Big Data

Stanisław Matwin
stan@cs.dal.ca

Canada Research Chair
Professor Faculty of Computer Science
Director, Big Data Analytics Institute
Dalhousie University
Kanada

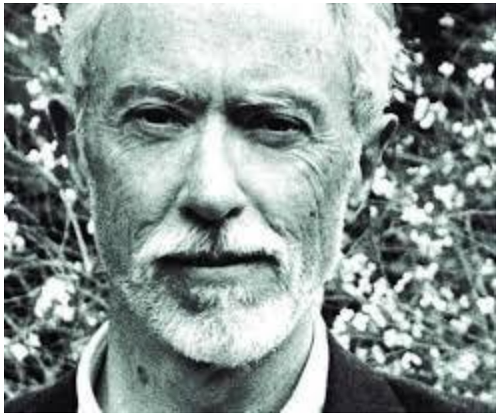
WSE

Warsaw, December 2013

DALHOUSIE UNIVERSITY
Inspiring Minds

Institute for Big Data Analytics

“He who comes to teach learns the keenest of lessons”



J. M. Coetzee Nobel Prize for literature, 2003

Contents

1. General Big Data discussion
   – What is BD?
   – Why BD?
   – How to do it?
2. Data mining for BD –
   – The CRISP model
   – Decision trees
   – Random Forest
   – Bayesian learning
     • Scalability
     • Text data
     • DMNB
   – Clustering
   – Finite mixture model
   – Association rules

Contents

3. High Performance Computing
   – Intro. to Map-Reduce
4. Vizualization
5. Privacy
   – Privacy-preserving Data Mining
   – Data publishing
     - Obfuscation
     - Cryptographic approaches
6. How to tech Big Data?

- a possible curriculum in the CS context
- a possible graduate curriculum in the CS context

7. Discussion on Big Data within WSE

Big Data

- Volume
- Velocity
- Variety
- Veracity
- … and Value

Rationale – why?

- McKinsey anticipates shortage of 140,000-190,000 “deep analytical positions” in the US by 2018
- Davos World Economic Forum – “big data” creates unprecedented opportunities for international development

Volume

1000^4	TB	terabyte
1000^5	PB	petabyte
1000^6	EB	**exabyte**
1000^7	ZB	zettabyte
1000^8	YB	yottabyte

- Library of Congress: 10TB of books, about 3PB of digitized material
- as of 2012, every day 2.5 exabytes (2.5×10^{18}) of data were created (IBM)
- All of data created until 2003 = all of data created since (Google)

Velocity

- Sensor data
- Streaming data
- Internet data
- Soc net data
- Etc.

Variety

- Eg medical data
  - Patient data (database, structured)
  - Doctor/nurse notes: text, unstructured
  - Tests: imaging data, graph data
- Challenge: to connect it

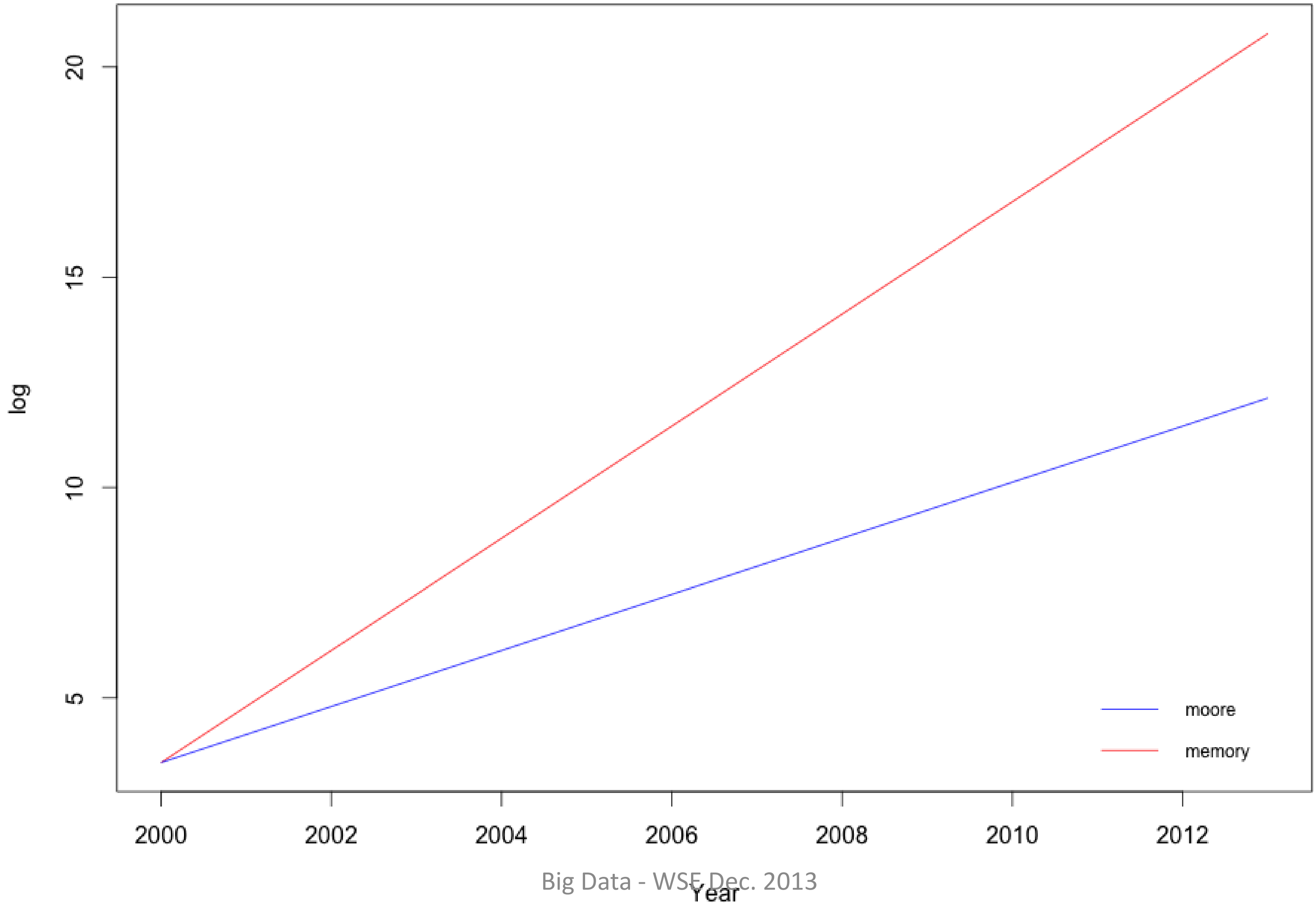
Veracity

- Quality of the data:
  - Noise
  - Missing data
  - Incorrectly entered data
  - …

Another view of Big Data

- Assetization of data
- From data to….
- Actionable knowledge

Moore's law *vs* memory law



Some history

- Technologies behind big data:
  - Data capture/transmission
  - Data bases/storage
  - Data mining
  - HPC (High-Performance Computing)/the Cloud
  - Visualization

More history…

1980s

Machine learning….

2000…

Data mining….

2012…

Big data…


IN 60 SECONDS...
1 NEW DEFINITION IS ADDED ON urban
1,600+ READS ON Scribd.
THE LARGEST SOCIAL READING PUBLISHING COMPANY!!
13,000+ HOURS MUSIC STREAMING ON PANDORA
12,000+ NEW ADS POSTED ON craigslist
New Craigslist Ads
370,000+ MINUTES VOICE CALLS ON skype
98,000+ TWEETS
320+ NEW twitter ACCOUNTS
100+ NEW LinkedIn ACCOUNTS
1 associatedcontent NEW ARTICLE IS PUBLISHED
THE WORLD'S LARGEST COMMUNITY CREATED CONTENT!!
6,600+ NEW PICTURES ARE UPLOADED ON flickr
50+ WORDPRESS DOWNLOADS
125+ PLUGIN DOWNLOADS
695,000+ facebook STATUS UPDATES
79,364 WALL POSTS
510,040 COMMENTS
1,700+ Firefox DOWNLOADS
694,445 SEARCH QUERIES
Google
168 MILLION EMAILS ARE SENT
60+ NEW BLOGS
1,500+ BLOG POSTS
70+ DOMAINS REGISTERED
600+ NEW VIDEOS
You Tube
25+ HOURS TOTAL DURATION
QUESTIONS ASKED ON THE INTERNET...
100+ Answers.com
40+ YAHOO! ANSWERS
20,000+ NEW POSTS ON tumblr.
13,000+ iPhone APPLICATIONS DOWNLOADED
GO-Globe.com

Nowcasting epidemics

Google

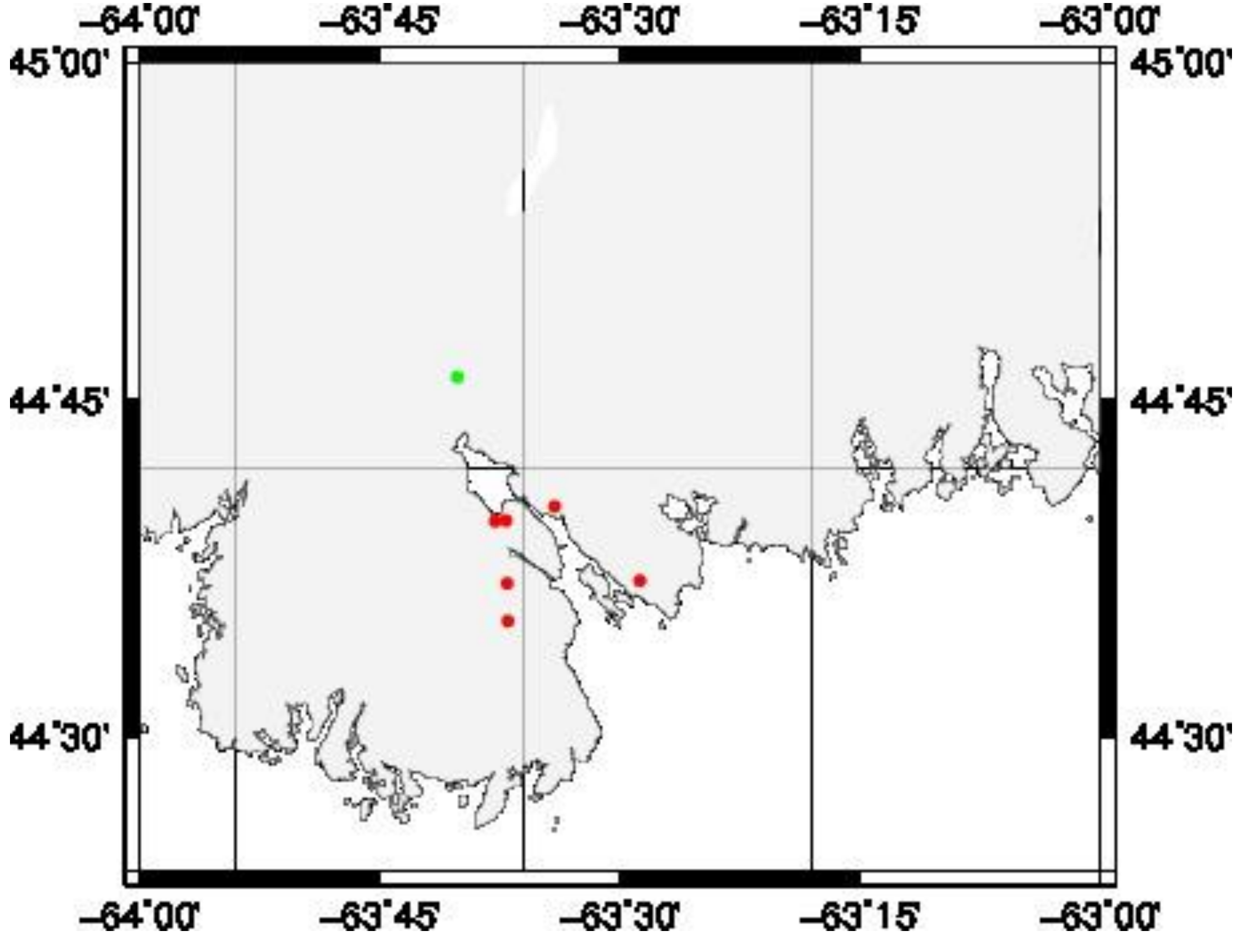
Detecting influenza epidemics using search engine query data

Jeremy Ginsberg[1], Matthew H. Mohebbi[1], Rajan S. Patel[1], Lynnette Brammer[2], Mark S. Smolinski[1] & Larry Brilliant[1]

[1]Google Inc. [2]Centers for Disease Control and Prevention

Nature 457, 1012-1014 (2009)

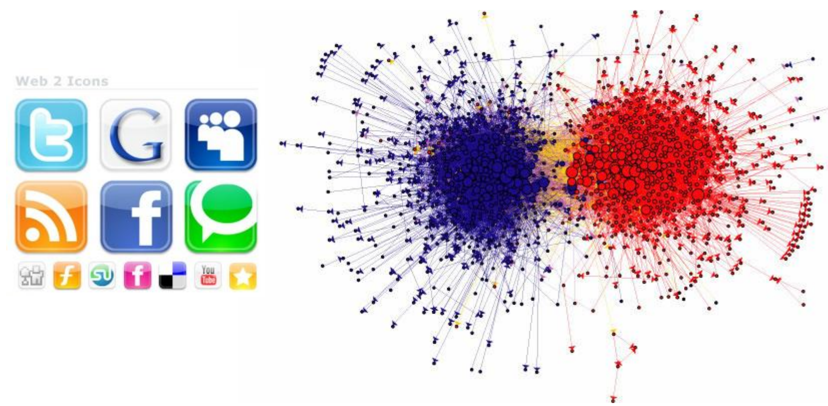
Detecting flu “chat” in twitter



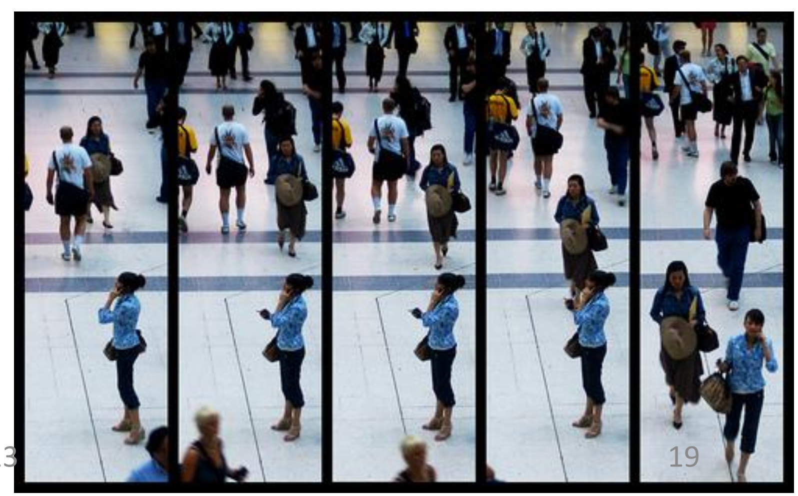
Big data "proxies" of social life

Shopping patterns & lifestyle

Relationships & social ties

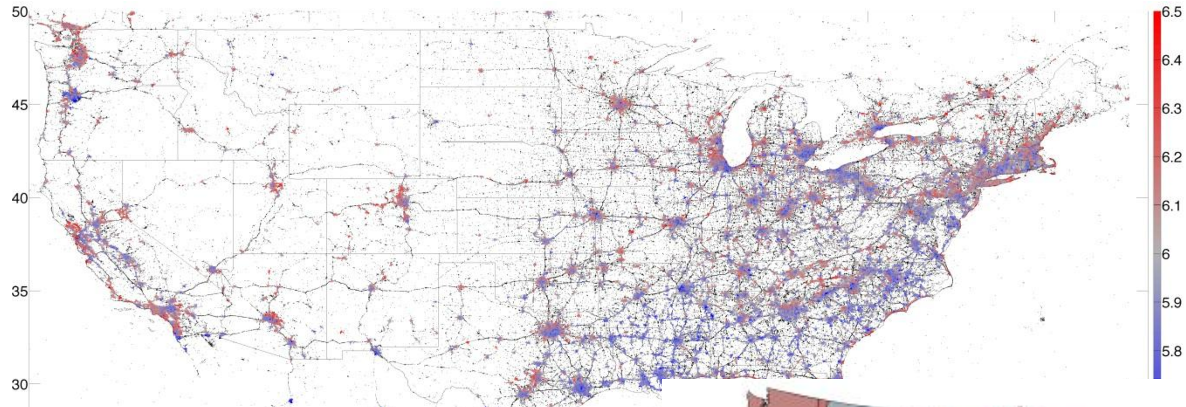


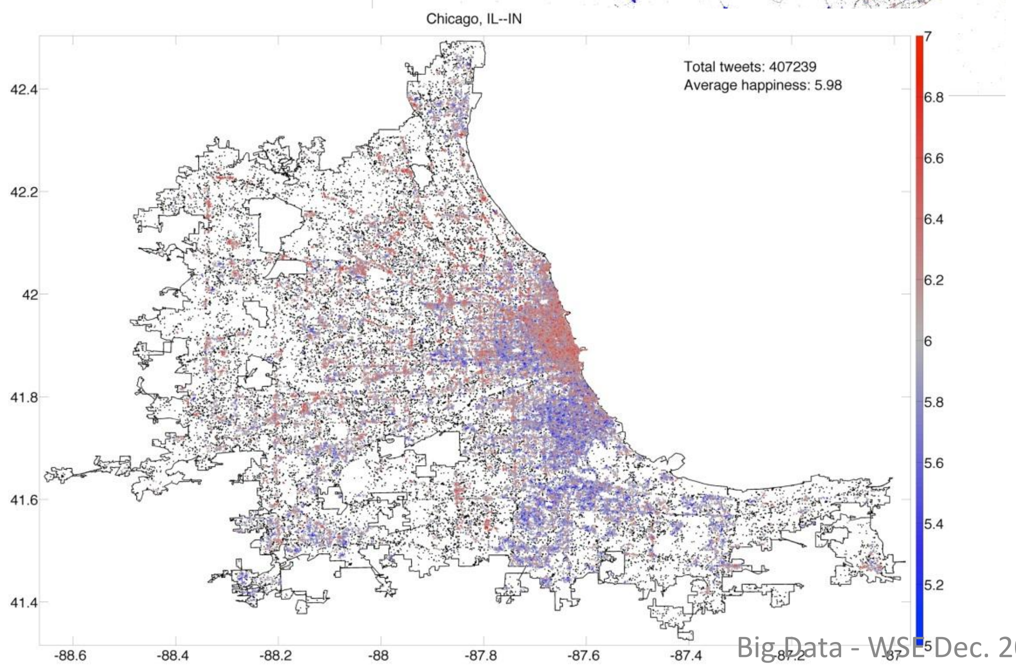
Movements

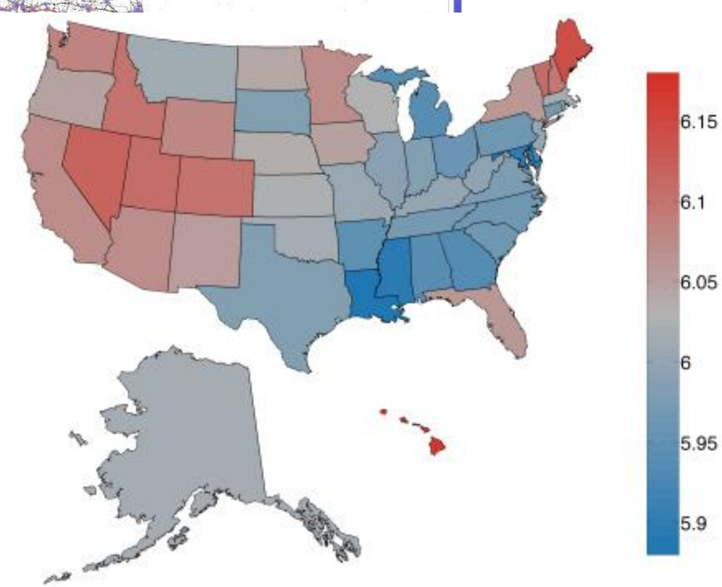


Desires, opinions, sentiments

Where is the happiest city in the US?

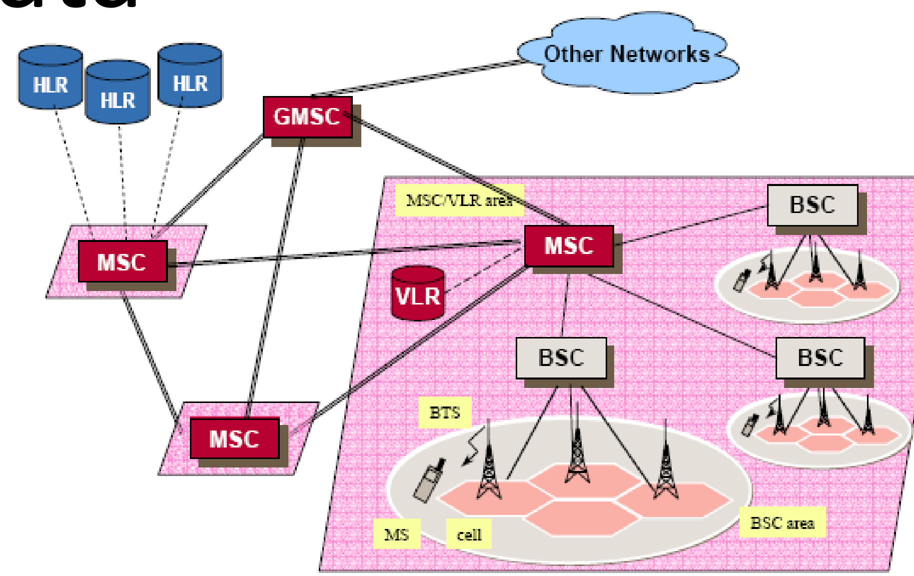


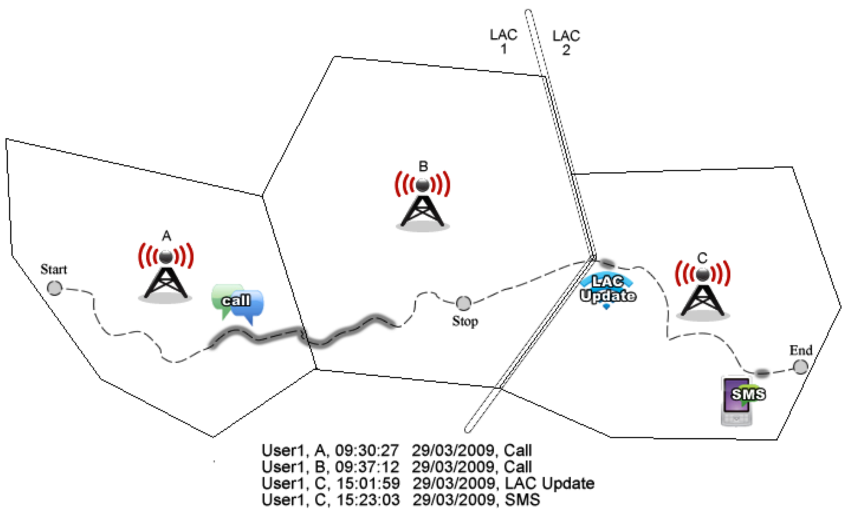




GSM data

- Mobile Cellular Networks handle information about the positioning of mobile terminals
  - CDR Call Data Records: call logs (tower position, time, duration,..)
  - Handover data: time of tower transition
- More sophisticated Network Measurement allow tracking of all active (calling) handsets



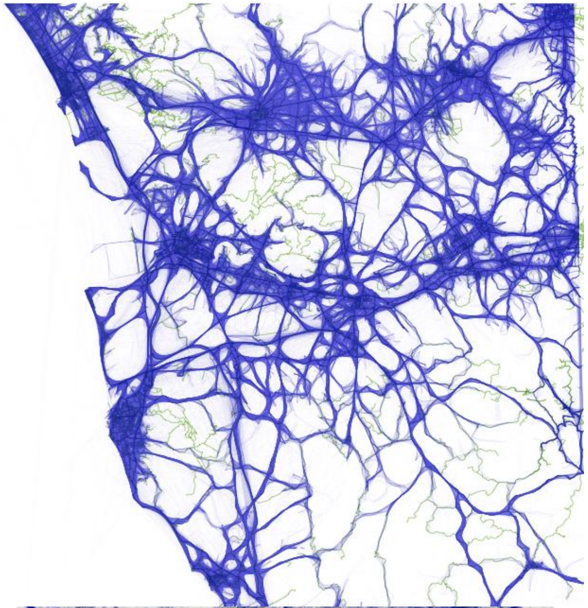


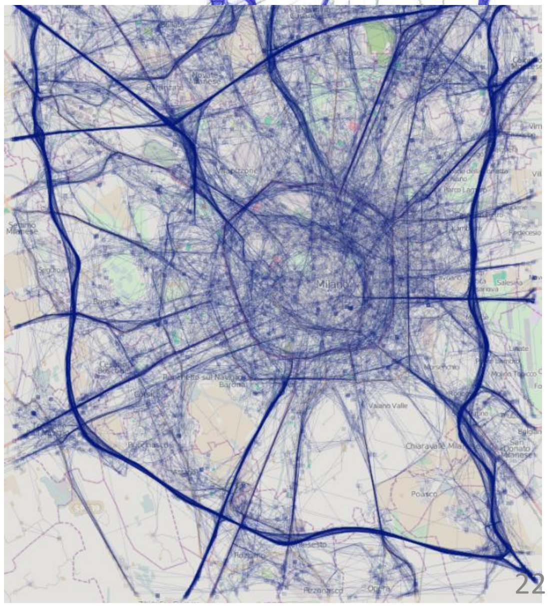
GPS tracks

- Onboard navigation devices send GPS tracks to central servers

```
Ide;Time;Lat;Lon;Height;Course;Speed;PDOP;State;NSat
…
8;22/03/07 08:51:52;50.777132;7.205580; 67.6;345.4;21.817;3.8;1808;4
8;22/03/07 08:51:56;50.777352;7.205435; 68.4;35.6;14.223;3.8;1808;4
8;22/03/07 08:51:59;50.777415;7.205543; 68.3;112.7;25.298;3.8;1808;4
8;22/03/07 08:52:03;50.777317;7.205877; 68.8;119.8;32.447;3.8;1808;4
8;22/03/07 08:52:06;50.777185;7.206202; 68.1;124.1;30.058;3.8;1808;4
8;22/03/07 08:52:09;50.777057;7.206522; 67.9;117.7;34.003;3.8;1808;4
8;22/03/07 08:52:12;50.776925;7.206858; 66.9;117.5;37.151;3.8;1808;4
8;22/03/07 08:52:15;50.776813;7.207263; 67.0;99.2;39.188;3.8;1808;4
8;22/03/07 08:52:18;50.776780;7.207745; 68.8;90.6;41.170;3.8;1808;4
8;22/03/07 08:52:21;50.776803;7.208262; 71.1;82.0;35.058;3.8;1808;4
8;22/03/07 08:52:24;50.776832;7.208682; 68.6;117.1;11.371;3.8;1808;4
…
```

- Sampling rate ~30 secs
- Spatial precision ~ 10 m



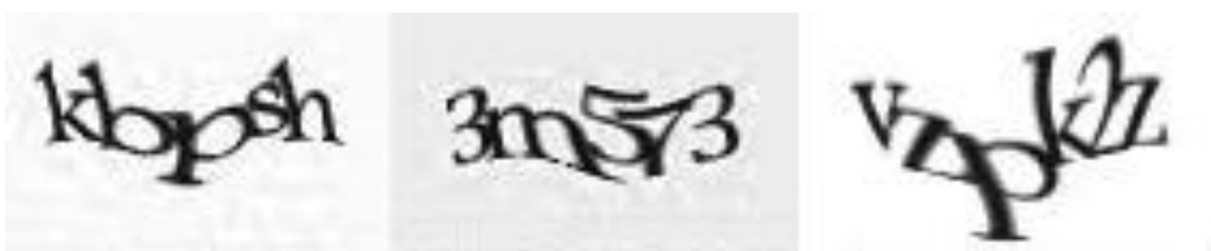


Big data – replacement for knowledge?

