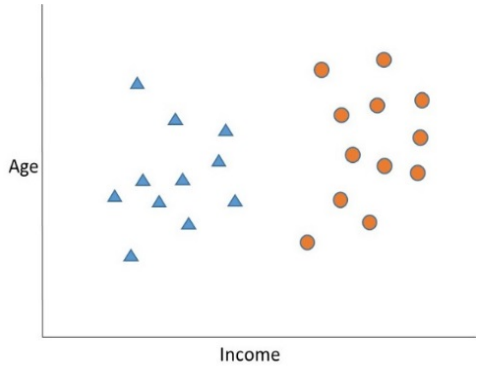
Random forests

- 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.

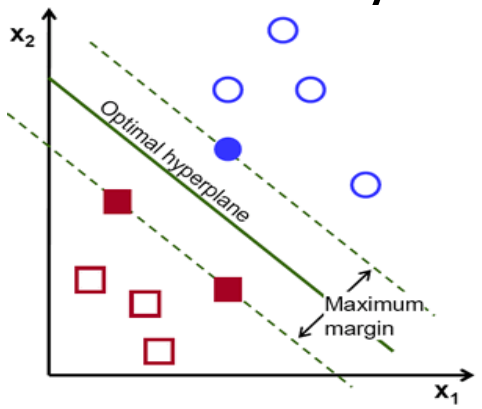
https://www.r-bloggers.com/k-means-clustering-in-r/



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)



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.

For interest

Some package resources

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

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

For interest

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

Day 1
Session 3

Digging Deeper: Classification and Regression

1. General and Generalised linear regression
2. Spatial and time series models
3. Tree-based approaches: CART, RF, BRT, bagging boosting
4. Support vector machines

Linear 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

Linear Regression and lasso

- **lasso (least absolute shrinkage and selection operator)**: a regression analysis method that performs both variable selection and regularisation (introducing additional information to prevent overfitting) in order to enhance the prediction accuracy and interpretability of the statistical model it produces.
- Lasso regularization is easily extended to a wide variety of statistical models including glms in a straightforward fashion.

https://en.wikipedia.org/wiki/Lasso_(statistics)

(Online) real-time regression

- 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

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

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

Generalised regression (GLM)

- Data generating distribution for the response
- Link function (to the input variables)
- Linear predictor

Day 1
Session 3

Digging Deeper: Classification and Regression

1. Generalised linear regression
2. Spatial and time series models
3. Tree-based approaches: CART, RF, BRT, bagging boosting
4. Support vector machines

Spatial Modelling

1. Interpolation: Kriging

2. Local smoothing: BYM

BYM Model

Consider the distribution of observed counts within an area:

$y_i \sim \text{Poisson}(e_i\theta_i)$, θ_i = constant RR

Spatial Modelling

Consider the distribution of observed counts within an area:

$$y_i \sim \text{Poisson}(e_i\theta_i), \quad \theta_i = \text{constant RR}$$

Can add fixed effects
eg spatial trend or long-range variation over the study area:
fit to area centroids with x_1=easting, x_2=northing:

$$\theta_i = \exp(\beta_0+\beta_1 x_{1i}+\beta_2 x_{2i}) \text{ , ie } \theta_i = \exp(X_i\beta)$$

Spatial Modelling

Consider the distribution of observed counts within an area:

$y_i \sim \text{Poisson}(e_i\theta_i)$, θ_i = constant RR

Can add random effects

eg, variation in individual susceptibility (frailty); variation due to unmodelled covariates (overdispersion); error in interpolation of spatial covariate to locations of case events or area centroids; spatial autocorrelation

add to model or use Bayesian priors for parameters (eg θ_i)

BYM Spatial Model

Model with area level spatial random effect:

$$y_i \sim \text{Poisson}(\mu_i)$$
$$\mu_i = e_i \theta_i$$
$$\theta_i = \exp(\beta_0 + \nu_i + u_i)$$

