Math Camp

Justin Grimmer

Associate Professor
Department of Political Science
University of Chicago

September 7th, 2017

Multivariate Optimization

Optimizing multivariate functions

- Parameters $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_n)$ such that $f(\boldsymbol{\beta}|\boldsymbol{X}, \boldsymbol{Y})$ is maximized
- Policy $\boldsymbol{x} \in \Re^n$ that maximizes $U(\boldsymbol{x})$
- Weights $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_K)$ such that a weighted average of forecasts $\boldsymbol{f} = (f_1, f_2, \ldots, f_k)$ have minimum loss

$$\min_{\boldsymbol{\pi}} \quad = \quad -(\sum_{j=1}^{K} \pi_j f_j - y)^2$$

Today we'll describe analytic and computational approaches to optimization

- Analytic recipe for optimization
- Computational optimization
    - Multivariate Newton-Raphson
    - BFGS
    - Approximate Optimization: k-means

Multivariate Optimization

Definition

Let $\boldsymbol{x} \in \Re^n$ and let $\delta > 0$. Define a neighborhood of $\boldsymbol{x}$, $B(\boldsymbol{x}, \delta)$, as the set of points such that,

$$B(\boldsymbol{x}, \delta) \quad = \quad \{\boldsymbol{y} \in \Re^n : ||\boldsymbol{x} - \boldsymbol{y}|| < \delta\}$$

Definition

Suppose $f : X \rightarrow \Re$ with $X \subset \Re^n$. A vector $\boldsymbol{x}^* \in X$ is a global maximum if , for all other $\boldsymbol{x} \in X$

$$f(\boldsymbol{x}^*) \quad > \quad f(\boldsymbol{x})$$

A vector $\boldsymbol{x}^{local}$ is a local maximum if there is a neighborhood around $\boldsymbol{x}^{local}$, $Q \subset X$ such that, for all $x \in Q$,

$$f(\boldsymbol{x}^{local}) \quad > \quad f(\boldsymbol{x})$$

Multivariate Optimization

Definition

A set $X \subset R^n$ is compact if it is closed and bounded

Theorem

Multivariate Extreme Value Theorem Suppose $f : X \rightarrow \Re$ be continuous and $X \subset \Re^n$ and X compact. Then f takes on its maximum and minimum values on X.

We're going to come up with the multivariate equivalent of the first order and second order conditions now

Gradient

Definition

Suppose $f : X \rightarrow \Re^n$ with $X \subset \Re^1$ is a differentiable function. Define the gradient vector of f at $\boldsymbol{x}_0$, $\nabla f(\boldsymbol{x}_0)$ as,

$$\nabla f(\boldsymbol{x}_0) = \left(\frac{\partial f(\boldsymbol{x}_0)}{\partial x_1}, \frac{\partial f(\boldsymbol{x}_0)}{\partial x_2}, \frac{\partial f(\boldsymbol{x}_0)}{\partial x_3}, \ldots, \frac{\partial f(\boldsymbol{x}_0)}{\partial x_n}\right)$$

Gradient First Order Condition

Theorem

Suppose $f : X \rightarrow \Re^1$, $X \subset \Re^n$. Suppose $\boldsymbol{a} \in X$ is a local extremum. Then,

$$\begin{aligned} \nabla f(\boldsymbol{a}) &= \mathbf{0} \\ &= (0, 0, \ldots, 0) \end{aligned}$$

- Proof (intuition): same as one dimensional case (left-hand, right hand), just do it dimension by dimension
- Critical Values:
  1) Maximum
  2) Minimum
  3) Saddle point
- Second Derivative Test!

Second Order Conditions: Hessian

Definition

Suppose $f : X \rightarrow \Re^1$, $X \subset \Re^n$, with f a twice differentiable function. We will define the Hessian matrix as the matrix of second derivatives at $\boldsymbol{x}^* \in X$,

$$\boldsymbol{H}(f)(\boldsymbol{x}^*) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1}(\boldsymbol{x}^*) & \frac{\partial^2 f}{\partial x_1 \partial x_2}(\boldsymbol{x}^*) & \ldots & \frac{\partial^2 f}{\partial x_1 \partial x_n}(\boldsymbol{x}^*) \\ \frac{\partial^2 f}{\partial x_2 \partial x_1}(\boldsymbol{x}^*) & \frac{\partial^2 f}{\partial x_2 \partial x_2}(\boldsymbol{x}^*) & \ldots & \frac{\partial^2 f}{\partial x_2 \partial x_n}(\boldsymbol{x}^*) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1}(\boldsymbol{x}^*) & \frac{\partial^2 f}{\partial x_n \partial x_2}(\boldsymbol{x}^*) & \ldots & \frac{\partial^2 f}{\partial x_n \partial x_n}(\boldsymbol{x}^*) \end{pmatrix}$$

General test $\rightsquigarrow$ Two Dimensional Test $\rightsquigarrow$ Example

Hessians

Definition

Consider $n \times n$ matrix $\boldsymbol{A}$. If, for all $\boldsymbol{x} \in \Re^n$ where $\boldsymbol{x} \neq 0$:

$$\begin{aligned} \boldsymbol{x}^{'}\boldsymbol{A}\boldsymbol{x} &> 0 \; \boldsymbol{A} \text{ is positive definite} \\ \boldsymbol{x}^{'}\boldsymbol{A}\boldsymbol{x} &< 0 \; \boldsymbol{A} \text{ is negative definite} \end{aligned}$$

If $\boldsymbol{x}^{'}\boldsymbol{A}\boldsymbol{x} > 0$ for some $\boldsymbol{x}$ and $\boldsymbol{x}^{'}\boldsymbol{A}\boldsymbol{x} < 0$ for other $\boldsymbol{x}$, then we say $\boldsymbol{A}$ is indefinite

Approximating functions and second order conditions

Theorem

Taylor's Theorem *Suppose* $f : \Re \rightarrow \Re$, $f(x)$ *is infinitely differentiable function. Then, the taylor expansion of* $f(x)$ *around* a *is given by*

$$
\begin{aligned}
f(x) &= f(a) + \frac{f^{'}(a)}{1!}(x-a) + \frac{f^{''}(a)}{2!}(x-a)^2 + \frac{f^{'''}(a)}{3!}(x-a)^3 + \ldots \\
f(x) &= \sum_{n=0}^{\infty} \frac{f^{n}(a)}{n!}(x-a)^n
\end{aligned}
$$

Example Function

Suppose $a = 0$ and $f(x) = e^x$. Then,

$$\begin{aligned} f^{'}(x) &= e^x \\ f^{''}(x) &= e^x \\ \vdots &\vdots \vdots \\ f^n(x) &= e^x \end{aligned}$$

This implies

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} \ldots + \frac{x^n}{n!} + \ldots$$

Multivariate Taylor's Theorem

Theorem

Suppose $f : \Re^n \to \Re$ is a three-times continously differentiable function, then around $\boldsymbol{a} \in \Re^n$,

$$f(\boldsymbol{x}) \quad = \quad f(\boldsymbol{a}) + \nabla f(\boldsymbol{a})(\boldsymbol{x} - \boldsymbol{a}) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{a})^{'}\mathbf{H}(f)(\boldsymbol{a})(\boldsymbol{x} - \boldsymbol{a}) + R(\boldsymbol{a}, \boldsymbol{x})$$

where $\frac{R(\boldsymbol{x},\boldsymbol{a})}{||\boldsymbol{x}-\boldsymbol{a}||^2} \to 0$ *as* $\boldsymbol{x} \to \boldsymbol{a}$

