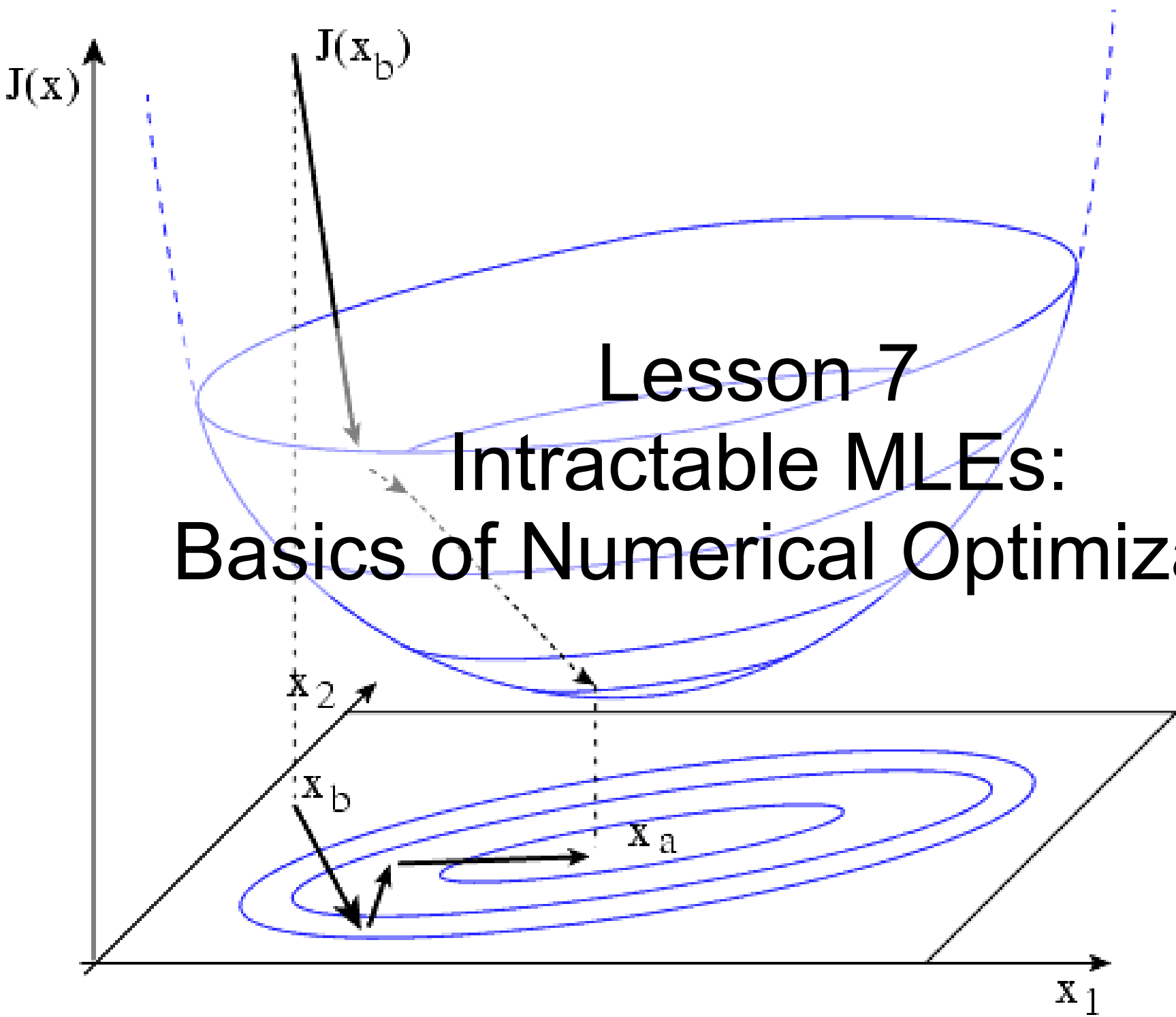


Lesson 7

Intractable MLEs: Basics of Numerical Optimization



Maximum Likelihood

Write down the Likelihood

Take the log

Take the derivatives w.r.t. each parameter

Set equal to 0 and solve for parameter θ

Maximum Likelihood Estimate (MLE)

