Advanced computational statistics, lecture 1

Frank Miller, Department of Computer and Information Science, Linköping University

Department of Statistics; Stockholm University

March 16, 2023

Course schedule

Optimisation

- Topic 1: **Gradient based optimisation**
- Topic 2: **Stochastic gradient based optimisation**
- Topic 3: **Gradient free optimisation**
- Topic 4: **Optimisation with constraints**
- Topic 5: **EM algorithm and bootstrap**
- Topic 6: **Simulation of random variables**
- Topic 7: **Importance sampling**

Simulation
and Integration

Course homepage:
http://www.adoptdesign.de/frankmillereu/adcompstat2023.html

Includes schedule, reading material, lecture notes, assignments

Optimisation in statistics

- Maximum Likelihood
- Minimising risk in (Bayesian) decision theory
- Minimising sum of squares (Least Squares Estimate)
- Maximising information in experimental design
- Machine learning
- Common problem in these examples:
  - $\boldsymbol{x}$ p-dimensional vector, $g\colon \mathbb{R}^p \to \mathbb{R}$ function
  - We search $\boldsymbol{x}^*$ with $g(\boldsymbol{x}^*) = \max g(\boldsymbol{x})$
  - Typical: $g = \sum_{i=1}^{n} g_i$ with a (large) sample size n where $g_i\colon \mathbb{R}^p \to \mathbb{R}$
  - Minimisation problem turns into maximisation by considering $-g$

Least squares estimation (LSE)

- We search a Least Squares estimate $\widehat{\boldsymbol{\beta}}$ for $\boldsymbol{\beta}$ minimising the distance $g(\widehat{\boldsymbol{\beta}}) = \|\widehat{\boldsymbol{y}} - \boldsymbol{y}\|^2$ from $\widehat{\boldsymbol{y}} = \boldsymbol{X}\widehat{\boldsymbol{\beta}}$ to $\mathbf{y}=\mathbf{X}\,\boldsymbol{\beta} + \boldsymbol{\varepsilon}$
- $g(\widehat{\boldsymbol{\beta}}) = \left\|\boldsymbol{X}\widehat{\boldsymbol{\beta}} - \boldsymbol{y}\right\|^2 = \left(\boldsymbol{X}\widehat{\boldsymbol{\beta}} - \boldsymbol{y}\right)^T\left(\boldsymbol{X}\widehat{\boldsymbol{\beta}} - \boldsymbol{y}\right) = \widehat{\boldsymbol{\beta}}^T\boldsymbol{X}^T\boldsymbol{X}\widehat{\boldsymbol{\beta}} - \mathbf{2}\widehat{\boldsymbol{\beta}}^T\boldsymbol{X}^T\boldsymbol{y} + \boldsymbol{y}^{\boldsymbol{T}}\boldsymbol{y}$
- Setting the derivative to 0 ($\frac{\partial f}{\partial \widehat{\beta}} = 2\boldsymbol{X}^T\boldsymbol{X}\widehat{\boldsymbol{\beta}} - \mathbf{2}\boldsymbol{X}^T\boldsymbol{y} = 0$), we get $\widehat{\boldsymbol{\beta}} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}$

- Note that $g(\widehat{\boldsymbol{\beta}}) = \left\|\boldsymbol{X}\widehat{\boldsymbol{\beta}} - \boldsymbol{y}\right\|^2 = \sum_{i=1}^{n}\left(\boldsymbol{x}_i^T\widehat{\boldsymbol{\beta}} - y_i\right)^2 = \sum_{i=1}^{n} g_i(\widehat{\boldsymbol{\beta}})$
- Optimisation problem:
  - $\widehat{\boldsymbol{\beta}}$ p-dimensional vector, $g\colon \mathbb{R}^p \to \mathbb{R}$ function
  - We search $\widehat{\boldsymbol{\beta}}$ with $g(\widehat{\boldsymbol{\beta}}) = \min g(\boldsymbol{b}) = \min \sum_{i=1}^{n} g_i(\boldsymbol{b})$
- Here, we do not need to iteratively compute this minimum since we have an algebraic solution $\widehat{\boldsymbol{\beta}} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}$