Intuition for Quadratic Form

Suppose $\boldsymbol{x}^*$ is some critical value,

$$f(\boldsymbol{x}) \quad = \quad f(\boldsymbol{x}^*) + \nabla f(\boldsymbol{x}^*)(\boldsymbol{x} - \boldsymbol{x}^*) + (\boldsymbol{x} - \frac{1}{2}\boldsymbol{x}^*)\mathbf{H}(f)(x^*)(\boldsymbol{x} - \boldsymbol{x}^*) + R(\boldsymbol{x}^*, \boldsymbol{x}$$

$$f(\boldsymbol{x}) - f(\boldsymbol{x}^*) \quad = \quad 0(\boldsymbol{x} - \boldsymbol{x}^*) + (\boldsymbol{x} - \frac{1}{2}\boldsymbol{x}^*)\mathbf{H}(f)(\boldsymbol{x}^*)(\boldsymbol{x} - \boldsymbol{x}^*) + R(\boldsymbol{x}^*, \boldsymbol{x})$$

For $\boldsymbol{x}$ near $\boldsymbol{x}^*$, $R(\boldsymbol{x}^*, \boldsymbol{x}) \approx 0$

$\boldsymbol{H}(f)(\boldsymbol{x}^*)$ positive definite $\rightarrow f(\boldsymbol{x}) > f(\boldsymbol{x}^*) \rightarrow$ local minimum
$\boldsymbol{H}(f)(\boldsymbol{x}^*)$ negative definite $\rightarrow f(\boldsymbol{x}) < f(\boldsymbol{x}^*) \rightarrow$ local maximum

Theorem

Second Derivative Test

- If $\boldsymbol{H}(f)(\boldsymbol{a})$ is *positive definite* then $\boldsymbol{a}$ is a local minimum
- If $\boldsymbol{H}(f)(\boldsymbol{a})$ is *negative definite* then $\boldsymbol{a}$ is a local maximum
- If $\boldsymbol{H}(f)(\boldsymbol{a})$ is *indefinite* then $\boldsymbol{a}$ is a saddle point

Second Derivative Test

Many ways to assess definiteness $\rightsquigarrow$ use determinant

Theorem

Two Dimensional, Second Derivative Test. Suppose $f : X \rightarrow \Re$ with $X \subset \Re^2$ and f twice differentiable. Write the Hessian of f at a critical value $\boldsymbol{a}$,

$$\boldsymbol{H}(f)(\boldsymbol{a}) \quad = \quad \begin{pmatrix} A & B \\ B & C \end{pmatrix}$$

Then, we can conduct the second derivative test as:

- $AC - B^2 > 0$ and $A > 0 \rightsquigarrow$ *positive definite* $\rightsquigarrow$ $\boldsymbol{a}$ *is a local minimum*
- $AC - B^2 > 0$ and $A < 0 \rightsquigarrow$ *negative definite* $\rightsquigarrow$ $\boldsymbol{a}$ *is a local maximum*
- $AC - B^2 < 0 \rightsquigarrow$ *indefinite* $\rightsquigarrow$ *saddle point*
- $AC - B^2 = 0$ *inconclusive*

Multivariate Recipe

1) Calculate gradient
2) Set equal to zero, solve system of equations
3) Calculate Hessian
4) Assess Hessian at critical values
5) Boundary values? (if relevant)

Example 1: A Simple Optimization Problem

Suppose $f : \Re^2 \rightarrow \Re$ with

$$f(x_1, x_2) \quad = \quad 3(x_1 + 2)^2 + 4(x_2 + 4)^2$$

Calculate gradient

$$\begin{aligned} \nabla f(\boldsymbol{x}) &= (6x_1 + 12, 8x_2 + 32) \\ \mathbf{0} &= (6x_1^* + 12, 8x_2^* + 32) \end{aligned}$$

We now solve the system of equations to yield $x_1^* = -2$ and $x_2^* = -4$

Example 1: A Simple Optimization Problem

$$\mathbf{H}(f)(\boldsymbol{x}^*) = \begin{pmatrix} 6 & 0 \\ 0 & 8 \end{pmatrix}$$

$\det(\mathbf{H}(f)(\boldsymbol{x}^*)) = 48$ and $6 > 0$ so $\mathbf{H}(f)(\boldsymbol{x}^*)$ is positive definite. local minimum

Example 2: Two Dimensional Ideal Points

Suppose legislators are considering legislation $\boldsymbol{x} \in \Re^2$. And suppose legislator i has utility function $U_i : \Re^2 \rightarrow \Re$,

$$U(\boldsymbol{x})_i = -(x_1 - \mu_1)^2 - (x_2 - \mu_2)^2$$

What is legislator i's optimal policy?
$\nabla f(\boldsymbol{x}) = (-2(x_1 - \mu_1), -2(x_2 - \mu_2))$
$\nabla f(\boldsymbol{x}) = \mathbf{0}$

$$\begin{aligned} -2(x_1^{*} - \mu_1) &= 0 \\ -2(x_2^{*} - \mu_2) &= 0 \end{aligned}$$

Solving yields $x_1^{*} = \mu_1$ and $x_2^{*} = \mu_2$.

Example 2: Two Dimensional Ideal Points

$$U(\boldsymbol{x})_i = -(x_1 - \mu_1)^2 - (x_2 - \mu_2)^2$$

Call $\boldsymbol{\mu} = (\mu_1, \mu_2)$
The Hessian at the critical value is

$$\begin{aligned} \boldsymbol{H}(f)(\boldsymbol{\mu}) &= \begin{pmatrix} \frac{\partial^2 U_i}{\partial x_1 \partial x_1}(\boldsymbol{\mu}) & \frac{\partial^2 U_i}{\partial x_1 \partial x_2}(\boldsymbol{\mu}) \\ \frac{\partial^2 U_i}{\partial x_2 \partial x_1}(\boldsymbol{\mu}) & \frac{\partial^2 U_i}{\partial x_2 \partial x_2}(\boldsymbol{\mu}) \end{pmatrix} \\ &= \begin{pmatrix} -2 & 0 \\ 0 & -2 \end{pmatrix} \end{aligned}$$

So, $-2 * -2 - 0 = 4 > 0$ and $-2 < 0 \rightsquigarrow$ negative definite, maximum
$\boldsymbol{\mu} = (\mu_1, \mu_2)$ are legislator i's two dimensional ideal point.

