Mathematical Challenges and Aides to Tuning Models to Complex Social Systems

W. L. Hare

May 15th, 2008

Abstract

Complex social systems are any system that is governed, at least in part, by social dynamics. In developing mathematical models of complex social systems, one is often left with a incomplete picture of the system being modelled. In particular most models contain parameters that are unmeasurable by standard data collection techniques. (For example, the societal ratio to which one values family-time over increased income.) Determining best approximations for these parameters is typically done by a system of trial and error, seeking a combination of parameters that result in the model fitting a known data set. In this talk we discuss how using derivative free optimization can automate this process and vastly improve over all fit. An example model regarding Home and Community Care in British Columbia is given.

Outline

1. Introduction: Modelling of Complex Social Systems
2. Example: Home and Community Care
3. Tuning Unknown Parameters
4. Summary and Future Research

Introduction

Introduction

A complex social system (CSS) is any system that is governed, at least in part, by social dynamics.

The problem of "modelling of complex social systems" (MoCSS) is rapidly becoming one of the most interesting and challenging problems in Mathematics.

DARPA Mathematical Challenge Two:

Develop the high-dimensional mathematics needed to accurately **model** *and* **predict** *behaviour*

. . .

occurring in social systems.

Examples

There are many real reasons to Model Complex Social Systems:

- Criminal Networks: understanding how social networks influence crime rates.
- Health Care: understanding how social behaviour influences the demand for health care.
- Epidemiology: understanding how social status impacts spread of disease.

An Example

Home and Community Care

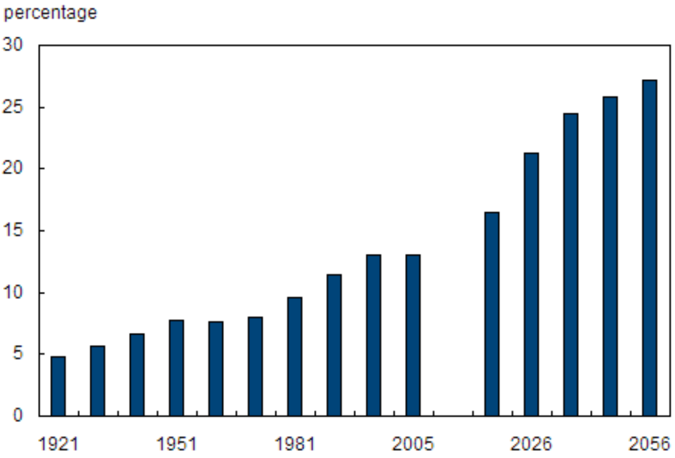
Example: modelling the future of Home and Community Care (HCC)

- HCC is the branch of healthcare which deals with long term health services outside of a hospital:
  - post-surgery rehabilitation,
  - home care nursing,
  - adult day centres
  - assisted living,
  - residential care, etc...
- In British Columbia HCC is opt-in and has user fees based on patient income levels.

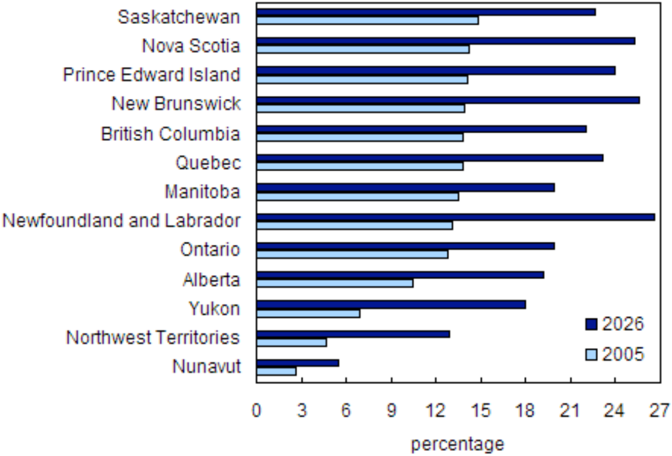
Why Model HCC?

- About 80% of H.C.C. clients are seniors (65+).
- The proportion of seniors in Canada is increasing.

