Overview

In theory, there is no difference between theory and practice. In practice, there is.
- Jan L.A. van de Snepscheut

- Evolutionary Computing and the business model
- Implementation guidelines
- Integrate & Conquer
- Key application areas
- Open issues

Academic vs. industrial data analysis

Transfer data into knowledge Transfer data into value

Special Features of Industrial Data Analysis

Operators intervention

Curse of closed loops

Operators manually modify the process

The majority of process variables are in closed loops and depend on controller adjustments

Multiple time scales

Time scales vary from milliseconds to months

Real-time pressure

Most of models operate in real time
Intelligent Systems in Industrial Data Analysis: Lessons From the Past

The Expert Systems campaign (late 80s) “We’ll put engineers in the box”
- static rule-based models not linked to numerical world
- the politics of knowledge acquisition
- the efforts of knowledge acquisition

The Neural Networks campaign (early 90s) “We’ll turn data into gold”
- black-box models with inefficient structure
- fragile models and model validation
- maintenance nightmare

Application Issues in the Chemical Industry

High dimensionality of the data
- Highly correlated data with time delays
- Outlier detection
- Multiple optima
- Intensive number crunching needed
- Too much or too little data

Industrial data analysis components

The role of evolutionary computing (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.

Why industry needs Evolutionary Computing?

No a priori modeling assumptions
- Derivative-free optimization
- Few design parameters
- Natural selection of most important inputs
- Parsimonious analytical functions as a final result
- Facilitates human understanding of derived models
- Efficient modeling approach in terms of human time investment
Economic benefits from Evolutionary Computing

- Resolve complex optimization problems
- Physical interpretation & insight (Symbolic regression)
- Reduce model development cost (significantly reduced development time relative to alternatives)
- Reduce model exploitation cost
- Reduce cost of industrial experiments (minimizes the number of additional experiments)

Benefits of integrating Evolutionary Computing with other approaches

- Data with high information content
- Increased quality of generated models
- Model complexity measure
- Reduced model development time and cost
- Condensed data sets
- Faster model selection
- Broader support from different stakeholders
- Final users
- First-principle modelers
- Statistical community
- Machine learning community

Application areas with impact

- Understand Variable Relationships
- Cues to Physical Mechanisms
- Explore Multivariate Relationships
- Infer System States
- Online Monitoring 
- Alarm
- Model Discrimination
- DOE
- Research Acceleration
- Industrial Applications
- Emulators
- System Modeling
- Course Optimization
- Insight into System
- Meaningful Combinations
- Convert into less nonlinear problem
- Identify Variables which drive system
- Variable Transforms
- Nonlinear DOE
- Variable Sensitivity
- Inferential Sensors

Implementation guidelines

- Requirements for successful empirical modeling
- Key issues to be overcome
- Implementation strategy
- Implementation tools
Requirements for successful data-driven modeling

**Objective function:**
Minimizing modeling cost and maximizing data analysis efficiency under broad range of operating conditions

- Good Model Aspects
  - Cost-Effective
  - Interpretability
  - Robustness
  - Extrapolations
  - Credibility
  - Self-Assessment
  - ability to withstand minor changes in targeted system
  - ability to estimate quality of predictions
  - the model matches the observed behavior
  - humans are able to agree that the model is "reasonable"

The total cost-of-ownership (development + operation + maintenance) is proper

- Key issues to overcome
  - Data pre-processing and condensation
  - Model selection
  - User-friendly implementation tools
  - Marketing of EC to different modeling communities
  - Resistance to implement empirical models (inherited from black-box models)
  - Seamless integration into existing maintenance and support infrastructure
  - Critical mass of model developers familiar with EC

**Implementation Strategy**

- Business Opportunities
- Emerging Technologies
- Known Technologies
- Small Projects
- Implementation Methodology
- Large Projects

**Understanding lags application**

(Good judgement comes from experience; experience comes from bad judgement)

**Implementation tools**

- MATLAB (Dow developed)
  - GA
  - GP
  - PSO (single objective and multi-objective)
  - Analytic neural networks
  - Support vector machines
- Mathematica (Dow developed)
  - Symbolic regression package
  - AutoAnalysisTools
  - Analytic neural networks
  - PSO
  - Group Methods of Data Handling (GMDH)
- Tools for model distribution
  - Delphi
  - Web Mathematica
  - Excel
  - Process control systems
Exploitation/Implementation Sequence of Computational Intelligence Approaches in Dow Chemical