Linear Regression

$$a_0 = \bar{y} - a_1 \bar{x}$$

$$a_1 = \frac{\overline{xy} - \bar{x} \bar{y} + a_1 \bar{x}^2}{\overline{x^2}}$$

$$\sigma_{ML}^2 = \frac{1}{n} \sum_{i=1}^n (y_i - a_0 - a_1 x_i)^2$$

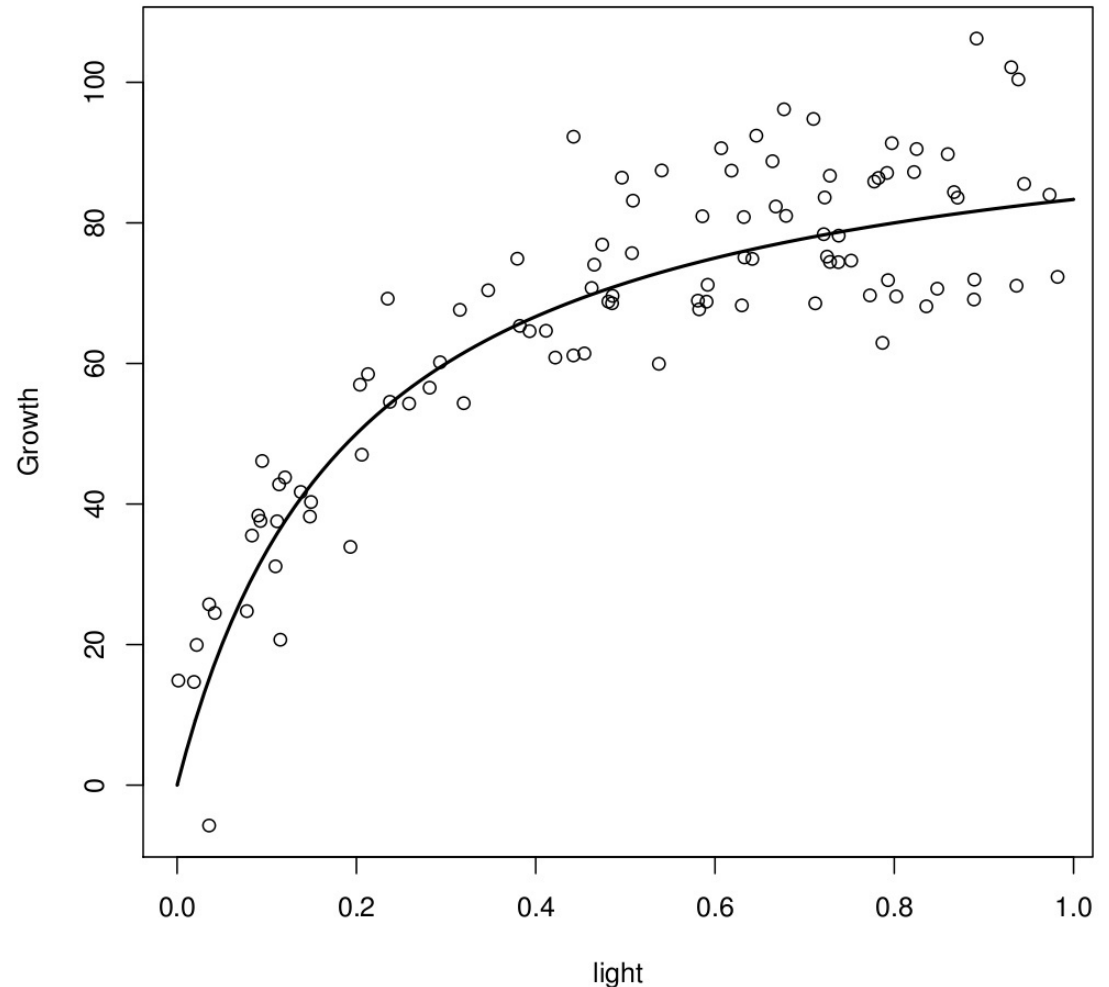
Nonlinear Model Fitting

$$Growth_i = \frac{\beta_1 light_i}{\beta_2 + light_i} + \epsilon_i$$

$$\epsilon_i \sim N(0, \sigma^2)$$

How do we fit this??

