PSY8219 : Week 12

Homework 10 Due today

Readings for Today
  None
ACCRE Presentation (after Thanksgiving break)

students who were previously not ACCRE users should have received an email with instructions for logging into the cluster, changing their passwords, etc.
Connecting to ACCRE

http://www.accre.vanderbilt.edu/?page_id=326

Logging onto the Cluster

For Mac or Linux users:

For access to ACCRE resources, you will need a Secure Shell (SSH client) on your local machine. Unix, Linux, and Mac OSX users can open their Terminal application (which is installed by default on these operating systems) to use SSH. In Mac OSX look in Applications--->Utilities for the Terminal application. To log in to the cluster from the Terminal, type:

```
ssh vunetid@login.accre.vanderbilt.edu
```

where vunetid is your unique Vanderbilt ID. You may see a message asking if you trust the system you are attempting to log in to. Type “yes” and hit return. You will then be prompted for your password, which you should have received in an email from ACCRE. Enter your password and hit return. Note that for security reasons the password will not appear as you are typing.

For Windows users:

- Download and install PuTTY for Windows

Once installed, run PuTTY, type login.accre.vanderbilt.edu into the hostname box, and click Open. You may see a message asking if you trust the system you are attempting to log in to. Click “yes.” Connect to the gateway, and enter the username and password assigned to you when you requested an account. Note that for security reasons the password will not appear as you are typing. You are now logged onto the cluster!

Please continue to your first cluster session.
Copying Files to ACCRE

https://filezilla-project.org
Optimization and Curve Fitting
slope and intercept parameters are calculated using closed-form mathematical expressions

Linear Regression Model

vs

Nonlinear Statistical, Cognitive, Neural Model

model parameters can be estimated using a variety of search/optimization approaches
Week12_Figure.m

imagine trying to fit data by adjusting all the parameters by hand
what values of $\alpha$, $\beta$, $\gamma$, $\lambda$ provide the best account of the observed data?

fitting a psychometric function to observed data

$$P(\text{correct} \mid x) = \gamma + (1 - \lambda - \gamma)F(x; \alpha, \beta)$$

$$F(x; \alpha, \beta) = 1 - \exp(-(x / \beta)^\alpha)$$
Motion coherence and accuracy, e.g., what value of coherence gives chance performance.

Fitting a psychometric function to observed data.
is the same psychometric function observed when another variable is manipulated (or not)?

is the threshold significantly higher in one condition than the other?
fit a power law function to a learning curve

\[ RT(n) = A + Bn^{-c} \]
is a power law or an exponential law a better function?

\[ RT(n) = A + Bn^{-C} \quad \text{vs.} \quad RT(n) = A + Be^{-C(n-1)} \]
Figure 1 | Experimental approach and results of experiment 1. a, Colour recall task. b, Mixture model of performance, showing the probability of reporting each colour value given a sample colour at 180°. When the probed item is present in memory, the reported colour tends to be near the original colour (blue broken line). When the probed item is not present in memory, the observer is equally likely to report any colour value (red broken line). When collapsed across trials, the data comprise a mixture of these two trial types (solid line), weighted by the probability that the probed item was stored in memory. c, Results of experiment 1 (N = 8). P_m and s.d. are defined in the text.
\[ \text{Prob}(x \mid \mu, \sigma, P_m) = P_m \frac{e^{\cos(x-\mu)/\sigma}}{2\pi I_0(1/\sigma)} + (1 - P_m) \frac{1}{2\pi} \]

- \( \mu, \sigma, P_m \) used to characterize VWM in a given condition
- probability an item is stored in VWM
- fidelity of the memory
- mean content of the memory
Stochastic Accumulation of Evidence Models

perceptual processing time

\( T_R \)

drift

\( a \)

Response times

motor response

\( T_M \)

time
Different kinds of models

- purely mathematical / statistical models
  - nonlinear regression
  - power law of learning
  - psychometric function

- hybrid mathematical / process model
  - mixture models of power law functions
  - mixture model of VWM

