The *DataModeler* Package

Evolving Insight from Data

Mark Kotanchek

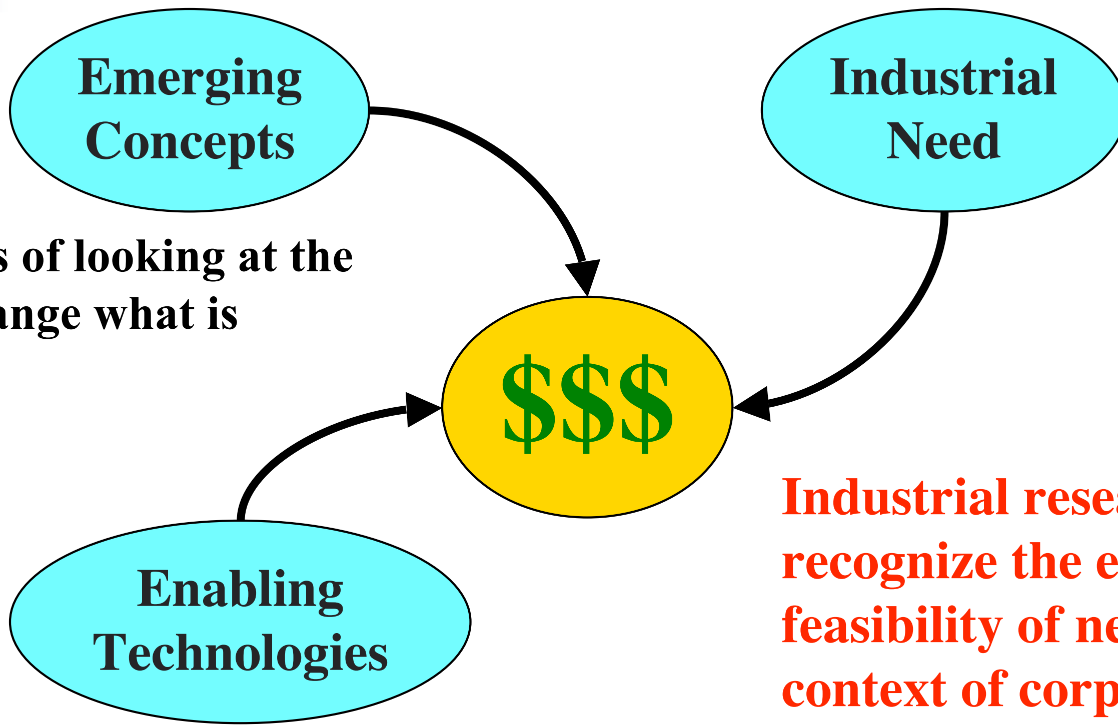
Evolved Analytics

mark@evolved-analytics.com

Agenda

- Motivation
  - Why we care about data modeling
- Technologies
  - Context: Complementary Technologies
  - Symbolic Regression Overview
- Symbolic Regression Examples
  - A Toy Problem
  - Industrial Successes
- DataModeler Package Design
  - Design Philosophy
  - Data Exploration
  - Model Development
  - Model Exploration
  - Model Management
  - Utility Functions
  - GUIKit Interface
- … plus a few diversions

Data Modeling
At the Intersection of Opportunity & Need



The economic context prioritizes the possibilities.

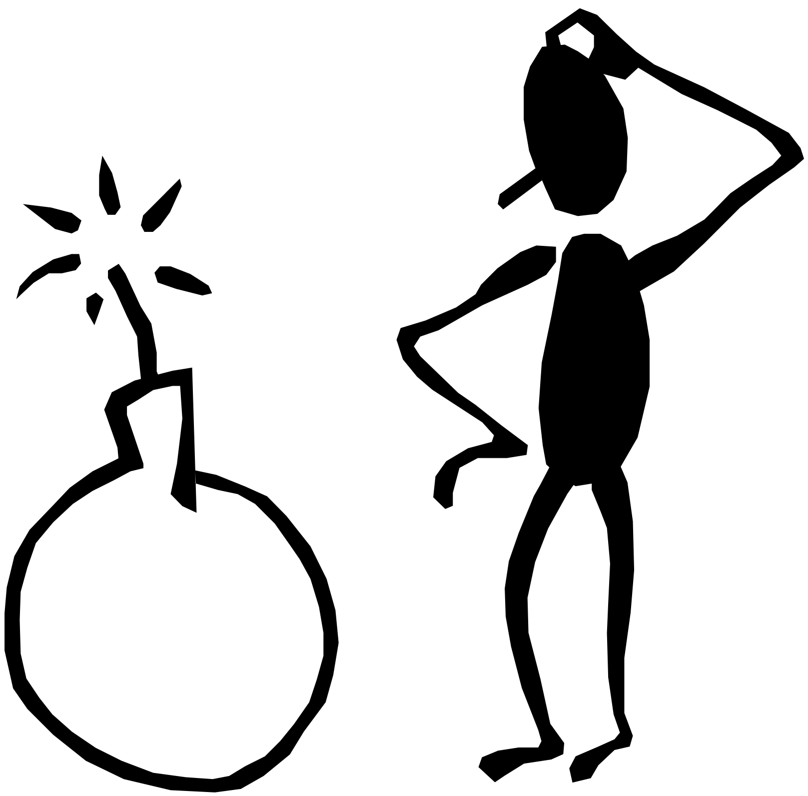
New ways of looking at the world change what is possible.

Technology & Price-Performance shifts enable implementing new concepts and implementing old concepts better.

Industrial research needs to recognize the evolving potential and feasibility of new ideas within the context of corporate needs. The DataModeler package resulted from consciously exploring this intersection.

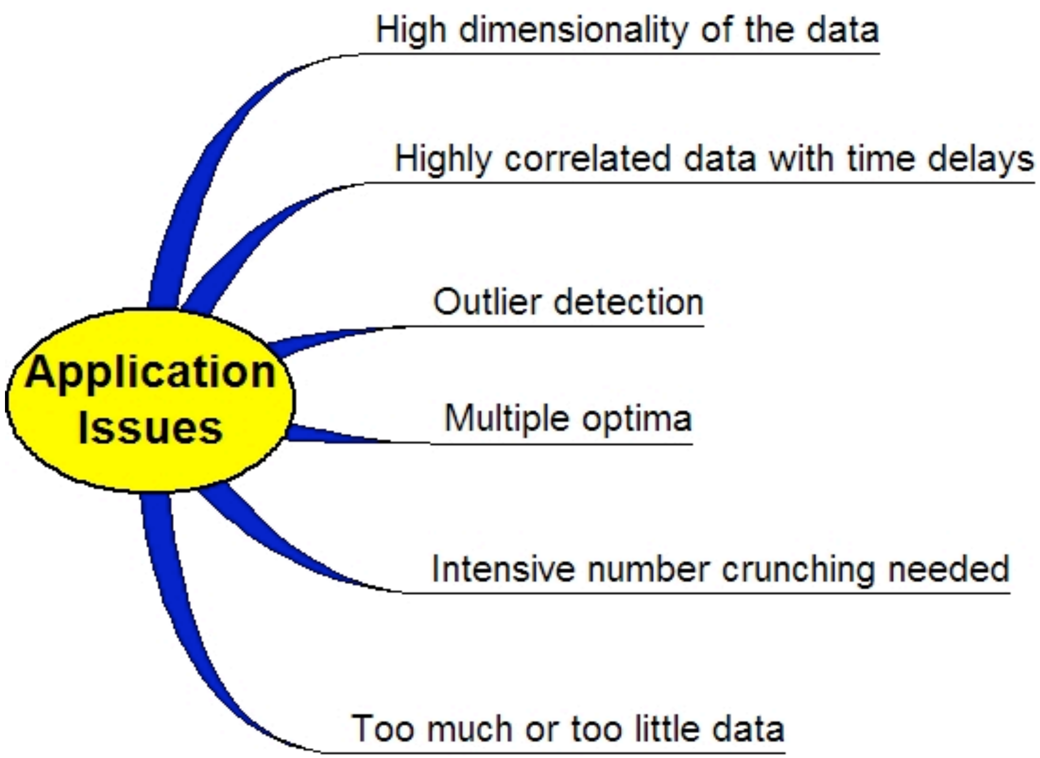
Motivation

"We are drowning in information and starving for knowledge" – R.D. Roger



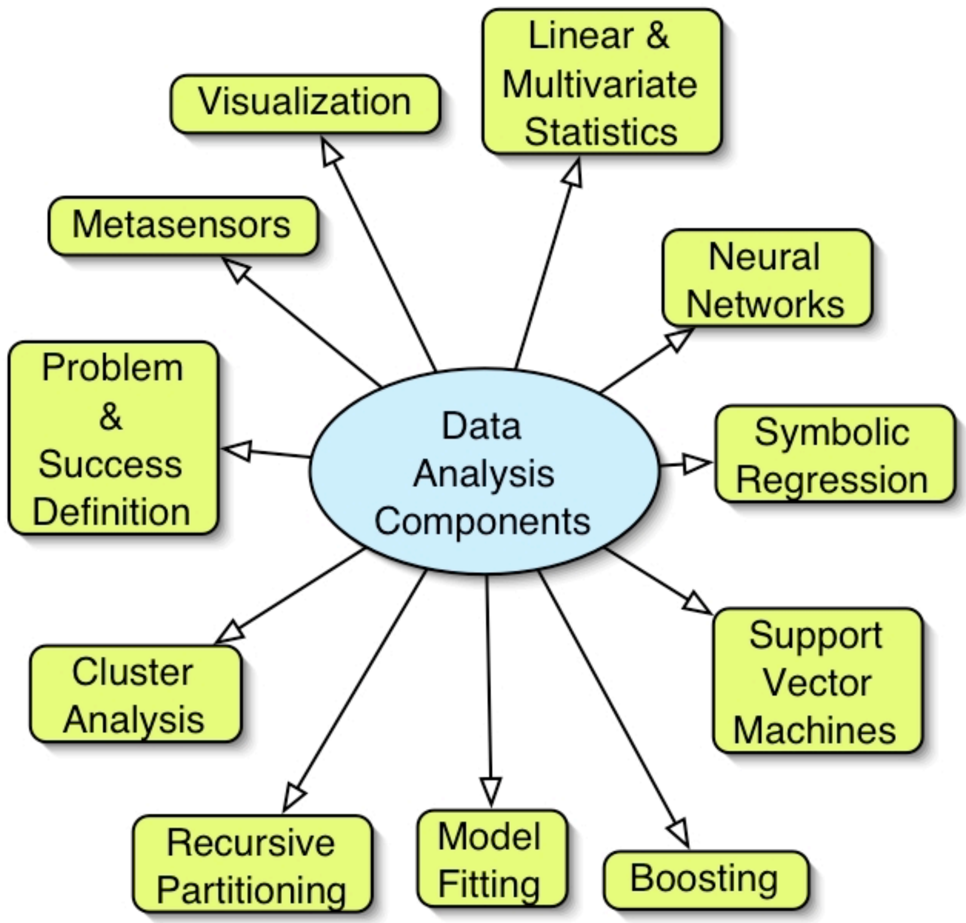
- Industry is great at collecting data … and then performing records retention
- Extracting insight from multivariate data is hard
- Time and money is being wasted

Industrial Data Modeling Issues



"The most exciting phrase to hear in science, the one that heralds new discoveries, is not 'Eureka!' (I found it!) but 'That's funny …'" — Isaac Asimov (1920 - 1992)

Empirical Modeling Context



- The role of symbolic regression is to …
  - Facilitate physical/mechanism insight and understanding
  - Summarize data behavior
  - Identify data transforms and metasensors
  - Perform variable selection
  - Enable response surface exploration and optimization
  - Visualize behavior in the form of a symbolic expression
- The overall goal is to achieve speed, accuracy & efficiency.
- Symbolic regression is part of an integrated methodology.

