Complex Algorithm R&D
A Guide for the Perplexed

John F. McGowan, Ph.D.
jmcgowan11@earthlink.net

Outline

- Introduction
- Example Projects
- Project Scope and Feasibility
- Return on Investment
- Genuine Breakthroughs
- Mathematical Models
- Software Engineering

Introduction

- Complex algorithm research and development projects can have a return many times the amount invested.
- Working Definition: A complex algorithm is an algorithm that requires at least 1000 lines of C/C++ code embodying advanced mathematical or logical concepts.

Introduction

- Complex algorithms are now widely used.
- Markets of billions of dollars. Digital video is a prominent example. HD DVD/Blu Ray, DVD, YouTube…
- U.S. DVD Video market is around $25 billion dollars in revenues per year.

Introduction

- Potential markets for new algorithms and applications are even larger.
- Complex algorithms may solve big, trillion dollar problems.
  - Energy
  - Disease (cancer, heart attack, etc.)

Introduction

- Global annual energy market is over $1 Trillion
  - Cost of refined petroleum products is soaring (2008)
  - World may be running out of oil
  - Raising the standard of living of the world to the US level requires a vast increase in world energy production (5-15 times the current level).
- Development of new energy technologies such as thermonuclear fusion may require sophisticated mathematical modeling of electromagnetic, nuclear, and plasma effects.

Current Examples

- Video Compression (MPEG, H.264, H.263, HD DVD, BluRay, DVD, Windows Media)
- Speech Recognition Engines (Phone Help Systems, Dragon Naturally Speaking, etc.)
  - Speech leader Nuance reports over $600 million in 2007 revenues.
- Encryption Software (AES, RSA, etc.)

Current Examples

- 3D Graphics Rendering Engines
- Pricing Optimization
- Options Pricing Models (Finance)
- Automobile Traffic Models (Inrix Traffic Fusion Engine)
- Spam Modeling and Detection

The Future

- Telecommuting using video compression and high resolution wall displays replaces the daily commute avoiding billions in transportation costs and millions of man-hours.
  - Present day annual transportation costs in the USA exceed $300 billion (and rising with oil prices).

The Future

- Human-like speech recognition enables hands-free command and control and dictation avoiding many costs, including:
  - Order entry by voice
  - Billing by voice
  - Customer service by voice
  - Transcription of live and recorded speech

The Future

- Computer-aided design of electrostatic fusion devices enables powerful tabletop reactors the size of a basketball and costing a few thousand dollars.
- Molecular modeling of cancer and other diseases results in cures.

Typical Projects Today

- Converting a prototype to a production system.
- Porting a complex algorithm to a different platform
- Technical feasibility assessment
- Developing a prototype or proof of concept
- Statistical Data Analysis
- Mathematical Modeling
- Research and development of a new algorithm

A Little About Math

- Most commercial software today involves at most lower level high school math.
  - Addition, subtraction, multiplication, division
  - Simple averages, elementary statistics.
- Even many high school mathematical methods are rare outside of computer graphics
  - Trigonometry
  - Pythagorean Formula $a^2 = b^2 + c^2$
  - Square Roots and Powers
  - Quadratic Formula (roots of $ax^2 + bx + c = 0$)

A Little About Math

- Most complex algorithms today utilize mathematics currently taught in 1st and 2nd year math courses at a top university.
- More advanced math (3rd, 4th year, graduate) occurs occasionally
  - The General Theory of Relativity is used in the Global Positioning System (GPS).
  - Some advanced group theory is used in encryption.
- Some common types of math in complex algorithms
  - Linear Algebra/Matrices
  - Statistics
  - Fourier Transform
  - Other Linear Transforms

A Little About Math

- Higher level advanced math may become more common in the future.
  - Pattern recognition probably requires more advanced math.
- The math found in complex algorithms is closest to modern applied mathematics or the advanced mathematics of the 19th century before the triumph of abstraction in pure mathematics.

Project Scope

- Most projects require between four (4) man-months and several man-years.
- Individuals or small teams.
- A careful review of data (SEC filings, web sites, press coverage, research papers etc.) from past projects in a specific area will usually confirm similar project scopes.

Shorter Projects

- Some projects can be done in a few weeks or months. Not typical.
  - Technical Feasibility Assessments
  - Proposal Development and Writing
  - Some proof of concepts and prototypes embodying known, proven algorithms.
    - Often done with a tool such as Matlab or Mathematica
  - Miscellaneous small projects almost always involving known, proven algorithms.

Most Projects are Longer

- Converting a prototype to a production system.
  - In most cases, this takes several months to years.
- Porting a working algorithm to a new platform.
  - This usually takes a while. Occasionally, it goes smoothly as one might naively expect, but usually not.

