DNYANSAGAR ARTS AND COMMERCE COLLEGE, BALEWADI,PUNE – 45

SUBJECT : BIGDATA

UNIT 1: Introduction to Big Data

What is Big Data?

What makes data, "Big" Data?

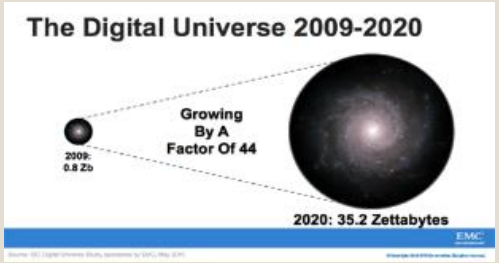
Big Data Definition

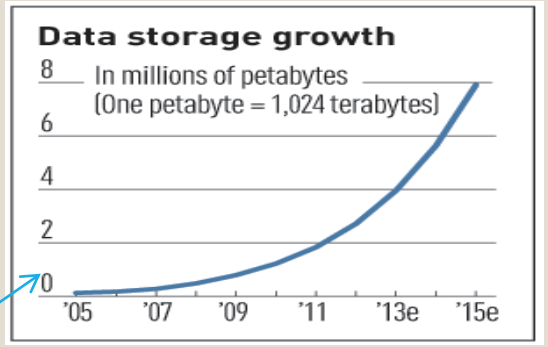
- No single standard definition…

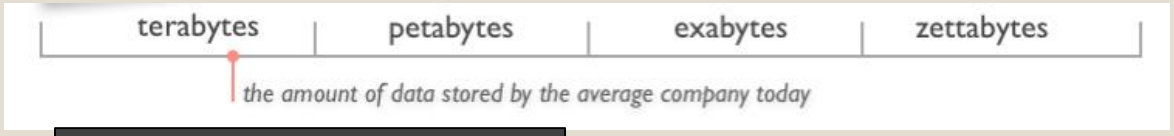
"***Big Data***" is data whose scale, diversity, and complexity require new architecture, techniques, algorithms, and analytics to manage it and extract value and hidden knowledge from it…

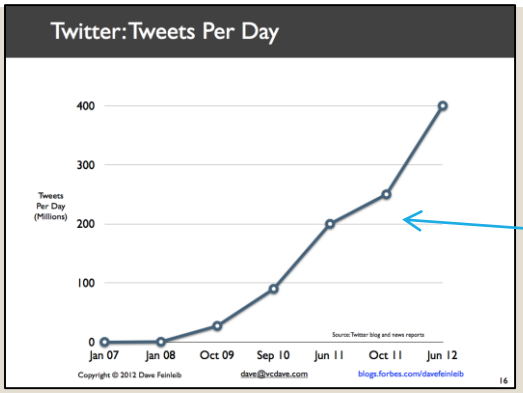
Characteristics of Big Data: 1-Scale (Volume)

- **Data Volume**
  - 44x increase from 2009 2020
  - From 0.8 zettabytes to 35zb
- Data volume is increasing exponentially





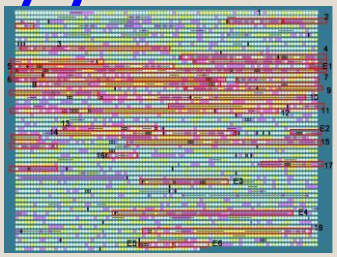


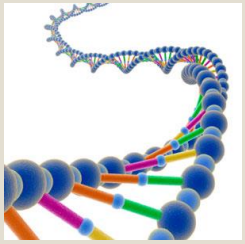


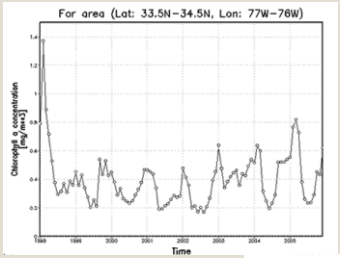
Exponential increase in collected/generated data

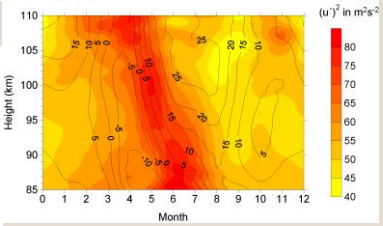
Characteristics of Big Data: 2-Complexity (Varity)

- Various formats, types, and structures
- Text, numerical, images, audio, video, sequences, time series, social media data, multi-dim arrays, etc…
- Static data vs. streaming data
- A single application can be generating/collecting many types of data







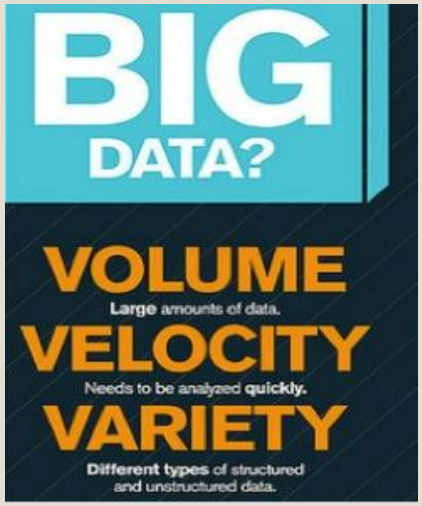


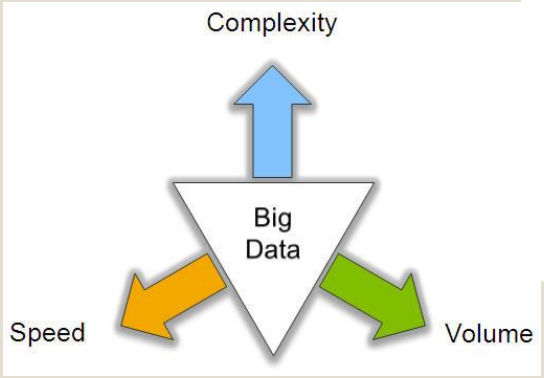
To extract knowledge➔ all these types of data need to linked together

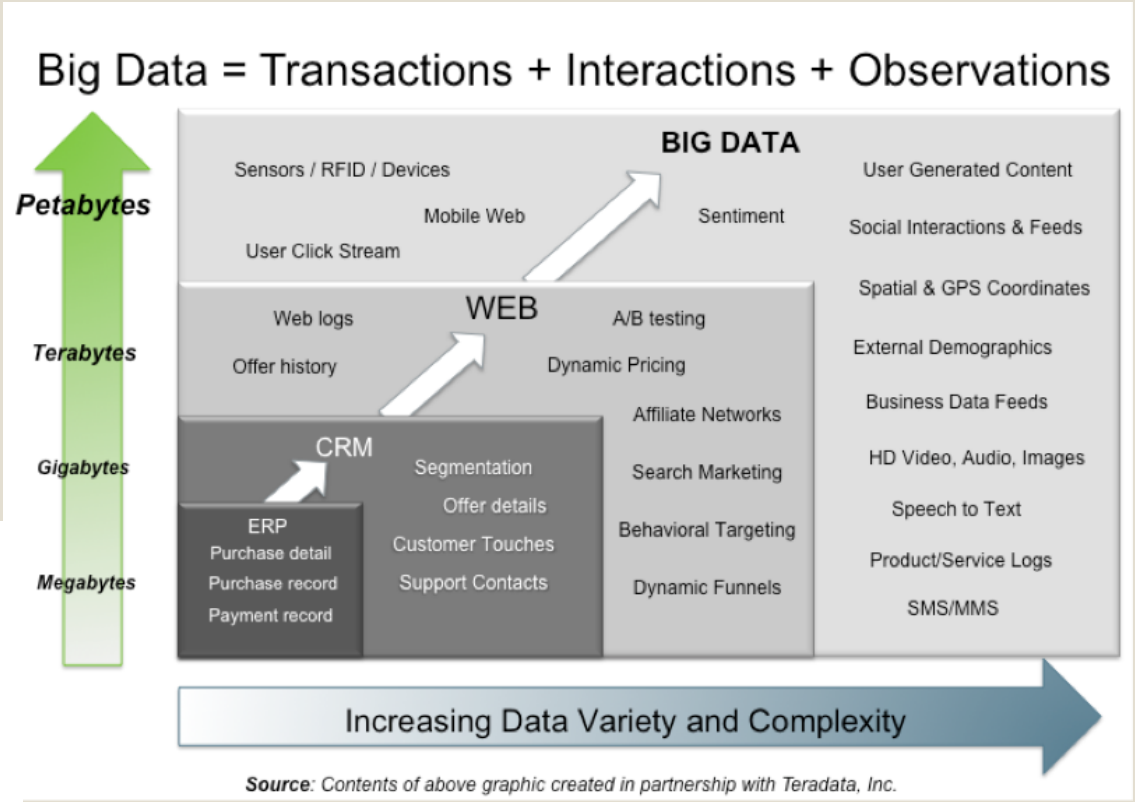
Characteristics of Big Data: 3-Speed (Velocity)

- Data is begin generated fast and need to be processed fast
- Online Data Analytics
- Late decisions ➔ missing opportunities
- **Examples**
  - **E-Promotions:** Based on your current location, your purchase history, what you like ➔ send promotions right now for store next to you
  - **Healthcare monitoring:** sensors monitoring your activities and body ➔ any abnormal measurements require immediate reaction

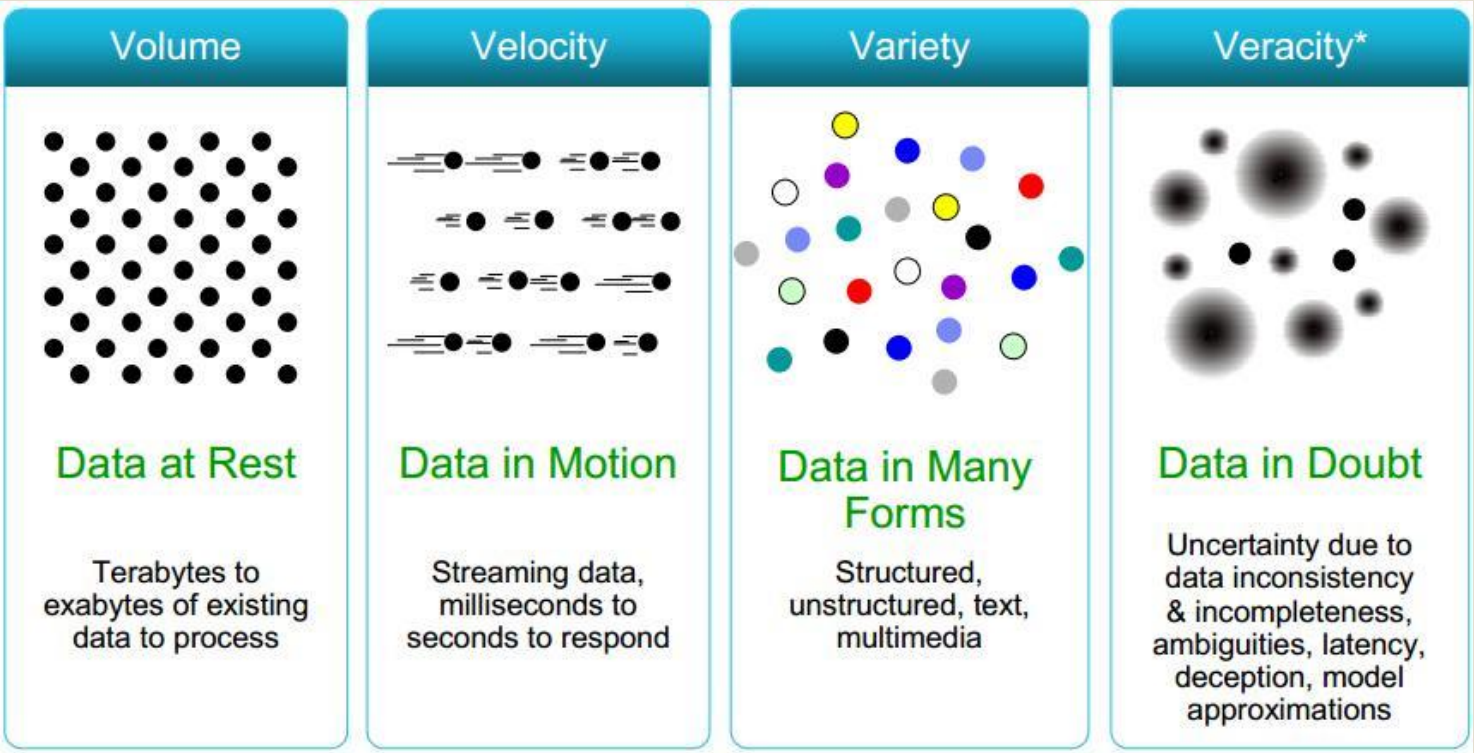
Big Data: 3V's


BIG
DATA?
VOLUME
Large amounts of data.
VELOCITY
Needs to be analyzed quickly.
VARIETY
Different types of structured and unstructured data.