Competing/Complementary Technologies

Linear Models

- Linear in coefficients, not necessarily linear in model
- Often "good enough" and simple
- Well developed criteria and foundations in linear statistical analysis
- Typically easy and fast to develop (unless subtleties are involved)

Neural networks

- Often good performance but lots of “trust me”
- A good reference for nonlinear modeling potential
- The *Mathematica* Neural Networks package is ***very*** good

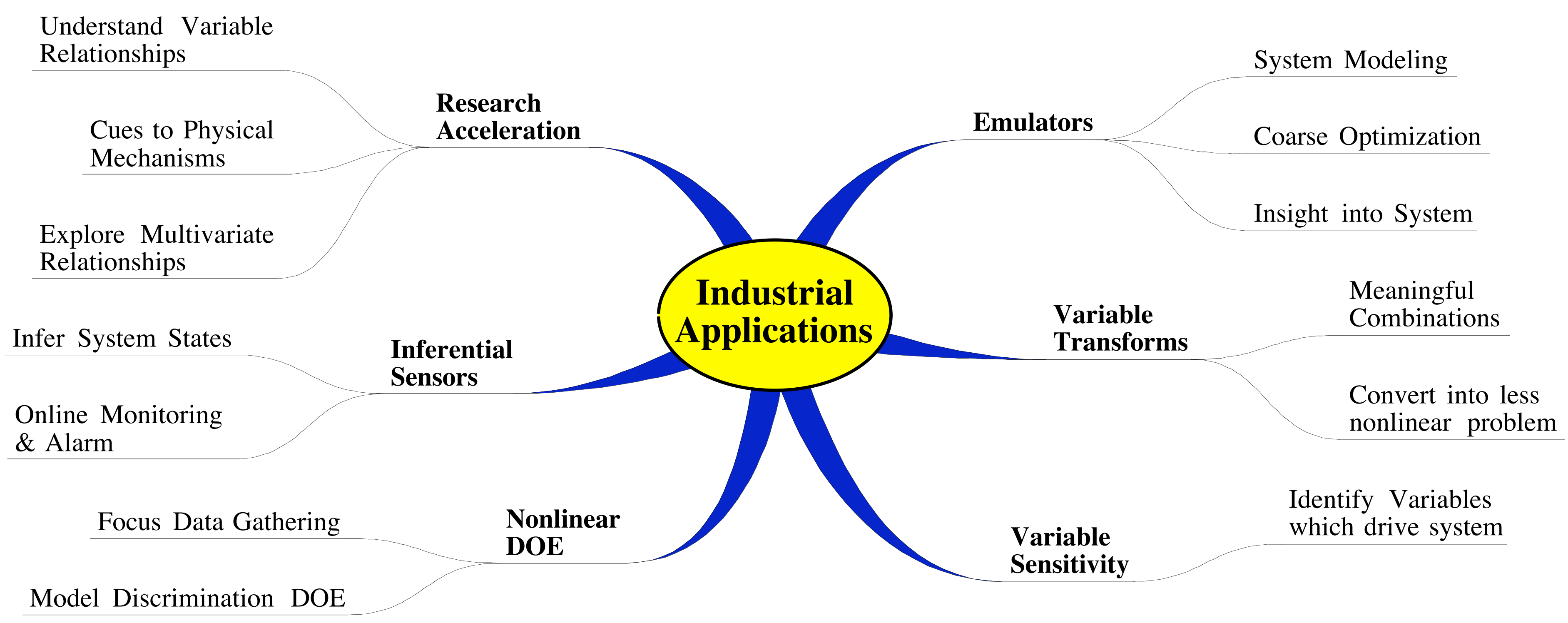
Support Vector Machines

- Useful for data compression to match information content
- Computationally demanding
- Unique nonlinear outlier detection capability

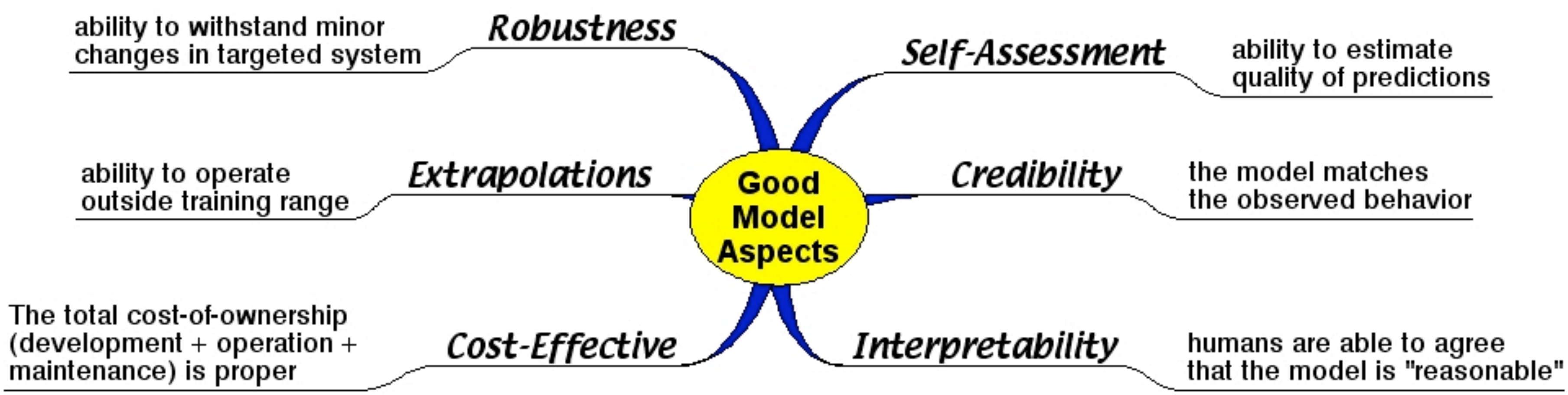
Fuzzy Rules/Recursive Partitioning

- Human interpretability — if simple
- Can handle categorical data
- The Machine Learning Framework is strong here

Data modeling impact areas

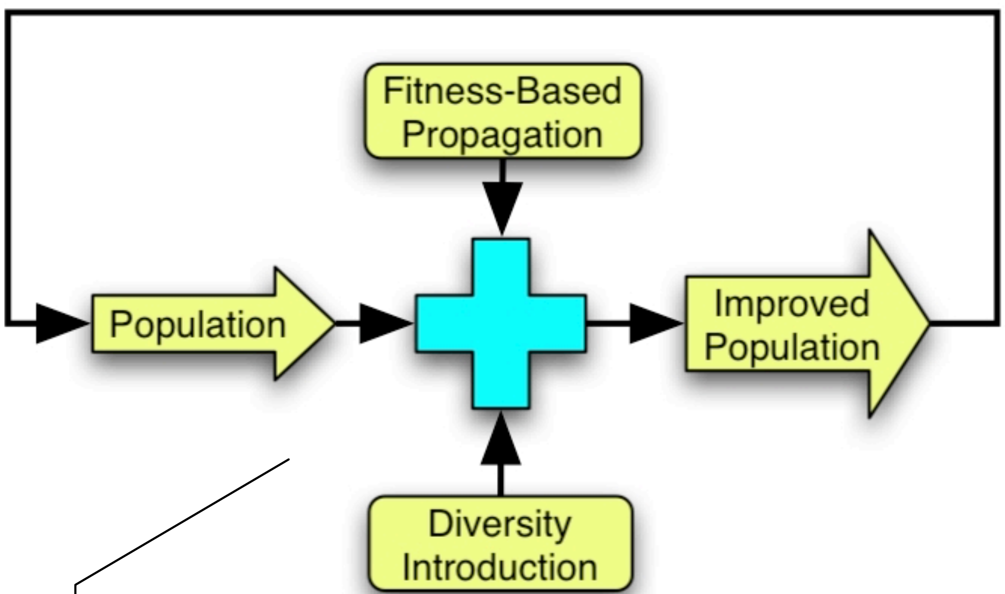


Characteristics of a Good Empirical Model



Symbolic regression has unique abilities in each of these aspects

Evolutionary Computing Theory



It is this simple!

Variants:

- Genetic Algorithms (GA)
- Evolutionary Strategies (ES)
- Evolutionary Programming (EP)
- Genetic Programming (GP)
- Particle Swarm Optimization (PSO)
- Gene Expression Programming (GEP)
- etc.

Genetic Programming

- Genome (genetic code) evolves
- Phenotype (realization) judged for fitness
- Goal is to evolve *programs* which solve problems
- The search space is *infinite*!
- Symbolic regression is one application of genetic programming

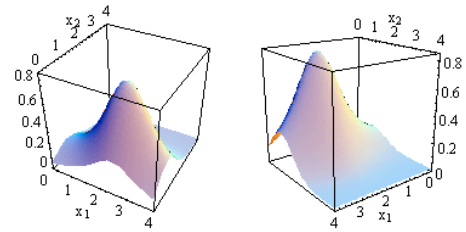
Symbolic Regression

- Goal is to identify expressions which summarize data
- NOT parameter fitting — discovery of both structure and parameters
- The search space is infinite!
- In practice, symbolic regression is part of an integrated methodology

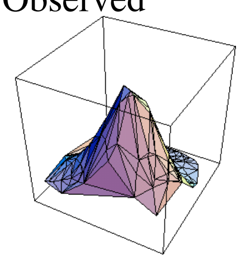
Symbolic Regression via Genetic Programming

Truth

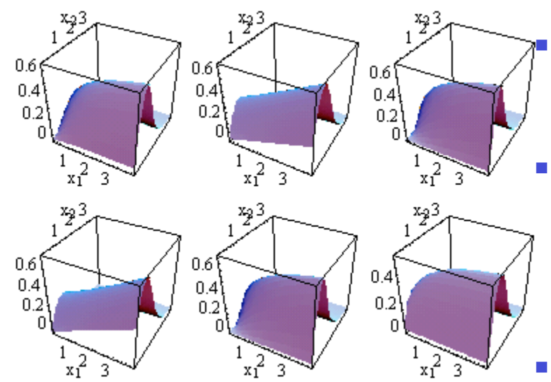
$$\frac{e^{-(-1+b)^2}}{1.2+(-2.5+a)^2}$$

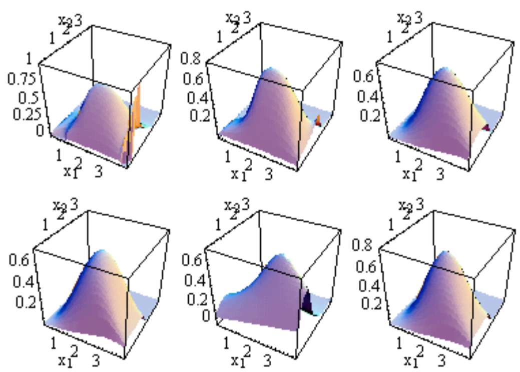


Observed



Early Results





Later Results

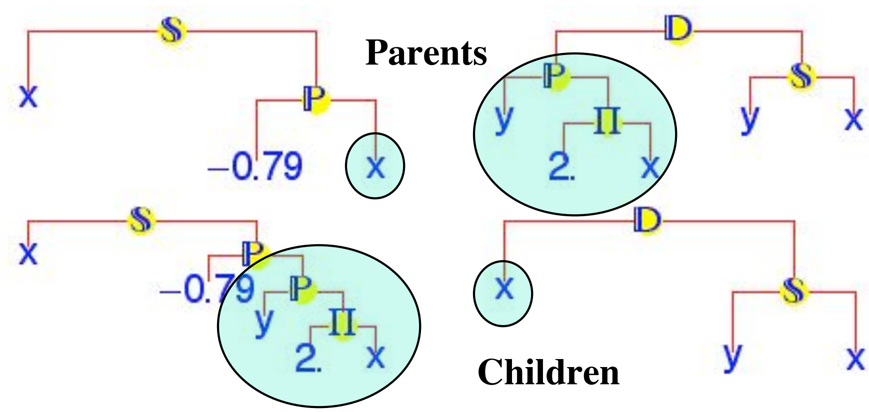
1. 0.969 0.975 x1 x2

$$\frac{x_1}{\left|\left|\frac{x_1\left|x_1\right|^{\frac{(x_1-3.665)\,x_1}{x_2}}-2.718^{x_1}+7}{x_1^2}\right|^{x_1}+\left|\left|x_2\right|^{x_2}\right|^2+x_2+\frac{x_1}{x_2}+1\right|}$$