Return on Investment

- Return on investment for a successful project can be very high.
- Cost of even a large multi-year project is a few million dollars.
  - Ten (10) algorithm developers (large team) over five (5) years (long project) has a total cost of about $7.5 million at current U.S. rates. (Using 150 K/year per Full Time Employee)
  - One (1) algorithm developer over six (6) months (small project) has a total cost of about $75,000 at current U.S. rates.
- A home run can solve a billion dollar or larger problem and bring in hundreds of millions of dollars, even billions.
  - Return = $100 M / $7.5 M = 13.3 (small home run)
  - Return = $1 B / $7.5 M = 133 (big home run)

Common Project Problems

- Complex algorithms often have a high degree of coupling between different parts.
- Similar to a mechanical clock or automobile engine where all the parts must work together within small tolerances for the entire system to work at all.
- Hence the common use of the term engine to describe implementations of complex algorithms.

Common Project Problems

- The tight coupling of the parts means more and longer debugging per line of code than other software projects.
- Often, every bit must be correct.
- Experience with other software development projects such as web sites, user interfaces, or database reports is often misleading.
- Not so agile. Turnaround time between requests and results is usually at least weeks, often months, even years for major projects.

Project Feasibility

- Technical feasibility is difficult to assess.
- Some types of projects are usually feasible
  - Porting a working algorithm to a new platform
  - Converting a working prototype to a production system
  - Implementing a proven algorithm for a new application
  - Minor refinements of proven algorithms

Project Feasibility

- Once again, it is easy to misjudge the technical feasibility of projects!
- Long history of exaggerated claims regarding complex algorithms that duplicate aspects of human intelligence.
- Long history of exaggerated claims for advances in data compression.
- Caveat emptor

Some Famous Flops

- Pen Computing (early 1990's)
  - See Jerry Kaplan's StartUp, one of the few books on one of the many failed startups that hinged on complex algorithms, that is handwriting recognition.
- Lernout and Hauspie (speech recognition)
  - Major financial scandal
  - See press coverage and court records

Data Compression Hype

- There has been enormous success in data compression over the last several decades. See next slide.
- Nonetheless, there is also a long history of exaggerated and questionable claims about data compression.
- Video and other data compression involves complex algorithms that are difficult to evaluate.

Some Famous Successes

- MPEG Audio and Video Compression
  - Video CD (MPEG-1)
  - DVD (MPEG-2)
  - Digital Cable TV (MPEG-2)
  - High Definition Successors to DVD (H.264/MPEG-4)
    - BluRay
    - HD-DVD
  - MP3

Some Famous Successes

- Major advances in video compression reach market in 2003. Enable web video, YouTube, etc.
  - Embodied in H.264/MPEG-4
  - Windows Media 10
  - Flash video formats
  - Other video formats

Some Famous Successes

- Pre-2003 Bitrates
  - MPEG-1 (roughly VHS quality) 1 Megabit/second
  - MPEG-2 (DVD) 4-8 Megabits/second
- 2003
  - MPEG-4/H.264 275 Kilobits/second (basic DSL)
    - With tuning, a subjective quality close to DVD level can be achieved at around 275 Kilobits/second
  - Less obtrusive compression artifacts
    - "cartoony" look, fine details washed out.
    - Occasional perceived jitter of edges or images

Project Feasibility

- Artificial Intelligence (AI) is an unsolved problem. Many names and sub-fields:
  - Human Reasoning
  - Speech Recognition
  - Object Recognition and Tracking
  - New buzzword or phrase every few years

Project Feasibility

- These areas probably require fundamental research and the development of new mathematical or logical methods.
- Vast market opportunities exist in these areas. Billion, even trillion dollar markets.
- They are difficult.

Genuine Breakthroughs Needed

- Genuine breakthroughs are rare.
- Appear to have been more common before the "professionalization" of science during and after World War II.
- Often involve a new system architecture, concept, or mathematical expression.