- process / mechanistic model (cognitive or neural)
  - diffusion model of perceptual decision making
  - temporal context model of memory
optimization and parameter estimation

model
optimization and parameter estimation

model

parameters
optimization and parameter estimation

model

parameters

prediction
optimization and parameter estimation

\[ \text{model} \rightarrow \text{prediction} \equiv \text{data} \]

\[ \text{parameters} \]
we need to find parameters that cause the model to make predictions that match the data
optimization and parameter estimation

unknown and unobservable

parameters → social, behavioral, or neural process → observable
data
optimization and parameter estimation

unknown and unobservable parameters → social, behavioral, or neural process → observable data

model of that process

hypothesized

perhaps one of several competing models being considered
optimization and parameter estimation

**unknown and unobservable**

parameters → social, behavioral, or neural process → data

**observable**

parameters → model of that process → prediction

**hypothesized**

**generated**
optimization and parameter estimation

Parameters → social, behavioral, or neural process → data

unknown and unobservable

maximize the correspondence

minimize the discrepancy

observable

data

model of that process

prediction

hypothesized

parameters → model of that process → prediction

generated
optimization and parameter estimation

unknown and unobservable

parameters → social, behavioral, or neural process → data

minimize the discrepancy
maximize the correspondence

observable

hypothesized

parameters by adjusting these parameters

model of that process → prediction

generated
optimization and parameter estimation

- unknown and unobservable parameters

- social, behavioral, or neural process

- data

- observable

- minimize the discrepancy

- maximize the correspondence

- hypothesis

- model of that process

- prediction

- generated

- parameters

- by adjusting these parameters

- if we’ve optimized competing models, we can statistically determine whether some models predict better than others
optimization and parameter estimation

unknown and unobservable

parameters \rightarrow social, behavioral, or neural process \rightarrow data

minimize the discrepancy
maximize the correspondence

hypothesized

parameters \rightarrow model of that process \rightarrow prediction

generated

if we’ve optimized competing models, we can statistically determine whether some models predict better than others
optimization and parameter estimation

unknown and unobservable

parameters → social, behavioral, or neural process → data

minimize the discrepancy
maximize the correspondence

observable

parameters → model of that process → prediction
generated

by adjusting these parameters

hypothesized

sometimes, parameters can be set based on one set of data and used (without fitting) for another situation ...
Some examples (often called “objective functions”)

**Sum-of-Squared Error (SSE)**

\[ SSE = \sum_{i} (obs_i - prd_i)^2 \]

**Correlation**

\[ r = \sqrt{\frac{\sum_{i} (obs_i - \overline{obs})(prd_i - \overline{prd})}{\sum_{i} (obs_i - \overline{obs})^2 \sum_{i} (prd_i - \overline{prd})^2}} \]

**\(\chi^2\)**

\[ \chi^2 = \sum_{i} \frac{(obs_i - prd_i)^2}{prd_i} \]

some objective functions are more conducive to statistical tests
with only one parameter, we want to find the value of parameter $p$ that makes the $SSE$ as small as possible – best fitting value of $p$
Two-parameter model

values of $p1$ and $p2$ that minimize $SSE$
Three-parameter (or more) model

cannot visualize graphically, but the concept is the same ...

what values of parameters minimize SSE
(or maximize correlation, or maximize likelihood)
a brief survey of optimization methods
a brief survey of optimization methods

**Brute Force : Grid Search**

try every combination of parameters and keep the one that minimizes/maximizes your objective function (e.g., SSE)
a brief survey of optimization methods

Brute Force : Grid Search
	ry every combination of parameters and keep the one that minimizes/maximizes your objective function (e.g., SSE)

basically, it's the same programming you would use to graph the function, only that you're also trying to find the min
a brief survey of optimization methods

**Brute Force: Grid Search**

try every combination of parameters and keep the one that minimizes/maximizes your objective function (e.g., SSE)

like an earlier homework problem
Brute Force: Grid Search

try every combination of parameters and keep the one that minimizes/maximizes your objective function (e.g., SSE)

okay for models with discrete-valued parameters ... when they have continuous values it is *impossible* to try every combination

okay for models with a couple of parameters ... *impossible* for models with dozens or hundreds of parameters
a brief survey of optimization methods

**Brute Force : Grid Search**

Imagine we have a model with 5 parameters ...