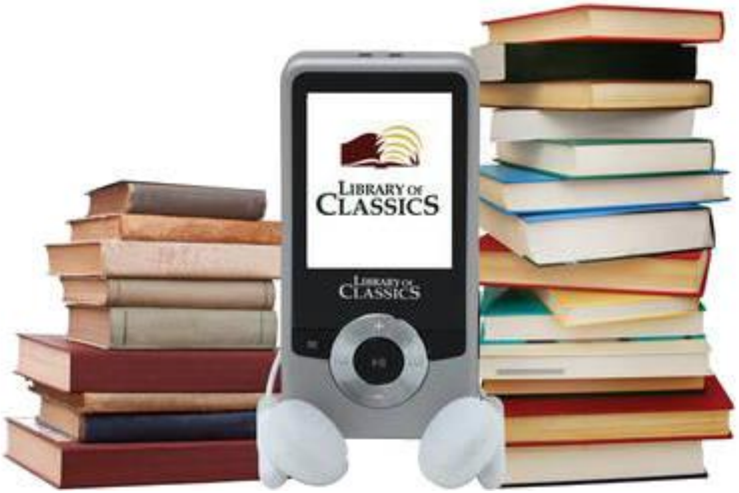
- Google translate: use data in the wild…
- Currently, statistical translation models consist mostly of phrase tables that give candidate mappings between specific source- and target-languages (Norvig 2009)
- **Simple models and more data beat elaborate models based on less data**

- Where does it cease, if at all?

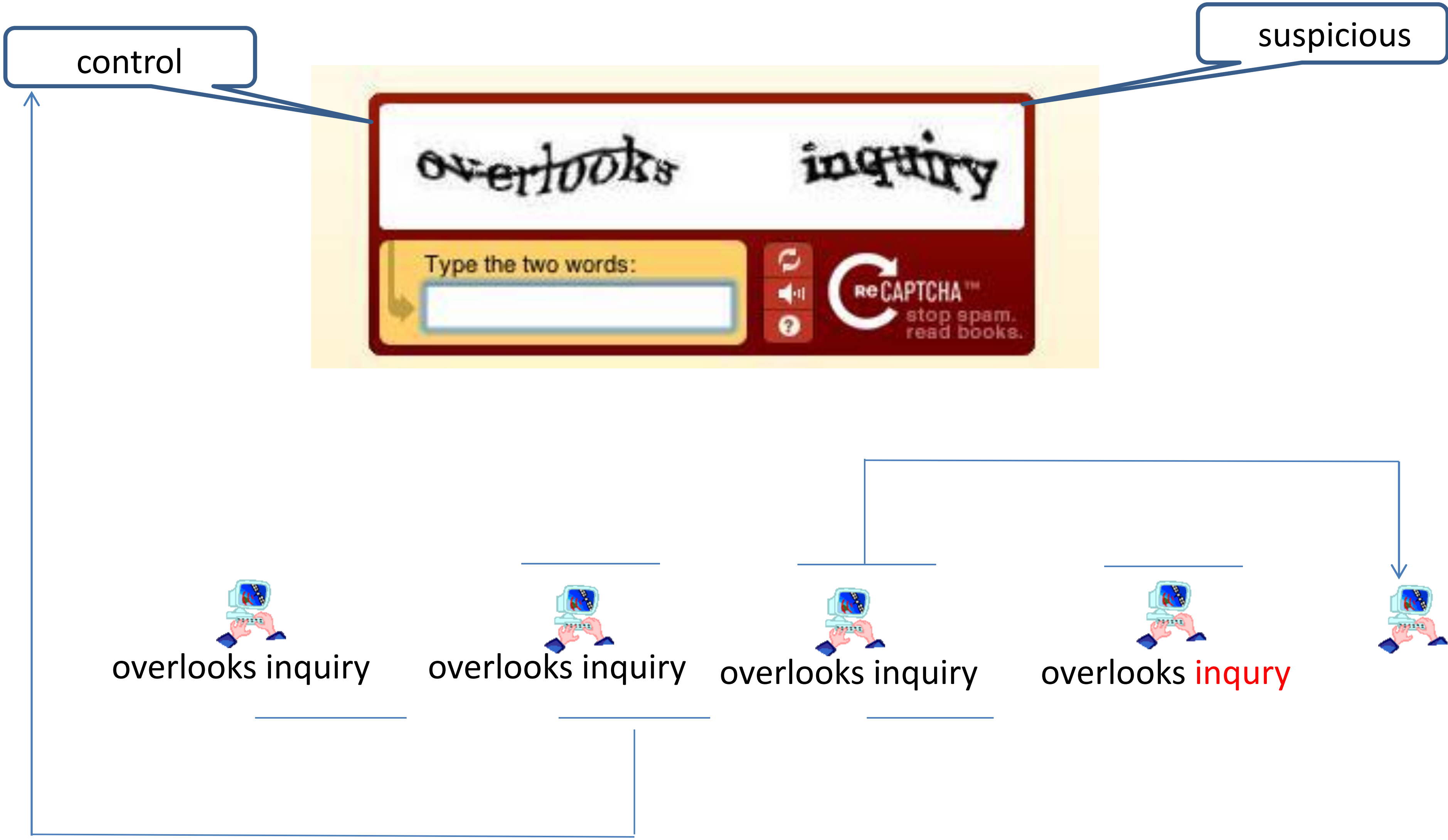
- Crowdsourcing – a solution for “curated” data
- CAPTCHA = Completely Automated Public Turing test to tell Computers and Humans Apart [von Ahn]



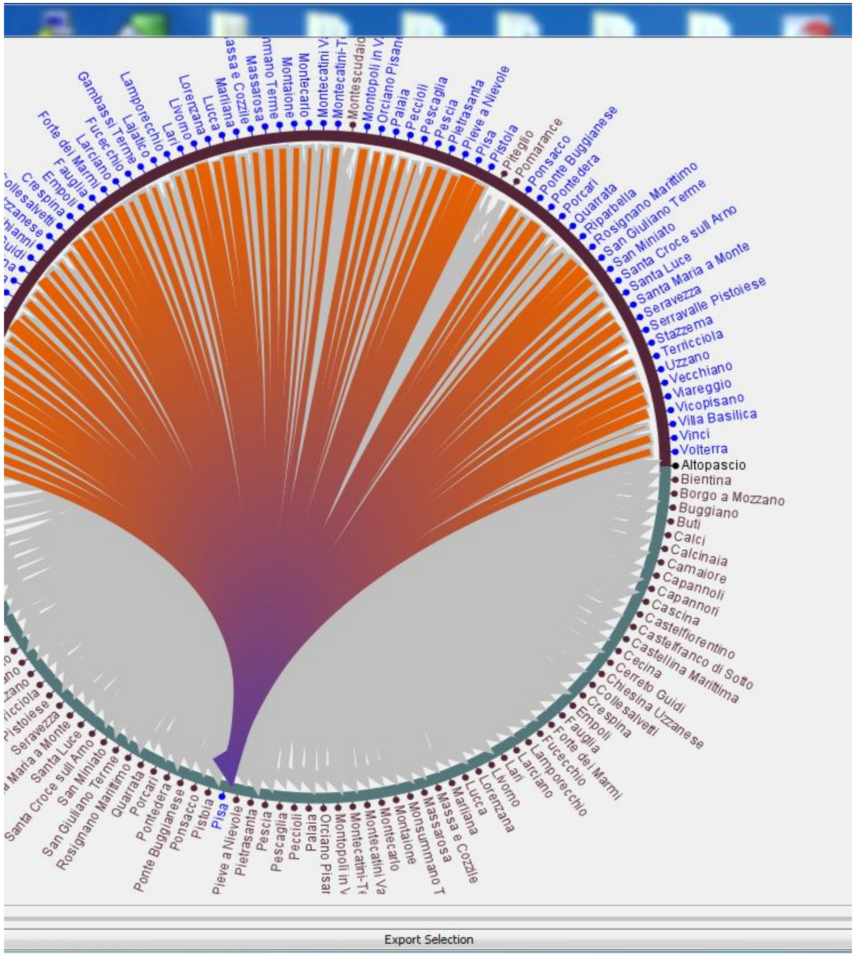
Digital libraries

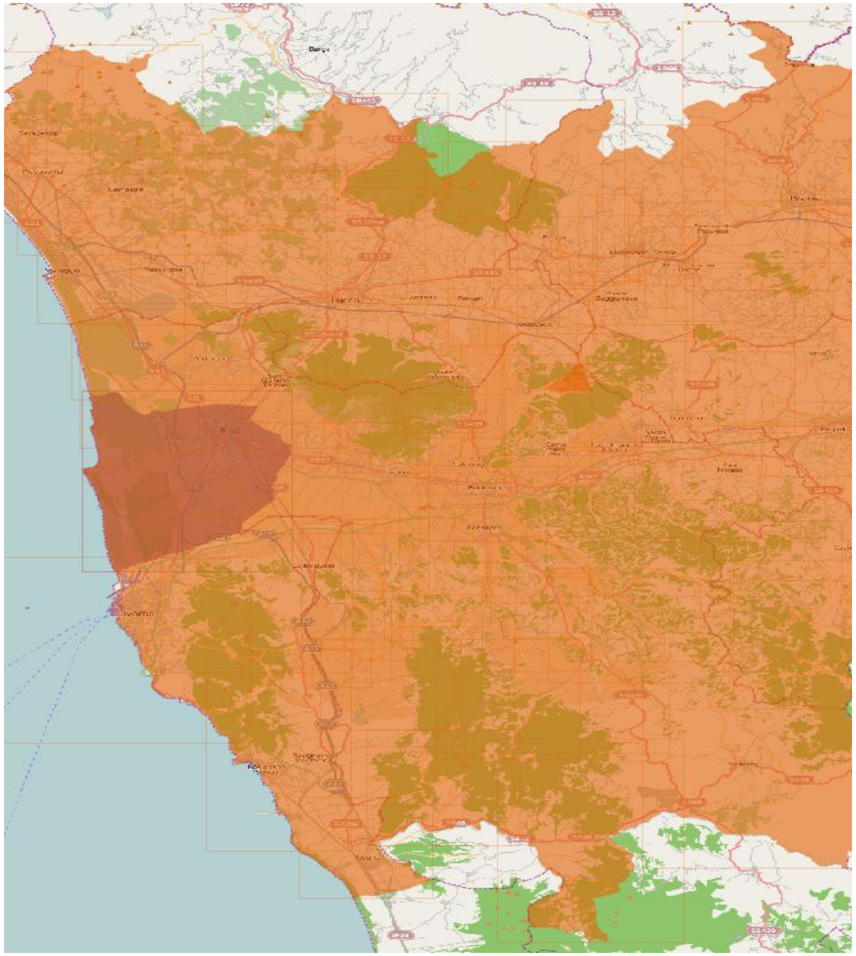


reCAPTCHA

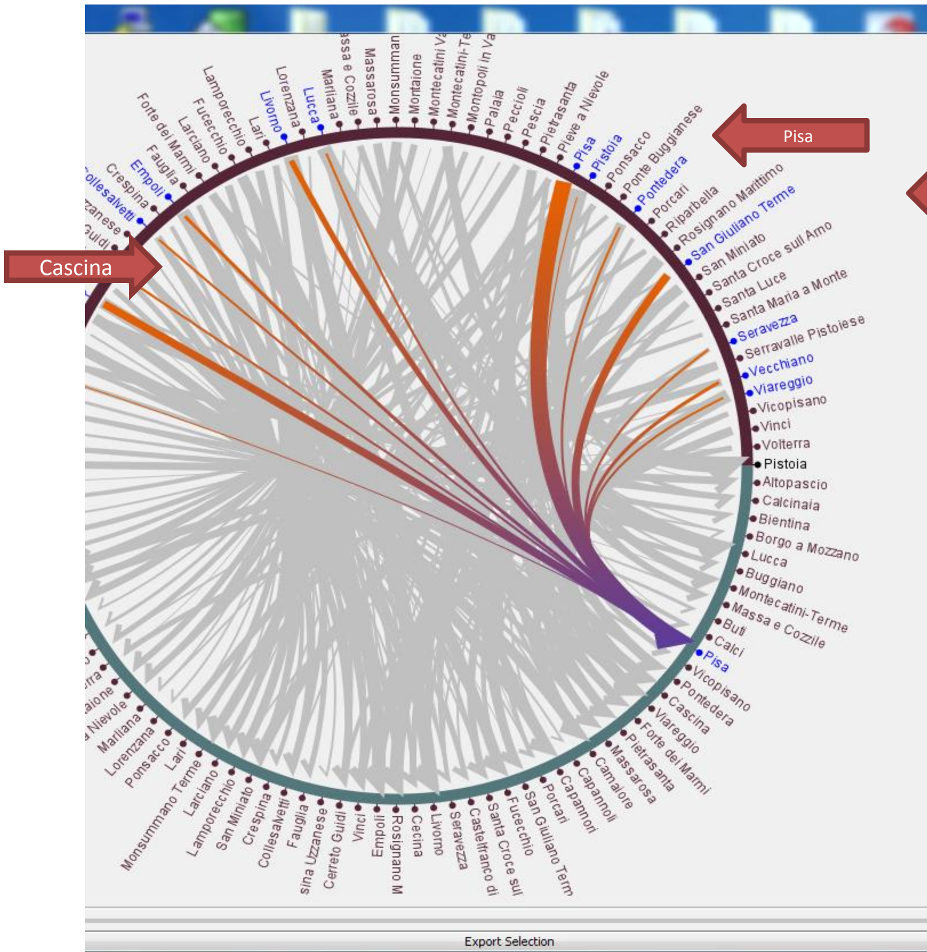

control
suspicious
overlooks inquiry
Type the two words:
reCAPTCHA™
stop spam.
read books.
overlooks inquiry
overlooks inquiry
overlooks inquiry
overlooks inqury

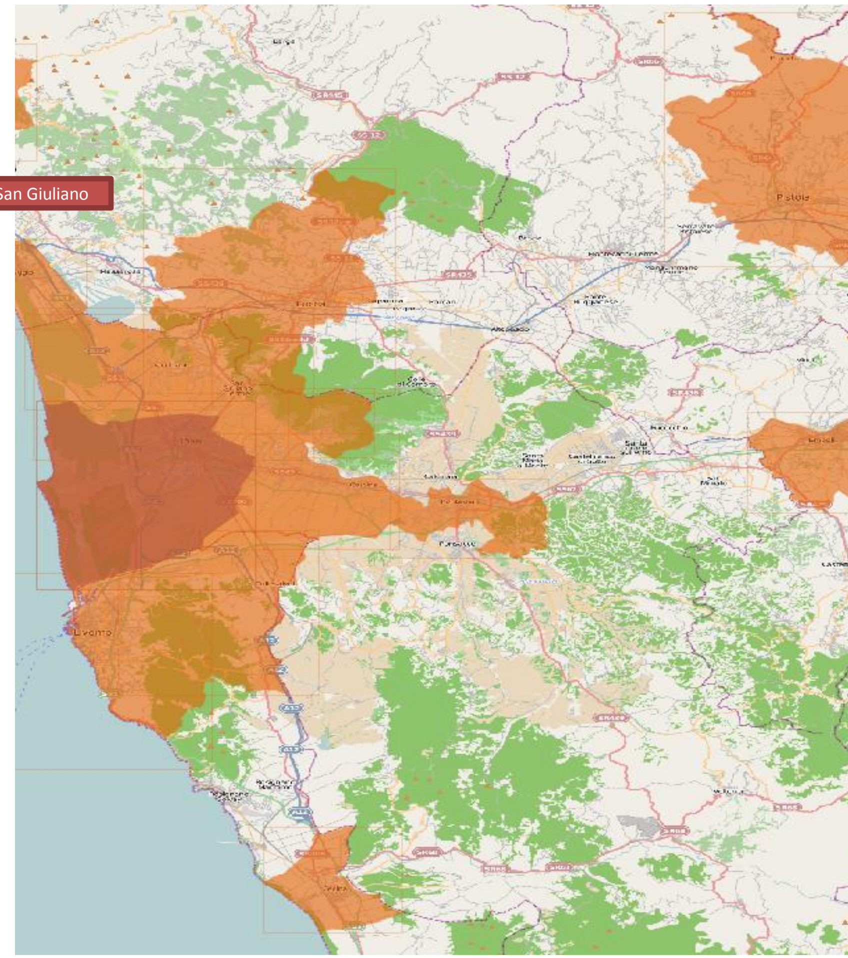
Drill down: from cities to cities





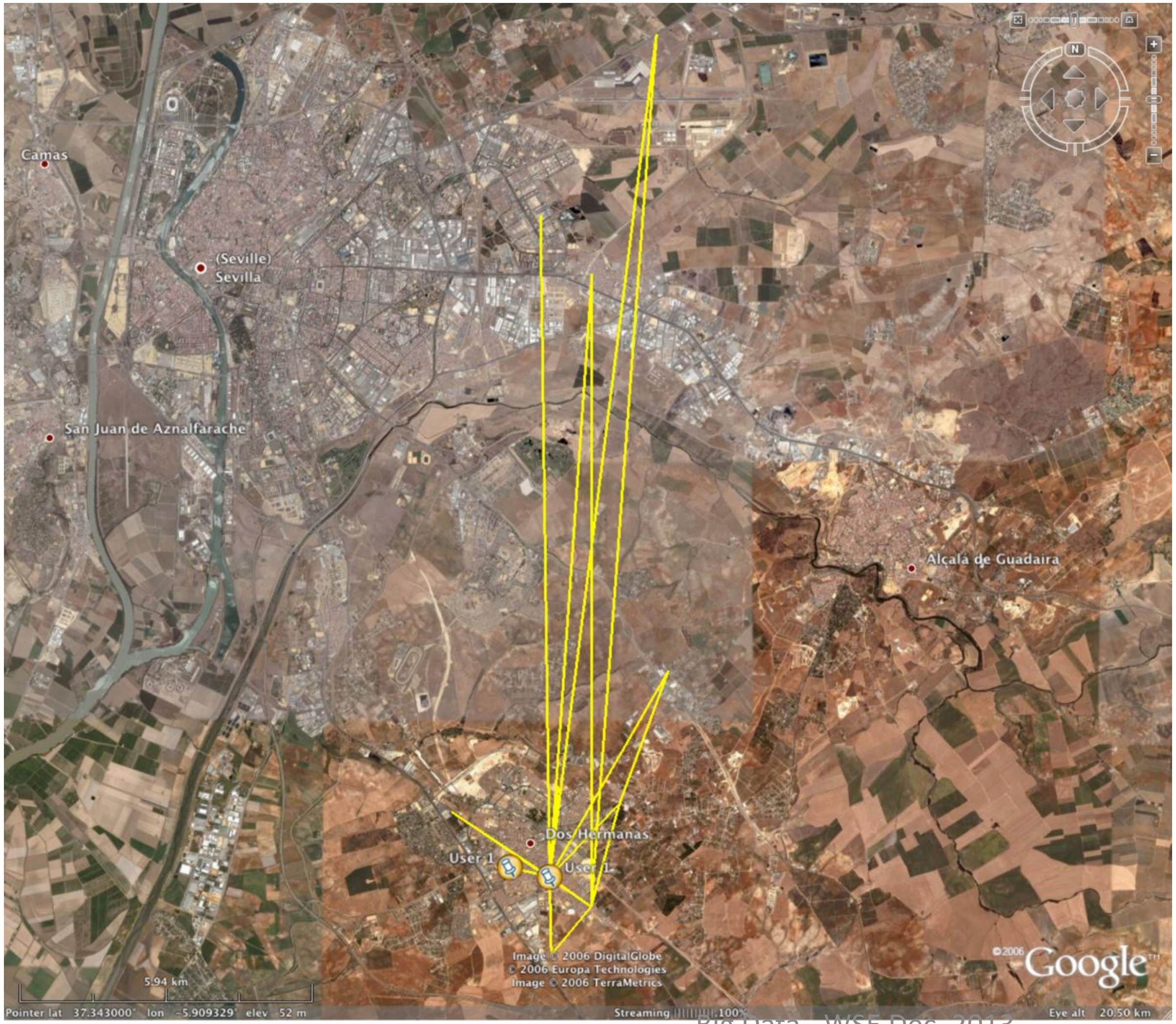
Drill down: from cities to cities (filtered)





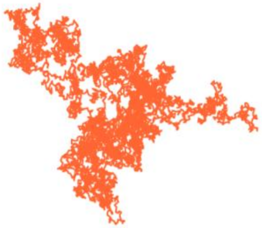
Restrict visualization to flows above a given threshold.
Select specific flows: from Cascina, San Giuliano, and Pisa

Model of human travel?



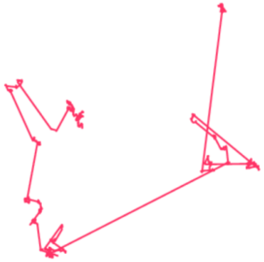
Random Walk

$$f(\Delta r)=C$$



Lévy Flight

$$f(\Delta x) \sim \frac{1}{\Delta x^{1+\beta}}$$



Data-driven Decision Making

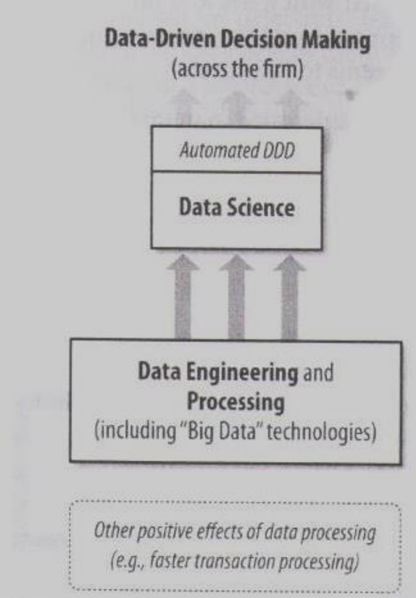


Figure 1-1. Data science in the context of various data-related processes in th organization.

From Provost, Fawcett, "Data Science for Business", O'Reilly 2012

Data science problems

- Finding similarity (similar customers…)
  - Recommender systems
- Predicting things (often done with
    - Classification
    - Probability estimation
    - Regression
    - Link prediction
  - Customers most likely to buy product
  - Length of patient's stay in hospital
  - Churn
  - How much will a customer use the service?
- Exploring things
    - Association rules – co-occurrence - Market Basket Analysis
    - Clustering – groups - what are customer types?

Data science problems

- Explaining things
  - Profiling: pattern of movement for a fishing ship? Or: what is the typical cellphone use of this customer segment?
  - what factors cause churn?
- Causal modelling
  - Randomized experiments
- Data science projects ***are not like*** IT projects, but like ***R&D*** projects

Modeling

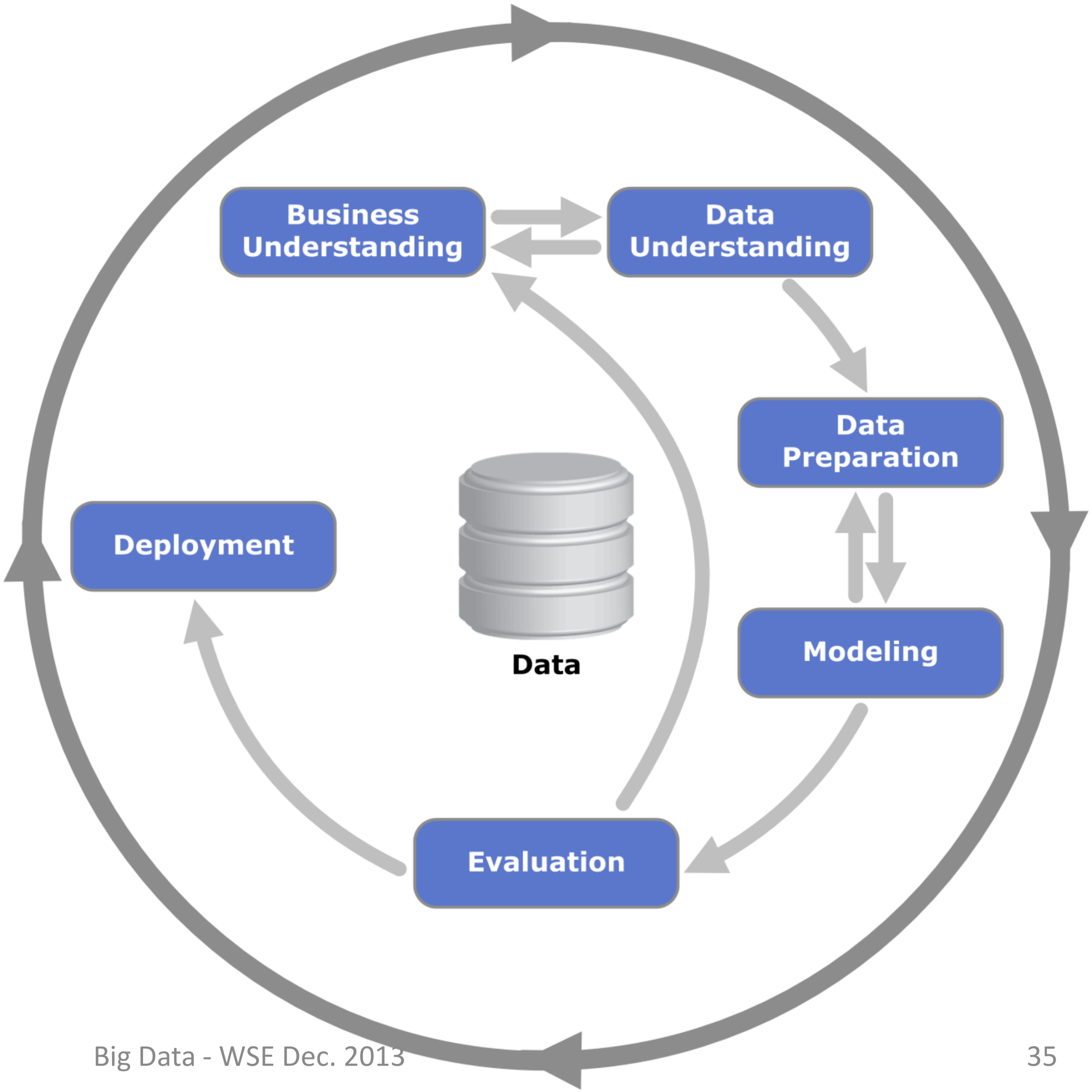
- Classification and class probability estimation
- Regression
- Similarity matching
- Clustering
- Co-occurrence grouping
- Profiling
- Link prediction
- Data reduction
- Causal modeling

Supervised *vs* unsupervised tasks

- Can we find groups of customers wrt their transactional behavior?
- Can we have groups of customers with particularly high liklelihood of cancelling the service after their contract expires (*CHURN*)
- There must be ***data*** for supervised tasks

Data mining

The CRISP (Cross Industry Standard Process for Data Mining) model



Business understanding

- Key part of the process
- Mapping the business problem into a data and data mining problem
- Think of use scenarions

Data understanding

- What data is available?
- What is the cost of the data?
  - CC fraud *vs* insurance fraud

Data preparation

- Conversions
- Mapping to tabular format
- “attribute engineering”
- Data “leaks” – from historical data to target variable

Modeling

- …coming soon…

Evaluation

- What is the right measure?
  - Accuracy
  - MSE
  - AUC
  - …application-dependent…
- Cross-validation for temporal data and data leak

Deployment

- Depends on the business problem
- Often involves recoding
- Who does it?