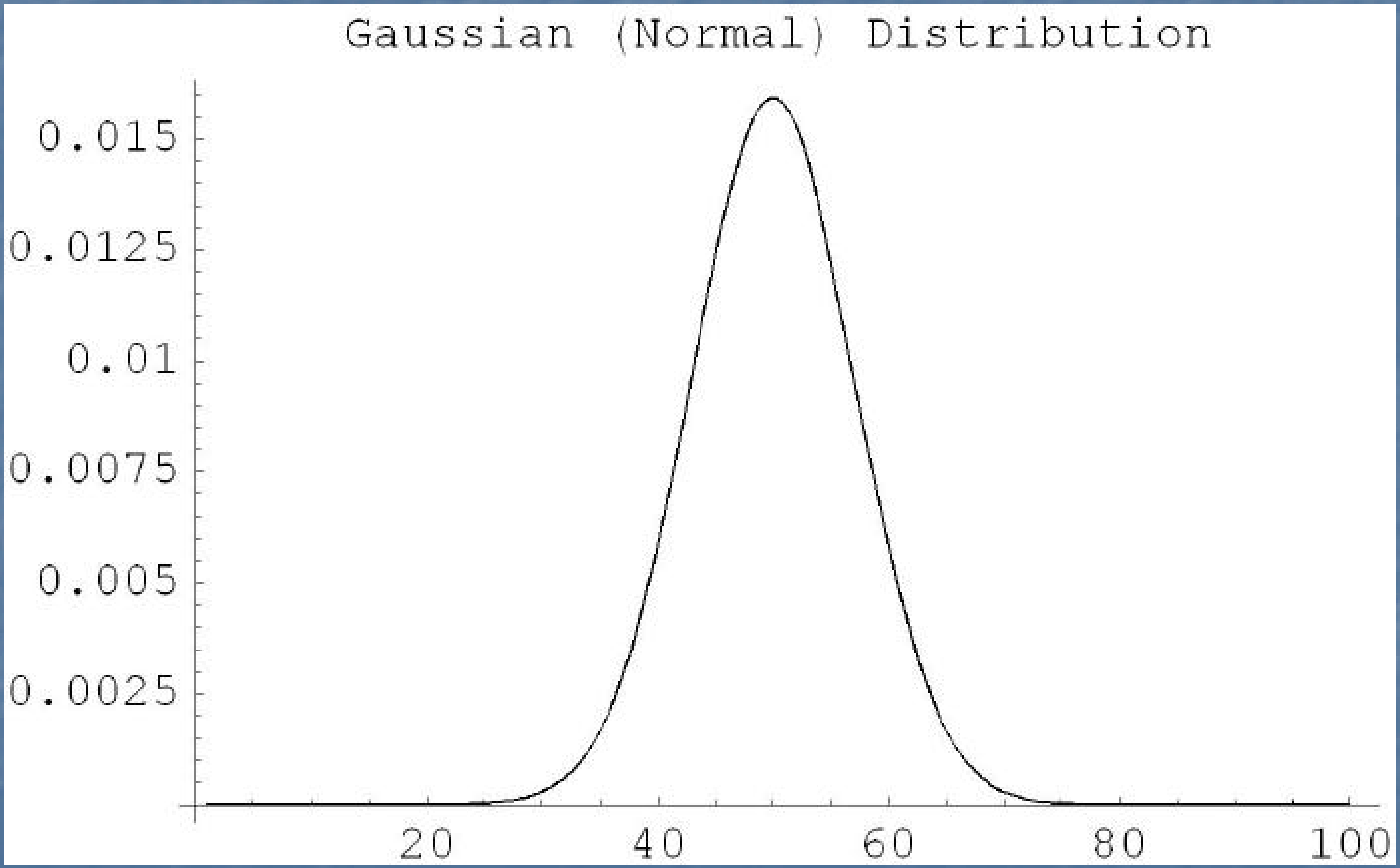
Genuine Breakthroughs Needed

- Breakthroughs typically require between five (5) and twenty (20) years of work.
- Large amount of trial and error.
  - Mistakes
  - Blind alleys
  - Luck
- Most historical breakthrough scientific discoverers or inventors spent several years with negligible demonstrable progress before their insight or insights.