Variations of least squares estimation

- Algebraic solution exists for the LSE, but not if we vary the problem
- Lasso estimate: $g(\widehat{\boldsymbol{\beta}}) = \left\|\boldsymbol{X}\widehat{\boldsymbol{\beta}} - \boldsymbol{y}\right\|^2 + \lambda\left\|\widehat{\boldsymbol{\beta}}\right\|_1 = \sum_{i=1}^{n}\left(\boldsymbol{x}_i\widehat{\boldsymbol{\beta}} - y_i\right)^2 + \lambda\left\|\widehat{\boldsymbol{\beta}}\right\|_1 = \sum_{i=1}^{n} g_i(\widehat{\boldsymbol{\beta}})$
- L_1-estimation: $g(\widehat{\boldsymbol{\beta}}) = \left\|\boldsymbol{X}\widehat{\boldsymbol{\beta}} - \boldsymbol{y}\right\|_1 = \sum_{i=1}^{n}\left|\boldsymbol{x}_i\widehat{\boldsymbol{\beta}} - y_i\right| = \sum_{i=1}^{n} g_i(\widehat{\boldsymbol{\beta}})$
- Many further variations of estimates have been considered
- In all cases, we search $\widehat{\boldsymbol{\beta}}$ with $g(\widehat{\boldsymbol{\beta}}) = \min g(\boldsymbol{b}) = \min \sum_{i=1}^{n} g_i(\boldsymbol{b})$

- Recall: Norms for $\boldsymbol{x} = (x_1, \dots, x_p)^T$: $\|\boldsymbol{x}\| = \|\boldsymbol{x}\|_2 = \sqrt{x_1^2 + \cdots + x_p^2}$ (Euklid), $\|\boldsymbol{x}\|_1 = |x_1| + \cdots + |x_p|$, $\|\boldsymbol{x}\|_\infty = \max\{|x_1|, \dots, |x_p|\}$ (max-norm)

Maximizing information of experimental designs

- Regression model $\mathbf{y}=\mathbf{X}\,\boldsymbol{\beta}+\boldsymbol{\varepsilon}$ (where $\boldsymbol{\varepsilon}$ has iid components)
- $\mathbf{X}$ design matrix (depends on choice of observational points)
- Covariance matrix of Least Squares estimate $\widehat{\boldsymbol{\beta}}$ is
$$\mathrm{Cov}\big(\widehat{\boldsymbol{\beta}}\big) = (\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X})^{\boldsymbol{-1}}\cdot const$$
- Choose design of an experiment such that $\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}$ “large”
- D-optimality: $g(\text{"}\mathbf{design}\text{"}) = \det(\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X})$
- We search $\mathbf{design}^*$ with $g(\mathbf{design}^*) = \max g(\mathbf{design})$

Maximizing information of experimental designs

- Regression model **y**=**X β** + **ε**, $\mathrm{Cov}(\hat{\boldsymbol{\beta}}) = (\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X})^{-\mathbf{1}} \cdot const$
- We search **design*** with $g(\textbf{design}^*) = \max g(\textbf{design})$
- Example: cubic regression, $y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \varepsilon$, n observations in each of following 4 points: $-1, -a, a, 1$. How should $a \in (0,1)$ be chosen?

$$\boldsymbol{X} = \begin{pmatrix} 1 & -1 & 1 & -1 \\ 1 & -a & a^2 & -a^3 \\ 1 & a & a^2 & a^3 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

$$g(a) = \det(\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}) = \det(\boldsymbol{X}(a)^{\boldsymbol{T}}\boldsymbol{X}(a))$$

- We search a^* with $g(a^*) = \max g(a)$

Today's schedule

- Univariate Optimisation (bi-section, Newton, secant)
- Multivariate Optimisation
  - Analytical opt.
  - Newton
  - Steepest ascent
  - Accelerated steepest ascent
  - Quasi-Newton

