Embracing Statistical Challenges in the Information Technology Age

Bin Yu
Department of Statistics
University of California at Berkeley, CA
binyu@stat.berkeley.edu, www.stat.berkeley.edu/users/binyu

Abstract
This paper examines the role of statistics in the current information technology developments. We start by reviewing the current state of computer technologies and the initiative of cyberinfrastructure from the computer science community to integrate computer technologies into the very fabric of science and beyond. Areas of science and text mining are reviewed to motivate a 7-component statistical investigation framework which is used to situate the current achievements of statistics (including machine learning). The challenges that we face are highlighted in order for us to join the cyberinfrastructure design and taking advantage of the trend of parallel or distributed or grid computing. They include developing Exploratory Data Analysis (EDA) for massive data sets, using non-traditional mathematical results for insights in high dimensions, and frameworks for streaming data analysis, data fusion, and taking into account computation, transmission and compression constraints.

We conclude strongly that, for the healthy existence of our field, computer technologies must be integrated into statistics, and statistical thinking must be integrated into computer technologies or into the making of the cyberinfrastructure.

1 Introduction

Information technology (IT) is a broad subject concerned with technology and other aspects of managing and processing information, especially in large organizations. In particular, IT deals with the use of electronic computers and computer software to convert, store, protect, process, transmit, and retrieve information. – Wikipedia

Information technology began no later than the invention in 1946 of the Electronic Numerical Integrator and Computer (ENIAC), the first device able to solve a large range of computing
problems. It weighed 27 tons, however, and was a very different creature from what we know today as a computing device. Since ENIAC, IT has seen an evolutionary process of computing devices from the 27 ton ENIAC to the compact desktops in offices and homes, to the book sized laptops at airports and cafes, and to the tiny hand-held devices like cell phones, MP3’s and iPods. Leap-and-bound advances have also been witnessed in computer software and network technologies, wired (metallic or optical) or wireless (radio-wave or infra-red). These advances have drastically increased our abilities to collect data, in the forms of numerical, text, image, video, audio, multimedia and beyond.

Electronic documents or records stored on computers are hallmarks of the IT revolution: government, industry, university, individual alike are using computer technology to create a gigantic amount of text in various file formats (e.g. doc, txt, pdf, ps) and in all human languages (currently in use or endangered or long-dead). Images and videos are much larger than text files. They are generated by optical sensors on satellites, by medical scanners such as PET, MRI, fMRI, by biological imaging tools to understand macro, micro and nano scale activities of cells and molecules, by digital sky surveys, by security surveillance cameras, and by personal digital cameras and videos. Audio waveform data such as those created by human speech, radio broadcasts, movie sound tracks, and concerts are stored in digital form on computers. Multimedia data consist of text, image and audio all at once, and they are often the norm for TV programs, movie DVDs and websites of major newspapers. We are in an IT data deluge!

Since ENIAC in 1946, the computing technologies have been growing exponentially, and are responsible for the data deluge. In 1964, Gordon Earle Moore, a co-founder of Intel, predicted that the number of macro-, micro-, and nano-scale components on a chip increase every year by a factor of 1.5 or 2 and this Moore’s law is valid. An exponential law is also found with semiconductor memory or storage technology, with the consequence that its price decreases by 28% on average every year. Access to the internet from homes is being made available by the backbone optical fibers. However, we are meeting physical limitations in all three areas: we are getting to the limit on how many switches (transistors) we can put on a unit area of a chip before the quantum effect occurs; the access speed to the ever-cheaper memory devices is limited by the materials of the devices; and the transmission speed of optical fibers can not exceed the speed of light (cf. [5]) and network switches induce further delays.

Parallelism is now used to mitigate these limitations (cf. [5]). Within a computer, multiple chips form parallel processing units; paralleled or clustered high-speed computers are connected to further increase the computing power; for storage, since disk prices are falling, many disks are used to host databases of a few terabytes. Wired and wireless networks are used for distributed (parallel) computation. The costs of displays are dropping and useful 3D interaction is becoming feasible on possibly many displays at the same time. Amidst the parallelism developments of computer and network hardwares, parallel computing is still useful.
However, the new computing modes are virtual computing and grid computing. Relative to virtual and grid computing, parallel computing is rigid because it specifies how iterative algorithms are to be cut up and requires a central supervisor. Grid computing attempts to break this bond and allow clusters to determine their own computation during free-cycle time. Virtual computing further abstracts this environment so that different programs and different operating systems can handle slices of data. In all these modes, the biggest difficulty is that it takes more time to cut up the data than to run the actual computations. True distributed (peer-to-peer) algorithms can process data locally and merge their results pairwise.

In 2003, the U.S. NSF blue-ribbon advisory panel on cyberinfrastructure proclaimed in their report ([5]) that “we now have the opportunity and responsibility to integrate and extend the products of the digital revolution to serve the next generation of science and engineering research and education.” In 2005, the report “Towards 2020 Science” [65] written by a dozen prominent scientists invited by Microsoft concluded that “an important development in science is occurring at the intersection of computer science and the sciences that has the potential to have a profound impact on science. It is a leap from the application of computing to support scientists to do science (i.e. computational science) to the integration of computer science concepts, tools, and theorems into the very fabric of science.” Both reports send out the same unmistakable message: the IT revolution has progressed to a tipping point that we need to build a computer cyberinfrastructure that is well integrated into the very fabric of science, engineering and education to create useful information from the deluge of data. To quote from Wikipedia,

*The term “cyberinfrastructure” ... describes the new research environments that support advanced data acquisition, data storage, data management, data integration, data mining, data visualization and other computing and information processing services over the Internet. In scientific usage, cyberinfrastructure is a technological solution to the problem of efficiently connecting data, computers, and people with the goal of enabling derivation of novel scientific theories and knowledge.*

The “Towards 2020 Science” report lists the following specific areas of science needing this new cyberinfrastructure: earth’s life-support systems, biology (cell, immune system, brain), the origin of life, global epidemics, revolutionized medicine, and future energy. It is noted that data from these fields are not only from experiments but also from simulations. Outside science, massive data sets are also collected by governments, financial institutions, health and educational organizations.

In his recent article to commemorate the 40th anniversary of Tukey’s “The future of data

---

Footnote: These descriptions of computing modes are basically taken from an email exchange from Dr. Leland Wilkinson commenting on an earlier version of this paper.
analysis” paper in 1962 [69], Mallows (2006) [51] gives the following definition of statistics, emphasizing the aim of statistics:

Statistics concerns the relation of quantitative data to a real-world problem, often in the presence of variability and uncertainty. It attempts to make precise and explicit what the data has to say about the problem of interest.

Wikipedia has the following definition, emphasizing the process of conducting statistics and its users:

Statistics is a mathematical science pertaining to the collection, analysis, interpretation, and presentation of data. It is applicable to a wide variety of academic disciplines, from the physical and social sciences to the humanities; it is also used for making informed decisions in all areas of business and government. – Wikipedia

With both definitions in mind, we ask what kind of a role statistics will play in this new computer cyberinfrastructure outlined by both the NSF and the “Towards 2020 Science” reports. Should we make modifications to our existing paradigms or must we make fundamental changes to our paradigms to meet the challenges of the data deluge? One thing is clear: whatever we develop in statistics should fit well with the other developments of computer technology in this new cyberinfrastructure. This is because for the healthy existence of our field, we have to find our way to be part of this new cyberinfrastructure making.

The rest of the paper is organized as follows. In Section 2, we review selected important subject areas where the data deluge is happening, expanding on two projects from the author’s own research to other problems in science and text mining. One project is online arctic cloud detection using multi-angle satellite images, while the other is re-ranking candidate sentences in Japanese using features from the phonetic string. A 7-component process emerges from the cloud problem to encompass the common aspects of an interdisciplinary statistical investigation. Then text mining problems of a broad range (information retrieval, web search, information extraction, question answering) are reviewed and discussed in terms of this 7-component framework. In Section 3, we examine the responses or achievements of the greater statistics community (including machine learning) to the data deluge: exploratory data analysis through R or Matlab, and statistical machine learning advances which take into account computation such as loss-function-based methods and graphical models and sparse model building for interpretation. Moreover, we point out exciting directions for future research by thinking about statistics within the cyberinfrastructure. By taking advantage of this cyberinfrastructure and situating statistics in this cyberinfrastructure, we argue that the statistics community has to tackle the following important issues in massive data modeling: visualization beyond R and Matlab (possibly through parallel or dis-
tributed or grid computing), interaction with databases (in a distributed manner), merging considerations of transmission, compression and computation seamlessly into our statistical framework (parallel or distributed algorithm development is a key), need for non-traditional mathematical tools (such as random matrices), streaming data analysis, and data fusion. Section 4 concludes the paper with a parting message.