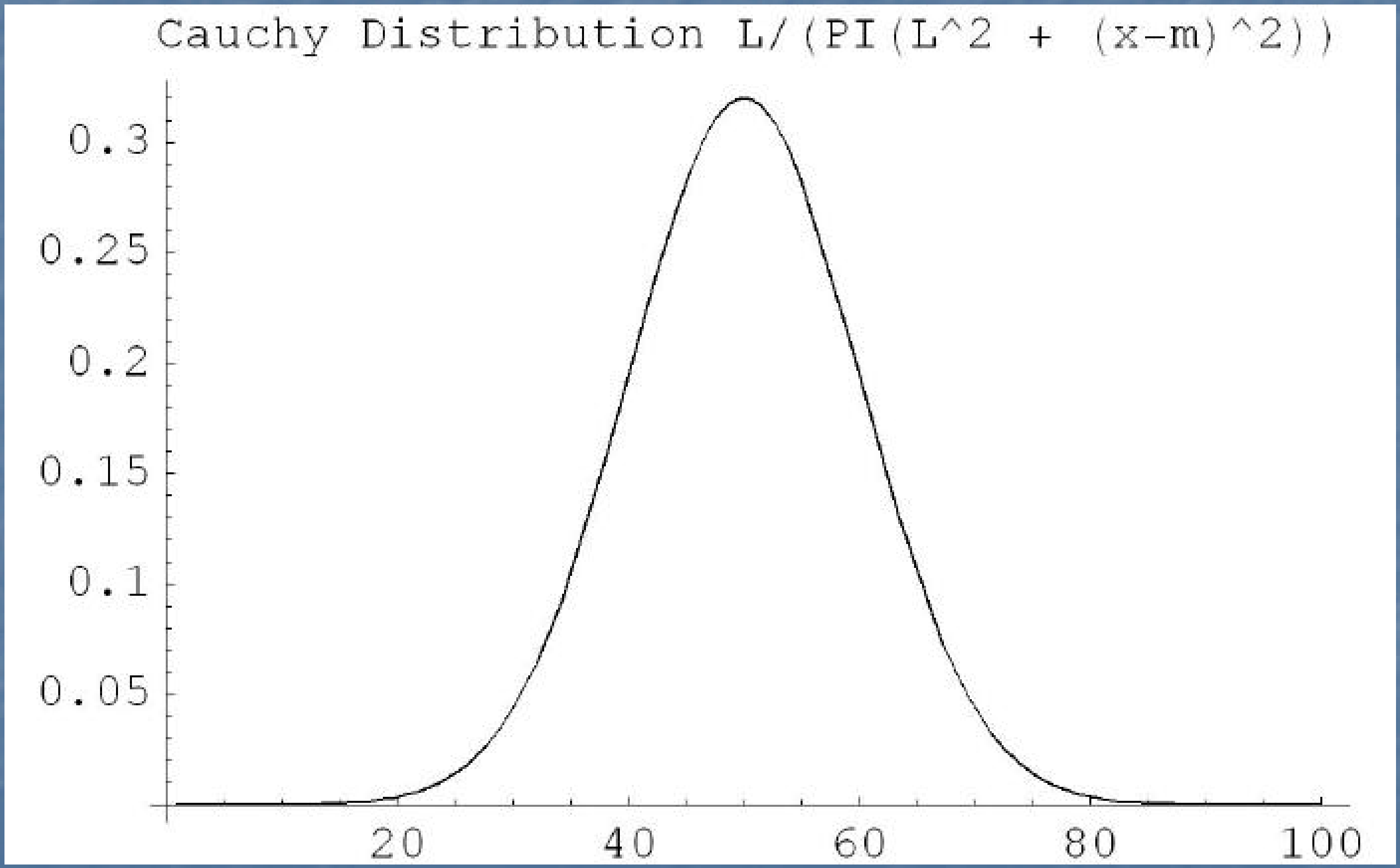
Genuine Breakthroughs Needed

- Not like most mainstream science or engineering today.
- Requires developing or identifying a new, unknown mathematical expression that matches data.
- Most mainstream science and engineering involves constructing mathematical models from a small set of known, often widely studied functions such as the Gaussian (also known as Normal) distribution, Cauchy distribution, etc.