$$\nu_i \sim N(\Sigma \nu_{\sim i}/n_i, \sigma_\nu^2)$$

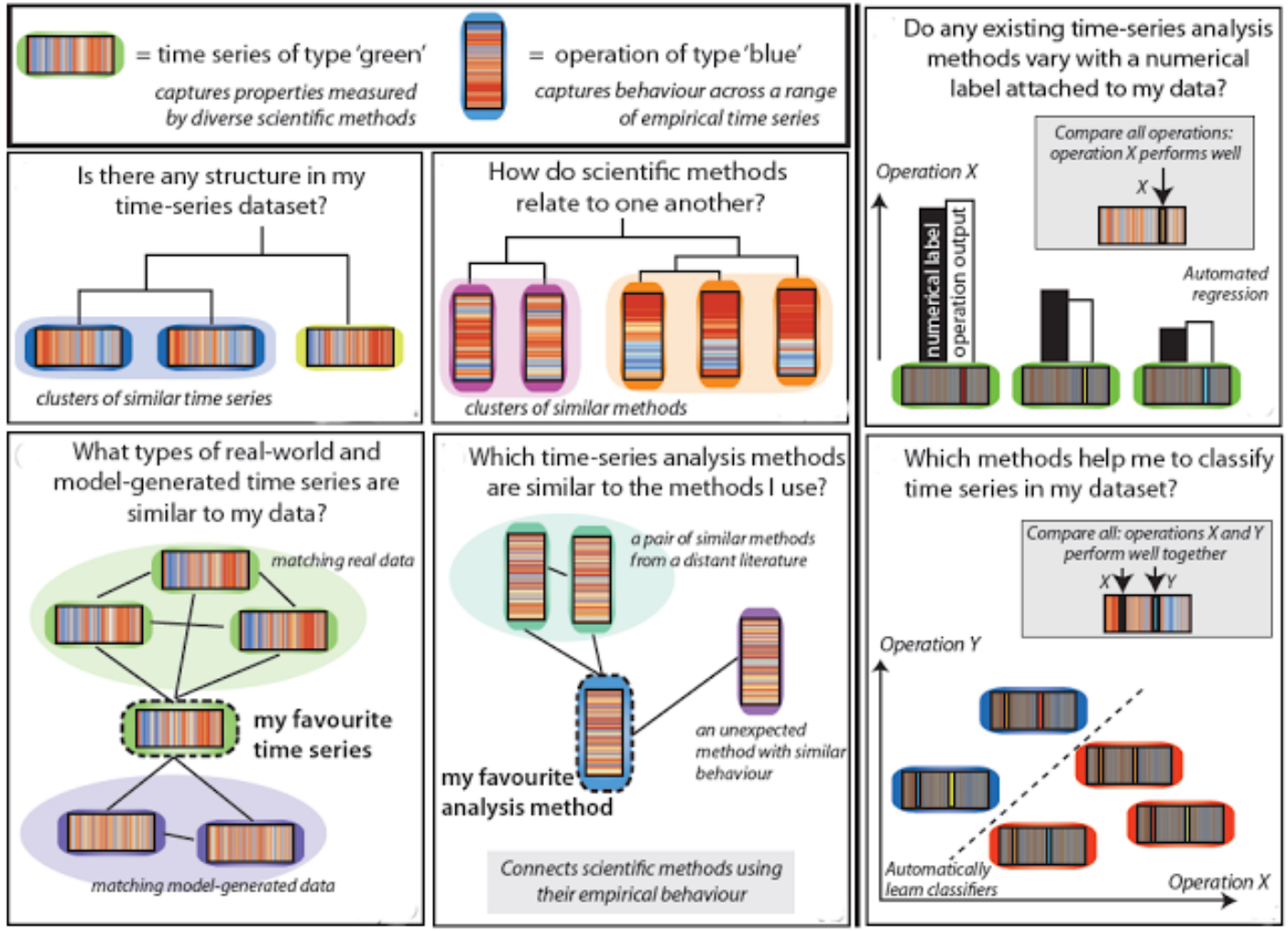
$$u_i \sim N(0, \sigma_u^2)$$

Time Series Modelling

1. Autoregressive, moving average and ARIMA models

2. State space models

Time series: taxonomy of aims and methods

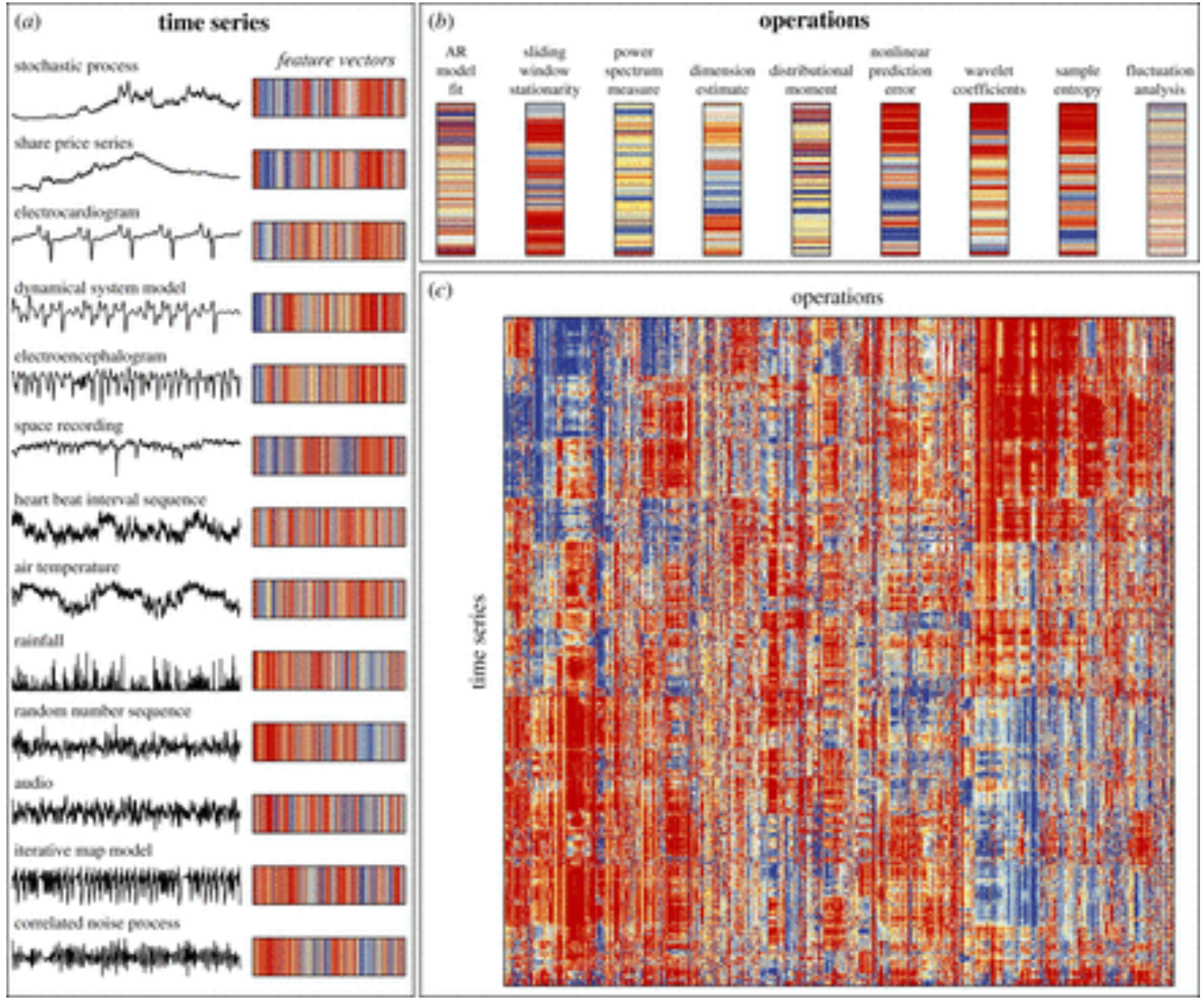


>9000 methods for analysing signals

>35 000 real-world & model-generated time series

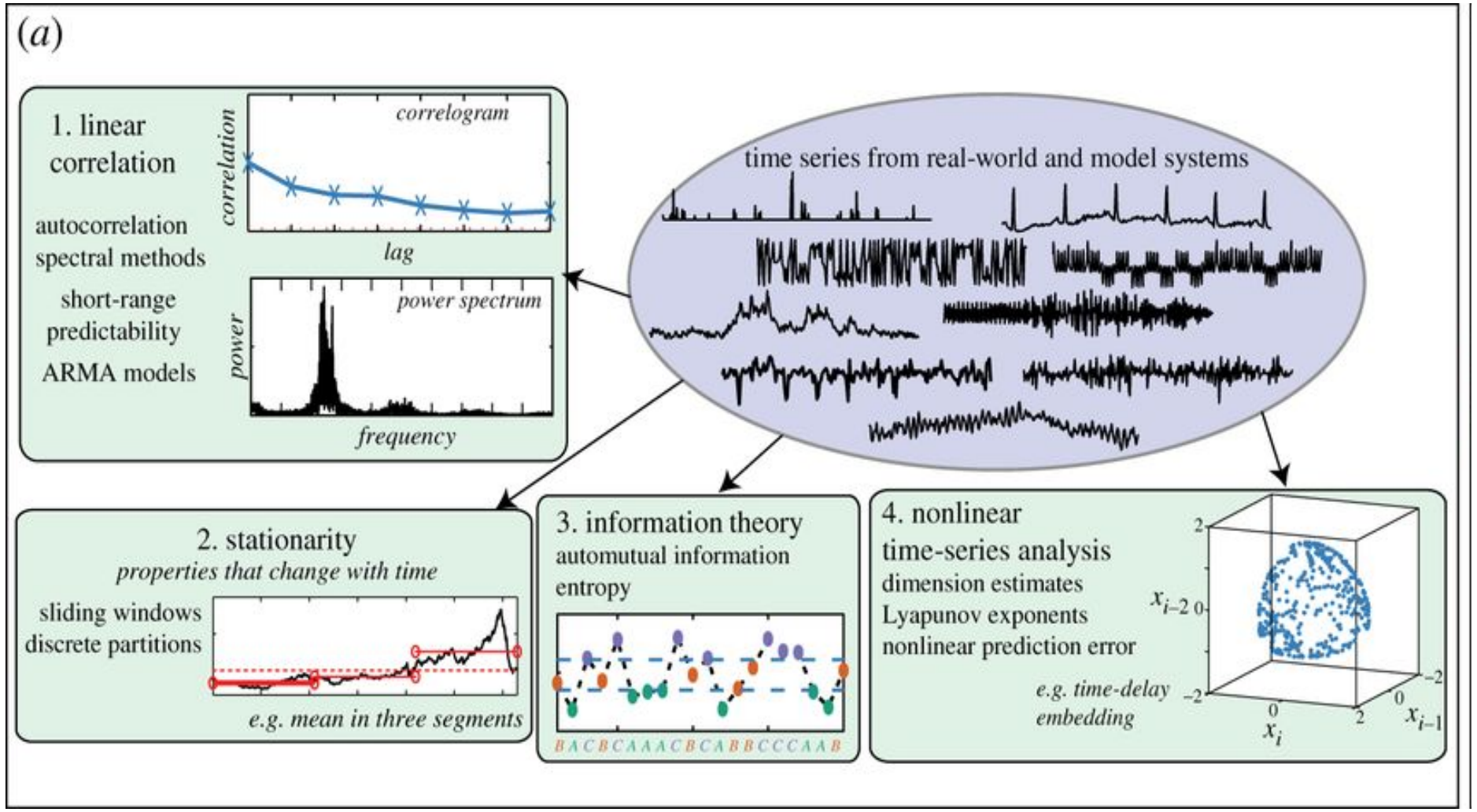
~ 200 types of methods

http://systems-signals.blogspot.com.au/2013/04/a-compound-methodological-eye-on.html



http://rsif.royalsocietypublishing.org/content/10/83/20130048

Cluster analysis of operations



http://rsif.royalsocietypublishing.org/content/10/83/20130048

Model based approaches: State space models

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 θ_t from time t to $t+1$,

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

Model based approaches: State space models

Write as a structural time series (trend, seasonality, error terms):