And for each parameter we are going to evaluate 100 values of it ...

*How many total evaluations is that?*
a brief survey of optimization methods

Brute Force: Grid Search

imagine we have a model with 5 parameters ...

and for each parameter we are going to evaluate 100 values of it ...

*How many total evaluations is that?*

$100^5$ or $10^{10} = 10,000,000,000$

*Is that a lot? Well it depends.*
a brief survey of optimization methods

**Brute Force : Grid Search**

Imagine we have a model with 5 parameters ...

And for each parameter we are going to evaluate 100 values of it ...

*How many total evaluations is that?*

$100^5$ or $10^{10} = 10,000,000,000$

*Is that a lot? Well it depends.*

If it takes 1 nanosecond ($10^{-9}$ seconds) per evaluation, then it would take only 10 seconds to do all $10^{10}$ evaluations ... but that would have to be a very simple model
Brute Force: Grid Search

Imagine we have a model with 5 parameters ...

And for each parameter we are going to evaluate 100 values of it ...

**How many total evaluations is that?**

\[100^5 \text{ or } 10^{10} = 10,000,000,000\]

*Is that a lot? Well it depends.*

If it takes 100 seconds (\(10^2\) seconds) per evaluation, then it would take nearly 32,000 years to do all the evaluations ...
A brief survey of optimization methods

Brute Force: Grid Search

Imagine we have a model with 5 parameters ...

And for each parameter we are going to evaluate 100 values of it ...

How many total evaluations is that?

$100^5$ or $10^{10} = 10,000,000,000$

Is that a lot? Well it depends.

If it takes 100 seconds ($10^2$ seconds) per evaluation, then it would take nearly 32,000 years to do all the evaluations ... even if we ran this on 1000 processors, it would take 32 years
a brief survey of optimization methods

Calculus

for mathematically simple models and objective functions, you can sometimes optimize using calculus ...
a brief survey of optimization methods

Calculus

derivative of a function is another function that specifies the slope of that function

\[ \frac{dSSE(p)}{dp} \]
a brief survey of optimization methods

Calculus

The derivative of a function is another function that specifies the slope of that function.

$SSE(p)$

$p$
a brief survey of optimization methods

Calculus

The derivative of a function is another function that specifies the slope of that function.

\[ \frac{dSSE(p)}{dp} \]
minimizing or maximizing starts with finding any places on the function where the derivative is zero.
a brief survey of optimization methods

Calculus

\[
\frac{dSSE(p)}{dp} = 0 \text{ can often be done analytically, solving for values of } p
\]
a brief survey of optimization methods

Calculus

\[
\frac{\partial \text{SSE}(p_1)}{\partial p_1} = 0 \quad \text{and} \quad \frac{\partial \text{SSE}(p_2)}{\partial p_2} = 0
\]
a brief survey of optimization methods

Hill-climbing Algorithms
a brief survey of optimization methods

Hill-climbing Algorithms

$SSE(p)$
a brief survey of optimization methods

Hill-climbing Algorithms

$SSE(p)$
a brief survey of optimization methods

Hill-climbing Algorithms

$SSE(p)$
a brief survey of optimization methods

Hill-climbing Algorithms

$SSE(p)$
a brief survey of optimization methods

Hill-climbing Algorithms

$SSE(p)$
a brief survey of optimization methods

Hill-climbing Algorithms

$SSE(p)$
a brief survey of optimization methods

Hill-climbing Algorithms

\[ SSE(p) \]
a brief survey of optimization methods

Hill-climbing Algorithms

$SSE(p)$
a brief survey of optimization methods

Hill-climbing Algorithms

$SSE(p)$
a brief survey of optimization methods

Hill-climbing Algorithms

$SSE(p)$
Hill-climbing Algorithms

Enrico Fermi and Nicholas Metropolis used one of the first digital computers, the Los Alamos Maniac, to determine which values of certain theoretical parameters (phase shifts) best fit experimental data (scattering cross sections). They varied one theoretical parameter at a time by steps of the same magnitude, and when no such increase or decrease in any one parameter further improved the fit to the experimental data, they halved the step size and repeated the process until the steps were deemed sufficiently small. Their simple procedure was slow but sure, and several of us used it on the Avidac computer at the Argonne National Laboratory for adjusting six theoretical parameters to fit the pion-proton scattering data we had gathered using the University of Chicago synchrocyclotron [7].