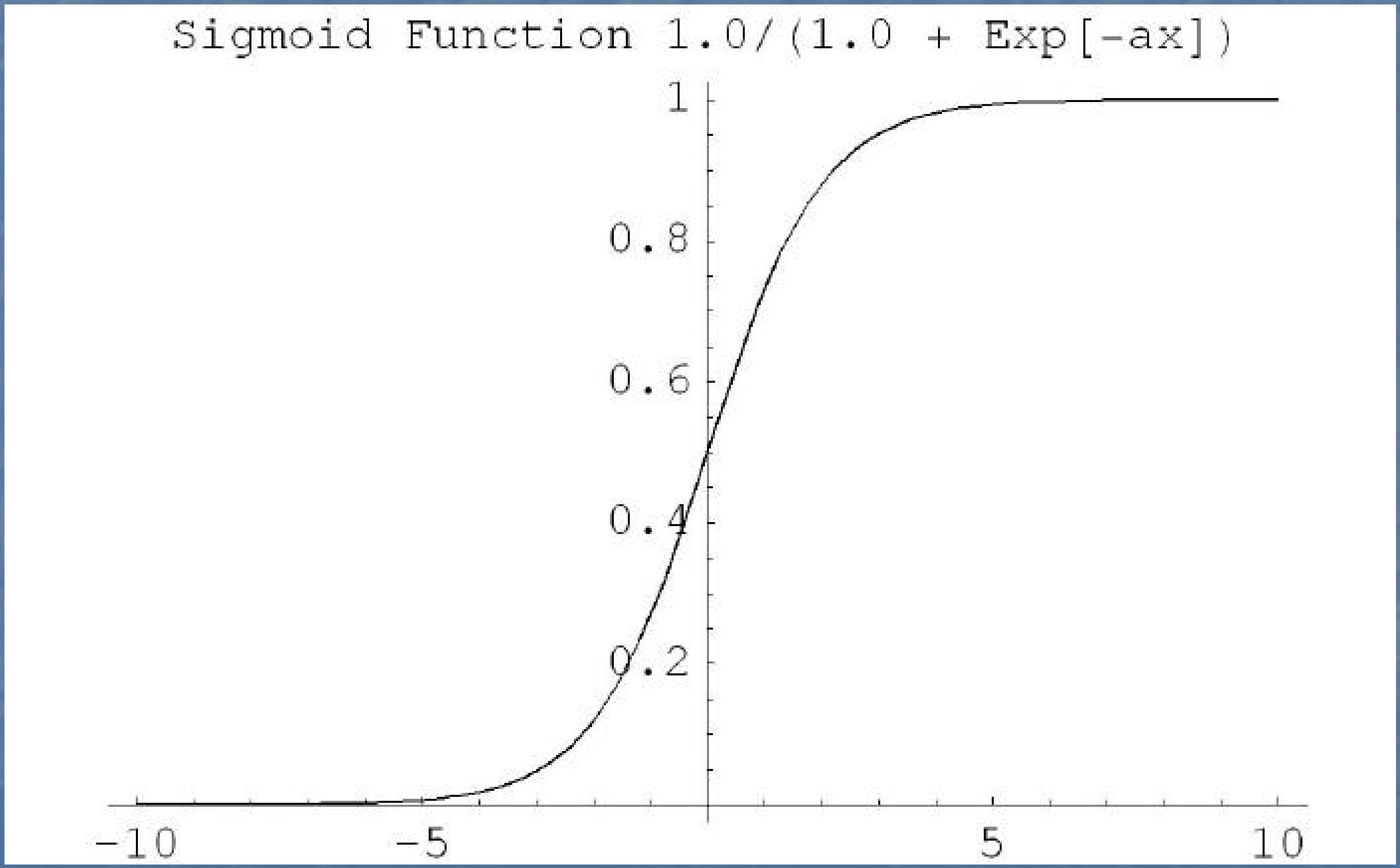
Common Building Blocks



Common Building Blocks



Common Building Blocks



Common Building Blocks

- Gaussian
  - Key building block of speech recognition models (Dragon, MS Speech, Sphinx, Nuance, etc.)
- Cauchy
  - Accurate model of the frequency response of an LRC electrical circuit (for example, a radio receiver)
  - Key building block of articulatory models of speech production (Gunnar Fant and successors)
- Sigmoid
  - Key buildng block of many neural network pattern classifiers (object, handwriting, face, etc. recognition).

Common Building Blocks

- In many cases, complex models constructed from common building blocks fail to reproduce reality.
- Speech synthesized using articulatory models (Cauchy distribution) does not sound like human speech although it has the same gross spectral characteristics.
- State of the art speech recognition algorithms (Gaussian distribution) are much less accurate than humans.
- Neural networks (sigmoid functions) are rarely successful in practical applicatons.

Common Building Blocks

- The persistence of this problem after decades of intensive heavily-funded research suggests that some new or at least not widely known mathematics is involved.
- A conceptual breakthrough is probably needed.
- Easier said than done.

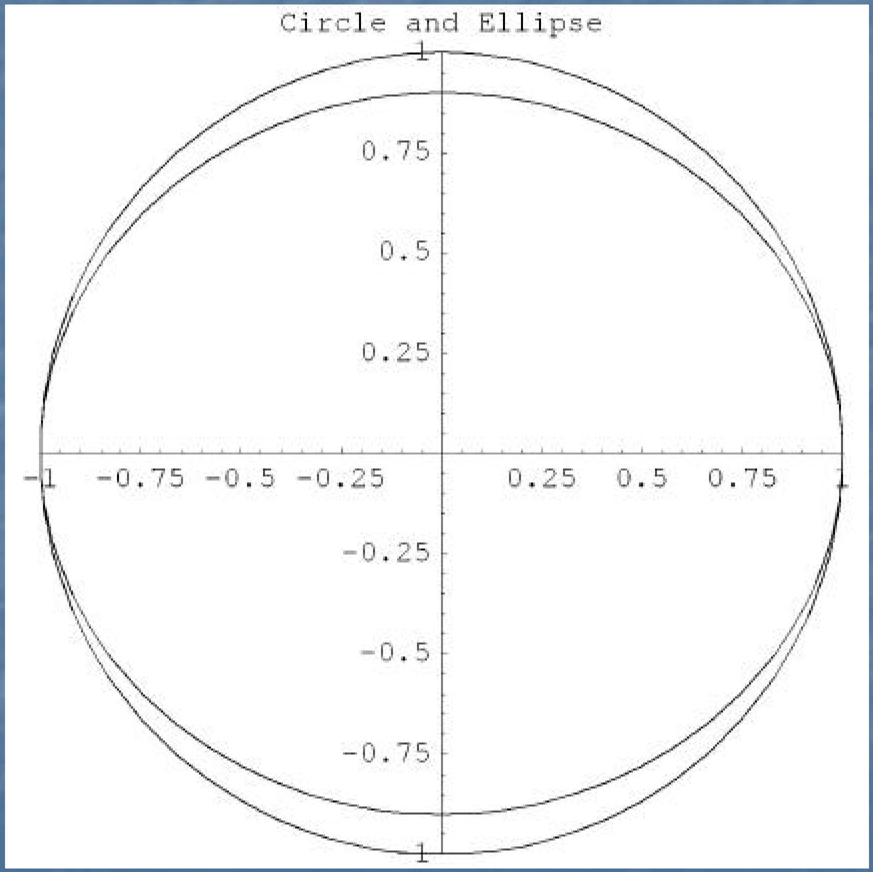
The True Breakthrough: A Lost Art

- Lengthy written discussions of concepts often illustrated with simple drawings are common.
- Use of various methods to convert the words and pictures to new mathematical expressions.