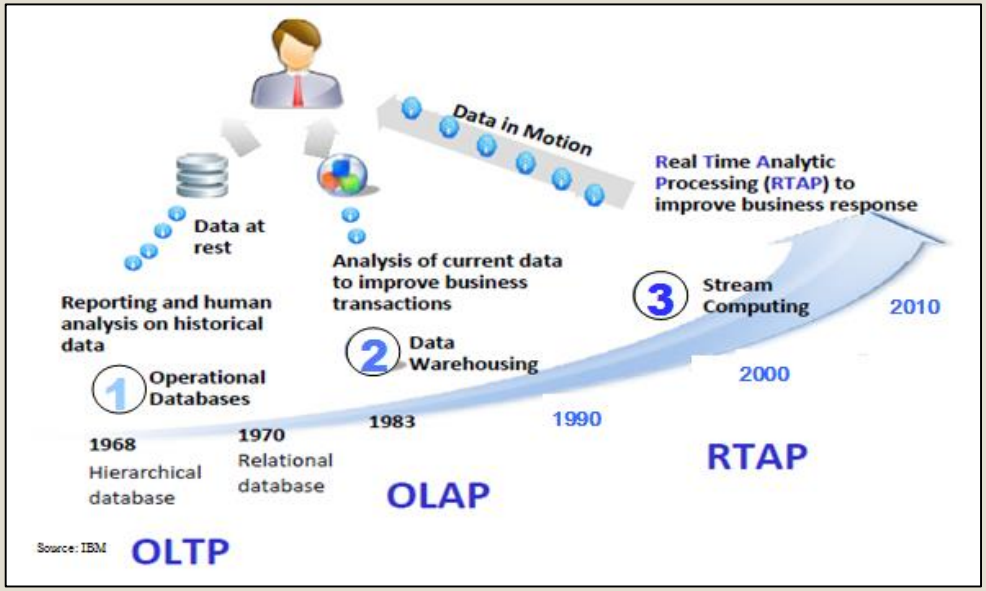

Complexity
Big Data
Speed
Volume


Big Data = Transactions + Interactions + Observations
Petabytes
Terabytes
Gigabytes
Megabytes
BIG DATA
Sensors / RFID / Devices
Mobile Web
User Click Stream
Sentiment
User Generated Content
Social Interactions & Feeds
Spatial & GPS Coordinates
External Demographics
Business Data Feeds
HD Video, Audio, Images
Speech to Text
Product/Service Logs
SMS/MMS
WEB
Web logs
Offer history
A/B testing
Dynamic Pricing
Affiliate Networks
Search Marketing
Behavioral Targeting
Dynamic Funnels
CRM
Segmentation
Offer details
Customer Touches
Support Contacts
ERP
Purchase detail
Purchase record
Payment record
Increasing Data Variety and Complexity
Source: Contents of above graphic created in partnership with Teradata, Inc.

Some Make it 4V's



Harnessing Big Data



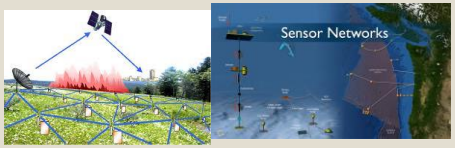
- **OLTP:** Online Transaction Processing (DBMSs)
- **OLAP:** Online Analytical Processing (Data Warehousing)
- **RTAP:** Real-Time Analytics Processing (Big Data Architecture & technology)

Who's Generating Big Data

Social media and networks
(all of us are generating data)

Scientific instruments
(collecting all sorts of data)

Mobile devices
(tracking all objects all the tim



Sensor technology and networks
(measuring all kinds of data)

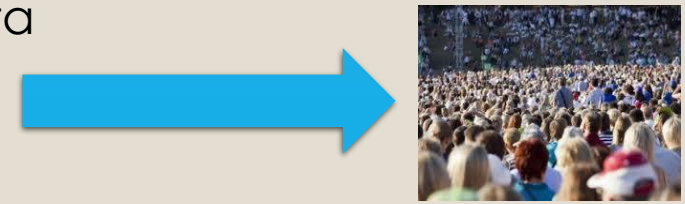
- The progress and innovation is no longer hindered by the ability to collect data
- But, by the ability to manage, analyze, summarize, visualize, and discover knowledge from the collected data in a timely manner and in a scalable fashion

The Model Has Changed…

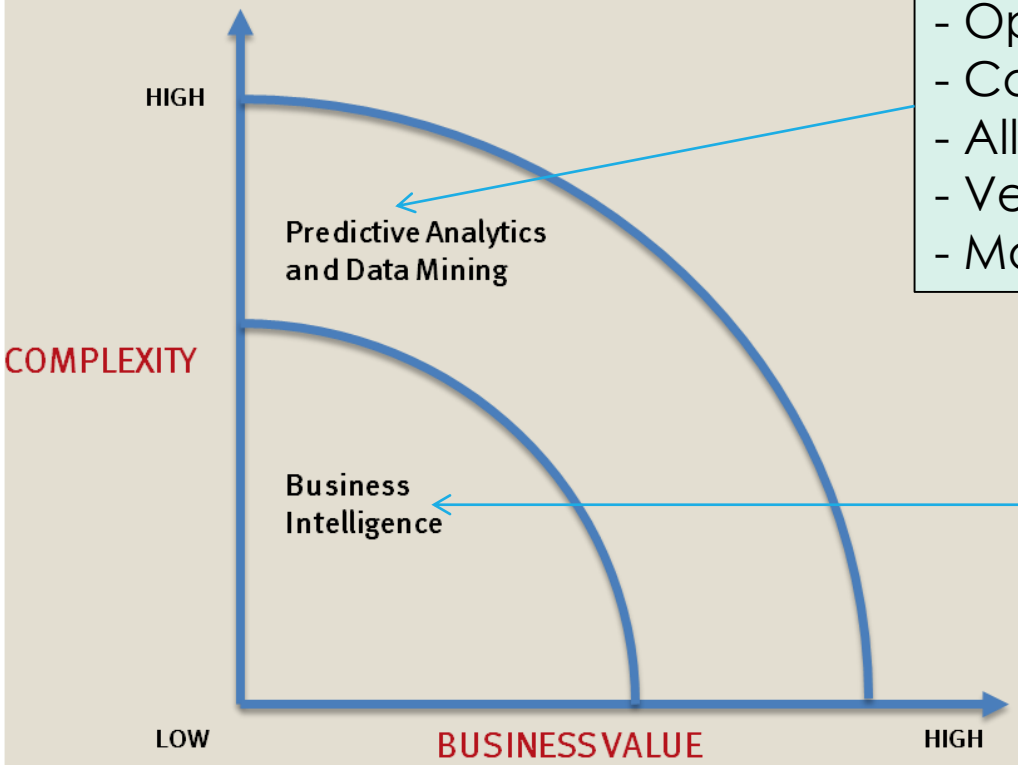
- **The Model of Generating/Consuming Data has Changed**

Old Model: Few companies are generating data, all others are consuming dat


New Model: all of us are generating data, and all of us are consuming data

What's driving Big Data

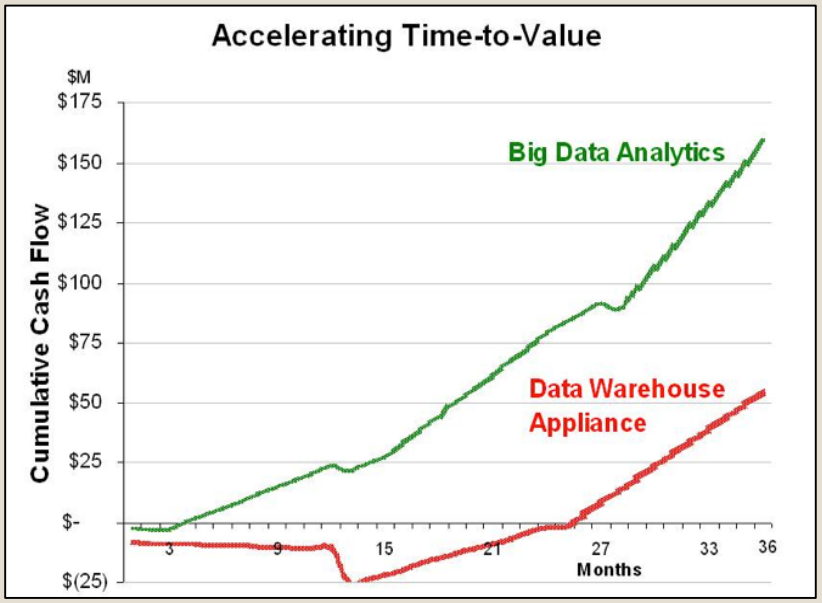


- Optimizations and predictive analytics
- Complex statistical analysis
- All types of data, and many sources
- Very large datasets
- More of a real-time

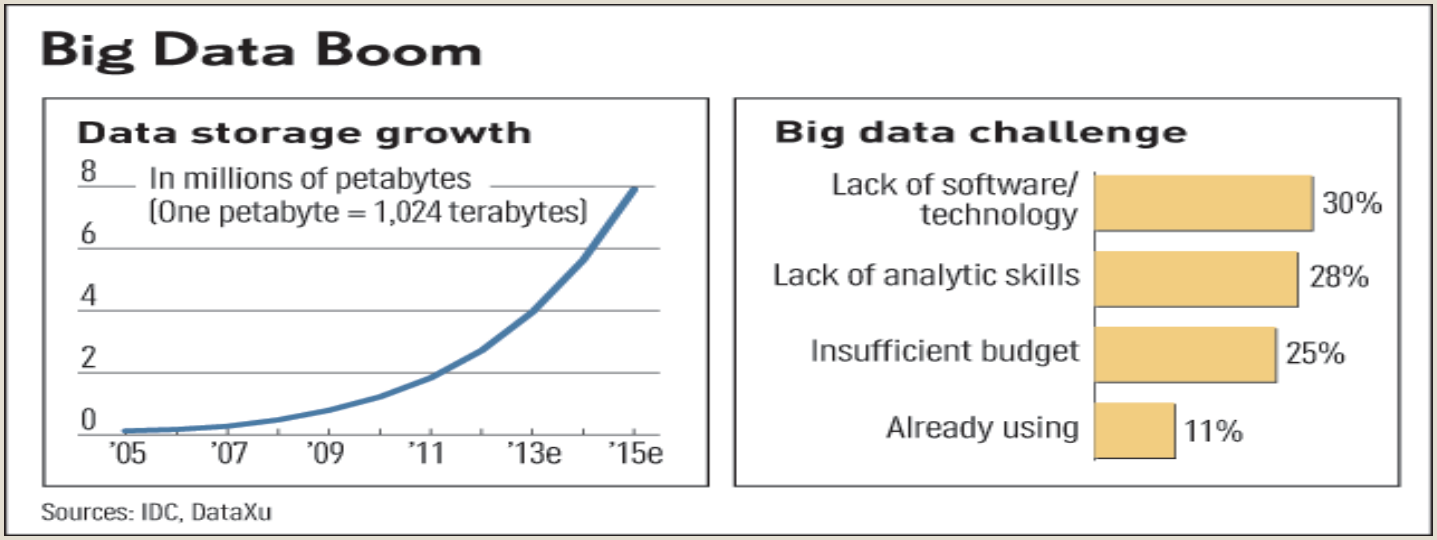
- Ad-hoc querying and reporting
- Data mining techniques
- Structured data, typical sources
- Small to mid-size datasets

Value of Big Data Analytics

- Big data is more real-time in nature than traditional DW applications
- Traditional DW architectures (e.g. Exadata, Teradata) are not well-suited for big data apps
- Shared nothing, massively parallel processing, scale out architectures are well-suited for big data apps



Challenges in Handling Big Data



- **The Bottleneck is in technology**
  - New architecture, algorithms, techniques are needed
- **Also in technical skills**
  - Experts in using the new technology and dealing with big data

What Technology Do We Have For Big Data ??

Big Data Landscape

Vertical Apps

PREDICTIVE POLICING
bloomreach
MYRRIX

Log Data Apps

splunk> loggly sumologic

Ad/Media Apps

rocketfuel
collective[i]
bluefin
Recorded Future
LuckySort
Media Science
TURN
DataXu

Business Intelligence

ORACLE | Hyperion
SAP Business Objects RJMetrics
Microsoft | Business Intelligence
IBM COGNOS birst
MicroStrategy
Autonomy
bime
QlikView
Chart.io
GoodData

Analytics and Visualization

tableau
Palantir
OPERA
metaLayer
METAMARKETS
dataspora
centrifuge
TERADATA ASTER
SAS TIBCO
KARMASPHERE
panopticon
Real-Time Visual Data Analysis
pentaho
Datameer
platfora
ClearStory
CIRRO
alteryx
visual.ly
AYATA

Data As A Service

factual
kaggle
knoema
GNIP
DATASIFT
Windows Azure Marketplace
INRIX
LexisNexis
LOQATE

Analytics Infrastructure

Hortonworks
VERTICA
MAPR
cloudera
INFOBRIGHT
PARACCEL
EMC²
GREENPLUM
NETEZZA
kognitio
DATASTAX
EXASOL
calpont