Example 3: Maximum Likelihood Estimation, Normal Distribution

Example 3: Maximum Likelihood Estimation, Normal Distribution

Suppose that we draw an independent and identically distributed random sample of n observations from a normal distribution,

Example 3: Maximum Likelihood Estimation, Normal Distribution

Suppose that we draw an independent and identically distributed random sample of n observations from a normal distribution,

$$Y_i \sim \text{Normal}(\mu, \sigma^2)$$

Example 3: Maximum Likelihood Estimation, Normal Distribution

Suppose that we draw an independent and identically distributed random sample of n observations from a normal distribution,

$$\begin{aligned} Y_i &\sim \text{Normal}(\mu, \sigma^2) \\ \boldsymbol{Y} &= (Y_1, Y_2, \ldots, Y_n) \end{aligned}$$

Example 3: Maximum Likelihood Estimation, Normal Distribution

Suppose that we draw an independent and identically distributed random sample of n observations from a normal distribution,

$$\begin{aligned} Y_i &\sim \text{Normal}(\mu, \sigma^2) \\ \boldsymbol{Y} &= (Y_1, Y_2, \ldots, Y_n) \end{aligned}$$

Our task:

Example 3: Maximum Likelihood Estimation, Normal Distribution

Suppose that we draw an independent and identically distributed random sample of *n* observations from a normal distribution,

$$\begin{aligned} Y_i &\sim \text{Normal}(\mu, \sigma^2) \\ \boldsymbol{Y} &= (Y_1, Y_2, \ldots, Y_n) \end{aligned}$$

Our task:

- Obtain likelihood (summary estimator)

Example 3: Maximum Likelihood Estimation, Normal Distribution

Suppose that we draw an independent and identically distributed random sample of n observations from a normal distribution,

$$\begin{aligned} Y_i &\sim \text{Normal}(\mu, \sigma^2) \\ \boldsymbol{Y} &= (Y_1, Y_2, \ldots, Y_n) \end{aligned}$$

Our task:

- Obtain likelihood (summary estimator)
- Derive maximum likelihood estimators for μ and σ^2

Example 3: Maximum Likelihood Estimation, Normal Distribution

Suppose that we draw an independent and identically distributed random sample of n observations from a normal distribution,

$$\begin{aligned} Y_i &\sim \text{Normal}(\mu, \sigma^2) \\ \boldsymbol{Y} &= (Y_1, Y_2, \ldots, Y_n) \end{aligned}$$

Our task:

- Obtain likelihood (summary estimator)
- Derive maximum likelihood estimators for μ and σ^2
- Characterize sampling distribution

Example 3: Maximum Likelihood Estimation, Normal Distribution

Example 3: Maximum Likelihood Estimation, Normal Distribution

$$L(\mu, \sigma^2 | \boldsymbol{Y}) \quad \propto \quad \prod_{i=1}^{n} f(Y_i | \mu, \sigma^2)$$

Example 3: Maximum Likelihood Estimation, Normal Distribution

$$
\begin{aligned}
L(\mu, \sigma^2 | \boldsymbol{Y}) & \propto \prod_{i=1}^{n} f(Y_i | \mu, \sigma^2) \\
& \propto \prod_{i=1}^{N} \frac{\exp[-\frac{(Y_i - \mu)^2}{2\sigma^2}]}{\sqrt{2\pi\sigma^2}}
\end{aligned}
$$

Example 3: Maximum Likelihood Estimation, Normal Distribution

$$
\begin{aligned}
L(\mu, \sigma^2 | \boldsymbol{Y}) & \propto \prod_{i=1}^{n} f(Y_i | \mu, \sigma^2) \\
& \propto \prod_{i=1}^{N} \frac{\exp[-\frac{(Y_i - \mu)^2}{2\sigma^2}]}{\sqrt{2\pi\sigma^2}} \\
& \propto \frac{\exp[-\sum_{i=1}^{n} \frac{(Y_i - \mu)^2}{2\sigma^2}]}{(2\pi)^{n/2}\sigma^{2n/2}}
\end{aligned}
$$

Example 3: Maximum Likelihood Estimation, Normal Distribution

$$
\begin{aligned}
L(\mu, \sigma^2|\boldsymbol{Y}) & \propto \prod_{i=1}^{n} f(Y_i|\mu, \sigma^2) \\
& \propto \prod_{i=1}^{N} \frac{\exp[-\frac{(Y_i-\mu)^2}{2\sigma^2}]}{\sqrt{2\pi\sigma^2}} \\
& \propto \frac{\exp[-\sum_{i=1}^{n} \frac{(Y_i-\mu)^2}{2\sigma^2}]}{(2\pi)^{n/2}\sigma^{2n/2}}
\end{aligned}
$$

Taking the logarithm, we have

Example 3: Maximum Likelihood Estimation, Normal Distribution

$$
\begin{aligned}
L(\mu, \sigma^2 | \boldsymbol{Y}) &\propto \prod_{i=1}^{n} f(Y_i | \mu, \sigma^2) \\
&\propto \prod_{i=1}^{N} \frac{\exp[-\frac{(Y_i - \mu)^2}{2\sigma^2}]}{\sqrt{2\pi\sigma^2}} \\
&\propto \frac{\exp[-\sum_{i=1}^{n} \frac{(Y_i - \mu)^2}{2\sigma^2}]}{(2\pi)^{n/2}\sigma^{2n/2}}
\end{aligned}
$$

Taking the logarithm, we have

$$
l(\mu, \sigma^2 | \boldsymbol{Y}) = -\sum_{i=1}^{n} \frac{(Y_i - \mu)^2}{2\sigma^2} - \frac{n}{2} log(2\pi) - \frac{n}{2} \log(\sigma^2) + {\color{red} c}
$$

Example 3: Maximum Likelihood Estimation, Normal Distribution

$$
\begin{aligned}
L(\mu, \sigma^2 | \boldsymbol{Y}) & \propto \prod_{i=1}^{n} f(Y_i | \mu, \sigma^2) \\
& \propto \prod_{i=1}^{N} \frac{\exp[-\frac{(Y_i - \mu)^2}{2\sigma^2}]}{\sqrt{2\pi\sigma^2}} \\
& \propto \frac{\exp[-\sum_{i=1}^{n} \frac{(Y_i-\mu)^2}{2\sigma^2}]}{(2\pi)^{n/2}\sigma^{2n/2}}
\end{aligned}
$$

Taking the logarithm, we have

$$
\begin{aligned}
l(\mu, \sigma^2 | \boldsymbol{Y}) & = -\sum_{i=1}^{n} \frac{(Y_i - \mu)^2}{2\sigma^2} - \frac{n}{2} log(2\pi) - \frac{n}{2}\log(\sigma^2) + c \\
& = -\sum_{i=1}^{n} \frac{(Y_i - \mu)^2}{2\sigma^2} - \frac{n}{2}\log(\sigma^2) + c^{'}
\end{aligned}
$$