Some Historical Examples

- Johannes Kepler's New Astronomy
  - Planets travel in elliptical orbits with the Sun at one focus, sweeping out equal areas in equal time.
- Michael Faraday, William Thomson (Lord Kelvin), and James Clerk Maxwell, discovery of Maxwell's Equations, the basis of modern electrical technology.
- Pythagorean Theorem (Note: the formula, if not the proof, was known in ancient Sumeria, long before Pythagoras)

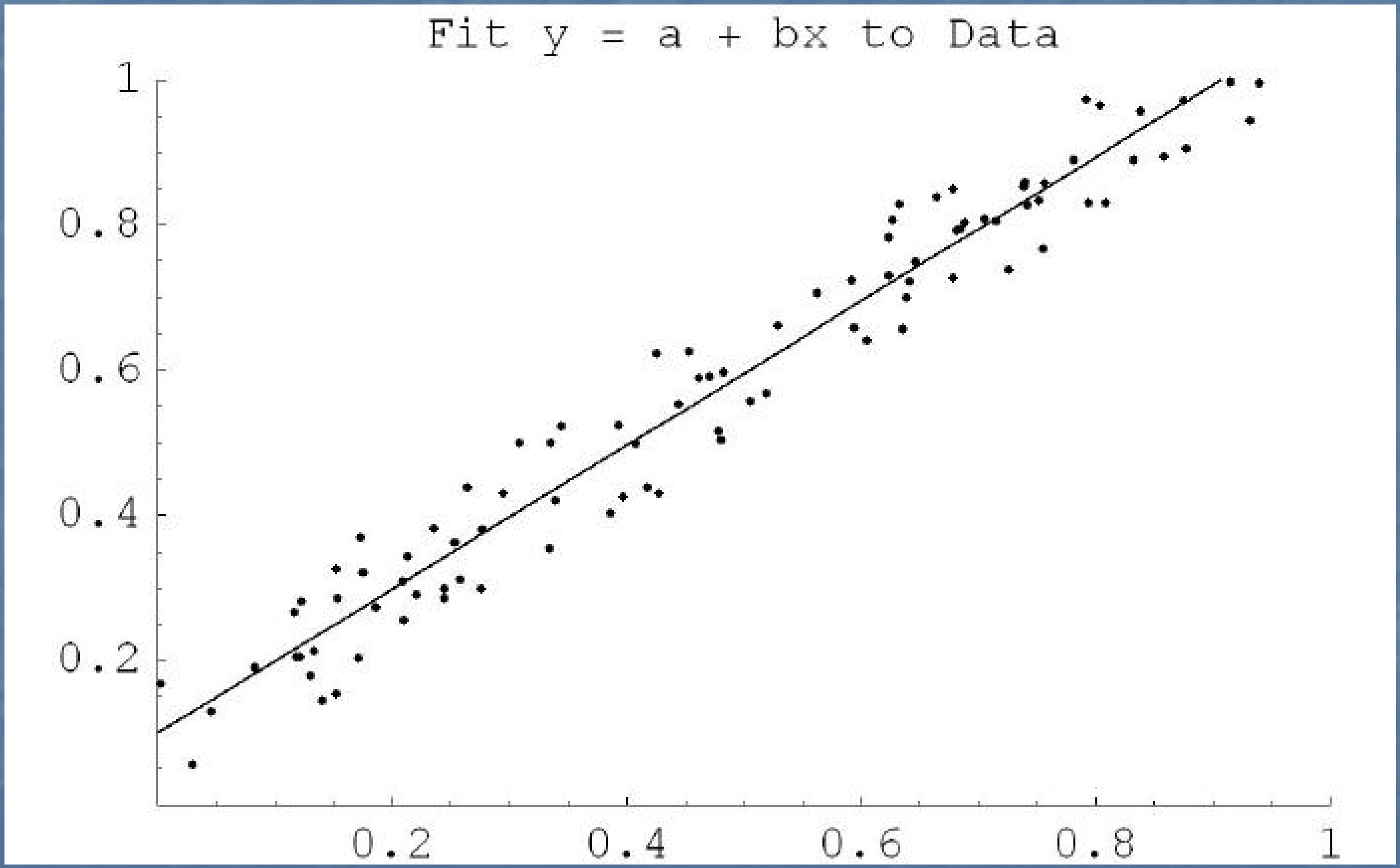
Kepler's New Astronomy



Mathematical Modeling

- Many complex algorithms are mathematical or statistical models of real world processes.
  - Speech Recognition
  - Financial Models
  - Automobile Traffic Models
- Models usually have model parameters that are found by fitting the model to real world data.