Univariate optimisation

- x real number, $g: \mathbb{R} \to \mathbb{R}$ continuously differentiable function
- We search x^* with $g(x^*) = \max g(x)$

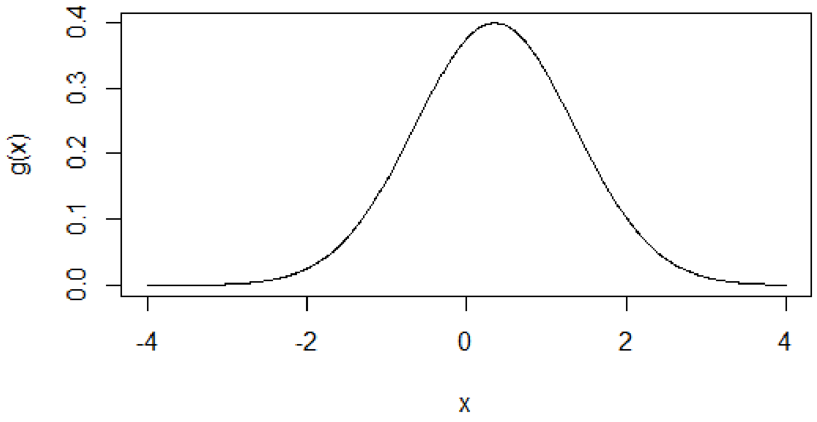
- Compute $g'(x)$ and search x^* with $g'(x^*) = 0$
- One has then to check if the result is maximum, minimum, possibly local optimum…

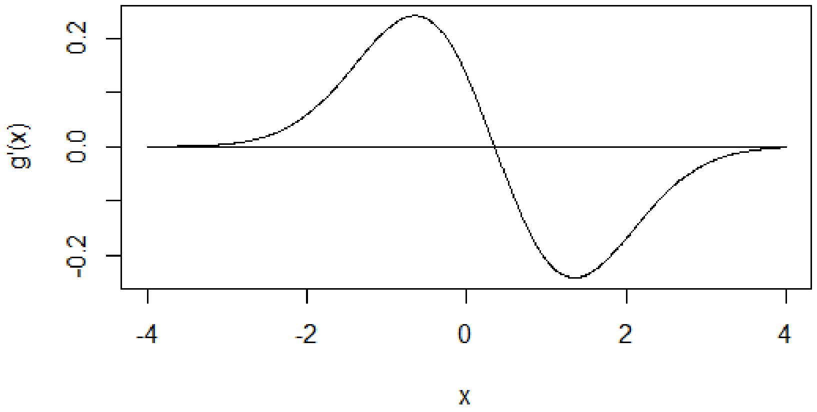
Univariate optimisation: bisection

- Search x^* with $g'(x^*) = 0$:

- See video on course homepage

- We iteratively improve approximations for x^*:
  $x^{(0)}$ -> $x^{(1)}$ -> $x^{(2)}$ -> …





Optimisation: convergence criterion

- Compare $x^{(t)}$ and $x^{(t+1)}$ and stop if they are "close enough"
- Absolute convergence criterion:

$$\left|x^{(t+1)} - x^{(t)}\right| < \epsilon$$

- Relative convergence criterion:

$$\frac{\left|x^{(t+1)} - x^{(t)}\right|}{|x^{(t)}|} < \epsilon$$

Univariate Newton(-Raphson)

- x real number, $g: \mathbb{R} \to \mathbb{R}$ twice differentiable function
- Search x^* with $g(x^*) = \max g(x)$ by searching x^* with $g'(x^*) = 0$

- Taylor expansion around x^* motivates:

$$0 = g'(x^*) \approx g'\left(x^{(t)}\right) + \left(x^* - x^{(t)}\right)g''\left(x^{(t)}\right)$$
$$-\left(x^* - x^{(t)}\right)g''\left(x^{(t)}\right) \approx g'\left(x^{(t)}\right)$$
$$x^* \approx x^{(t)} - g'\left(x^{(t)}\right)/g''(x^{(t)})$$