Operational Infrastructure

COUCHBASE
10gen
TERADATA
HADAPT
TERRACOTTA
VoltDB
MarkLogic
INFORMATICA

Infrastructure As A Service

amazon web services
Windows Azure
infochimps
Google BigQuery

Structured Databases

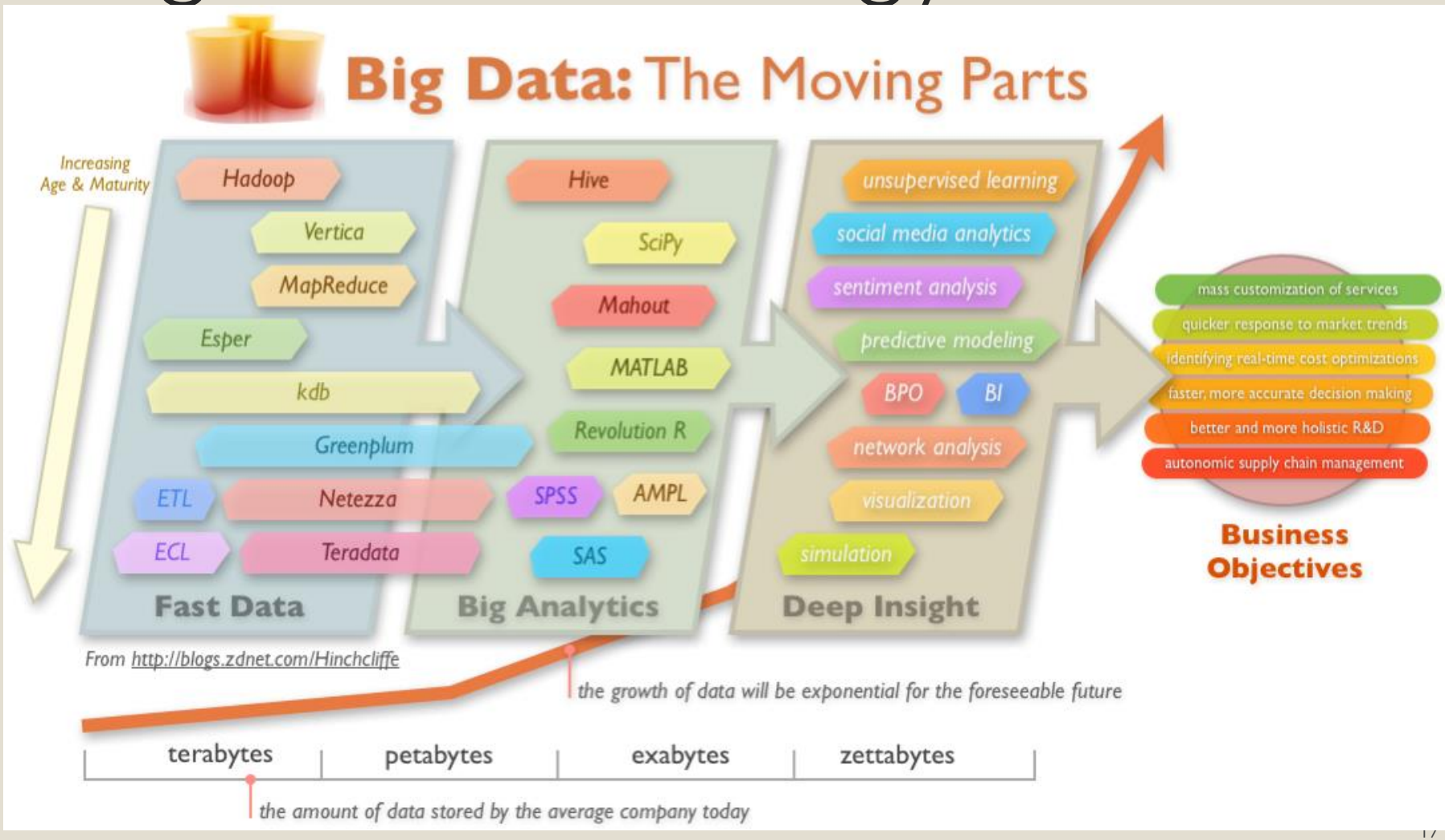
ORACLE
MySQL
Microsoft SQL Server
PostgreSQL
IBM DB2
SYBASE
memsql

Technologies

hadoop
hadoop MapReduce
mahout
APACHE HBASE
Cassandra

dave@vcdave.com
blogs.forbes.com/davefeinleib

Big Data Technology



What You Will Learn…

- We focus on ***Hadoop/MapReduce technology***
- **Learn the platform (how it is designed and works)**
  - How big data are managed in a scalable, efficient way
- **Learn writing Hadoop jobs in different languages**
  - Programming Languages: Java, C, Python
  - High-Level Languages: Apache Pig, Hive
- **Learn advanced analytics tools on top of Hadoop**
  - RHadoop: Statistical tools for managing big data
  - Mahout: Data mining and machine learning tools over big data
- **Learn state-of-art technology from recent research papers**
  - Optimizations, indexing techniques, and other extensions to Hadoop

Course Logistics

Course Logistics

- Web Page: http://web.cs.wpi.edu/~cs525/s13-MYE/
- Electronic WPI system: blackboard.wpi.edu
- Lectures
  - Tuesday, Thursday: (4:00pm - 5:20pm)

Textbook & Reading List

- **No specific textbook**
  - Big Data is a relatively new topic (so no fixed syllabus)

- **Reading List**
  - We will cover the state-of-art technology from research papers in big conferences
  - Many Hadoop-related papers are available on the course website

- **Related books:**
  - Hadoop, The Definitive Guide [pdf]

Requirements & Grading

- **Seminar-Type Course**
  - Students will read research papers and present them (Reading List)
- **Hands-on Course**
  - No written homework or exams
  - Several coding projects covering the entire semester

Done in teams of two

Course grades are divided as follows:

Item	Percentage	Notes
Projects (6 or 7)	50%	Each project will be done in teams of two.
Presentations (1 or 2)	25%	Each presentation will be done in teams of two. If the number of teams is large, some teams may do one presentation + an extra project.
Reviews	15%	Reviews are done individually. Whenever a team is presenting a paper, other students are expected to read the presented paper and submit a review on it.
Class Participation	10%	Includes discussions in class and attendance.

Requirements & Grading (Cont'd)

- **Reviews**
  - When a team is presenting (*not the instructor*), the other students should prepare a review on the presented paper
  - Course website gives guidelines on how to make good reviews
  - ***Reviews are done individually***

Course grades are divided as follows:

Item	Percentage	Notes
Projects (6 or 7)	50%	Each project will be done in teams of two.
Presentations (1 or 2)	25%	Each presentation will be done in teams of two. If the number of teams is large, some teams may do one presentation + an extra project.
Reviews	15%	Reviews are done individually. Whenever a team is presenting a paper, other students are expected to read the presented paper and submit a review on it.
Class Participation	10%	Includes discussions in class and attendance.

Late Submission Policy

- **For Projects**
  - One-day late → 10% off the max grade
  - Two-day late → 20% off the max grade
  - Three-day late → 30% off the max grade
  - Beyond that, no late submission is accepted
  - **Submissions:**
    - Submitted via blackboard system by the due date
    - Demonstrated to the instructor within the week after
- **For Reviews**
  - No late submissions
  - Student may skip at most 4 reviews
  - **Submissions:**
    - Given to the instructor at the beginning of class

More about Projects

- **A virtual machine is created including the needed platform for the projects**
  - Ubuntu OS (Version 12.10)
  - Hadoop platform (Version 1.1.0)
  - Apache Pig (Version 0.10.0)
  - Mahout library (Version 0.7)
  - Rhadoop
  - In addition to other software packages
- **Download it from the course website (link)**
  - Username and password will be sent to you
- **Need Virtual Box (Vbox)** [free]

Next Step from You…

1. Form teams of two
2. Visit the course website (Reading List), each team selects its first paper to present (1st come 1st served)
   - Send me your choices top 2/3 choices
3. You have until Jan 20th
   - Otherwise, I'll randomly form teams and assign papers
4. Use Blackboard "Discussion" forum for posts or for searching for teammates

Course Output: What You Will Learn…

- We focus on ***Hadoop/MapReduce technology***
- **Learn the platform (how it is designed and works)**
  - How big data are managed in a scalable, efficient way
- **Learn writing Hadoop jobs in different languages**
  - Programming Languages: Java, C, Python
  - High-Level Languages: Apache Pig, Hive
- **Learn advanced analytics tools on top of Hadoop**
  - RHadoop: Statistical tools for managing big data
  - Mahout: Analytics and data mining tools over big data
- **Learn state-of-art technology from recent research papers**
  - Optimizations, indexing techniques, and other extensions to Hadoop

UNIT 2: Introduction to Data Science

Topics

- databases and data architectures
- databases in the real world
  - scaling, data quality, distributed
- machine learning/data mining/statistics
- information retrieval

- Data Science is currently a popular interest of employers
  - our Industrial Affiliates Partners say there is high demand for students trained in Data Science
    - databases, warehousing, data architectures
    - data analytics – statistics, machine learning
  - Big Data – gigabytes/day or more
  - Examples:
    - Walmart, cable companies (ads linked to content, viewer trends), airlines/Orbitz, HMOs, call centers, Twitter (500M tweets/day), traffic surveillance cameras, detecting fraud, identity theft…
  - supports "Business Intelligence"
    - quantitative decision-making and control
    - finance, inventory, pricing/marketing, advertising
    - need data for identifying risks, opportunities, conducting "what-if" analyses

Data Architectures

- traditional databases (CSCE 310/608)
  - tables, fields
  - tuples = records or rows
    - <yellowstone,WY,6000000 acres,geysers>
  - key = field with unique values
    - can be used as a reference from one table into another
    - important for avoiding redundancy (normalization), which risks inconsistency
  - join – combining 2 tables using a key
  - metadata – data about the data
    - names of the fields, types (string, int, real, mpeg...)
    - also things like source, date, size, completeness/sampling

Name	HomeTown	Grad school	PhD	teaches	title
John Flaherty	Houston, TX	Rice	2005	CSCE 411	Design and Analysis of Algorithms
Susan Jenkins	Omaha, NE	Univ of Michigan	2004	CSCE 121	Introduction to Computing in C++
Susan Jenkins	Omaha, NE	Univ of Michigan	2004	CSCE 206	Programming in C
Bill Jones	Pittsburgh, PA	Carnegie Mellon	1999	CSCE 314	Programming Languages
Bill Jones	Pittsburgh, PA	Carnegie Mellon	1999	CSCE 206	Programming in C

Instructors:

Name	HomeTown	Grad school	PhD
John Flaherty	Houston, TX	Rice	2005
Susan Jenkins	Omaha, NE	Univ of Michigan	2004
Bill Jones	Pittsburgh, PA	Carnegie Mellon	1999

TeachingAssignments:

Name	teaches
John Flaherty	CSCE 411
Susan Jenkins	CSCE 121
Susan Jenkins	CSCE 206
Bill Jones	CSCE 314
Bill Jones	CSCE 206

Courses:

course	title
CSCE 411	Design and Analysis of Algorithms
CSCE 121	Introduction to Computing in C++
CSCE 314	Programming Languages
CSCE 206	Programming in C

SQL: Structured Query Language

```
>SELECT Name,HomeTown FROM Instructors WHERE PhD<2000;
Bill Jones Pittsburgh, PA

>SELECT Course,Title FROM Courses ORDER BY Course;
CSCE 121    Introduction to Computing in C++
CSCE 206    Programming in C
CSCE 314    Programming Languages
CSCE 411    Design and Analysis of Algorithms
```

can also compute sums, counts, means, etc.

example of JOIN: find all courses taught by someone from CMU:

```
>SELECT TeachingAssignments.Course
  FROM Instructors JOIN TeachingAssignments
   ON Instructors.Name=TeachingAssigmnents.Name
   WHERE Instructor.PhD="Carnegie Mellon"
CSCE 314
CSCE 206
```

because they were both taught by Bill Jones

- SQL servers
  - centralized database, required for concurrent access by multiple users
  - ODBC: Open DataBase Connectivity – protocol to connect to servers and do queries, updates from languages like Java, C, Python
  - Oracle, IBM DB2 - industrial strength SQL databases

some efficiency issues with real databases

- indexing
  - how to efficiently find all songs written by Paul Simon in a database with 10,000,000 entries?
  - data structures for representing sorted order on fields
- disk management
  - databases are often too big to fit in RAM, leave most of it on disk and swap in blocks of records as needed – could be slow
- concurrency
  - transaction semantics: either all updates happen *en batch* or none (commit or rollback)
  - like delete one record and simultaneously add another but guarantee not to leave in an inconsistent state
  - other users might be blocked till done
- query optimization
  - the order in which you JOIN tables can drastically affect the size of the intermediate tables

Unstructured data

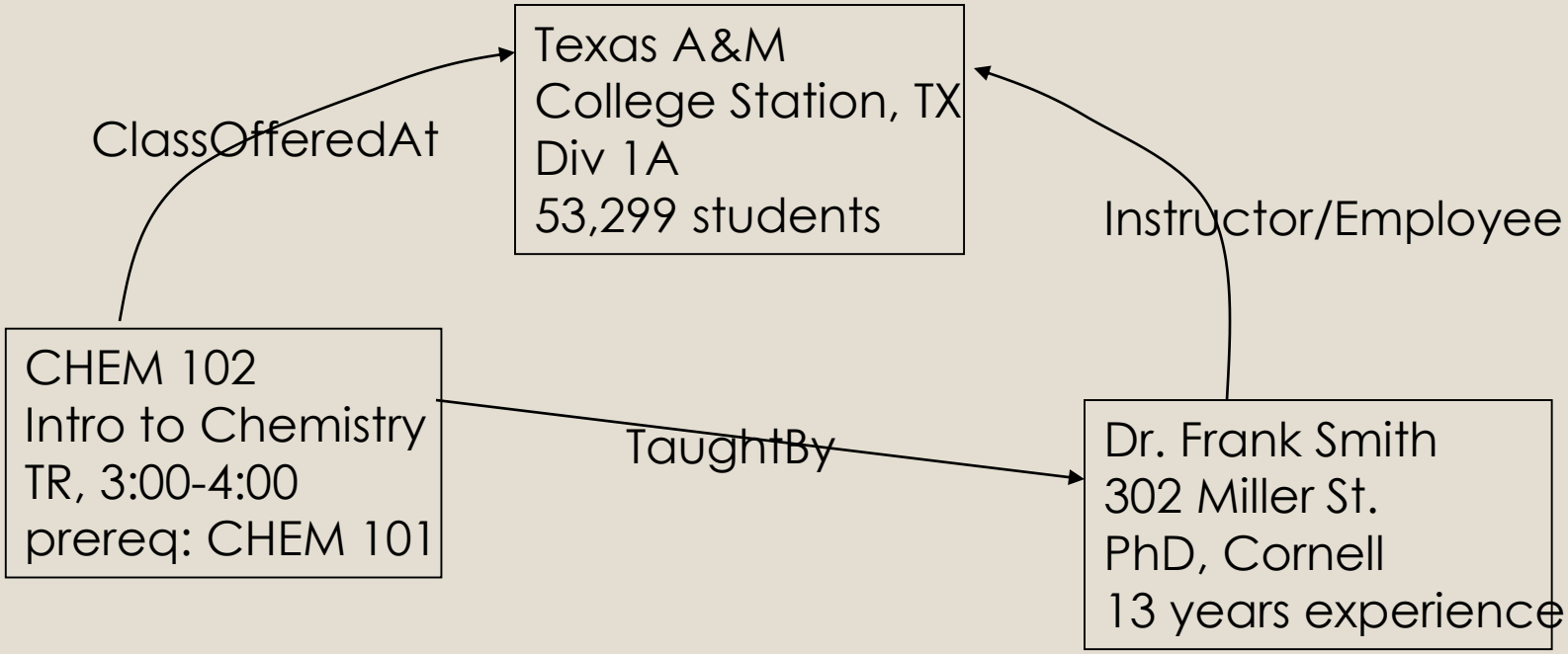
- raw text
  - documents, digital libraries
  - grep, substring indexing, regular expressions
    - like find all instances of "[aA]g+ies" including "aggggies"
  - Information Retrieval (CSCE 470)
  - look for synonyms, similar words (like "car" and "auto")
    - *tfIdf* (term frequency/inverse doc frequency) – weighting for important words
    - *LSI* (latent semantic indexing) – e.g. 'dogs' is similar to 'canines' because they are used similarly (both near 'bark' and 'bite')
  - Natural Language parsing
    - extracting requirements from jobs postings