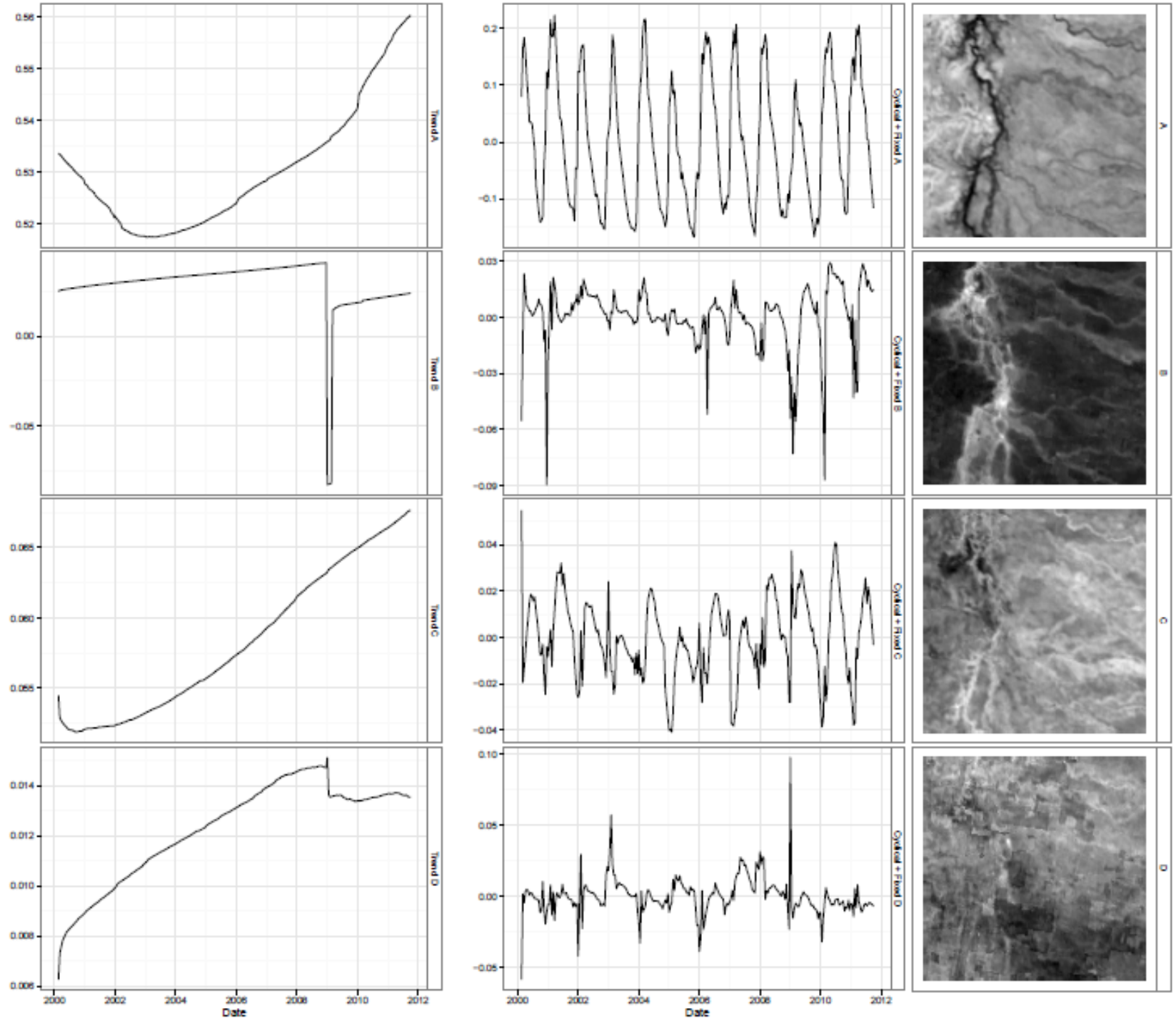
$$
\begin{aligned}
y_t &= \mu_t + \varepsilon_t, \quad & \varepsilon_t &\sim N(0, \sigma_\varepsilon^2) \\
\mu_{t+1} &= \mu_t + \nu_t + \xi_t, \quad & \xi_t &\sim N(0, \sigma_\xi^2) \\
\nu_{t+1} &= \nu_t + \varsigma_t, \quad & \varsigma_t &\sim N\left(0, \sigma_\xi^2\right)
\end{aligned}
$$

Model based approaches: State space models

Write as a state space model:

$$y_t = (1 \quad 0)\begin{pmatrix}\mu_t \\ \nu_t\end{pmatrix} + \varepsilon_t,$$

$$\begin{pmatrix}\mu_{t+1} \\ \nu_{t+1}\end{pmatrix} = \begin{bmatrix}1 & 1 \\ 0 & 1\end{bmatrix}\begin{pmatrix}\mu_t \\ \nu_t\end{pmatrix} + \begin{pmatrix}\xi_t \\ \varsigma_t\end{pmatrix}.$$


Trend A
Trend B
Trend C
Trend D
Cyclical + Fixed A
Cyclical + Fixed B
Cyclical + Fixed C
Cyclical + Fixed D
A
B
C
D
Date
Date

Time series regression

Gruber and West (2016): graphical dynamical linear models (SGDLMs) for forecasting and scalable multivariate volatility analysis

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

These involve:

(i) A set of decoupled univariate dynamic linear models for individual series

(ii) Sparse graphical modelling to recouple the series

(iii) (iii) variational Bayesian methods combined with importance sampling to integrate/couple the series for forecasting and decisions.

(iv) Parallel, GPU-based implementation enables on-line analysis of increasingly high-dimensional time series

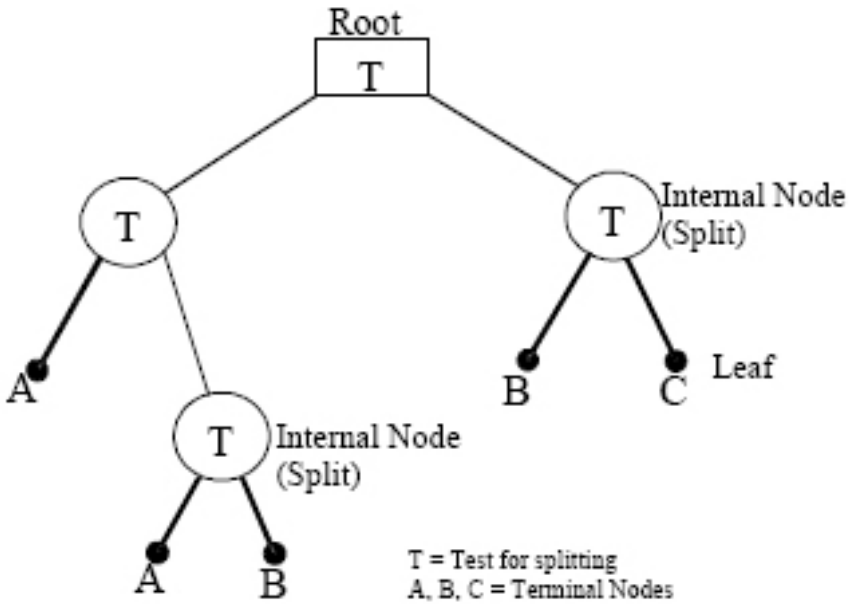
Day 1
Session 3

Digging Deeper: Classification and Regression

1. Generalised linear regression
2. Spatial and time series models
3. Tree-based approaches: CART, RF, BRT, bagging boosting
4. Support vector machines

CART

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



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).

CART

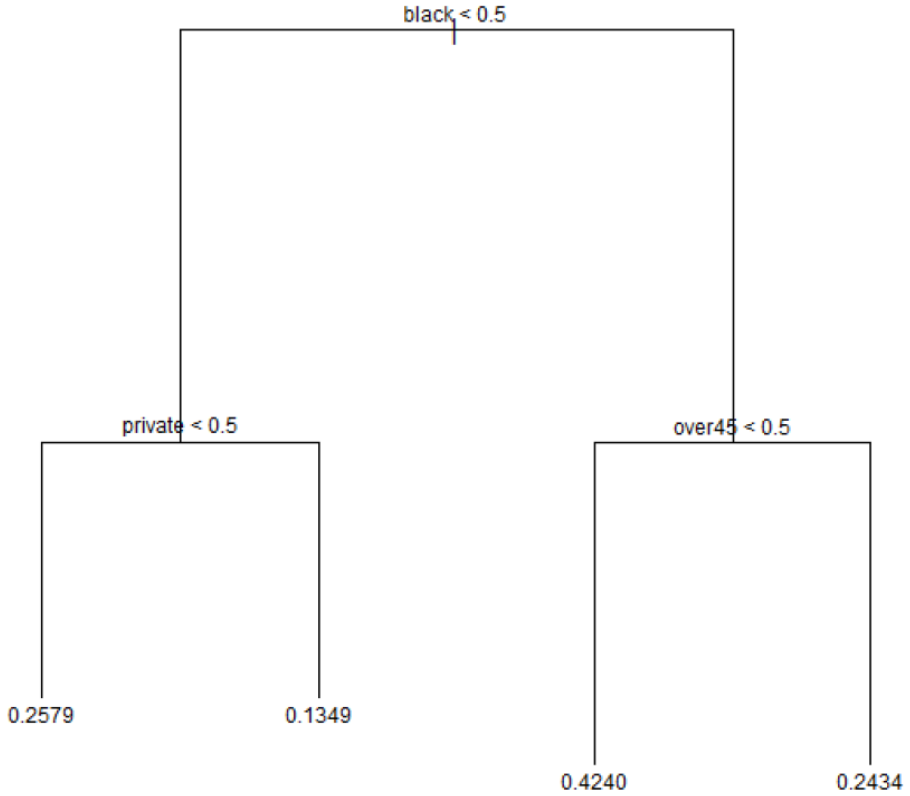
https://www.bu.edu/sph/files/2014/05/MorganCART.pdf

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).

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

Drawbacks of 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.

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.

Boosting

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

Boosting combines existing weak learners in a new form so the classification process achieves higher accuracy, and therefore less errors.

Weak learners often have just one feature and are simply structured, but can be interpreted well and quickly.

Boosting combines all weak classifiers with a weighting factor. Basically the residuals will be used to fit a new model to it.