Why Model HCC?



Why Model HCC?



Challenges in Modelling HCC

The Question:

- How many people will use publicly funded HCC services in the year 2011, 2016, 2021, 2026, 2031?
- Which services will they use?

Our Model

Deterministic Multi-state Markov Model

- details aren't important

Our Model

Deterministic Multi-state Markov Model

- details aren't important

The model incorporates:

- Relationship of Population Wealth to Private Care Options

No data is available on this topic

Our Model Challenge

The model contains 13 unknown parameters that control how

- Population Wealth Distribution and Private HCC

impact HCC service usage

Our Model Challenge

The model contains 13 unknown parameters that control how

- Population Wealth Distribution and Private HCC

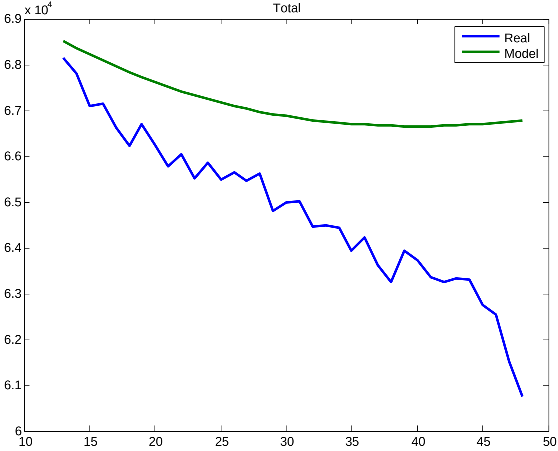
impact HCC service usage

If these parameters are set to zero, the model assumes that

- nobody ever uses private HCC

Baseline Model Fit

Model Fit if all parameters are zero



$$r^2 = -1.454329$$

Tuning Unknown Parameters

Math

Let us consider the problem as follows

- Let $x_1, x_2, ...x_{13}$ represent the 13 unknown parameters
- Let $f(x)$ map a parameter selection to its resulting r^2 value

Then we seek to locate a solution to

$$\max\{f(x_1, x_2 ... x_{13})\}$$

such that

$$x_1, x_2, ...x_{13} \quad \text{are acceptable parameters}$$

Challenges

There are two challenges here

1. f is not analytically defined
2. the parameters cannot take any value

Challenges

There are two challenges here

1. f is not analytically defined
2. the parameters cannot take any value

In this case the following constraints must hold:

$$
\begin{aligned}
&x_i \geq 0 \quad \text{for all } i \\
&x_3 \leq 21000 \\
&x_4 \leq x_5 \leq x_6 \leq x_7 \leq x_8 \leq 1 \\
&x_9 \leq 0.75 \\
&x_{10}, x_{11}, x_{12}, x_{13} \leq 1
\end{aligned}
$$

Derivative Free Optimization

So **how do you solve**

$$\max\{f(x) : x \in S\}$$

when f is not analytically known?

Derivative Free Optimization

So **how do you solve**

$$\max\{f(x) : x \in S\}$$

when f is not analytically known?

1. Best Guess (ask an expert)
2. Grid Search
3. Random Search
4. Advanced Random Search
   (Simulated Annealing, Genetic Algorithms, etc...)
5. Basic Pattern Search
6. Generalized Pattern Search
   (Nelder-Mead, MADS, CONDOR, etc...)

Derivative Free Optimization

Best Guess: Examine what each parameter represents and take an educated guess

Derivative Free Optimization

Best Guess: Examine what each parameter represents and take an educated guess

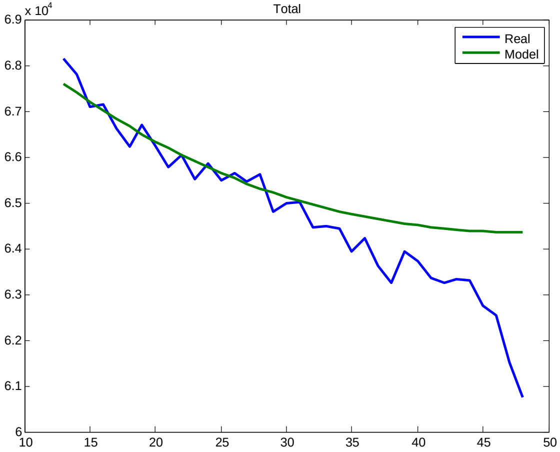
Advantages:

