

INFT/CSI 979
Statistical Data Mining
of Massive Data



Edward J. Wegman



Administrative

Contact Information



Edward J. Wegman

155 Science-Technology 2

Office Hours: 3:30 – 4:30 pm on
Mondays and by appointment

Phone: (703) 993-1691 (good luck)

Email: ewegman@gmu.edu (best bet)

My Stuff on the Web



Lecture Notes from Fall 96

<http://www.galaxy.gmu.edu/stats/syllabi/inft979.wegman.html>

Short Course Lecture Notes from Spring 2000

<ftp://www.galaxy.gmu.edu/pub/PowerPoint/StatisticalDataMining.ppt>

Lecture Notes from Spring 2001

<http://www.galaxy.gmu.edu/stats/syllabi/INFT979.spring2001.html>

Outline of Lecture



- Complexity
- Data Mining: What is it?
- Data Mining: Some statistical strategies
- Data Mining: Computational architectures

Complexity



Descriptor	Data Set Size in Bytes	Storage Mode
Tiny	10^2	Piece of Paper
Small	10^4	A Few Pieces of Paper
Medium	10^6	A Floppy Disk
Large	10^8	Hard Disk
Huge	10^{10}	Multiple Hard Disks e.g. RAID Storage
Massive	10^{12}	Robotic Magnetic Tape Storage Silos

The Huber Taxonomy of Data Set Sizes

Complexity



$O(r), O(n^{1/2})$

Plot a scatterplot

$O(n)$

Calculate means, variances, kernel density estimates

$O(n \log(n))$

Calculate fast Fourier transforms

$O(nc)$

Calculate singular value decomposition of an rc matrix; solve a multiple linear regression

$O(n^2)$

Solve most clustering algorithms.

Algorithmic Complexity

Complexity

Table 2: Number of Operations for Algorithms of Various Computational Complexities and Various Data Set Sizes

n	$n^{1/2}$	n	$n \log(n)$	$n^{3/2}$	n^2
<i>tiny</i>	10	10^2	2×10^2	10^3	10^4
<i>small</i>	10^2	10^4	4×10^4	10^6	10^8
<i>medium</i>	10^3	10^6	6×10^6	10^9	10^{12}
<i>large</i>	10^4	10^8	8×10^8	10^{12}	10^{16}
<i>huge</i>	10^5	10^{10}	10^{11}	10^{15}	10^{20}

Complexity

Table 4: Computational Feasibility on a Pentium PC
10 megaflop performance assumed

n	$n^{1/2}$	n	$n \log(n)$	$n^{3/2}$	n^2
<i>tiny</i>	10^{-6} seconds	10^{-5} seconds	2×10^{-5} seconds	.0001 seconds	.001 seconds
<i>small</i>	10^{-5} seconds	.001 seconds	.004 seconds	.1 seconds	10 seconds
<i>medium</i>	.0001 seconds	.1 seconds	.6 seconds	1.67 minutes	1.16 days
<i>large</i>	.001 seconds	10 seconds	1.3 minutes	1.16 days	31.7 years
<i>huge</i>	.01 seconds	16.7 minutes	2.78 hours	3.17 years	317,000 years

Complexity

**Table 5: Computational Feasibility on a Silicon Graphics Onyx Workstation
300 megaflop performance assumed**

n	$n^{1/2}$	n	$n \log(n)$	$n^{3/2}$	n^2
<i>tiny</i>	3.3×10^{-8} seconds	3.3×10^{-7} seconds	6.7×10^{-7} seconds	3.3×10^{-6} seconds	3.3×10^{-5} seconds
<i>small</i>	3.3×10^{-7} seconds	3.3×10^{-5} seconds	1.3×10^{-4} seconds	3.3×10^{-3} seconds	.33 seconds
<i>medium</i>	3.3×10^{-6} seconds	3.3×10^{-3} seconds	.02 seconds	3.3 seconds	55 minutes
<i>large</i>	3.3×10^{-5} seconds	.33 seconds	2.7 seconds	55 minutes	1.04 years
<i>huge</i>	3.3×10^{-4} seconds	33 seconds	5.5 minutes	38.2 days	10,464 years