these techniques only emerged 60 years ago
(Calculus was invented 400 years ago)
a brief survey of optimization methods

Hill-climbing Algorithms

Enrico Fermi and Nicholas Metropolis used one of the first digital computers, the Los Alamos Maniac, to determine which values of certain theoretical parameters (phase shifts) best fit experimental data (scattering cross sections). They varied one theoretical parameter at a time by steps of the same magnitude, and when no such increase or decrease in any one parameter further improved the fit to the experimental data, they halved the step size and repeated the process until the steps were deemed sufficiently small. Their simple procedure was slow but sure, and several of us used it on the Avidac computer at the Argonne National Laboratory for adjusting six theoretical parameters to fit the pion-proton scattering data we had gathered using the University of Chicago synchrocyclotron [7].


these techniques only emerged 60 years ago (Calculus was invented 400 years ago)
a brief survey of optimization methods

Simple Hill Climbing

how many points do you need to evaluate with each step?

2 parameters
a brief survey of optimization methods

Simple Hill Climbing

how many points do you need to evaluate with each step?

<table>
<thead>
<tr>
<th>N parameters</th>
<th>$3^N - 1$ evaluations per step</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 parameters</td>
<td>242 evaluations per step</td>
</tr>
<tr>
<td>10 parameters</td>
<td>59049 evaluations per step</td>
</tr>
</tbody>
</table>

this ends up being inefficient because you can need to take 1000’s of steps
a brief survey of optimization methods

More sophisticated algorithms


Optimization Toolbox

Global Optimization Toolbox

Curve Fitting Toolbox
a brief survey of optimization methods

Nelder-Mead Simplex

MATLAB: fminsearch(…)

http://www.scholarpedia.org/article/Nelder-Mead_algorithm
opt fit = fminsearch(@myfun, p0)

change params to try to decrease fit

params

fit

myfun()
p0 = -2;
[opt fit] = fminsearch(@myfun1, p0);
fprintf('p0 = %5.2f \t optimal p = %5.2f \t f(p) = %6.3f', ...
    p0, opt, fit);

function y = myfun1(x)
    y = -humps(x);
end
p0 = [-2 0];
[opt fit] = fminsearch(@myfun2, p0);

function z = myfun2(x, y)
    z = 3*(1-x(:,1)).^2 .* exp(-(x(:,1)).^2 - (x(:,2)+1).^2) ...
        - 10*(x(:,1)/5 - x(:,1).^3 - x(:,2).^5).* ...
        exp(-x(:,1).^2 - x(:,2).^2) ...
        - 1/3*exp(-(x(:,1)+1).^2 - x(:,2).^2);
end
p0 = [1 0 2 3];
[opt fit] = fminsearch(@myfun3, p0);

function SSE = myfun3(p)
    % calculate model predictions given parameter vector p

    % then calculate SSE comparing predictions to data

    % return SSE as the value to minimize
end

this could be a nonlinear function if you are curve fitting
or
this could be a neural or cognitive model if you are theorizing
p0 = [3 5 1 4 3 1];
[opt fit] = fminsearch(@myfun4, p0);

function lnL = myfun4(p)
    % calculate model predictions given parameter vector p
    % then calculate lnL comparing predictions to data
    % return -lnL as the value to minimize
end

maximizing is simply -minimizing
what values of $\alpha$, $\beta$, $\gamma$, $\lambda$ provide the best account of the observed data?