- Therefore, the Newton-iteration works as:

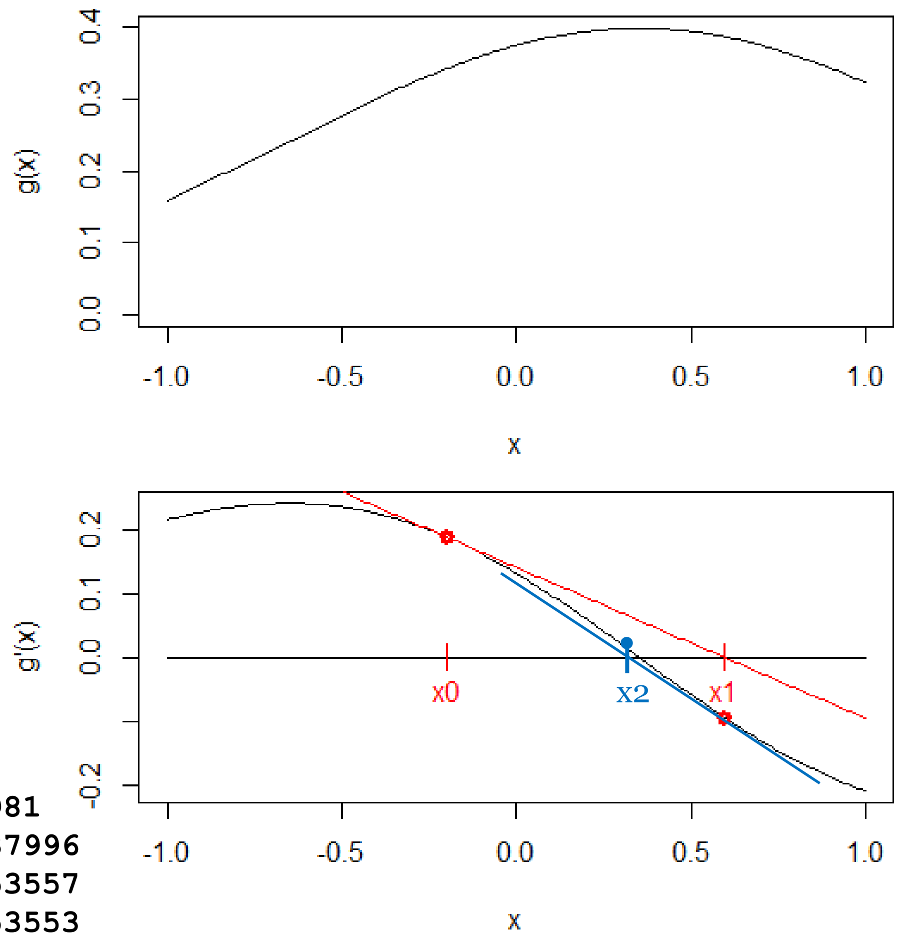
$$x^{(t+1)} = x^{(t)} - g'\left(x^{(t)}\right)/g''(x^{(t)})$$

Univariate Newton(-Raphson)

- $x^{(t+1)} = x^{(t)} - g'(x^{(t)})/g''(x^{(t)})$
- Start with a $x^{(0)}$
- Tangent in $(x^{(0)}, g'(x^{(0)}))$ determines $x^{(1)}$
- Tangent in $(x^{(1)}, g'(x^{(1)}))$ determines $x^{(2)}$
- ...
- until convergence criterion met

+Newton method is fast

- Requires existence and computation of g''