It is acknowledged that the goal of the paper is ambitious, and it is difficult to do justice to all the exciting research conducted in and outside statistics on solving massive data problems. What appears in this paper is a personal view, admittedly limited in scope and incomplete, although efforts were made to be inclusive. It is hoped, nevertheless, that some degree of synthesis is achieved and challenges we face are distilled. At the minimum, our discussions here might stimulate more exchanges in and outside the statistics community so that statistics as a field adequately positions itself in the making of the new cyberinfrastructure.

2 Examples of data deluge

As discussed in the 2020 Science report, many fields in science are collecting enormous amounts of data and are the users of the new cyberinfrastructure of which statistics has to be part of. In this section, we select a few examples from an almost infinite pool of fields. They are in no sense exhaustive of all IT data problems. Some of them are selected and covered in more detail because of our familiarity with them, others are briefly described because more details can be found in papers of this special issue, and still others are mentioned because they are at the frontier of science and technology. Their origins differ, but they share with statistics a dependency on the new cyberinfrastructure. Their high-dimensionality and high data rate pose challenges in data management, exploratory data analysis, formal modeling, and model validation. Meeting these challenges requires the integration of statistics (thinking and methodology) into the cyberinfrastructure at the design stage, rather than as an add-on later.

2.1 Tales from science

Atmospheric science is accumulating huge amounts of data through model simulations and observations from remote sensors on satellites and airborne instruments. There is no doubt that atmospheric science is very important because answers to questions like global warming will rest on this science. It presents great opportunities and challenges for statistics (cf. Second NASA Data Mining Workshop Final Report, [3]). One simulation model in atmospheric science, is Mesoscale Model Version 5, a joint effort of the National Center for Atmospheric Research (NCAR) and Penn State University (cf. [8]); and this model is being replaced
by the Weather Research and Forecast Model ([1]), a joint effort of five agencies including NCAR. Both models utilize atmospheric observations as initial values, and solve partial differential equations regarding physical thermodynamic and micro-physical processes on a 3-dim grid. Besides atmospheric science, complex computer model simulation is used also in wide-ranging areas from meteorology, wildfire control, transportation planning, to immune system function as evident in the workshop on this topic (cf. [8]). Policy and decision making use simulation models, but many difficult issues have to be confronted in this simulation world. For example, how to evaluate or validate a computer model such as the Mesoscale Model Version 5, which takes much computing power for just one run. For collective thoughts on steps necessary to integrate statistics into this model simulation world, we refer interested readers to Berk et al (2003) [8].

Much of the remotely-sensed observation data in atmospheric science are publicly available, and thus provide a great resource for statistical investigation. Over the past four years, we have been working on the arctic cloud detection problem using Multi-angle Imaging SpectroRadiometer (MISR) satellite data. Our team is interdisciplinary and multi-institutional with a (former) graduate student Dr. Tao Shi (now an Assistant Professor at Ohio State Univ), a statistician at the Jet Propulsion Laboratory, Dr. Amy Braverman, and a meteorologist, Professor Eugene Clothiaux, at Penn State University.

Arctic cloud detection

In Shi et al (2006a) [63] we motivate the problem by stating that “global climate models predict that the strongest dependences of surface temperatures on increasing atmospheric carbon dioxide levels will occur in the Arctic and this regional temperature increment can lead to global temperature increase. A systematic study of this relationship requires accurate global scale measurements, especially the cloud coverage, in the arctic regions. Ascertaining the properties of clouds in the Arctic is a challenging problem because liquid and ice water cloud particles often have similar properties to the snow and ice particles that compose snow- and ice-covered surfaces. As a result, the amount of visible and infrared electromagnetic radiation emanating from clouds and snow- and ice-covered surfaces is often similar, which leads to problems in the detection of clouds over these surface types. Without accurate characterization of clouds over the Arctic we will not be able to assess their impact on the flow of solar and terrestrial electromagnetic radiation through the Arctic atmosphere and we will not be able to ascertain whether they are changing in ways that enhance or ameliorate future warming in the Arctic.”

Multi-angle Imaging SpectroRadiometer (MISR) is a sensor aboard NASA’s EOS satellite Terra launched in 1999. It makes novel electromagnetic radiation measurements at 9 different viewing angles at wavelengths (red, green, blue, and near-infrared) of which all are collected in the 275 m × 275 m resolution initially (leading to about 1 million pixels per image). Due to the high data rate, all other bands except for the red are aggregated to the coarser 1.1
km $\times$ 1.1 km resolution before transmission to the base station on earth.

It was known to the MISR team that the MISR operational arctic detection algorithms had not worked very well in the Arctic (or the polar regions). We were invited to work on this problem by Braverman and later had the fortune to have an open-minded scientist, Professor Eugene Clothiaux, join our team as the subject expert. Our goal in this project is to provide a better cloud labeling for each pixel based on MISR’s red band 9-viewing angle data (other bands have a coarser resolution and do not seem to offer more information for the cloud label). For MISR operational purposes, we would like to have an on-line algorithm which outputs a label while data come in from the MISR sensor. The first step of obtaining data from NASA data center turned out to be lengthy (over 3 months) and the data format initially could not be read directly into Matlab – a special program had to be borrowed from the MISR team at the Jet Propulsion Laboratory to convert the data into a suitable form for Matlab. Because the data volume was too large for Matlab to carry out computations in a timely manner, we programmed a Graphical User Interface just to obtain some very simple summary statistics. Due to the advancing computer technology, Stereo-visualization of MISR images is now available through the Leica Photogrammetry Suite (LPS) by Leica Geosystems (www.leica-geosystems.com), with a customized interface for MISR data. This provides the necessary means to obtain validation data from experts to estimate the clouds heights, the next goal of our project.

The MISR red-band data is 9 dimensional per pixel corresponding to 9 angles and there are about 3 million pixels per image block (which consist of three original images). The MISR operational algorithm is called Stereo Derived Cloud Mask (SDCM), which uses the red-band data to first retrieve cloud height based on matching of clouds in images of different angles. The cloud mask, or estimated cloud or not label, is obtained by thresholding the cloud height based on the terrain height. The matching step is computationally expensive and error-prone in the polar regions.

Our first breakthrough came six months after we embarked on the project: we realized that we could look for “snow/ice” pixels, bypassing the error-prone cloud height retrieval underlying the MISR operational algorithm. After three years of interactions with the MISR team by giving talks at their science meetings, email exchanges, and visits, we were successful in devising and testing a simple cloud detection algorithm called Enhanced linear matching clustering (ELCMC) (Shi et al, 2006 [62]). The cornerstone of our algorithm is three physically meaningful features from the MISR red-band measurements. We then fused the MISR-ELCMC labels with MODIS cloud labels (MODIS is another sensor on Terra which is hyperspectral but one angle) to get the training data to apply Quadratic Discriminant Analysis (QDA) for a probability label of a pixel ([61]). When compared with the best “ground-truth” data, expert labels, our algorithm gives an average of 94% accuracy on labelled pixels of 60 blocks of data of millions of pixels (while MISR-SDCM gave only 80%). It is worth noting that two expert labelings typically differ by about 5%, so our method is
basically reproducing expert labels automatically.

This experience reveals several essential steps or components of interdisciplinary research to extract useful information from the data deluge:

1. Access to good scientific or subject problems and expertise;
2. Collection and management of large data sets (including effective transmission and storage and possibly data reduction or feature selection);
3. EDA (visualization and descriptive statistics and possibly also data reduction or feature selection);
4. Processing mode: off-line or on-line (streaming data);
5. Formal modeling with computation and accuracy considerations (estimation and uncertainty assessment);
6. Data fusion from various sources;
7. Validation using information from outside statistics (quantitative test data or qualitative validation based on subject matter).