fitting a psychometric function to observed data

\[
P(\text{correct} \mid x) = \gamma + (1 - \lambda - \gamma)F(x; \alpha, \beta)
\]

\[
F(x; \alpha, \beta) = 1 - \exp(-(x / \beta)^\alpha)
\]
Week12_Psy2.m

minimizing SSE

alpha = 1; beta = 1; gamma = 0.5; lambda = 0.0;
p0 = [alpha beta gamma lambda];
[opt fit] = fminsearch(@psysse3, p0);
alpha = opt(1); beta = opt(2); gamma = opt(3); lambda = opt(4);

function sse = psysse3(p)
a = p(1); b = p(2); g = p(3); l = p(4);
sse = 0.0;
for i=1:length(x)
    sse = sse + (data(i) - mypsyfunc(x(i),a,b,g,l)).^2;
end
end
alpha = 1; beta = 1; gamma = 0.5; lambda = 0.0;
p0 = [alpha beta gamma lambda];
[opt fit] = fminsearch(@psylnl3, p0);
alpha = opt(1); beta = opt(2); gamma = opt(3); lambda = opt(4);

function lnL = psylnl3(p)
a = p(1); b = p(2); g = p(3); l = p(4);
lnL = 0.0;
for i=1:length(x)
    lnL = lnL + ln_nchoosek((raw(1,i)+raw(2,i)),raw(1,i)) + ...
         raw(1,i)*log(mypsyfunc(x(i),a,b,g,l)) + ...
         raw(2,i)*log(1-mypsyfunc(x(i),a,b,g,l));
end
lnL = -lnL; % maximize lnL, same as minimize -lnL
end
maximizing lnL
computationally tractable version of maximizing likelihood

\[ p(data \mid \theta) \]
maximizing \( \ln L \)

**computationally tractable version of maximizing likelihood**

\[
p(data \mid \theta)
\]

\[
p(data \mid \alpha, \beta, \gamma, \lambda)
\]

what values of \( \alpha, \beta, \gamma, \lambda \) provide the best account of the observed data?
maximizing lnL
computationally tractable version of maximizing likelihood

\[ p(data \mid \theta) \]

\[ p(data \mid \alpha, \beta, \gamma, \lambda) \]

\[ p(data \mid \alpha, \beta, \gamma, \lambda) = \prod_{i} p(data_{i} \mid \alpha, \beta, \gamma, \lambda) \]

what values of \( \alpha, \beta, \gamma, \lambda \) provide the best account of the observed data?
maximizing lnL
computationally tractable version of maximizing likelihood

\[ p(data \mid \theta) \]

\[ p(data \mid \alpha, \beta, \gamma, \lambda) \]

\[ p(data \mid \alpha, \beta, \gamma, \lambda) = \prod_i p(data_i \mid \alpha, \beta, \gamma, \lambda) \]

\[ p(data \mid \alpha, \beta, \gamma, \lambda) = \prod_i \binom{N_i}{x_i} p(x_i) (1 - p)^{N_i - x_i} \]

\[ p(data \mid \alpha, \beta, \gamma, \lambda) = \prod_i \frac{N_i!}{x_i! (N_i - x_i)!} p(x_i) (1 - p)^{N_i - x_i} \]
\begin{align*}
\text{maximizing } \ln L & \quad \text{computationally tractable version of maximizing likelihood} \\
\log \text{taking a log of both sides turns this into a sum} \\
p(data \mid \alpha, \beta, \gamma, \lambda) &= \prod_i \frac{N_i!}{x_i!(N_i-x_i)!} p^{x_i} (1-p)^{N_i-x_i} \\
p(\text{correct} \mid x) &= \gamma + (1-\lambda-\gamma)F(x; \alpha, \beta) \\
F(x; \alpha, \beta) &= 1 - \exp(-\frac{x}{\beta})^\alpha
\end{align*}
What is a simplex?