Select scalable modeling techniques

- Decision trees
- Random forest
- Bayesian

Classification: a definition

- Data are given as vectors of attribute values, where the domain of possible values for attribute *j* is denoted as *Aj*, for 1 *<= j <= N*. Moreover,a set *C* = {*c*1,., *ck*} of *k* classes is given; this can be seen as a special attribute or label for each record. Often *k* = 2, in which case we are learning a binary classifier.
- Inducing, or learning a classifier, means finding a mapping

  $F: A_1 \times A_2 \times A_N \rightarrow C,$

  given a finite <u>training</u> <u>set</u> $X1 = \{<x_{ij}, c_i>, 1 <= j <= N, c_i \in C, 1 <= i <= M\}$ of M labeled examples [comment on noise]

- We assume that data is represented as fixed size vectors of attributes (AVL representation): eg all patients are represented by the same 38 attributes, perhaps in conceptual groupings into personal, social, medical
- F belongs to a fixed language, e.g. F can be
  - a set of n -1 dimensional hyperplanes partitioning an n-dimensional space into k subspaces, or
  - a decision tree with leaves belonging to C, or
  - a set of rules with consequents in C.
- We also want F to perform well, in terms of its predictive power on (future) data not belonging to X1 [predictive power]

- In data base terminology, we “model” one relation
- There are methods that deal with multi-relational representations (multiple tables), - multi-relational learning AKA Inductive Logic Programming

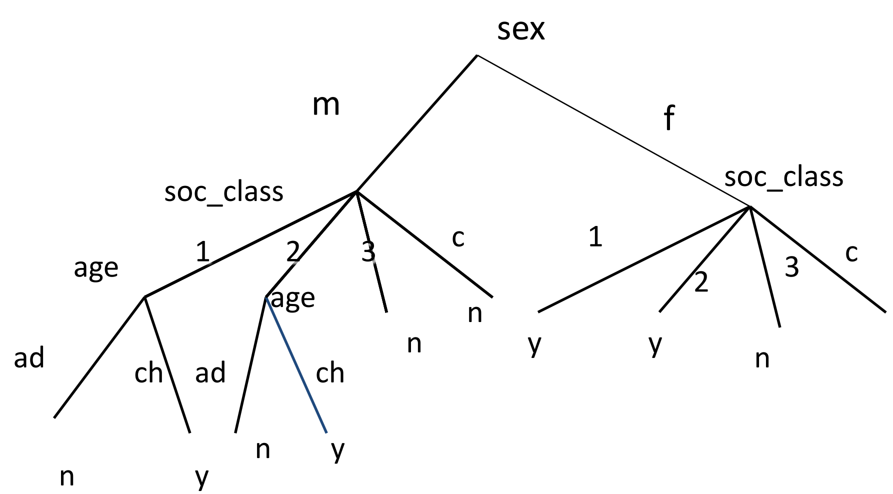
Example 2: Who would survive Titanic's sinking

- Predict whether a person on board would have survived the tragic sinking
- Classification: yes (survives), no (does not survive)
- Data:The data is already collected and labeled for all 2201 people on board the Titanic.

Example 2: Representation for the Titanic Survivor Prediction

- Each example records the following *attributes*
- social class {first class,second class, third class, crew member}
- age {adult, child}
- sex {male, female}
- survived {yes, no}

Titanic Survivor model



y

Induction of decision trees: an algorithm building a DT from data…

building a *univariate (single attribute is tested)* decision tree from a set T of training cases for a concept C with classes C_1,…C_k

Consider three possibilities:

- **T contains 1 or more cases all belonging to the same class C_j. The decision tree for T is a leaf identifying class C_j**
- **T contains no cases. The tree is a leaf, but the label is assigned heuristically, e.g. the majority class in the parent of this node**

- T contains cases from different classes. T is divided into subsets that seem to lead towards collections of cases. A test t based on a single attribute is chosen, and it partitions T into subsets $\{T_1,\ldots,T_n\}$. The decision tree consists of a decision node identifying the tested attribute, and one branch for ea. outcome of the test. Then, the same process is applied recursively to ea.T_i

Choosing the test

- why not explore all possible trees and choose the simplest (Occam's razor)? But this is an NP complete problem. E.g. in the 'Titanic' example there are millions of trees consistent with the data

- idea: to choose an attribute that best separates the examples according to their class label
- This means to maximize the difference between the info needed to identify a class of an example in T, and the same info after T has been partitioned in accordance with a test X
- Entropy is a measure from information theory [Shannon] that measures the quantity of information

- information measure (in bits) of a message is -$\log_2$ of the probability of that message
- notation: S: set of the training examples; freq(C_i, S) = number of examples in S that belong to C_i;

selecting 1 case and announcing its class has info measure - $\log_2(\text{freq}(C_i, S)/|S|)$ bits

to find information pertaining to class membership in all classes:

$$\text{info}(S) = -\sum_1 (\text{freq}(C_i, S)/|S|)*\log_2(\text{freq}(C_i, S)/|S|)$$

after partitioning according to outcome of test X:

$$\text{info}_X(T) = \sum |T_i|/|T|*\text{info}(T_i)$$

$\text{gain}(X) = \text{info}(T) - \text{info}_X(T)$ measures the gain from partitioning T according to X

We select X to maximize this gain

- The basic idea in evaluating classifier performance is to count how many times the classifier is correct and incorrect when applied on the testing set.
- This is nicely represented in a *confusion matrix*

label	assigned=T	assigned=F
true=T	TP	FN
true=F	FP	TN

- The most common measure of classifier performance is accuracy ACC$= \frac{TP+TN}{N}$ or its complement error rate $=$ $1-$ACC$= 1 - \frac{TP+TN}{N} = \frac{FN+FP}{N}$

Computing accuracy: in practice

– partition the set E of all *labeled* examples (examples with their classification labels) into a *training set* X1 and a *testing (validation) set X2.* Normally, X1 and X2 are disjoint

– use the training set for learning, obtain a hypothesis H, set acc := 0

– for ea. element t of the testing set,

  apply H on t; if H(t) = label(t) then acc := acc+1

– acc := acc/|testing set|

Testing - cont'd

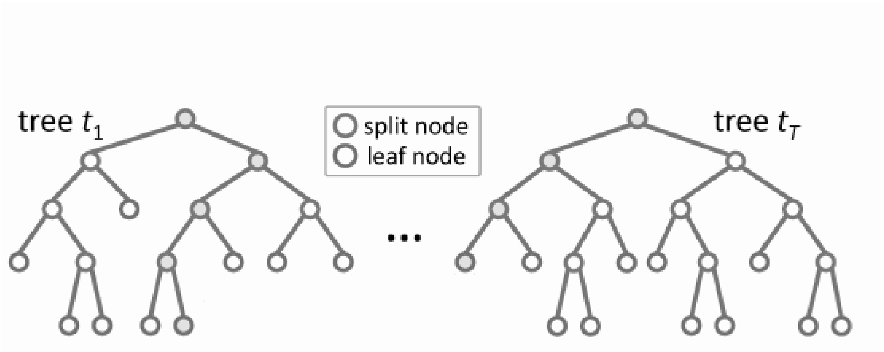
- Given a dataset, how do we split it between the training set and the test set?
- cross-validation (n-fold)
  - partition E into n groups
  - choose n-1 groups from n, perform learning on their union
  - repeat the choice n times
  - average the n results
  - usually, n = 3, 5, 10
- another approach - learn on all but one example, test that example.

  "Leave One Out"

MSE

- Mean Square Error – a measure appropriate for
  - Binary setting (two classes)
  - Numerical predictions (regression)

Random Forests (from Zhuowen Tu, UCLA)



- Random forests (RF) are a combination of tree predictors
- Each tree depends on the values of a random vector sampled independently
- The generalization error depends on the strength of the individual trees and the correlation between them
- Using a random selection of features yields results favorable to AdaBoost, and are more robust w.r.t. noise

The Random Forest Algorithm

Given a training set S

For i = 1 to k do:

  Build subset Si by sampling with replacement from S

  Learn tree Ti from Si

    At each node:

      Choose best split from random subset of F features

    Each tree grows to the largest extend, and no pruning

Make predictions according to majority vote of the set of k trees.

from Zhuowen Tu, UCLA

Features of Random Forests

- It is unexcelled in accuracy among current algorithms.
- It runs efficiently on large data bases.
- It can handle thousands of input variables without variable deletion.
- It gives estimates of what variables are important in the classification.
- It generates an internal unbiased estimate of the generalization error as the forest building progresses.
- It has an effective method for estimating missing data and maintains accuracy when a large proportion of the data are missing.
- It has methods for balancing error in unbalanced data sets.

from Zhuowen Tu, UCLA

Bayesian learning

- Highly scalable
- Applicable to BD

Bayesian learning

- incremental, noise-resistant method
- can combine prior Knowledge (the K is probabilistic)
- predictions are probabilistic

Courtesy of Eammon Keogh, UCR,
eamonn@cs.ucr.edu

Naïve Bayes Classifier

Thomas Bayes
1702 - 1761

Let us start with an example of “Bayesian inference”:…

Bayes' law of conditional probability:

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

results in a simple "learning rule": choose
the most likely (Maximum APosteriori)hypothesis

$$h_{MAP} = \arg\max_{h \in H} P(D|h)P(h)$$

Example:
Two hypo:
(1) the patient has cancer
(2) the patient is healthy

Priors: 0.8% of the population has cancer;

$\oplus$ is 98% reliable: it returns positive in 98% of cases when the the disease is present, and returns 97% negative when the disease is actually absent.

P(cancer) = .008

P(+ |cancer) = .98

P(+|not cancer) = .03

P(not cancer) = .992

P(- |cancer) = .02

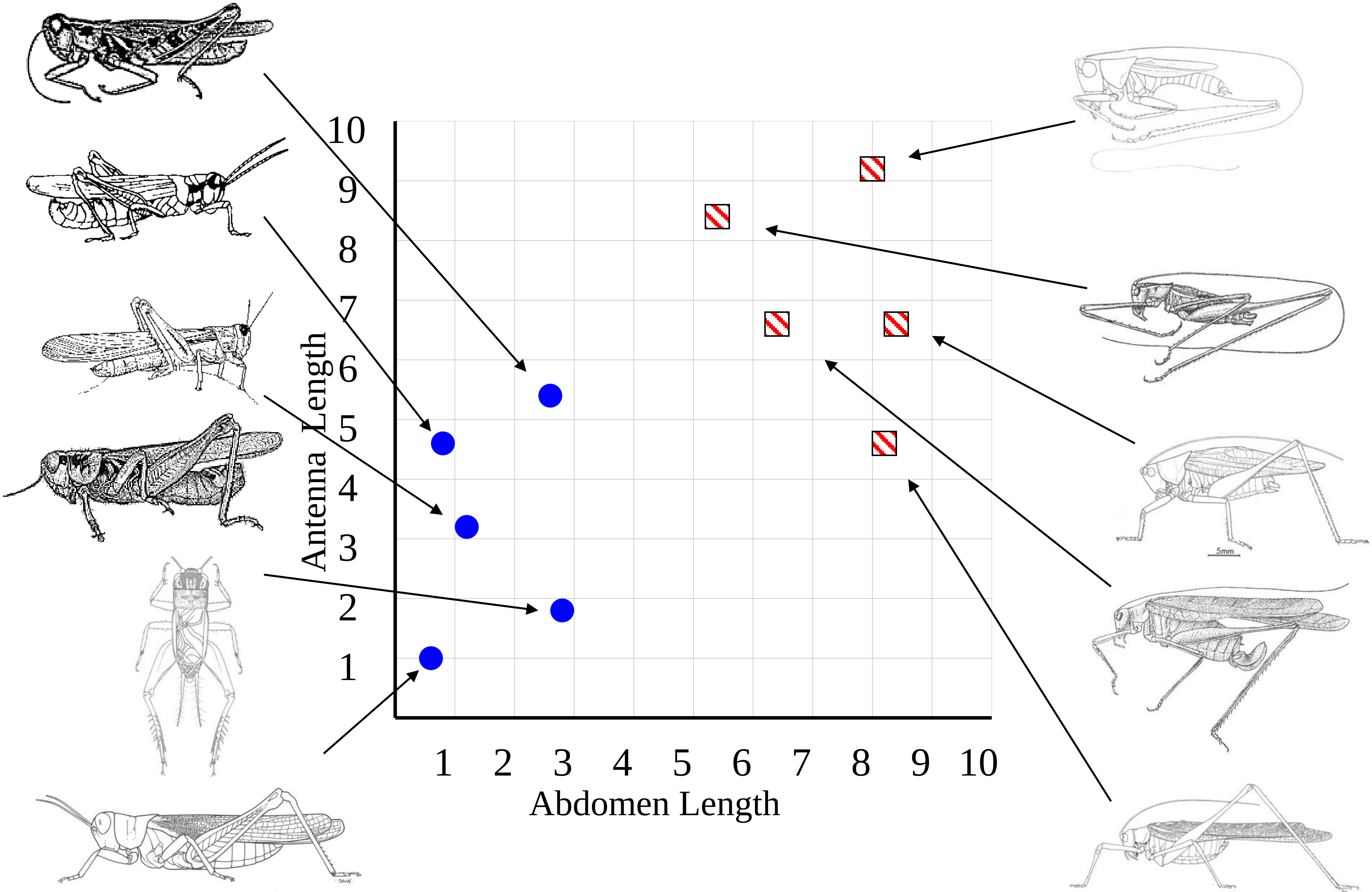
P(-|not cancer) = .97

We observe a new patient with a positive test.
How should they be diagnosed?

P(cancer|+) = P(+|cancer)P(cancer) = .98*.008 = .0078
P(not cancer|+) = P(+|not cancer)P(not cancer) = .03*.992=.0298

Grasshoppers

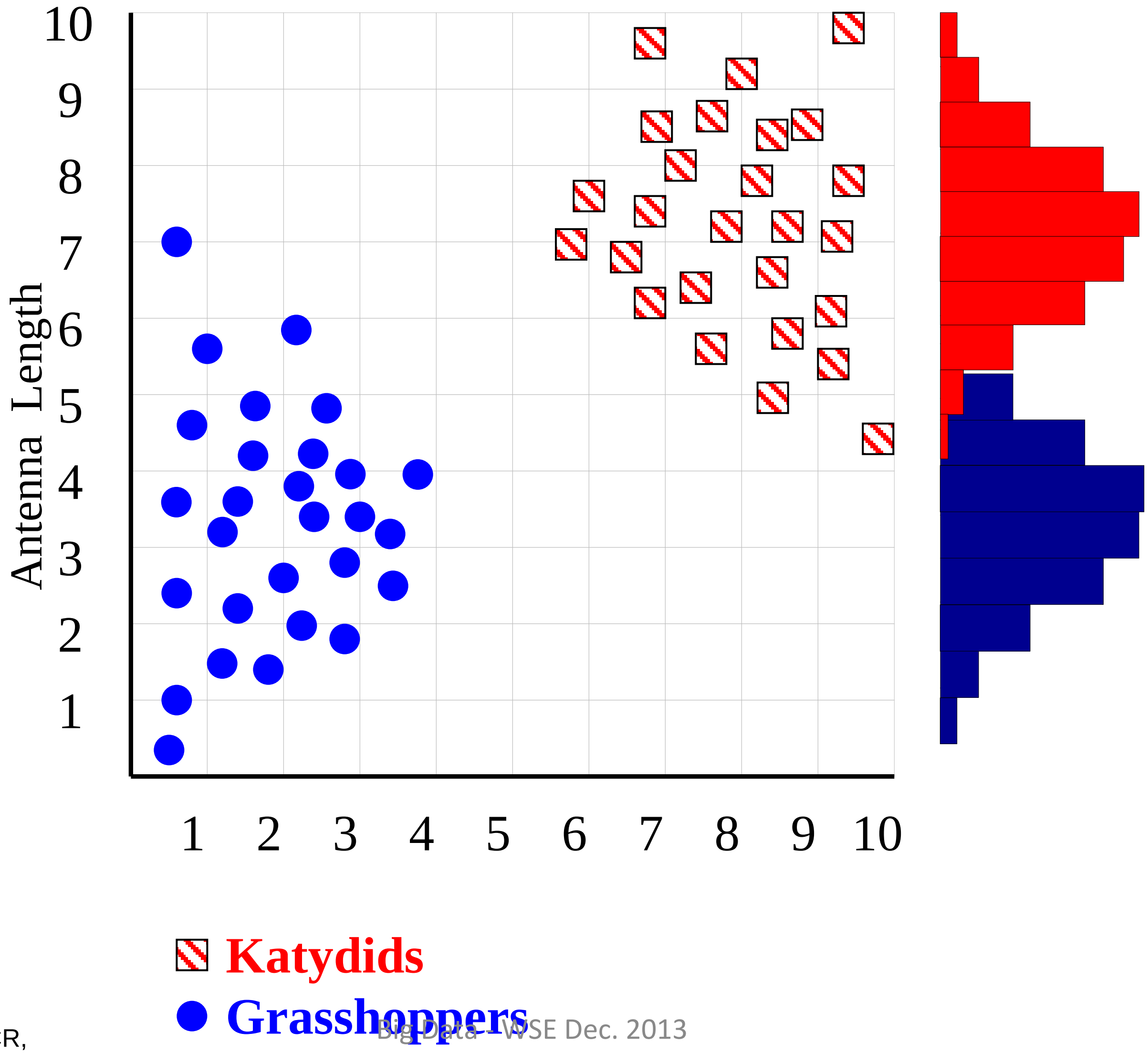
Katydids



Naïve Bayes classifier: the very foundation

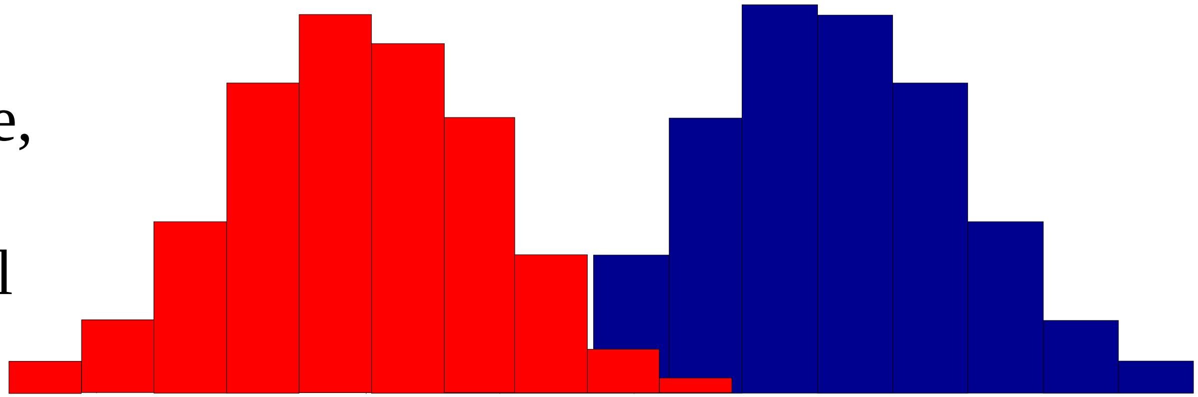
Courtesy of Eammon Keogh, UCR, eamonn@cs.ucr.edu

With a lot of data, we can build a histogram. Let us just build one for "Antenna Length" for now…

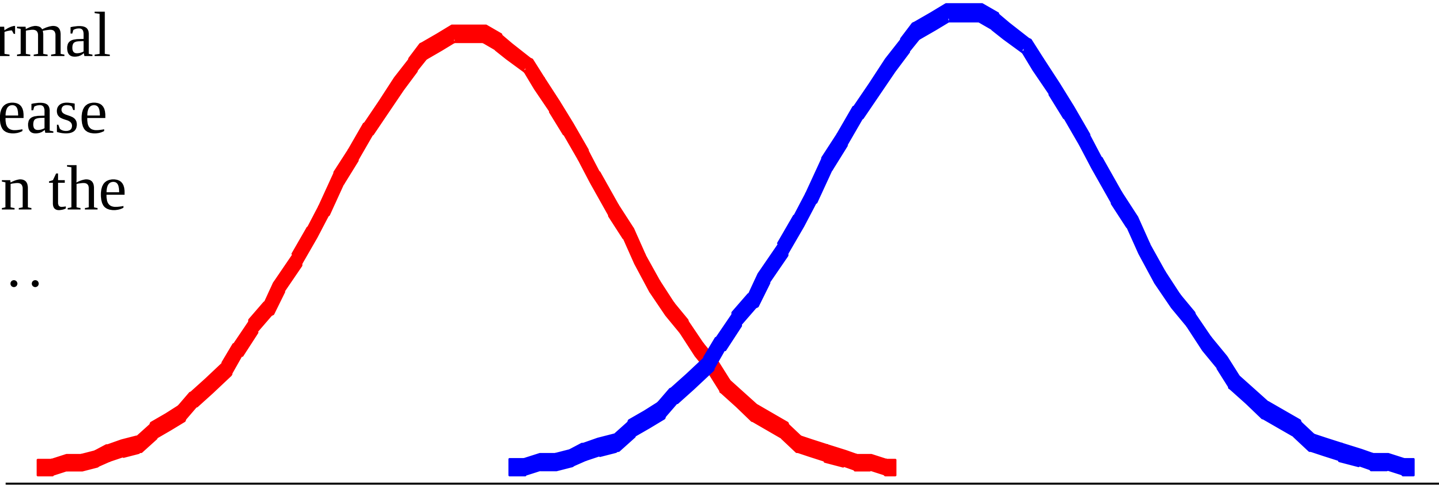


Courtesy of Eammon Keogh, UCR, eamonn@cs.ucr.edu

We can leave the histograms as they are, or we can summarize them with two normal distributions.



Let us us two normal distributions for ease of visualization in the following slides…



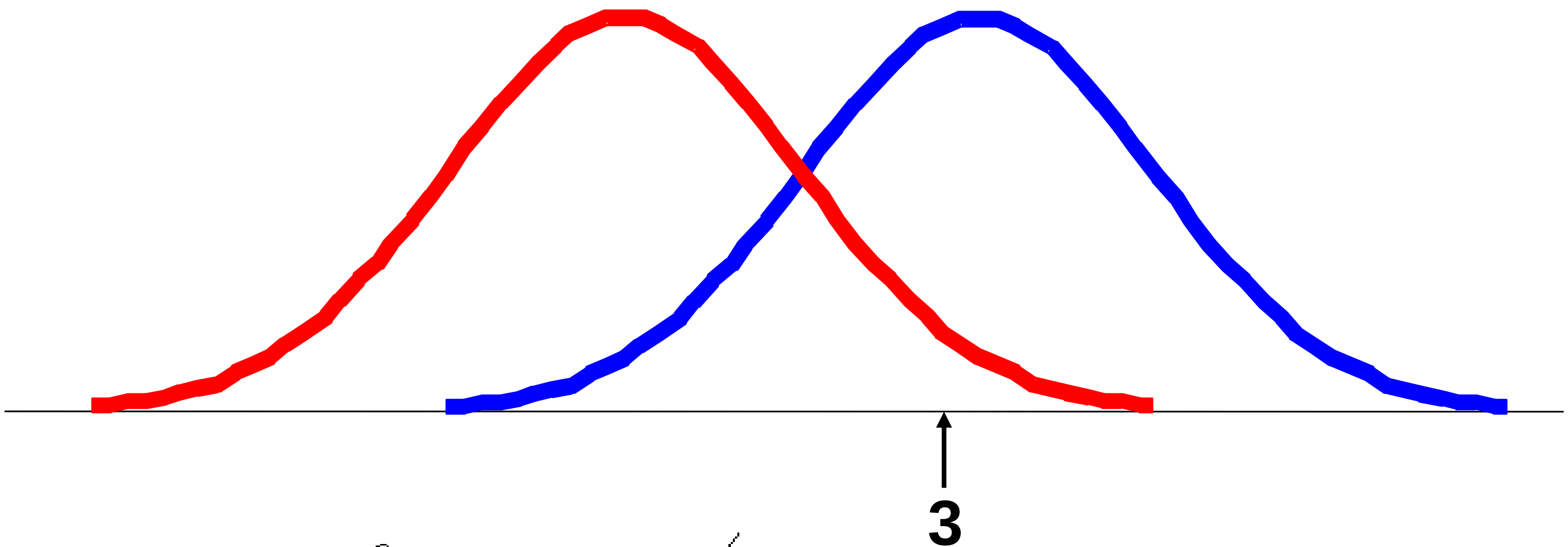
Courtesy of Eammon Keogh, UCR, eamonn@cs.ucr.edu

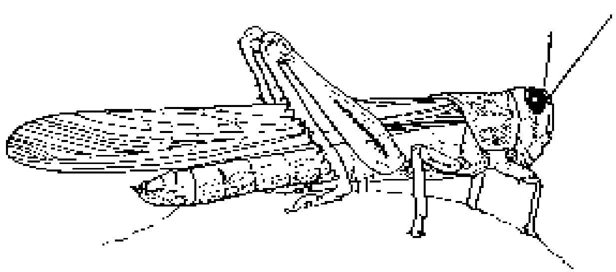
• We want to classify an insect we have found. Its antennae are 3 units long. How can we classify it?