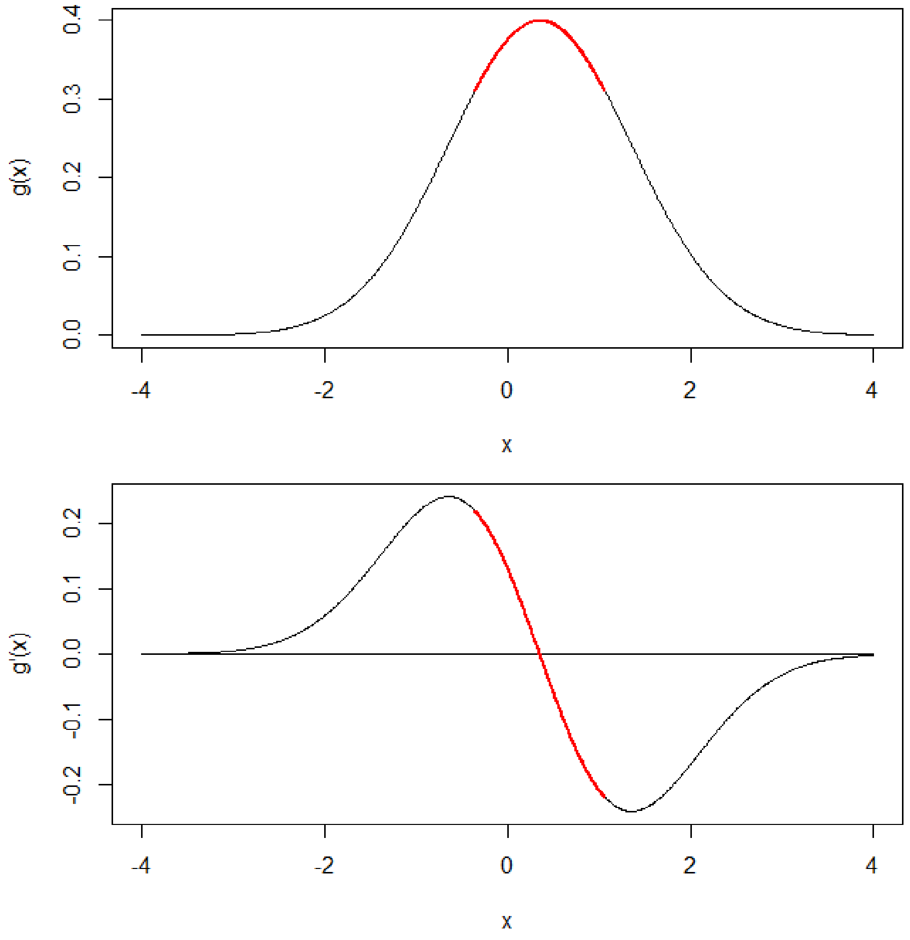



```
x0  -0.2
x1   0.5981
x2   0.337996
x3   0.353557
x4   0.353553
x5   0.353553
STOP
```

Univariate Newton(-Raphson)

- $x^{(t+1)} = x^{(t)} - g'\big(x^{(t)}\big)/g''(x^{(t)})$

- What about the starting value $x^{(0)}$?



Univariate secant method

- x real number, $g: \mathbb{R} \to \mathbb{R}$ once differentiable function
- Search x^* with $g(x^*) = \max g(x)$ by searching x^* with $g'(x^*) = 0$
- Recall: The Newton-iteration works as:

$$x^{(t+1)} = x^{(t)} - g'\left(x^{(t)}\right)/g''(x^{(t)})$$

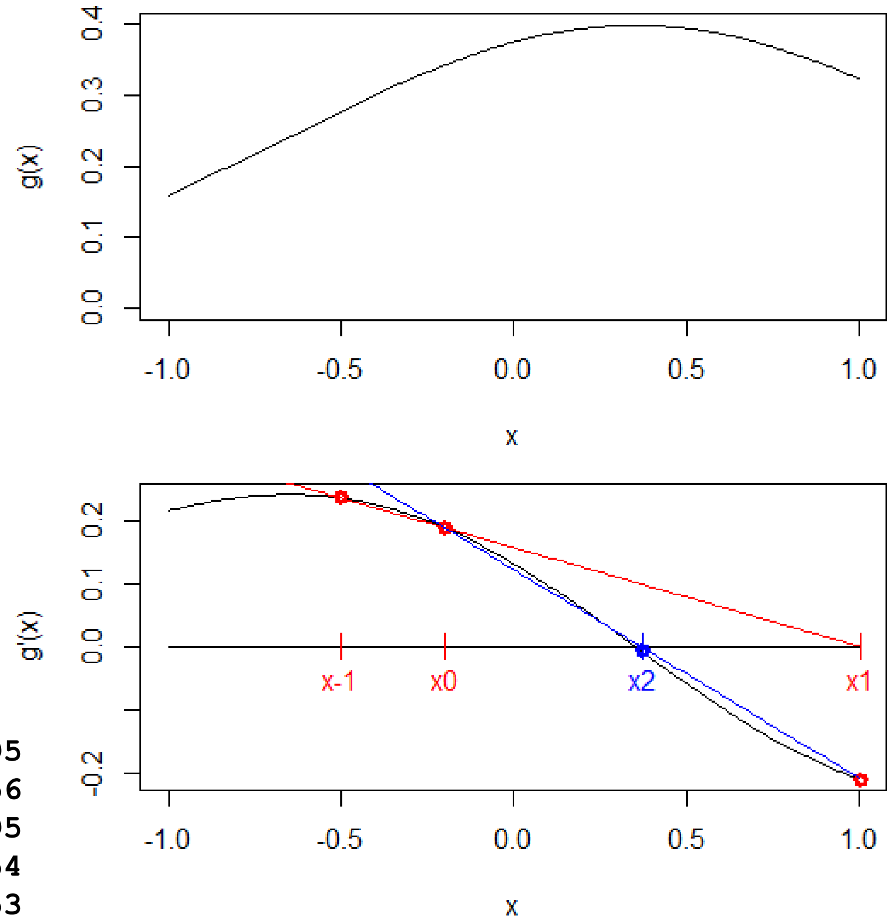
- Need to compute g'' which might be difficult. Instead:
- Approximate $g''(x^{(t)})$ by $\left[g'\left(x^{(t)}\right) - g'(x^{(t-1)})\right]/\left(x^{(t)} - x^{(t-1)}\right)$

Univariate secant method

- $x^{(t+1)} = x^{(t)} - g'\left(x^{(t)}\right)\dfrac{x^{(t)}-x^{(t-1)}}{g'\left(x^{(t)}\right)-g'\left(x^{(t-1)}\right)}$
- Start with $x^{(0)}$ and $x^{(-1)}$
- Secant through $x^{(0)}$ and $x^{(-1)}$ determines $x^{(1)}$
- Secant through $x^{(1)}$ and $x^{(0)}$ determines $x^{(2)}$
- …
- until stopping crit. fulfilled

- Quite fast
- No 2nd derivative necessary

```
x0  -0.2
x1   1.006995
x2   0.371656
x3   0.349095
x4   0.353554
x5   0.353553
x6   0.353553
STOP
```



Convergence speed of optimisation algorithms

Convergence order

- Convergence speed can be quantified by q and c as follows:
  - Let $\varepsilon^{(t)} = x^{(t)} - x^*$,
  - Find q and c such that $\lim_{t\to\infty} \varepsilon^{(t+1)}/(\varepsilon^{(t)})^q = c$

Convergence rate

Intuitively, $\varepsilon^{(t+1)} \approx c \cdot (\varepsilon^{(t)})^q$

- ε = 1, 0.5, 0.25, 0.125, 0.063, 0.031, … ➡ q=1, c=0.5,
- ε = 1, 0.1, 0.01, 0.001, 0.0001, … ➡ q=1, c=0.1,
- If q=1, we say that convergence is ”linear”
- ε = 1, 0.5, 0.125, 0.008, 0.00003, … ➡ q=2, c=0.5.
- If q=2, we say that convergence is ”quadratic”