0 dimensions  point  1 vertex
1 dimension  line  2 vertices
2 dimensions  triangle  3 vertices
3 dimensions  tetrahedron  4 vertices
4 dimensions  pentachoron  5 vertices

basically just a generalization of a triangle to $N$ dimensions
contract
shrink
high point

simplex at start of procedure

low point

(a)

reflection

(b)

reflection and expansion

(c)

contraction

(d)

contraction in all directions
other optimization variants in Matlab

<table>
<thead>
<tr>
<th>Constraint Type</th>
<th>Linear</th>
<th>Quadratic</th>
<th>Least Squares</th>
<th>Smooth nonlinear</th>
<th>Nonsmooth</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>n/a (f = const, or min = $-\infty$)</td>
<td>quadprog, Theory, Examples</td>
<td>, lsqcurvefit, lsqnonlin, Theory, Examples</td>
<td>fminsearch, fminunc, Theory, Examples</td>
<td>fminsearch, *</td>
</tr>
<tr>
<td>Bound</td>
<td>linprog, Theory, Examples</td>
<td>quadprog, Theory, Examples</td>
<td>1sqcurevfit, 1sqlin, 1sqnonlin, 1sqnonneg, Theory, Examples</td>
<td>fminbnd, fmincon, fseminf, Theory, Examples</td>
<td>fminbnd, *</td>
</tr>
<tr>
<td>Linear</td>
<td>linprog, Theory, Examples</td>
<td>quadprog, Theory, Examples</td>
<td>1sqlin, Theory, Examples</td>
<td>fmincon, fseminf, Theory, Examples</td>
<td>*</td>
</tr>
<tr>
<td>General smooth</td>
<td>fmincon, Theory, Examples</td>
<td>fmincon, Theory, Examples</td>
<td>fmincon, Theory, Examples</td>
<td>fmincon, fseminf, Theory, Examples</td>
<td>*</td>
</tr>
<tr>
<td>Discrete</td>
<td>bintprog, * Theory, Examples</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>
Search Algorithms

**Genetic Algorithms**: parameters are like genes ... combinations of parameters that lead to better fits are allowed to “reproduce” (in a sense it’s more like eugenics) ... by combining sets of their parameter values ... the children “inherit” the parameters (genes) from best-fitting (surviving) parents ...
a brief survey of optimization methods

Search Algorithms

**Genetic Algorithms**: parameters are like genes ... combinations of parameters that lead to better fits are allowed to “reproduce” (in a sense it’s more like eugenics) ... by combining sets of their parameter values ... the children “inherit” the parameters (genes) from best-fitting (surviving) parents ...

**Simulated Annealing**: start by searching the parameter space randomly ... eventually, the probability of trying a new location in parameters space is a function of the fit value, but you can move probabilistically to a worse-fitting location ... but as the “temperature” is reduced, the probability of such a move decreases ... the system “anneals”
a brief survey of optimization methods

Genetic Algorithms and Simulated Annealing can be slow but are particularly good at problems with multiple “local minima”
hill-climbing
hill-climbing
hill-climbing
hill-climbing
genetic algorithm
genetic algorithm
genetic algorithm
genetic algorithm
### Generation 1

<table>
<thead>
<tr>
<th>SSE</th>
<th>.12</th>
<th>.15</th>
<th>.99</th>
<th>.08</th>
<th>1.4</th>
<th>1.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>.143</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SSE</th>
<th>.88</th>
<th>.15</th>
<th>.99</th>
<th>.11</th>
<th>2.4</th>
<th>1.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>.043</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SSE</th>
<th>.33</th>
<th>.52</th>
<th>.11</th>
<th>.87</th>
<th>1.5</th>
<th>9.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>.198</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SSE</th>
<th>.08</th>
<th>.53</th>
<th>.73</th>
<th>.11</th>
<th>2.4</th>
<th>4.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>.298</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SSE</th>
<th>.54</th>
<th>.72</th>
<th>.43</th>
<th>.32</th>
<th>1.2</th>
<th>8.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>.088</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Generation 1