Boosted Regression Trees (BRTs)

- Iteratively create new trees trained using the residual errors of the previous steps
- Observations which are predicted poorly are given a higher weight for the next step in creating a new tree.
- Repeat successively to generate each new set of residuals until a specified number of trees have been created.

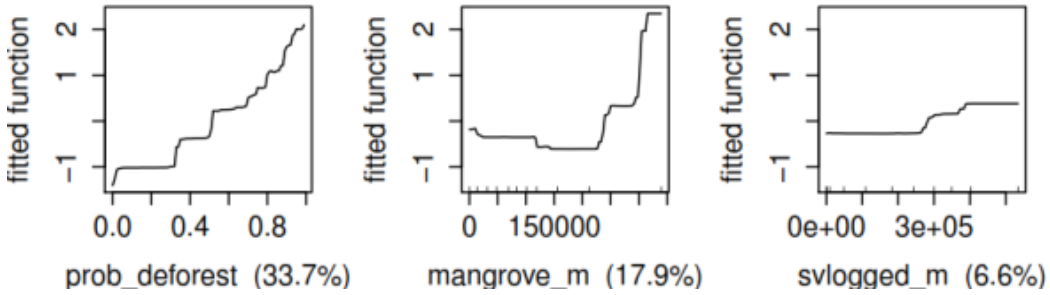
The result from the single steps in creating those trees is summed to give a successive accumulative model.

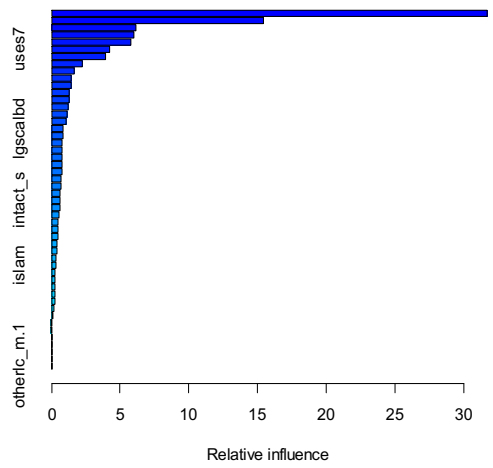
Shrinkage/learning rate: We can make smaller steps by only adding a fraction of the result given by each tree to achieve a more stable convergence towards the correct values.

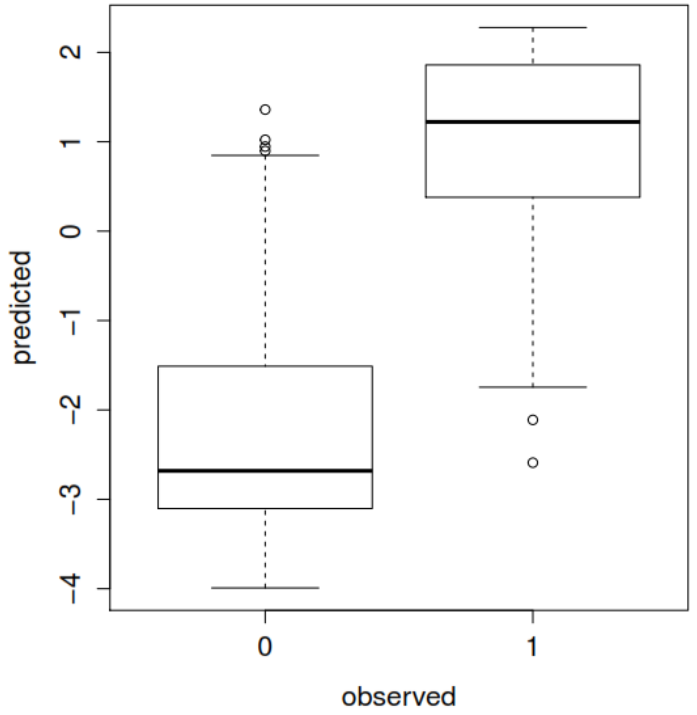
Example: orangutans!