Example 3: Log-Likelihood Plot

- In R, drew 10,000 realizations from

Example 3: Log-Likelihood Plot

- In R, drew 10,000 realizations from

$$Y_i \sim \text{Normal}(0.25, 100)$$

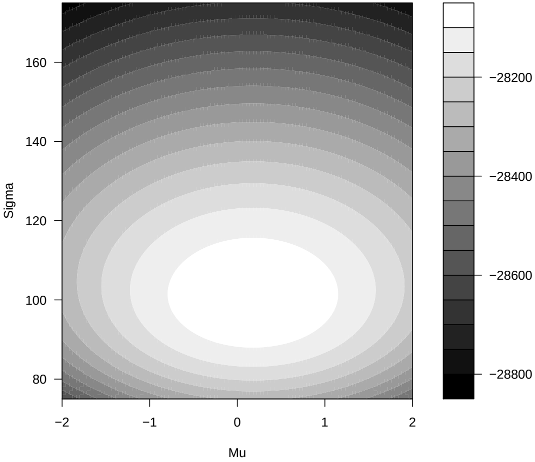
Example 3: Log-Likelihood Plot

- In R, drew 10,000 realizations from

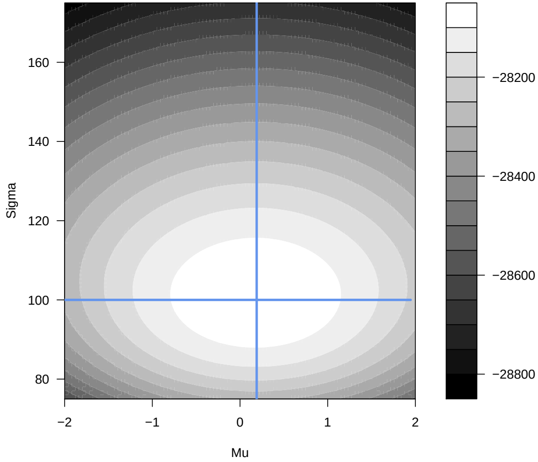
$$Y_i \sim \text{Normal}(0.25, 100)$$

- Used realized values y_i evaluate $l(\mu, \sigma^2|\boldsymbol{y})$

Example 3: Log-Likelihood Plot



Example 3: Log-Likelihood Plot



Example 3: Maximum Likelihood Estimation, Normal Distribution

Let's find $\widehat{\mu}$ and $\widehat{\sigma}^2$ that maximizes log-likelihood.

Example 3: Maximum Likelihood Estimation, Normal Distribution

Let's find $\widehat{\mu}$ and $\widehat{\sigma}^2$ that maximizes log-likelihood.

$$l(\mu, \sigma^2 | \boldsymbol{Y}) = -\sum_{i=1}^{n} \frac{(Y_i - \mu)^2}{2\sigma^2} - \frac{n}{2}\log(\sigma^2) + c^{'}$$

Example 3: Maximum Likelihood Estimation, Normal Distribution

Let's find $\widehat{\mu}$ and $\widehat{\sigma}^2$ that maximizes log-likelihood.

$$
\begin{aligned}
l(\mu, \sigma^2 | \boldsymbol{Y}) &= -\sum_{i=1}^{n} \frac{(Y_i - \mu)^2}{2\sigma^2} - \frac{n}{2} \log(\sigma^2) + c^{'} \\
\frac{\partial l(\mu, \sigma^2) | \boldsymbol{Y})}{\partial \mu} &= \sum_{i=1}^{n} \frac{2(Y_i - \mu)}{2\sigma^2}
\end{aligned}
$$

Example 3: Maximum Likelihood Estimation, Normal Distribution

Let's find $\widehat{\mu}$ and $\widehat{\sigma}^2$ that maximizes log-likelihood.

$$
\begin{aligned}
l(\mu, \sigma^2 | \boldsymbol{Y}) &= -\sum_{i=1}^{n} \frac{(Y_i - \mu)^2}{2\sigma^2} - \frac{n}{2}\log(\sigma^2) + c^{'} \\
\frac{\partial l(\mu, \sigma^2)|\boldsymbol{Y})}{\partial \mu} &= \sum_{i=1}^{n} \frac{2(Y_i - \mu)}{2\sigma^2} \\
\frac{\partial l(\mu, \sigma^2)|\boldsymbol{Y})}{\partial \sigma^2} &= -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_{i=1}^{n}(Y_i - \mu)^2
\end{aligned}
$$

Example 3: Maximum Likelihood Estimation, Normal Distribution

$$
\begin{aligned}
0 &= -\sum_{i=1}^{n} \frac{2(Y_i - \widehat{\mu})}{2\widehat{\sigma}^2} \\
0 &= -\frac{n}{2\widehat{\sigma}^2} + \frac{1}{2\widehat{\sigma}^4}\sum_{i=1}^{n}(Y_i - \mu^*)^2
\end{aligned}
$$

Example 3: Maximum Likelihood Estimation, Normal Distribution

$$
\begin{aligned}
0 &= -\sum_{i=1}^{n} \frac{2(Y_i - \widehat{\mu})}{2\widehat{\sigma}^2} \\
0 &= -\frac{n}{2\widehat{\sigma}^2} + \frac{1}{2\widehat{\sigma}^4}\sum_{i=1}^{n}(Y_i - \mu^*)^2
\end{aligned}
$$

Solving for $\widehat{\mu}$ and $\widehat{\sigma}^2$ yields,

Example 3: Maximum Likelihood Estimation, Normal Distribution

$$
\begin{aligned}
0 &= -\sum_{i=1}^{n} \frac{2(Y_i - \widehat{\mu})}{2\widehat{\sigma}^2} \\
0 &= -\frac{n}{2\widehat{\sigma}^2} + \frac{1}{2\widehat{\sigma}^4}\sum_{i=1}^{n}(Y_i - \mu^*)^2
\end{aligned}
$$

Solving for $\widehat{\mu}$ and $\widehat{\sigma}^2$ yields,

$$
\widehat{\mu} = \frac{\sum_{i=1}^{n} Y_i}{n}
$$

Example 3: Maximum Likelihood Estimation, Normal Distribution

$$
\begin{aligned}
0 &= -\sum_{i=1}^{n} \frac{2(Y_i - \widehat{\mu})}{2\widehat{\sigma}^2} \\
0 &= -\frac{n}{2\widehat{\sigma}^2} + \frac{1}{2\widehat{\sigma}^4}\sum_{i=1}^{n}(Y_i - \mu^*)^2
\end{aligned}
$$

Solving for $\widehat{\mu}$ and $\widehat{\sigma}^2$ yields,

$$
\begin{aligned}
\widehat{\mu} &= \frac{\sum_{i=1}^{n} Y_i}{n} \\
\widehat{\sigma}^2 &= \frac{1}{n}\sum_{i=1}^{n}(Y_i - \overline{Y})^2
\end{aligned}
$$

Example 3: Maximum Likelihood Estimation, Normal Distribution

Example 3: Maximum Likelihood Estimation, Normal Distribution

$$\mathbf{H}(f)(\widehat{\mu}, \widehat{\sigma}^2) = \begin{pmatrix} \frac{\partial^2 l(\mu,\sigma^2|\boldsymbol{Y})}{\partial \mu^2} & \frac{\partial^2 l(\mu,\sigma^2|\boldsymbol{Y})}{\partial \sigma^2 \partial \mu} \\ \frac{\partial^2 l(\mu,\sigma^2|\boldsymbol{Y})}{\partial \sigma^2 \partial \mu} & \frac{\partial^2 l(\mu,\sigma^2|\boldsymbol{Y})}{\partial^2 \sigma^2} \end{pmatrix}$$

Example 3: Maximum Likelihood Estimation, Normal Distribution

$$\mathbf{H}(f)(\widehat{\mu}, \widehat{\sigma}^2) = \begin{pmatrix} \frac{\partial^2 l(\mu,\sigma^2|\boldsymbol{Y})}{\partial \mu^2} & \frac{\partial^2 l(\mu,\sigma^2|\boldsymbol{Y})}{\partial \sigma^2 \partial \mu} \\ \frac{\partial^2 l(\mu,\sigma^2|\boldsymbol{Y})}{\partial \sigma^2 \partial \mu} & \frac{\partial^2 l(\mu,\sigma^2|\boldsymbol{Y})}{\partial^2 \sigma^2} \end{pmatrix}$$