Unstructured data

- images, video (BLOBs=binary large objects)
  - how to extract features? index them? search them?
  - color histograms
  - convolutions/transforms for pattern matching
  - looking for ICBM missiles in aerial photos of Cuba
- streams
  - sports ticker, radio, stock quotes...
- XML files
  - with tags indicating field names

```
<course>
    <name>CSCE 411</name>
    <title>Design and Analysis of Algorithms</title>
</course>
```

- Object databases



In a database with millions of objects, how do you efficiently do queries (i.e. follow pointers) and retrieve information?

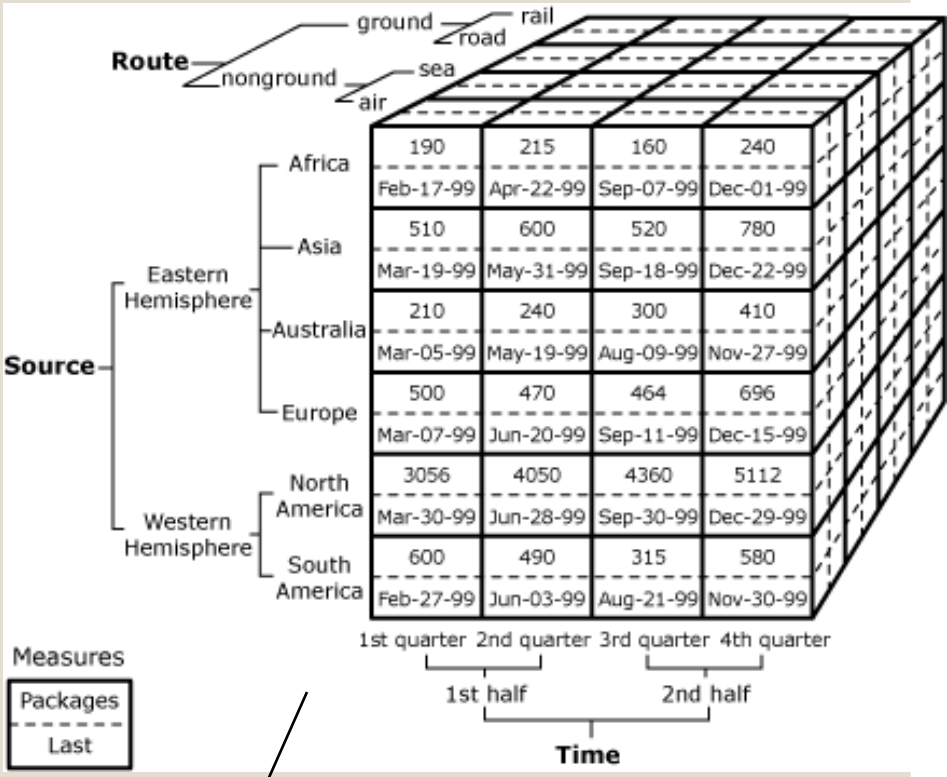
Real-world issues with databases

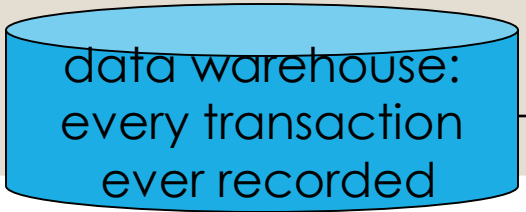
- it's all about scaling up to many records (and many users)
- data warehousing:
  - full database is stored in secure, off-site location
  - slices, snapshots, or views are put on interactive query servers for fast user access ("staging")
    - might be processed or summarized data
- databases are often distributed
  - different parts of the data held in different sites
  - some queries are local, others are "corporate-wide"
  - how to do distributed queries?
  - how to keep the databases synchronized?
  - CSCE 438 – Distributed Object Programming

OLAP: OnLine Analytical Processing

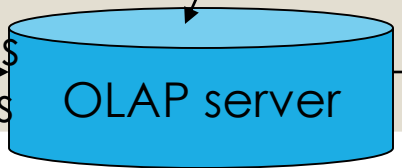
- multi-dimensional tables of aggregated sales in different regions in recent quarters, rather than “every transaction”
- users can still look at seasonal or geographic trends in different product categories
- project data onto 2D spreadsheets, graphs

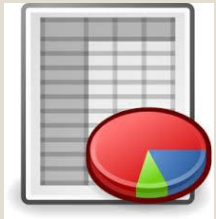
http://technet.microsoft.com/en-us/library/ms174587.aspx





nightly updates and summaries





data integrity

- missing values
  - how to interpret? not available? 0? use the mean?
- duplicated values
  - including partial matches (Jon Smith=John Smith?)
- inconsistency:
  - multiple addresses for person
- out-of-date data
- inconsistent usage:
  - does “destination” mean of first leg or whole flight?
- outliers:
  - salaries that are negative, or in the trillions
- most database allow “integrity constraints” to be defined that validate newly entered data

Interoperability

- how can data from one database be compared or combined with another?
- what if fields are not the same, or not present, or used differently?
- think of medical or insurance records
- translation/mapping of terms
- standards
  - units like ft/s, or gallons, etc.
  - identifiers like SSN, UIN, ISBN
- "federated" databases – queries that combine information across multiple servers

"Data cleansing"

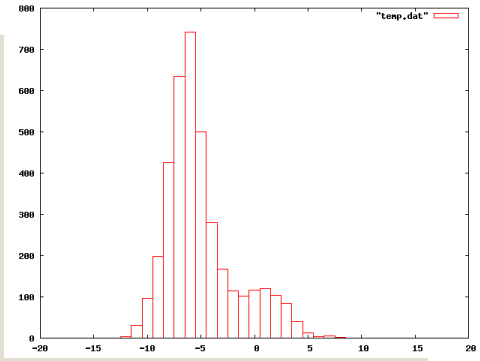
- filling in missing data (imputing values)
- detecting and removing outliers
- smoothing
  - removing noise by averaging values together
- filtering, sampling
  - keeping only selected representative values
- feature extraction
  - e.g. in a photo database, which people are wearing glasses? which have more than one person? which are outdoors?

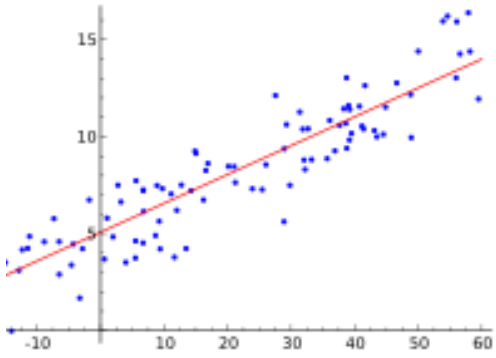
Data Mining/Data Analytics

- finding patterns in the data
- statistics
- machine learning

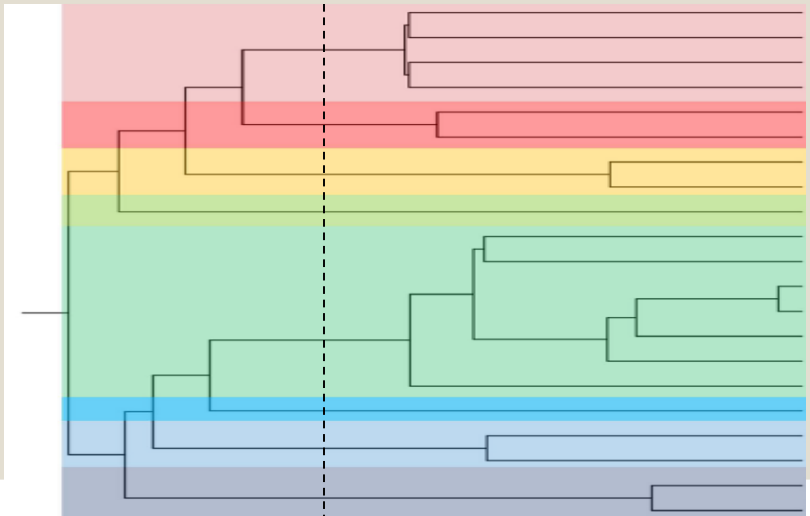
(CSCE 633)

- Numerical data
  - correlations
  - multivariate regression
  - fitting "models"
    - predictive equations that fit the data
    - from a real estate database of home sales, we get
    - *housing price = 100*SqFt - 6*DistanceToSchools + 0.1*AverageOfNeighborhood*
  - ANOVA for testing differences between groups
  - *R* is one of the most commonly used software packages for doing statistical analysis
    - can load a data table, calculate means and correlations, fit distributions, estimate parameters, test hypotheses, generate graphs and histograms





- clustering
  - similar photos, documents, cases
  - discovery of "structure" in the data
  - example: accident database
    - some clusters might be identified with "accidents involving a tractor trailer" or "accidents at night"
  - top-down vs. bottom-up clustering methods
  - granularity: how many clusters?

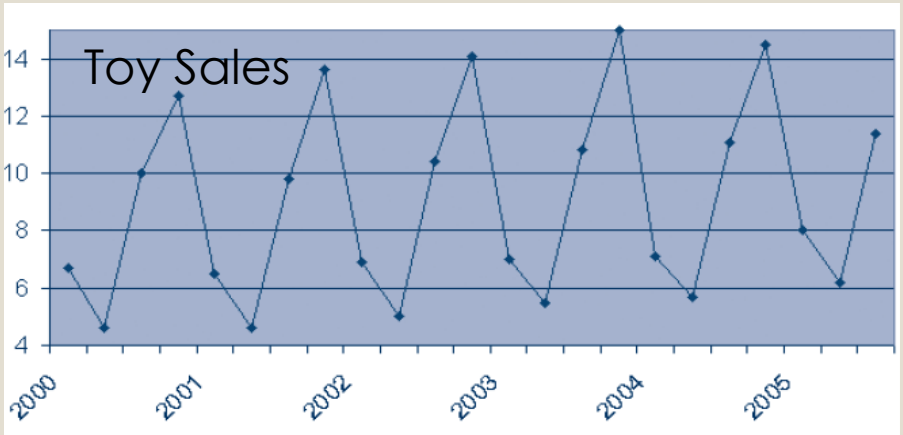


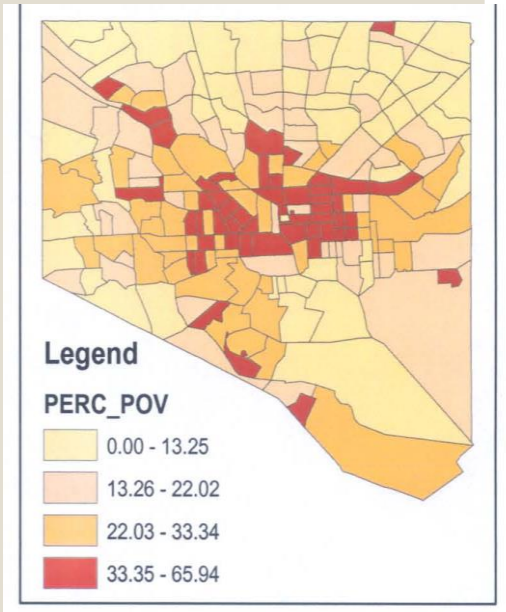
- decision trees (classifiers)
  - what factors, decisions, or treatments led to different outcomes?
  - recursive partitioning algorithms
  - related methods
    - "discriminant" analysis
      - what factors lead to return of product?
    - extract "association rules"
      - boxers dogs tend to have congenital defects
      - covers 5% of patients with 80% confidence

Veterinary database - dogs treated for disease

breed	gender	age	drug	sibsp	outcome
terrier	F	10	methotrexate	4.0	died
spaniel	M	5	cytarabine	2.3	survived
doberman	F	7	doxorubicin	0.1	died