Complexity

**Table 6: Computational Feasibility on an Intel Paragon XP/S A4
4.2 gigaflop performance assumed**

n	$n^{1/2}$	n	$n \log(n)$	$n^{3/2}$	n^2
<i>tiny</i>	2.4×10^{-9} seconds	2.4×10^{-8} seconds	4.8×10^{-8} seconds	2.4×10^{-7} seconds	2.4×10^{-6} seconds
<i>small</i>	2.4×10^{-8} seconds	2.4×10^{-6} seconds	9.5×10^{-6} seconds	2.4×10^{-4} seconds	.024 seconds
<i>medium</i>	2.4×10^{-7} seconds	2.4×10^{-4} seconds	.0014 seconds	.24 seconds	4.0 minutes
<i>large</i>	2.4×10^{-6} seconds	.024 seconds	.19 seconds	4.0 minutes	27.8 days
<i>huge</i>	2.4×10^{-5} seconds	2.4 seconds	24 seconds	66.7 hours	761 years

Complexity

**Table 7: Computational Feasibility on a Teraflop Grand Challenge Computer
1000 gigaflop performance assumed**

n	$n^{1/2}$	n	$n \log(n)$	$n^{3/2}$	n^2
<i>tiny</i>	10^{-11} seconds	10^{-10} seconds	2×10^{-10} seconds	10^{-9} seconds	10^{-8} seconds
<i>small</i>	10^{-10} seconds	10^{-8} seconds	4×10^{-8} seconds	10^{-6} seconds	10^{-4} seconds
<i>medium</i>	10^{-9} seconds	10^{-6} seconds	6×10^{-6} seconds	.001 seconds	1 second
<i>large</i>	10^{-8} seconds	10^{-4} seconds	8×10^{-4} seconds	1 second	2.8 hours
<i>huge</i>	10^{-7} seconds	.01 seconds	.1 seconds	16.7 minutes	3.2 years

Complexity

**Table 8: Types of Computers for Interactive Feasibility
Response Time < 1 second**

n	$n^{1/2}$	n	$n \log(n)$	$n^{3/2}$	n^2
<i>tiny</i>	<i>Personal Computer</i>	<i>Personal Computer</i>	<i>Personal Computer</i>	<i>Personal Computer</i>	<i>Personal Computer</i>
<i>small</i>	<i>Personal Computer</i>	<i>Personal Computer</i>	<i>Personal Computer</i>	<i>Personal Computer</i>	<i>Super Computer</i>
<i>medium</i>	<i>Personal Computer</i>	<i>Personal Computer</i>	<i>Personal Computer</i>	<i>Super Computer</i>	<i>Teraflop Computer</i>
<i>large</i>	<i>Personal Computer</i>	<i>Workstation</i>	<i>Super Computer</i>	<i>Teraflop Computer</i>	---
<i>huge</i>	<i>Personal Computer</i>	<i>Super Computer</i>	<i>Teraflop Computer</i>	---	---

Complexity

**Table 9: Types of Computers for Feasibility
Response Time < 1 week**

n	$n^{1/2}$	n	$n \log(n)$	$n^{3/2}$	n^2
<i>tiny</i>	<i>Personal Computer</i>	<i>Personal Computer</i>	<i>Personal Computer</i>	<i>Personal Computer</i>	<i>Personal Computer</i>
<i>small</i>	<i>Personal Computer</i>	<i>Personal Computer</i>	<i>Personal Computer</i>	<i>Personal Computer</i>	<i>Personal Computer</i>
<i>medium</i>	<i>Personal Computer</i>	<i>Personal Computer</i>	<i>Personal Computer</i>	<i>Personal Computer</i>	<i>Personal Computer</i>
<i>large</i>	<i>Personal Computer</i>	<i>Personal Computer</i>	<i>Personal Computer</i>	<i>Personal Computer</i>	<i>Teraflop Computer</i>
<i>huge</i>	<i>Personal Computer</i>	<i>Personal Computer</i>	<i>Personal Computer</i>	<i>Super Computer</i>	---