• We can just ask ourselves, give the distributions of antennae lengths we have seen, is it more *probable* that our insect is a **Grasshopper** or a **Katydid**.

• There is a formal way to discuss the most *probable* classification…

$p(c_j \mid d)$ = probability of class c_j, *given* that we have observed d





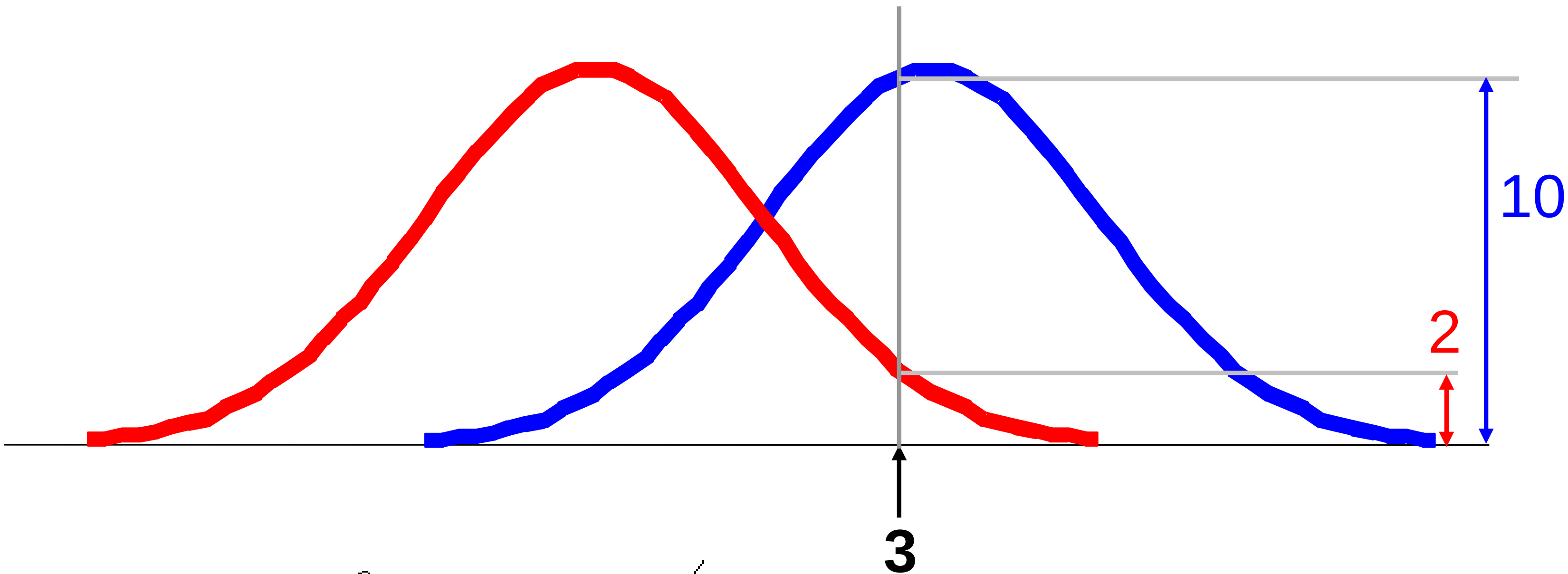
Antennae length is **3**

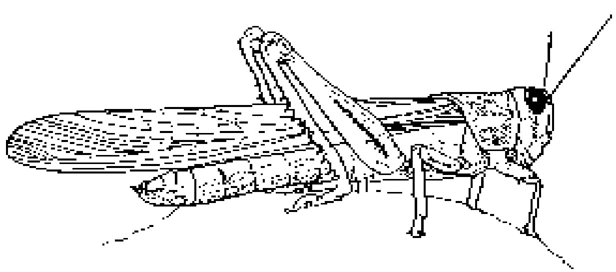
Courtesy of Eammon Keogh, UCR, eamonn@cs.ucr.edu

$p(c_j \mid d)$ = probability of class c_j, given that we have observed d

P(**Grasshopper** | **3**) = 10 / (10 + 2) = 0.833

P(**Katydid** | **3**) = 2 / (10 + 2) = 0.166





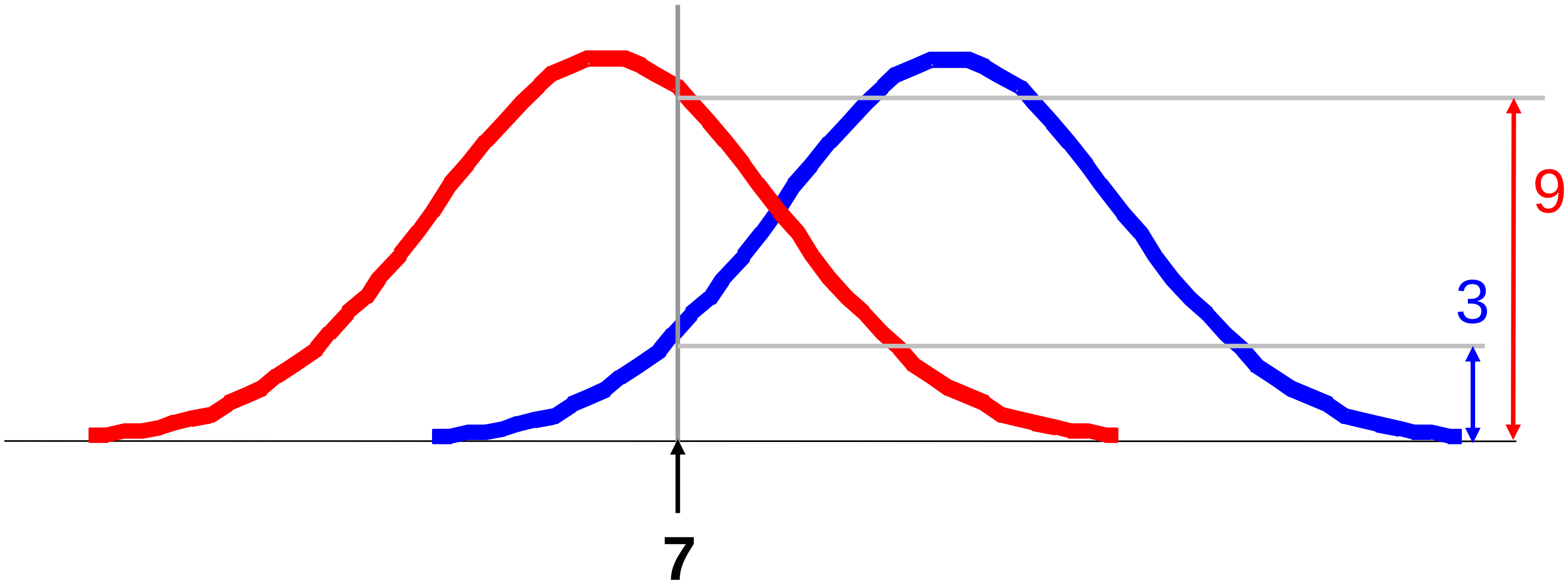
Antennae length is **3**

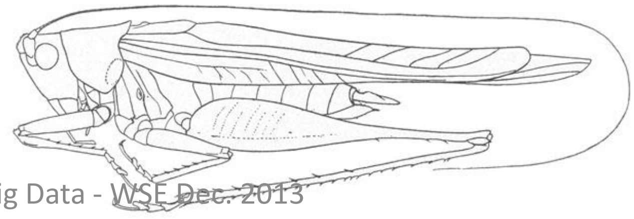
Courtesy of Eammon Keogh, UCR, eamonn@cs.ucr.edu

$p(c_j \mid d)$ = probability of class c_j, given that we have observed d

P(**Grasshopper** | **7**) = 3 / (3 + 9) = 0.250

P(**Katydid** | **7**) = 9 / (3 + 9) = 0.750





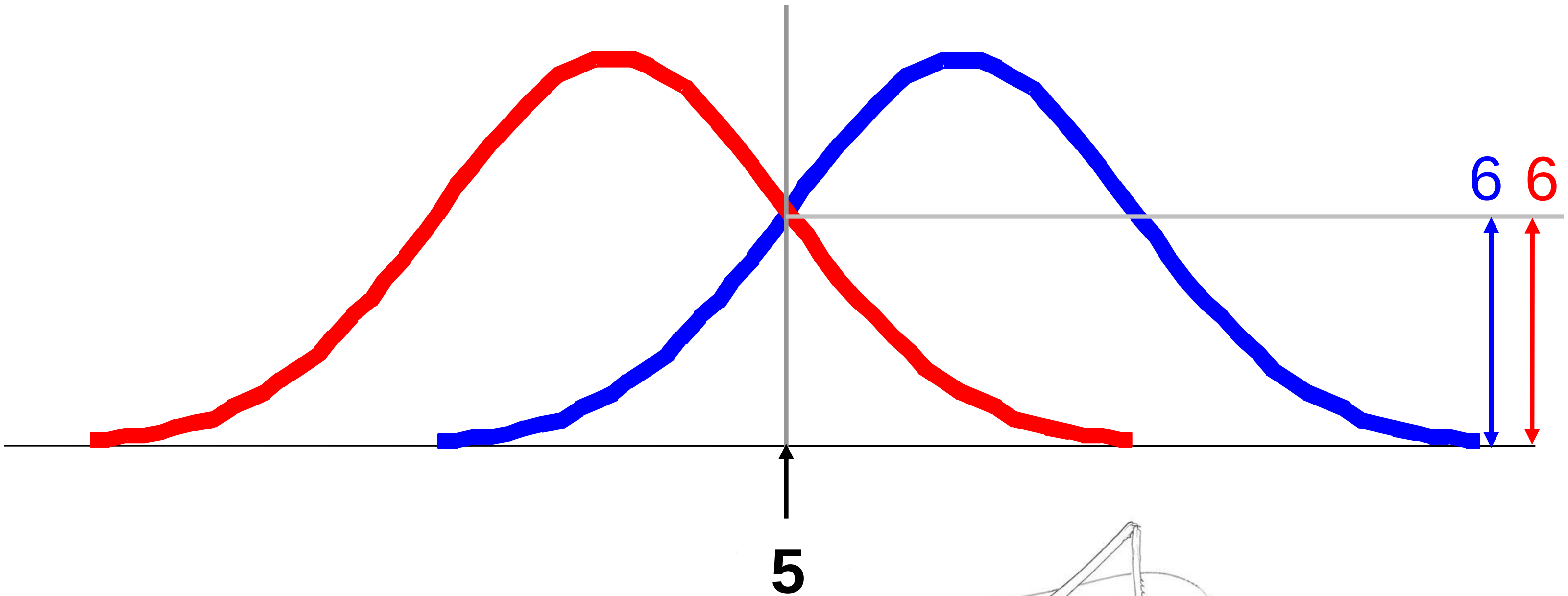
Antennae length is **7**

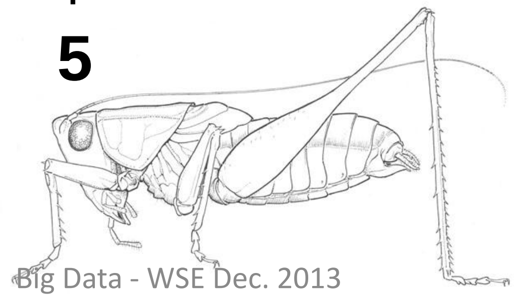
Courtesy of Eammon Keogh, UCR, eamonn@cs.ucr.edu

$p(c_j \mid d)$ = probability of class c_j, given that we have observed d

P(**Grasshopper** | **5**) = 6 / (6 + 6) = 0.500

P(**Katydid** | **5**) = 6 / (6 + 6) = 0.500





Antennae length is **5**

Courtesy of Eammon Keogh, UCR, eamonn@cs.ucr.edu

Minimum Description Length

revisiting the def. of h*MAP*:

$$h_{MAP} = \arg\max_{h \in H} P(D|h)P(h)$$

we can rewrite it as:

$$h_{MAP} = \arg\max_{h \in H} \log_2 P(D|h) + \log_2 P(h)$$

or

$$h_{MAP} = \arg\min_{h \in H} -\log_2 P(D|h) - \log_2 P(h)$$

But the first log is the cost of coding the data *given* the theory, and the second - the cost of coding the theory

Observe that:

for data, we only need to code the exceptions; the others are correctly predicted by the theory

MAP principles tells us to choose the theory which encodes the data in the shortest manner

the MDL states the trade-off between the complexity of the hypo. and the number of errors

Bayes optimal classifier

- so far, we were looking at the "most probable hypothesis, given a priori probabilities". But we really want the most probable classification
- this we can get by combining the predictions of all hypotheses, weighted by their posterior probabilities:
- this is the bayes optimal classifier BOC:

$$P(v_j|D) = \sum_{h_i} P(v_j|h_i)P(h_i|D)$$

$$\arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D)$$

Example of hypotheses
h1, h2, h3 with posterior probabilities
.4, .3. .3
A new instance is classif. pos. by h1 and neg. by h2, h3

Bayes optimal classifier

$V = \{+, -\}$

$P(h1|D) = .4$, $P(-|h1) = 0$, $P(+|h1) = 1$

…

Classification is " –" (show details!)

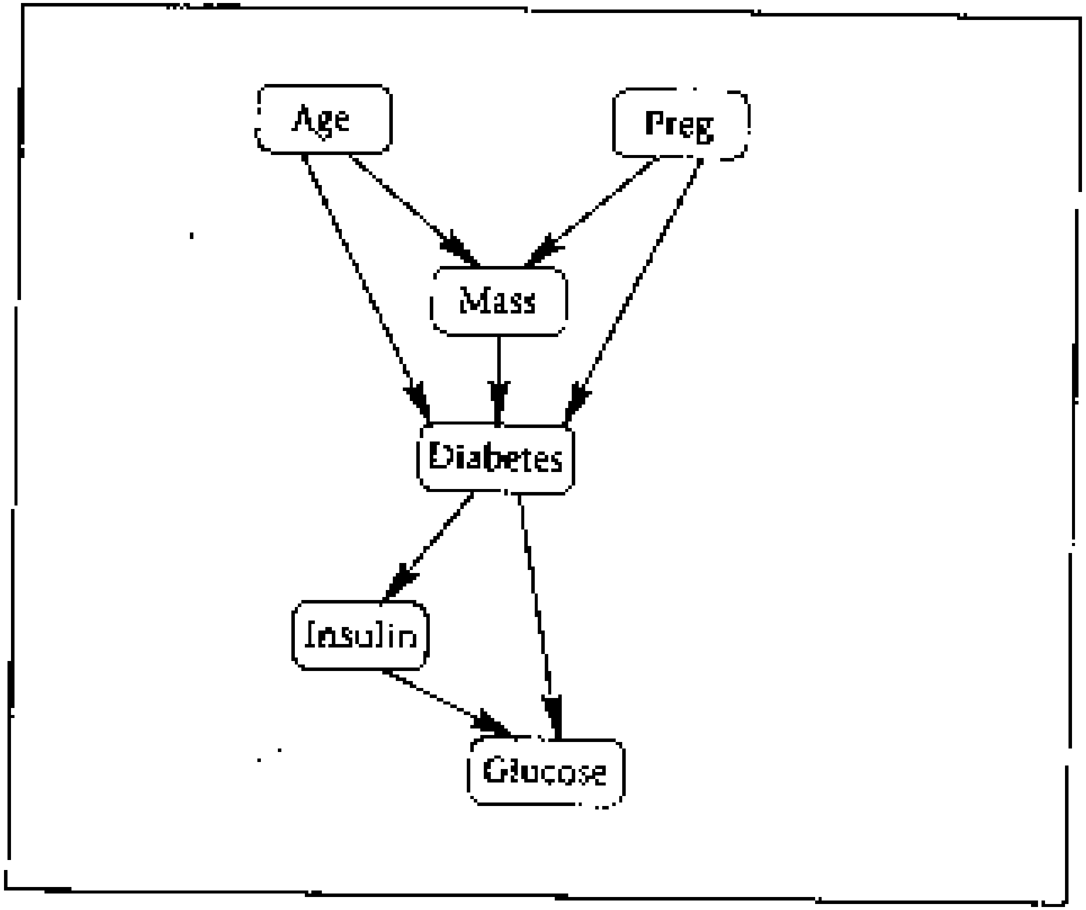


Figure 19. A Probabilistic Network for Diabetes Diagnosis.

- Captures probability dependencies
- ea node has probability distribution: the task is to determine the join probability on the data
- In an appl. a model is designed manually and forms of probability distr. Are given
- Training set is used to fit the model to the data
- Then probabil. Inference can be carried out, eg for prediction

First five variables are observed, and the model is Used to predict diabetes

*P(A, N, M, I, G, D)=P(A)*P(n)*P(M|A, n)*P(D|M, A, N)*P(I|D)*P(G|I,D)*

Age	$P(A)$
0–25	
26–50	
51–75	
> 75	

Preg	$P(N)$
0	
1	
>1	

		$P(M\|A, N)$		
Age	Preg	0–50	51–100	>100
0–25	0			
0–25	1			
0–25	> 1			
26–50	0			
26–50	1			
26–50	>1			
51–75	0			
51–75	1			
51–75	>1			
>75	0			
>75	1			
>75	> 1			

Table 3. Probability Tables for the Age, Preg, and Mass Nodes from Figure 19.
A learning algorithm must fill in the actual probability values based on the observed training data.

- how do we specify prob. distributions?
- discretize variables and represent probability distributions as a table
- Can be approximated from frequencies, eg table P(M|A, N) requires 24parameters
- For prediction, we want (D|A, n, M, I, G): we need a large table to do that

- no other classifier using the same hypo. space e and prior K can outperform BOC
- the BOC has mostly a theoretical interest; practically, we will not have the required probabilities
- another approach, Naive Bayes Classifier (NBC)

$$v_{MAP} = \arg\max_{v_j \in V} P(v_j \mid a_1, \ldots a_n) = \arg\max_{v_j \in V} \frac{P(a_1, \ldots a_n \mid v_j) P(v_j)}{P(a_1, \ldots a_n)} =$$

$$\arg\max_{v_j \in V} P(a_1, \ldots a_n \mid v_j) P(v_j)$$

under a simplifying assumption of independence of the attribute values given the class value:

To estimate this, we need (#of possible values)*(#of possible instances) examples

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_i P(a_i \mid v_j)$$

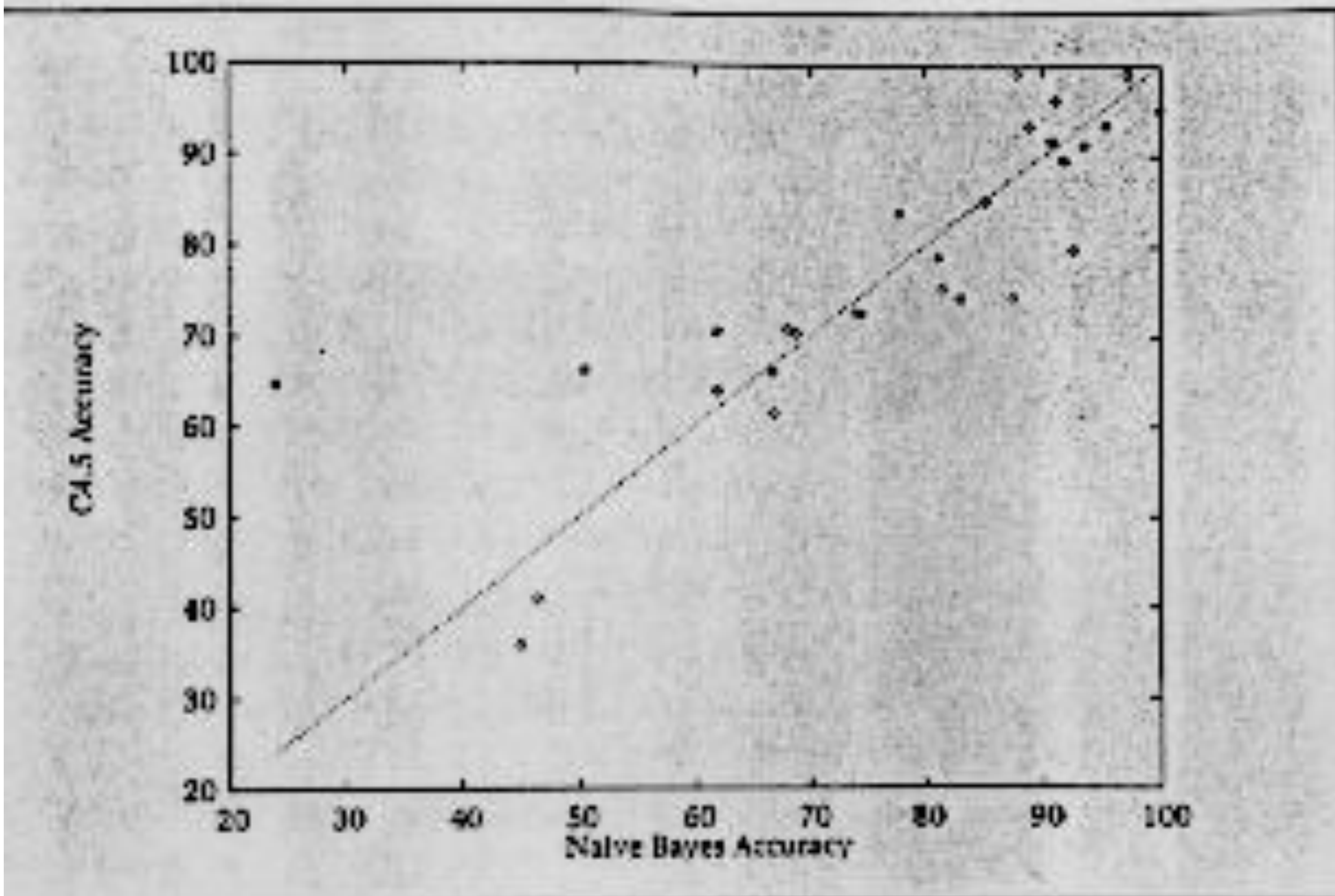


Figure 21. Comparison of C4.5 and the Naive Bayesian Classifier on 28 Data Sets.

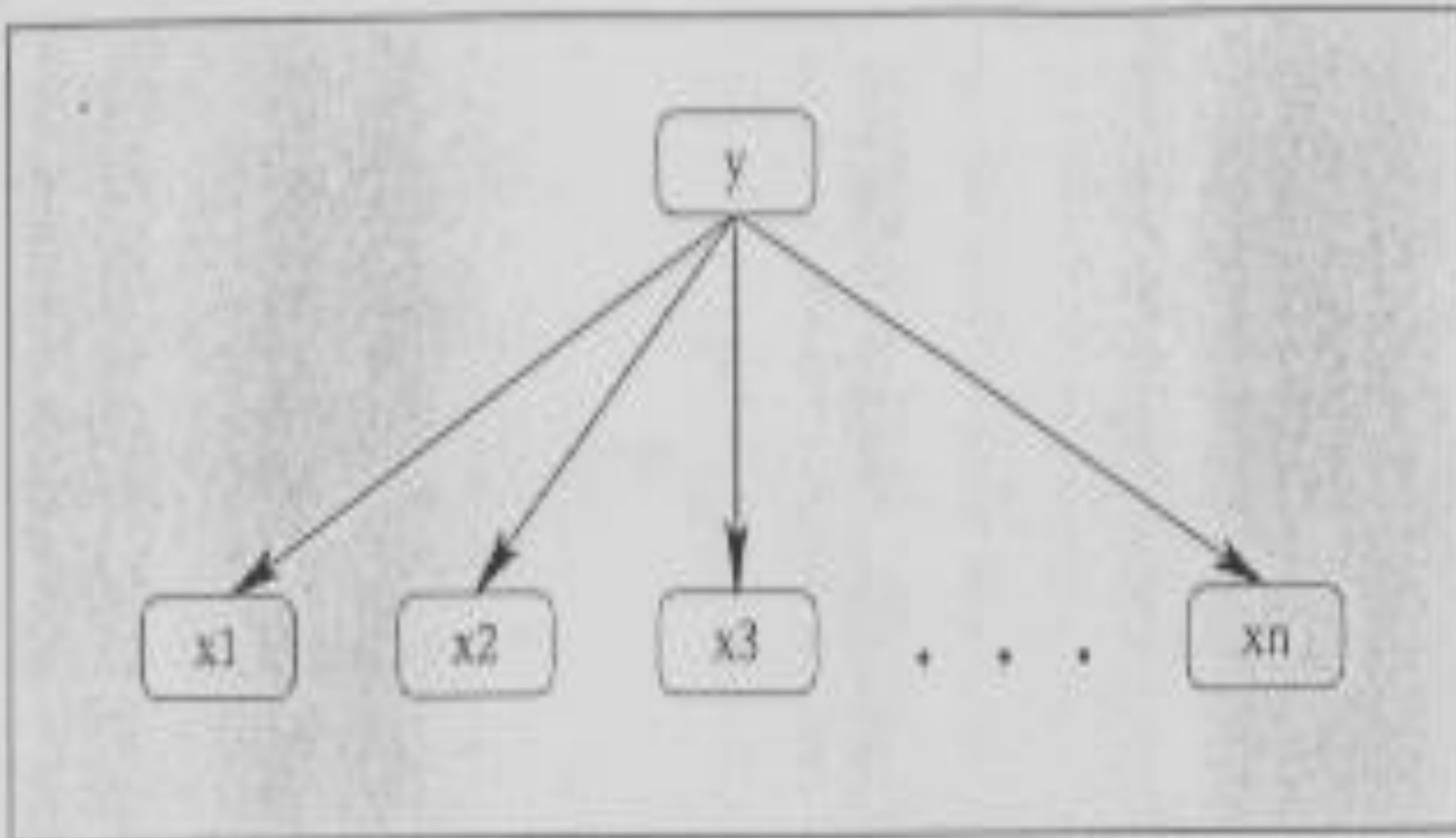


Figure 20. Probabilistic Network for the Naive Bayes Classifier.