Classical NN GA/GP Integrated methodology Pareto GP

Analytic NN SVM PSO

Integrate & Conquer

• Integrated methodology for successful EC implementation
• Related approaches
• A case study

Hybrid approach integrating multiple technologies exploits the strengths of each

Advantages:
– Fast development (days)
– Robust performance (compact models)
– Direct implementation in any Distributed Control System (no need for specialized software)
– Very low capital cost (only if hardware for data collection is unavailable)
– Low average cost of ownership (reduced development and maintenance cost)
– Process engineers like it (preferable to black-box models)
Steps Based on Analytic Neural Nets

Objective: to supply GP with clean, informative, and parsimonious data set

Key idea behind analytic neural networks

If input-to-hidden layer weights \( a_{ij} \) are fixed, there is an analytical solution for the weights \( b_i \) and \( c_i \)

Analytic Neural Network Benefits

- **Robust** algorithm
  - No tunable parameters
  - One global optimum
- **Very fast**, 
  - possible to use a whole range of cross-validation principles from statistics
  - No longer an NP-complete problem
- **Strong theoretical foundation**
  - statistical learning theory
  - Direct measure for the model capacity (VC-dimension)

Key technique for input-to-hidden layer initialization

Hidden nodes have to be within the active region of the nonlinear function

The width of the active zone is defined by the steepness of the function or the “temperature”

The “temperature” depends also on the number of inputs to the hidden node

Empirical expression for a normalized “temperature” of a sigmoid function

\[
T_n = \eta \cdot \frac{\log(2 + \sqrt{3})}{\sqrt{ni - 0.5}}
\]

Weights from the input-to-hidden layer are Sampled from a normal distribution
Stacked Analytic Neural Nets (SANN)

- Fast development
- Diverse subnet consensus indicator of model output quality
- Allows explicit calculations of input/output sensitivity
- Can handle time-delayed inputs by convolution functions
- Gives more reliable estimates based on multiple models statistics

Internally developed in Dow Chemical by Guido Smits

Steps Based on Support Vector Machines

- Representative data collection
- Data preprocessing and classification
- Sensitivity analysis of all inputs
- Convolution parameters’ estimation
- Outlier detection and data-set condensation
- GP function generation
- Analytical function selection/verification
- On-line implementation
- Model maintenance

Reliable outlier detection

Data compression (industrial data from a chemical reactor) Only 40% of data used

Advantages
- Solid theoretical basis => Statistical Learning Theory
- Model building is based on global optimum
- Explicit control over model complexity

Issues
- ad hoc Kernel selection
- Complex theory
- No commercial software
- Computationally intensive

Support Vector Machines for Classification

Only 3 support vectors needed to define optimal hyperplane

Key to robust modeling

Support vector

Optimal hyperplane

Key to robust modeling

Sensitivity analysis of various process parameters on catalyst efficiency

NN model performance with model disagreement indicator

Model disagreement indicator
The generic scheme of SVM

- Decision rule based on weights and support vectors
- Nonlinear transformations based on support vectors: $x_1, x_2, \ldots, x_N$
- Weights $\alpha_1, \alpha_2, \ldots, \alpha_N$
- Decision rule based on weights and support vectors

Support Vector Machines and Neural Networks

- Neural Network
- Support Vector Machine
- Optimal hyperplane in feature space
- Feature space
- Input space

Structural Risk Minimization Principle

- Trade-off between quality of approximation of the given data and the complexity of the approximating function.
- The VC-dimension is now a controlling variable.
- Chooses the set of functions with the lowest VC-dimension for which minimizing the empirical risk gives the best bound on the actual risk.
- Minimize

$$R(\alpha) \leq R_{\text{emp}}(\alpha) + \Phi(\frac{1}{n})$$

Where $\alpha$ is the model parameter of interest, $n$ is the sample size and $h$ is the complexity measure.
Structural Risk Minimization in learning algorithms

- Keep $\Phi(\frac{\cdot}{n})$ fixed, minimize $R_{\text{emp}}(\alpha)$
  - Neural Networks

- Keep $R_{\text{emp}}(\alpha)$ fixed, minimize $\Phi(\frac{\cdot}{n})$
  - Support Vector Machines

Neural Networks and Support Vector Machines are two sides of the same coin

SVM for Regression: Constructing a tube

Insensitive zone (tuning parameter)

Generalization capabilities of SVM based on mixed kernels

Support Vector Machine model based on mixed polynomial and RBF kernels

Data set from a chemical reactor inferential sensor

Impressive generalization 50% outside the training range!