- other types of data
  - time series and forecasting:
    - model the price of gas using *autoregression*
    - a function of recent prices, demand, geopolitics...
    - de-trend: factor out seasonal trends
  - GIS (geographic information systems)
    - longitude/latitude coordinates in the database
    - objects: city/state boundaries, river locations, roads
    - find regions in B/CS with an excess of coffee shops





from: Basic Statistics for Business and Economics, Lind et al (2009), Ch 16.

credit: Frank Curriero

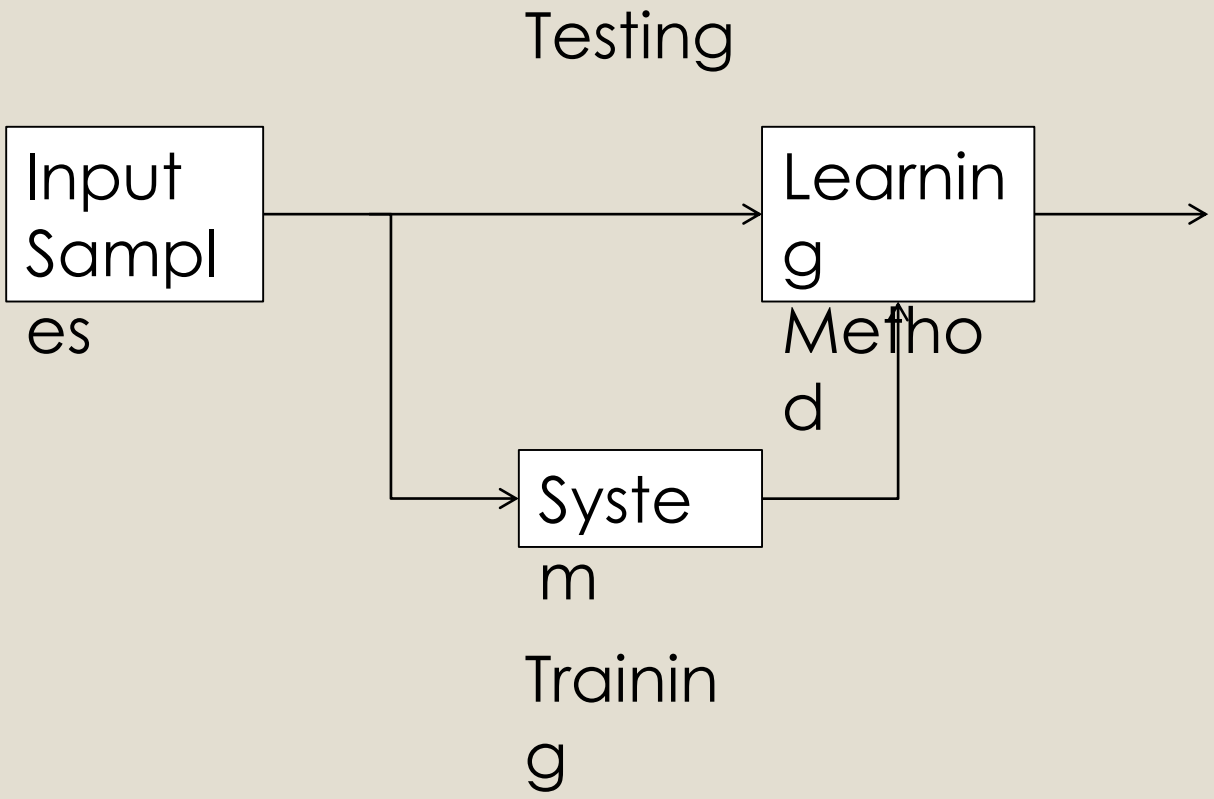
UNIT 3: Introduction to Machine learning

What is machine learning?

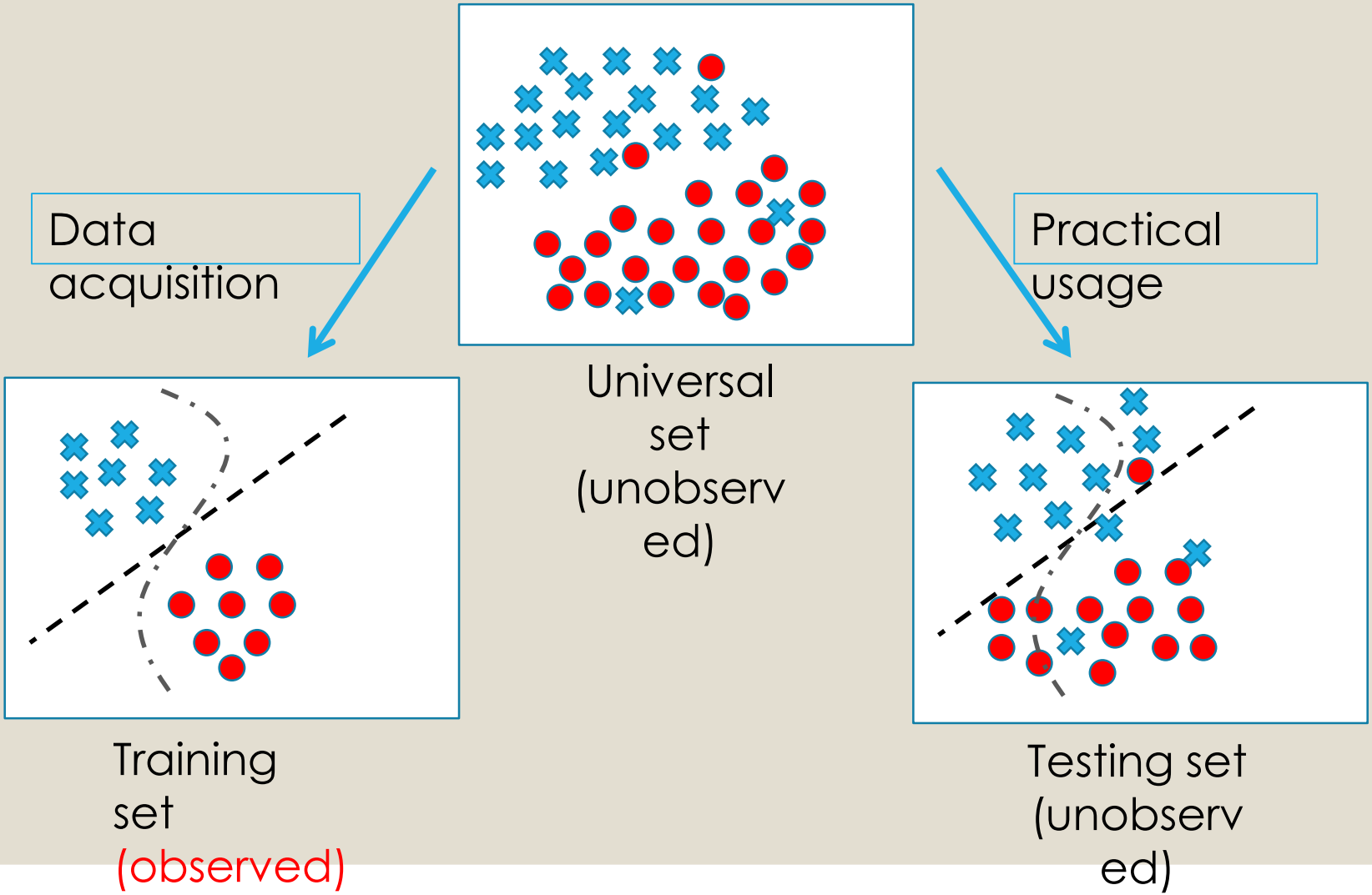
- A branch of **artificial intelligence**, concerned with the design and development of algorithms that allow computers to evolve behaviors based on empirical data.

- As intelligence requires knowledge, it is necessary for the computers to acquire knowledge.

Learning system model



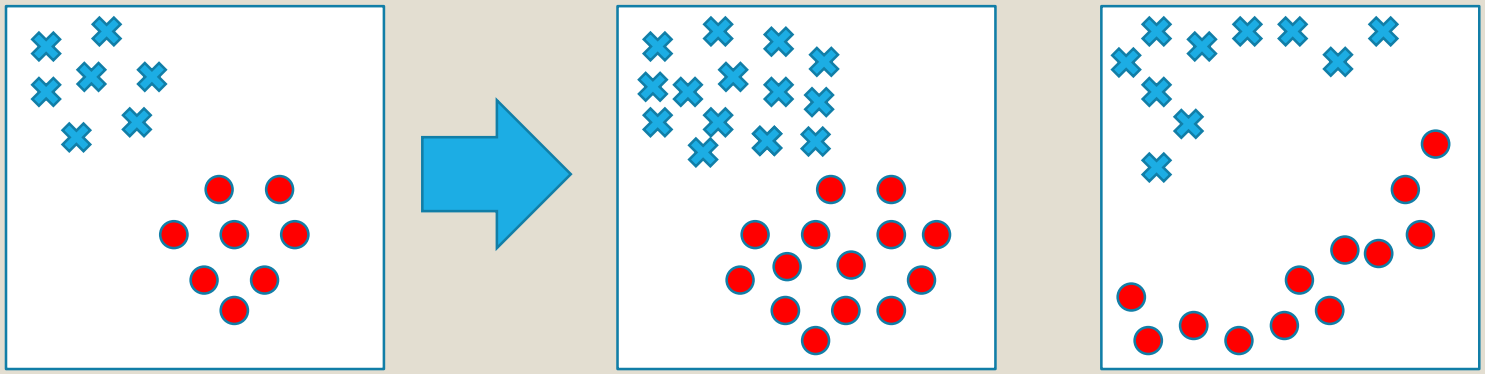
Training and testing



Training and testing

- Training is the process of making the system able to learn.

- No free lunch rule:
  - Training set and testing set come from the same distribution
  - Need to make some assumptions or bias



Performance

- There are several factors affecting the performance:
  - **Types of training** provided
  - The form and extent of any initial **background knowledge**
  - The **type of feedback** provided
  - The **learning algorithms** used

- Two important factors:
  - Modeling
  - Optimization

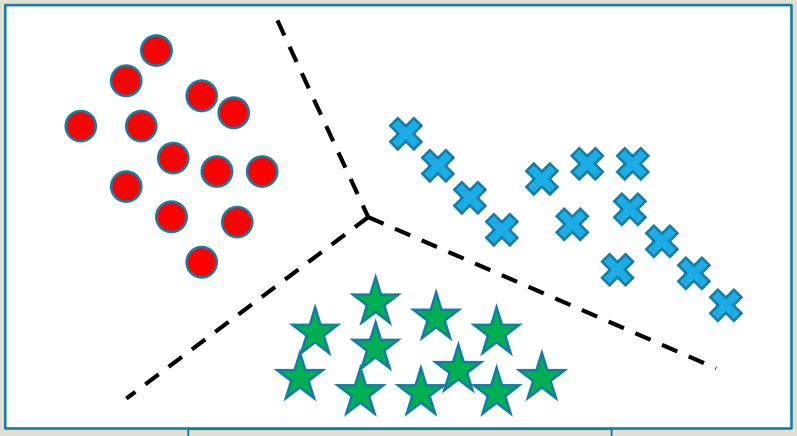
Algorithms

- The success of machine learning system also depends on the algorithms.
- The algorithms control the search to find and build the knowledge structures.
- The learning algorithms should extract useful information from training examples.

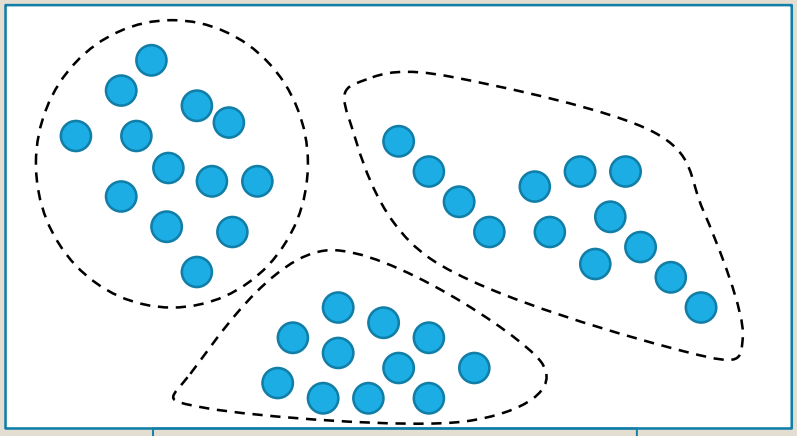
Algorithms

- **Supervised learning** ($\{x_n \in R^d, y_n \in R\}_{n=1}^N$)
  - Prediction
  - Classification (discrete labels), Regression (real values)
- **Unsupervised learning** ($\{x_n \in R^d\}_{n=1}^N$)
  - Clustering
  - Probability distribution estimation
  - Finding association (in features)
  - Dimension reduction
- **Semi-supervised learning**
- **Reinforcement learning**
  - Decision making (robot, chess machine)