Michaelis-Menten

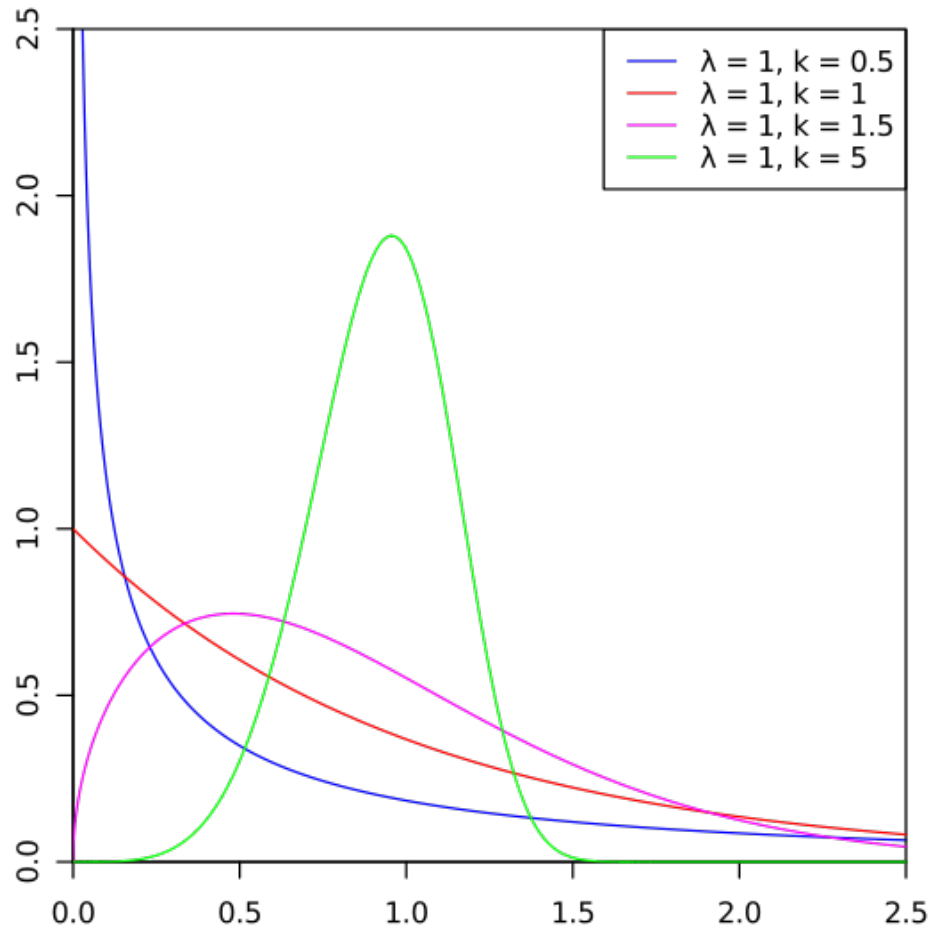


The Problem

- Analytical optimization requires setting the derivative = 0 and solving for the parameter
- For complicated problems a closed-form analytical solution may not exist or may be difficult to solve for.
 - Nonlinear models
 - Multi-parameter models
 - Multiple Constraints

Example – Weibull distribution

$$Weibull(x|c, \lambda) = \left(\frac{c}{\lambda}\right) \left(\frac{x}{\lambda}\right)^{(c-1)} \exp\left(-\left(x/\lambda\right)^c\right)$$