A Simple Fit to Data



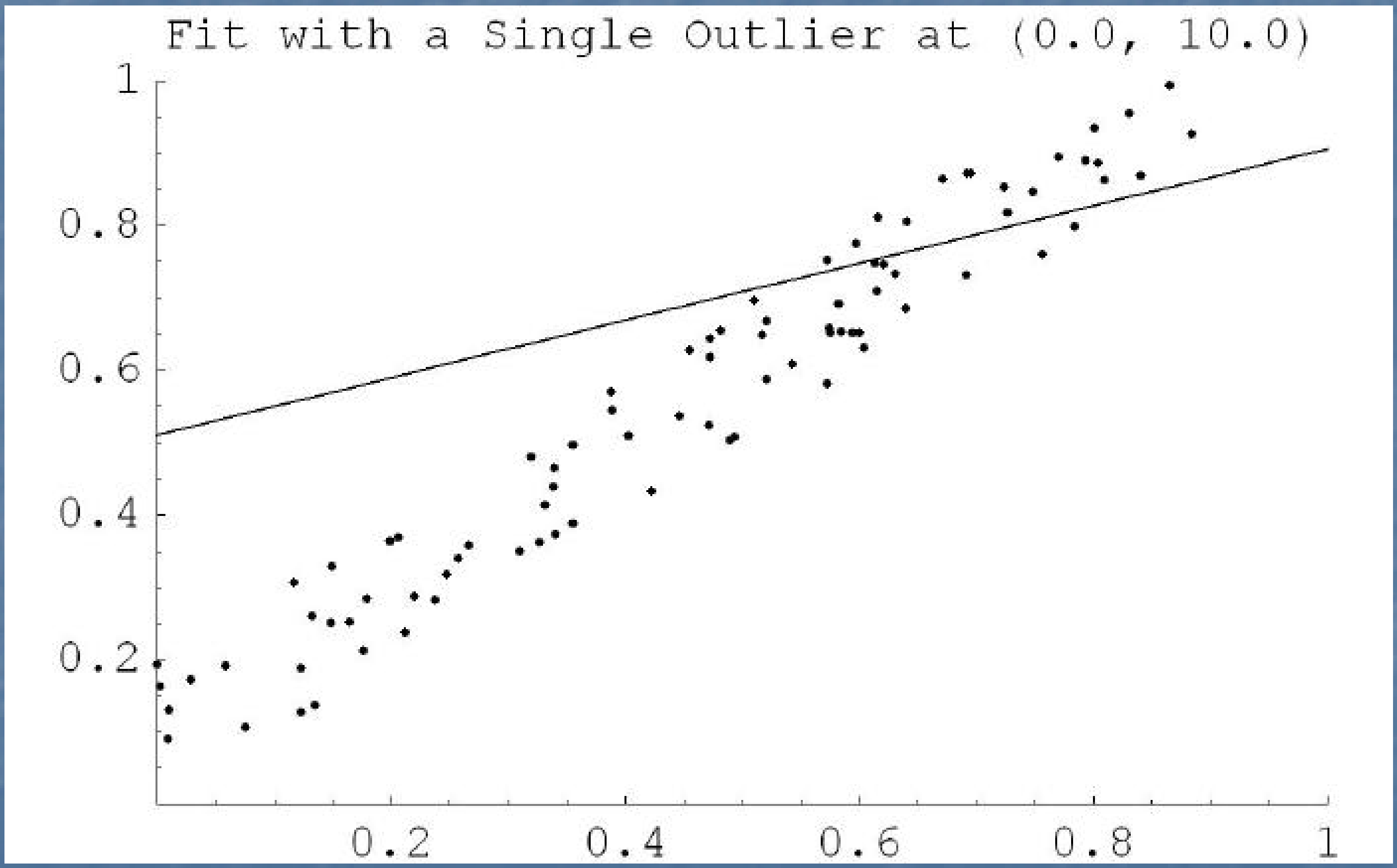
Mathematical Modeling

- Many advanced fitting methods known
  - Least Squares Fitting
  - Maximum Likelihood Estimation
  - Robust Methods
  - Levenberg-Marquardt
  - Davidon-Powell-Fletcher
  - Polytope (or simplex) method
- Many subtleties

Model Fitting Pitfalls

- Model Fitting Remains an Art
- Outliers
- Goodness of Fit
- Fit Methods Sometimes Fail
- Correlations between fitted parameters
- Complex models with many parameters often match current data but predict the future poorly!
- Simple Models often predict best (Holy Grail)

A Simple Fit to Data with an Outlier



Simple Models

- Predict system based on a few input parameters.
- Analyze data to find simple predictive relationships.
- This is hard to do. Complex models are easier to create.
- Kepler, for example, spent over five years from 1599-1605 discovering the elliptical orbit of Mars and other planets.
- The Lost Art

Software Engineering

- Often easier to research and develop models using a tool such as Matlab, Mathematica, AXIOM, or Maxima.
  - Scripting languages similar to Python or Perl
  - Implicit variable declaration
  - Comprehensive, well-integrated library of mathematical, numerical, and statistical functions.