- in NB, the conditional probabilities are *estimated* from training data simply as normalized frequencies: how many times a given attribute value is associated with a given class wrt to all classes: $\dfrac{n_c}{n}$
- no search!
- example

Example we are trying to predict *yes* or *no* for *Outlook=sunny, Temperature=cool, Humidity=high, Wind=strong*

$$v_{NB} = \underset{v_j \in [yes,no]}{\arg\max}\, P(v_j)\prod_i P(a_i \mid v_j) = \underset{v_j \in [yes,no]}{\arg\max}\, P(v_j)P(Outlook = sunny \mid v_j)$$

$$P(Temperature = cool \mid v_j)P(Humidity = high \mid v_j)P(Wind = strong \mid v_j)$$

P(yes)=9/14   P(no)=5/14

P(Wind=strong|yes)=3/9 P(Wind=strong|no)=3/5 etc.

P(yes)P(sunny|yes)P(cool|yes)P(high|yes)Pstrong|yes)=.0053

P(no)P(sunny|no)P(cool|no)P(high|no)Pstrong|no)=.0206

so we will predict *no*

Geometric decision boundary

- Assume a binary NB classifier f with instances $[x_1,\ldots,x_n,y]$, $y=0$ or $y=1$. Denote by v_0 (v_1) the vector of probabilities of all instances belonging to class *0 (1),* respectively.

$$f(x) = \log \frac{P(y=1 \mid x)}{P(y=0 \mid x)} = \log P(y=1 \mid x) - \log P(y=0 \mid x) =$$

$$(\log v_1 - \log v_0)x + \log p(y=1) - \log p(y=0)$$

- This expression is linear in x. Therefore the decision boundary of the NB classifier is linear in the feature space *X,* and is defined by $f(x) = 0$.

- Further, we can not only have a decision, but also the prob. of that decision: $\frac{n_c}{n}$ $\frac{.0206}{.0206+.0053}=.795$
- we rely on n for the conditional probability, where n is the total number of instances for a given class, n_c is how many among them have a specific attribute value
- if we do not observe any values of , or very few, this is a problem for the NB classifier (multiplications!)
- So: smoothen; see Witten p. 91

- we will use the estimate $\dfrac{n_c + mp}{n+m}$

  where p is the prior estimate of probability,

  m is *p=1/k* for *k* values of the attribute; *m* has the effect of augmenting the number of samples of class ;

  large value of m means that priors p are important wrt training data when probability estimates are computed, small – less important
- In practice often *1* is used for *mp* and *m*

Application: text classification

- setting: newsgroups, preferences, etc. Here: ‘like’ and ‘not like’ for a set of documents
- text representation: “bag of words”: Take the union of all words occurring in all documents. A specific document is represented by a binary vector with 1’s in the positions corresponding to words which occur in this document
- high dimensionality (tens of thou. of features)

$$v_{NBC} = \max_{v_j \in like, notlike} \{P(like)P(w_1 \mid like)...P(w_n \mid like),$$

$$(P(notlike)(P(w_1 \mid notlike)...P(w_n \mid notlike)\}$$

- We will estimate $P(w_k|v_j)$ as m-estimate with equal priors

$$\frac{n_k + 1}{n + |vocabulary|}$$

- incorrectness of NB for text classification (e.g. if ‘Matwin’ occurs, the previous word is more likely to be ‘Stan’ than any other word; violates independence of features)
- but amazingly, in practice it does not make a big difference

LEARN_NAIVE_BAYES_TEXT(*Examples*, V)

Examples is a set of text documents along with their target values. V is the set of all possible target values. This function learns the probability terms $P(w_k|v_j)$, describing the probability that a randomly drawn word from a document in class v_j will be the English word w_k. It also learns the class prior probabilities $P(v_j)$.

1. collect all words, punctuation, and other tokens that occur in Examples

- *Vocabulary* $\leftarrow$ the set of all distinct words and other tokens occurring in any text document from *Examples*

2. calculate the required $P(v_j)$ and $P(w_k|v_j)$ probability terms

- For each target value v_j in V do
  - $docs_j \leftarrow$ the subset of documents from *Examples* for which the target value is v_j
  - $P(v_j) \leftarrow \frac{|docs_j|}{|Examples|}$
  - $Text_j \leftarrow$ a single document created by concatenating all members of $docs_j$
  - $n \leftarrow$ total number of distinct word positions in $Text_j$
  - for each word w_k in *Vocabulary*
    - $n_k \leftarrow$ number of times word w_k occurs in $Text_j$
    - $P(w_k|v_j) \leftarrow \frac{n_k+1}{n+|Vocabulary|}$

CLASSIFY_NAIVE_BAYES_TEXT(*Doc*)

Return the estimated target value for the document Doc. a_i denotes the word found in the ith position within Doc.

- *positions* $\leftarrow$ all word positions in *Doc* that contain tokens found in *Vocabulary*
- Return v_{NB}, where

$$v_{NB} = \underset{v_j \in V}{\operatorname{argmax}}\, P(v_j) \prod_{i \in positions} P(a_i|v_j)$$

Taking into account frequencies of words

- In order to determine the weight of term *k* for the representation of document *j*, the *term frequency inverted document frequency* (*tfidf*) is often used. This function is defined as:
- $tfidf(t_k,d_j) = \#(t_k, d_j) * \log (|Tr| / \#(t_k))$
- where Tr is the training set, $\#(t_k, d_j)$ is the number of times t_k occurs in d_j, and $\#(t_k)$ is the number of documents in Tr in which t_k occurs at least once (the document frequency of tk.) <u>Meaning?</u>
- To make the weights fall in the [0,1] interval and for the documents to be represented by vectors of equal length, the following cosine normalization is used:
- $w_{k,j} = tfidf(t_k, d_j) / sqrt(\sum_{s=1..r} (tfidf(t_s,d_j))^2)$

Measures for text classification

Refer to the contingency table:

- Precision (Pr) = TP / (TP + FP)
- Recall (Re) = TP / (TP + FN)

Complementarity of R & P, break-even

- Also, the F_α-measure:= $(1+\alpha)P*R/(\alpha P+R)$
- For $\alpha=1$, F-measure

Bayesian algorithms for text categorization Naive Bayes for and against

- Naive Bayes attractive features: simple model, easy to implement and fast
- Naive Bayes has its share of shortcomings, primarily due to its strict assumptions
- If only presence/absence of word is represented, we have a multi-variate Bernoulli model for NB

Naive Bayes. Next step ahead

- improving Naive Bayes by

    1. Learning better classification weights
    2. Modeling text better (transforming the data)

- The final goal is to have a fast classifier that performs almost as well as the SVM (on text)

Multinomial Naïve Bayes (MNB)

- designed for text categorization - requires BOW input data

- attempts to improve the performance of text classification by the incorporation the words frequency information

- models the distribution of words (features) in a document as a multinomial distribution

Multinomial model and classifying documents

- We assume the *generative* model: a "source" generates an *n*-word long document, from a vocabulary of *k* words *(|V| = k*)
- Here we usually find the hypothesis (model) *most likely to have generated the data* (whereas in MAP we are looking for a model most likely *given* the observed data
- Word occurrences are *independent*
- A new document can then be modeled by a multinomial distribution

Multinomial distribution

- in probability theory, the multinomial distribution is a generalization of the binomial distribution.

- The binomial distribution is the probability distribution of the number of "successes" in n independent Bernoulli trials, with the same probability of "success" on each trial. (n tosses of a coin)

- In a multinomial distribution, each trial results in exactly one of some fixed finite number k of possible outcomes, with probabilities p1, ..., pk (so that pi ≥ 0 for i = 1, ..., k and $\sum_{j=1}^{k} p_j = 1$), and there are n independent trials. Then let the random variables Xi indicate the number of times outcome number i was observed over the n trials. X=(X1,…,Xk) follows a multinomial distribution with parameters n and p, where p = (p1, ..., pk).

From pdf file!

- Pp. 34 to 49

Discrminative Naïve Bayes for Text Classification

- See course webpage for the original paper Discriminative Multinominal Naive Bayes for Text Classification by Su, Sayyad Shirabad, Matwin, and Huang
- Software incorporated in weka

MNB (Multinomial naïve Bayes classifier)

- MNB model: $$P(d \mid c\ ) = \frac{(\sum_i f_i)!}{\prod_i f_i!} \prod_{i=1} P(w_i \mid c)^{f_i}$$
- where f_i = # of occurrences of word w_i in d
- Three independence assumptions:
  - occurrence of w_i is independent of occurrences of all the other words
  - occurrence of w_i is independent of itself
  - $|d|$ is independent of class of d
- MNB classifier:

$$P(c \mid d) = \frac{P(c) \prod_{i=1}^{n} P(w_i \mid c)^{f_i}}{P(d)} \qquad (1)$$

Frequency Estimate

- How do we get $P(w_i|c)$?
- We estimate it by Frequency Estimate (FE): this is the essence of the generative approach:

$$\hat{P}(w_i \mid c) = \frac{f_{ic}}{f_c} \qquad (2)$$

- where f_{ic} = # of occurrences of w_i in docs of class c
- f_c = total # of word occurrences in documents of class c
- FE is efficient: a single scan thru all the instances

Problems with MNB

- FE is not meant to optimize accuracy! It is meant to optimize likelihood
- If the independence assumptions are true, then FE also maximizes accuracy. But they are not true.

MNB is efficient

- Using the conditional probability (from the multinomial framework of MNB), we easily get the aposteriori probability:

$$P(c \mid d) = \alpha P(c) \prod P(w_i \mid c)^{f_i} \qquad (**)$$

and

$$C(d) = \arg\max_c P(c) \prod P(w_i \mid c)^{f_i}$$

- This means that we can ignore all the words from the corpus missing in a given document! (why?). In practice, this saves a lot of time!

Frequency estimate problems

- Objective function of FE is

$$LL(T) = \sum_{t=1}^{|T|} \log \hat{P}(c^{t|} \mid w^{t}) + \sum_{i=1}^{T} \log \hat{P}(w^{t})$$

- First term: how well the model estimates the probabil. Of class given the words
- Second term: how well the model estimates the joint distribution of words
- What happens when the number of words gets large?

Basic idea of DMNB

- Keep FE, but extend it so that the discriminative character of classification is taken into account
- Note that in each step of FE we in fact have a classifier: it's a classifier whose conditional (local) probabilities are built in (**)

Basic idea of DMNB

- Do this by computing in each step

$$L(d) = P(c \mid d) - \hat{P}(c \mid d) \qquad (5)$$

- We intialize *P(c|d) = 1* (for the true class c of d) . Also initially for each class (in the first turn of the loop)

$$\hat{P}(c \mid d) = \frac{1}{C}$$

Algorithm 1 Discriminative Multinominal Naive Bayes

1. Initialize each word frequency entry F_{ic} to 0
2. **For** t from 1 to M **Do**
   - Randomly draw a training document d^t from the training data set T.
   - Estimate the probabilities parameters using Equation 2 and the current frequencies f_{ic}^t
   - Compute the posterior probability $\hat{P}(c|d^t)$. ***from (1)***
   - Compute the loss $L(d^t)$ using Equation 5.
   - **For** each non-zero word w_i in the document d^t
     - Let f_i^t= the frequency of the word w_i in the t_{th} document d^t
     - Let f_{ic}^{t+1}=f_{ic}^t+$L(d^t) * f_i^t$.

Example

	docID	words in document	class label
training data	1	NB classifier performance	Evaluation
	2	SVM classifier performance	Classification
	3	NB classifier	Classification
test data	4	NB	Classification

$C) = \frac{1}{5}$

$$\hat{P}(c=E)=\frac{1}{3}\hat{P}(c=C)=\frac{2}{3},\hat{P}(w_1=NB\,|\,c=E)=\frac{1}{3},\hat{P}(w_1=NB\,|\,c=C)=\frac{1}{5}$$

$$\frac{\hat{P}(c=E \mid w_1 = NB)}{\hat{P}(c=C \mid w_1 = NB)} = \frac{1}{2} \times \frac{\frac{1}{3}}{\frac{1}{5}} = \frac{5}{6} < 1$$

- **So MNB classifies the test case as class C (correct)**
- **But now substitute “Naïve Bayes” for “NB” throughout the training and test data**

$$\frac{\hat{P}(c=E \mid w_1 = N, w_1 = \ B)}{\hat{P}(c=C \mid w_1 = N, w_2 = B)} = \frac{1}{2} \times \left(\frac{6}{4}\right)^2 = \frac{9}{8} > 1$$

ratio of $\dfrac{\hat{P}(w_1 = N \mid c)}{\hat{P}(w_2 = B \mid c)}$

- Here, MNB will classify the test instance as of class E, incorrect! It is because the assumption of independence between occurrences of word "N" and "B" is not true in this data

But for DMNB

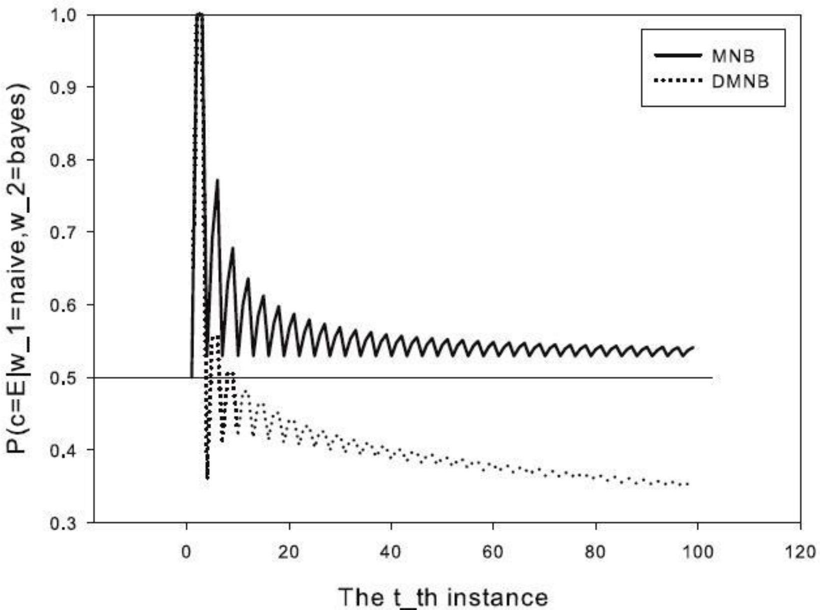


Figure 2. The y-axis is the predicted probability. The x-axis is the t_{th} document fed into the algorithms.

- DMNB converges to ~ 0.35 for this ratio

Extensive empirical tests of DMNB

- …indicate it is competitive wrt SVM, but MUCH faster (50-600 times!)

Table 2. Summary of accuracy comparisons on Multi Class Datasets.

	LibSVM	CNB	MNB
DMNB	0/12/7	1/7/11	0/5/14
LibSVM		4/6/9	3/6/10
CNB			4/7/8

Spam filtering

- In some countries SMS spam is a very serious challenge for mobile operators
- GuangXi telecom (China) is using a spam filter based on our algorithm (Machine Learning/Text classification)
  - Reads 30M messages a day
  - 1% is spam
  - Detects 99.99% of spam
  - Velocity/Real time
  - Volume

http://honeyblog.org/archives/32-Towards-Proactive-Spam-Filtering.html

DALHOUSIE UNIVERSITY
INSTITUTE FOR BIG DATA ANALYTICS

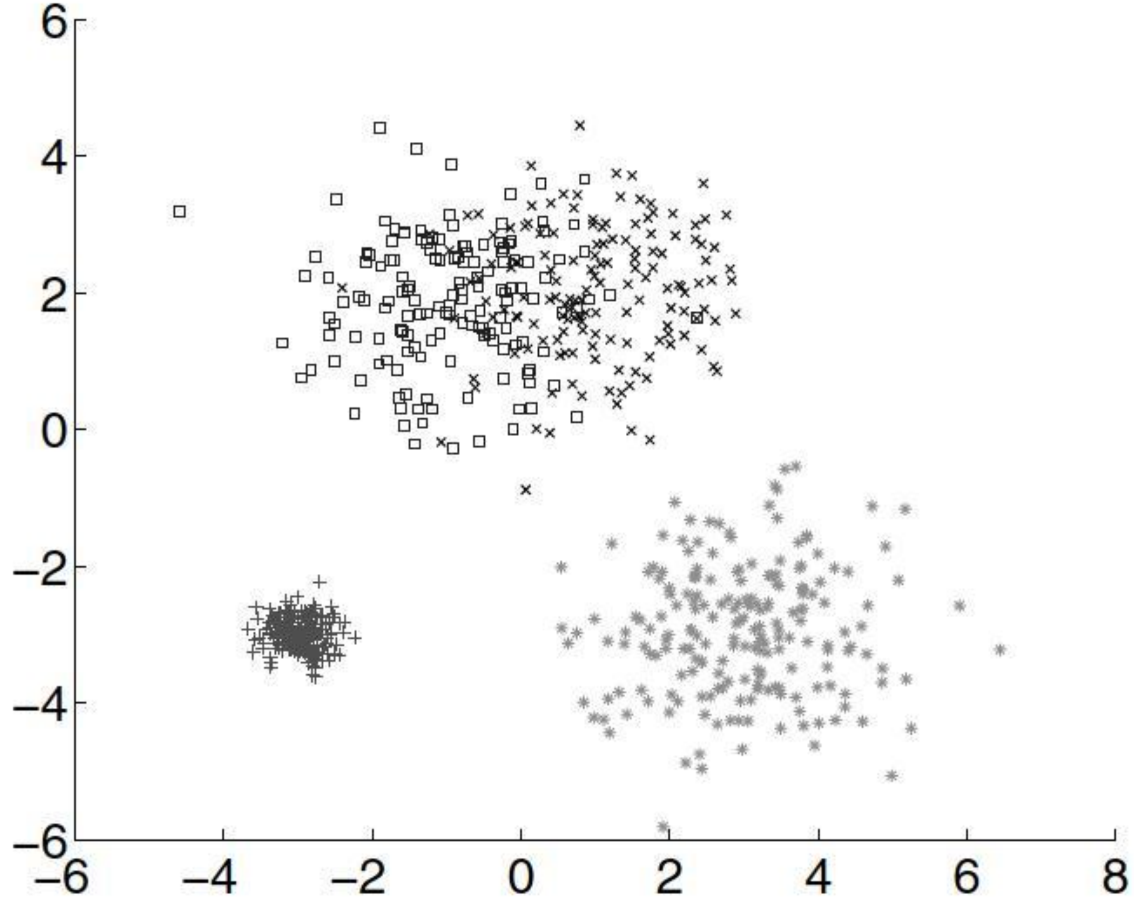
Clustering

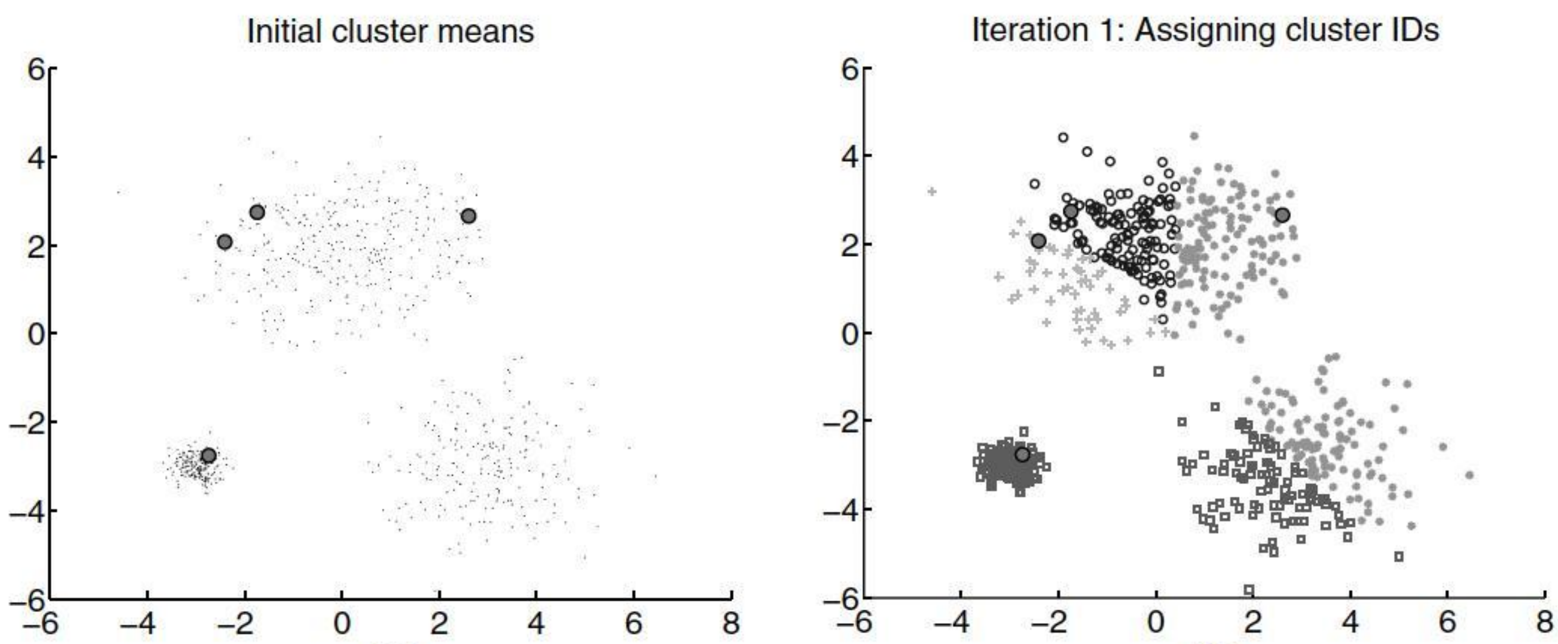
- Unsupervised learning task; data has **no labels**
- The task is to find “natural groupings” in data
- Practically important, often the first step in exploratory data analysis
- Comes in different variants:
  - “Exclusive” clusters
  - “Shared” clusters
  - Probabilistic cluster membership

Clustering – k means

1. Define *k*- the number of clusters
2. Choose *k* points randomly as cluster centres
3. For any instance, assign it to the cluster whose centre is the closest
4. If no cluster gets modified, STOP
5. Make centroids ("instances" created by taking means of all instances in the cluster) new clusters
6. go to 3

iterative relocation

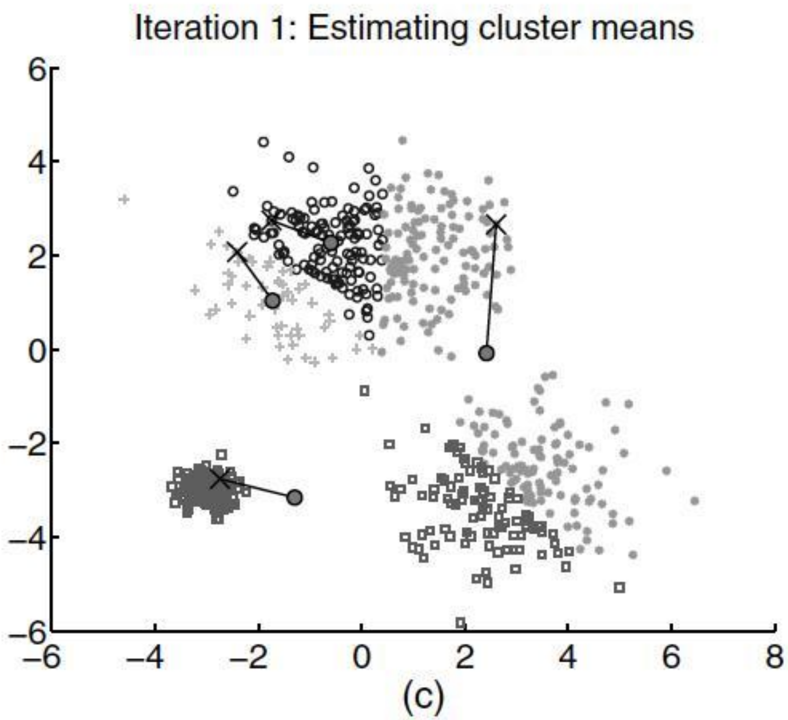

6
4
2
0
−2
−4
−6
−6
−4
−2
0
2
4
6
8

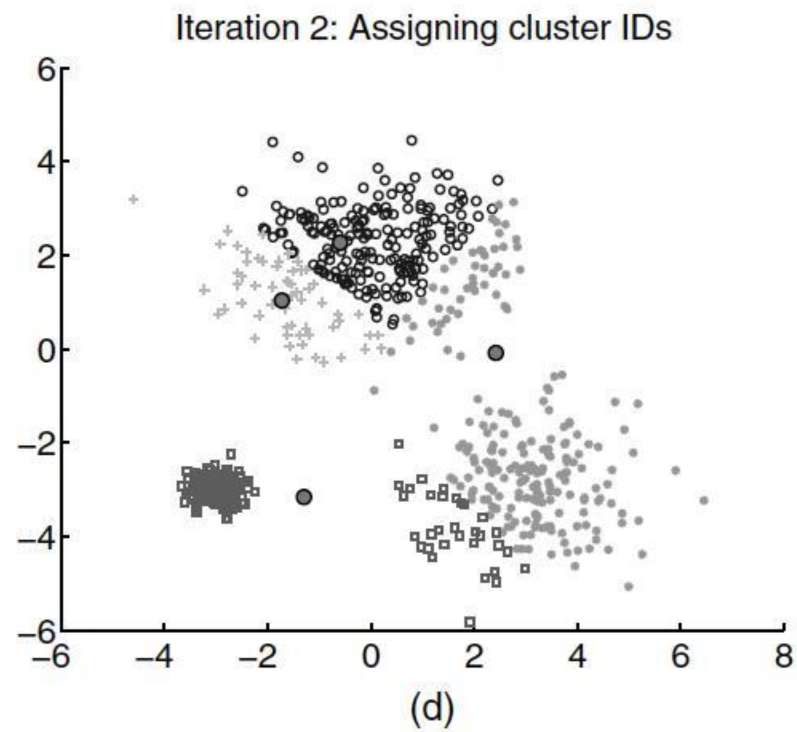


Initial cluster means
Iteration 1: Assigning cluster IDs
6
4
2
0
−2
−4
−6
−6
−4
−2
0
2
4
6
8

(a)