Example – Weibull distribution

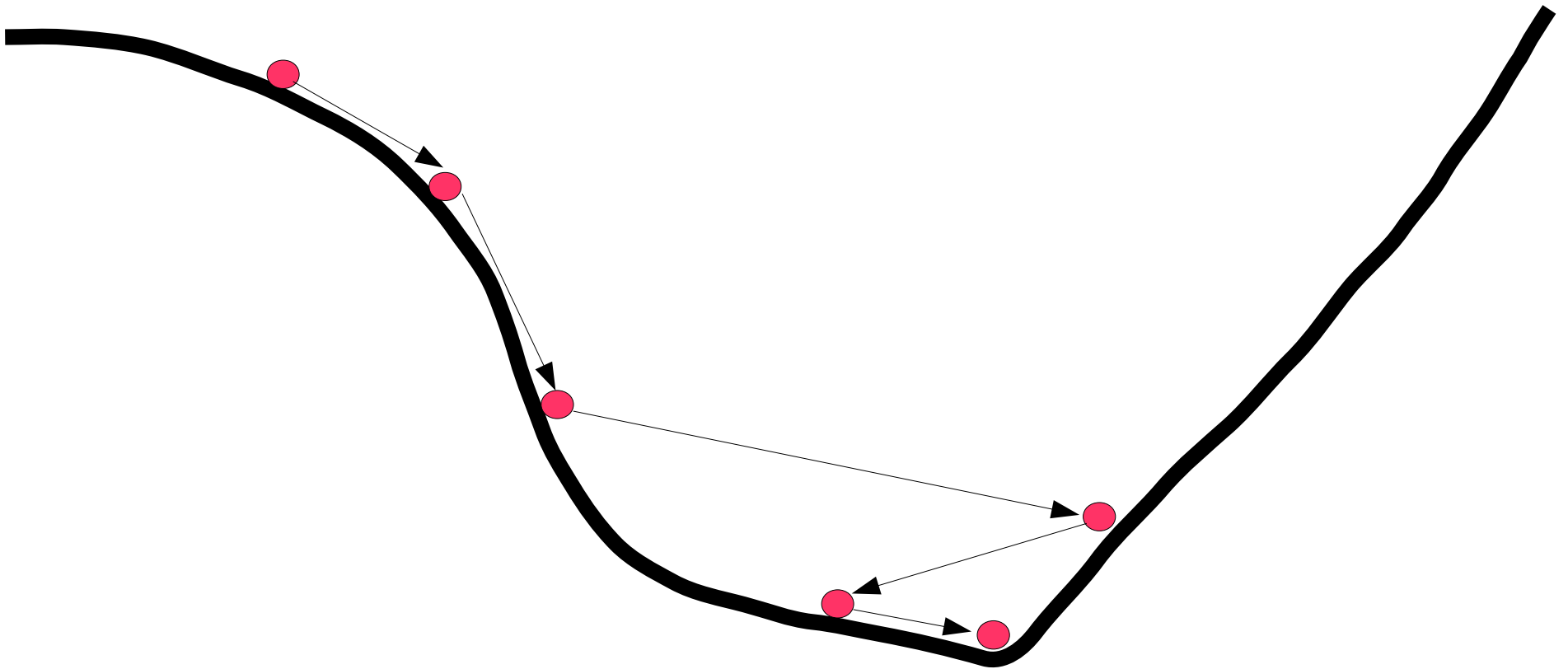
$$\text{Weibull}(x|c, \lambda) = \left(\frac{c}{\lambda}\right) \left(\frac{x}{\lambda}\right)^{(c-1)} \exp\left(-\left(x/\lambda\right)^c\right)$$

$$\ln L = \ln c - \ln \lambda + (c-1) [\ln x - \ln \lambda] - (x/\lambda)^c$$

$$\frac{\partial \ln L}{\partial c} = \frac{1}{c} + \ln x - \ln \lambda - \ln c (x/\lambda)^c = 0$$