Mathematica vs. C/C++

- Adding Two Vectors
- ```
  A = {1.0, 2.0, 3.0};
  B = {1.1, 0.0, 4.0};
  C = A + B
  Out[1]={2.1, 2.0, 7.0}
  ```
- ```
  #include <iostream.h>
  double A[3] = {1.0, 2.0, 3.0};
  double B[3] = {1.1, 0.0, 4.0};
  double C[3];
  int index;

  for(index = 0; index <3; index++)
  {
  C[index] = A[index] + B[index];

  }

  cout << “{“ << C[1] << “,” <<
  C[2] << “,” << C[3] << “}” <<
  endl;
  ```

AXIOM vs. C/C++

- Adding Two Vectors
- A := vector[1.0, 2.0, 3.0];
  B := vector[1.1, 0.0, 4.0];
  C := A + B
  [2.1, 2.0, 7.0]

```
#include <iostream.h>
double A[3] = {1.0, 2.0, 3.0};
double B[3] = {1.1, 0.0, 4.0};
double C[3];
int index;

for(index = 0; index <3; index++)
{
C[index] = A[index] + B[index];

}

cout << "{" <<  C[1] << "," <<
C[2] << "," << C[3] << "}" <<
endl;
```

Software Engineering

- Drawback is speed and memory constraints often require conversion of the finished algorithm to a faster programming language such as C/C++, Java, <insert your favorite language here>.
- If throughput is low, can use a server or servers with Matlab or Mathematica code to compute results.

Software Engineering

- May need to convert the finished algorithm to a fast programming language.
  - Require good libraries of mathematical, numerical, and statistical functions for rapid conversion.
- One can research and develop the algorithm in a fast programming language.
  - Avoids conversion costs, speed and memory issues.
  - My experience is that tools such as Matlab and Mathematica are often much better for algorithm R&D.

Leading Algorithm R&D Tools

- Matlab
  - widely used in commercial world (Digital Signal Processing).
- Mathematica
  - works just as well
  - widely used in academic and government R&D, Wall Street finance.
- AXIOM
  - free, open-source, reputedly just as good.
  - Berkeley style license
  - Many features. Started in 1971. 300 man-years.
- Maxima
  - Free, open-source
  - GNU Public License (GPL)

Fast Programming Languages

- ANSI C
  - Almost universally available for any device, processor etc.
  - Fast, efficient, low memory use, hard to reverse engineer compiled binaries.
- C++
  - Object-oriented
  - Usually less efficient, more memory use than C.
- Java
  - Object-oriented
  - Compiled to byte codes, often slower, less efficient.
  - Easier to reverse engineer.

The Dream Tool

- Algorithm R&D Tool similar to Matlab, Mathematica, AXIOM, etc.
- Available for all platforms
- Compiled to optimized binaries
  - Same speed as C/C++ binaries
  - Same memory requirements as C/C++ binaries
- Integrated GUI Builder similar to Visual Basic
- Integrated network and web support

Conclusion

- Complex Algorithms
  - Project scope is significant ($75K to $7.5M)
  - Project feasibility is difficult to assess.
  - Breakthroughs are unpredictable, take time.
  - Some standard software methods exist.
- Return for a success can be 5-1000 times investment
- Questions
- Contact: jmcgowan11@earthlink.net

References

- Some Complex Algorithms
  - http://www.chiariglione.org/mpeg/ (MPEG compression, one of the great success stories)
  - http://www.videolan.org/developers/x264.html (x264 is a free, open-source h.264 video encoder)
  - http://cmusphinx.sourceforge.net/ (The Carnegie Mellon Sphinx Project, an open-source speech recognition engine)
  - http://www.itk.org/ (National Library of Medicine Insight Image Registration and Segmentation Toolkit)
- Algorithm R&D Tools
  - http://www.mathworks.com/ (Matlab)
  - http://www.wolfram.com/ (Mathematica)
  - http://www.axiom-developer.org/ (AXIOM)
  - http://maxima.sourceforge.net/ (Maxima)

References

- StartUp: A Silicon Valley Adventure, by Jerry Kaplan, Houghton Mifflin Co, Boston, 1995
  - Note that the author devotes only a few pages to the development of the handwriting recognition algorithms which would have been essential for the success of GO, his failed pen computing startup.
- "How High-Tech Dream Shattered in Scandal at Lernout & Hauspie", by Mark Maremont, Jesse Eisinger, and John Carreyrou, Wall Street Journal, December 7, 2000

References

- New Astronomy, by Johannes Kepler, Translated from the Latin original by William H. Donahue, Cambridge University Press, Cambridge, UK, 1992
  - One of the most important and difficult mathematical analyses of data in history.
  - Originally published in 1609.
  - Major advances in artificial intelligence and related fields probably require a similar mathematical advance.