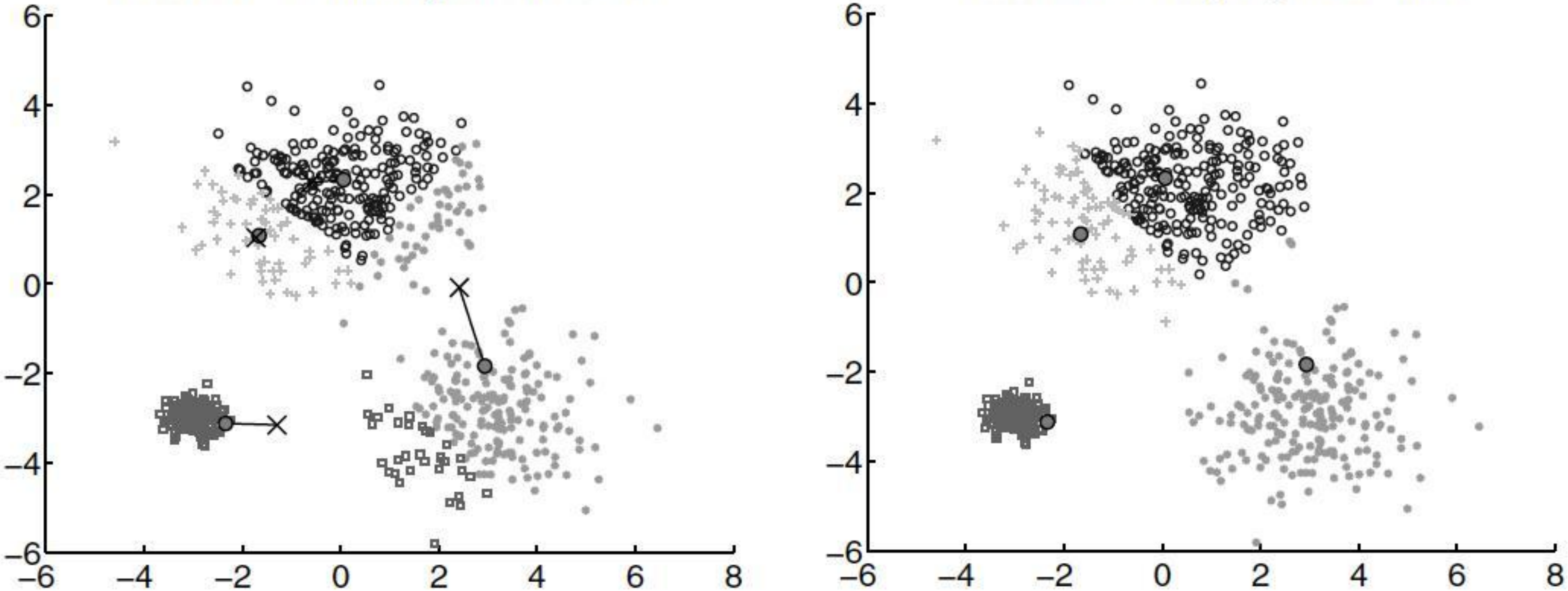
(b)


Iteration 1: Estimating cluster means
6
4
2
0
−2
−4
−6
−6
−4
−2
0
2
4
6
8
(c)


Iteration 2: Assigning cluster IDs
6
4
2
0
−2
−4
−6
−6
−4
−2
0
2
4
6
8
(d)

Iteration 2: Estimating cluster means

Iteration 3: Assigning cluster IDs



(e) (f)

- When *k*-means terminates, the sum of all distances of points to their cluster centres is minimal
- This is only local, i.e. depends on the initial choice of *k*
- Efficiency problem - #iterations**k***N*
- *k*D trees can be used to improve efficiency
- *k*-medoids *vs* *k*-means

Sensitivity to outliers

- Example: {1, 2, 3, 8, 9, 10, 25}
- Clustering {1, 2, 3}, {8, 9, 10, 25} *vs* clustering {1, 2, 3, 8}, {9, 10, 25}

- How to choose *k*?
- x-val on the minimum distance: expensive
- Iterative on *k*; create 2 clusters, split recursively. “freeze” the initial 2-clustering
- When to stop splitting? Pitfall of a non-solution with 1-instance clusters; remedy – MDL-based splitting crietrion:
  - if (info. required to represent 2 new cluster centres and instances wrt these centres) > *(*info required to represent 1 original cluster centre and instances wrt that centre) then don’t split else split

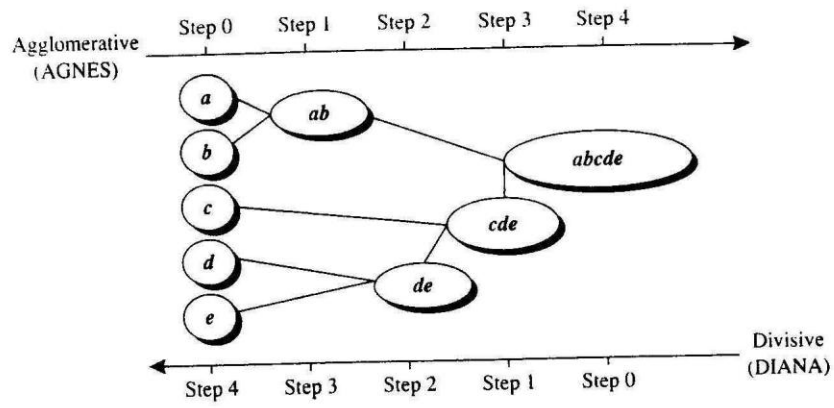
k-medoids clustering

- Instead of the mean as the cluster centre, use an instance
- More robust and less sensitive to outliers

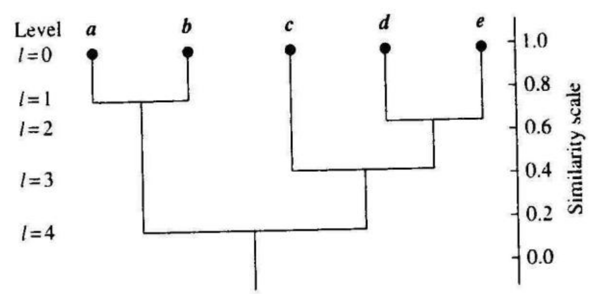
Hierarchical clustering

- Grouping instances into a hierarchy (itself not given)
- Agglomerative clustering (bottom-up) and divisive clustering (top-down)

Hierarchical clustering – example



Agglomerative and divisive hierarchical clustering on data objects $\{a, b, c, d, e\}$.



Dendrogram representation for hierarchical clustering of data objects $\{a, b, c, d, e\}$.

Evaluation of clustering

- Difficult task
- Intrinsic measures exist
- Often done on classification datasets, which is a bit of a miss
- Human comprehensibility of clusters a valuable part of evaluation

Probabilistic clustering

- Finite mixture model
- Set of *k* probability distributions represents *k* clusters: each distribution determines the probability that an instance *x* would have a certain set of attribute values ***if it was known that*** *x* ***belongs to this cluster***
- There is also a probability distribution that reflects the relative population sizes of each cluster

Finite mixture problem

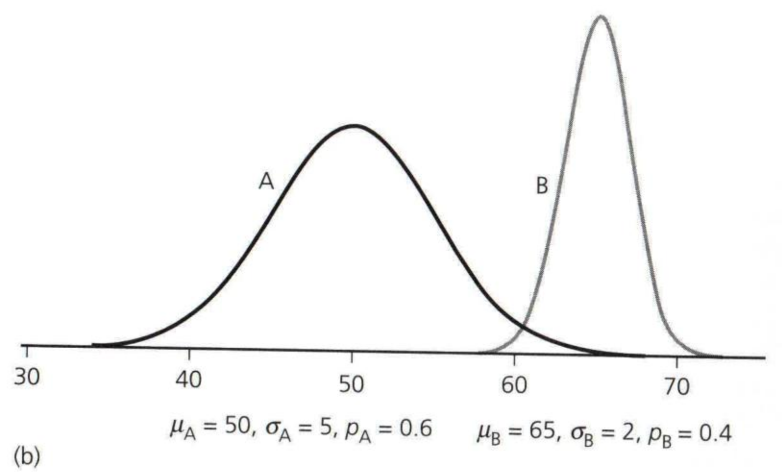


Figure 6.19 A two-class mixture model.

- Given set of instances without knowing which gaussian generated which imnstance, determine μ_A, σ_A, p_A, μ_B, σ_B ($p_B = 1 - p_A$)

Mixed model cont'd

- Had we known from which distribution (*A* or *B*) a instance comes from, we could easily compute the two μ, σ, and *p*
- If we knew the five parametrs, we would assign a new *x* to cluster *A* if

$$\frac{\Pr[A \mid x]}{\Pr[B \mid x]} = \frac{f(x, \mu_A, \sigma_A)}{f(x, \mu_B, \sigma_B)} > 1$$

- where

$$f(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{(x-\mu)^2}{2\sigma^2}}$$

The EM algorithm

- Since we do **not** know any of the five parameters, we estimate and maximize:
  - Start with a random assignment of the 5
  - Compute cluster probabilities for each instance ("expected" cluster assignments)
  - Use these cluster assignments to compute the 5 parameters ("maximize" the likelihood of the distribution given the data)
- Note that the same algorithm, with label assignment instead of cluster assignment, can be used to assign labels to unlabeled data generated by a mixture model!

EM cont'd

- But when to stop?
- Essentially, when the learning curve flattens. Specifically, when the overall probability that the data comes from this model

$$\prod_i (p_A \Pr[x_i \mid A] + p_B \Pr[x_B \mid B])$$

(where the cluster probabilities are given by the $f(x,\mu,\sigma)$ starts to yield very small differences in a number of consecutive iterations

- in practice EM works with log-likelihoods to avoid multiplications

EM cont'd

- The framework is extended to mixtures of *k* gaussians (two-class to *k*-class, but *k* must be known)
- The framework is further easily extended to multiple attributes, under the assumption of independence of attributes…
- …and further extended with dropping the independence assumption and replacing the standard deviation by the covariance matrix

EM cont'd

- Parameters: for n independent attributes, $2n$ parameters; for covariant attributes, $n+n(n+1)/2$ parameters: n means and the symmetric nxn covariance matrix
- For (independent) nominal attributes, EM is like Naïve Bayes: instead of normal distribution, kv parameters per attribute are estimated, where v is the number of values of the attribute:
  – Expectation: determine the cluster (like the class in NB)
  – Maximization: like estimating NB priors (attribute-value probabilities) from data

Associations

Given:

I = {i1,…, im} set of items

D set of transactions (a database), each transaction is a set of items $T \subset 2^I$

Association rule: $X \Rightarrow Y$, $X \subset I$, $Y \subset I$, $X \cap Y = 0$

confidence c: ratio of # transactions that contain both X and Y to # of *all* transaction that contain X

support s: ratio of # of transactions that contain both X and Y to # of transactions in D

Itemset is *frequent* if its support > θ

An *association rule* $A \Rightarrow B$ is a conditional implication among itemsets A and B, where $A \subset I$, $B \subset I$ and $A \cap B = \emptyset$.

Support of an association rule = P(A $\cup$ B). The *confidence* of an association rule r: $A \Rightarrow B$ is the conditional probability that a transaction contains B, given that it contains A. Confidence = $P(B|A)$

The support of rule r is defined as: $sup(r) = sup(A \cup B)$. The confidence of rule r can be expressed as $conf(r) = sup(A \cup B)/sup(A)$.

Formally, let $A \subset 2^I$; $sup(A) = |\{t: t \in D, A \subset t\}|/|D|$, if $R = A \Rightarrow B$ then $sup(R) = sup(A \cup B)$, $conf(R) = sup(A \cup B)/sup(A)$

Itemsets and association rules

- Itemset = set of items
- k-itemset = set of k items
- Finding association rules in databases:
  - Find all frequent (or large) itemsets (those with support > min_s
  - Generate rules that satisfy minimum confidence

Example

- Computer store
- Customers buying computers and financial software
- What does the rule mean:

computer ➔ financial_mgmt_software

[support = 2%, conf = 60%]

Associations - mining

Given D, generate all assoc rules with c, s > thresholds min_c, min_s

(items are ordered, e.g. by barcode)

Idea:

find all itemsets that have transaction support > min_s : large itemsets

Associations - mining

to do that: start with indiv. items with large support

in ea next step, k,

- use itemsets from step k-1, generate new itemset C_k,
- count support of C_k (by counting the candidates which are contained in any t),
- prune the ones that are not large

Apriori property

- All [non-empty] subsets of a frequent itemset must be frequent

- Based on the fact that an itemset *i* that is NOT frequent has support < min_s
- But inserting an additional item *A* in *i* will not increase the support of $i \cup A$

Associations - mining

```
L1 = {large 1-itemsets};
for ( k = 2; L_{k-1} ≠ ∅; k++ ) do begin
    C_k = apriori-gen(L_{k-1}); // New candidates
    forall transactions t ∈ D do begin
        C_t = subset(C_k, t); // Candidates contained in t
        forall candidates c ∈ C_t do
            c.count++;
    end
    L_k = {c ∈ C_k | c.count ≥ minsup}
end
Answer = ∪_k L_k;
```

*subset(c_k,***I***)* denotes those itemsets that are contained In transaction **I**

Candidate generation

C_k = apriori-gen(L_{k-1})

insert into C_k
select p.item$_1$, p.item$_2$, ..., p.item$_{k-1}$, q.item$_{k-1}$
from L_{k-1} p, L_{k-1} q
where p.item$_1$ = q.item$_1$, . . ., p.item$_{k-2}$ = q.item$_{k-2}$,
p.item$_{k-1}$ < q.item$_{k-1}$;

Next, in the *prune* step, we delete all itemsets $c \in C_k$ such that some $(k-1)$-subset of c is not in L_{k-1}:

forall itemsets $c \in C_k$ **do**
  forall $(k-1)$-subsets s of c **do**
    if $(s \notin L_{k-1})$ **then**
      delete c **from** C_k;

Select from *k-1*-frequent itemsets two overlapping subsets, add the differences

Example

TID	List of item_IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

From Han, Kamber,"Data Mining", p. 232

I = {I1,...,I5}

$min_s = 2$

Scan D for count of each candidate →

C_1

Itemset	Sup. count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Compare candidate support count with minimum support count →

L_1

Itemset	Sup. count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Generate C_2 candidates from L_1 →

C_2

Itemset
{I1, I2}
{I1, I3}
{I1, I4}
{I1, I5}
{I2, I3}
{I2, I4}
{I2, I5}
{I3, I4}
{I3, I5}
{I4, I5}

Scan D for count of each candidate →

C_2

Itemset	Sup. count
{I1, I2}	4
{I1, I3}	4
{I1, I4}	1
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2
{I3, I4}	0
{I3, I5}	1
{I4, I5}	0

Compare candidate support count with minimum support count →

L_2

Itemset	Sup. count
{I1, I2}	4
{I1, I3}	4
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2

Generate C_3 candidates from L_2 →

C_3

Itemset
{I1, I2, I3}
{I1, I2, I5}

Scan D for count of each candidate →

C_3

Itemset	Sup. count
{I1, I2, I3}	2
{I1, I2, I5}	2

Compare candidate support count with minimum support count →

L_3

Itemset	Sup. count
{I1, I2, I3}	2
{I1, I2, I5}	2

Firstly, C3 ={{I1,I2,I3},{I1,I2,I5},{I1,I3,I5}{I2,I3,I4},{I2,I3,I5},{I2,I4,I5}}
Only {I1,I2,I3},{I1,I2,I5} are left
C4={I1,I2,I3,I5} is attempted but pruned, C4=Ǿ terminates the algorithm

From itemsets to association rules

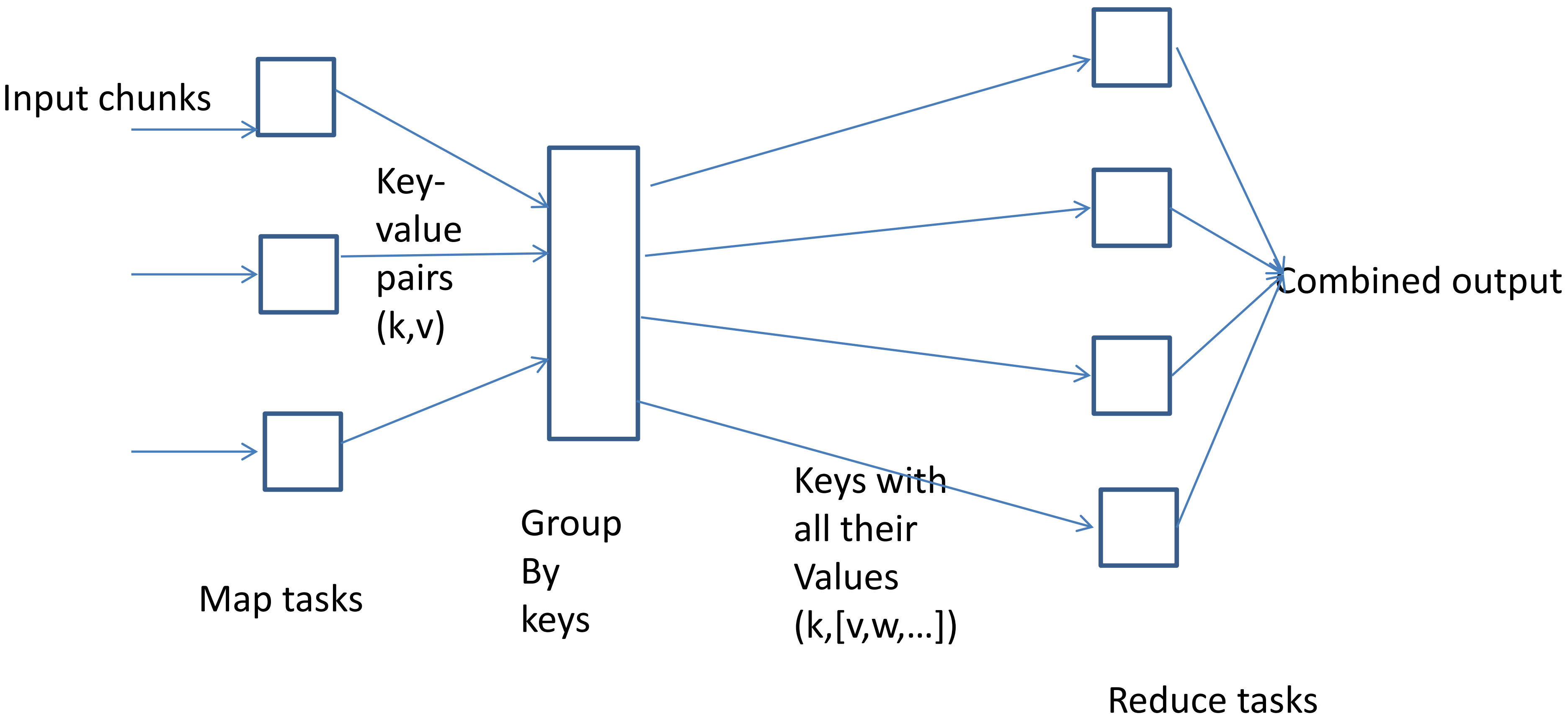
- For ea. frequent itemset *I* generate all the partitions of *I* into *s, I-s*
- Attempt a rule *s* ➔ *I-s* iff $support\_count(I)/support\_count(s) > min_c$
- e.g. for $min_c = 0.5$, what rules do we get?

[$conf(r) = sup(A \cup B)/sup(A)$]

Map/reduce

- A model for distributed computation and parallelization for very large datasets
- Based on Distributed File System (DFS)
- First proposed by Google for computing Page Rank
- Implemented in open source Hadoop architecture
- Excellent book on this topic is publicly available at http://infolab.stanford.edu/~ullman/mmds/book.pdf

Overall scheme



Wordcount example

- Counting the number of occurrences for each word in a collection of documents
- Input: repository of documents, each document is an element
- Map function: keys are strings (words), values are integers. Map reads a document and emits a sequence of key-value pairs, where value = 1:
- $(w_1,1), (w_2, 1),\ldots,(w_n,1)$

Wordcount example

- Note: a single Map tasks will typically process multiple documents
- If a word *w* occurs m times in the chunk assigned to a given Map task, there will be m pairs (*w*,1)

Wordcount example

- The Reduce task adds up all the values: output is (w, m), w is a word occurring at least once, and m is the number of occurrences of w in those docs

Master/worker (slave)

- Master assigns map and Reduce tasks to slave processes
- Each Map task is assigned chunks of the input file
- A file for ea. Reduce task is created on disk of ea. Map task; Master has the location info and for which Reduce task the file is made

Node failure

- When Master fails, the whole MR job must be restarted
- When Map fails, its task needs to be redone by another slave, even if completed. Reduce tasks are informed of changed input location
- When Reduce fails, its task is rescheduled to another slave later

Initial uses of MR

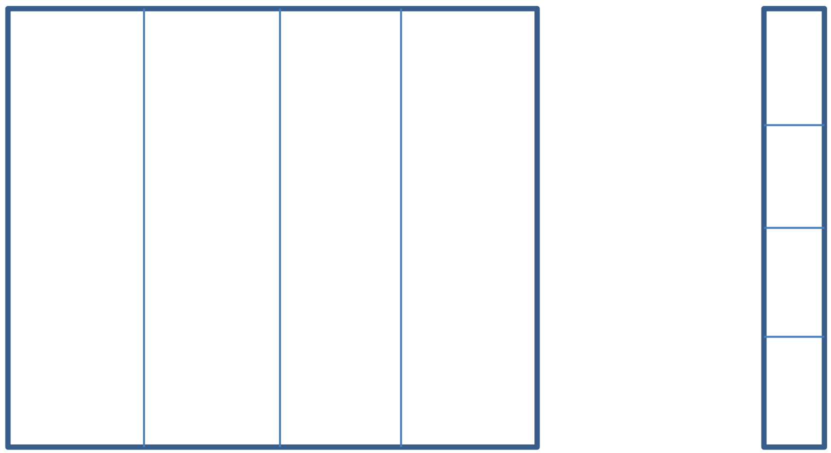
- Finding similar buying patterns between users
- Matrix-vector multiplication

Matrix-vector multiplication

- M = nxn matrix; v – vector of length n
- M x v = $[x_i]_1{}^n$ $x_i = \sum_{i=1}^{n} m_{ij} v_j$
- Ea. M task takes the whole vector v and a chunk of the matrix and produces a key-value pair (i, $m_{ij}v_j$)
- R task sums all the kv pairs for a given key i

When vector does not fit in memory

- Portion of the vector in one stripe fits in memory



- Each M task is assigned one chunk from one stripes of the matrix and the corresp. stripe of the vector

Basic algebra of relations

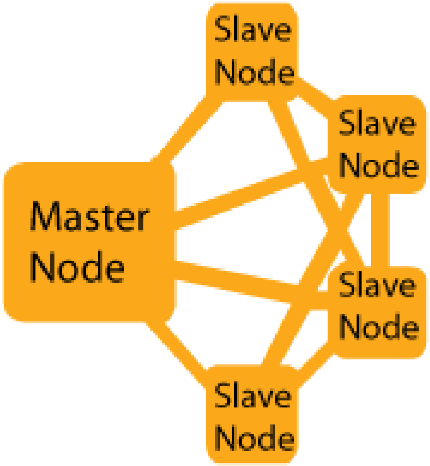
- Union: ea. input is made into kv pair (t,t)
- Make (t,t) when either there is one or two (t,t) pairs
- Intersection: make (t,t) only of kv list (t,t), otherwise make (t, NULL)
- Difference R-S: kv pairs (t,R), (t,S); for kv in R (R,R) make (t,t), otherwise – (R,S), (S,R), R – make (t,NULL)

See https://www.coursera.org/course/datasci for a good intro tutorial to Mr/Hadoop

Amazon Elastic MapReduce Tutorial

(Xuan Liu, uOttawa)

Introduction

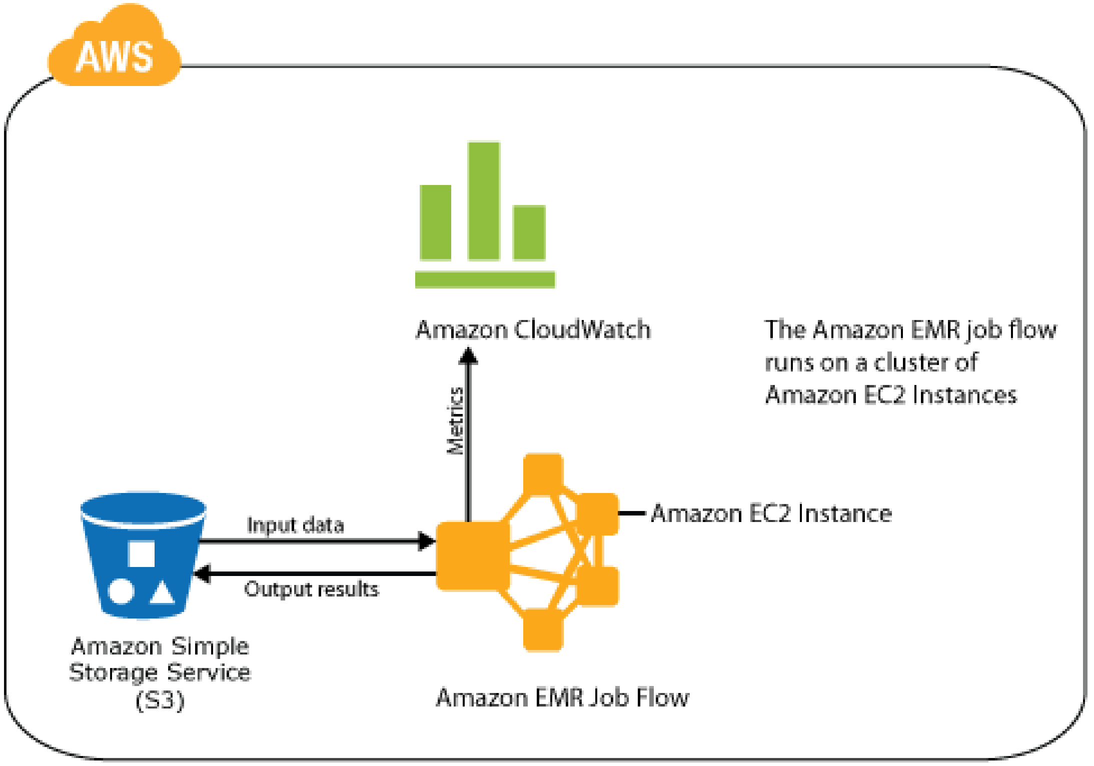


What is Amazon EMR

- Analyze and process vast amounts of data;
- Distribute computational work across Amazon cloud;
- The cluster is managed using Hadoop;
- Hadoop uses a distributed processing architecture called MapReduce.

Introduction-con't

- Amazon EMR make Hadoop work seamlessly with other Amazon Web Services (AWS)



Get Started-Count Words with Amazon EMR

- A tutorial using mapper and reducer functions to analyze data in a streaming cluster;
- Use Amazon EMR to count the frequency of words in a text file;
- The mapper logic is written as a Python script;
- The reducer is the built-in *aggregator* function provided by Hadoop;
- Use the Amazon EMR console to launch a cluster of virtual servers into a cluster to process the data in a distributed fashion.

Sign up for the service

- Your AWS account gives you access to all services;
- You are charged only for the resources that you use;
- Go to http://aws.amazon.com and click **Sign Up Now**;
- Follow the on-screen instructions;
- For console access, use your IAM user name and password to sign in to the AWS Management Console using the IAM sign-in page;
- For more information about creating access keys, see How Do I Get Security Credentials?