Not possible to solve for c analytically

The Solution: Numerical Optimization

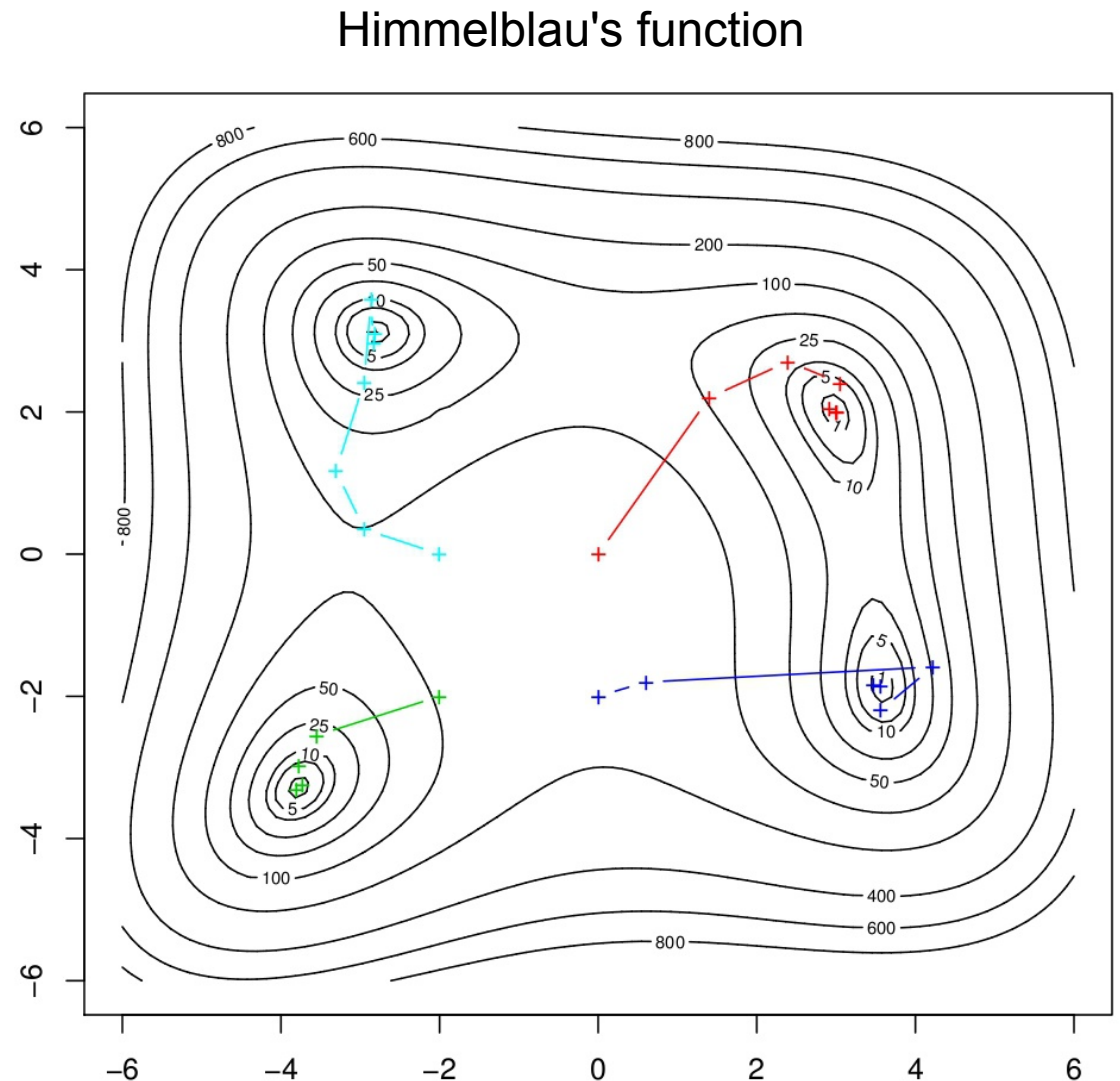


Optimization Algorithm

- 1) Start from some initial parameter value
- 2) Evaluate the likelihood
- 3) Propose a new parameter value
- 4) Evaluate the new likelihood
- 5) Decide whether or not to accept the new value
- 6) Repeat 3-5 until you can't find a better value

1) Start from some initial parameter value

- Far values take a long time to converge
- Can get stuck in local minima
- Want to try multiple initial values



3) Propose a new parameter value

- Diversity of approaches
- Deterministic
 - Gradient decent
 - Nelder-Mead
- Stochastic
 - Genetic algorithms
 - Simulated annealing
- Curvature
 - Newton's method

5) Decide whether or not to accept the new value

- Almost all algorithms accept a new step if it has a lower negative log likelihood
- What if the step has a higher value?
 - Always reject a worse step
 - Efficient
 - More susceptible to local minima
 - Occasionally accept a worse step with some probability
 - Slower convergence
 - Less likely to get caught in local minima