The first step could be the most challenging for mathematically trained people. Good problems rarely fall from the sky. It takes open-mindedness, readiness, inter-personal skills, and good luck to find them, and requires teamwork to solve them successfully. Yet not all collaborations, as not all relationships, end well. We were fortunate to have assembled an excellent multi-disciplinary team for the cloud problem. Another example of successful multi-disciplinary collaboration was the probabilistic weather forecasting project at the University of Washington at Seattle. As required by the grant, psychologists were involved, which turned out to be a great asset. Cognitive science was used by the psychologists to decide on how to best display uncertainty information. The team was a mix of statisticians Adrian Raftery and Tilmann Gneiting, atmospheric scientist Clifford F. Mass, and psychologists Susan Joslyn and Earl Hunt ([?]). (See Speed (2005) [64] for more discussions on interdisciplinary research.)

Often, as in the arctic cloud detection problem, statisticians are not involved in data collection (the second step). It would be ideal, however, to get involved in data collection design for the next generation multi-angle sensor that is currently under consideration by NASA and similarly for other problems. The reason that data reduction/feature selection is included in both steps 2 and 3 is that it could occur in either stage or both depending on the problem. For the cloud problem, data reduction was done when the 275m resolution data was aggregated into the 1.1 km resolution data due to the transmission/storage constraint. Further data reduction/feature selection was conducted when we derived the 3 physically
meaningful features in the 1.1 km resolution from a patch of 8 \times 8 \, 275 \, \text{m resolution pixels}
and using 4 of the 9 viewing angles. These three features incorporated previous work and
subject knowledge from atmospheric science and were the key to our success.

As alluded to earlier, the cloud problem demands the on-line mode for processing. This
streaming data mode determines our choice of QDA instead of the front-runner machine
learning method, Support Vector Machines, due to QDA’s low computation cost in the
feature space.

In terms of the formal modeling stage, for the arctic cloud detection problem we used
a mixture of ideas from clustering (thresholding and 1-dim EM) and classification (QDA).
Clustering, classification and regression are common problems nowadays and the field of
machine learning (which we regard as new developments of statistics in the IT age) is re-
sponding to the needs of these problems and making impressive advances. We will cover
machine learning in more details in Sec. 3.2.1.

The same set of components of interdisciplinary research can be found in other science
and engineering areas such as the two examples below.

**Digital sky surveys**

Enormous amounts of data are flooding astronomers from the next generation of sky
surveys such as the 2 Micro All Sky survey and the Sloan Digital Sky Survey (cf. Welling and
Derthick, 2001 [74], Jacob and Husman, 2001 [39]). From the SDSS website (www.sdss.org),

“Simply put, the Sloan Digital Sky Survey (SDSS) is the most ambitious astronomical
survey ever undertaken. When completed, it will provide detailed optical images covering
more than a quarter of the sky, and a 3-dimensional map of about a million galaxies and
quasars. As the survey progresses, the data are released to the scientific community and the
general public in annual increments.”

In a five year period, the 2 Micron All Sky Survey and Sloan Digital Sky Survey pro-
duced 550 gigabytes of reduced data and 1 terabyte of cutout images around each detected
object. These volumes surpass humans’ ability to study them even if we want to. The only
alternative is to rely on computing power to sift through them. This also leads to on-line or
streaming data analysis that require real-time processing or analysis of data to extract use-
ful information, with the possibility of discarding data on the fly due to storage limitations.
This processing could have a visualization component and human interaction, but it has to
be automated to consume the huge amount of data. Clustering, classification, and multiple
testing are useful inference frameworks to answer questions from these sky surveys.

**Particle matters**

In particle physics, gigantic experiments are undertaken to understand the most elemen-
tary ingredients of matter and their interactions. One of the experiments, the Compact
Muon Solenoid at CERN in Geneva, generates about 40 terabytes per second, which has to be reduced to about 10 terabytes per day in real time for later analysis. This is another example of streaming data analysis. For more details, See Knuteson and Padley (2003) [46].

2.2 Text data

Texts are the data for information retrieval (e.g. web search), information extraction (e.g. title extraction from documents), natural language processing (e.g. machine translation), and question answering (e.g. “what is the distance between Berkeley and San Francisco?”). To help readers to follow up on these topics beyond this paper, we list below references and pointers to conferences. For information retrieval, information extraction, and natural language processing, read Manning and Schütze (1999) [53], Jurafsky and Martin (2000) [43], Manning et al (2007) [52]; for information retrieval and question answering, read TREC Publications at http://trec.nist.gov/pubs.html. For current developments in these areas, go to websites of the following conferences: StatNLP: Association for Computational Linguistics (ACL), North American ACL (NAACL), Empirical methods for NLP (EMNLP), European ACL (EACL), ICASSP, ICSLP, SIGIR, and WWW, and http://trec.nist.gov/pubs.html.

The research on text data is happening mostly outside the traditional statistics community in areas such as signal processing, machine learning, and artificial intelligence. However, statisticians are getting involved as illustrated by Genkin et al (2006) [31] in this special issue. Below is another example of employing a statistical method to a language modeling problem in which we have taken part.

Input method editor

In order to use a standard keyboard to enter complex Asian scripts (e.g. characters in Chinese or Japanese), we can enter transliterated or phonetic input symbols (e.g. Ramonji for Japanese or pinyin for Chinese). An Input Method Editor (IME) converts these transliterated symbols into texts in the original language. The performance of an IME is measured by the character error rate, that is, the number of characters wrongly converted from the phonetic string divided by the number of characters in the correct transcript.

In Gao et al (2006) [30], a training data set consisted of a set of input/output pairs with the input $A_i$ as an input phonetic string and the output the reference transcript word string $W_i$ in Japanese. First a word trigram model (Gao et al, 2002 [29]) is used to generate a set $GEN(A_i)$ of candidate word strings given the input $A_i$. A trigram model was built by maximum likelihood estimation applied to the Nikkei Newspaper corpus in Japanese (of 36 million data units). Our task was to develop a rule to choose one word string from $GEN(A_i)$ as the text input for $A_i$ for future $A_i$, and the rule was to be derived using the training data set. Computing a linear score is currently a popular approach and the score is based on the feature vector of each candidate word string. The weights in the score are determined or
estimated using the training data. The covariate vector or feature vector $f_d(W)$ is a real-vector valued function of $W$. In Gao et al (2006) [30], the feature vector composed the log probability that the word trigram model assigned to $W$, and the counts of the word 1-gram and word 2-grams in $W$. The dimension of this vector in the Gao et al paper is around 860,000.

Following up on the success of boosting in ranking problems (Koo and Collins, 2005 [20]), we used BLasso, a newly developed machine learning method related to boosting and an approximate algorithm for solving Lasso (Zhao and Yu, 2004 [78]). The same exponential loss as in Collins and Koo (2005) [20] was employed, but an $L_1$ penalty was added to find the weight vector. BLasso produced significantly better character error rate results than boosting on data sets of Japanese language input which are cross-domain. That is, the training data is mostly from a different domain (i.e. Nikkei Newspaper corpus of 36 millions) than the testing or adaptation domain (e.g. Yomiuri Newspaper corpus) for which only a small amount of training data is available. See Gao et al (2006) [30] for more details.

IME is a relatively easy language modeling problem and small cross-domain character error rates have been achieved (from 3% to 12%). Other tasks could be more general and harder as to be seen next.

**Information retrieval**

Information Retrieval (IR) is the science and practice of indexing and searching data, especially in text or other unstructured forms. A typical IR task could be searching for an image with horses in an image database, or searching for a document with a specific name on a computer. The volume of data is daunting and the data structure is not traditional for statisticians.

**Web search**

Web search is something we all now rely on for seeking information on almost anything and everything. Searches for papers on a topic, papers with specific titles, show times and locations of a particular movie, mortgage rates, email addresses of colleagues, their telephone numbers, are just few examples of web searches. Web search is the hottest topic in IR, but its scale is gigantic and desires a huge amount of computation. First, the target of web search is moving: the content of many websites changes within a week for 30% to 40% of the web (Fetterly et al, 2004 [26]). A crawler is the tool that a search engine uses to collect websites into its database to answer queries. Because the web content is interlinked, very clumpy, and very diverse, one can not easily carry out a random sampling to crawl. So the crawling results, or search results for given queries, might be biased. Worse yet, the content of a website can include more than just text, for instance, images and videos, as well as interactive content. Web data is therefore very unstructured and its processing or data reduction/feature extraction very challenging. We refer the readers to Henzinger (2000)
and Henzinger et al (2002) for more details on data collection and algorithm issues related to web search.