Complexity

Table 10: Transfer Rates for a Variety of Data Transfer Regimes

<i>n</i>	<i>standard ethernet</i> 10 mega-bits/sec	<i>fast ethernet</i> 100 mega-bits/sec	<i>hard disk transfer</i> 2027 kilo-bytes/sec	<i>cache transfer @ 200 megahertz</i>
	1.25×10^6 bytes/sec	1.25×10^7 bytes/sec	2.027×10^6 bytes/sec	2×10^8 bytes/sec
<i>tiny</i>	8×10^{-5} seconds	8×10^{-6} seconds	4.9×10^{-5} seconds	5×10^{-6} seconds
<i>small</i>	8×10^{-3} seconds	8×10^{-4} seconds	4.9×10^{-3} seconds	5×10^{-5} seconds
<i>medium</i>	.8 seconds	.08 seconds	.49 seconds	5×10^{-3} seconds
<i>large</i>	1.3 minutes	8 seconds	49 seconds	.5 seconds
<i>huge</i>	2.2 hours	13.3 minutes	1.36 hours	50 seconds

Complexity

Table 11: Resolvable Number of Pixels Across Screen for Several Viewing Scenarios

	19 inch monitor @ 24 inches	25 inch TV @ 12 feet	15 foot screen @ 20 feet	immersion
Angle	39.005°	9.922°	41.112°	140°
5 seconds of arc resolution (Valyus)	28,084	7,144	29,601	100,800
1 minute of arc resolution	2,340	595	2,467	8,400
3.6 minute of arc resolution (Wegman)	650	165	685	2,333
4.38 minutes of arc resolution (Maar 1)	534	136	563	1,918
.486 minutes of arc/foveal cone (Maar 2)	4,815	1,225	5,076	17,284

Complexity



Scenarios

Typical high resolution workstations,

$$1280 \times 1024 = 1.31 \times 10^6 \text{ pixels}$$

Realistic using Wegman, immersion, 4:5 aspect ratio,

$$2333 \times 1866 = 4.35 \times 10^6 \text{ pixels}$$

Very optimistic using 1 minute arc, immersion, 4:5 aspect ratio,

$$8400 \times 6720 = 5.65 \times 10^7 \text{ pixels}$$

Wildly optimistic using Maar(2), immersion, 4:5 aspect ratio,

$$17,284 \times 13,828 = 2.39 \times 10^8 \text{ pixels}$$

Massive Data Sets



One Terabyte Dataset

VS

One Million Megabyte Data Sets

Both difficult to analyze,
but for different reasons.

Massive Data Sets: Commonly Used Language

- Data Mining = DM
- Knowledge Discovery in Databases = KDD
- Massive Data Sets = MD
- Data Analysis = DA

Massive Data Sets



DM \neq MD

DM \neq DA

Even DM + MD \neq DA

1. Computationally Feasible Algorithms
2. Little or No Human Intervention

Data Mining of Massive Datasets



Data Mining is Exploratory Data Analysis with Little or No Human Interaction using Computationally Feasible Techniques, i.e., the Attempt to find Interesting Structure unknown a priori

Statistical Data Mining



Techniques

- Classification
- Clustering
- Neural Networks & Genetic Algorithms
- CART
- Nonparametric Regression
- Time Series: Trend & Spectral Estimation
- Density Estimation, Bumps and Ridges

Massive Data Sets



- Major Issues
 - Complexity
 - Non-homogeneity
- Examples
 - Huber's Air Traffic Control
 - Highway Maintenance
 - Ultrasonic NDE

Massive Data Sets



- Air Traffic Control

- 6 to 12 Radar stations, several hundred aircraft, 64-byte record per radar per aircraft per antenna turn
- megabyte of data per minute