- Simple
- Quick

Disadvantages:

- Naive
- Parameters usually end up rounded off
- Seldom accurate

Best Guess



$$r^2 = 0.824967$$

Grid Search

Grid Search: Lay down a grid over constraint area and check every possible combination of parameters

Grid Search

Grid Search: Lay down a grid over constraint area and check every possible combination of parameters

Advantages:

- Simple
- Can control how many significant digits you want
- Reasonable guarantee of global optimality

Disadvantages:

- Scales horribly

  (one significant digit for 13 parameters takes 10^{13} model runs)
- Unbounded parameters are difficult to deal with

Random Search

Random Search: Pick a starting point x^0, initialize $i = 1$ then:

1. Generate a random vector δ
2. Evaluate Function at point $x^{i-1} + \delta$,

   a) if $f(x^{i-1} + \delta) > f(x^{i-1})$ set $x^i = x^{i-1} + \delta$

   b) otherwise set $x^i = x^{i-1}$
3. Increase $i \to i + 1$ and repeat as desired

Random Search

Random Search: Pick a starting point x^0, initialize $i = 1$ then:

1. Generate a random vector δ
2. Evaluate Function at point $x^{i-1} + \delta$,

   a) if $f(x^{i-1} + \delta) > f(x^{i-1})$ set $x^i = x^{i-1} + \delta$

   b) otherwise set $x^i = x^{i-1}$
3. Increase $i \to i + 1$ and repeat as desired

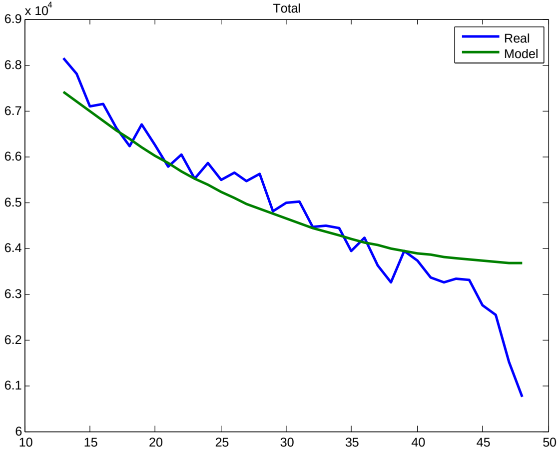
Advantages:

- Simple
- Probabilistic proof of local optimality

Disadvantages:

- No convergence proof of optimality
- Easily gets stuck at local maxima
- Bounded parameters are hard to deal with

Random Search (1000 Model Runs)



$$r^2 = 0.909752$$

Advanced Random Search: Simulated Annealing

Simulated Annealing: Pick a starting point x^0, initialize $i = 1$ then:

1. Generate a random vector δ and a scale factor s^i
2. Evaluate Function at point $x^{i-1} + s^i\delta$,

   a) if $f(x^{i-1} + s^i\delta) > f(x^{i-1})$ set $x^i = x^{i-1} + s^i\delta$

   b) otherwise set $x^i = x^{i-1}$
3. Increase $i \rightarrow i + 1$ and repeat as desired

By sliding the scale factor s^i up and down, one allows the algorithm to jump out of local maxima.

Advanced Random Search: Simulated Annealing

Simulated Annealing: Pick a starting point x^0, initialize $i = 1$ then:

1. Generate a random vector δ and a scale factor s^i
2. Evaluate Function at point $x^{i-1} + s^i\delta$,
   a) if $f(x^{i-1} + s^i\delta) > f(x^{i-1})$ set $x^i = x^{i-1} + s^i\delta$
   b) otherwise set $x^i = x^{i-1}$
3. Increase $i \to i + 1$ and repeat as desired

By sliding the scale factor s^i up and down, one allows the algorithm to jump out of local maxima.