Taking derivatives and evaluating at MLE's yields,

Example 3: Maximum Likelihood Estimation, Normal Distribution

$$\mathbf{H}(f)(\widehat{\mu}, \widehat{\sigma}^2) = \begin{pmatrix} \frac{\partial^2 l(\mu,\sigma^2|\boldsymbol{Y})}{\partial \mu^2} & \frac{\partial^2 l(\mu,\sigma^2|\boldsymbol{Y})}{\partial \sigma^2 \partial \mu} \\ \frac{\partial^2 l(\mu,\sigma^2|\boldsymbol{Y})}{\partial \sigma^2 \partial \mu} & \frac{\partial^2 l(\mu,\sigma^2|\boldsymbol{Y})}{\partial^2 \sigma^2} \end{pmatrix}$$

Taking derivatives and evaluating at MLE's yields,

$$\mathbf{H}(f)(\widehat{\mu}, \widehat{\sigma}^2) = \begin{pmatrix} \frac{-n}{\widehat{\sigma}^2} & 0 \\ 0 & \frac{-n}{(\widehat{\sigma}^2)^2} \end{pmatrix}$$

Example 3: Maximum Likelihood Estimation, Normal Distribution

$$\mathbf{H}(f)(\widehat{\mu}, \widehat{\sigma}^2) = \begin{pmatrix} \frac{\partial^2 l(\mu,\sigma^2|\boldsymbol{Y})}{\partial \mu^2} & \frac{\partial^2 l(\mu,\sigma^2|\boldsymbol{Y})}{\partial \sigma^2 \partial \mu} \\ \frac{\partial^2 l(\mu,\sigma^2|\boldsymbol{Y})}{\partial \sigma^2 \partial \mu} & \frac{\partial^2 l(\mu,\sigma^2|\boldsymbol{Y})}{\partial^2 \sigma^2} \end{pmatrix}$$

Taking derivatives and evaluating at MLE's yields,

$$\mathbf{H}(f)(\widehat{\mu}, \widehat{\sigma}^2) = \begin{pmatrix} \frac{-n}{\widehat{\sigma}^2} & 0 \\ 0 & \frac{-n}{(\widehat{\sigma}^2)^2} \end{pmatrix}$$

$\det(\mathbf{H}(f)(\widehat{\mu}, \widehat{\sigma}^2)) = n^2/\widehat{\sigma}^5$ and $-n/\widehat{\sigma}^2 < 0 \rightsquigarrow$ maximum

Computational Optimization

Analytic solutions: often hard.

Computational Optimization

Analytic solutions: often hard.
Computational solutions: simplify. Trade offs

Computational Optimization

Analytic solutions: often hard.

Computational solutions: simplify. Trade offs

- Newton-Raphson: expensive

Computational Optimization

Analytic solutions: often hard.
Computational solutions: simplify. Trade offs

- Newton-Raphson: expensive
- BFGS: less expensive

Computational Optimization

Analytic solutions: often hard.
Computational solutions: simplify. Trade offs

- Newton-Raphson: expensive
- BFGS: less expensive
- EM-like optimization: solve intractable problems, parallelizable

Multivariate Newton Raphson

Suppose $f : \Re^n \rightarrow \Re$. Suppose we have guess $\boldsymbol{x}_t$.

Multivariate Newton Raphson

Suppose $f : \Re^n \rightarrow \Re$. Suppose we have guess $\boldsymbol{x}_t$. Then our update is:

Multivariate Newton Raphson

Suppose $f : \Re^n \rightarrow \Re$. Suppose we have guess $\boldsymbol{x}_t$. Then our update is:

$$\boldsymbol{x}_{t+1} \quad = \quad \boldsymbol{x}_t - \mathbf{H}(f)(\boldsymbol{x}_t)^{-1} \nabla f(\boldsymbol{x}_t)$$

Multivariate Newton Raphson

Suppose $f : \Re^n \rightarrow \Re$. Suppose we have guess $\boldsymbol{x}_t$. Then our update is:

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \mathbf{H}(f)(\boldsymbol{x}_t)^{-1}\nabla f(\boldsymbol{x}_t)$$

Derivation (intuition):

Multivariate Newton Raphson

Suppose $f : \Re^n \rightarrow \Re$. Suppose we have guess $\boldsymbol{x}_t$. Then our update is:

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \mathbf{H}(f)(\boldsymbol{x}_t)^{-1} \nabla f(\boldsymbol{x}_t)$$

Derivation (intuition): Approximate function with tangent plane.

Multivariate Newton Raphson

Suppose $f : \Re^n \rightarrow \Re$. Suppose we have guess $\boldsymbol{x}_t$. Then our update is:

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \mathbf{H}(f)(\boldsymbol{x}_t)^{-1}\nabla f(\boldsymbol{x}_t)$$

Derivation (intuition): Approximate function with tangent plane. Find value of x_{t+1} that makes the plane equal to zero. Update again.

Multivariate Newton Raphson

Suppose $f : \Re^n \rightarrow \Re$. Suppose we have guess $\boldsymbol{x}_t$. Then our update is:

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \mathbf{H}(f)(\boldsymbol{x}_t)^{-1}\nabla f(\boldsymbol{x}_t)$$

Derivation (intuition): Approximate function with tangent plane. Find value of x_{t+1} that makes the plane equal to zero. Update again.

`R Code`

Multivariate Newton Raphson

Multivariate Newton Raphson

- Expensive to calculate (requires inverting Hessian)

Multivariate Newton Raphson

- Expensive to calculate (requires inverting Hessian)
- Very sensitive to starting points

Multivariate Newton Raphson

- Expensive to calculate (requires inverting Hessian)
- Very sensitive to starting points
- Ideally: method that exploits Newton-like structure, but is cheaper and more robust

Multivariate Newton Raphson

- Expensive to calculate (requires inverting Hessian)
- Very sensitive to starting points
- Ideally: method that exploits Newton-like structure, but is cheaper and more robust

`BFGS`: Quasi-Newton method

Multivariate Newton Raphson

- Expensive to calculate (requires inverting Hessian)
- Very sensitive to starting points
- Ideally: method that exploits Newton-like structure, but is cheaper and more robust

`BFGS`: Quasi-Newton method

`R` code

Optimization that is Both Discrete and Continuous

K-means: most commonly used clustering algorithm.

Optimization that is Both Discrete and Continuous

K-means: most commonly used clustering algorithm.

Story: Data are grouped in K clusters and each cluster has a center or mean.

Optimization that is Both Discrete and Continuous

K-means: most commonly used clustering algorithm.

Story: Data are grouped in K clusters and each cluster has a center or mean.

$\rightarrow$ Two types of parameters to estimate

Optimization that is Both Discrete and Continuous

K-means: most commonly used clustering algorithm.

Story: Data are grouped in K clusters and each cluster has a center or mean.