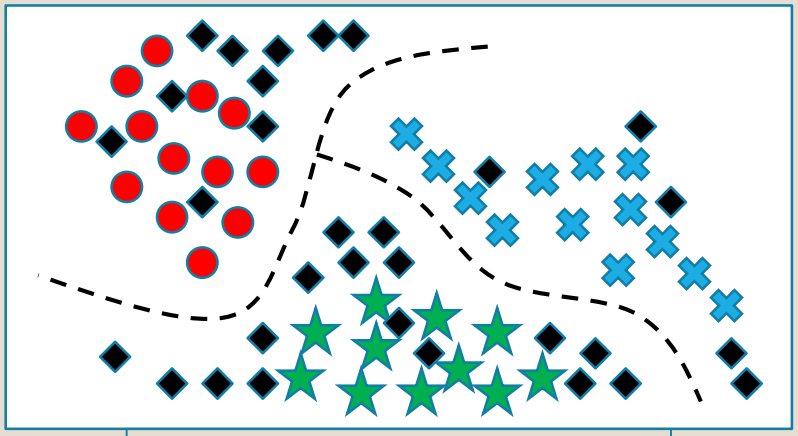
Algorithms



Supervised learning



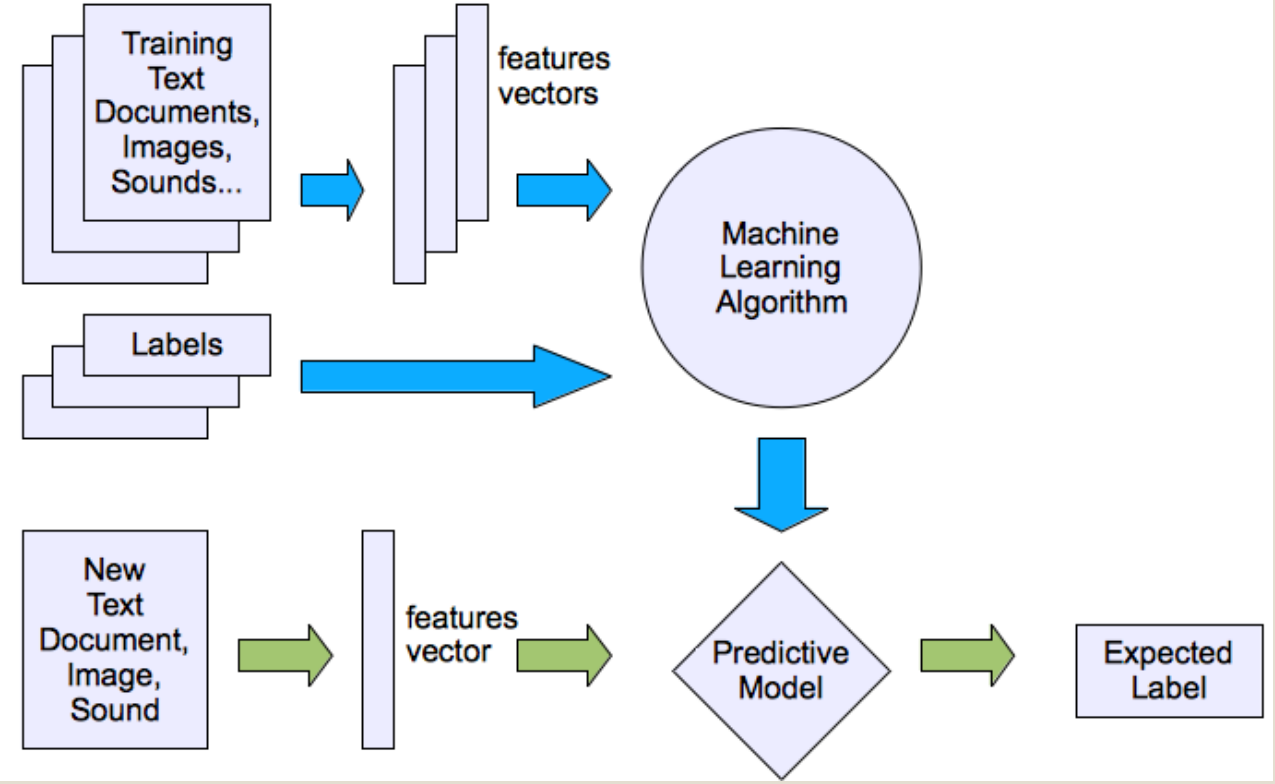
Unsupervised learning



Semi-supervised learning

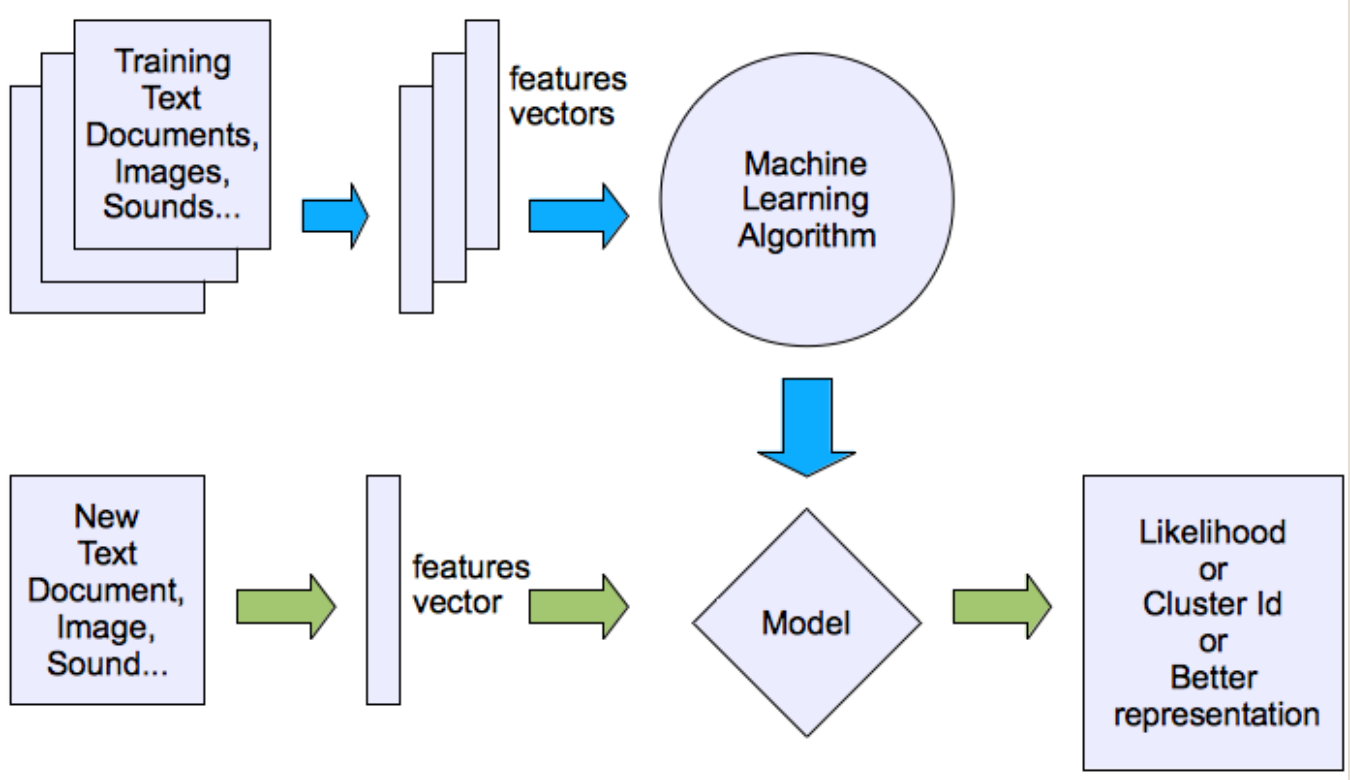
Machine learning structure

- Supervised learning



Machine learning structure

- Unsupervised learning



What are we seeking?

- Supervised: Low E-out or maximize probabilistic terms

$$error = \frac{1}{N}\sum_{n=1}^{N}[y_n \neq g(x_n)]$$

E-in: for training set

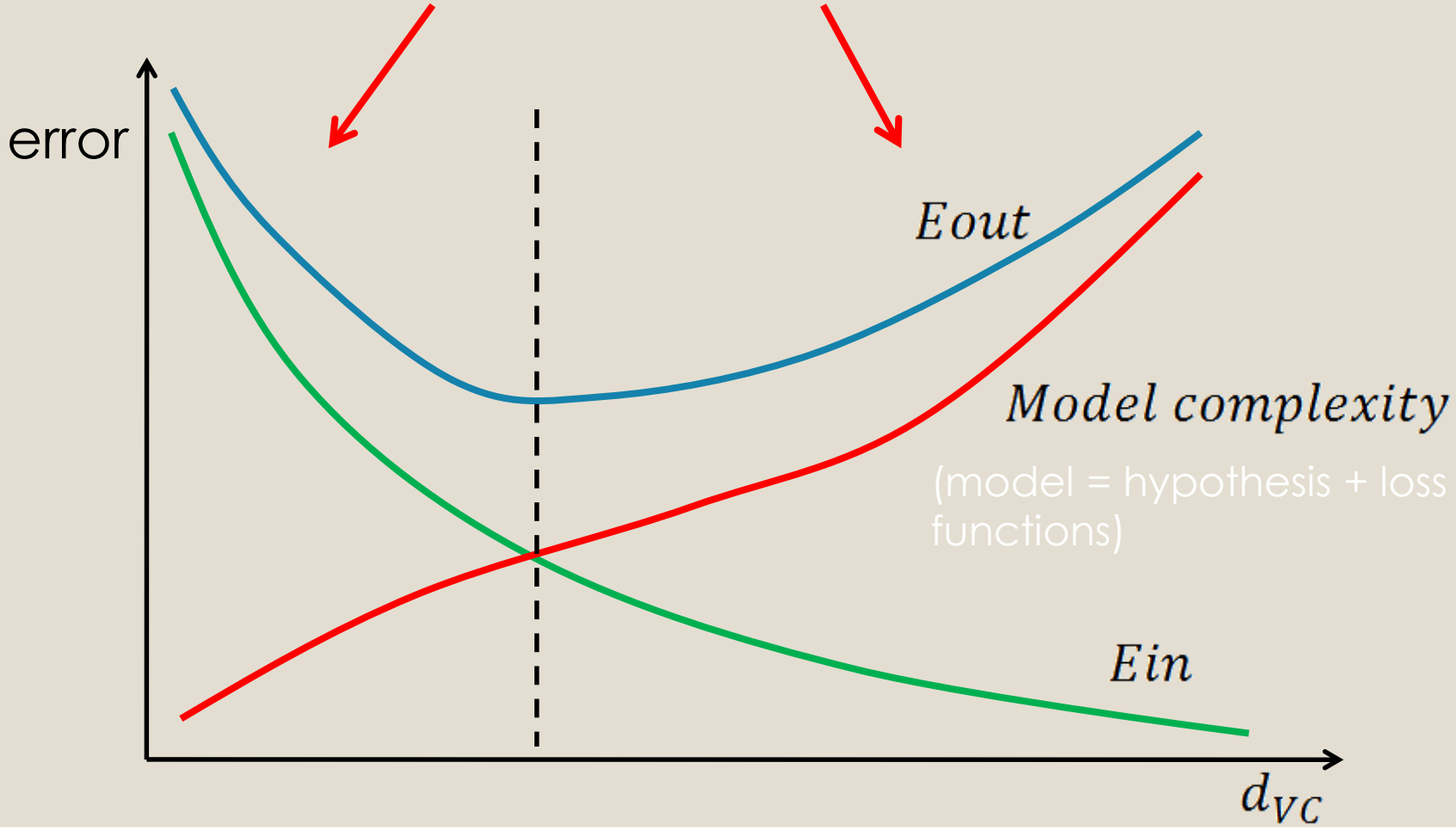
E-out: for testing

$$Eout(g) \leq Ein(g) \pm O\left(\sqrt{\frac{d_{VC}}{N} ln\, N}\right)$$

- Unsupervised: Minimum quantization error, Minimum distance, MAP, MLE(maximum likelihood estimation)

What are we seeking?

Under-fitting VS. Over-fitting (fixed *N*)

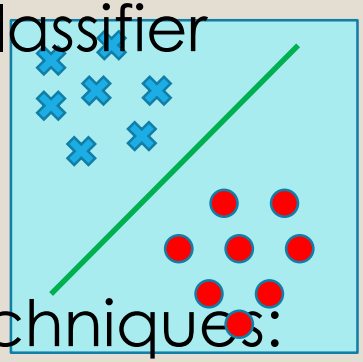


Learning techniques

- Supervised learning categories and techniques
  - **Linear classifier** (numerical functions)
  - **Parametric** (Probabilistic functions)
    - Naïve Bayes, Gaussian discriminant analysis (GDA), Hidden Markov models (HMM), Probabilistic graphical models
  - **Non-parametric** (Instance-based functions)
    - *K*-nearest neighbors, Kernel regression, Kernel density estimation, Local regression
  - **Non-metric** (Symbolic functions)
    - Classification and regression tree (CART), decision tree
  - **Aggregation**
    - Bagging (bootstrap + aggregation), Adaboost, Random forest

Learning techniques

- Linear classifier



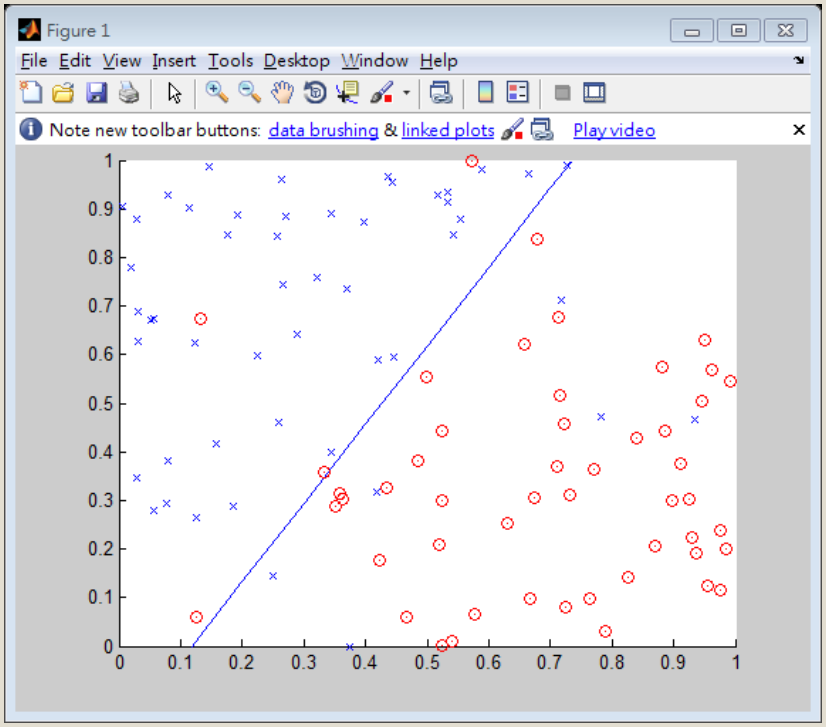
$$g(x_n) = sign(w^T x_n)$$

, where w is an *d*-dim vector (learned)