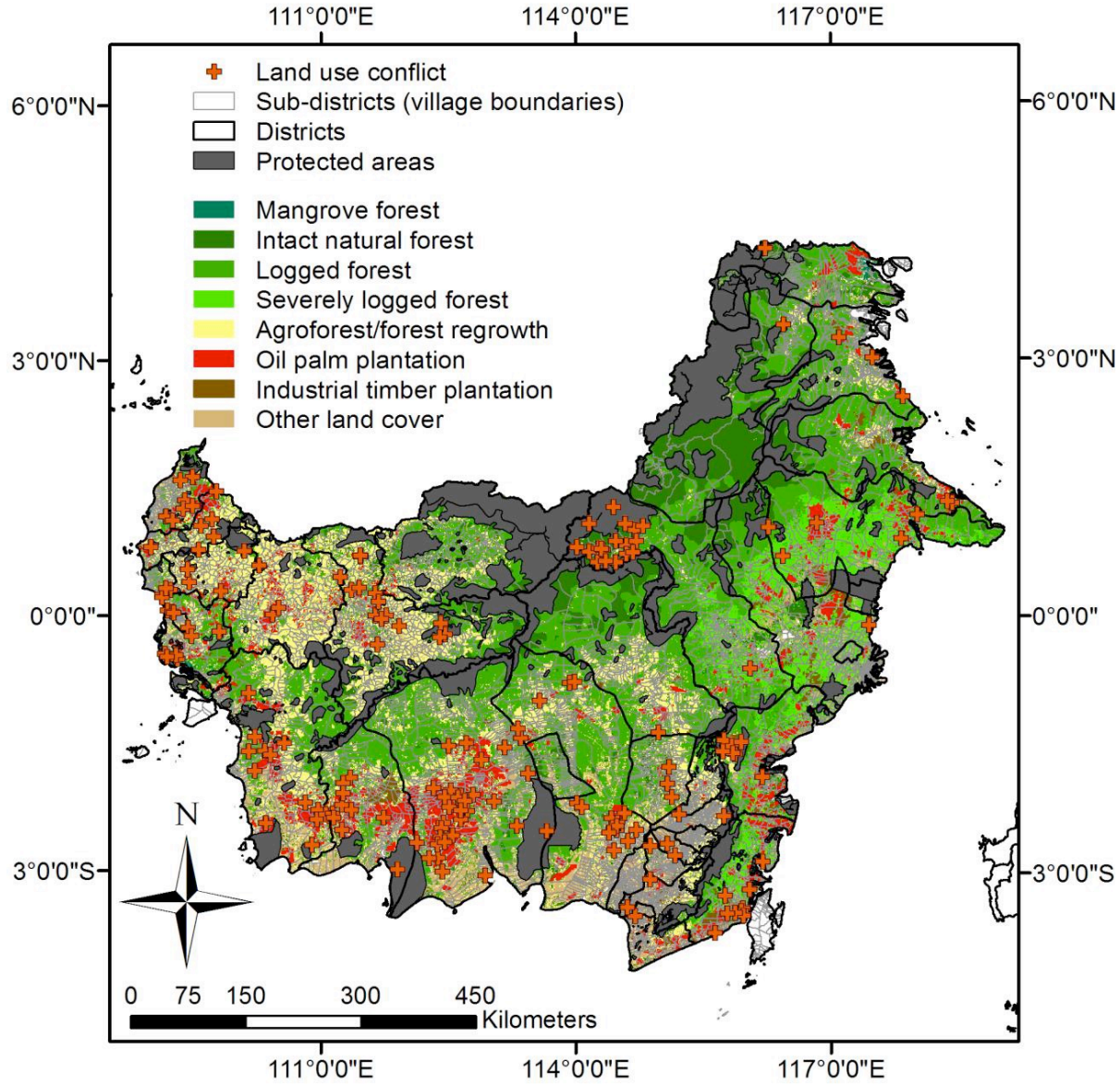
With Nicola Abram *et al.*

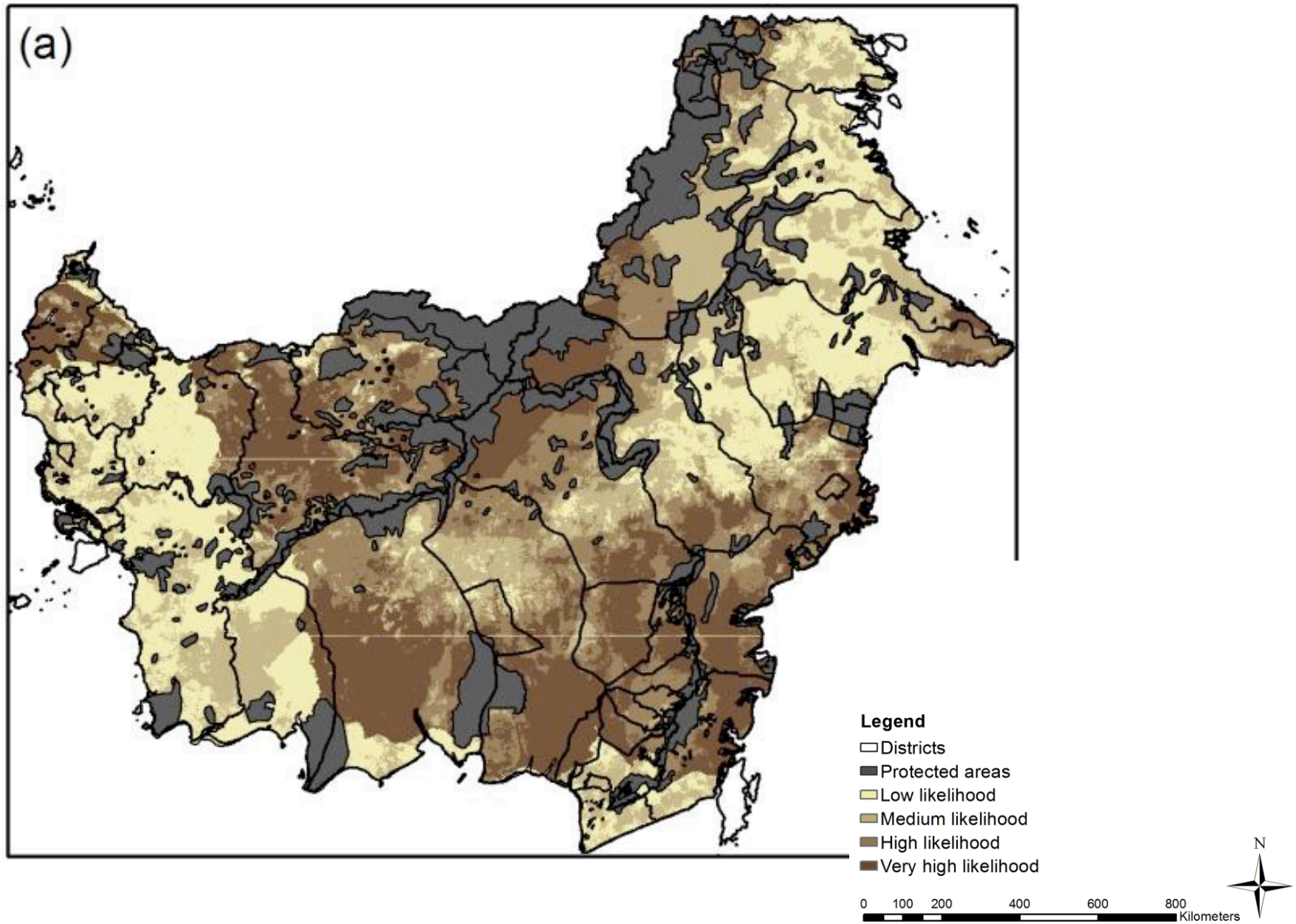
Understand conflict associated with oil palm








111°0'0"E
114°0'0"E
117°0'0"E
6°0'0"N
3°0'0"N
0°0'0"
3°0'0"S
Land use conflict
Sub-districts (village boundaries)
Districts
Protected areas
Mangrove forest
Intact natural forest
Logged forest
Severely logged forest
Agroforest/forest regrowth
Oil palm plantation
Industrial timber plantation
Other land cover
N
0 75 150 300 450
Kilometers


(a)
Legend
Districts
Protected areas
Low likelihood
Medium likelihood
High likelihood
Very high likelihood
N
0 100 200 400 600 800
Kilometers

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, $\sqrt{p}$ (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

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

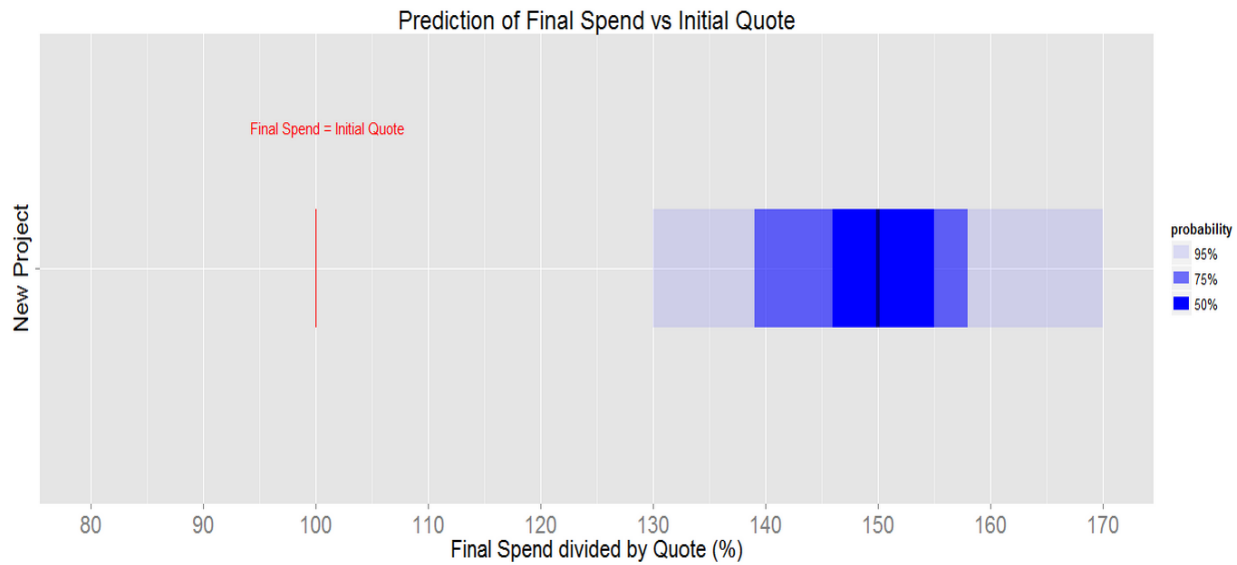
Client: Please start typing

Client size: individual

Client sector: Government

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

Calculate



Day 1
Session 3

Digging Deeper: Classification and Regression

1. Generalised linear regression
2. Spatial and time series models
3. Tree-based approaches: CART, RF, BRT, bagging boosting
4. Support vector machines

Support Vector Machines

Useful reference: Mountrakis, G., Im, J. and Ogole, C. (2011). Support vector machines in remote sensing: A review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(3), 247–259.

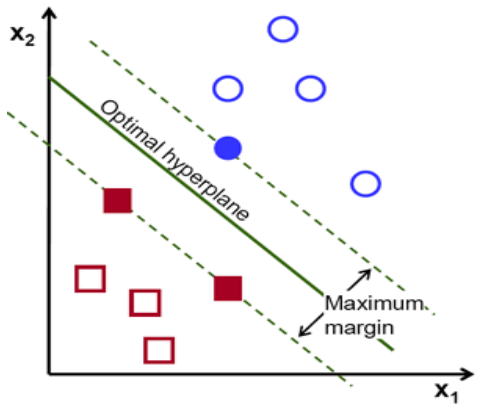
Support vector machines (SVMs) is a supervised non-parametric statistical learning technique, therefore there is no assumption made on the underlying data distribution.

In its simplest form, SVMs are linear binary classifiers that assign a given test sample a class from one of the two possible labels. An instance of a data sample to be labeled in the case of remote sensing classification is normally the individual pixel derived from the multi-spectral or hyperspectral image.

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)



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

Benefits of SVMs

- reasonably good results from relatively small amounts of training data
- Don’t“overfit”; that is, they tend to strike a balance between accurately differentiating groups in training data and generalising well to unobserved data
- Don’t rely on prohibitive assumptions regarding distribution of the data
- Don’t become trapped at local minima like neural networks

Disadvantages of SVMs

- A key issue with remote sensing and crop identification is choice of kernel e.g. polynomial or radial basis functions
- It is often not clear which kernel will yield best results
- Training data quality is also important with SVMs, as they are clearly sensitive to group mislabelling (however, this is an issue with all supervised classification techniques)
- SVMs are generally a binary classifier and so their use in multi-category applications generally takes the form of reducing the problem to a series of multiple binary comparisons.

References to SVM case studies for remote sensing

Szuster, B. W., Chen, Q. and Borger, M. (2011). A comparison of classification techniques to support land cover and land use analysis in tropical coastal zones. *Applied Geography*, 31(2), 525–532. This compares maximum likelihood classification and ANNs to SVMs for land use and land cover classification in tropical coastal zones.

Yang, C., Everitt, J. H. and Murden, D. (2011). Evaluating high resolution SPOT 5 satellite imagery for crop identification. *Computers and Electronics in Agriculture*, 75(2), 347–354. Assess various supervised classifiers using SPOT 5 imagery (a high resolution satellite). They found that the maximum likelihood and support vector machine techniques performed best of those considered. Increased pixel size from $10m$ to $30m$ had minimal effect on classification accuracy.