$\rightarrow$ Two types of parameters to estimate

1) For each cluster j, $(j = 1, \ldots, K)$

   r_{ij} =Indicator, Document i assigned to cluster j

   $\boldsymbol{r}_j = (r_{1j}, r_{2j}, \ldots, r_{Nj})$

   $\boldsymbol{r} = (\boldsymbol{r}_1^{'}, \boldsymbol{r}_2^{'}, \ldots, \boldsymbol{r}_K^{'})$ ($N \times K$ matrix)

Optimization that is Both Discrete and Continuous

K-means: most commonly used clustering algorithm.

Story: Data are grouped in K clusters and each cluster has a center or mean.

$\rightarrow$ Two types of parameters to estimate

1) For each cluster j, $(j = 1, \ldots, K)$

   r_{ij} =Indicator, Document i assigned to cluster j

   $\boldsymbol{r}_j = (r_{1j}, r_{2j}, \ldots, r_{Nj})$

   $\boldsymbol{r} = (\boldsymbol{r}^{'}_1, \boldsymbol{r}^{'}_2, \ldots, \boldsymbol{r}^{'}_K)$ ($N \times K$ matrix)

2) For each cluster j

   $\boldsymbol{\mu}_j$ a cluster center for cluster j.

   $\boldsymbol{\mu}_j = (\mu_{1j}, \mu_{2j}, \ldots, \mu_{Mj})$

Optimization that is Both Discrete and Continuous

K-means: most commonly used clustering algorithm.

Story: Data are grouped in K clusters and each cluster has a center or mean.

$\rightarrow$ Two types of parameters to estimate

1) For each cluster j, ($j = 1, \ldots, K$)

   r_{ij} =Indicator, Document i assigned to cluster j

   $\boldsymbol{r}_j = (r_{1j}, r_{2j}, \ldots, r_{Nj})$

   $\boldsymbol{r} = (\boldsymbol{r}_1^{'}, \boldsymbol{r}_2^{'}, \ldots, \boldsymbol{r}_K^{'})$ ($N \times K$ matrix)

2) For each cluster j

   $\boldsymbol{\mu}_j$ a cluster center for cluster j.

   $\boldsymbol{\mu}_j = (\mu_{1j}, \mu_{2j}, \ldots, \mu_{Mj})$

Notation. Representation of document i:

$$\boldsymbol{y}_i = (y_{i1}, y_{i2}, \ldots, y_{iM})$$

Specifying the Method

1) Assume Euclidean distance between objects.
2) Objective function

$$f(\boldsymbol{r}, \boldsymbol{\mu}, \boldsymbol{y}) = \sum_{i=1}^{N} \sum_{j=1}^{K} r_{ij} \left(\sum_{m=1}^{M} (y_{im} - \mu_{km})^2 \right)$$

Specifying the Method

1) Assume Euclidean distance between objects.
2) Objective function

$$f(\boldsymbol{r}, \boldsymbol{\mu}, \boldsymbol{y}) = \sum_{i=1}^{N} \sum_{j=1}^{K} r_{ij} \left(\sum_{m=1}^{M} (y_{im} - \mu_{km})^2 \right)$$

Goal:

Specifying the Method

1) Assume Euclidean distance between objects.
2) Objective function

$$f(\boldsymbol{r}, \boldsymbol{\mu}, \boldsymbol{y}) \quad = \quad \sum_{i=1}^{N}\sum_{j=1}^{K} r_{ij}\left(\sum_{m=1}^{M}(y_{im} - \mu_{km})^2\right)$$

Goal:
Choose $\boldsymbol{r}^*$ and $\boldsymbol{\mu}^*$ to minimize $f(\boldsymbol{r}, \boldsymbol{\mu}, \boldsymbol{y})$

Specifying the Method

1) Assume Euclidean distance between objects.
2) Objective function

$$f(\boldsymbol{r}, \boldsymbol{\mu}, \boldsymbol{y}) \quad = \quad \sum_{i=1}^{N}\sum_{j=1}^{K} r_{ij}\left(\sum_{m=1}^{M}(y_{im} - \mu_{km})^2\right)$$

Goal:
Choose $\boldsymbol{r}^*$ and $\boldsymbol{\mu}^*$ to minimize $f(\boldsymbol{r}, \boldsymbol{\mu}, \boldsymbol{y})$
Two observations:

- If $K = N$ $f(r^*, \boldsymbol{\mu}^*, \boldsymbol{y}) = 0$ (Minimum)
    - Each observation in own cluster
    - $\boldsymbol{\mu}_i = \boldsymbol{y}_i$
- If $K = 1$, $f(r^*, \boldsymbol{\mu}^*, \boldsymbol{y}) = N \times \sigma^2$
    - Each observation in one cluster
    - Center: average of documents

Specifying the Method

1) Assume Euclidean distance between objects
2) Objective function
3) Algorithm for optimization

Iterative algorithm, Each Iteration t

- Conditional on $\boldsymbol{\mu}^{t-1}$ (from previous iteration), choose $\boldsymbol{r}^t$
- Conditional on $\boldsymbol{r}^t$, choose $\boldsymbol{\mu}^t$

Repeat until convergence, measured as change in f.

$$\text{Change} \quad = \quad f(\boldsymbol{\mu}^t, \boldsymbol{r}^t, \boldsymbol{y}) - f(\boldsymbol{\mu}^{t-1}, \boldsymbol{r}^{t-1}, \boldsymbol{y})$$

Specifying the Method

$$f(\boldsymbol{r}, \boldsymbol{\mu}, \boldsymbol{y}) = \sum_{i=1}^{N}\sum_{j=1}^{K} r_{ij}\left(\sum_{m=1}^{M}(y_{im} - \mu_{km})^2\right)$$

Algorithm for estimation:
Begin: initialize $\boldsymbol{\mu}_1^{t-1}, \boldsymbol{\mu}_2^{t-1}, \ldots, \boldsymbol{\mu}_K^{t-1}$
Choose $\boldsymbol{r}^t$

$$r_{ij}^t = \begin{cases} 1 \text{ if } j = \arg\min_k \sum_{m=1}^{M}(y_{im} - \mu_{km})^2 \\ 0 \text{ otherwise }, \end{cases} .$$

In words: Assign each document $\boldsymbol{y}_i$ to the closest center $\boldsymbol{\mu}_k$

$$f(\boldsymbol{r}, \boldsymbol{\mu}, \boldsymbol{y}) = \sum_{i=1}^{N}\sum_{j=1}^{K} r_{ij}\left(\sum_{m=1}^{M}(y_{im} - \mu_{km})^2\right)$$

Conditional on $\boldsymbol{r}^t$, choose $\boldsymbol{\mu}^t$
Let's focus on $\boldsymbol{\mu}_k$

$$f(\boldsymbol{r}, \boldsymbol{\mu}_k, \boldsymbol{y})_k = \sum_{i=1}^{N} r_{ik}\left(\sum_{m=1}^{M}(y_{im} - \mu_{km})^2\right)$$

Focus on just μ_{km}

$$f(\boldsymbol{r}, \mu_{km}, \boldsymbol{y})_{km} = \sum_{i=1}^{N} r_{ik}(y_{im} - \mu_{km})^2$$