Massive Data Sets



- Highway Maintenance
 - Records of maintenance records and measurements of road quality for several decades
 - Records of uneven quality
 - Records missing

Massive Data Sets



- NDE using Ultrasound
 - Inspection of cast iron projectiles
 - Time series of length 256, 360 degrees, 550 levels = 50,688,000 observations per projectile
 - Several thousand projectiles per day

Massive Data Sets: A Distinction



Human Analysis of the Structure of
Data and Pitfalls

VS

Human Analysis of the Data Itself

- Limits of HVS and computational complexity limit the latter
- Former is the basis for design of the analysis engine

Massive Data Sets



■ Data Types

- Experimental
- Observational
- Opportunistic

■ Data Types

- Numerical
- Categorical
- Image

Massive Data Sets



Thinning vs Binning

- Every time I speak about Massive Data, statistical subsampling is suggested.
- Quantization is engineering's success story
- Binning is statistician's quantization

Massive Data Sets



- Images are quantized in 8 to 24 bits, i.e. 256 to 16 million levels.
- Signals (audio on CDs) are quantized in 16 bits, i.e. 65,536 levels
- Ask a statistician how many bins to use, likely response is a few hundred.
- For a terabyte data set, 10^6 bins

Massive Data Sets



- Using 10^6 bins is computationally and visually feasible
- Fast binning, for data in the range $[a,b]$, and for k bins
$$j = \text{fixed}[k*(x_i-a)/(b-a)]$$
gives the index of the bin for x_i .
- Can be generalized to multidimensions
- Memory requirements drop to 2 times 10^6 - location of bin + # items in bin.

Massive Data Sets



- Binning does not lose fine structure in tails as sampling might.
- Roundoff analysis applies.
- With scale of binning, discretization not likely to be much less accurate than accuracy of recorded data.
- Discretization - finite number of bins implies discrete variables more compatible with categorical data.

Massive Data Sets



- Analysis on a finite subset of the integers has theoretical advantages
 - Analysis is less delicate
 - | different forms of convergence are equivalent
 - Analysis is often more natural since data is already quantized or categorical
 - Graphical analysis of numerical data is not much changed since 10^6 pixels is at limit of HVS

Data Mining: Prototype Systems



- Digital Library
- Digital Data Library
 - Numerical or Symbolic Data
 - Distributed Linked Scientific Databases
 - | Databases Linked to other Databases
 - | Databases Linked to Digital (Text) Libraries

Data Mining: An Architecture for the Problem



The phrase, **siftware**, has its origins in a typographical error (o is next to i on the qwerty keyboard), but in fact massive databases (terabytes and larger) will not simply be one massive data set, but many, many somewhat smaller data sets. A terabyte database could easily be a million 10^6 data sets. However you slice it, this is not something that is feasible for an individual to browse in an afternoon. Thus, data mining software must also be data siftware ... software designed to aid in isolating interesting worthwhile data sets for the researcher to examine.

Data Mining: An Architecture for the Problem



NASA has created a **Dilbert-type** concept to deal with the massive data sets anticipated from the Earth Observing System (EOS). Called the **DAAC, Distributed Active Archive Centers**, NASA manages to encode two oxymorons in a single name (distributed centers and active archives). The DAACs are intended as central repositories for the massive amounts of data expected from EOS. One proposal currently under development for access data in the DAACs is the **Virtual Domain Application Data Center (VDADC)**. The VDADC is designed to accommodate large data sets in the sense that a large number of descriptors (metadata) characterize the data, as opposed to a large quantity of data with relatively few characterizing parameters.