Determine empirically convergence rate (and order) of optimisation algorithms

- You have a given optimisation algorithm and you have determined or know the maximiser $\boldsymbol{x}^*$. To check convergence speed in an optimisation-run, you can calculate

$$D^{(t)} = \frac{\left|x^{(t)} - x^*\right|}{\left|x^{(t-1)} - x^*\right|}$$

(see Givens and Hoeting, 2013, page 101/102, for an example)

- If $D^{(t)} \rightarrow 1$, there is not even linear convergence (bad, order q<1),
  If $D^{(t)} \rightarrow c \in (0,1)$, linear convergence (order q=1) with rate c,
  If $D^{(t)} \rightarrow 0$, better than linear convergence (order q>1).

Comparison of univariate optimisation methods

Bisection	Secant	Newton
g' required	g' required	g'' required
finds always an optimum between a_0 and b_0 (but could be local)	converges only when the two starting values ”close” to optimum	converges only when starting value ”close” to optimum
slow q=1	$q = \frac{1+\sqrt{5}}{2} = 1.62$	fast q=2

Today's schedule

- Univariate Optimisation (bi-section, Newton, secant)
- Multivariate Optimisation
  - Analytical opt.
  - Newton
  - Steepest ascent
  - Accelerated steepest ascent
  - Quasi-Newton

Multivariate optimisation – gradient and Hessian

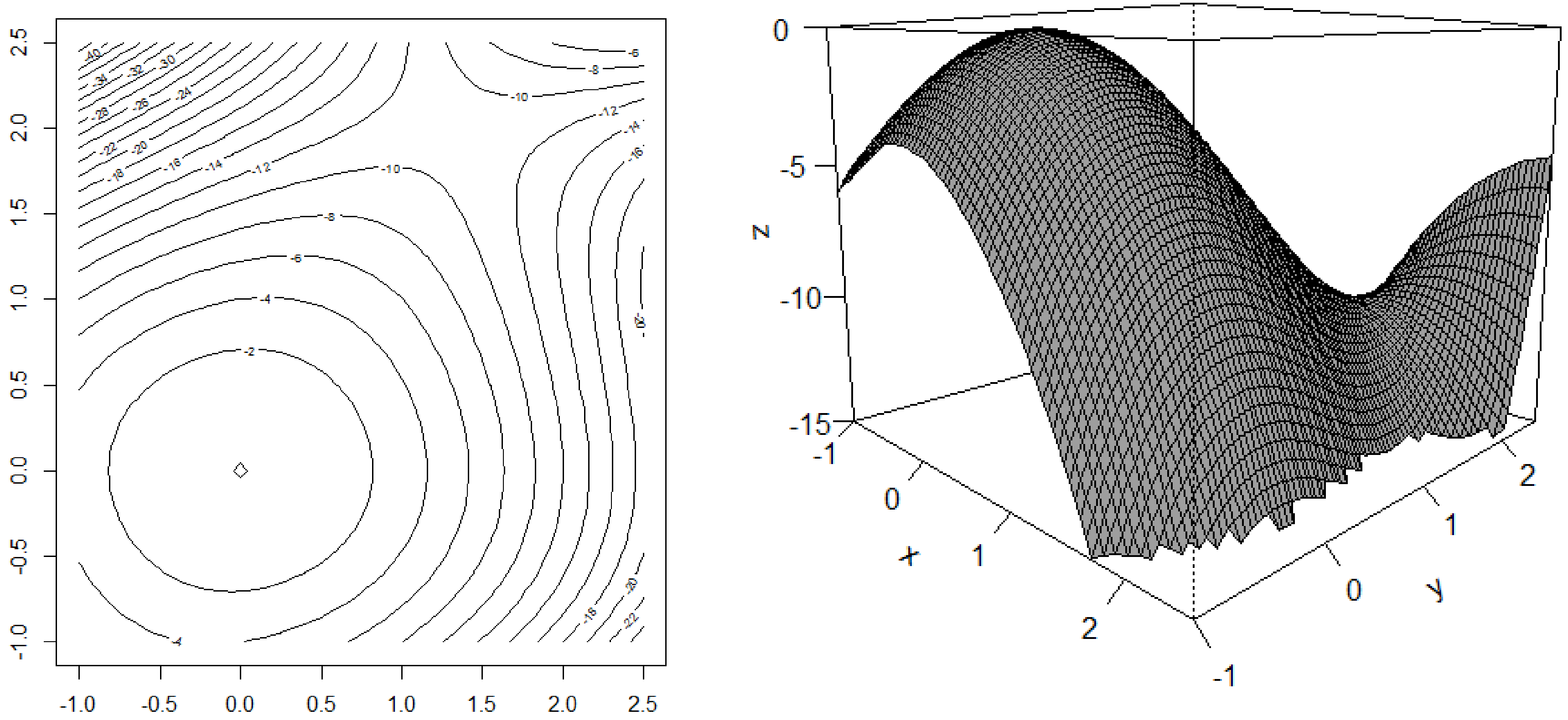
- $\boldsymbol{g}\begin{pmatrix} x_1 \\ \vdots \\ x_p \end{pmatrix}$ is a real-valued function