6) Repeat until you can't find a better value

- “Stopping condition”
- Improvement in estimate is below some threshold (gradient)
- Step size is below some threshold
- Failure to converge?
 - Too many steps taken
 - Converged to boundary condition
 - Step size too big (divergence)

Reasons for working with negative log likelihoods

- Log
 - Numerical precision
 - Likelihood is degenerate if a probability = 0
 - In R, taking the log of the returned value is less precise than using “log = TRUE”
- Negative
 - Most numerical optimization routines set up for minimization
- Deviance = $-2 \log(L)$
 - Used in model selection and CI

Limits to Numerical Methods

- Accuracy
- Generality / Understanding
- Local Minima
- Dimensionality
 - Difficult to explore high dimensional parameter space

Optimization Algorithm

- 1) Start from some initial parameter value
- 2) Evaluate the likelihood
- 3) Propose a new parameter value
- 4) Evaluate the new likelihood
- 5) Decide whether or not to accept the new value
- 6) Repeat 3-5 until you can't find a better value

MLE Optimization Algorithm

- 1) Start from some initial parameter value
- 2) Evaluate **the likelihood**
- 3) Propose a new parameter value
- 4) Evaluate **the new likelihood**
- 5) Decide whether or not to accept the new value
- 6) Repeat 3-5 until you can't find a better value

Simple Example: $y = N(a, \sigma^2)$

Name of the function

Parameter vector

```
lkNormal <- function (beta,y){
```

```
-sum(
```

Normal density

```
dnorm(y,
```

Response data

```
beta[1],
```

a

```
beta[2],
```

Standard deviation

```
log=TRUE)
```

Returns the log likelihood

```
)
```

```
}
```

Simple Example: $y = N(a, \sigma^2)$

```
> y = rnorm(10,3.5,2)
```

```
> mean(y)
[1] 3.897316
```

```
> sd(y)
[1] 2.4483
```

```
> lkNormal(c(0,1),y)
[1] 112.1085
```

```
> lkNormal(c(3.5,2),y)
[1] 23.06163
```

```
> optim(c(0,1),lkNormal,y=y)
$par
[1] 3.897709 2.323170
```

```
$value
[1] 22.61652
```

```
$counts
function gradient
      87      NA
```

```
$convergence
[1] 0
```

Nonlinear example

$$Growth_i = \frac{\beta_1 light_i}{\beta_2 + light_i} + \epsilon_i$$

$$\epsilon_i \sim N(0, \sigma^2)$$

“Pseudodata”