- First, we define **building blocks**: operators, variables, and terminals (constants)
- Starting from an initial **population** of expressions (either randomly synthesized or dictated), we assign **breeding rights based upon the fitness** of the functions -- i.e., how well they match the observed behavior
- Amazingly, **expressions will evolve** which capture the behavior of the underlying data (although, not necessarily the true expression)
- Note that **multiple solutions** will evolve which are functionally similar; we can sort through the expressions to gain insight into variable relationships or forms appropriate for online implementation
- There is a **trade-off** which must be made between accuracy and simplicity (which we assume corresponds to robustness and better generalization capability)

Genetic Programming

Genome Tree Plots



Example of Crossover Operation

Phenotypes (Expressions)

Parents

$$-(-0.787701)^{x} + x \qquad \frac{y^{2}\,x}{-x+y}$$

Children

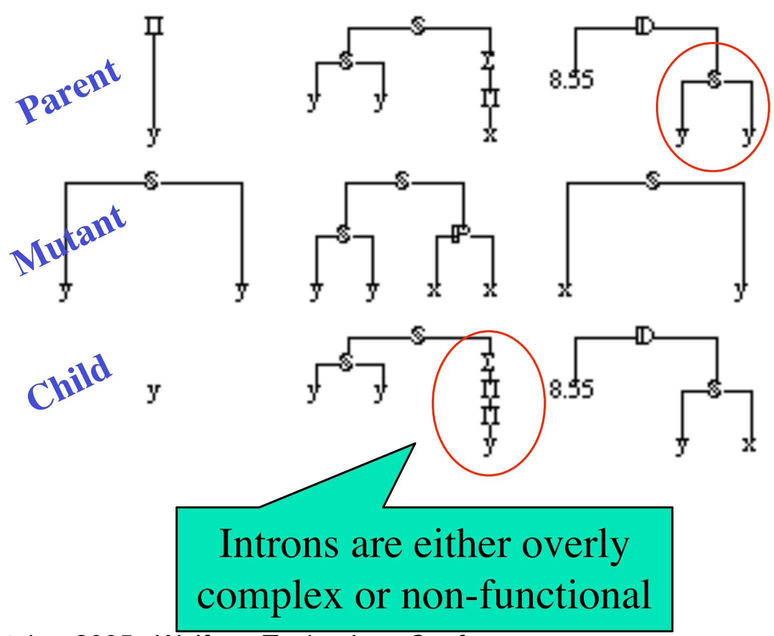
$$-(-0.787701)^{y^{2}\,x} + x \qquad \frac{x}{-x+y}$$

- Based on artificial evolution of millions of potential nonlinear functions => **survival of the fittest**
- **Many possible solutions** with different levels of complexity
- The final result is an **explicit (nonlinear) function**
- *Can* have better **generalization capabilities** than neural nets
- Low **implementation** requirements
- Issues include …
    - Time delays
    - Sensitivity analysis of large data sets
    - Relatively slow development (hours of computation time)

Symbolic Regression via GP

Nuances…

```
GenomeTreePlot[{parents,
   MutateSubtree[parents,
    MaximumTreeDepth → 3,
    MaximumArity → 2,
    DataVariables → {x, y}],
   Crossover[parents]}];
```



choice of operators
- functional building blocks

parsimony pressure
- preference for simpler/smaller solutions

diversity operators
- modify fit solutions and the relative presence of each mechanism

fitness-based breeding rights
- proportional, ranking, elitist, tournament, random, etc.

evolution environment
- population size, number of generations, population interaction, fitness criteria, etc.

genetic modifications
- coefficient & structure optimization

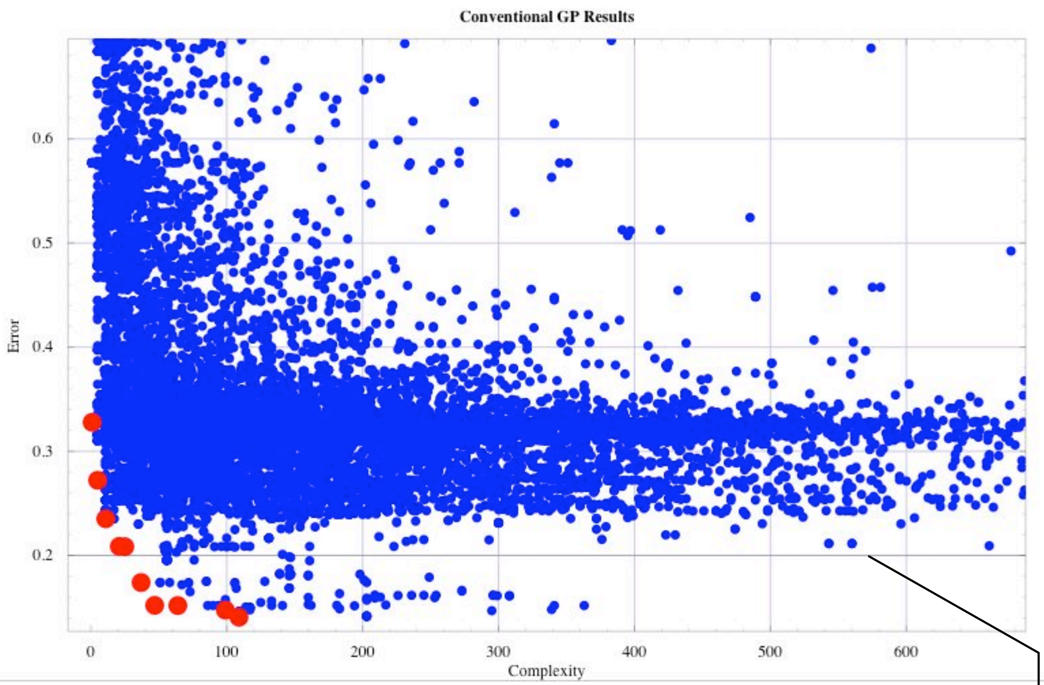
automatically defined functions
- dynamically determined building blocks

metasensor definitions
- dynamically determined transforms and variable combinations

Classic Problems with Genetic Programming

- Relatively Slow Discovery
  - Computational demands are intense
- Selection of “Quality” Solutions
  - Trade-off of Complexity vs. Performance
- Good-but-not-Great Solutions
  - Other nonlinear techniques (e.g., neural nets) outperform in raw performance
- Bloat (overly complex expressions)
  - Parsimony control requires user intervention and is problem dependent

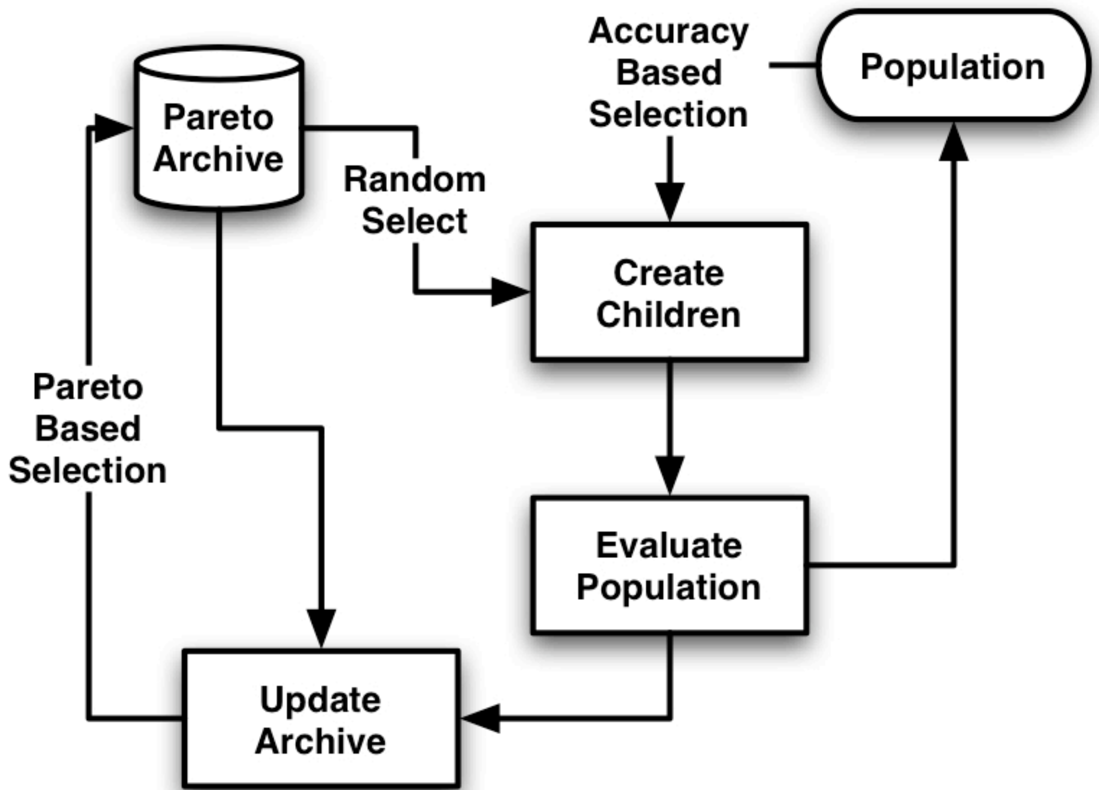
The Pareto Front



- Identifies trade-off surface between competing objectives
  - e.g., performance vs. complexity
- Pareto front solutions are the best “bang-for-the-buck”
- Introns are punished automatically
- How can we exploit?