Based on the websites collected in a database by a crawler, when a query is entered, the relevant websites will be found and ranked. This fuels a very active research area in machine learning: ranking function estimation, of which the IME problem is also an example. A ranking function in web search usually depends on weighting the content of websites and links among the sites, as in the PageRank algorithm used by Google (Brin and Page, 1998 [12]). When a weighting scheme is open to the public, however, opportunities arise for the so-called SEO’s (search engine optimizers) to mislead the search engine to irrelevant websites of an SEO’s customers. Search engines therefore have to outwit SEOs in their search and ranking strategies in addition to dealing with the fast-changing and growing world of websites.

**Information extraction**

Information extraction (IE) attempts to do more than IR. Its goal is to extract useful facts for users in electronic documents (in most natural languages), that is, it aims to use text as a demonstration of understanding. IE had already existed early in natural language processing, but its potential is boosted dramatically by the IT revolution. For example, Lloyds of London, a shipping company, performed an IE task with human analysts for hundreds of years. The company wants to know all the ship sinking incidents around the world and put the information in a database. The IE question in the IT age is whether we could replace human analysts by computer algorithms automated to collect this information from data such as newspapers, web broadcasts, and government documents, possibly in different languages.

**Question answering**

Question answering (QA) takes open domain questions and searches over a collection of documents to find concise answers to the questions. It takes IR and IE further to deal with natural language sentence queries and return answers that need to be precise. Current QA systems could answer simple questions about facts like the one about the distance between San Francisco and Berkeley, but have difficulty answering complex questions requiring reasoning or analysis such as ”What are the differences between a private university and a public university?”.

### 2.3 Text data investigation from a statistical view

In all three areas of information retrieval, information extraction, and question answering, texts need to be represented by numeric forms before further actions. Programming skills are required to process these text data, and statistical thinking in natural language processing
is necessary to keep the key information in the numeric form (or forming the feature vector) for downstream comparisons between data units. In addition, statistical modeling is often a must to relate the feature vector to the goal of the task as illustrated in the IME example. In this section, we situate the text data problems within the 7-component framework to highlight their common challenges.

1. Access to good scientific or subject problems and expertise

Information gathering is a basic intelligence trait of all humans. Using computer technologies to aid this process undoubtedly takes us way beyond what would have been possible with our innate abilities, as it makes accessible data that humans would be incapable of manually processing. IE, IR (websearch) and QA are extremely important and exciting since they help us obtain information. These areas all involve a certain level of natural language processing (NLP), and there is a general question of how to incorporate linguistic rules in statistical or other quantitative approaches. The move in natural language processing seems now towards using the vast text data that a machine can handle to derive structures and various levels of linguistic generalizations. That is, natural language processing has generally become very empirical or data driven. In the empirical approach, large corpora are needed for implementation and evaluation. Often, creating an evaluation data set involves humans. A scoring system or loss function also has to be devised to reflect the human evaluation as in the IME example, leading to statistical machine learning research.

2. Collection and management of large data sets (including transmission, storage and data reduction)

Data collection is present in information retrieval (websearch), information extraction and question answering for they all rely on various databases. The most interesting is web search in which web crawling is data collection. Web crawling is a difficult task, but an exciting area for statisticians to investigate. Because of the dynamic nature of the web, new research might be called for in experimental design on how and when to select websites to crawl and what content and how much to take to store. For the IME project, the data were already collected when we got involved.

Text data sets are often huge (e.g. the sample size is 36 million and the dimension of features 860,000 for the IME data). It is non-trivial to transmit and store such data. Often sending medium such as DVD’s via postal mail is the best route. The data obtained by us for the IME project were already in the form of feature vectors so some data reduction was already done by researchers who came before us. And it is common that for IE, IR, QA and web-search, there are standard ways to extract numerical features from the texts.

3. EDA (visualization and descriptive statistics and possibly data reduction or feature selection)
For the IME project, we have not carried out any EDA because the sample size is 36 million and random sampling is not a good idea because of the sparsity of the feature vector (most of the features are zeros). It would be original and new research to figure out how to visualize these feature vectors in a meaningful way and conduct EDA in general for text data.

4. **Processing mode: off-line or on-line (streaming data)**

Most works on text data are off-line. Although the results are impressive, they are still not comparable to results by humans, even if the task is the relatively simple IME. When enough knowledge is accumulated and accuracy is much improved, the demand for on-line training of rules for IR, IE, etc, would drive the research into fast and accurate prediction rule estimation. For example, we can envision the need for an on-line IME algorithm so that the estimation of the parameters can adapt on the fly to the individual users of a computer, as the vocabulary used varies by user and document topic.

5. **Formal modeling with computation and accuracy considerations**

In the IME project, a classification algorithm BLasso was called upon directly and it produced improved results over boosting for the Japanese input data sets. BLasso is a machine learning algorithm and machine learning methods are very useful in text mining. The paper by Genkin et al (2006) [31] also uses a method similar to Lasso, but the optimization method is different from BLasso. We will address machine learning in detail in Section 3.

6. **Data fusion from various sources**

Data fusion is an emerging active topic in text mining, because it is natural to think about fusing information across languages, as in the Lloyds of London example above. For web-search, it is certainly the case that different modes of data (e.g. text, image, audio) at a webpage should be integrated to form the feature vector for the webpage. Search engines like Google do use these multi-mode data (e.g. local.google.com). The most obvious fusion should be based on numerical features from each kind. For information retrieval of images, the title of an image should be combined with the image content. For a movie data base, one needs to fuse information from videos, sound track and subtitles. It is a challenge for statistics to provide a useful conceptual framework for this data fusion related to text.

7. **Validation using information from outside statistics**

Training data are often readily available for text mining; or least they can be produced with some cost by having humans label the data. Machine learning methods such as BLasso are thus validated based on improved prediction performance on test data. Machine learning derived rules have now been used to provide insights into linguistics as in the works of Dan Jurafsky and co-authors. As we will see in the machine learning section, there is a general trend to build and study sparse models as a step towards interpretability.
3 Accomplishments and challenges

The need for statistics or analysis of data is rooted in questions outside statistics. The data form or structure varies from problem to problem, but statistically they share two common characteristics: high dimensionality and volume and high data rate. Moreover, from a statistical point of view, the above examples are aimed at building a model either for prediction through clustering, classification or regression (e.g. atmospheric science and astronomy) or for interpretation (most desirable in sciences and possibly in social sciences as well).

Now let us time travel back to 1922, the year that Fisher published his foundational work Fisher (1922) [27] giving statistics a sound framework to truly become a science. First as a scientist himself, Fisher formulated his statistical inference framework to solve scientific problems in areas such as agriculture and genetics.

Fisher clearly states three “types” of statistical investigation:

1. Problems of Specification
2. Problems of Estimation
3. Problems of Distributions

Fisher cast these three types of investigation into a mathematical framework based on the assumption that data are random samples from an underlying population. Consequently he relied on the available mathematical tools, probability theory and calculus, to define the concepts of consistency and efficiency and prove results about them.

Eighty four years later, we are seeing an urgent need to investigate into an ever-exploding amount of data. As clear from the NSF and the “Towards 2020 Science” reports, we are also at the verge of integrating concepts of computer science into the very fabric of science. The implication for statistics is that we must integrate concepts of computer science into statistics and therefore bring our community into the making and design of the new infrastructure. Only by doing so will we able to keep or develop a competitive edge in accessing important science and technology problems that are the reason for the existence of our field.

Comparing Fisher’s list with ours developed above, our list is obviously quite an expansion and an evolution to adapt to new problems. This is not surprising, of course. He made a rigorous and useful framework to rest statistics on in order to tackle problems of his age, and it is now our turn to tackle ours. In the sections to follow, we first give our view on where we are in selected areas of statistics and then move on to describe challenges that we face.