<table>
<thead>
<tr>
<th>SSE</th>
<th>.88</th>
<th>.15</th>
<th>.99</th>
<th>.11</th>
<th>2.4</th>
<th>1.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSE</td>
<td>.54</td>
<td>.72</td>
<td>.43</td>
<td>.32</td>
<td>1.2</td>
<td>8.2</td>
</tr>
<tr>
<td>SSE</td>
<td>.12</td>
<td>.15</td>
<td>.99</td>
<td>.08</td>
<td>1.4</td>
<td>1.5</td>
</tr>
<tr>
<td>SSE</td>
<td>.33</td>
<td>.52</td>
<td>.11</td>
<td>.87</td>
<td>1.5</td>
<td>9.3</td>
</tr>
<tr>
<td>SSE</td>
<td>.08</td>
<td>.53</td>
<td>.73</td>
<td>.11</td>
<td>2.4</td>
<td>4.2</td>
</tr>
</tbody>
</table>

...
### Generation 1

<table>
<thead>
<tr>
<th>SSE</th>
<th>0.043</th>
<th>0.088</th>
<th>0.143</th>
<th>0.198</th>
<th>0.298</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>.88</td>
<td>.15</td>
<td>.99</td>
<td>.11</td>
<td>2.4</td>
</tr>
<tr>
<td></td>
<td>.54</td>
<td>.72</td>
<td>.43</td>
<td>.32</td>
<td>1.2</td>
</tr>
<tr>
<td></td>
<td>.12</td>
<td>.15</td>
<td>.99</td>
<td>.08</td>
<td>1.4</td>
</tr>
<tr>
<td></td>
<td>.03</td>
<td>.52</td>
<td>.11</td>
<td>.87</td>
<td>1.5</td>
</tr>
<tr>
<td></td>
<td>.06</td>
<td>.53</td>
<td>.73</td>
<td>.11</td>
<td>.44</td>
</tr>
</tbody>
</table>
### Generation 1

<table>
<thead>
<tr>
<th>SSE</th>
<th>0.43</th>
<th>0.88</th>
<th>0.15</th>
<th>0.99</th>
<th>0.11</th>
<th>2.4</th>
<th>1.3</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>SSE</th>
<th>0.88</th>
<th>0.15</th>
<th>0.99</th>
<th>0.11</th>
<th>2.4</th>
<th>1.3</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>SSE</th>
<th>0.143</th>
<th>0.33</th>
<th>0.52</th>
<th>0.72</th>
<th>0.43</th>
<th>0.32</th>
<th>1.2</th>
<th>8.2</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>SSE</th>
<th>0.198</th>
<th>0.54</th>
<th>0.72</th>
<th>0.43</th>
<th>0.32</th>
<th>1.2</th>
<th>8.2</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>SSE</th>
<th>0.298</th>
<th>0.53</th>
<th>0.73</th>
<th>0.31</th>
<th>0.4</th>
<th>4.2</th>
</tr>
</thead>
</table>

### Generation 2

<table>
<thead>
<tr>
<th>SSE</th>
<th>0.88</th>
<th>0.15</th>
<th>0.99</th>
<th>0.11</th>
<th>2.4</th>
<th>1.3</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>SSE</th>
<th>0.88</th>
<th>0.15</th>
<th>0.99</th>
<th>0.11</th>
<th>2.4</th>
<th>1.3</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>SSE</th>
<th>0.198</th>
<th>0.54</th>
<th>0.72</th>
<th>0.43</th>
<th>0.32</th>
<th>1.2</th>
<th>8.2</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>SSE</th>
<th>0.298</th>
<th>0.53</th>
<th>0.73</th>
<th>0.31</th>
<th>0.4</th>
<th>4.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generation 1</td>
<td>Generation 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSE=.043</td>
<td>.88 .15 .99 .11 2.4 1.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSE=.088</td>
<td>.54 .72 .43 .32 1.2 8.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSE=.143</td>
<td>.12 .15 .99 .08 1.4 1.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSE=.298</td>
<td>.06 .53 .73 .11 0.4 4.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.88 .15 .99 .32 1.2 8.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.12 .15 .99 .11 2.4 1.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.30 .52 .11 .87 .5 9.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.06 .53 .73 .11 0.4 4.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- 
- 
-
### Generation 1