Genetic Programming

- Based on artificial evolution of millions of potential nonlinear functions $\Rightarrow$ survival of the fittest
- Many possible solutions with different levels of complexity
- The final result is an explicit nonlinear function
- Better generalization capabilities than neural nets
- Low implementation requirements
- Time delays
- Sensitivity analysis of large data sets
- Relatively slow (several hours of computational time)
Steps Based on Genetic Programming

- Representative data collection
- Data preprocessing and classification
- Sensitivity analysis of all inputs
- Convolution parameters' estimation
- Outlier detection and data set condensation
- GP function generation
- Analytical function selection/verification
- On-line implementation
- Model maintenance

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
  - 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?

Pareto Performance

- Characterizing Pareto Performance
  - Dominance
  - Domination
  - Layer
  - Combinations …
- Computational Issues
  - Brute force is $M^2 N$
  - Can do $M N \log M^2 (N)$ if clever
    - $M = \#$ of objectives
    - $N = \#$ population size
  - Computation demands need to be considered in algorithm design

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

- What is complexity?
  - # of nodes?
  - Tree depth?
  - Included functions?
  - Number of variables?
  - Combinations?
- Chosen function is sum of sum of node counts
  - Provides more resolution at low end of complexity than simply using node count
  - Rewards fewer layers
- Real goal is to characterize the (relative) “smoothness” of the evolved function

\[
\text{Complexity} = 36 - 27 \times \frac{1}{2} = 17
\]

Pareto GP Algorithm

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

Pareto GP Features

- 60-100x speedup relative to prior symbolic regression implementation
- Improved solutions
- Implicit intron suppression
- Facilitates …
  - Generation of diverse ensembles
  - Metavariable identification
  - Faster post-run analysis

Solutions of similar performance but diverse structure

Pareto Front

- 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
**Particle swarm optimization**

An efficient technique to find the global optimum for model inversion and non-linear parameter estimation

At each time step $t$

For each particle $i$

Update the position change (velocity)

$$V_i(t+1) = \chi \cdot (V_i(t) + c_1 \cdot \text{rand}(0,1) \cdot (P_i(t) - X_i(t)) + c_2 \cdot \text{rand}(0,1) \cdot (P_g(t) - X_i(t)))$$

Then move

$$X_i(t+1) = X_i(t) + V_i(t+1)$$

Note: - stochastic component
- parameters $c_1, c_2, \chi$ default values (2.05, 2.05, 0.73)

• Note the pathologies at high complexity when extrapolating
Multi-Objective PSO

Efficient technique to determine the Pareto front for problems with convex, non-convex and even disconnected Pareto fronts.

Case Study: Inferential Sensors

Key objective:
To predict difficult-to-measure parameter (melt index) from easy-to-measure data (temperature, pressure, flow, etc.)

Inferential Sensor
An empirical model based on analytical equations with built-in self-assessment capability

Software tools

Issues with neural net-based inferential sensors:
- High sensitivity to process changes
- Frequent re-training
- Complicated development & maintenance
- Low survival rate after 3 years in operation
- Engineers hate black-boxes
Inferential sensor for emission monitoring: A case study

Data Collection

- 251 training data points
- 107 test data points (~140% outside training range)

Chemical Process

Design Of Experiments

- 6 inputs

Emission variable

Sensitivity analysis by SANN

- A NN with 4 inputs: x2, x5, x6, and x8 is selected after discussion with process engineers
- Input x3 removed after first sequence
- Input x7 removed after second sequence
- Input x6 has the strongest sensitivity

Bad extrapolation (test data is 140% outside the range of training data)

Model based on 30 stacked NN with 10 neurons in hidden layer

Reduced number of inputs from 8 to 4

Fast test of the hypothesis about potential nonlinear relationship (in 20-30 min)

Measured emission variable

Predicted emission variable

SVM parameters

- Parameters:
  - % support vectors: 10
  - C = 10^6
  - Mixed Kernels: Polynomial and RBF
  - Range of Polynomial kernels: 1-3
  - Range of RBF kernel: 0.25-0.75
  - Range of ratio 0.5 – 0.99
Inferential sensor for emission monitoring: A case study (SVM model performance)

Impressive extrapolation (test data is 140% outside the range of training data).