How much does it cost to run this tutorial?

- Cost of running an Amazon EMR cluster containing three m1.small instances for one hour: 29 cents;
- Cost of storing the input, output, and log files in Amazon S3: 13 cents a month (for new customer, free for the first year).

Visualization

- „let your data talk to you”
- Important to communicate
- Tools, eg:
  - Tableau
  - JIGSAW
  - ....

Windmap

- http://hint.fm/wind/ [Martin Wattenberg]
- Data from the National Digital Forecast Database
- It was exhibited in MOMA as graphical art

Visualization

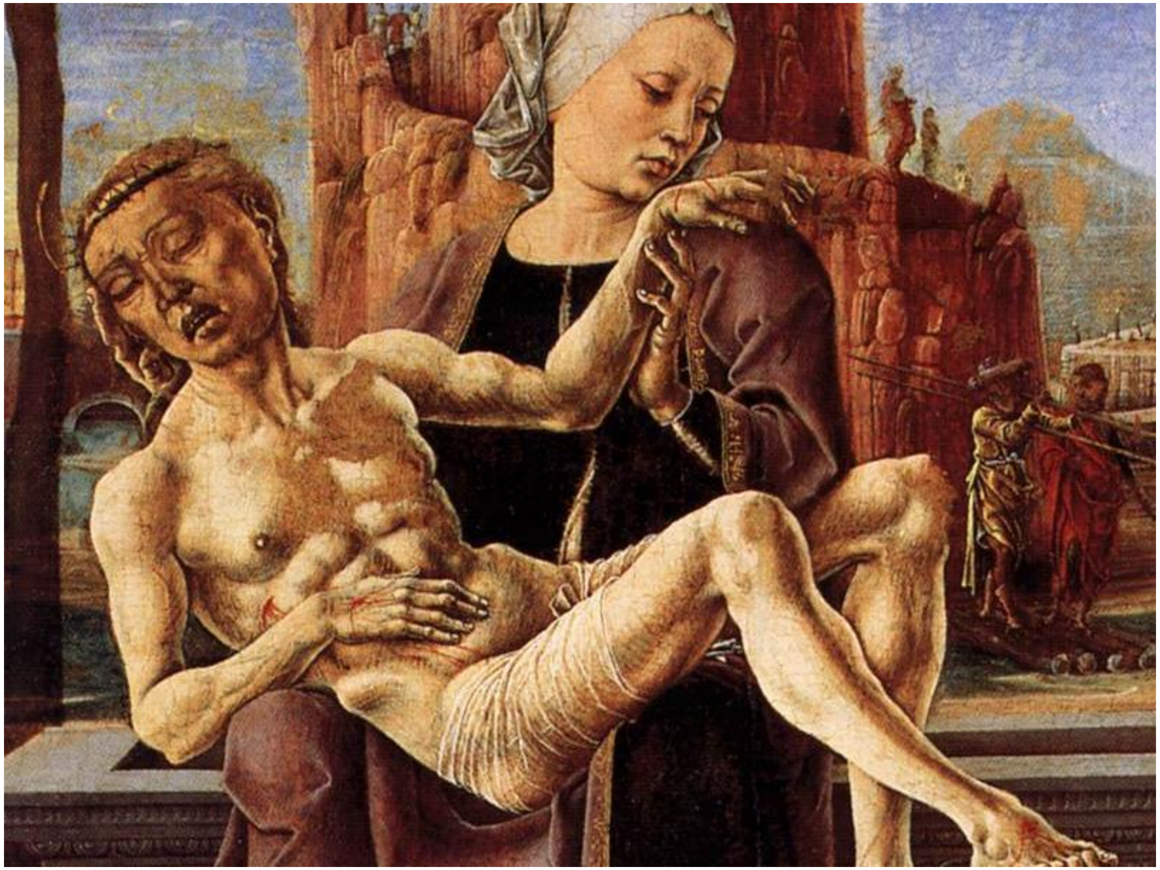
- Why?
  - right (imagery) and left (analytical) brain hemisphere
- What makes a good one?
  - Informative
  - Esthetically pleasing
  - Often, the right abstraction of the data

How to do it?

Art is good

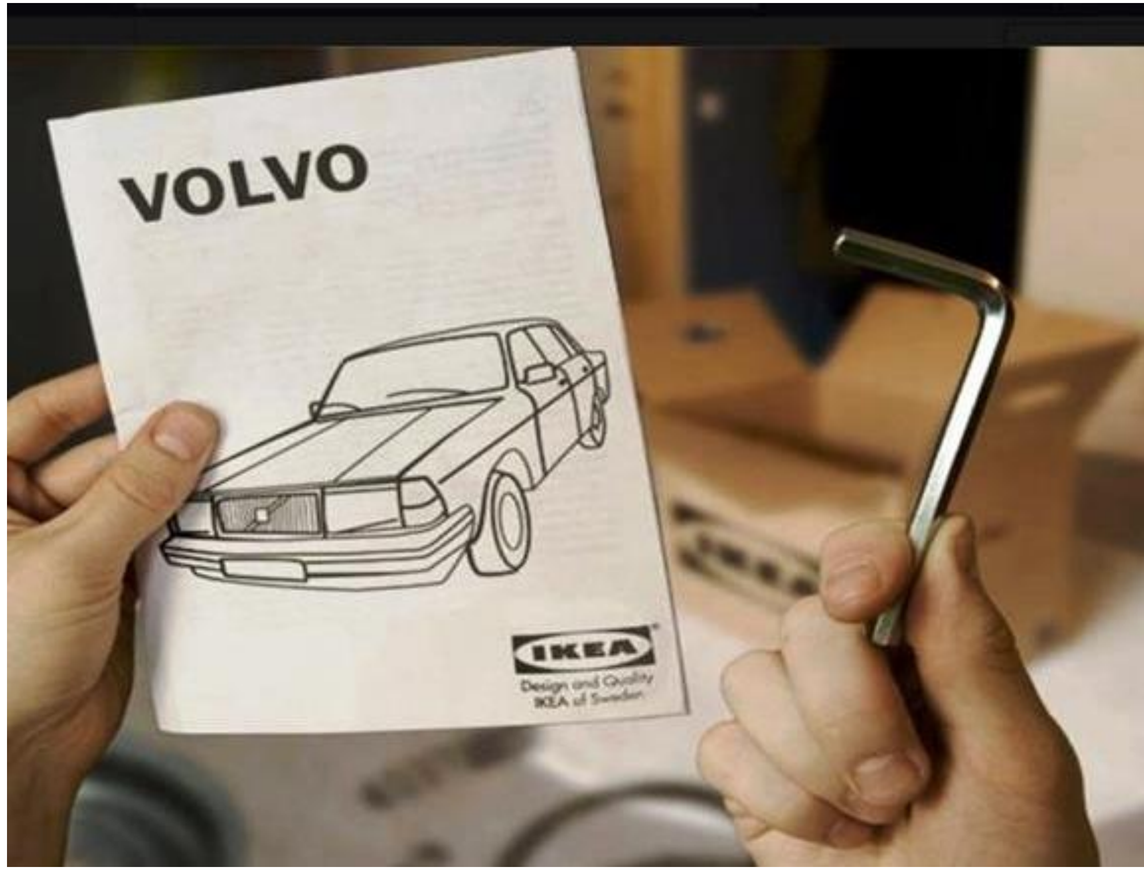
in conveying

complex concepts



Cosme Tura, La Pietà

Advertising makes great visualizations …

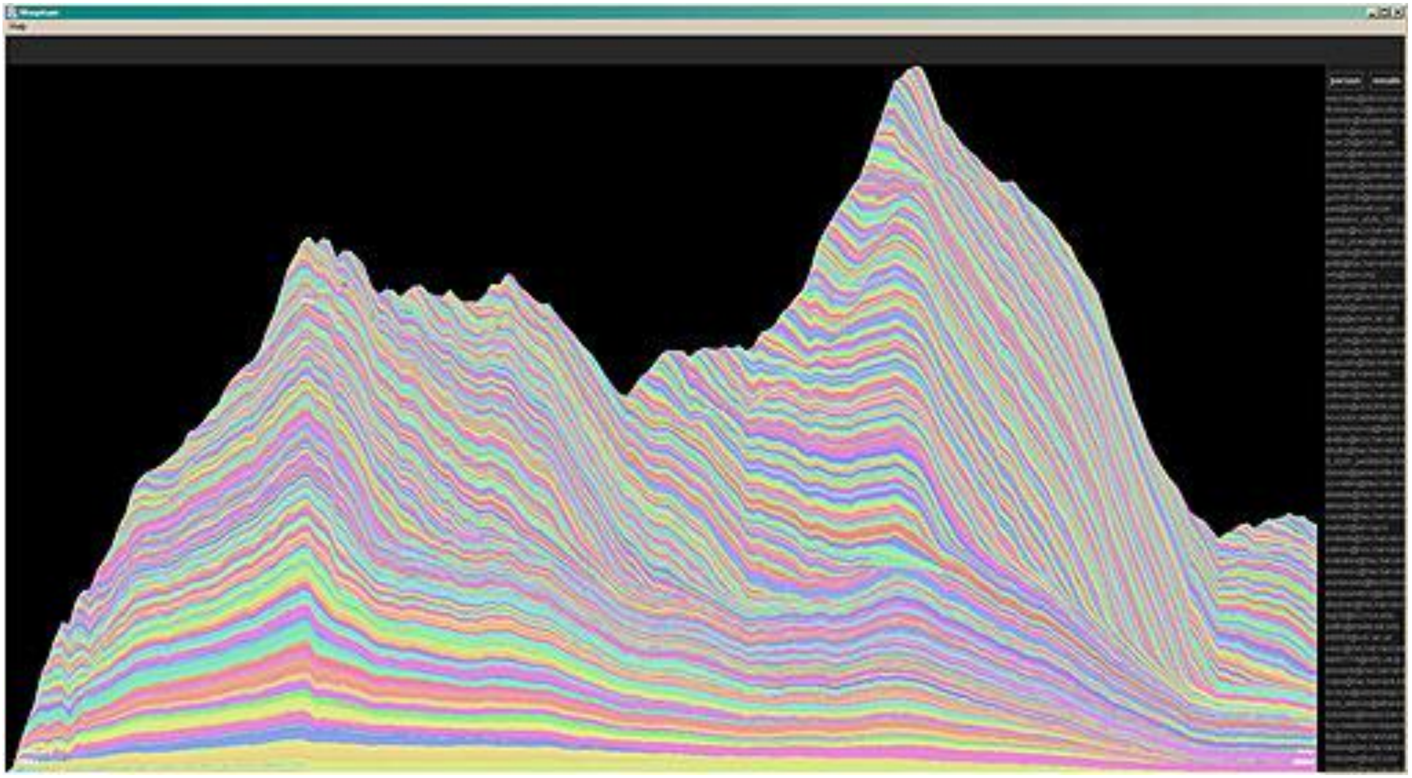


- but art conveys emotions, mental states…
- Visualizing data is different, but still…
  - A good visualization should require no explanation

Word cloud


Visualization
Text
mining
classification
Learning
Matwin
dream
representation
Dream
Visualizing
also
Current
approach
concepts
Active
instances
level
Algorithm
user
corpus
task
BOW
emotions
emotional
complex
art
visualization
highly
tools
dreams
pos/neg
right
analysis
Interactive
high-yield
good
text
vs
data
large-scale
Informative
structures
user-in-the
Analytics
cosine
arrays
Dynamic
extraction
e.g
Dimensions
Semi-supervised
Topic
Semantic
description
chessboard
Labeled
Advertising
Attributes
pleasing
Tagging
project
loop
participants
methods
attributes
hemisphere
eg
protocol
Art
Vera
Joint
detection
linguistic
fish
future
Tura
animation
label
easy
engineering
sadness
message
Binary
Attribute
Naïve
Word
butter
Viergas
come
classifier
resources
Static
Example
Projection
Sazonova
Beyond
Classification
states
imagery
learning
Deep
view
descriptions
http://hint.fm/wind
Kiss
Vectorial
left
model
anger
et
Klimt
Exploration
define
classifying
Information
tonight
judges
makes
Email
ECAI-PAIS
understand
Relatedness
oil
Computer
Distance
bank
Gaussian
set
al
much
Caron
preprocessing
conveying
today
mountain
etc
LIWC
Munch
Sentinet
features
Pietà
SVM
talk
progression
net
conveys
discovery
labels
using
brain

Email “mountain” [F. Viergas]



Dimensions of visualization

- Final show vs Exploration
- Static vs Interactive (eg. Drilling)
- German political donations [G. Aisch]

Privacy and Data Mining

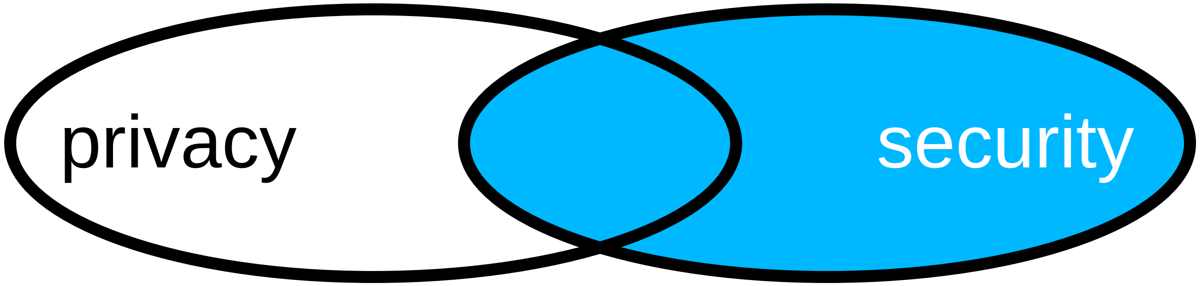
- Why privacy??
- Classification of Privacy-preserving Data Mining research (PPDM)
- Examples of current PPDM work
- Challenges

Why privacy and data mining?…

- Like any technology can be used for « good » and « bad » purposes …
- It's Computer Science that has developed these tools, so…
- A moral obligation to develop solutions that will alleviate [potential] abuses and problems

Privacy

- „fuzzy”, over-general concept
  - legal
  - economic
- Security?



Privacy

- Freedom from being watched ("*to be left alone*")
- …being able to control who knows what about us, and when [Moor]

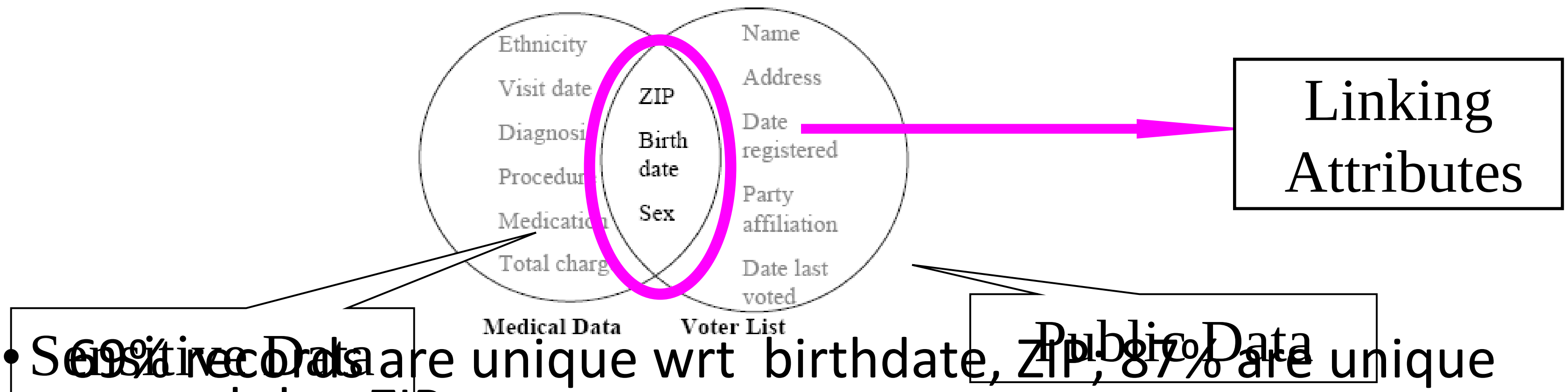
Privacy

- A CS « perspective»
  - I am a database
  - Privacy is the ability to control the *views*
- Threats to privacy due to:
  - The Internet
  - Distributed databases
  - Data mining
- « greased » data

…more precisely

- Privacy preservation: what does that mean?
- Given a table of instances (rows), we cannot associate any instance with a given person
- Naive anonymization…
- …is not sufficient, due to pseudo-identifiers

- L. Sweeney published this « attack » in 2001:
- **<u>anoymized</u>** (*de-linked*) health records of all 135,000 employees+families of the state of Massachussetts was placed on-line
- Electoral list of Cambridge, MA – bought for $20 (54 805 people)



- 69% records are unique wrt birthdate, ZIP; 87% are unique wrt to bday, ZIP, sex…
- Governor's health records were identified
- …naive anonymization is not sufficient

Other privacy fiascos

- AOL search engine queries published

  2006
- Netflix publicly released a data set containing movie ratings of 500,000 Netflix subscribers *between December* 1999 and December 2005.
- By matching no more than 8 movie ratings and approximate dates, 96% of subscribers can be uniquely identified.

In statistics

- Statistical Disclosure Control
- A table is published, and the whole table has to be protected
- Risk/quality dilemma
- SDC ignores the use of the table
  - Classification
  - Associations
  - Distributed data

Privacy-preserving Data Mining PPDM

- Data sharing
- Data publishing
- Cloud
- Two main dimensions:
  - What is being protected: data, results?
  - Data centralized or distributed?

PPDM - dimensions

	Data centralized	Data distributed
Protecting the data	•generalization/suppression [Sweeney] •randomization [Du]/perturbation [Aggrawal]	•Horizontal/vertical: SMC-based [Clifton], •Homomorphic encryption [Wright], [Zhang Matwin]
Protecting the results	*k*-anonymization of results :[Gianotti/Pedreschi]	[Jiang, Atziori], [Felty, Matwin]

Privacy Goal: *k*-Anonymity

- Quasi-identifier (QID): The set of re-identification attributes.
- *k*-anonymity: Each record cannot be distinguished from at least *k-1* other records in the table wrt *QID*. [Sween98]

Raw patient table			
Job	**Sex**	**Age**	**Disease**
Engineer	Male	36	Fever
Engineer	Male	38	Fever
Lawyer	Male	38	Hepatitis
Musician	Female	30	Flu
Musician	Female	30	Hepatitis
Dancer	Female	30	Hepatitis
Dancer	Female	30	Hepatitis

3-anonymous patient table			
Job	**Sex**	**Age**	**Disease**
Professional	Male	[36-40]	Fever
Professional	Male	[36-40]	Fever
Professional	Male	[36-40]	Hepatitis
Artist	Female	[30-35]	Flu
Artist	Female	[30-35]	Hepatitis
Artist	Female	[30-35]	Hepatitis
Artist	Female	[30-35]	Hepatitis

Homogeneity Attack on *k*-anonymity

- A data owner wants to release a table to a data mining firm for classification analysis on *Rating*

Job	Country	Child	Bankruptcy	Rating	# Recs
Cook	US	No	Current	0G/4B	4
Artist	France	No	Current	1G/3B	4
Doctor	US	Yes	Never	4G/2B	6
Trader	UK	No	Discharged	4G/0B	4
Trader	UK	No	Never	1G/0B	1
Trader	Canada	No	Never	1G/0B	1
Clerk	Canada	No	Never	3G/0B	3
Clerk	Canada	No	Discharged	1G/0B	1
				Total:	24

- Inference: {Trader,UK} → fired
- Confidence = 4/5 = 80%
- An inference is sensitive if its confidence > threshold.

p-Sensitive k-Anonymity

- for each equivalence class EC there is at least p distinct values for each sensitive attribute
- **Similarity attack** occurs when the values of sensitive attribute in an EC are distinct but have similar sensitivity.

Age	Country	Zip Code	Health Condition
<30	America	142**	HIV
<30	America	142**	HIV
<30	America	142**	Cancer
<30	America	142**	Cancer
>40	Asia	130**	Hepatitis
>40	Asia	130**	Phthisis
>40	Asia	130**	Asthma
>40	Asia	130**	Heart Disease
3*	America	142**	Flu
3*	America	142**	Flu
3*	America	142**	Flu
3*	America	142**	Indigestion

2-Sensitive 4-Anonymity

l-Diversity

- every equivalence class in this table has at least l *well represented* values for the sensitive attribute
- **Distinct *l*-diversity**: the number of distinct values for a sensitive attribute in each equivalence class to be at least l.
- l -Diversity may be difficult and unnecessary to achieve and it may cause a huge information loss.

	Non-Sensitive			Sensitive
	Zip Code	Age	Nationality	Condition
1	1305*	≤ 40	$*$	Heart Disease
4	1305*	≤ 40	$*$	Viral Infection
9	1305*	≤ 40	$*$	Cancer
10	1305*	≤ 40	$*$	Cancer
5	1485*	> 40	$*$	Cancer
6	1485*	> 40	$*$	Heart Disease
7	1485*	> 40	$*$	Viral Infection
8	1485*	> 40	$*$	Viral Infection
2	1306*	≤ 40	$*$	Heart Disease
3	1306*	≤ 40	$*$	Viral Infection
11	1306*	≤ 40	$*$	Cancer
12	1306*	≤ 40	$*$	Cancer

3-diverse data [4]

t-closeness

- An equivalence class EC is said to have t-closeness if the distance between the distribution of a sensitive attribute in this class and the distribution of the attribute in the whole table is no more than a threshold *t*. [5].

	ZIP Code	Age	Salary	Disease
1	4767*	≤ 40	3K	gastric ulcer
3	4767*	≤ 40	5K	stomach cancer
8	4767*	≤ 40	9K	pneumonia
4	4790*	≥ 40	6K	gastritis
5	4790*	≥ 40	11K	flu
6	4790*	≥ 40	8K	bronchitis
2	4760*	≤ 40	4K	gastritis
7	4760*	≤ 40	7K	bronchitis
9	4760*	≤ 40	10K	stomach cancer

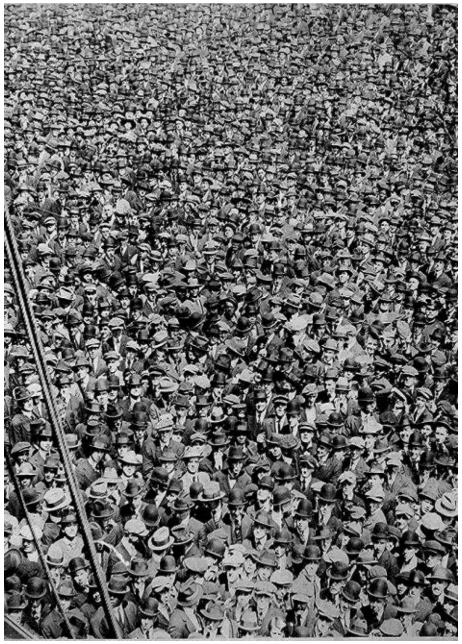
0.167-closeness w.r.t. salary and
0.278-closeness w.r.t. Disease[5]

- It solves the attribute disclosure problems of l-diversity, i.e. skewness attack and similarity attack, [6]

Two basic approaches

camouflage

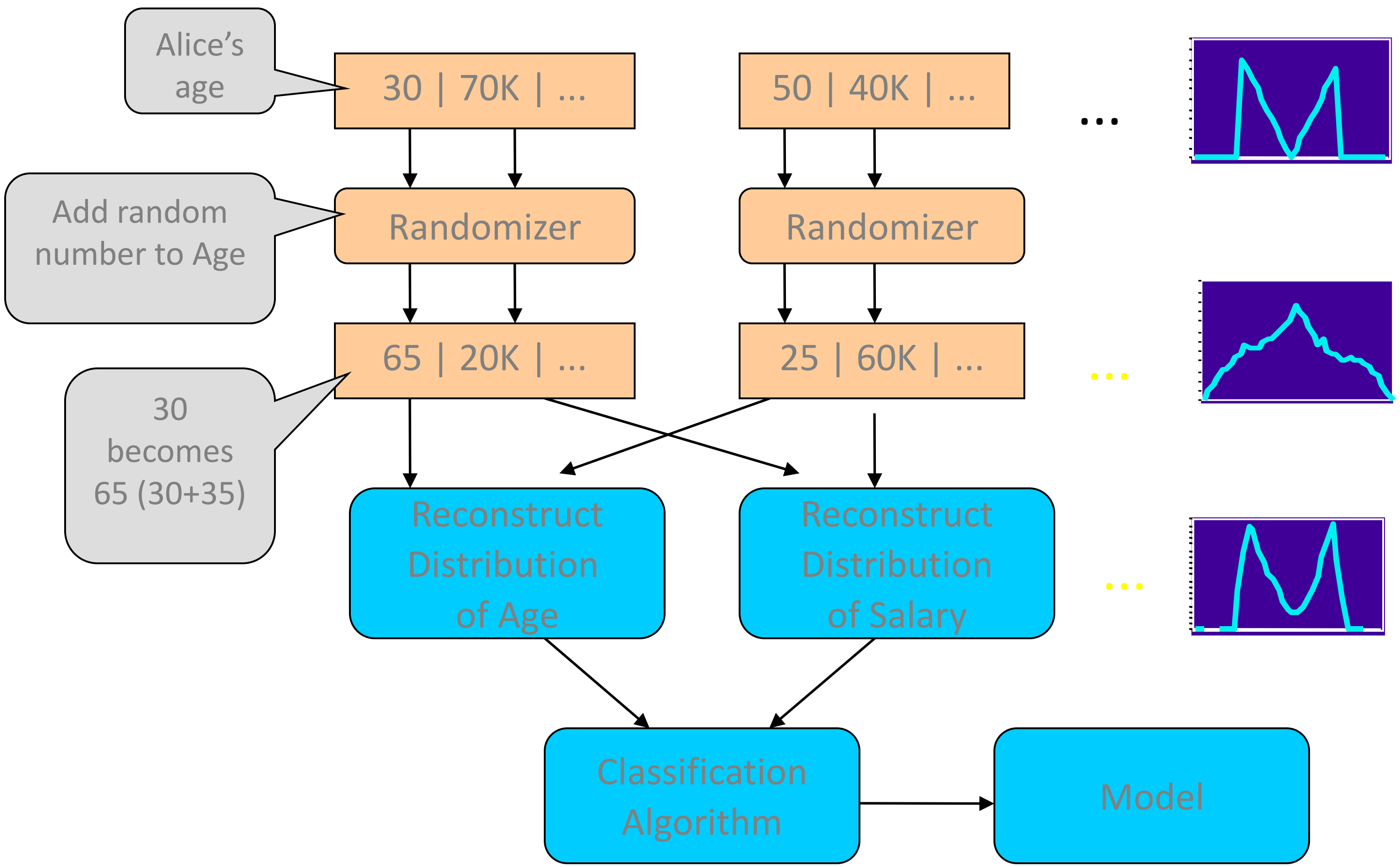
hiding in the crowd



Data modification/perturbation

k-anonymization

Randomization

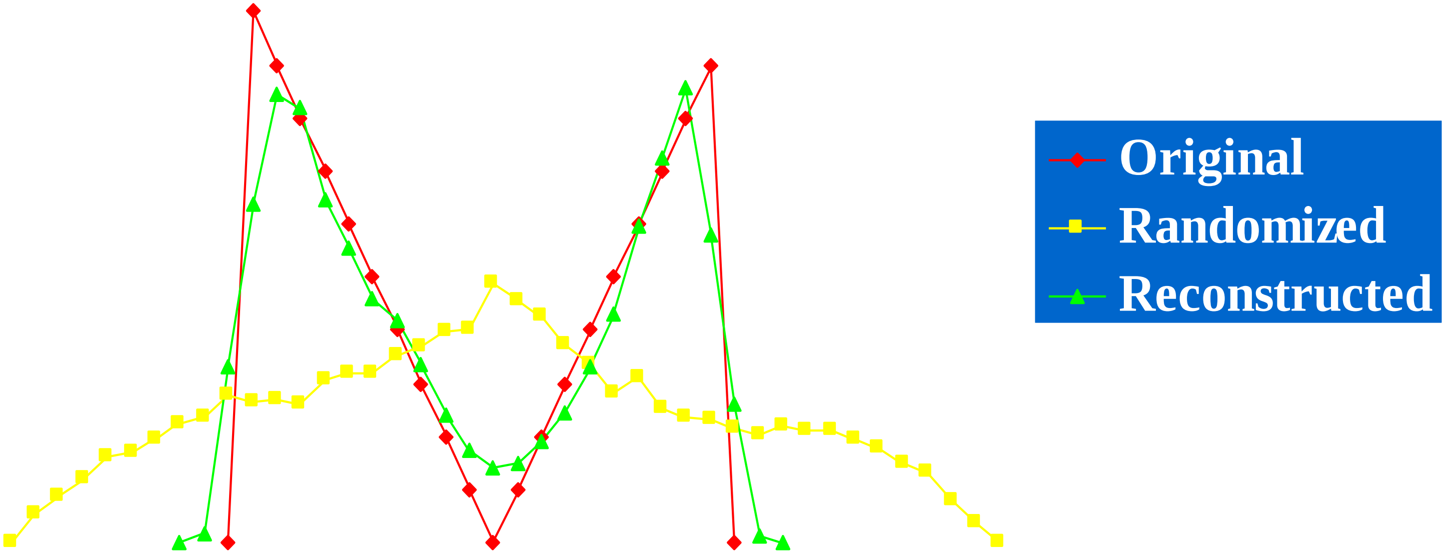


Reconstruction (*linking*)

- initial (confidential) values $x_1, x_2, \ldots, x_n$ have an (uknown) distribution X
- For protection, we perturb them with values $y_1, y_2, \ldots, y_n$ with a *known* distribution Y
- given
  - $x_1+y_1, x_2+y_2, \ldots, x_n+y_n$
  - distribution Y

  Find an estimation of the distribution X.