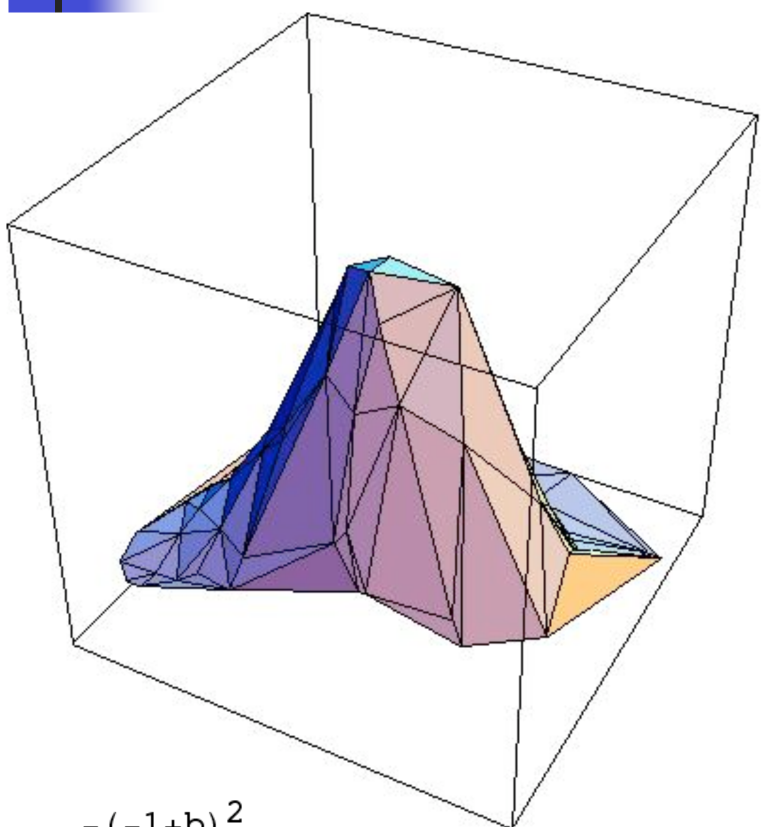
Note that much evolutionary effort is spent exploring high complexity & high fitness regions

Pareto GP Algorithm



- Select from population based upon model accuracy
- Select randomly from Pareto archive
- Cascades …
  - Pareto archive maintained
  - Population wiped out (fresh genes!)
- Independent runs with independent archives for diversity
- There are other variants along these lines

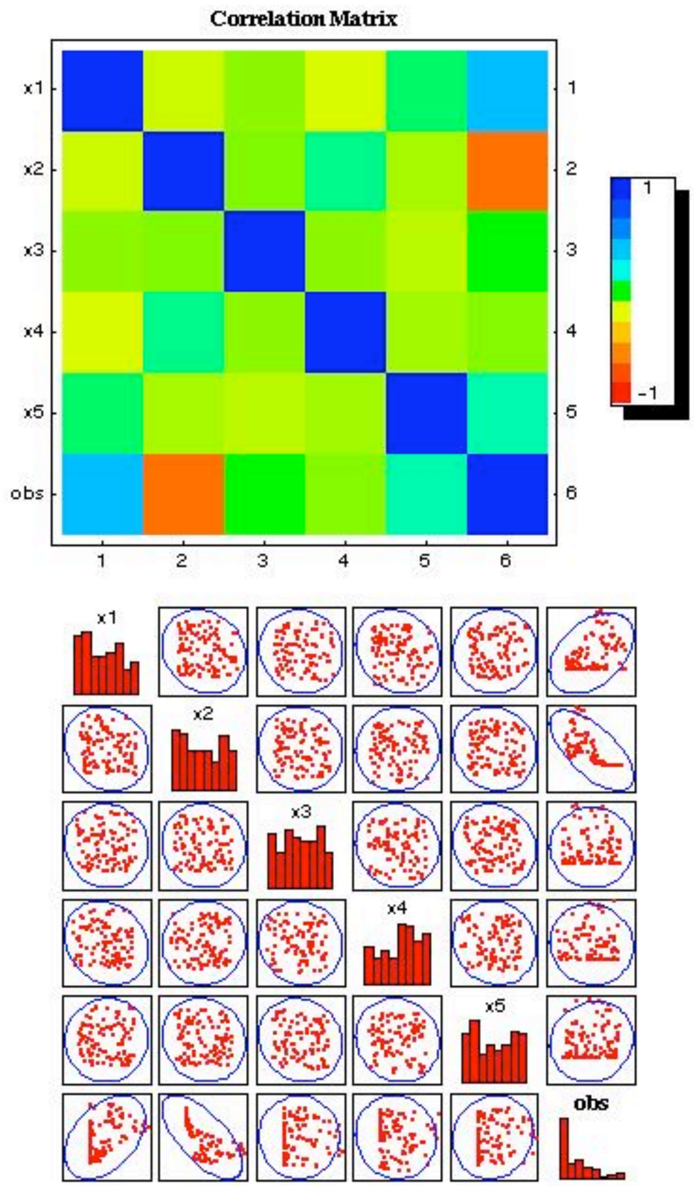
A Toy Problem for Illustration



$$\frac{e^{-(-1+b)^2}}{1.2+(-2.5+a)^2}$$

- We sampled a function of two variables at 100 random points in the range [0,4]
- The data matrix has three random spurious variables in the range [0,4]
- Notice that the entire parameter space is not covered

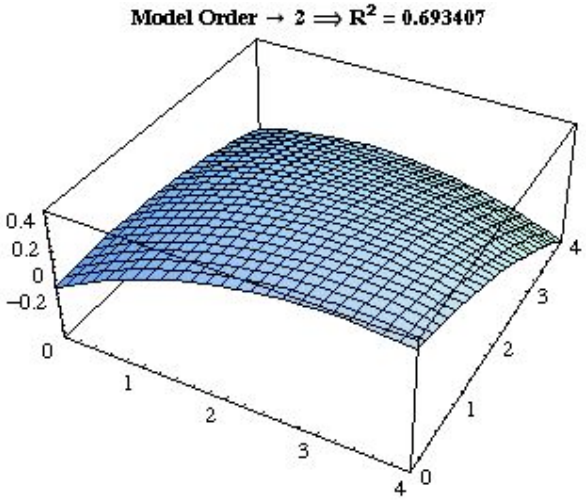
Getting the Zen of the Data

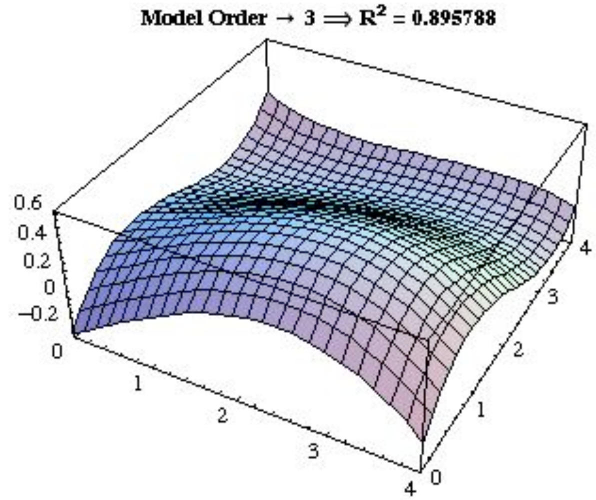


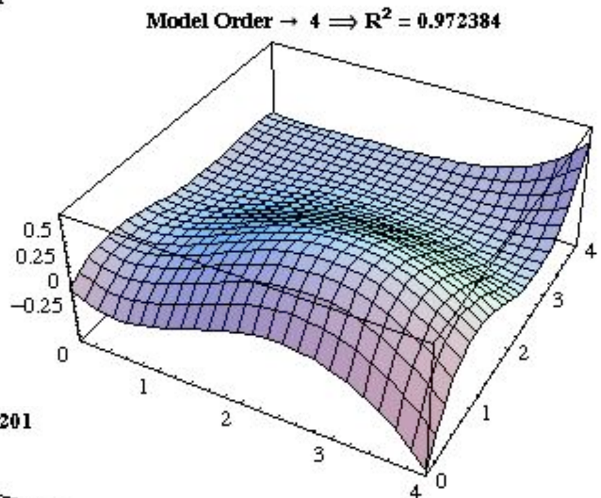
- In this simple example, we could probably guess that only two variables were important for model building
- Correlated inputs can be a problem for some other modeling techniques
- However, lack of correlation to the response does not necessarily correspond to lack of importance

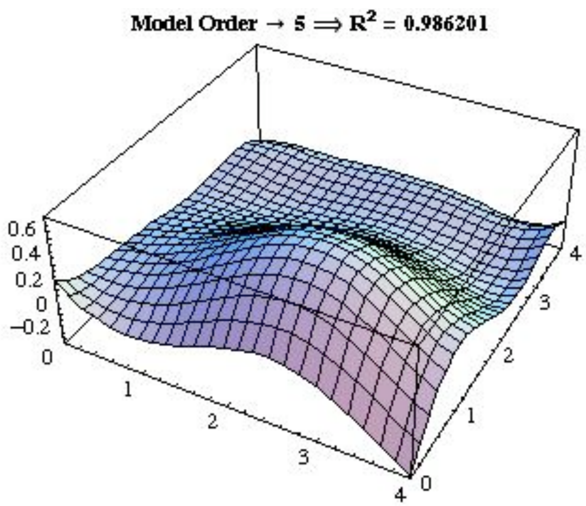
Context-free analysis leads to confidently wrong answers!

Linear Models





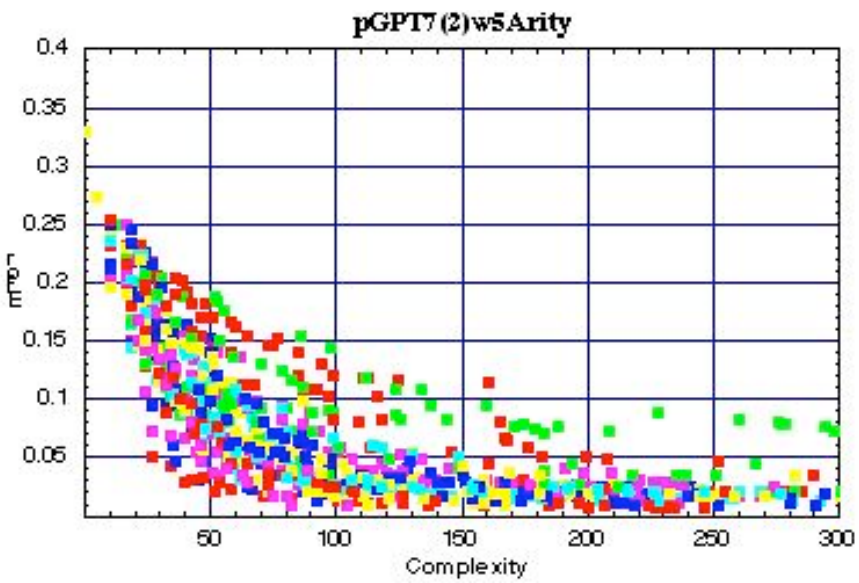




- Here we look at 2nd through 5th order models of the two driving variables (a 3rd order model with all five variables has 56 terms)
- Notice the edges -- these models would likely not extrapolate well!
- However, not much time was required to achieve a poor model!

The Pareto Front: Handling Competing Objectives

No more things should be presumed to exist than are absolutely necessary — W. Occam [1280–1349]



These are the error vs. complexity results of multiple independent symbolic regressions. Note that there is variability from run to run due to the random nature of the evolutionary process.