Quadratic: take derivative, set equal to zero (second derivative test works)

$$\begin{aligned}
\frac{\partial f(\boldsymbol{r}, \mu_{km}, \boldsymbol{y})_{km}}{\partial \mu_{km}} &= -2\sum_{i=1}^{N} r_{ik}(y_{im} - \mu_{km}) \\
2\sum_{i=1}^{N} r_{ik}(y_{im} - \mu_{km}^{t}) &= 0 \\
\sum_{i=1}^{N} r_{ik} y_{im} - \mu_{km}^{t} \sum_{i=1}^{N} r_{ik} &= 0 \\
\frac{\sum_{i=1}^{N} r_{ik} y_{im}}{\sum_{i=1}^{N} r_{ik}} &= \mu_{km}^{t}
\end{aligned}$$

$$\boldsymbol{\mu}_k^t = \frac{\sum_{i=1}^N r_{ik}\boldsymbol{y}_i}{\sum_{i=1}^N r_{ik}}$$

In words:

- $\boldsymbol{\mu}_k^t$ is the average of documents assigned to the k^{th} cluster

Algorithm, In Words

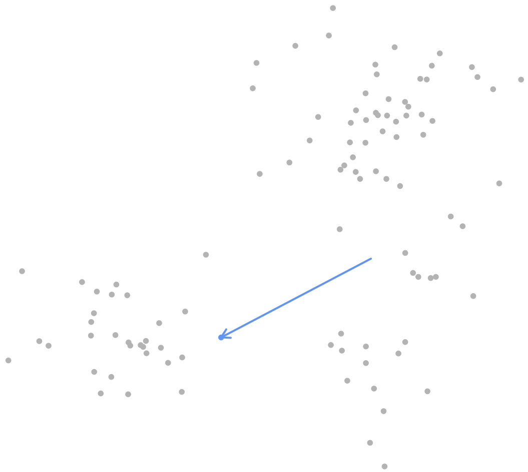
- Conditional on center estimates, assign documents to closest cluster centers
- Conditional on document assignments, cluster centers are averages of documents assigned to the cluster

Expectation-Maximization (EM) [connection guarantees convergence]

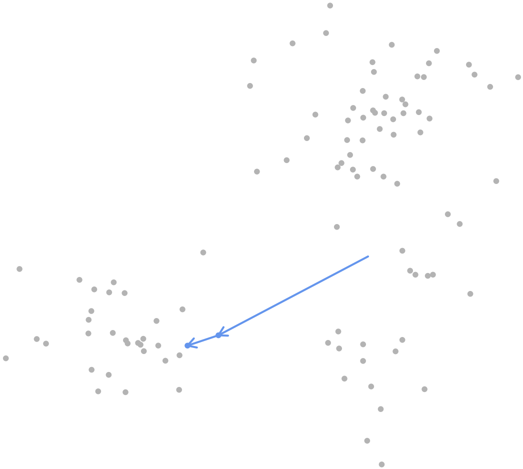
- Estimation of $r \rightsquigarrow$ Expectation step (data augmentation)
- Estimation of $\boldsymbol{\mu}_k \rightsquigarrow$ Maximization Step