Data Mining: An Architecture for the Problem



- Automated Generation of Metadata
- Query and Search
 - | Client Browser
 - | Expert System for Query Refinement
 - | Search Engine
 - | Reporting Mechanism

Data Mining: An Architecture for the Problem



Automated Generation of Metadata

- **Metadata** that describe file and variable type and organization but has minimal information on scientific content of the data
- In the raw form, a data set and its metadata has minimal usability
- Link the data set to digital objects that are used to index the data set.
- The digital objects become part of the searchable metadata associated with the data set
- The concept is to have a background process, launched either by the database owner or via applet created by the virtual data center examining databases available on the dataweb and doing autonomous data mining

Data Mining: An Architecture for the Problem



Automated Generation of Metadata (continued)

- When a pattern is found in a particular data set, the digital object corresponding to that pattern is made part of the metadata associated with that data set
- Pointers would be added to that metadata pointing to other distributed databases containing the same pattern
- Metadata will be located in the virtual data center and through this metadata, distributed databases will be linked
- Linking is to be done on the fly as data is accumulated in the database
- On existing databases, the background process would run as compute cycles are available.

Data Mining: An Architecture for the Problem



Automated Generation of Metadata (continued)

- Patterns to be searched for are to be generated by one of at least three different methods
 - | empirical or statistical patterns
 - | model-based patterns
 - | patterns found by clustering algorithms

Data Mining: An Architecture for the Problem



Query and Search

- The idea of the automated creation of metadata is to develop metadata that reflects the scientific content of the data sets within the database rather than just data structure information
- The locus of the metadata is the virtual data center. The end user would see only the virtual data center
- The original metadata, resident in the actual data centers, would be reproduced in the virtual center.
- The original metadata would be augmented by metadata collected by the automated creation procedures, by pointers used to link related data sets in distributed databases, and by metadata collected in the process of interacting with system users.

Data Mining: An Architecture for the Problem



■ Query and Search

- client browser
- expert system for query refinement
- search engine
- reporting mechanism

Data Mining: An Architecture for the Problem



Client Browser

- The client browser would be a piece of software running on the scientist's client machine
- The client machine is likely to be a PC or a workstation
- The idea is to have a GUI interface that would allow the user to interact with a more powerful server in the virtual data center
- The client software is essentially analogous to the myriad of browsers available for the world-wide web

Data Mining: An Architecture for the Problem



Expert System for Query Refinement

- Two basic scenarios 1) the scientist knows precisely the location and type of data he desires, 2) the scientist knows generally the type of question he liked to ask, but has little information about the nature of the databases with which he hopes to interact
- The approach is to match a vague query formulated by the scientist to one or more of the digital objects discovered in the automated-generation-of-metadata phase
- The expert system would initially be given rules devised by discipline experts for performing this match and would attempt to match the query to one or more digital objects (patterns)
- The expert system would then engage the search engine in order to synthesize the appropriate data sets

Data Mining: An Architecture for the Problem



Expert System for Query Refinement

- The expert system would also take advantage of the interaction with the scientist to form a new rule for matching the original query to the digital objects developed in the refinement process
- The scientist would be informed how his particular query was resolved; this allows him to reformulate the query efficiently
- Log files of these iterative queries would be processed automatically to inspect the query trees and possibly, improve their structure
- Other experts not necessarily associated with the data repository itself may have examined certain data sets and have commentary in either informal annotations or in the refereed scientific literature
- Provide a mechanism for indicating reliability of data

Data Mining: An Architecture for the System



Search Engine

- Large scale scientific information systems will be distributed and contain not only the basic data but both structured metadata
- Given the volume of the data, high performance engines that integrate the processing of the structured and unstructured data would be required
- Both DBMS and information retrieval systems provide some functionality to maintain data
- DBMS allow users to store unstructured data as binary large objects (BLOB)
- Information retrieval systems allow users to enter structured data in zoned fields

Data Mining: An Architecture for the Problem



Reporting Mechanism

- The basic idea is not only to retrieve data sets appropriate to the needs of the scientist, but also to scale down the potentially large databases the scientist must consider
- The scientist would consider megabytes instead of terabytes of data. The search and retrieval process may still result in a massive amount of data
- The reporting mechanism would thus initially report the nature and magnitude of the data sets to be retrieved
- If the scientist agrees that the scale is appropriate to his needs, the data will be delivered by an FTP or similar mechanism to his local client machine or to another server where he wants the synthesized data to be stored