3.1 Selected landscapes in statistics

In this section we examine selected topics in statistics from the point of view of the 7-component framework introduced in Section 2.
We have been utilizing Fisher’s framework for a long time. However, things have been moving or changing in statistics. Sometimes the moves came from within (e.g. Tukey developed EDA in the seventies and R is becoming a freeware of choice for researchers for interactive data analysis, both at Bell Labs), but other times the moves came from outside (e.g. machine learning, streaming data analysis, CS or engineering people solving IT problems). From inside or outside, the moves are made by people who were solving real data problems just as Fisher was solving real data problems. Although all of us recognize the importance of computation, it seems that for most of us that R and Matlab are the extent of our computational activities. Data visualization, a main ingredient in EDA, is carried out also in R and Matlab through histograms, scatter plots, and their extensions to dimensions larger than 2. We have been users, not part, of computer technologies. Hence more moves or changes are necessary for statistics to become part of the new cyberinfrastructure by integrating considerations of computation, transmission, and storage into the design and analysis of our methodologies, and by bringing statistical thinking into the making of the cyberinfrastructure.

3.1.1 Data management

At the present time, if a statistician decides to help answer a question outside statistics, most likely this person is not involved in data collection and this person would move the data to his/her location. This move is formally data transmission. The satellite images at NASA websites are often too huge to download, as we experienced in our arctic cloud detection project (Shi et al, 2006a, 2006b [63, 61]). (A couple months in 2001 was spent on waiting for the MISR data to arrive in CDs from the east coast.) Alternatively, a first preliminary analysis is possible on compressed data via website downloading. Because of the spatial dependence of the pixels on an image, it is not desirable to randomly select pixels to download for analysis. It is much better to compress the image to download. JPEG or other general image compression tools are designed for visual inspection, however. They remove irrelevant image features that our vision system does not need. They might not be suitable for image compression for statistical analysis. This brings up the need to develop image or general data compression tools aimed to keep statistical information or remove statistically irrelevant features in the images (cf. Jörnsten et al (2003) [42] and Braverman et al (2003) [10]). The next step is to analyze the data now on the statistician’s own computer. R or Matlab is the popular interface between research statisticians and data. For image data from the arctic cloud project, we found Matlab much easier to use, although we also use R as well depending on the skills and familiarities of the researcher with either software. More and more frequently, the speed of R or Matlab and storage are not sufficient for large data set visualization or summary statistics calculation in a reasonable amount of time as in our cloud project, and C or C++ programs have to be written to interact with R or Matlab.
for numerical calculations. For visualization, we are often forced to downsize the data set to accommodate the limitations of R or Matlab. Hence the analyst relying on R or Matlab usually cannot see all the data and often opts to develop a formal statistical model or a procedure which is more likely to be tested on all the available data if C++ programming is used after the initial exploration stage of the data (both visually and numerically). The results of such formal modeling again is difficult to visualize for the entire data set on R or Matlab in a research environment, making it difficult to revise or improve the fitted model.

3.1.2 Statistical machine learning

Machine learning is at the frontier of Statistics because its serious utilization of computation in statistical inference. It is called algorithmic modeling in the most interesting and thought-provoking paper Breiman (2001) [11]. Breiman argues that we have to aim at solving real data problems and we have to consider more diverse tools often driven by computation than those dependent on data models. The same view on solving real problems has been strongly advocated by Tukey ([69] and more recently by Mallows (2006) [51].

Computation has entered statistics much earlier than data compression and transmission. Retrospectively, we might view the development of computation in statistics in three phases. The first phase was pre-computer where we depended on closed-form solutions. The second phase used computers, but not in an integrated manner; we would design a statistical method and then worry about how to compute it later. Frequently calling a numerical optimization routine was the solution and we relied on the routine to determine how numerical convergence would be achieved, that is, convergence parameters were tuned for numerical reasons and the optimization routine was used as a black-box by statisticians. The third phase is the IT phase where data volume is so gigantic that procedures designed without computational considerations might not be implementable. This is also the cyberinfrastructure phase. Machine learning methods and Markov Chain Monte Carlo algorithms are examples of approaches that intrinsically integrate computation.

The computationally-feasible loss function approach

Two machine learning methodologies stand out: one is boosting (Freund and Schapire, 1997 [28], Hastie et al, 2001 [36]) and the other support vector machines (cf. Schölkopf and Smola, 2002, [59]). They enjoy impressive empirical performances and now have much theoretical understanding within both the machine learning and statistics communities. The current view of boosting is that it fits an additive model via gradient descent (or its variant) to minimize an objective or loss function. It is stopped early by monitoring the generalization or prediction error of the fitted model either estimated by cross-validation or assessed over a proper test set. That is, the minimization of the loss function is a “pretense” – we are really interested in the solutions along the way to the minimum, not the minimum itself, and
prepared to stop early. This way the numerical convergence is not important at all, but the prediction performance is. For support vector machines, computation is also the main focus via the “kernel trick” via the route of a penalized optimization of a hinge loss function. An implicit Reproducing Hilbert Space is induced by a kernel function and in this space, a linear model is fitted. However, all the computation is conveniently done via the kernel function.

The machine learning approach based on minimizing a loss function can be viewed as a natural extension of the maximum likelihood approach where the loss function is the negated log likelihood function. The penalized version can be viewed as an extension of the Maximum A Posteriori (MAP) approach in Bayesian inference. What is new is the liberation from the negated log likelihood function to a general loss function, in a way reminiscent of M-estimation. The motivation is not the same. In M-estimation, the goal is to obtain robust estimators in a parametric setting. In loss function machine learning, the goal is to have computationally feasible loss functions (often convex) to optimize over large data sets. Since robustness can be desirable also in the machine learning context, we now see some of the convex Huber functions being integrated into the machine learning literature.

This loss function machine learning approach has been very successful in building up models for prediction. The measure of uncertainty has been based on perturbing the data one way or the other (permutation, resampling and cross validation). But a fundamental assumption to justify these perturbations to the original data is the iid assumption beneath most of the current machine learning methods.

**Graphical models**

Parallel to loss function machine learning methods, graphical models are the other important development in statistics and machine learning. Graphical models effectively deal with the intricate dependences and structures in the thousands or more variables which consist of our data today. For example: spatial-temporal modeling of temperature and precipitation in atmospheric science, image processing in the multi-resolution framework of wavelets, gene network discovery based on gene expression and other modes of data, Hidden Markov Models in speech recognition, hierarchical models in information retrieval, and error-correction codes in communication. Obviously models for dependent structures existed long before the formalism of graphical models that uses the graph representation with variables as nodes so that general algorithms can be devised to compute marginal and conditional probabilities of interest. Graphical models are widely accepted and used in the engineering and science communities. See Jordan (2004) [41] for an insightful review of graphical models from the algorithmic point of view and references of graphical models in applications.

One popular inference tool for graphical models is sampling algorithms, of which Markov Chain Monte Carlo (MCMC) is the most prominent. The MCMC method gives, if the Markov chain converges, a posterior distribution estimate which provides an uncertainty measure. The design of an MCMC scheme to assure a good mixing speed or convergence
has to be taken into account when one lays out the distributions/conditional distributions for such a model. This is another example of a third generation computation. For many graphical models of high dimension, which we encounter more and more, Markov Chain Monte Carlo methods are more easily trapped in local modes of a posterior distribution and convergence is even more difficult to guarantee. Adaptive sampling algorithms are more effective in these situations. See Liu (2001) [49] for MCMC and other sampling methods.

On the other hand, the maximum likelihood principle turns a graphical model inference problem into an optimization problem. When the graphical model corresponds to a tree graph, we could use efficient message passing algorithms (or junction trees) for exact ML parameter estimation. This algorithm is very efficient when only local dependence exists in the graphical model and is very expensive otherwise (when a general graphical model gets embedded in a tree structure on an enlarged space). In general, the optimization function from the log likelihood is often not convex so that direct optimization is difficult. Searching for approximate loss functions that are more feasible computationally has been a thrust of current graphical model research (cf. Wainwright and Jordan, 2005 [73]) such that the message-passing algorithms can be effectively used or extended. In particular, Wainwright (2006a) [71] shows that for a specific graphical (mixture) model, computationally efficient algorithms could also bring estimation advantages as we have seen in the case of boosting and other methods discussed earlier.

The question of uncertainty measure via data perturbation is much harder than the iid case, however. Parametric bootstrap seems a reasonable solution, but theoretical studies are necessary to validate such an approach, especially because optimizing an approximate loss function may lead to inconsistent estimates (cf. Wainwright, 2006a [71]).