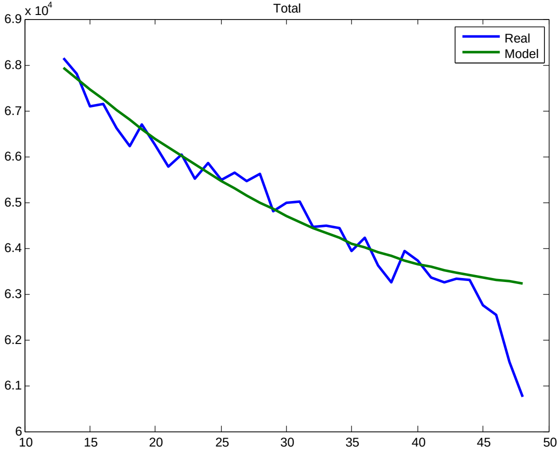
Advantages:

- Probabilistic proof of local optimality
- Can jump away from local maxima

Disadvantages:

- No convergence proof of optimality
- Can still get stuck at local maxima
- Bounded parameters are hard to deal with

Simulated Annealing (1000 Model Runs)



$$r^2 = 0.949591$$

Other Advanced Random Searches

Genetic Algorithms

- Breed good solutions together with random error to mimic evolution.

Tabu Algorithms

- Enforce new "tabu" constraints to help algorithm jump away from local maxima.

Ant Colony Algorithms

- Good solutions lay down a scent that attracts the random process.

Basic Pattern Search

Pattern Search: Pick a starting point x^0, a pattern P, set initial scale $s = 1$, initialize $i = 1$ then:

1. Evaluate Function at all points in $x^{i-1} + sP$
   a) if $f(x^{i-1} + sp) > f(x)$ for some p, set $x^i = x^{i-1} + sp$
   b) otherwise reduce scale factor $s \rightarrow s/2$
2. Increase $i \rightarrow i + 1$ and repeat as desired

Basic Pattern Search

Pattern Search: Pick a starting point x^0, a pattern P, set initial scale $s = 1$, initialize $i = 1$ then:

1. Evaluate Function at all points in $x^{i-1} + sP$

    a) if $f(x^{i-1} + sp) > f(x)$ for some p, set $x^i = x^{i-1} + sp$

    b) otherwise reduce scale factor $s \rightarrow s/2$

2. Increase $i \rightarrow i + 1$ and repeat as desired

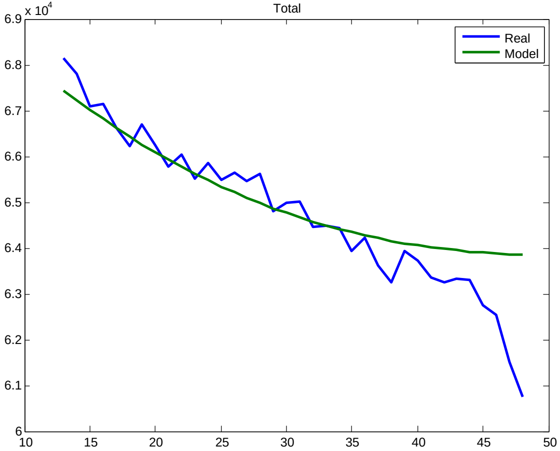
Advantages:

- Simple
- Convergence proof of optimality under some (strong) conditions
- Reproducible results

Disadvantages:

- Can be very slow if P is not the right "shape"
- Very easily gets stuck if P is not the right "shape"

Basic Pattern Search (1014 Model Runs)



$$r^2 = 0.908277$$

Generalized Pattern Searches

Various rules have been created on

- how to **select** the pattern set P and
- how to **adapt** the pattern set during the algorithm

Generalized Pattern Searches

Various rules have been created on

- how to **select** the pattern set P and
- how to **adapt** the pattern set during the algorithm

Advantages:

- Stronger convergence proofs of local optimality
- Often faster convergence

Disadvantages:

- More complicated to code
- Most algorithms cannot deal with constraints

Generalized Pattern Searches

Nelder-Mead

- by Nelder and Mead (1965)
- Classic taught in undergraduate Numerical Optimization
- Many free codes available (not all user friendly)