- Identifies trade-off surface between competing objectives
  - e.g., performance vs. complexity
- Pareto front solutions are the best “bang-for-the-buck”
- Accuracy and simplicity are automatically rewarded
- Pareto Front Benefits
  - Avoids need for *a priori* combination of objectives into a single metric
  - The shape of the front gives us insight into the problem
  - Identifies multiple candidate solutions simultaneously

Evolved Models

	model	complexity	vars	abs corr	R^2
1	x_2	1	x2	0.672146	0.45178
2	$x_1 - x_2$	5	x1 x2	0.72785	0.529765
3	$0.293716^{x_1}\, x_2^{x_1}$	11	x1 x2	0.805609	0.649005
4	$0.301214^{0.612354\, x_1}\, x_2^{0.612354\, x_1}$	17	x1 x2	0.818587	0.670084
5	$0.344236^{x_1 x_2^{[illegible]}}$	19	x1 x2	0.858864	0.737648
6	$\left(\frac{x_1}{x_2}\right)^{(5-x_1)\, x_2}$	25	x1 x2	0.895316	0.801592
7	$2.23888^{x_2}\, (-5 + x_1)\, x_1 \left(\frac{1}{x_2}\right)^{x_2}$	27	x1 x2	0.949914	0.902336
8	$2.15727^{x_2}\, (-4.89307 + x_1)\, x_1 \left(\frac{1}{x_2}\right)^{x_2}$	33	x1 x2	0.950524	0.903495
9	$1.78744^{x_2^{1.[illegible]}}\, (-5 + x_1)\, x_1 \left(\frac{1}{x_2}\right)^{x_2^{1.[illegible]}}$	35	x1 x2	0.958384	0.9185
10	$2.23888^{2 x_2}\, (-5 + x_1)^2\, x_1^2 \left(\frac{1}{x_2}\right)^{2 x_2}$	40	x1 x2	0.972973	0.946676
11	$0.415404^{-2 x_2}\, (5 - x_1)^2\, x_1^2\, x_2^{-2 x_2}$	49	x1 x2	0.976215	0.952995
12	$1.78744^{x_2^{1.[illegible]}}\, (-5 + x_1)^2\, x_1^2 \left(\frac{1}{x_2}\right)^{x_2^{1.[illegible]}}$	52	x1 x2	0.980611	0.961599
13	$(5 - x_1)^2\, x_1^2 \left(\frac{4-x_2}{x_2}\right)^{x_2}$	54	x1 x2	0.980731	0.961833
14	$\frac{2^{[illegible]} \left(\frac{1}{x_2}\right)^{[illegible]}}{3.51424+(1.96032-x_1)^2 - x_1}$	62	x1 x2	0.985068	0.970359
15	$(9.56047 + (9 - 2 x_1)^{x_1})^{x_2^{[illegible]}}$	65	x1 x2	0.989806	0.979716
16	$\frac{1.82619^{[illegible]} \left(\frac{1}{x_2}\right)^{[illegible]}}{3.51424+(1.96032-x_1)^2 - x_1}$	72	x1 x2	0.994007	0.988051
17	$(9 + (9 - 2 x_1)^{x_1} + x_1)^{x_2^{[illegible]}}$	73	x1 x2	0.99426	0.988554
18	$(8.18505 + (9 - 2 x_1)^{x_1} + x_1)^{x_2^{[illegible]}}$	78	x1 x2	0.994281	0.988596
19	$(9 + (9 - 2 x_1)^{x_1} + 2 x_1)^{x_2^{[illegible]}}$	83	x1 x2	0.994852	0.98973
20	$(9 + (9 - 2 x_1)^{x_1} + 2 x_1 - x_2)^{x_2^{[illegible]}}$	95	x1 x2	0.995965	0.991947
21	$(8.18505 + (9 - 2 x_1)^{x_1} + 2 x_1 - x_2)^{x_2^{[illegible]}}$	100	x1 x2	0.99604	0.992095

2nd

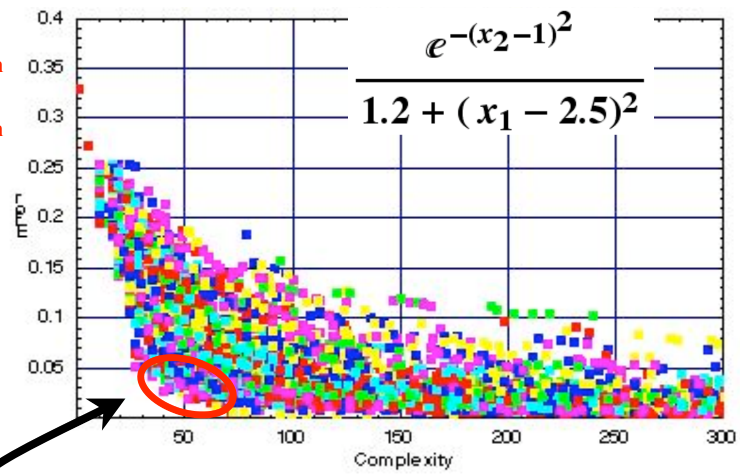
3rd

Equivalent linear model

4th

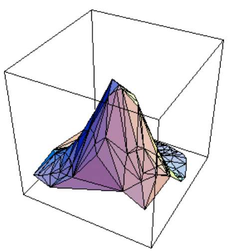
5th

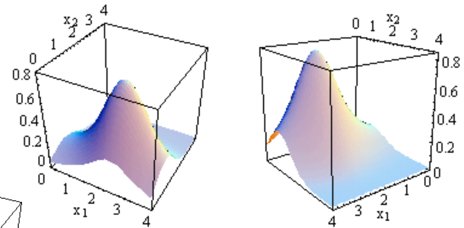
- A run tends to fully explore a foundation structure
- Independent evolutions will result in different (but still fit) structures
- Cascading results from independent evolutions seems to be beneficial
- Note that we are not strictly restricted to the Pareto front in selecting models -- many models may be “good enough” and have the benefit of being structurally different and diverse