Model based on a mixture of 2nd order polynomial global kernel and RBF local kernel with width of 0.5 and ratio of 0.95.

Reduced number of training data points from 251 to 34 (based on support vectors).

Inferential sensor for emission monitoring: A case study (GP parameters)

Parameters for a GP simulated evolution:

- Reference data: 34
- Random subset selection [%]: 100
- Number of runs: 20
- Population size: 500
- Number of generations: 100
- Probability for function as next node: 0.6
- Optimization function: Corr.
- Parsimony pressure: 0.1
- Prob. for random vs guided crossover: 0.5
- Probability for mutation of terminals: 0.3
- Probability for mutation of functions: 0.3

Inferential sensor for emission monitoring: A case study (Selected symbolic regression model)

Simple expression with acceptable performance ($R^2 = 0.87$).

Response surface of model according to process physics.

Selected model on Pareto front.

Inferential sensor for emission monitoring: A case study (Final solution: Stacked Symbolic Regression model)

Model based on 8 Stacked Symbolic Predictors.

In operation since August 2001.

Shorter evolutionary process based on 8.44% of the original training data set.
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

EC Applications in Dow Chemical

<table>
<thead>
<tr>
<th>Application Domains</th>
<th>Examples</th>
</tr>
</thead>
</table>
| **Material Design** | - Color Matching  
- Appearance Engineering  
- Polymer Design  
- Synthetic Leather |
| **Materials Research** | - Diverse Chemical Library Selection  
- Fundamental Model Building  
- Reaction Kinetics Modeling  
- Combi-Chem Catalyst Exploration  
- Combi-Chem Data Analysis |
| **Production Design** | - Acoustic Nullifier Emulator  
- EDC/VCM Nonlinear DOE  
- Bioreactor Optimization |
| **Production Monitoring & Analysis** | - Epoxy Holdup Monitoring  
- Iwanzynate Level Estimation  
- FTIR Calibration Variable Selection  
- Poly-3 Volatile Emission Monitoring  
- Epoxy Intelligent Alarm Processing  
- PerTet Emulator for Online Optimization  
- Emissions Monitoring |
| **Business Modeling** | - Diffusion of Innovation  
- Hydrocarbon Trading & Energy Systems Optimization  
- Scheduling Heuristics  
- Plant Capacity Drivers |

Automating Operating Discipline

- Heuristic rules defined verbally by process engineers/operators
- holdup predictor designed by stacked analytic NN and GP
- all decision blocks have fuzzy thresholds defined by membership functions
- simple empirical models and mass balances
- fundamental model predictions are used in the heuristic rules
- reduced major shutdowns
- reduced lab sampling

Emulator for optimization of an industrial chemical process

- Four levels DOE
- Training Data set
- Symbolic Regression Emulator
- On-line process optimization
- 10 inputs
- 12 outputs
Symbolic regression-based emulator’s performance

Simple expression for on-line implementation

\[ Y5 = 3x_0 + \frac{6x_1 + x_4 + x_5 + 2x_6 + x_2 + x_3 - 3x_0 - \sqrt{x_0 - 2x_1 - x_0 + 10}}{\log(x_2^2)} \]

Acceptable performance

Y5 Symbolic Regression Training \( (R^2 = 0.94) \)

<table>
<thead>
<tr>
<th>Sample Number</th>
<th>Value of Y5</th>
<th>Actual Y5</th>
<th>Predicted Y5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>21</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>41</td>
<td>41</td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>61</td>
<td>61</td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>81</td>
<td>81</td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>101</td>
<td>101</td>
<td></td>
</tr>
<tr>
<td>600</td>
<td>121</td>
<td>121</td>
<td></td>
</tr>
<tr>
<td>700</td>
<td>141</td>
<td>141</td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>161</td>
<td>161</td>
<td></td>
</tr>
<tr>
<td>900</td>
<td>181</td>
<td>181</td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>201</td>
<td>201</td>
<td></td>
</tr>
</tbody>
</table>

The evolution process identifies the key input variables as well as natural groupings & relationships. Combining this with a domain knowledge and first-principles insights is very powerful.

Accelerated Fundamental Model Building Based on GP

1. Problem definition
2. Run symbolic regression
3. Identify key factors
4. Construct the model
5. Solve the model
6. Select & verify the final model solution
7. Validate the model

Virtual modelers

Can we accelerate hypothesis search by simulated evolution?