References to SVM case studies for remote sensing

Melgani, F. and Bruzzone, L. (2004). Classification of hyperspectral remote sensing images with support vector machines. *IEEE Transactions on Geoscience and Remote Sensing*, 42(8), 1778–1790. Considers SVMs for remotely sensed data, comparing their use to that of neural networks (radial basis function NNs) and K-nearest neighbour (K-NN). Conclude SVMs are a viable option.

Huang, C., Davis, L. S. and Townshend, J. R. G. (2002). An assessment of support vector machines for land cover classification. *International Journal of Remote Sensing*, 23(4), 725–749. Investigate SVMs for land cover classification and compare to neural networks, maximum likelihood, and decision trees.

Day 1
Session 4

Extended Topics

1. Semiparametric regression

2. KNN

3. Ensembles and XGBoost

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 σ_ϵ^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
  - $U_1, \ldots, U_{992} | \sigma_U^2 \sim N(0, \sigma_U 2)$ are random intercepts for the rental agency
  - house_{ij} is an indicator for the property being a house or apartment
  - longitude_{ij} and 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

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.

Ensemble modelling

“Ensemble modeling is the process of running two or more related but different analytical models and then synthesizing the results in order to improve the accuracy of predictive analytics and data mining applications.”

“It is being said that ensemble modeling offers one of the most convincing way to build highly accurate predictive models. The availability of bagging and boosting algorithms further embellishes this method to produce awesome accuracy level.”

http://searchbusinessanalytics.techtarget.com/definition/Ensemble-modeling
https://www.analyticsvidhya.com/blog/2015/09/questions-ensemble-modeling/

5 questions about ensemble modelling

1. What is an ensemble model?
2. What are bagging, boosting and stacking?
3. Can we ensemble multiple models of same ML algorithm?
4. How can we identify the weights of different models?
5. What are the benefits of ensemble model?

https://www.analyticsvidhya.com/blog/2015/09/questions-ensemble-modeling/

1. What is an ensemble model?

Four reasons for a prediction – or inference – to be different:

- Different population
- Different hypothesis
- Different modelling technique
- Different initial seed

https://www.analyticsvidhya.com/blog/2015/08/introduction-ensemble-learning/

Example

Problem: Set rules for classification of spam emails
Solution: We can generate various rules for classification of spam emails:

- Spam
  - Have total length less than 20 words
  - Have only image (promotional images)
  - Have specific key words like "make money and grow" and "reduce your fat"
  - More miss spelled words in the email
- Not Spam
  - Email from Analytics Vidhya domain
  - Email from family members or anyone from e-mail address book

Can all these rules individually predict the correct class?
Combining these rules would provide more robust prediction.
This is the principle of Ensemble Modeling.
Ensemble model combines multiple 'individual' (diverse) models together and delivers superior prediction power.

https://www.analyticsvidhya.com/blog/2015/09/questions-ensemble-modeling/

Error in ensemble learning

$$Err(x) = \left(E[\hat{f}\,(x)] - f(x)\right)^2 + E\left[\hat{f}\,(x) - E[\hat{f}\,(x)]\right]^2 + \sigma_e^2$$

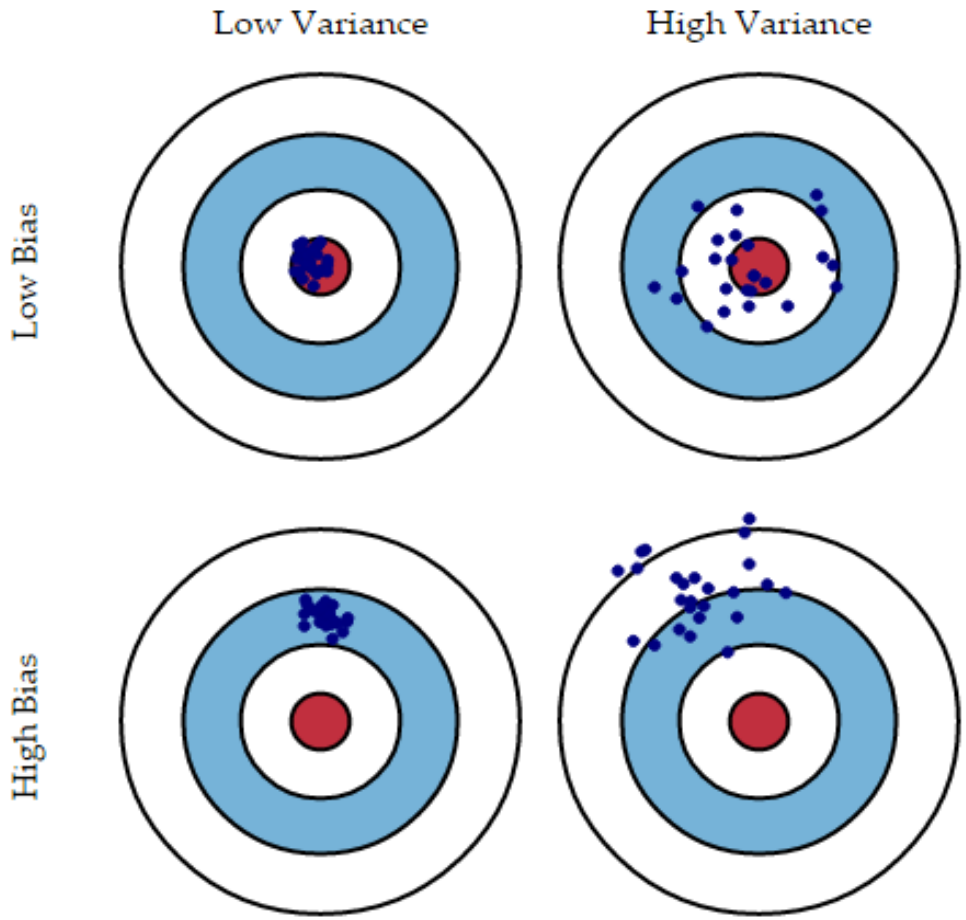
$$Err(x) = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error}$$

Bias error quantifies how much on an average the predicted values are different from the actual value.
High bias error = under-performing model which will miss important trends.

Variance quantifies how the prediction made on same observation differ from each other.
High variance = over-fitting on the training population and poor performance on any observation beyond training.