It is noted that most of the classical asymptotic distribution results have not found their way to the above third-generation methods because we need mathematical results in high dimensions which have not been used in traditional theoretical statistics.

**Sparsity and interpretability**

Interpretability of a statistical model is always desirable in any investigation and it is indispensable for model building in science. One computationally efficient means to obtain sparsity or interpretable models is through Lasso or the $L_1$ penalized Least Squares (Chen and Donoho, 1994 [19], Tibshirani, 1996 [67]). Fast algorithms to produce the whole Lasso path are known (Osborne et al, 2000, [57], 2000, Efron et al, 2004 [24]). Moreover, connections between Lasso and $L^2$Boosting are observed and understood (Efron et al, 2004 [24], Zhao and Yu, 2004 [78]) and provide understanding into the sparsity property of the boosting estimates. Sparsity is also a well-known principle in low-level vision as can be seen in Wu et al (2006) [75] of this issue. To capture sparsity of variables in a non-parametric setting, Rodeo by Lafferty and Wasserman (2005) [47] combines boosting (or gradient descent) with kernel estimation to build sparse non-parametric models.
Because of its usefulness in practice, Lasso has also been the focus of much theoretical research lately from statistics (and machine learning) and applied mathematics (cf. Chen and Donoho, 1994 [19], Donoho, 2004 [22], Tropp, 2004 [68], Tao and Candes, 2005 [14], Meinshausen and Bühlmann, 2006 [54], Zhao and Yu, 2006 [79], Wainwright 2006b [72], Zou, 2006 [81], Greenshtein and Ritov, 2004[33], vander Geer, 2006 [70]). Lasso is attractive because the $L_1$ penalty has a dual role: it simultaneously regularizes prediction and selects variables. What emerges from these studies is an incoherence or irrepresentable condition required for Lasso to select the correct variables if they exist and are sparse. This condition asks for the “irrelevant” variables to not be too correlated with the relevant or correct variables in the sparse model. These results hold also for the case of $p >> n$ which has emerged as a valuable asymptotic setup for deriving analytical results relevant to high dimensional data. On-going research with Meinshausen indicates that when the irrepresentable condition is violated, Lasso still behaves sensibly in the sense that the Lasso estimates keep the order of the original coefficients with high probability and the number of nonzero Lasso estimates can not be too much larger than the non-zero “true” coefficients. This results hold in the $p >> n$ case and the sparsity assumption for the true model can be $l_q$ for some $q \in [0, 1]$, suggesting the robustness of sparse model estimation via Lasso relative to the departure from the $l_0$ assumption which has been imposed conventionally in previous papers.

### 3.2 Challenges

It is certain that R and Matlab will be able to handle larger and larger data sets, but it is just as certain that there are always much larger data sources that R and Matlab would not be able to tackle. This right now includes databases where the sheer data volume is gigantic (e.g. digital sky surveys) and the data structure is sparse (e.g. IME) so that simple random sampling is not viable, and it also includes streaming data where data volume per unit time is too large to be even read into R or Matlab on the fly, not to mention conducting computation on the data. It is these areas where the biggest challenges lie and it is these areas where many ideas converge from statistics (including machine learning), optimization, and databases. This integration would put statistics firmly in the cyberinfrastructure, but it happens only if we decide to make it happen.

At the education level, the integration means experts from different academic disciplines (departments) has to interact and find themselves in one seminar or course for educational purposes. However, most if not all textbooks written by statisticians begin with EDA or the third component in our framework, a few cover data collection briefly and almost none discusses data management from the point of view of data transmission, compression and databases. This situation reflects the influence of Fisher’s framework of which computation is not a part. For statistics to get on the departing train of cyberinfrastructure, we need to re-think and re-design our undergraduate and graduate programs which is an important
At the research level, the best motivation and application of this integration is big science projects where huge amounts of data are collected or managed together, as is now happening in atmospheric science (e.g. model simulation and remote sensing data), astronomy (e.g. digital sky surveys) and biology (e.g. genome or brain databases across different species). The question is how we statisticians can participate in data collection and data modeling. Individual statisticians can find and are finding collaborative roles in these big projects, but it is very difficult for individual statisticians to influence the fundamentals of these projects, for example, to have a say in data collection and in what algorithms are used in mining the huge databases. It will take some collective thinking and leadership of our community to participate in these grand endeavors of science of our time. This is an interesting and important topic of its own and deserves more discussions, not here, but in other venues.

We envision that our integrated knowledge base would need components of

- data management (EDA or visualization),
- relevant mathematical analysis (e.g. random matrix theory),
- algorithm developments with data perturbation in mind (what are truly necessary computations to carry out in an optimization?), and
- the interaction of databases and methods (knowing database constraints while designing methods and knowing needed analysis while designing databases).

### 3.2.1 Turn computer technologies to our advantage

Use parallelism or distributed (grid) computing

R and Matlab, while useful for statistical research, are not adequate for exploring massive data sets on a single workstation. Now Matlab is equipped with a distributed computing engine and toolbox, which “enable(s) you to develop distributed and parallel MATLAB applications and execute them on a cluster of computers without leaving your technical computing development environment” (from http://www.mathworks.com/products/distribtb/). Meanwhile, Apple has developed a grid computing toolkit installed on all iMacs, called XGrid, which “turns a group of Macs into a supercomputer, so they can work on problems greater than each individually could solve” (http://www.apple.com/macosx/features/xgrid/). And Sun also provides its grid solutions (http://www.sun.com/software/grid/). On a much larger scale, BOINC (Berkeley Open Infrastructure for Network Computing) provides “open-source
software for volunteer computing and desktop grid computing” to solve problems in earth sciences, biology and medicine, mathematics, and physics (http://boinc.berkeley.edu/). However, as pointed it out to me by Dr. Wilkinson in an email exchange, “carving up the data can take longer than actually processing it on these (grid) systems. That’s why Google uses a distributed model, where the data never get consolidated in one place. That way, computations are local and the merging is done in a distributed fashion. Once data are merged into a single database, parallel architecture isn’t going to help much with speeding things up, because most of the time is spent accessing the data.”

These distributed computing developments are examples of the parallelism used by the computer science community to mitigate the limitations of the current computer technology and offer the users easy to use plug-in’s to increase users’ computing power.

*How are we statisticians taking this parallelism computing trend into account to boost our abilities to extract useful information from data?*

Using existing parallel environments in our statistical investigation into data, such as the Matlab toolbox and Xgrid, is a must first step. More fundamentally, however, we need to design statistical methods, exploratory or formal, which are well suited for parallel computing environments.

**EDA for massive data**

Since the advocacy and teaching of Tukey on Exploratory Data Analysis (EDA) in the 70’s, EDA has been part of statistician’s routine analysis of data. The most used tools of EDA are summary statistics and 2-dim or 3-dim visual displays such as histogram, box plots, scatter plots and time series plots. For high-dimensional data, 2-dim or 3-dim visualization abilities become quite limited for understanding the complex structures in them, even though efforts within the statistics community have been made to accommodate the high dimension through parallel plots and selective projections of data as in GGobi (cran.r-project.org/doc/packages/ggobi.pdf) and crystal-vision (http://www.crystalvision.tv/) and the grammar of graphics of Wilkinson ([?]). For other communities (e.g. machine learning and signal processing) dealing with similar high dimensional data, EDA is not part of their education so few employ it out before formal algorithmic analysis or modeling. For areas such as information retrieval and information extraction, the original form of data are often text, not always well formulated or structured. The obvious is to map these data into the numeric form so to use traditional summaries and EDA visualization tools. New summary and visualization tools seem necessary for text and other IT forms of data.

Data volume and complexity are one side of the computer technology. The other side is the increased computing power to visualize data (multi-media representation) and to fit sophisticated models. The field of data visualization is rapidly advancing outside statistics.
Since visual processing units take more than one third of the cortex in human’s brain, it is necessary to use the superb information gathering ability of our vision. The efficient use of our vision is even more desirable for the complex data we face today. It relies on understandings of our vision from neuroscience and computer vision to render images from data, using spatial locations, perspectives, color, and ray tracing (e.g. shading). That is, it goes well beyond the plots used in EDA. Distributed computing is also being used for data visualization through clusters of graphical processing units and CPUs. For example, Levit (2006) [48] brings out in real-time different aspects of the data coming in from a digital sky survey by parallelizing graphical processing units and hundreds of displays. The visualization group at Lawrence Berkeley Lab (http://vis.lbl.gov) also conducts research on parallel graphics and visualization. Moreover, one government agency headed by Dr. Jim Thomas is the National Visualization and Analytics Center (NVAC) chartered by the U.S. Department of Homeland Security. It has set its objective in 2004 as “to define a five-year research and development agenda for visual analytics to address the most pressing needs in R&D to facilitate advanced analytical insight” in order to help “counter future terrorist attacks in the U.S. and around the globe” (cf. [2]).