MDS (Multi-Directional Search)

- by Torczon (1991)

UOBYQA (Unconstrained Optimization by Quadratical Approximation)

- by Powell (2000)

MADS (Mesh Adaptive Direct Search)

- by Audit and Dennis (2004, 2006)

Generalized Pattern Searches dealing with Constraints

Most Generalized Patterns Searches do not allow for constraints

- One way to deal with this is rewrite

$$\max\{f(x) : x \in S\} = \max\{f(x) - i_S\}$$

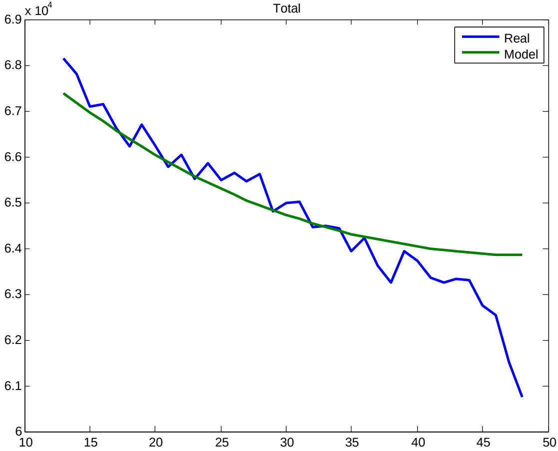
  where i_S is the indicator of S

$$i_S(x) = \begin{cases} 0 & x \in S \\ \infty & x \notin S \end{cases}$$

- We call this a boundary penalty

Nelder-Mead with Boundary Penalty (618 Model Runs)

MATLAB Code by C. T. Kelley, December 12, 1996



$$r^2 = 0.896671$$

Generalized Pattern Searches dealing with Constraints

- Another way to deal with constraint is Boundary Projection:
- Define

$$F(x) = f(P_S(x))$$

  where P_S is the projection of x onto S
- Then find

$$\max\{F(x)\}$$

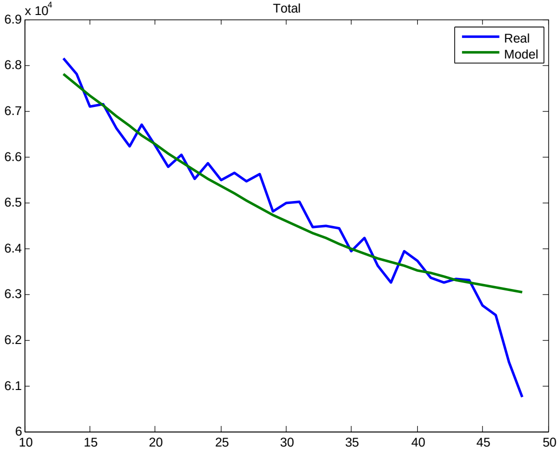
- The final parameter selection is

$$P_S(x^*)$$

  x^* is the best found solution to the above
- This is only useful if P_S is easy to solve

Nelder-Mead with Boundary Projection (1002 Model Runs)

MATLAB Code by C. T. Kelley, December 12, 1996



$$r^2 = 0.943960$$

Summary and Future Goals

Summary

- Modelling of Complex Social Systems (MoCSS) is rapidly growing and challenge area of mathematics
- Research in MoCSS often results in models with unknown parameters that need tuning
- **Derivative Free Optimization (DFO) provides mathematical tools to analyze unstructured optimization problems**
- DFO methods range from trivial (guess) to very complex (MADS)
- Most DFO methods do not allow for problem constraints, so are inapplicable
- Further study of current methods is a first step in the development of new methods that allow for DFO with constraints

DFO Method Comparison Summary

DFO Method	Model Runs	Resulting r^2
Base Line	1	-1.454
Best Guess	1	0.825
Random Search	1000	0.910
Simulated Annealing	1000	0.950
Pattern Search	1014	0.908
Nelder-Mead (penalty)	618	0.897
Nelder-Mead (project)	1002	0.944

Thank You