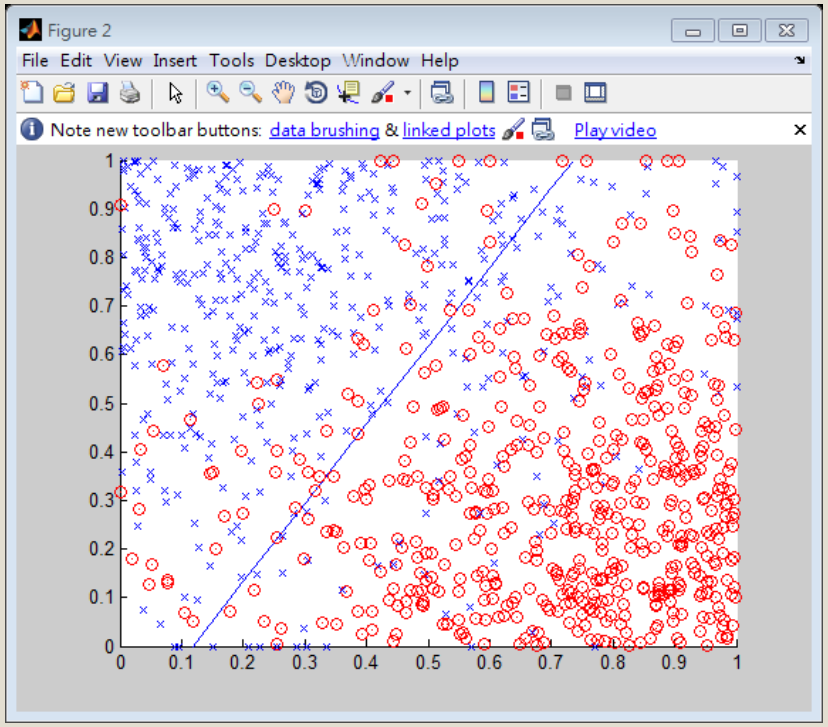
- Techniques:
  - Perceptron
  - Logistic regression
  - Support vector machine (SVM)
  - Ada-line
  - Multi-layer perceptron (MLP)

Learning techniques

Using **perceptron learning algorithm**(PLA)



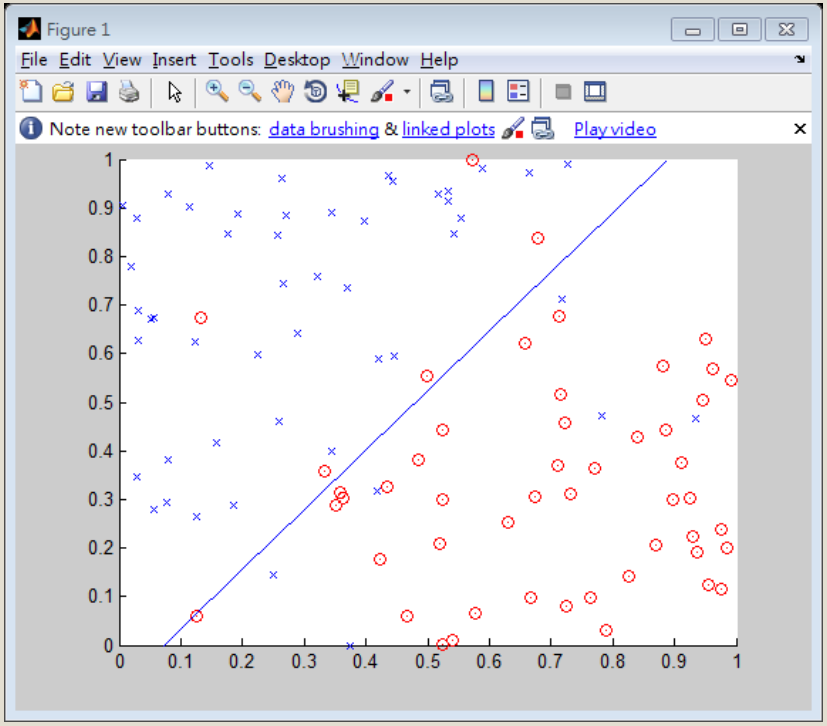
Training Error rate: 0.10



Testing Error rate: 0.156

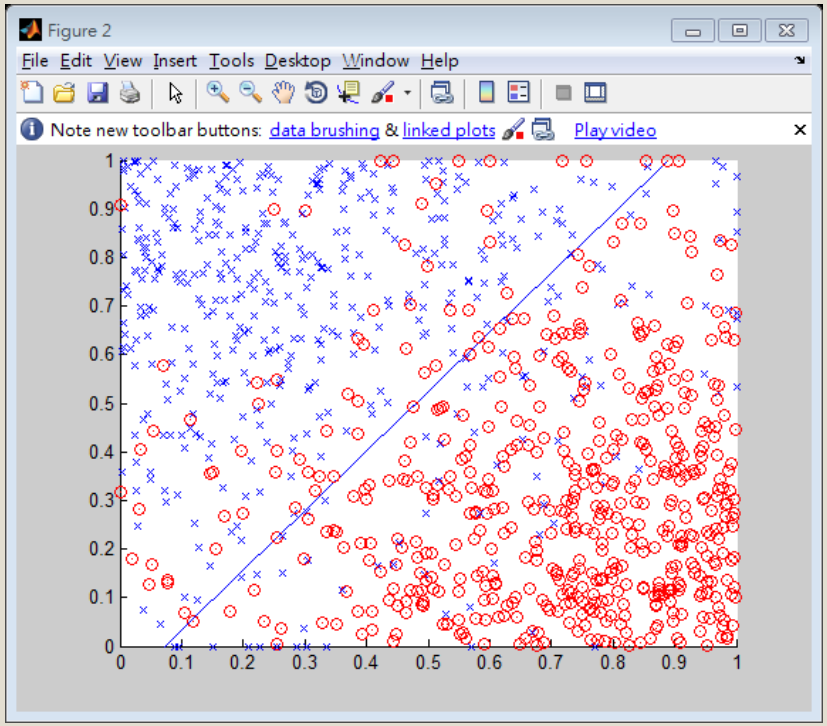
Learning techniques

Using **logistic regression**



Training

Error rate: 0.11

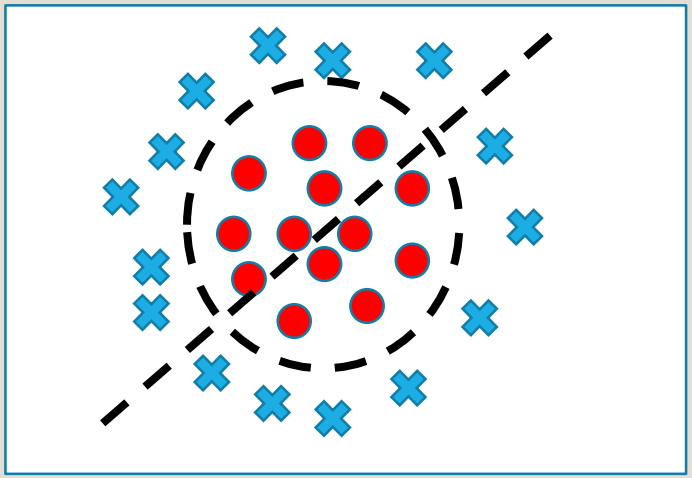


Testing

Error rate: 0.145

Learning techniques

- Non-linear case



$$x_n = [x_{n1}, x_{n2}]$$

$$x_n = [x_{n1}, x_{n2}, x_{n1} * x_{n2}, {x_{n1}}^2, {x_{n2}}^2]$$

$$g(x_n) = sign(w^T x_n)$$

- Support vector machine (SVM):
  - Linear to nonlinear: **Feature transform** and **kernel function**

Learning techniques

- Unsupervised learning categories and techniques
  - **Clustering**
    - K-means clustering
    - Spectral clustering
  - **Density Estimation**
    - Gaussian mixture model (GMM)
    - Graphical models
  - **Dimensionality reduction**
    - Principal component analysis (PCA)
    - Factor analysis

Applications

- Face detection
- Object detection and recognition
- Image segmentation
- Multimedia event detection
- Economical and commercial usage

UNIT 4: Data Analytics with R/Weka Machine learning

We'll Cover

- **What is R**
- **How to obtain and install R**
- **How to read and export data**
- **How to do basic statistical analyses**
- **Econometric packages in R**

What is R

- Software for Statistical Data Analysis
- Based on S
- Programming Environment
- Interpreted Language
- Data Storage, Analysis, Graphing
- Free and Open Source Software

Obtaining R

- **Current Version: R-2.0.0**
- **Comprehensive R Archive Network:**

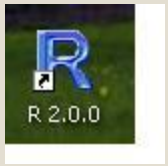
  http://cran.r-project.org
- **Binary source codes**
- **Windows executables**
- **Compiled RPMs for Linux**
- **Can be obtained on a CD**

Installing R

- **Binary (Windows/Linux): One step process**
  - **exe, rpm (Red Hat/Mandrake), apt-get (Debian)**
- **Linux, from sources:**

```
$ tar –zxvf “filename.tar.gz”
$ cd filename
$ ./configure
$ make
$ make check
$ make install
```

Starting R



Windows, Double-click on Desktop Icon

```
$ R
```

Linux, type R at command prompt

Strengths and Weaknesses

- **Strengths**
  - **Free and Open Source**
  - **Strong User Community**
  - **Highly extensible, flexible**
  - **Implementation of high end statistical methods**
  - **Flexible graphics and intelligent defaults**
- **Weakness**
  - **Steep learning curve**
  - **Slow for large datasets**

Basics

- Highly Functional
  - Everything done through functions
  - Strict named arguments
  - Abbreviations in arguments OK (e.g. T for TRUE)
- Object Oriented
  - Everything is an object
  - "< –" is an assignment operator
  - "X <- 5": X GETS the value 5

Getting Help in R

- From Documentation:
  - ?WhatIWantToKnow
  - help("WhatIWantToKnow")
  - help.search("WhatIWantToKnow")
  - help.start()
  - getAnywhere("WhatIWantToKnow")
  - example("WhatIWantToKnow")
- Documents: "Introduction to R"
- Active Mailing List
  - Archives
  - Directly Asking Questions on the List

Data Structures

- Supports virtually any type of data
- Numbers, characters, logicals (TRUE/ FALSE)
- Arrays of virtually unlimited sizes
- Simplest: Vectors and Matrices
- Lists: Can Contain mixed type variables
- Data Frame: Rectangular Data Set

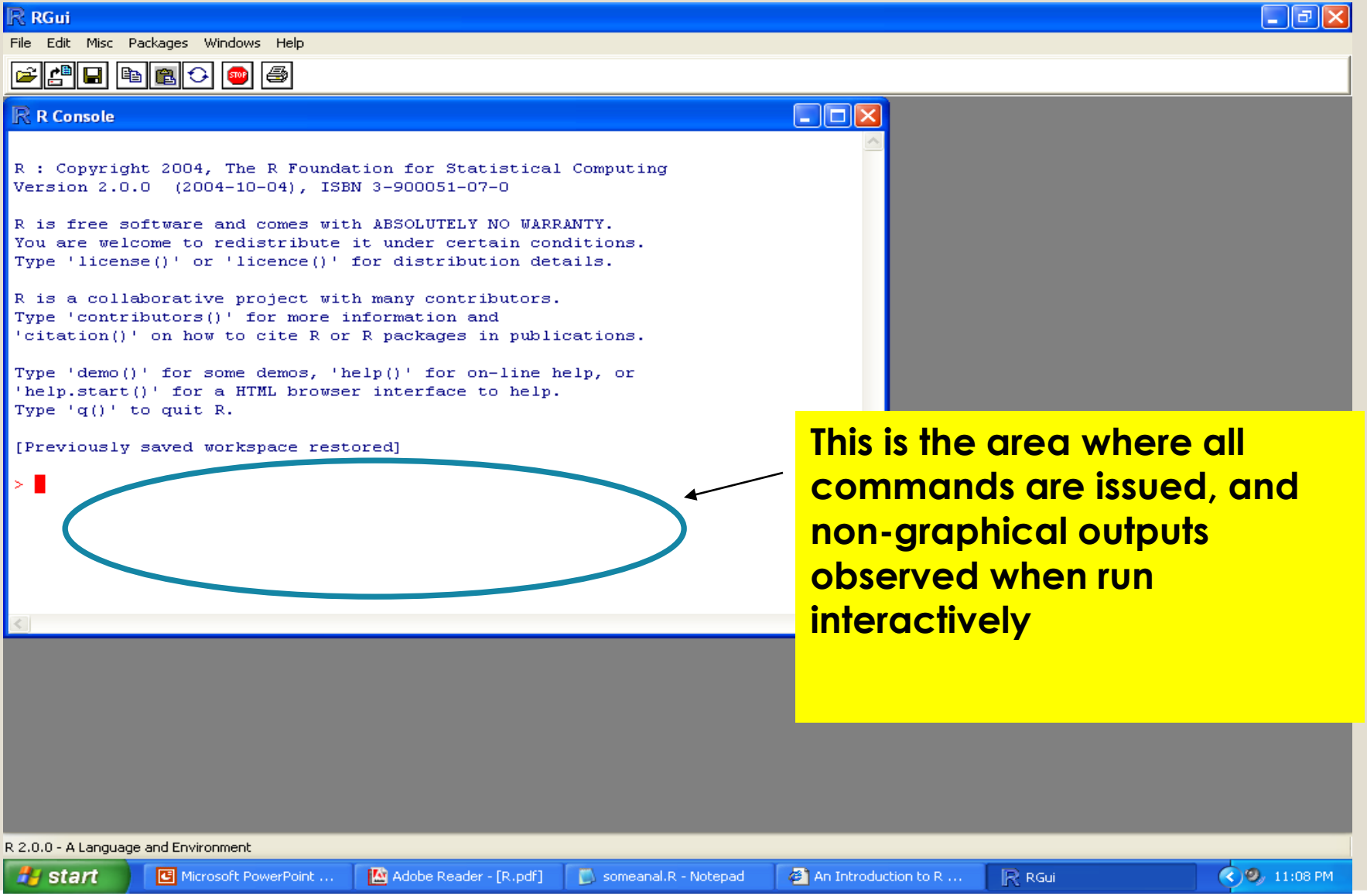
Data Structure in R

	Linear	Rectangular
All Same Type	VECTORS	MATRIX*
Mixed	LIST	DATA FRAME

Running R

- Directly in the Windowing System (Console)
- Using Editors
  - Notepad, WinEdt, Tinn-R: Windows
  - Xemacs, ESS (Emacs speaks Statistics)
- On the Editor:
  - source("filename.R")
  - Outputs can be diverted by using
    - sink("filename.Rout")