Visual Example

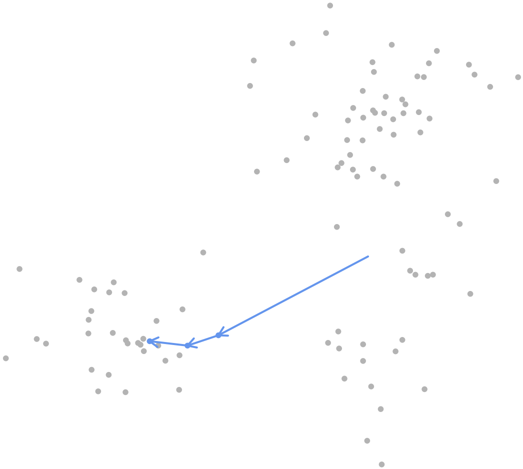
Visual Example



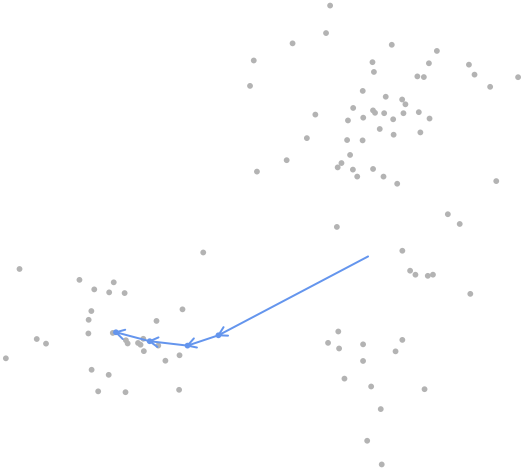
Visual Example



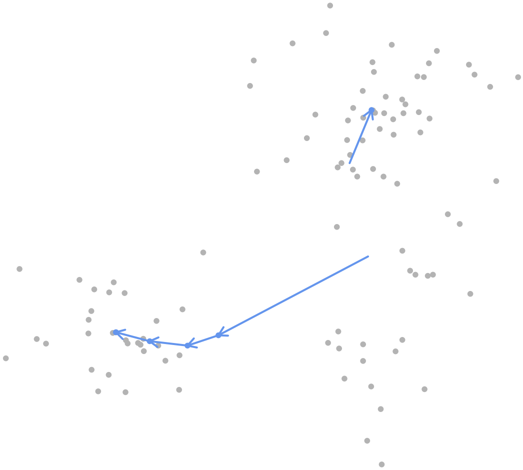
Visual Example



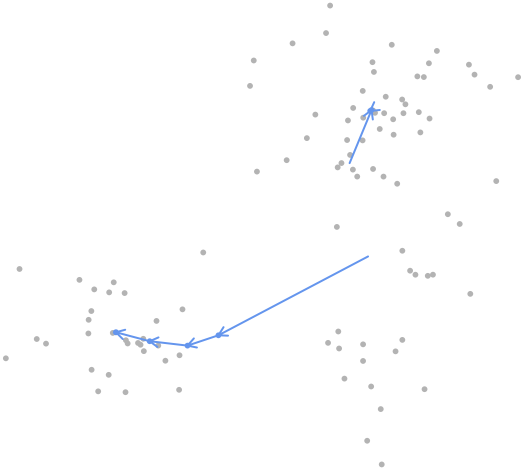
Visual Example



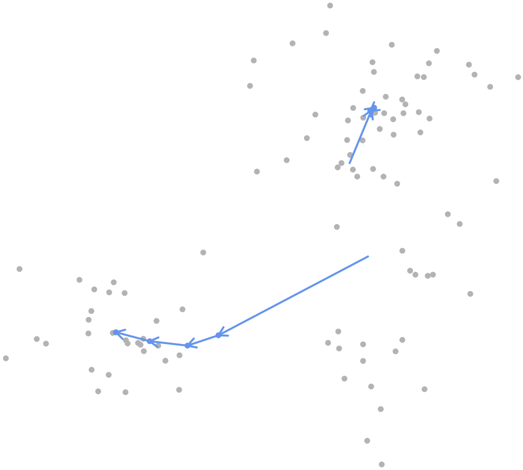
Visual Example



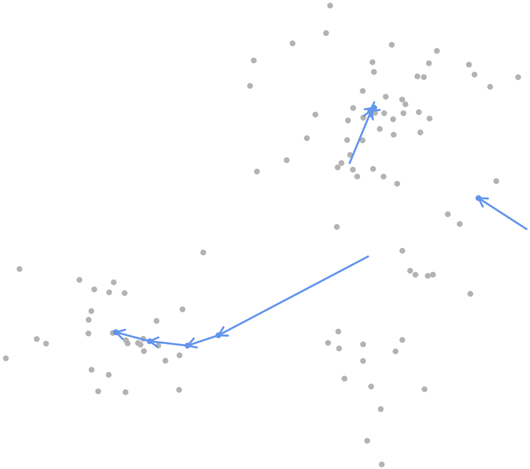
Visual Example



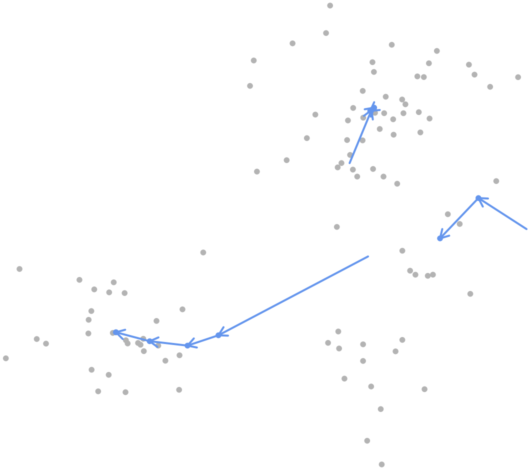
Visual Example



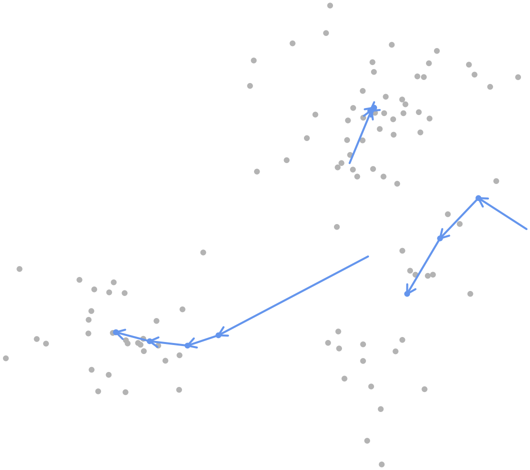
Visual Example



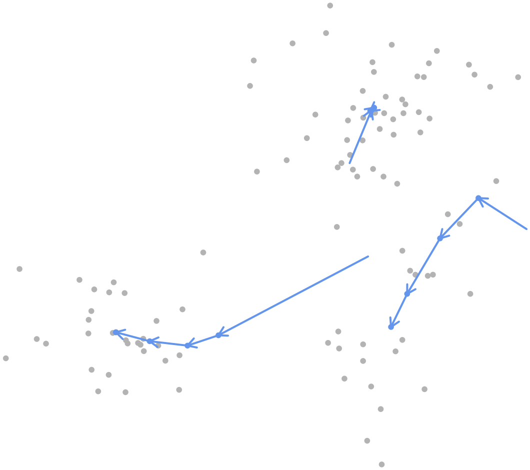
Visual Example



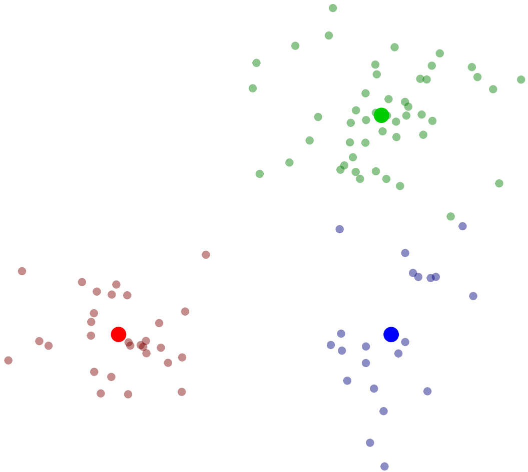
Visual Example



Visual Example



Visual Example



Many Optimization Procedures!!!

Many Optimization Procedures!!!

Nelder Mead:

Many Optimization Procedures!!!

Nelder Mead:

- Evaluate points on a simplex (triangle)

Many Optimization Procedures!!!

Nelder Mead:

- Evaluate points on a simplex (triangle)
- Either Reflect, Expand, or Contract (based on values)

Many Optimization Procedures!!!

Nelder Mead:

- Evaluate points on a simplex (triangle)
- Either Reflect, Expand, or Contract (based on values)
- Converges to local extrema

Many Optimization Procedures!!!

Nelder Mead:

- Evaluate points on a simplex (triangle)
- Either Reflect, Expand, or Contract (based on values)
- Converges to local extrema

Stochastic Optimization:

Many Optimization Procedures!!!

Nelder Mead:

- Evaluate points on a simplex (triangle)
- Either Reflect, Expand, or Contract (based on values)
- Converges to local extrema

Stochastic Optimization:

- Sample a subset of data, perform optimization

Many Optimization Procedures!!!

Nelder Mead:

- Evaluate points on a simplex (triangle)
- Either Reflect, Expand, or Contract (based on values)
- Converges to local extrema

Stochastic Optimization:

- Sample a subset of data, perform optimization
- Sample a new subset, perform optimization, combine with previous sample

Many Optimization Procedures!!!

Nelder Mead:

- Evaluate points on a simplex (triangle)
- Either Reflect, Expand, or Contract (based on values)
- Converges to local extrema

Stochastic Optimization:

- Sample a subset of data, perform optimization
- Sample a new subset, perform optimization, combine with previous sample
- Converges on local extrema (given regulatory conditions)

Many Optimization Procedures!!!

Nelder Mead:

- Evaluate points on a simplex (triangle)
- Either Reflect, Expand, or Contract (based on values)
- Converges to local extrema

Stochastic Optimization:

- Sample a subset of data, perform optimization
- Sample a new subset, perform optimization, combine with previous sample
- Converges on local extrema (given regulatory conditions)

Genetic Optimization:

Many Optimization Procedures!!!

Nelder Mead:

- Evaluate points on a simplex (triangle)
- Either Reflect, Expand, or Contract (based on values)
- Converges to local extrema

Stochastic Optimization:

- Sample a subset of data, perform optimization
- Sample a new subset, perform optimization, combine with previous sample
- Converges on local extrema (given regulatory conditions)

Genetic Optimization:

- Evaluate fitness of solutions

Many Optimization Procedures!!!

Nelder Mead:

- Evaluate points on a simplex (triangle)
- Either Reflect, Expand, or Contract (based on values)
- Converges to local extrema

Stochastic Optimization:

- Sample a subset of data, perform optimization
- Sample a new subset, perform optimization, combine with previous sample
- Converges on local extrema (given regulatory conditions)

Genetic Optimization:

- Evaluate fitness of solutions
- Randomly select most fit, then combine

Many Optimization Procedures!!!

Nelder Mead:

- Evaluate points on a simplex (triangle)
- Either Reflect, Expand, or Contract (based on values)
- Converges to local extrema

Stochastic Optimization:

- Sample a subset of data, perform optimization
- Sample a new subset, perform optimization, combine with previous sample
- Converges on local extrema (given regulatory conditions)

Genetic Optimization:

- Evaluate fitness of solutions
- Randomly select most fit, then combine
- Can converge to global maximum, but might require extensive run time

Where We Are Going

- Done with math component
- Start probability tomorrow