It is natural to ask whether we could bring out more quantitative information in our data with these new visualization tools. Progresses have been made by the field of scientific visualization in this direction (e.g. Ben Fry’s website: http://acg.media.mit.edu/people/fry/). See Johnson (2004) [40] for an overview and most pressing problems in scientific visualization. We could add more to this enterprise, I believe, if we collaborate with researchers in visualization to represent results from modern methods such as machine learning and MCMC for model revision and validation as we did with residual plots in simple linear regression.

Complementary to extending our abilities through computer graphics and visualization to see more in data, we could use the computing power to simplify the data to visualize. Sophisticated modeling methods can first be used on data so that patterns could present themselves in output plots (after some boosting or SVM fits for example) which are not possible in displays of the original data. Residual is an obvious output to investigate as in classical statistics, but additional visualizations of other outputs (margin plots from SVM, fitting error plots along a boosting path are easy ones to think about) should be added to the routine diagnostics of a model. We could also search for meaningful low-dimensional structures in even high-dimensional data. If we find these structures, then the high-dimensional data can be reduced to low dimensions for visualization. Subject knowledge often suggest such dimensionality reduction or models in low dimensions, as seen in the cloud project and in Faraway and Reed (2006) [25] and Buvaneswari et al (2006) [13] in this special issue. When subject knowledge is not adequate, automatic dimensionality reduction methods can be tried to suggest possible meaningful data reduction. These methods help search for these structures at a speed impossible before. Data visualization and model fitting ought to be conducted iteratively, however. Seeing suggests models to fit and model fits give data to
This is similar to what we do in residual analysis for regression models, but residual plots are replaced with multi-media data representation and regression models are replaced by more general methods. Admittedly, this is easier said than done.

Recent years have witnessed much activities in automatic data reduction, for example, Kernel PCA (Schölkopf et al, 1998), ISOMAP (Tenenbaum et al, 2000, [66]), LLE (Roweis and Saul, 2000, [58]) and its extension using the Hessian matrix (Donoho and Grimes, 2003,[23]), and spectral clustering (Shi and Malik, 2000 [60], Ng et al 2001 [55], Belkin and Ngoyi, 2003 [7], Zhou et al, 2004 [80]). These methods generalize the traditional PCA and MDS because an eigen analysis, either locally or globally, underlies all of them. See Ham et al (2003) [34] for a nice theoretical synthesis of different dimensionality reduction techniques from the kernel point of view. Before these dimensionality reduction methods become routine EDA analysis, much more experience of applying them to real data sets and theoretical analysis are needed to understand the pros and cons of each method, in absolute terms and relative to each other.

**Semi-supervised learning**

The dimensionality reduction research is also closely related to the new area of semi-supervised learning in machine learning where information on the structure of the inexpensive unlabelled data (clusters or low dimensional manifolds) is used for classification or regression (cf. Belkin et al, 2004 [6], Zhang and Ando, 2005 [77]). With the pervasive existence of $p >> n$ in the IT data sets and the high cost of obtaining labelling or response information as in website classification, the side information in the unlabelled data or the predictors is crucial to take into consideration for implicit regularization and increased well-posedness of our statistical inference problem. For more information along this line of research, see the recent edited book Chappele et al (2006) [18].

In the loss function optimization approach of machine learning (Sec. 3.1.2), side information on groupings of predictors is recently built in the penalized loss function approach by many researchers including Yuan and Lin (2006) [76], Kim and Kim (2005) [45], and Zou and Hastie (2004) [82]. Zhao et al (2006) [79] propose a general Composite Absolute Penalty (CAP) framework to include the grouping structure and at the same time extend to the hierarchical structure among predictors. It is worth noting that the CAP framework could also facilitate group selection and enforce selection orders of predictors.

Both approaches require computing power to carry out the necessary eigen analysis or convex optimization. It is of great interest to investigate parallel computation to mitigate the high demand on computation for large $n$ or large $p$ cases. However, as will be argued in the second part of Sec. 3.2.3, an “imprecise” parallel computation of eigen analysis and convex optimization is enough and may actually help the statistical accuracies of the fitted models in terms of tasks such as parameter estimation and prediction.
3.2.2 Use new mathematical tools

Like it or not, we are leaving the comfort of the classical paradigm founded mathematically on calculus and a large sample size relative to the dimension of the parameters. The off-line data now often have more attributes or predictors than the sample size, the so-called \( p >> n \) situation. We still need asymptotics to see regularity, but not with a fixed \( p \). Hence the dimension of the space is growing with the sample size and most of our intuitions are derived from the 3-dim physical space that we live in. To paraphrase Aldous (book preface, 1989 [4]), since we are at a point with not much intuition to go on, analytical derivations might help lead us towards light. It is encouraging to see that theoretical results for the \( p >> n \) case are appearing in random matrices, linear modeling, Lasso under deterministic and stochastic assumptions, boosting, and covariance estimation. It is expected that much insight will be gained through such analysis. Becoming part of the new cyberinfrastructure demands that the analytical results should take into account, as much as possible, the algorithmic implementation of the methods, instead of assuming the entities to be analyzed is the exact maxima of objective functions which in practice cannot be obtained due to computational reasons. For comprehensive tutorials and new research material, check out SAMSI's recent workshop on high dimensional inference and random matrices: http://www.samsi.info/programs/2006ranmatprogram.shtml. It remains to be seen whether the random matrix results are to become something of an equivalent of the central limit theorem in classical statistics. In any event, much distilling needs to happen to simplify the methods used to derive these results before they enter the analysis toolkit of routine statistical investigations of massive data to provide insights.

3.2.3 Bring communication and computation considerations into our framework

Statistics under access contraints

The “Towards 2020 Science” report states that “the way scientists interact with data and with one another is undergoing a fundamental paradigm shift.” The paradigm is shifting from the traditional “experiment \( \rightarrow \) analysis \( \rightarrow \) publication” to the new ”experiment \( \rightarrow \) data organization \( \rightarrow \) analysis \( \rightarrow \) publication”. In Section 3.1 above, data transmission by moving the data to R or Matlab does not deal with the new stage of “data organization” and this section intends to discuss the impacts of this new stage of science research on statistics and vice versa.

As discussed earlier, reducing the data size so we could handle the data at our site (via R or Matlab or C++ codes) runs the risk of losing important information. The recent story of Google’s machine translation success (Norvig, 2006, [56]) confirms exactly this point. Google’s translation system is based on analyzing an incredibly large database of documents and their human translations to carry out its own translation. Databases of such a size can only be housed by a few specialized companies such as Google and Microsoft. It is a com-
mon belief in the machine translation community that results from a reduced sized database housed by a research unit in a university would not come close to the Google results. It is also worth noting that memory constraints prompted the Google group to use binning (e.g. regularization) on the character strings to achieve better translation accuracy. This is another example of communication constraints leading to a better statistical accuracy. For a digital sky survey, the aim is to find very sparse signals (quasars, for example). Reducing the size by random sampling is likely to not include these targeted signals altogether. In other words, there are important technology and science problems to which the approach described in Section 3.1 would not be applicable. Thus arises the need for statistical methods to interact with databases. This interaction goes both ways. The easier direction is to understand database structures and design methods with fast implementations on the databases—the purpose of data mining. With the recent advances of research in machine learning and statistics, it is high time to integrate them into data mining software with necessary modifications to suit the database. The harder direction is to influence database design so that data in a database can be accessed swiftly by a slew of statistical algorithms. To achieve this, statisticians or data analysts have to reach consensus on what are the basic operations we need to conduct on a database for most if not all statistical analysis algorithms. In return, the size of the database might prohibit the use of any algorithm which is worse than linearly scalable to the size of the data. This points to the same question of what is the most efficient statistical method subject to a computational constraint which will be discussed later.