similar performance but diverse structure

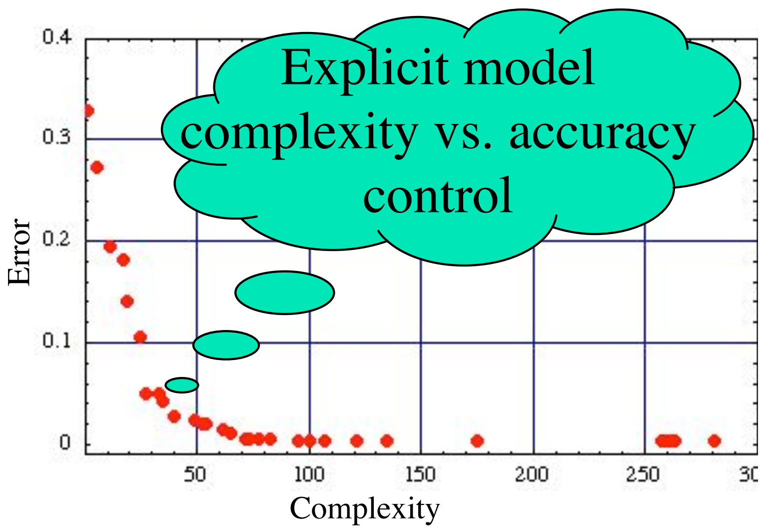
Pareto Front Models

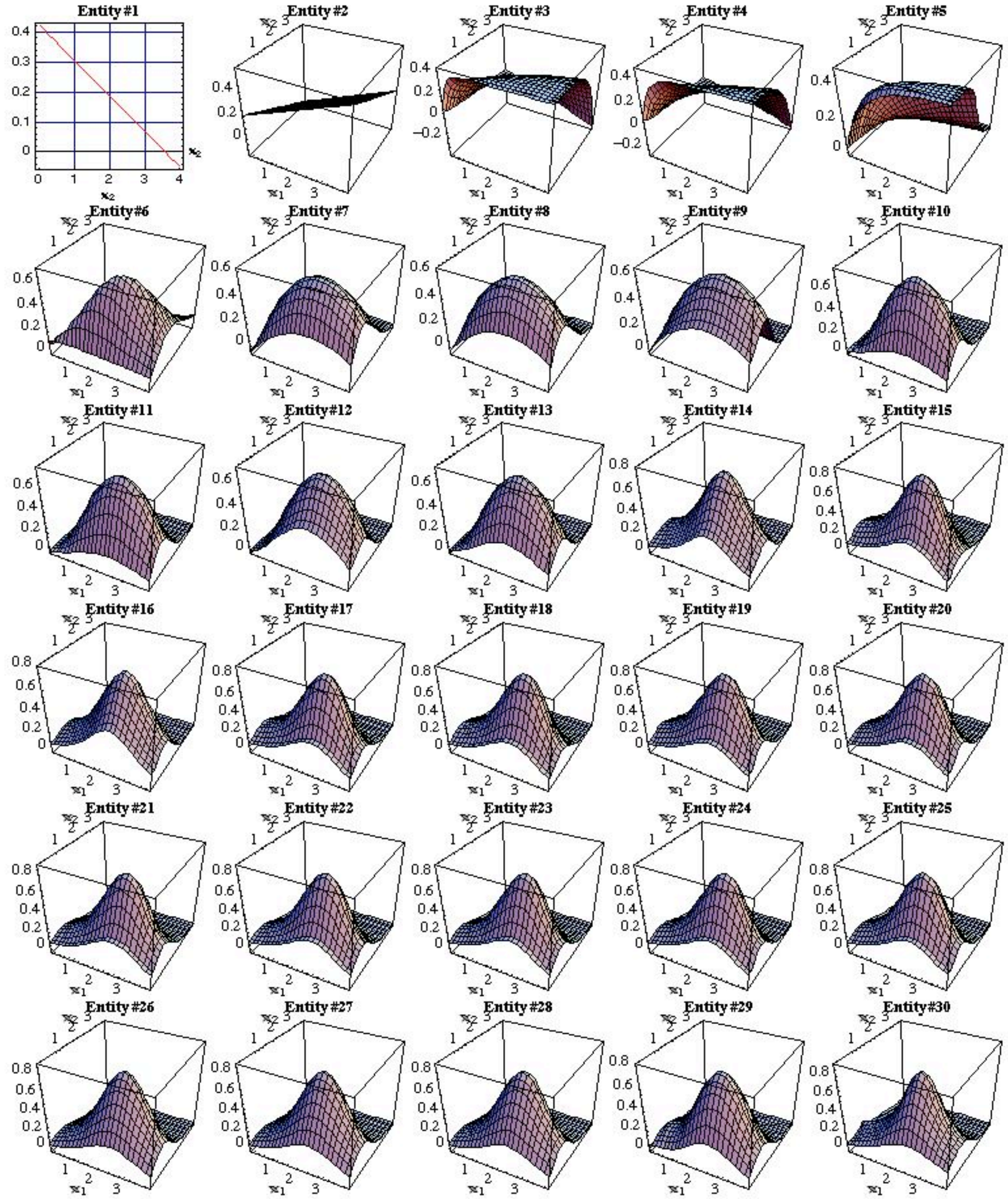




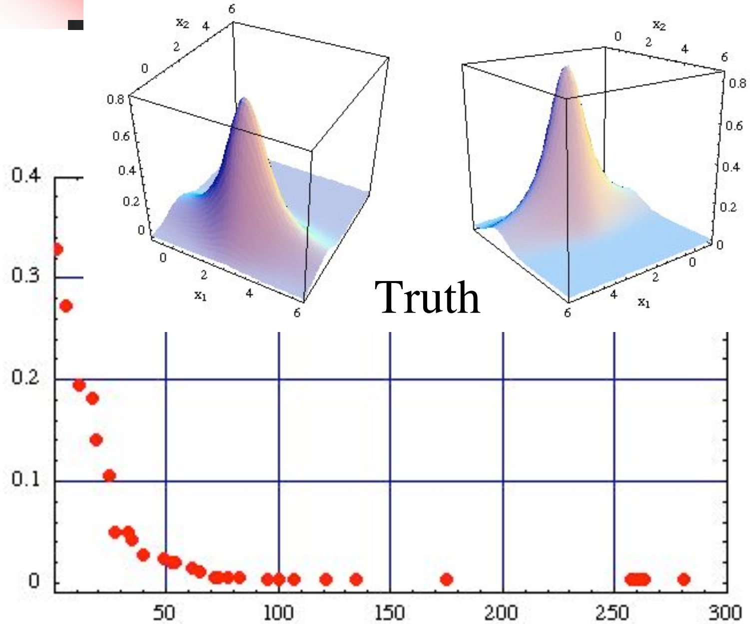
Truth

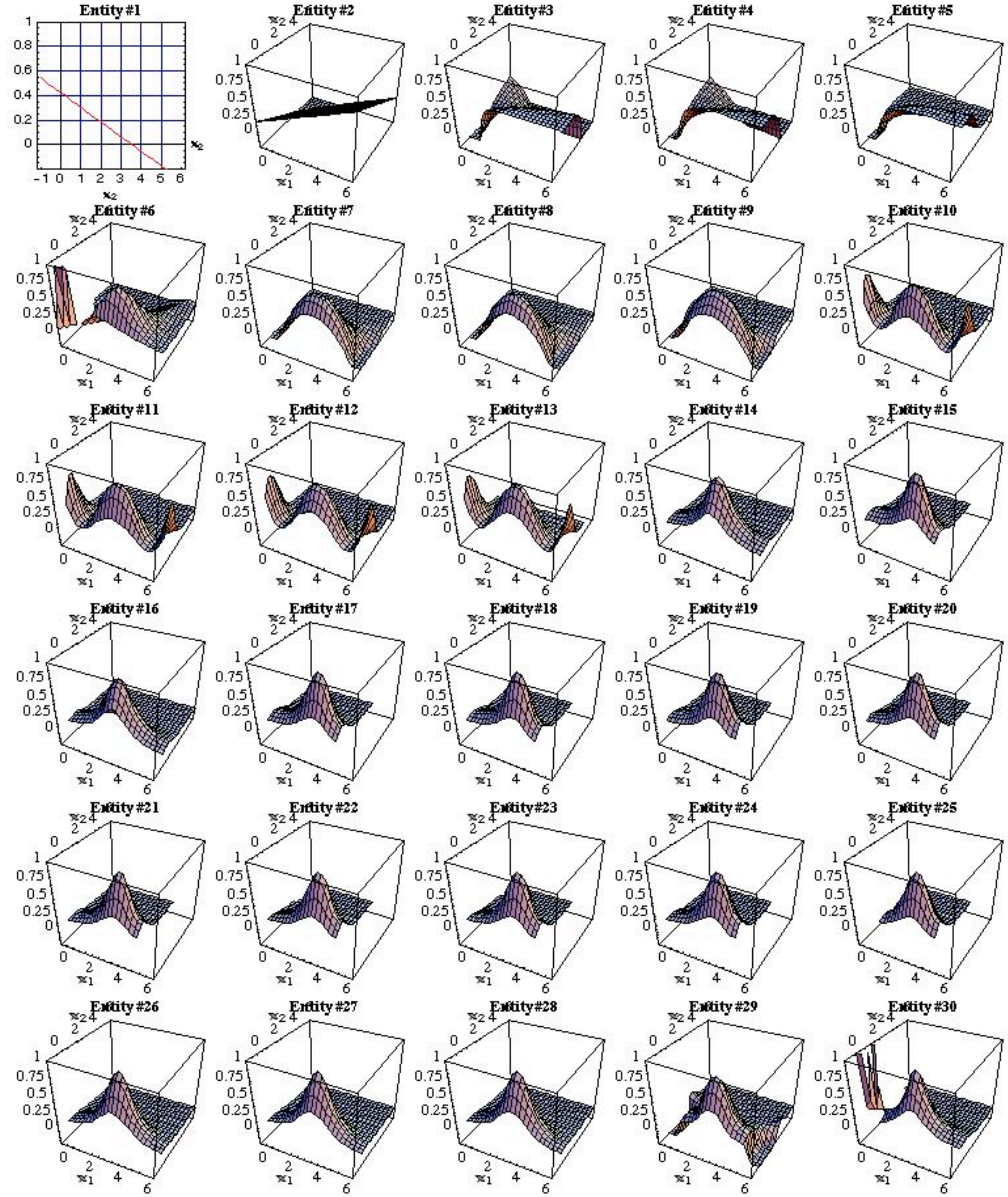
$$\frac{e^{-(x_2-1)^2}}{1.2 + (x_1 - 2.5)^2}$$





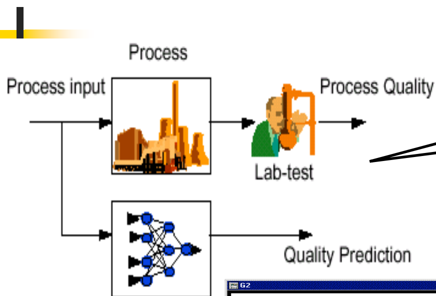
Parsimony & Extrapolation

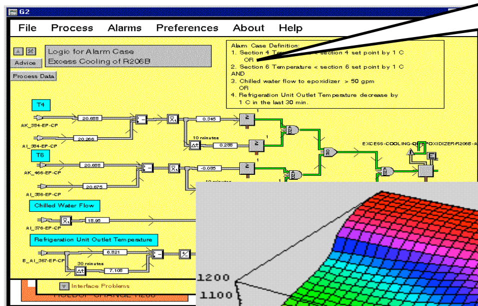


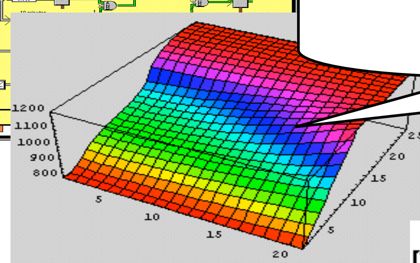


- Note the pathologies at high complexity when extrapolating
- In general, we want to avoid overmodeling!

Key application areas



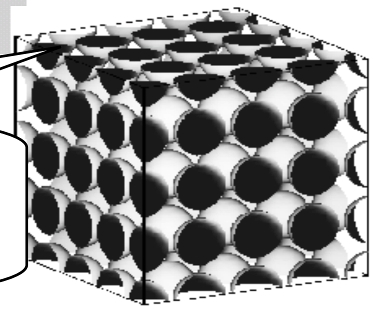




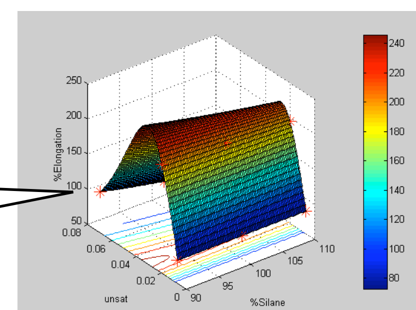
Robust Inferential Sensors
Mass-scale on-line empirical models

Automated Operating Discipline
Consistent intelligent on-line supervision

Empirical Emulators of Fundamental Models
Effective on-line process optimization



Fundamental model building based on GP
Accelerated new product development



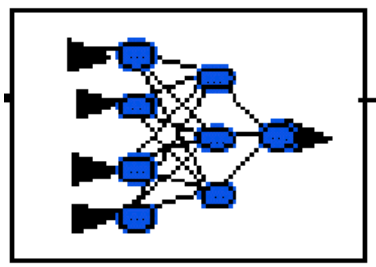
Nonlinear DOE based on GP
Minimizing expensive process experiments

Neural Net Issues

Operational Issues

- High sensitivity to **process changes**
- Frequent **re-training**
- Complicated development & **maintenance**
- Low **survival rate** after 3 years in operation
- **Engineers** hate black-boxes

Black box



VS.

$$\mathrm{Func2} = \frac{\mathrm{rate}^2 \left(\mathrm{vac} + \frac{\mathrm{rate} * \mathrm{hopp\ wt}}{\mathrm{temp}}\right) \mathrm{pllt\ wt\ temp}}{\mathrm{density} * \mathrm{temp}^2}$$

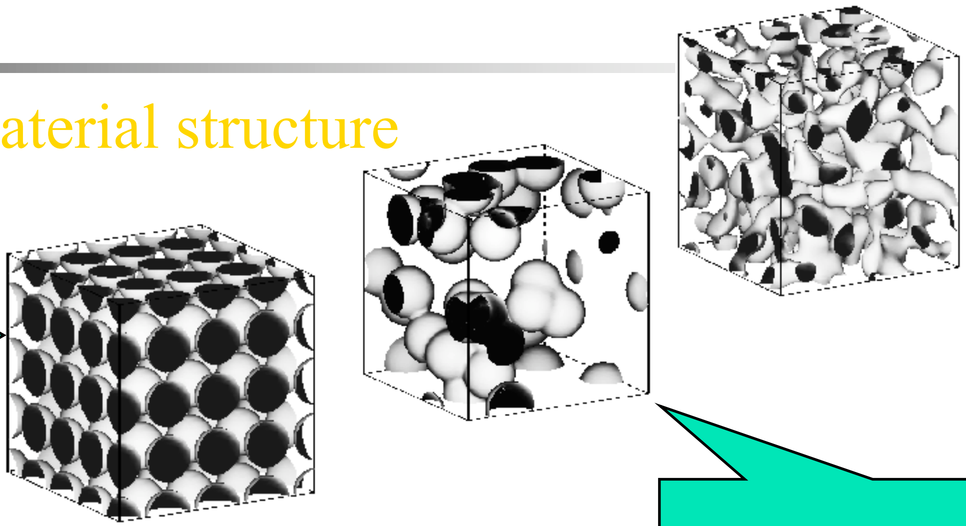
Analytical expression

Specialized run-time software

Directly coded into most on-line systems

The problem of structure-properties in fundamental modeling

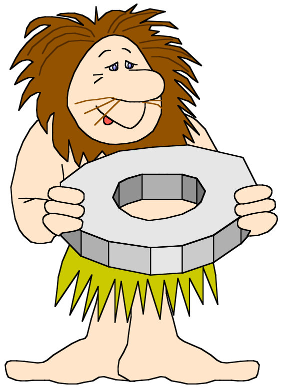
Material structure



Properties:
- molecular weight
- particle size
- crystallinity
- volume fraction
- material morphology
- etc.

Key modeling effort for new product development

Modeling issues:
- nonlinear interaction
- large number of preliminary expensive experiments required
- large number of possible mechanisms
- slow fundamental model building
- insufficient data for training neural nets



Results from hypothesis search

Selected symbolic regression empirical model

Fundamental model

$$\boxed{y = a + [b\,x_1 + c\log(x_2)]\,e^{kx_3} + d\,x_5}$$

Selected empirical model

GP-generated empirical model correctly captured the functional forms of the fundamental model

Square root form for x1

$$y = a + b\left(\frac{\sqrt{e^{\frac{-x_3}{\log(x_1 x_5^2)}}}}{e^{-x_3} + \log(x_2)} + \sqrt{x_1} + x_5\right)$$

Linear form for x5

Exponential form for x3

Logarithmic form for x2

GP and Design Of Experiments (DOE) Models Showing Lack of Fit

Situations of Lack of Fit

1. Simple factorial DOE

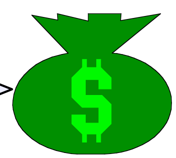
Enough experiments to fit first order model

$$y = \hat{a}_o + \sum_{i=1}^{k} \hat{a}_i x_i + \sum\sum_{i<j} \hat{a}_{ij} x_i x_j$$

Classical approach if LOF add experiments to fit second order model

$$S_k = \beta_o + \sum_{i=1}^{k} \beta_i x_i + \sum \beta_{ii} x_i^2 + \sum\sum_{i<j} \beta_{ij} x_i x_j$$

More costly experiments



2. A response surface DOE

already had all experiments to fit second order model

$$S_k = \beta_o + \sum_{i=1}^{k} \beta_i x_i + \sum \beta_{ii} x_i^2 + \sum\sum_{i<j} \beta_{ij} x_i x_j$$

Classical approach if LOF no alternative (use model as it is)

Suggested approach: Use GP to transform inputs

1. Generate GP models

$$S_k = \frac{3.13868\times10^{-17} e^{\sqrt{2x_1}} \ln\left[(x_3)^2\right] x_2}{x_4} + 1.00545 \quad (2)$$

Selected solution

2. Generate input transforms

Variable transformations suggested by GP model

Original Variable	Transformed Variable
x_1	$Z_1 = \exp\left(\sqrt{2x_1}\right)$
x_2	$Z_2 = x_2$
x_3	$Z_3 = \ln[(x_3)^2]$
x_4	$Z_4 = x_4^{-1}$

3. Fit response surface model in transformed variables

$$S_k = \beta_o + \sum_{i=1}^{4} \beta_i Z_i + \sum\sum_{i<j} \beta_{ij} Z_i Z_j + \sum_{i=1}^{4} \beta_{ii} Z_i^2$$

Source	DF	Sum of Square	Mean Square	F Ratio
Lack of Fit	2	0.00049190	0.000246	2.2554
Pure Error	2	0.00021810	0.000109	**Prob > F**
Total Error	2	0.00071000		0.3072
				Max RSq
				0.9999

No Lack Of Fit
(p=0.3037)

Symbolic Regression: Summary Benefits

Compact Nonlinear Models

- Compact empirical models can be suitable for **online implementation**
- Model(s) can be used as an **emulator** for coarse system optimization

Driving Variable Selection & Identification

- Appropriate models may be developed from **poorly structured data sets** (too many variables & not enough measurements)
- Identified driving variables may be used as **inputs into other modeling tools**

Metasensor (Variable Transform) Identification

- Identifying **variable couplings** can give insight into underlying physical mechanisms
- Identified metavariables can enable **linearizing transforms** to meld symbolic regression and more traditional statistical analysis
- Metavariables can also be used as **inputs into other modeling tools**

Diverse Model Ensembles

- The independent evolutions will produce **independent models**. Independent (but comparable) models may be stacked into ensembles whose divergence in prediction may be an indicator of extrapolation & model **trustworthiness**. This is an issue in high dimensional parameter spaces.