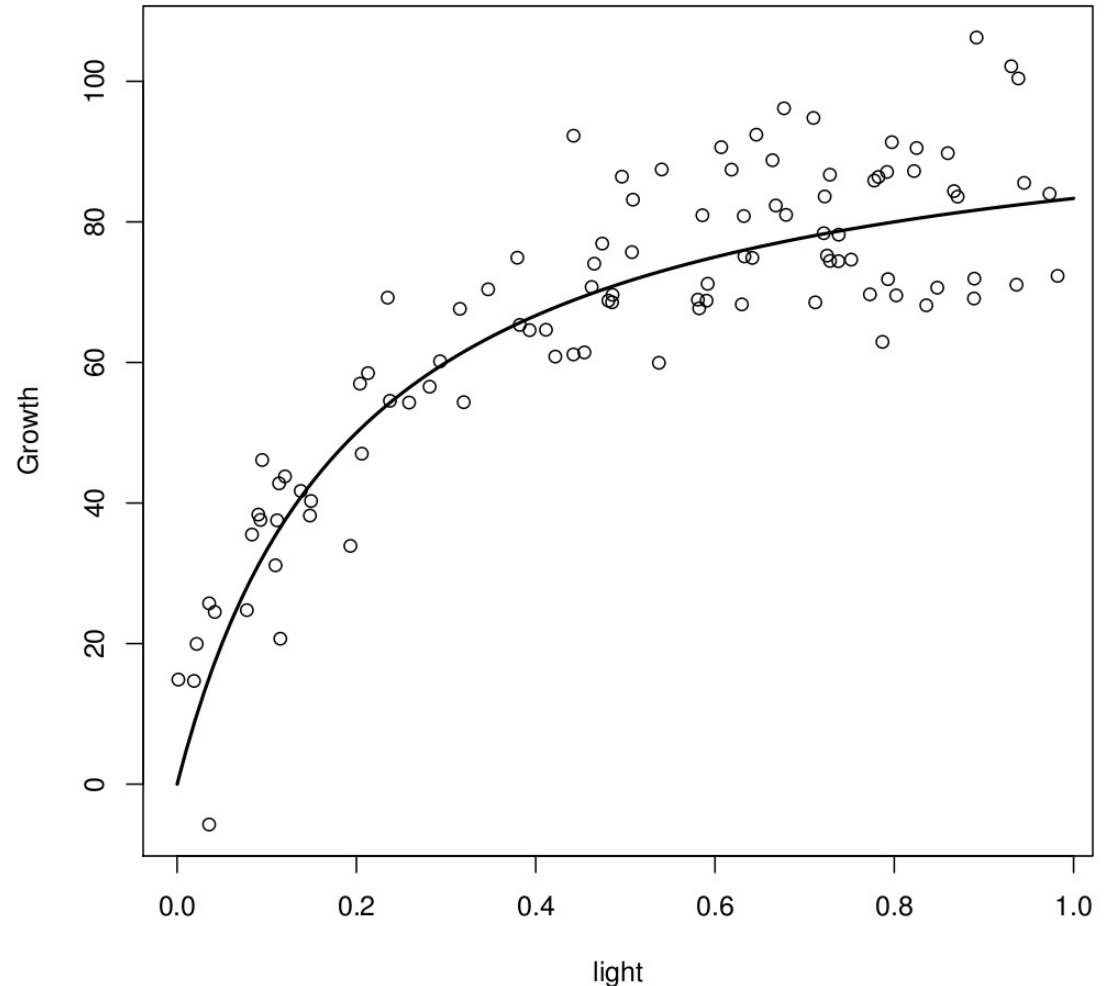
$$\beta_1 = 100$$

$$\beta_2 = 0.2$$

$$\sigma = 10$$

$$n = 100$$

Michaelis-Menten



Michaelis-Menten negative log likelihood

Name of the function

Parameter vector

```
lkMM <- function (beta){  
  -sum(  
    dnorm(y,  
          beta[1]*x/(beta[2]+x),  
          beta[3],  
          log=TRUE)  
  )  
}
```

Normal density

Response data (growth)

Michaelis-Menten

Standard deviation

Returns the log likelihood

Michaelis-Menten negative log likelihood

```
lkMM <- function (beta){
```

```
-sum(
```

```
dnorm(y,
```

```
beta[1]*x/(beta[2]+x),
```

```
beta[3],
```

```
log=TRUE)
```

```
)
```

```
}
```

$-\sum$

log

N

y

$\frac{\beta_1 x}{\beta_2 + x}$

σ^2

Optimization function

```
opt = optim(  
  c(max(y)*0.9,0.5,sd(y)/2),  
  lkMM,  
  method="L-BFGS-B",  
  lower=c(0,0,0),  
  upper=c(max(y)*2,1,sd(y)*1.1)  
)
```

Initial conditions

neg log likelihood function

Name of algorithm

Lower bound

Upper Bound

Optimization Output

```
> opt
```

```
$par
```

```
[1] 101.2937369  0.1916526  9.3997657
```

```
$value
```

```
[1] 365.9635
```

```
$counts
```

```
function gradient
```

```
    48    48
```

```
$convergence
```

```
[1] 0
```

```
$message
```

```
[1] "CONVERGENCE: REL_REDUCTION_OF_F <=
FACTR*EPSMCH"
```