https://www.analyticsvidhya.com/blog/2015/09/questions-ensemble-modeling/

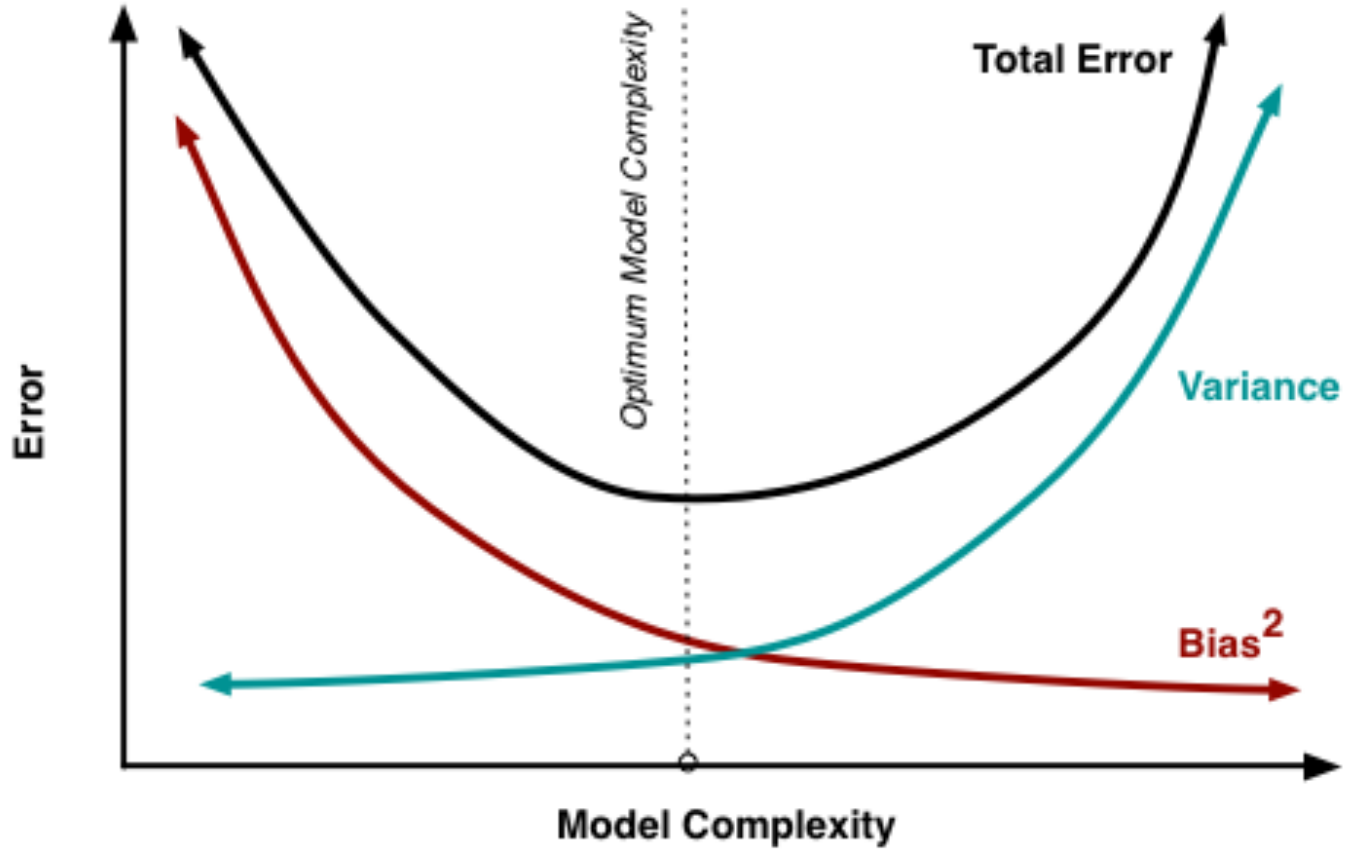
Bias versus variance



Red spot is the real value
Blue dots are predictions

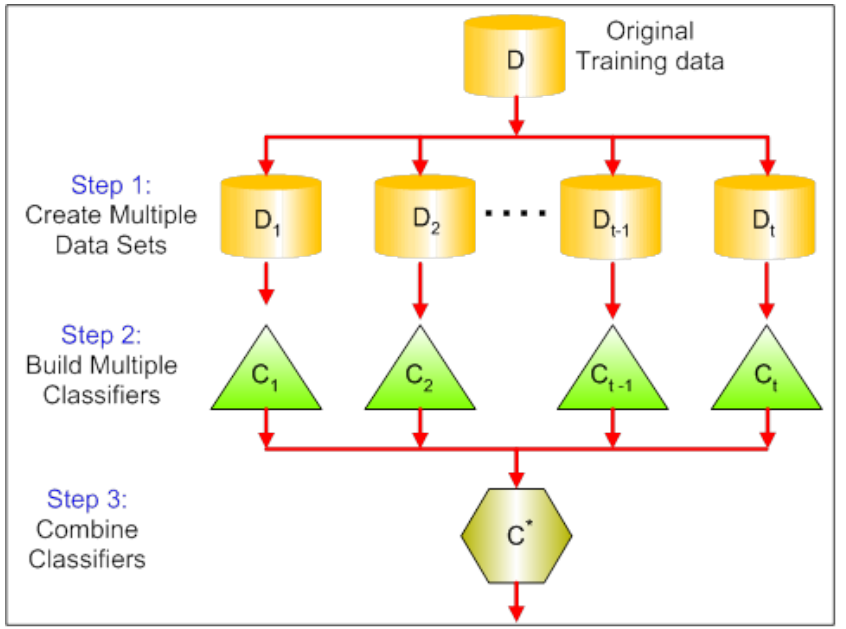
https://www.analyticsvidhya.com/blog/2015/09/questions-ensemble-modeling/

Trade-off management of bias and variance



https://www.analyticsvidhya.com/blog/2015/09/questions-ensemble-modeling/

2. What is bagging?



Bagging (Bootstrap Aggregating) is an ensemble method. First, we create random samples of the training data set (sub sets of training data set).

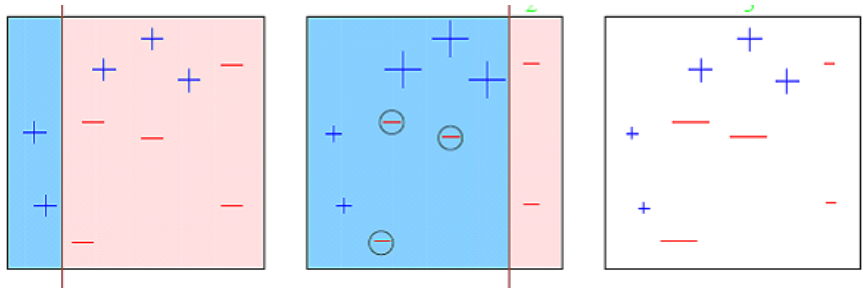
Then, we build a classifier for each sample.

Finally, results of these multiple classifiers are combined using average or majority voting.

Bagging helps to reduce the variance error.

https://www.analyticsvidhya.com/blog/2015/09/questions-ensemble-modeling/

2. What is boosting?



Boosting provides sequential learning of the predictors. The first predictor is learned on the whole data set, while the following are learnt on the training set based on the performance of the previous one**.**

It starts by classifying original data set and giving equal weights to each observation.

If classes are predicted incorrectly using the first learner, then it gives higher weight to the missed classified observation.

Being an iterative process, it continues to add classifier learner until a limit is reached in the number of models or accuracy.

Boosting has shown better predictive accuracy than bagging, but it also tends to over-fit the training data as well.

https://www.analyticsvidhya.com/blog/2015/09/questions-ensemble-modeling/

Boosting in practice

The most common example of boosting is AdaBoost and Gradient Boosting.

Getting smart with Machine Learning – AdaBoost and Gradient Boost

https://www.analyticsvidhya.com/blog/2015/05/boosting-algorithms-simplified/

Learn Gradient Boosting Algorithm for better predictions (with codes in R)

https://www.analyticsvidhya.com/blog/2015/09/complete-guide-boosting-methods/

https://www.analyticsvidhya.com/blog/2015/09/questions-ensemble-modeling/

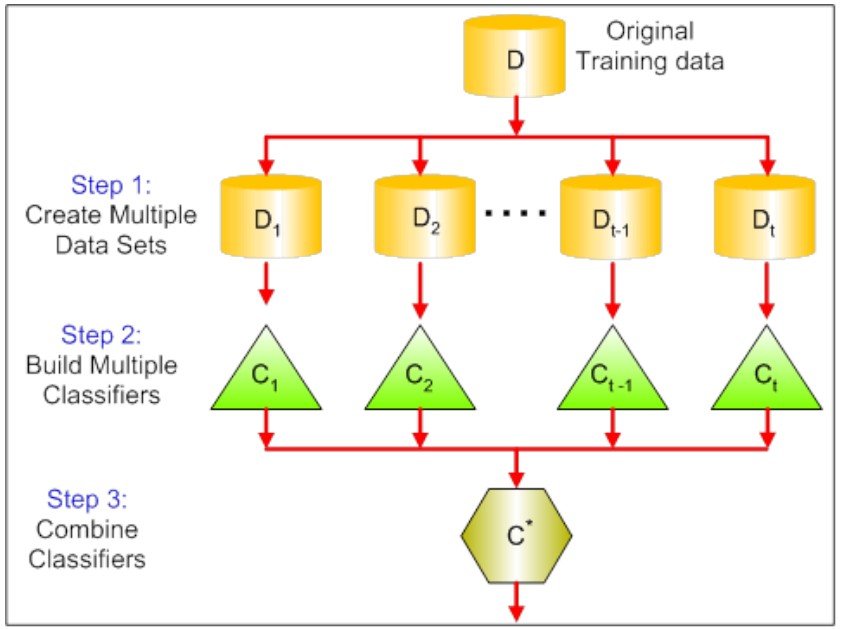
XGBoost

- Scalable, Portable and Distributed Gradient Boosting (GBDT, GBRT or GBM) Library
- Optimised, distributed gradient boosting library, designed to be highly efficient, flexible and portable.
- For Python, R, Java, Scala, C++ and more.
- Runs on single machine, Hadoop, Spark, Flink and DataFlow
- Provides a parallel tree boosting.

“Can solve problems beyond billions of examples”

https://github.com/dmlc/xgboost

2. What is stacking?



Bagging (Bootstrap Aggregating) is an ensemble method.
First, we create random samples of the training data set
(sub sets of training data set).
Then, we build a classifier for each sample.
Finally, results of these multiple classifiers are combined using average or majority voting.

Bagging helps to reduce the variance error.

https://www.analyticsvidhya.com/blog/2015/09/questions-ensemble-modeling/

3. Can we ensemble multiple models of the same ML algorithm?

Yes, but combining multiple predictions generated by different algorithms would normally give better predictions.

Example: the predictions of a random forest, a KNN, and a Naive Bayes may be combined to create a stronger final prediction set, compared to combining three random forest models.

The key to creating a powerful ensemble is model diversity. An ensemble with two techniques that are very similar in nature will perform poorly than a more diverse model set.

Exercise: Three models (A, B and C). A, B and C have prediction accuracy of 85%, 80% and 55% respectively. But A and B are found to be highly correlated whereas C is meagerly correlated with both A and B. Which models would you combine?

https://www.analyticsvidhya.com/blog/2015/09/questions-ensemble-modeling/

4. How can we identify the weights of different ensemble models?

In general, we assume equal weight for all models and takes the average of predictions.