Works well



privacy measures

- For modification methods
- First – wrt the interval to which we generalize a value
- We inject ”noise” with a random variable *A* with distribution *f*
- The privacy measure is

$$\Pi(A) = 2^{-\int_{\Omega_A} f_A(a)\log_2 f_A(a)\,da}$$

- We measure entropy

Differential privacy

- The desideratum: “access to a database should not enable one to learn anything about individual that could not be learned without access” [Dalenius 77]: simlar to semantic security of Goldwasser & Micali
- Impossible because of auxiliary knowledge *(AK*)*:* database of avg height of people of different nationalities + *AK* = SM is 1 cm shorter than avg Polish male

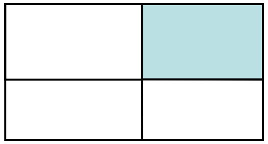
Differential privacy cont'd

- A randomized function K gives ε -differential privacy if for all data sets D_1 and D_2 differing on at most one element, and all $S \subseteq Range(K)$,
- $Pr[K(D_1) \in S] \leq \exp(\varepsilon) \times \Pr[K(D_2) \in S]$
- A ***relative*** guarantee of non-disclosure: any disclosure is as likely whether or not the individual participates in *D*
- *K* is a protection ("sanitization") scheme, $\in S$ represents a query about a database

Differential privacy cont’d

- For every pair of inputs that differ in one value
- For every output
- Adversary should not be able to distinguish between any D1and D2 based on any O:

$$\log\left[\frac{\Pr(D_1 \to O)}{\Pr(D_2 \to O)}\right] < \varepsilon(\varepsilon > 1)$$



Distributed data

- Vehicle/accident data
- To discover the causes of accidents we need to know the attributrs of different components from different manufacturers (brakes, tires)
- They will nolt disclose these values in the open
- **<u>Vertical</u>** partition

Distributed data

- A medical study carried out in several hospitals
- Would like to *merge* the data for bigger impact of results (results on 20 000 patients instead of 5 000 each)
- For legal reasons, cannot just share then open data
- Horizontal partition

Association Rule Mining Algorithm [Agrawal et al. 1993]

1. L_1 = large 1-itemsets
2. for $(k = 2; L_{k-1} \neq \phi; k{+}{+})$ do begin
3. $\quad C_k = \text{apriori-gen}(L_{k-1})$
4. for all candidates $c \in C_k$ do begin
5. compute ***c.count***
6. end
7. $L_k = \{c \in C_k \mid c.count \geq minsup\}$
8. end
9. Return $L = \bigcup_k L_k$

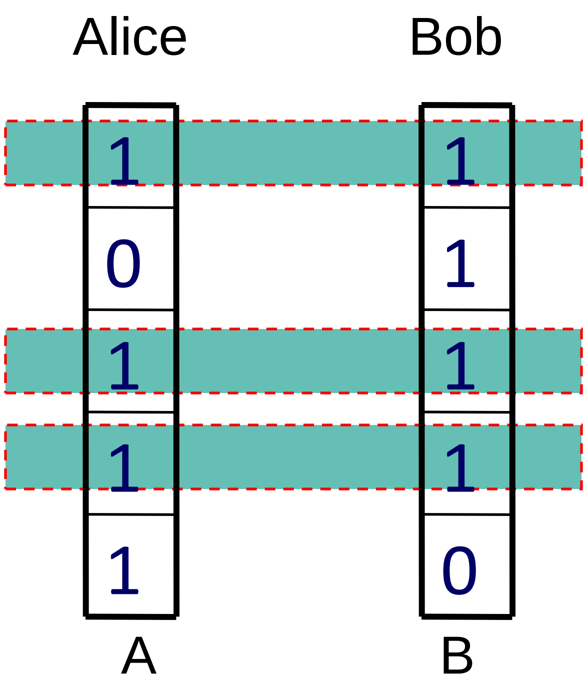
c.count is the frequency of an *itemset*.

to compute frequency, we need access to values of attributes belonging to different parties

Example

- c.count is the scalar product.
- A = Alice's attribute vector, B = Bob'
- AB is a candidate frequent itemset
- c.count = A • B = 3.

- How to perform the scalar product preserving the privacy of Alice and Bob?

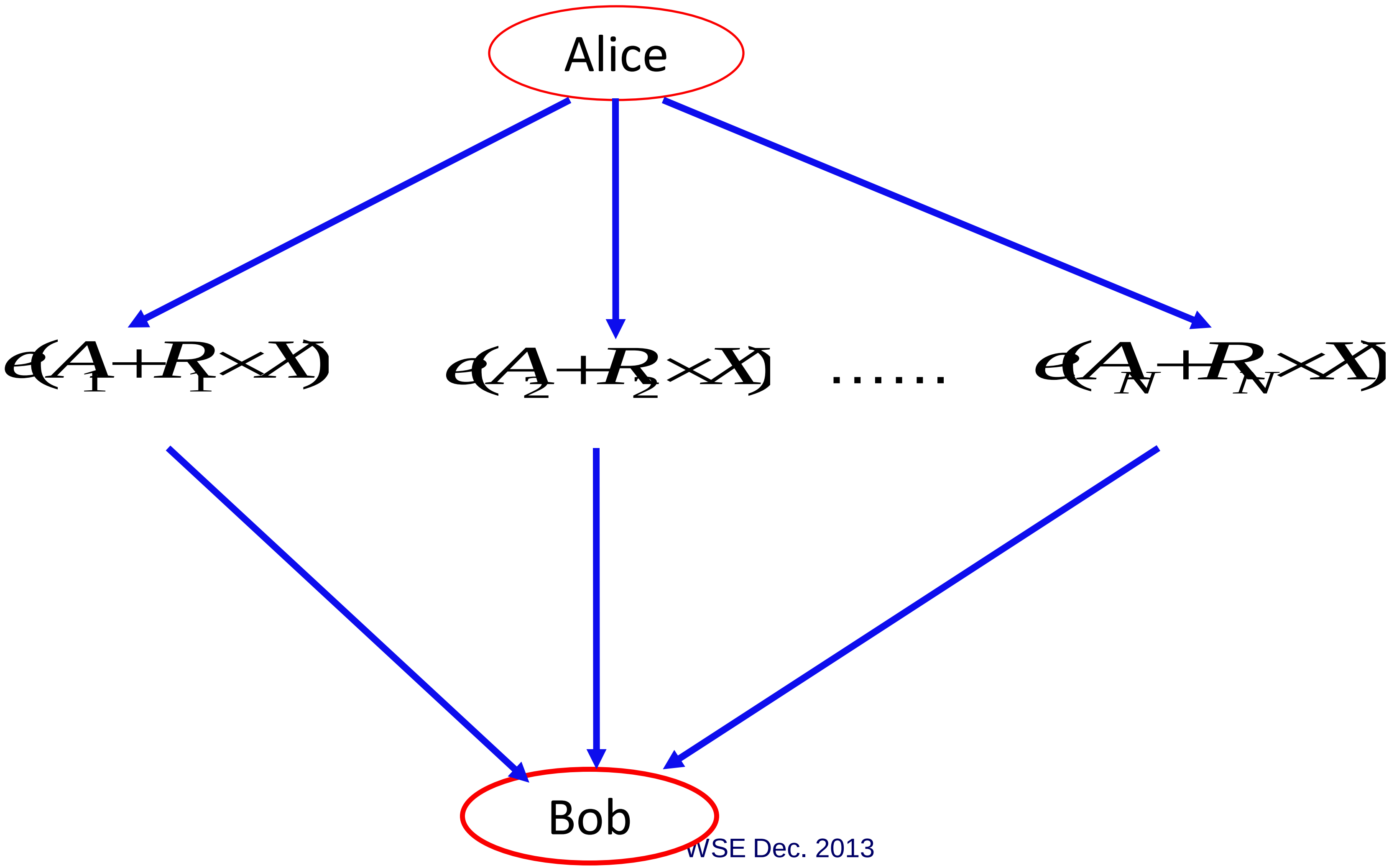


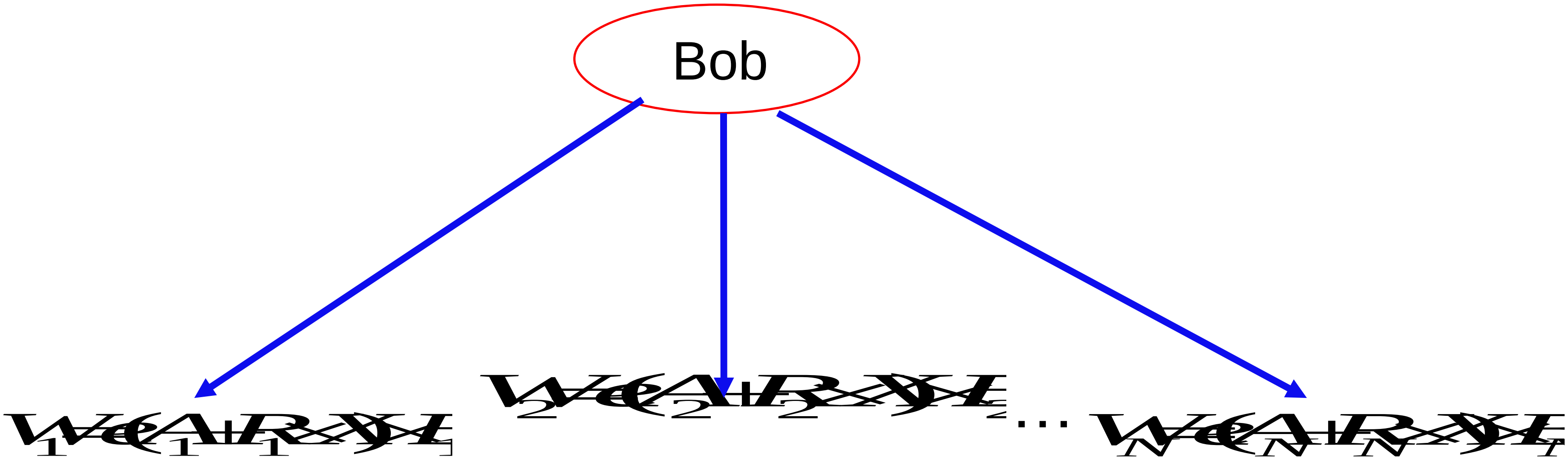
Homomorphic Encryption

[Paillier 1999]

- Privacy-preserving protocol based on the concept of homomorphic encryption
- The homomorphic encryption property is

- *e* is an encryption function $e(m_i) \neq 0$


Alice
$e(A_1 + R_1 \times X)$
$e(A_2 + R_2 \times X)$
......
$e(A_N + R_N \times X)$
Bob



Bob computes

encrypts , sends to Alice

Last stage

- Alice decrypts W and computes modulo X.

$$count$$

$$= ((A_1 + A_2 + \cdots + A_j) + (R_1 + R_2 + \cdots + R_j) X) \bmod X$$

$$(A_1 + A_2 + \cdots + A_j) \bmod X$$

$$((R_1 + R_2 + \cdots + R_j) X) \bmod X$$

- She obtains $A_1 + A_2 + \cdots + A_j$ for these A_j whose corresponding B_j are not 0, which is = c.count
- Privacy analysis

Now looking at data mining results…

Can data mining results reveal personal information? In some cases, yes: [Atzori et al. 05]:

An association rule :

$$a_1 \wedge a_2 \wedge a_3 \Rightarrow a_4 \quad [sup = 80, conf = 98.7\%]$$

Means that $sup(\{a_1, a_2, a_3, a_4\}) = 80$

So $sup(\{a_1, a_2, a_3\}) = \frac{sup(\{a_1, a_2, a_3, a_4\})}{0.987} = \frac{80}{0.987} = 81.05$

And $a_1 \wedge a_2 \wedge a_3 \wedge \neg a_4$ has support =1, and identifies a person!

Protecting data mining results

- A *k-anonymous patterns* approach and an algorithm (*inference channels*) detect violations of *k*-anonymity of results

Discrimination and data mining

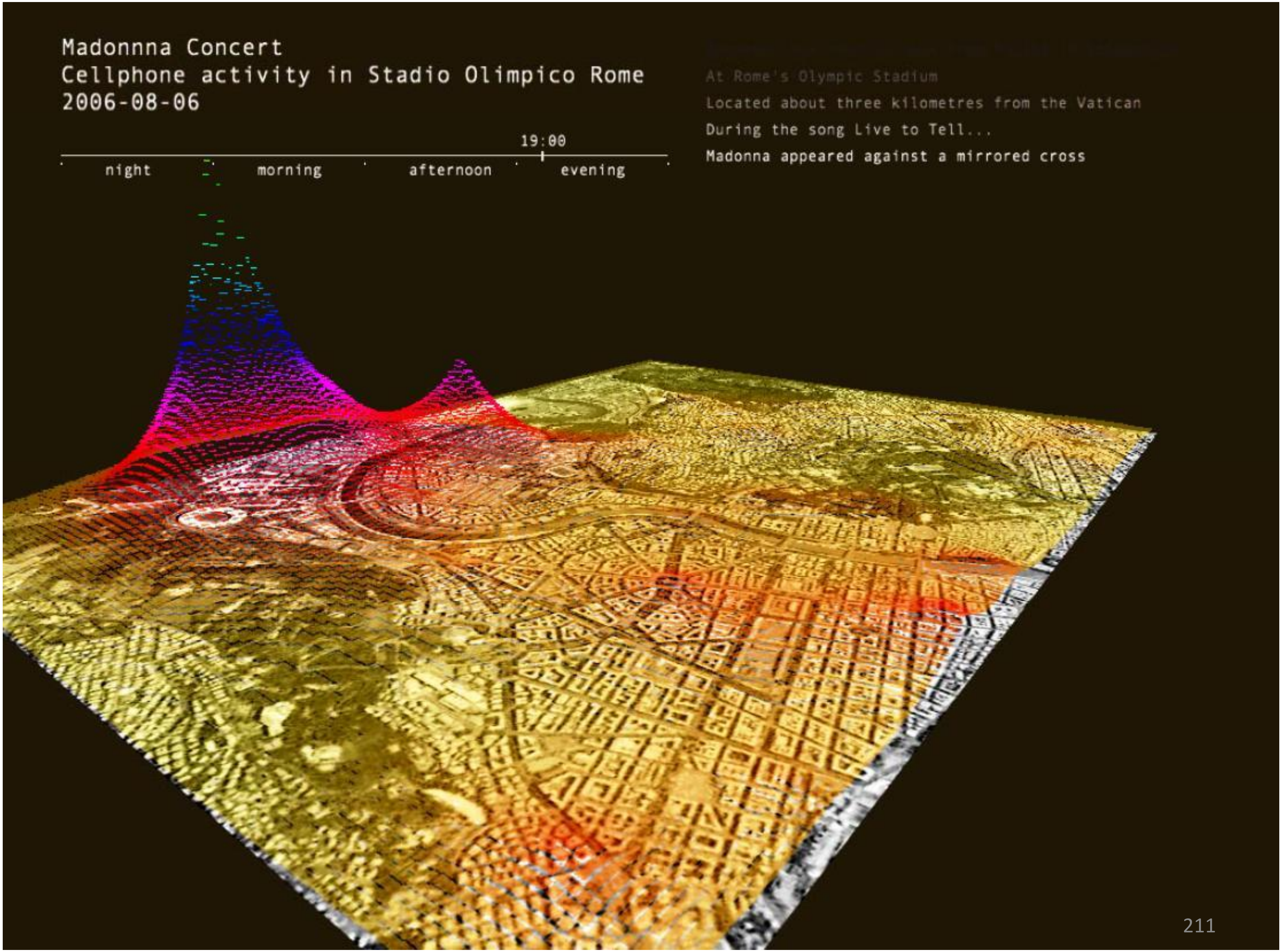
- [Pedreschi et al 07] shows how DM results can lead to discriminatory rules
- In fact, DM's goal is discrimination (between different sub-groups of data)
- They propose a measure of potential discrimination with lift : to what extent a sensitive is more assigned by a rule to a sensitive group than to an average group

Other challenges

- Privacy and social networks
- Privacy definition – where to look for inspiration (economics?)
- Text data – perturbation/anonymization methods don't work
- Medical data: trails [Malin], privacy of longitudinal data
- Mobile data -

GeoPKDD

- European project on Geographic Privacy-aware Knowledge Discovery and Delivery
- Data from GSM/UMTS and GPS


Madonnna Concert
Cellphone activity in Stadio Olimpico Rome
2006-08-06
19:00
night
morning
afternoon
evening
At Rome's Olympic Stadium
Located about three kilometres from the Vatican
During the song Live to Tell...
Madonna appeared against a mirrored cross

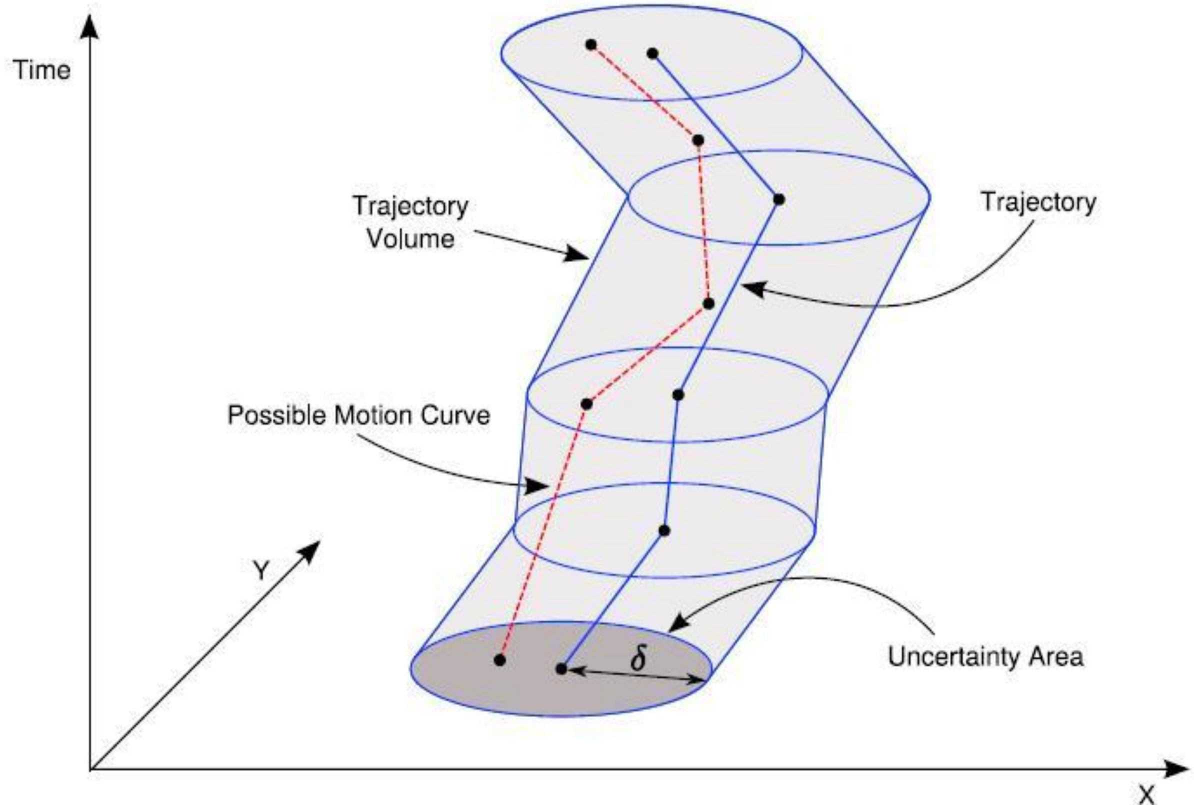
First obtaining spatio-temporal trajectories, then patterns

Trajectory = sequence of points visiteddans in a temporal order

pattern= set of frequent trajectories with similar transition times

Privacy of spatio-temporal data

- Modify the data in such a way each trajectory be indistinguishable from k other trajectories
- … by minimizing distorsion introduced into the data



Conclusion

- A major challenge for database/data mining research
- Lots of interesting contributions/papers, but lack of a systematic framework
- …?

What is Data Science for us

- Data Science = making big data accessible to decision makers
- extraction of insight from data is easier when the decision maker can **interact** with the data
- Hence focus of Data Science training beyond data and text mining, towards interaction and **vizualization**
- Need for HPC to support interaction

Scan

- As of May 2013, in Canada no undergraduate specialization in Data Science or Data Analytics
- ….but no doubt some programs are developed

Undergraduate CS Data Science curriculum

- 2nd year
  - Probability and stats
  - Databases
- 3rd year
  - Web intelligence
  - Digital media
  - Computer graphics with visualization

- 4th year
  - Cloud computing
  - HPC
  - Data science
  - Electives among:

    Machine learning applied to large scale problems

    Natural language processing

    Data Mining and Warehousing

Web intelligence

- Information retrieval,
- web crawlers,
- association rule mining,
- supervised learning (decision trees, k-nearest-neighbour classifiers, Naïve Bayes, generative models for text, support vector machines),
- unsupervised learning (k-means clustering, hierarchical agglomerative clustering),
- Natural language processing, automatic term recognition, sentiment classification, visual text analytics.

High Performance Computing

- Sequential Programming: code optimization, cache effects, I/O issues, compiler issues, vectorization, floating point issues, benchmarking and profiling practises;
- Multithreaded
- programming: shared memory multiprocessors, thread libraries, OpenMP, loop parallelization, untangling dependencies;
- Parallel programming: distributed memory multiprocessors, taxonomies, performance measures, clusters, MPI, performance evaluation, and parallel algorithms.

Cloud computing

- Cloud computing -
- basic concepts and terminology; Benefits vs. risks and costs;
- Cloud delivery and deployment models;
- Virtualization; Cloud infrastructure mechanisms: logical network perimeter, virtual server, cloud storage server, cloud usage monitor, resource replication; Specialized cloud mechanisms;
- Dynamic scaling;
- Google aps, Amazon web services, MS cloud service;
- Storage and computing models for Big Data (relational and non-relational storage models, Hadoop - MapReduce

Data Science

- Data model fundamentals;
- Data acquisition, ethics;
- Project objectives and planning;
- Analytical and Predictive model selecting
- Algorithmic approaches; Selecting models, converting data;
- Model evaluation;
- Model implementation and implementation issues;
- Communicating actionable, validated data analytical results;
- Managing organization project expectations

Graduate training in Big Text

- A graduate CS specialization
- Meant to attract students to Big Text
- An additional qualification or a stand-alone 1-yr GradCertificate degree
- Joint initiative with Simon Fraser's VIVA and Université de Montréal TALI

TRIBE: Training in Big Text Data: five pillars

- Structured curriculum
- Project
- Industrial internship
- Student mobility
- Respect for data privacy

Courses

- Data and text mining
- Applied computational linguistics with a bilingual data focus
- Data and Information Visualization and HCI
- High-performance Computing and the Cloud
- Professional practicum

Professional practicum course

- "soft skills" that we believe are particularly important for data scientists, i.e.,
  - data privacy and professional ethics,
  - Communications: business presentations, proposal writing, etc.
  - Project management
  - Intro to entrepreneurship, intellectual property, etc.

- focused training “camps” on particular tools students will use (e.g., R)
- Some courses delivered in a condensed format

How – IBDA - our model

- You have the data and a question/problem
- We help you answering it using state-of-the-art tools we have
- Do all this in a privacy-respectful way
- We are interested in the research aspect of such projects
- We train students through thesis topics inspired by your R&D needs

Discussion

- Many others are involved in similar initiatives
- Good time for discussion
- Time and experience will teach us all how what makes a really good Data Science program