R Working Area



In an R Session…

- First, read data from other sources
- Use packages, libraries, and functions
- Write functions wherever necessary
- Conduct Statistical Data Analysis
- Save outputs to files, write tables
- Save R workspace if necessary (exit prompt)

Specific Tasks

- To see which directories and data are loaded, type: search()
- To see which objects are stored, type: ls()
- To include a dataset in the searchpath for analysis, type: attach(NameOfTheDataset, expression)
- To detach a dataset from the searchpath after analysis, type: detach(NameOfTheDataset)

Reading data into R

- R not well suited for data preprocessing
- Preprocess data elsewhere (SPSS, etc…)
- Easiest form of data to input: text file
- Spreadsheet like data:
  - Small/medium size: use read.table()
  - Large data: use scan()
- Read from other systems:
  - Use the library "foreign": library(foreign)
  - Can import from SAS, SPSS, Epi Info
  - Can export to STATA

Reading Data: summary

- Directly using a vector e.g.: x <- c(1,2,3…)
- Using scan and read.table function
- Using matrix function to read data matrices
- Using data.frame to read mixed data
- library(foreign) for data from other programs

Accessing Variables

- edit(<mydataobject>)
- Subscripts essential tools
  - x[1] identifies first element in vector x
  - y[1,] identifies first row in matrix y
  - y[,1] identifies first column in matrix y
- $ sign for lists and data frames
  - myframe$age gets age variable of myframe
  - attach(dataframe) -> extract by variable name

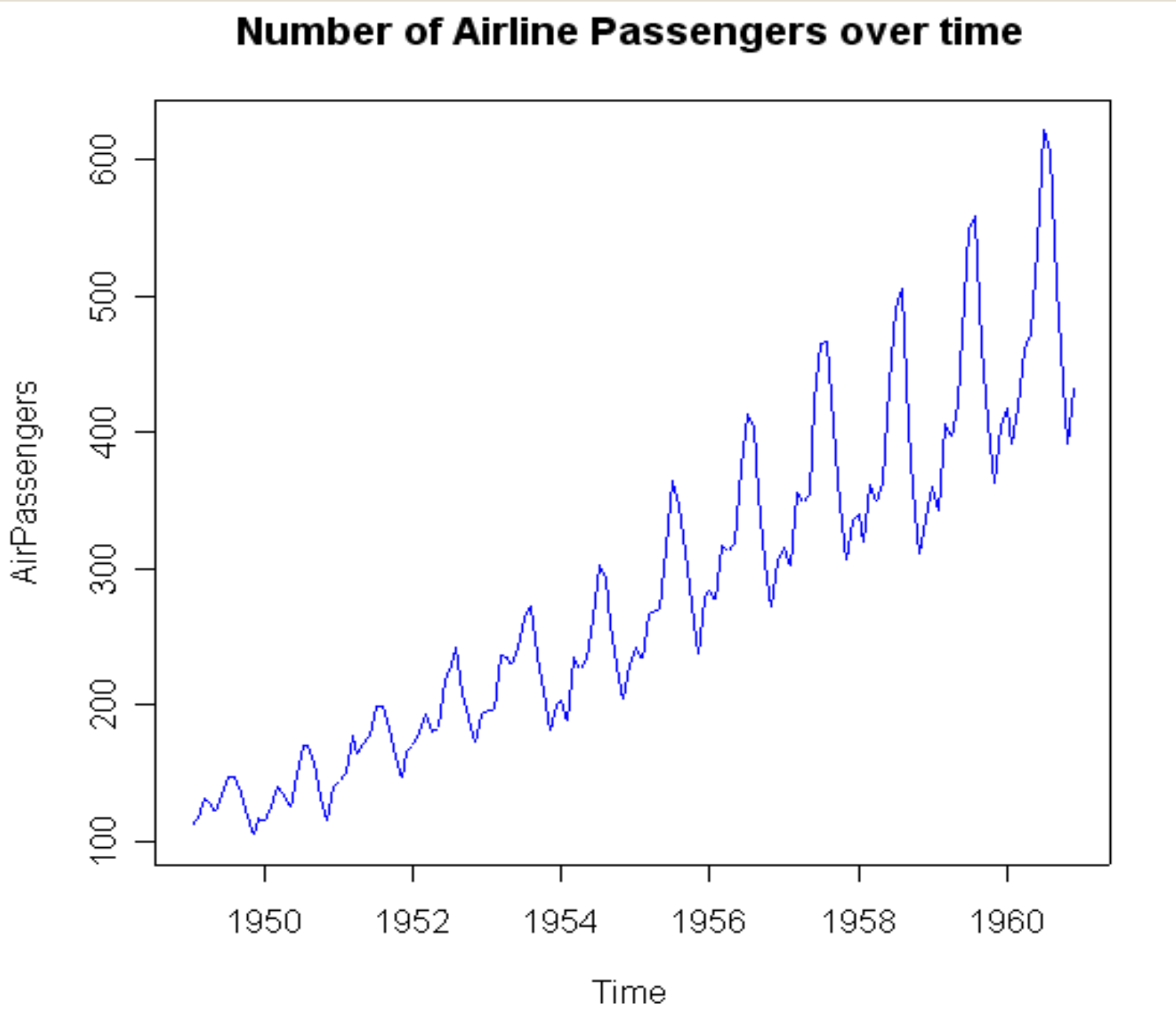
Subset Data

- Using subset function
  - subset() will subset the dataframe
- Subscripting from data frames
  - myframe[,1] gives first column of myframe
- Specifying a vector
  - myframe[1:5] gives first 5 rows of data
- Using logical expressions
  - myframe[myframe[,1], < 5,] gets all rows of the first column that contain values less than 5

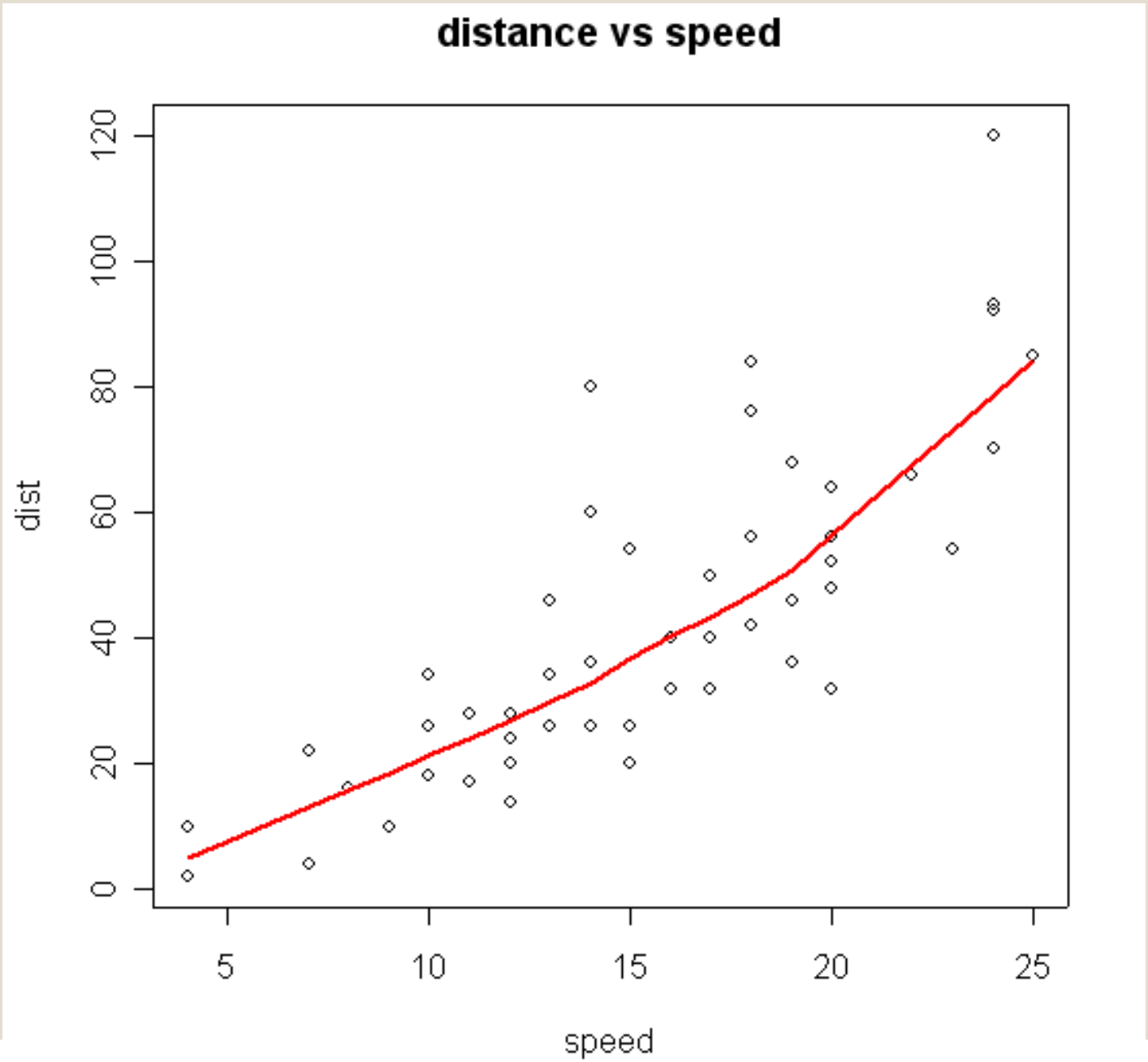
Graphics

- Plot an object, like: plot(num.vec)
  - here plots against index numbers
- Plot sends to graphic devices
  - can specify which graphic device you want
    - postscript, gif, jpeg, etc…
    - you can turn them on and off, like: dev.off()
- Two types of plotting
  - high level: graphs drawn with one call
  - Low Level: add additional information to existing graph

High Level: generated with plot()



Low Level: Scattergram with Lowess



Programming in R

- Functions & Operators typically work on entire vectors
- Expressions surrounded by {}
- Codes separated by newlines, ";" not necessary
- You can write your own functions and use them

Statistical Functions in R

- Descriptive Statistics
- Statistical Modeling
  - Regressions: Linear and Logistic
  - Probit, Tobit Models
  - Time Series
- Multivariate Functions
- Inbuilt Packages, contributed packages

Descriptive Statistics

- Has functions for all common statistics
- summary() gives lowest, mean, median, first, third quartiles, highest for numeric variables
- stem() gives stem-leaf plots
- table() gives tabulation of categorical variables

Statistical Modeling

- Over 400 functions
  - lm, glm, aov, ts
- Numerous libraries & packages
  - survival, coxph, tree (recursive trees), nls, …
- Distinction between factors and regressors
  - factors: categorical, regressors: continuous
  - you must specify factors unless they are obvious to R
  - dummy variables for factors created automatically
- Use of data.frame makes life easy

How to model

- Specify your model like this:
  - $y \sim x_i + c_i$, where
  - y = outcome variable, x_i = main explanatory variables, c_i = covariates, + = add terms
  - Operators have special meanings
    - + = add terms, : = interactions, / = nesting, so on…
- Modeling -- object oriented
  - each modeling procedure produces objects
  - classes and functions for each object

Synopsis of Operators

Operator	Usually means	In Formula means
+ or -	add or subtract	add or remove terms
*	multiplication	main effect and interactions
/	division	main effect and nesting
:	sequence	interaction only
^	exponentiation	limiting interaction depths
%in%	no specific	nesting only

Modeling Example: Regression

carReg <- lm(speed~dist, data=cars)

carReg = becomes an object

to get summary of this regression, we type

summary(carReg)

to get only coefficients, we type

coef(carReg), or carReg$coef

don't want intercept? add 0, so

carReg <- lm(speed~0+dist, data=cars)