Alternatives:

- Find the collinearity between base learners and based on this table, then identify the base models to ensemble. After that look at the cross validation score (ratio of score) of identified base models to find the weight.
- Find the algorithm to return the optimal weight for base learners using neural networks.
- Use other methods such as forward selection of learners, selection with replacement, bagging.

https://www.analyticsvidhya.com/blog/2015/09/questions-ensemble-modeling/

5. What are the benefits of an ensemble model?

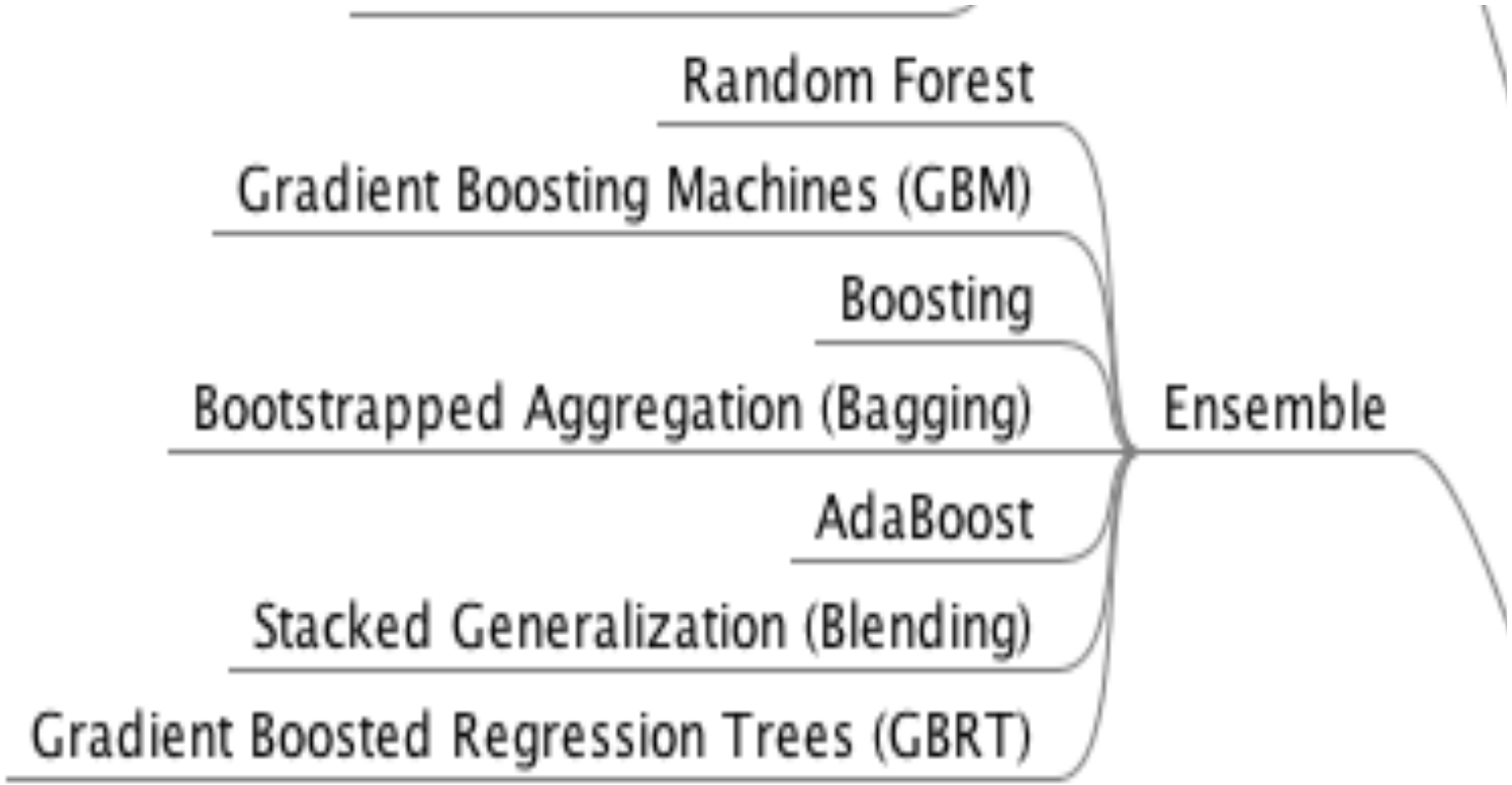
- Better prediction
- More stable model

The aggregate opinion of a multiple models is less noisy than other models.

In finance, this is called “Diversification”: a mixed portfolio of many stocks will be much less variable than just one of the stocks alone.

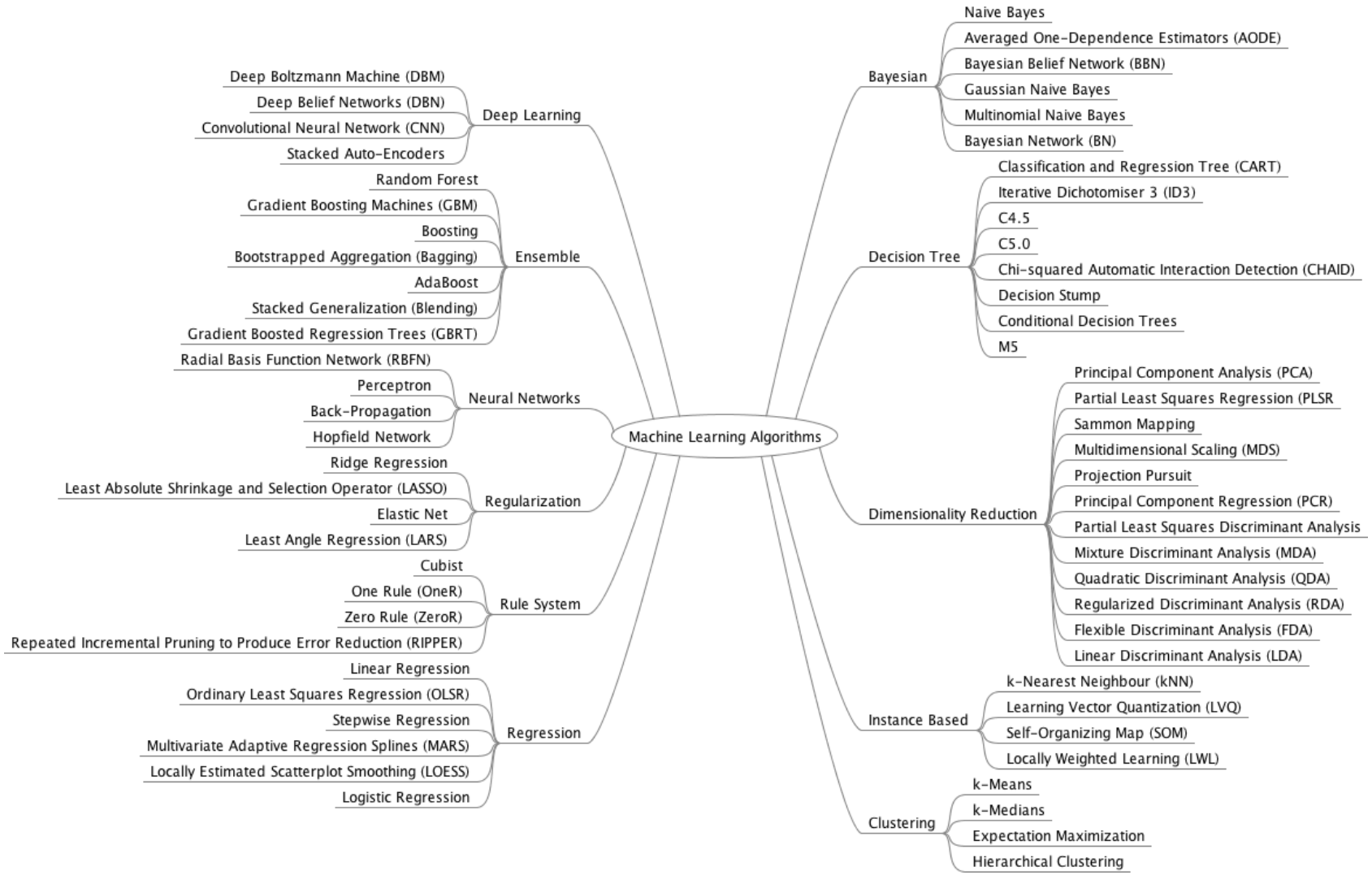
https://www.analyticsvidhya.com/blog/2015/09/questions-ensemble-modeling/

Ensemble models



https://s3.amazonaws.com/MLMastery/MachineLearningAlgorithms.png?__s=17oupmsuthrzaaubgepa

In context



https://s3.amazonaws.com/MLMastery/MachineLearningAlgorithms.png?__s=17oupmsuthrzaaubgepa

Day 1
Session 4

Concluding Discussion