<table>
<thead>
<tr>
<th>SSE</th>
<th>0.43</th>
<th>0.88</th>
<th>0.15</th>
<th>0.99</th>
<th>0.11</th>
<th>2.4</th>
<th>1.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSE</td>
<td>0.88</td>
<td>0.54</td>
<td>0.72</td>
<td>0.43</td>
<td>0.32</td>
<td>1.2</td>
<td>8.2</td>
</tr>
<tr>
<td>SSE</td>
<td>0.143</td>
<td>0.12</td>
<td>0.15</td>
<td>0.99</td>
<td>0.08</td>
<td>1.4</td>
<td>1.5</td>
</tr>
<tr>
<td>SSE</td>
<td>0.198</td>
<td>0.08</td>
<td>0.53</td>
<td>0.73</td>
<td>0.11</td>
<td>0.4</td>
<td>4.2</td>
</tr>
</tbody>
</table>

### Generation 2

<table>
<thead>
<tr>
<th>SSE</th>
<th>0.43</th>
<th>0.88</th>
<th>0.15</th>
<th>0.99</th>
<th>0.11</th>
<th>2.4</th>
<th>1.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSE</td>
<td>0.88</td>
<td>0.12</td>
<td>0.15</td>
<td>0.99</td>
<td>0.32</td>
<td>1.2</td>
<td>8.2</td>
</tr>
<tr>
<td>SSE</td>
<td>0.143</td>
<td>0.88</td>
<td>0.72</td>
<td>0.99</td>
<td>0.32</td>
<td>2.4</td>
<td>1.3</td>
</tr>
<tr>
<td>SSE</td>
<td>0.198</td>
<td>0.08</td>
<td>0.53</td>
<td>0.73</td>
<td>0.11</td>
<td>0.4</td>
<td>4.2</td>
</tr>
<tr>
<td>Generation 1</td>
<td>Generation 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSE=.043</td>
<td>.88 .15 .99 .11 2.4 1.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSE=.088</td>
<td>.54 .72 .43 .32 1.2 8.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSE=.143</td>
<td>.12 .15 .99 .08 1.4 1.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSE=.198</td>
<td>.54 .72 .99 .08 1.4 8.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSE=.298</td>
<td>.08 .53 .73 .11 0.4 4.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.88 .15 .99 .11 2.4 1.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.12 .15 .99 .32 1.2 8.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.88 .12 .15 .99 .11 2.4 1.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.12 .15 .32 1.2 8.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.12 .15 .86 .08 1.4 1.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Simulated Annealing


http://www.mathworks.com/discovery/simulated-annealing.html
functions with many local minima
Simulated Annealing

typically considers multiple candidate parameter vectors

move each vector in some random direction, always accept a new candidate parameter vector if it gives a better fit

but also accept a new candidate parameter vector with probability $P$ if it gives a WORSE fit:

$$P = \frac{1}{1 + \exp(\Delta \text{fit} / T)}$$

$T$ is the “temperature”, which decreases according to a schedule:

- $T$ starts at $\infty$, so $P$ starts at 1 (completely random)
- $T$ goes to 0, so $P$ goes to 0 (pure hill climbing)

depending on cooling schedule, simulated annealing can take far longer than a basic hill climbing algorithm
Bayesian approaches

\[ p(\theta \mid data) = \frac{p(data \mid \theta)p(\theta)}{p(data)} \]
Bayesian approaches

\[
p(\theta \mid data) = \frac{p(data \mid \theta)p(\theta)}{p(data)}
\]

\[
p(\theta \mid data) = \frac{p(data \mid \theta)p(\theta)}{\int p(data \mid \theta)p(\theta)\,d\theta}
\]

\[
p(\theta_1, \theta_2, \ldots, \theta_n \mid data) = \frac{\int \cdots \int p(data \mid \theta_1, \theta_2, \ldots, \theta_n)p(\theta_1, \theta_2, \ldots, \theta_n)\,d\theta_1\,d\theta_2\ldots d\theta_n}{\int \cdots \int p(data \mid \theta_1, \theta_2, \ldots, \theta_n)p(\theta_1, \theta_2, \ldots, \theta_n)\,d\theta_1\,d\theta_2\ldots d\theta_n}
\]

in lieu of computationally intensive optimization, you instead need computationally intensive algorithms to deal with this integral