Karr et al (2006) [44] in this special issue is a good example of interaction between statistical methodology (parametric) and parallel databases. They devised an incremental method to run most parametric estimation over multiple databases with security requirement. Another area of importance where access constraints are dominant is sensor networks. Sensor networks are self-networked small devices that are engineered to collaborate with each other and collect information concerning the environment around them. Their flexibility greatly extends our ability to monitor and control the physical environments from remote locations. Their applications range from seismic, natural environmental monitoring, to industrial quality control, and to military usages. However, the sensors are constrained by the battery power whose major consumer is communication (data transmission), and then to a much lesser extent computation. So far sensor network research is dominated by researchers from computer science and electrical engineering, but it provides one of the ideal platforms for us to integrate statistical analysis with computation, data compression and transmission because the overriding power constraint forces us to consider all the players in the same framework to maximize the utility of the limited battery energy. It would be interesting to try to devise a framework to encompass components 2 (transmission and compression) and 4 (formal modeling) to answer optimality questions. Distributed algorithms are also very desirable because data transmission among the sensors is expensive. For more discussions on sensor network research, see Hansen (2006) [35] in this special issue.
Both the database and sensor networks applications call for systematic studies of statistical inference under access constraints. Much research is necessary in this area, especially algorithm development and theoretical analysis which should be related, rather than isolated, activities, in order to put statistics squarely in the new cyberinfrastructure.

**Computation for data with uncertainty or noise**

Computation was not a concern in Fisher (1922), but is central to a statistical investigation today. There is something very novel about boosting (and fitting neural networks): the computation parameter, the number of iterations, serves also as a regularization parameter in statistical estimation. BLasso has a similar property. It is a component-wise gradient descent algorithm with a fixed step to minimize the generalized Lasso loss (convex loss and penalty functions) simultaneously for different values of \( \lambda \)'s. It shares much similarity with boosting when a component-wise gradient descent algorithm is used, or the forward stage-wise regression (FSR) as called by Efron et al (2004). That is, BLasso has a forward step just as in FSR, but with a backward step added to make sure the combined penalty is minimized not just the loss function part which is the aim of boosting. Moreover, BLasso solves a sequence of optimization problems corresponding to different \( \lambda \)'s similar to the Barrier method in optimization (Boyd and Vandenberghe, 2004 [9]).

The coupling of computation and regularization in boosting and BLasso is reminiscent of the equivalence of computation and modeling in K-complexity theory. Relative to a universal Turing Machine, the K-complexity of a binary string is defined as the length of the shortest program which prints out the string and stops. Because of an equivalence of a (prefix) program and a probability distribution, there is an equivalence of computation (program) and modeling represented by the distribution. In spite of the fact that K-complexity is not computable, this equivalence has an intriguing intellectual appeal.

Let us entertain ourselves further by looking into modeling and computation practiced by us today. We know that statistical model fitting uses scientific computing, but statistical computation is special. Even in the parametric case, there is a well-known result that only one Newton or second-order step is needed to make a \( \sqrt{n} \)-consistent estimator efficient. That is, since our objective function is a random quantity, we do not need convergence of the minimization algorithm to get a statistically satisfying solution, as shown in boosting. In nonparametric methods such as boosting, neural nets and BLasso, early stopping before convergence saves computation and regularizes the fitting procedure and hence results in a better statistical model. Again, computation and model fitting seem to be working in the same direction – less computation and better statistical accuracy. These facts indicate the intimate relationship between computation and model fitting. They prompt us to ask the following question:

*Is there a minimal amount of computation needed for a certain statistical accuracy?*
It is not clear whether this question is answerable because fast algorithms in scientific computation often rely on closed form equations or relationships derived through analytical means. Analytical calculations are infinite precision, while scientific computations are finite precision. Nevertheless, we believe it is a very interesting intellectual question to ask and the pursuit of the answer to this question could lead to useful practical consequences for modeling IT data.

**On-line or streaming data**

On-line or streaming data analysis is the result of the second characteristic of the IT data: high data rate. High dimensionality has attracted much attention within the statistics community, evident from the large number of conferences and workshops with this phrase in the titles. Streaming data analysis, however, remains mostly outside of the spotlight of research in statistics, with an exception of the special issue of Journal of Computational and Graphical Statistics in 2003 from which many papers are cited in this paper.

Due to the real-time requirement of streaming data analysis and the huge volumes of data coming in, the desired speed for extracting information from data is much higher than that in the batch or off-line mode. However, to design a fast on-line algorithm, batch data analysis may be necessary to identify which features of the data to retain. On the other hand, before batch data can be collected, data reduction or feature selection has to be carried out on-line to reduce the data volume for storage. For example, downsampling or aggregation may be necessary for batch mode analysis.

Similar to the statistical science discussion paper Chambers et al (2006) [17], many on-line or streaming data algorithms repeatedly update a low-dimensional feature distribution and identify outliers or anomalies relative to this distribution. Other streaming algorithms deal with records (of phone calls, for example) and have a distinctive discrete flavor. Interested readers should read Gilbert and Strauss (2006) [32] of this issue, which is a synthesis of (discrete) streaming algorithms from the point of view of group testing.

It would be very interesting to see what and how machine learning methods can be adopted, with necessary modifications or completely new designs, to the on-line setting.

On the analytical side, there has been an excellent body of research on on-line prediction in the individual sequence setting (i.e. results hold for all sequences). These results are now nicely collected in a 2006 book (Cesa-Bianchi and Lugosi, 2006, [16]). But the theory has been developed primarily without empirical data experience, but things are changing as shown in the recent paper by Cesa-Bianchi and Gentile (2005) [15].
3.2.4 Data fusion

In the arctic cloud detection project, we implemented a very simple form of data fusion of data from two sensors on the same satellite Terra: a consensus label was given only when our MISR-sensor based algorithm gives the same label as the MODIS-sensor operational algorithm. This is a fusion at the decision level. We have also fused the two sensor data at the feature level by applying quadratic discriminate analysis to the three MISR features and the five MODIS features for the final MISR-MODIS soft labeling. In Dass and Jain (2006) [21] of this issue, fusion of fingerprint with other biometric traits is addressed and three fusion levels are mentioned: feature level, matching score level and decision level. It is clear that other levels of fusion might be considered depending on the problem. As we have seen in Section 2 for text data, data fusion is necessary in many other fields such as genomics research (fusion of gene expression and sequence data) and climate modeling (fusion of simulation and observation data). These data fusion tasks are often related to large research or government projects and involve huge amounts of data and on-line or streaming data might be a requirement as well. It is therefore a research frontier for us to get involved and make significant contributions, for instance, by providing a framework to think about data fusion relative to communication and computation constraints.

4 Conclusion

After reviewing the current state of computing technologies, we use many examples to demonstrate the diverse sources of data and their equally diverse requirements for extracting information from them. The main difference and advantage (and/or disadvantage) of our time from Fisher’s is the availability of computing technology and consequently the availability of massive amounts of data. We argue that, for the sake of statistics’ healthy existence, solving real data problems has to be our aim, and we have to find a way to join the imminent cyberinfrastructure development.

To achieve the goal of solving real problems, many exciting challenges have to be met: we have to interact efficiently with databases and other data sources such as sensor networks, design new EDA visualization tools, use new or non-conventional mathematical results, develop new statistical algorithms satisfying communication and computation constraints, and devise new statistical inference paradigms to encompass such endeavors.

It is a time of data deluge, yet we could help build the Ark and ride on it, if we so choose.
5 Acknowledgements

This work was partially supported by NSF Grant DMS-03036508 and ARO Grant W911NF-05-1-0104, and a Guggenheim Fellowship. As incomplete as this work might seem, it would not have been possible without the support and encouragements of many colleagues. The author is very grateful for very helpful discussions and comments from Leland Wilkinson, David Purdy, Tilmann Gneiting, Martin Wainwright, John Rice, Mike Jordan, Nicolai Meinshausen, Dan Klein, Noureddine El Karoui, and Hang Li. She is also very grateful to two referees and the guest editor for their constructive criticisms on the first version of the paper so that technically the paper got re-written.

References


22 D. Donoho. For most large underdetermined systems of linear equations, the minimal l1-norm near-solution approximates the sparsest near-solution. Technical report, Department of Statistics, Stanford University, 2004.