“Pseudo”

$$n_1 = 100$$

$$n_2 = 0.2$$

$$O = 10$$

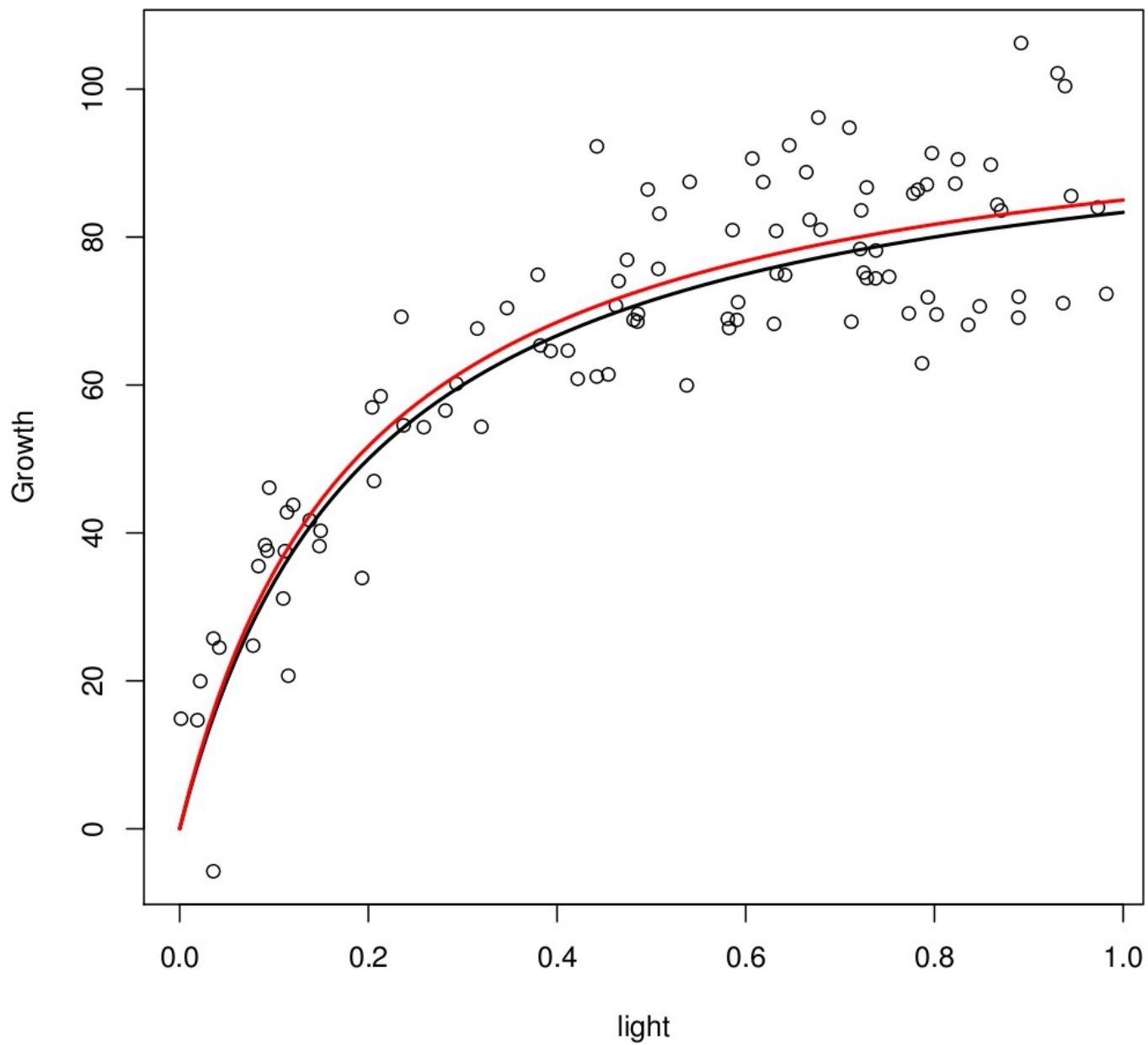
$$n = 100$$

Fit

$$101.3$$

$$0.192$$

$$9.40$$



More generally...

- Can fit any 'black box' function
- Can use any distribution
 - No Normality assumption
- Can model the variance explicitly
 - No equal variance assumption
- Various techniques for estimating uncertainties, CI, etc.
- Likelihood is backbone of more advanced approaches (e.g. Bayesian stats)

A few last thoughts on MLE

- More difficult as model complexity increases
- More challenging when not independent
 $P(x_1, x_2, x_3) \neq P(x_1)P(x_2)P(x_3)$
- Require additional assumptions/computation to estimate CI
- Analysis occurs in a “vacuum”
 - No way to update previous analysis