Classical fundamental model building steps

Accelerated fundamental model building steps

Approaches to accelerate fundamental model building process

AI approach

Reduce hypothesis search by GP

GP as automated invention machine

Mimic the expert

Eliminate the expert

Maximize creativity of the expert
The problem of structure-properties in fundamental modeling

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

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

Key modeling effort for new product development

Results from hypothesis search
Selected symbolic regression empirical model

Fundamental model

\[ y = a + [b \cdot x_1 + c \cdot \log(x_2)] \cdot e^{x_3} + d \cdot x_4 \]

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

Selected empirical model

- Square root form for x1
- Linear form for x5
- Exponential form for x3
- Logarithmic form for x2

Case Study with Structure-Property Relationships

Teoretical Analysis

\[ \frac{dT}{dt} = \frac{\beta^2}{\alpha^2} \cdot DH \cdot \frac{dt}{d\alpha} \]

Fundamental Model Building

$y = a + [b \cdot x_1 + c \cdot \log(x_2)] \cdot e^{d \cdot x_3} + d \cdot x_4$

Hypothesis Search

Fundamental model

Comparative Analysis of Symbolic Regression in Fundamental Model Building

Advantages of Symbolic Regression

- Model Development Speed
  - 10 hours vs. 3 months
- Summarize Multivariate Data
  - convert data into equations to facilitate human insight
  - can explore parameter sensitivity and play what-if games
- Accuracy
  - achieved > 90% correlation with experimental results
- Identify Key Variables and Transforms
  - with the exception of x1, symbolic regression captured correctly all other functional forms in the model
- Suggest Physical Mechanisms
  - evolved expressions and equation "building blocks" may be interpreted from a first-principles viewpoint
- Suggest Future Experiments
  - options in evolved expressions may be validated in future experiments

Disadvantages of Symbolic Regression

- Blind to Physics and Chemistry
  - genetic programming does not currently take into account the physical or chemical laws
  - expressions may have no physical meaning - mathematical consistency is how fitness is defined
  - inclusion of physical constraints is a research topic
- Garbage-In/Garbage Out
  - appropriate variables must be supplied
  - data is assumed to be accurate
  - operational range should be covered

Experts (Scientists) are Still Required

- domain expert is an absolute must for interpretation of evolved expressions
- The domain expert delivers the final fundamental model
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
   \[ y = \beta_0 + \sum \beta_i x_i + \sum \beta_{ij} x_i x_j \]
   Classical approach if LOF
   - add experiments to fit second order model
   \[ S_4 = \beta_4 + \sum \beta_i x_i + \sum \beta_{ij} x_i x_j + \sum \beta_{ijk} x_i x_j x_k \]
   Classical approach if LOF
   - no alternative (use model as it is)

2. A response surface DOE
   - already had all experiments to fit second order model
   \[ S_4 = \beta_4 + \sum \beta_i Z_i + \sum \beta_{ij} Z_i Z_j + \sum \beta_{ijk} Z_i Z_j Z_k \]

Suggested approach:
- Use GP to transform inputs

More costly experiments

---

**PSO application: Optimizing color spectrum of plastics**

Real-time optimization in 2-3 seconds

Multiple-objective PSO with 15 variables

**Other PSO applications**

- **Drug release predictor**
  - 6 parameters
  - population size = 30
  - optimization time: ~ 30 seconds
- **Foam acoustics performance predictor**
  - 8 parameters
  - population size = 50
  - optimization time: ~ 5 seconds
- **Crystallization kinetics predictor**
  - 4 parameters
  - population size = 30
  - optimization time: ~ 2 seconds
Open Issues

Complexity Control & Smoothness Characterization
Diversity Detection
Identifying Metavariabes
Convert Hard GP Problems into Easy Ones
Should we be doing Cultural Programming?
Blending Heuristics & Prior Knowledge

Acknowledgement

We would like to acknowledge the contribution of the following researchers from The Dow Chemical Company:

Alex Kalos
Kip Mercure
Flor Castillo
Elsa Jordaan
Leo Chiang
Irina Graf

Summary

- Evolutionary Computing can create significant value to industry by reducing model development time and model exploitation cost
- Integrating EC with Neural Networks, Support Vector Machines, and Statistics is recommended for successful industrial applications
- This strategy works for many real applications in the chemical industry
- The key application areas are:
  - Inferential sensors
  - Improved process monitoring and control
  - Accelerated new product development
  - Effective design of experiments
- And this is only the beginning …

References