Human Insight

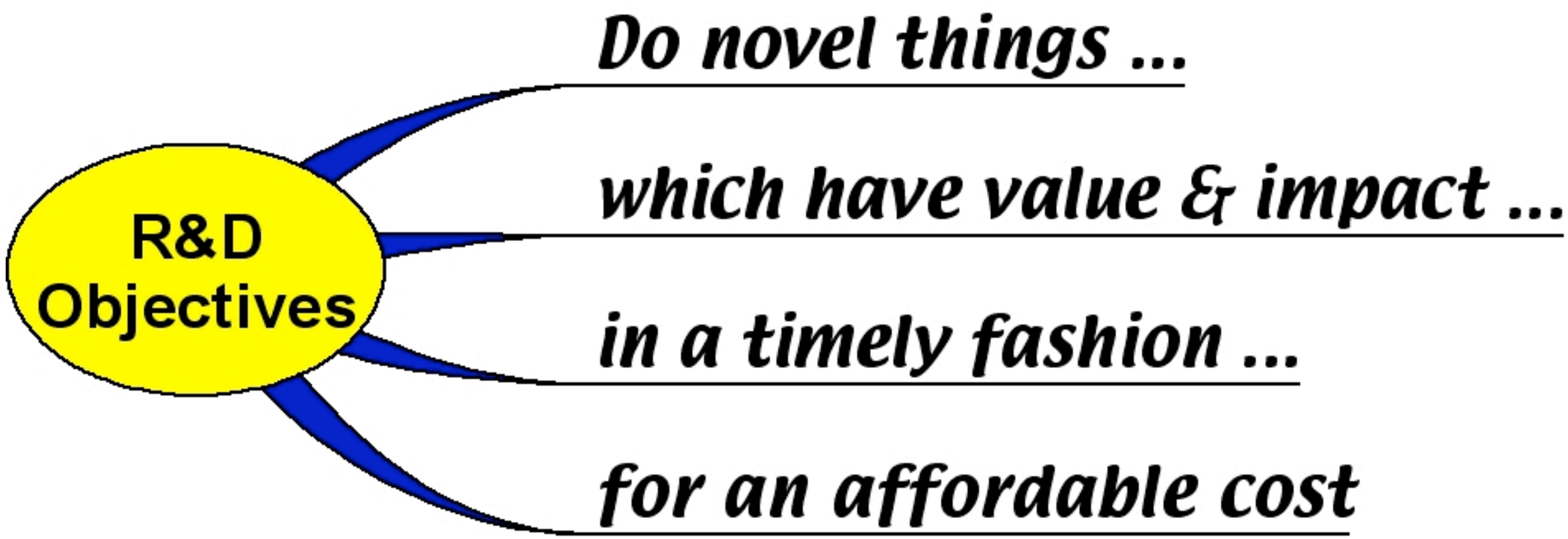
- The **transparency** of the evolved models as well as the explicit identification of the model **complexity-accuracy trade-off** is very compelling
- Examining an expression can be viewed as a **visualization** technique for high-dimensional data

There are many benefits to symbolic regression. These are enhanced when coupled with other analysis tools and techniques.

Mathematica Implementation

(finally … but first a diversion into system building …)

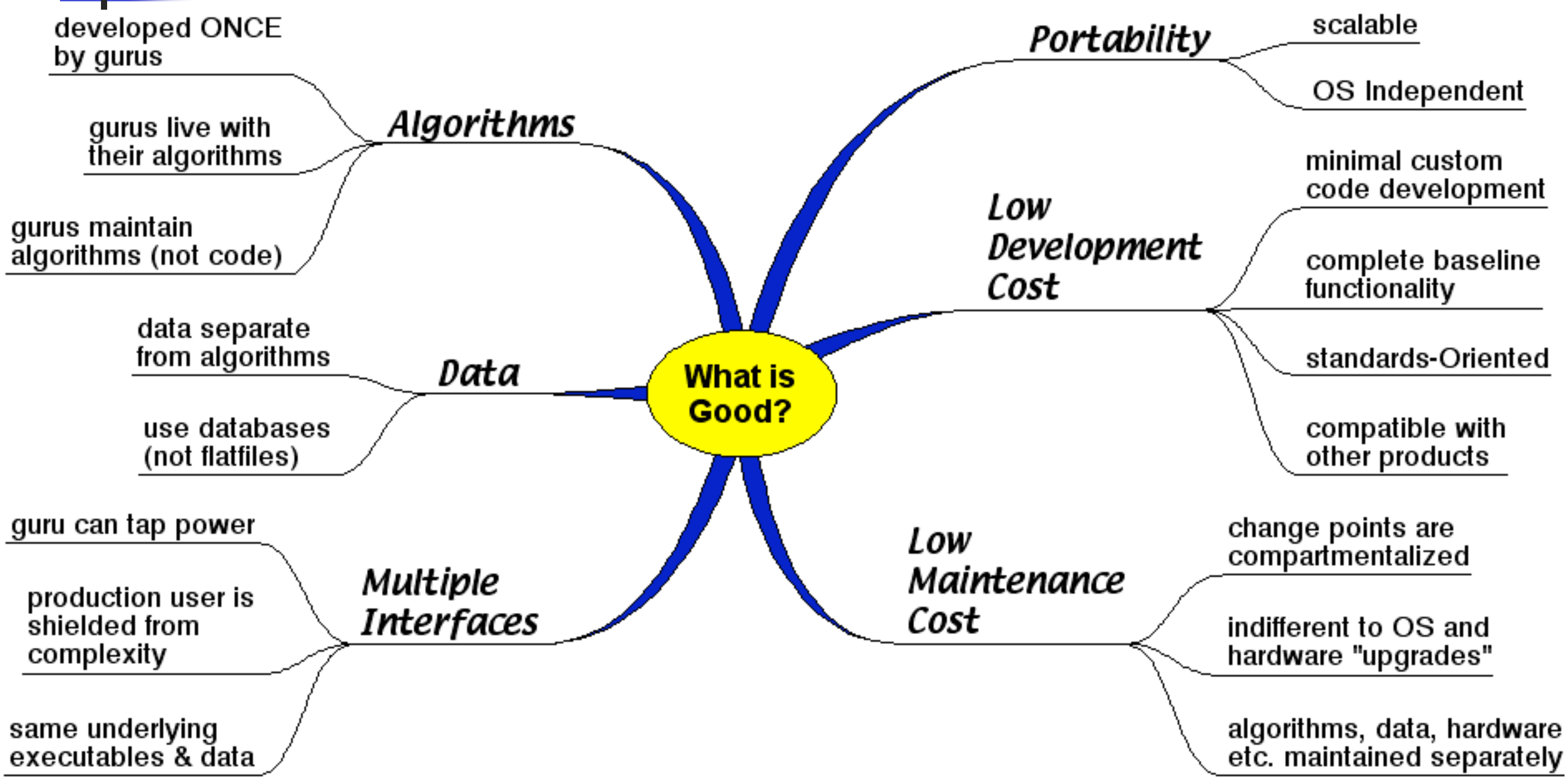
Corporate Research Objectives



The Researcher Dilemma

- The Problem
  - We want to learn and do new things -- a.k.a., “research”!
  - If we develop & build many useful solutions …
    - We are rewarded; however,
    - We eventually devote all our time to maintaining those solutions
    - This limits our ability to do new things which will lead to more rewards
  - Hence, success tends to be self-limiting!
- How do we resolve the researcher dilemma?

Characteristics of a “Good” Analysis System



Mathematica in the Analysis System Context

- Algorithm/Interface Partitioning
  - Developed package can be exploited in via *Mathematica* notebook, *webMathematica*, automated script, generated reports, GUI, etc.
  - Algorithms can be maintained in one place once by the guru
- Baseline Functionality
  - Many built-in functions + commercial packages
  - Tools for a variety of user interfaces
  - Supported on variety of compute platforms
- Flexibility
  - Totally scriptable operations
  - Extensible
  - Multiple programming paradigms

Packages & End-User Development

A Foundation For Capturing Value

(another diversion)

Why Packages?

- Important for analysis system development
- Benefits
  - Capture knowledge & expertise
  - Makes the experience transfer easy
  - Good even for the individual user
  - Documentation & usage examples
- Rant
  - Package development should be vigorously supported and encouraged by Wolfram Research
  - Students should be writing packages in *Mathematica* not toolboxes for MATLAB!!

The Package Development Process

- Develop algorithms in a notebook
- Transfer algorithms into a package context
  - Gotchas: contexts & hidden inclusions
  - Avoid stomping on other packages and definitions
- Write the help browser documentation
  - Help browser documentation is *easy* using **AuthorTools** (albeit, not well documented)
  - Ignore the *Mathematica Journal* article -- it is not that hard!
  - It isn't a package without the help browser!

The *Entire* Help Browser Build Process

- **Write the help in a notebook** using the HelpBrowser style
  - Use Section/Subsection/Subsubsection to define browser hierarchy
  - Use SubsectionIcon/SubsubsectionIcon for non-browser hierarchy
  - Content only at the bottom of the browser hierarchy tree (a.k.a., "strict outline form")
- Use *AuthorTools* : *MakeIndex* : ***Edit Notebook Index*** palette to tag cells with terms/phrases/functions/etc. which should be searchable in the browser
- Save the help notebook into the *packageName*/Documentation/English directory (a.k.a., the "help directory")
- Use *AuthorTools*: *Make Categories* : ***Make BrowserCategories*** palette to create a BrowserCategories.m file in the help directory
- Use *AuthorTools*: *Make Index* : ***Make Browser Index*** palette to create a BrowserIndex.nb file in the help directory
- Choose "Rebuild the Help Index" from the main menu
- You can use the *AuthorTools* : *Make Project* if you want to integrate multiple help documentation files

Summary:
After writing the help, we only need clicks on the AuthorTools palette buttons (in the right order) and some file renaming

Building the help for an end-user package is THIS simple!!

DataModeler System Design

(really!)

Design Philosophy

Life should be easy for the user

Design Philosophy Implementation

- Complete Tool Suite: tools for …
  - Data exploration
  - Model development (multiple methods)
  - Model validation and exploration
  - Model management (archival and retrieval)
  - Analysis documentation
- Make the Modeling Easy
  - Standard *Mathematica* function interface for the power user
  - GUIKit interface for the novice (and the lazy/smart power user)
  - Lots of help browser documentation

Package Functions Include…

Utility Functions

SyncFunctionOptions, LabelForm, GridTable, **EvaluationNotebookDirectory**, FileNamesOnly, ArchiveImage, AbsoluteCorrelation, SummaryStatistics, NumericCompile, MapThreadUnbalanced, PolynomialBasisSet, AutoSymbolList, **ParetoFront**, ParetoLayers

Data Exploration

ConfidenceEllipsoid, ConfidenceEllipsoidSelection, ConfidenceEllipsoidSelectionIndices, RobustCorrelationMatrix, **CorrelationMatrixPlot**, **ScatterPlotMatrix**

Model Development & Engineering

SymbolicRegression, RandomEntities, CreateEntityFromGenome, **CreateEntityFromExpression**, ExtractGenomeSubtrees, GenomeExpressions, SimplifyGenome, **SimplifyEntity**, ReplaceGenome, RemoveIntrons, EvaluateGenome, SelectEntity, MutateSubtree, Clone, Crossover, **AlignEntity**, OptimizeEntity,

Model Review

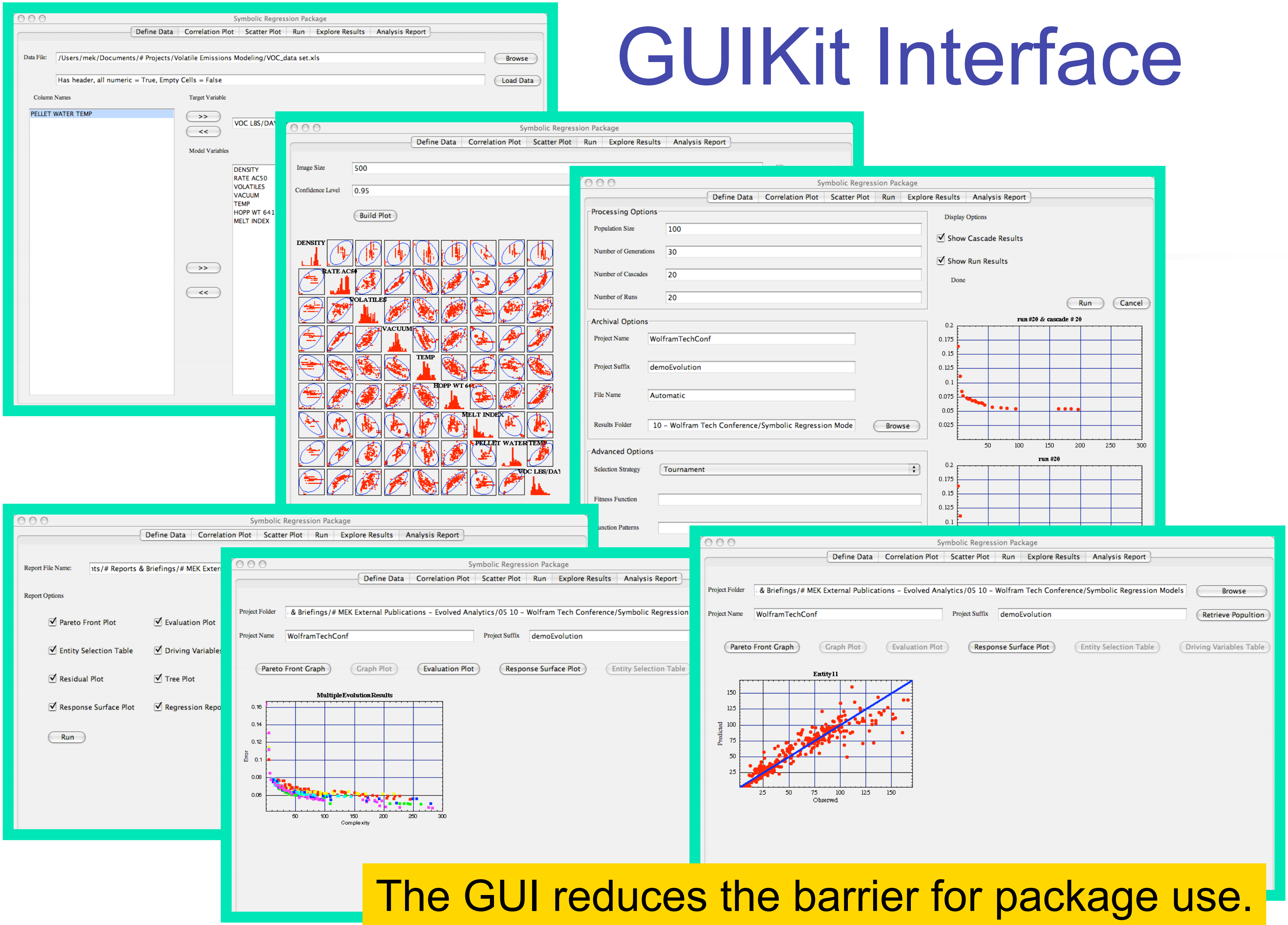
EvaluateModel, ExpressionGraphPlot, ExpressionTreePlot GenomeTreePlot, ModelEvaluationPlot, ModelResidualPlot, **ModelRegressionReport**, ParetoFrontPlot, **ResponseSurfacePlot**, **EntitySelectionTable**, **VariablePresence**

Model Management

StoreModelSets, RetrieveModelSets, MergeModelSets

In practice, only the SymbolicRegression function along with model review & management functions are generally used

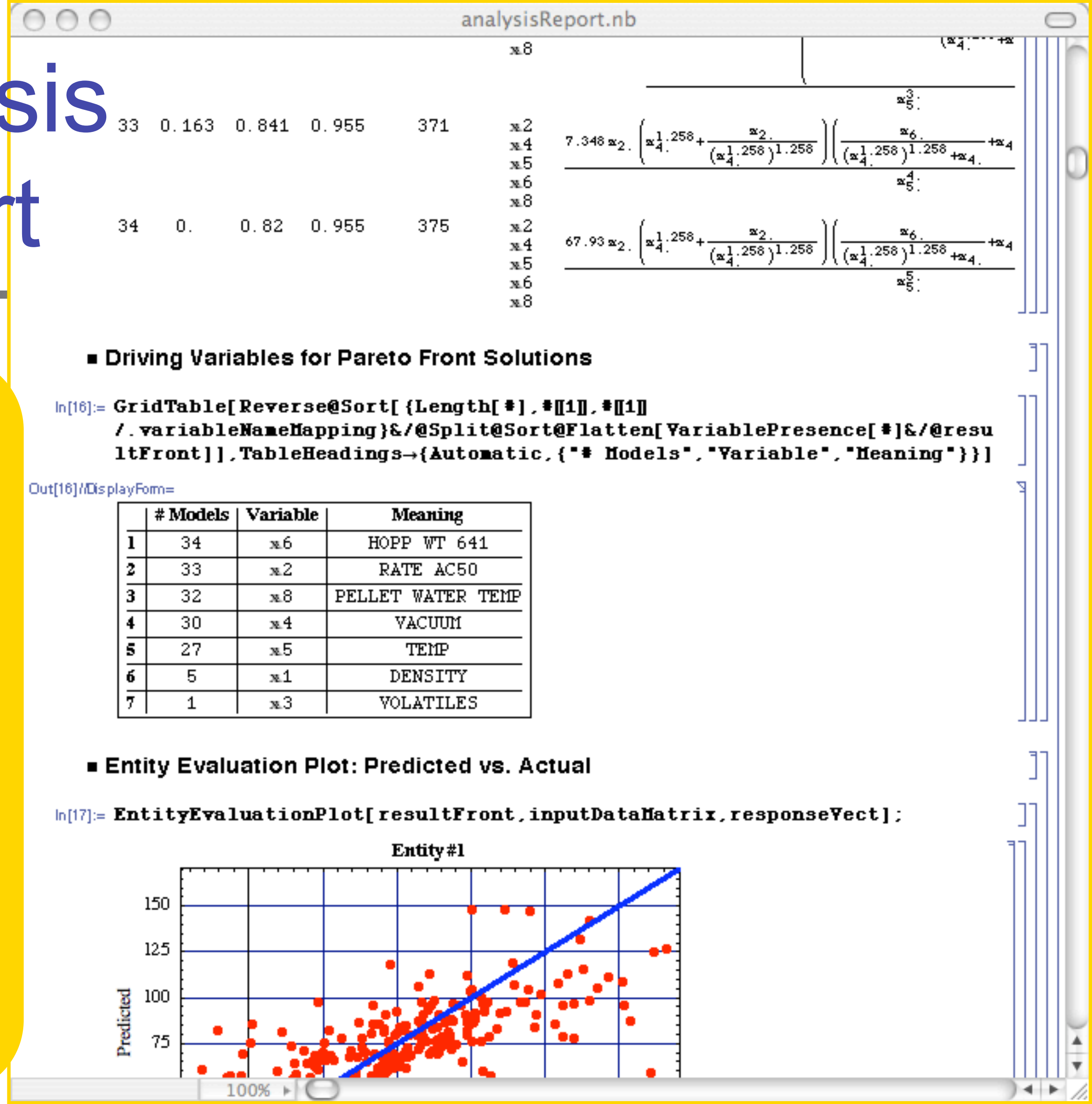
GUIKit Interface



The GUI reduces the barrier for package use.

Analysis Report

Automatically synthesizing the analysis report gives us the best of both worlds: a GUI for data exploration and model development and a notebook for documentation and basis for further exploration.



Summary

- Data modeling is important to industry and has a high impact … if it is done right
- Symbolic regression and other nonlinear data modeling tools can be an important part of successful modeling
- The DataModeler package provides some tools for nonlinear data modeling for both the expert Mathematica user as well as the novice via a GUIKit interface

References

- M. Kotanchek, G. Smits, and A. Kordon, Industrial Strength Genetic Programming, In GP Theory and Practice (R. Riolo and B. Worzel-Eds), Kluwer, 2003.
- A. Kordon, G. Smits, A. Kalos, and E. Jordaan, Robust Soft Sensor Development Using Genetic Programming, In Nature-Inspired Methods in Chemometrics, (R. Leardi-Editor), Elsevier, 2003.
- Kordon A.K, G.F. Smits, E. Jordaan and E. Rightor, Robust Soft Sensors Based on Integration of Genetic Programming, Analytical Neural Networks, and Support Vector Machines, Proceedings of WCCI 2002, Honolulu, pp. 896 – 901, 2002.
- Kotanchek M., A. Kordon, G. Smits, F. Castillo, R. Pell, M.B. Seasholtz, L. Chiang, P. Margl, P.K. Mercure, A. Kalos, Evolutionary Computing in Dow Chemical, Proceedings of GECCO'2002, New York, volume Evolutionary Computation in Industry, pp. 101-110., 2002
- Kordon A. K., H.T. Pham, C.P. Bosnyak, M.E. Kotanchek, and G. F. Smits, Accelerating Industrial Fundamental Model Building with Symbolic Regression: A Case Study with Structure – Property Relationships, Proceedings of GECCO'2002, New York, volume Evolutionary Computation in Industry, pp. 111-116, 2002
- Castillo F., K. Marshall, J. Greens, and A. Kordon, Symbolic Regression in Design of Experiments: A Case Study with Linearizing Transformations, Proceedings of GECCO'2002, New York, pp. 1043-1048.
- Kordon A., E. Jordaan, L. Chew, G. Smits, T. Bruck, K. Haney, and A. Jenings, Biomass Inferential Sensor Based on Ensemble of Models Generated by Genetic Programming, accepted for GECCO 2004, 2004.
- Smits G. and M. Kotanchek, Pareto-Front Exploitation in Symbolic Regression, In GP Theory and Practice (R. Riolo and B. Worzel-Eds), Kluwer, 2004.
- Kordon A., A. Kalos, and B. Adams, Empirical Emulators for Process Monitoring and Optimization, Proceedings of the IEEE 11th Conference on Control and Automation MED'2003, Rhodes, Greece, pp.111, 2003.