- $\boldsymbol{g}'\begin{pmatrix} x_1 \\ \vdots \\ x_p \end{pmatrix} = \begin{pmatrix} \frac{\partial g}{\partial x_1}(\boldsymbol{x}) \\ \vdots \\ \frac{\partial g}{\partial x_p}(\boldsymbol{x}) \end{pmatrix}$ is the gradient, $\boldsymbol{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_p \end{pmatrix}$

- $\boldsymbol{g}''\begin{pmatrix} x_1 \\ \vdots \\ x_p \end{pmatrix} = \begin{pmatrix} \frac{\partial g}{\partial x_1 \partial x_1}(\boldsymbol{x}) & \cdots & \frac{\partial g}{\partial x_1 \partial x_p}(\boldsymbol{x}) \\ \vdots & & \vdots \\ \frac{\partial g}{\partial x_1 \partial x_p}(\boldsymbol{x}) & \cdots & \frac{\partial g}{\partial x_p \partial x_p}(\boldsymbol{x}) \end{pmatrix}$ is the Hessian matrix

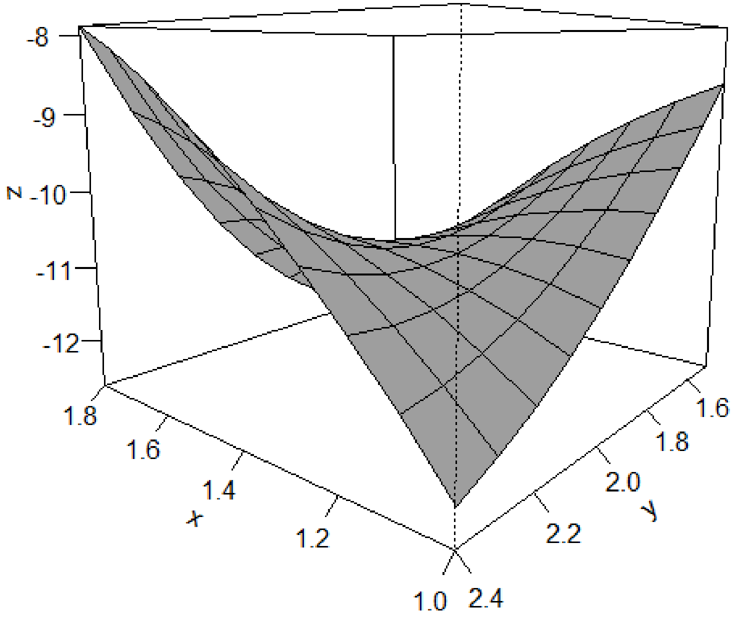
Bivariate optimisation – visualisation

- $g\begin{pmatrix} x \\ y \end{pmatrix} = -3x^2 - 4y^2 + xy^3$



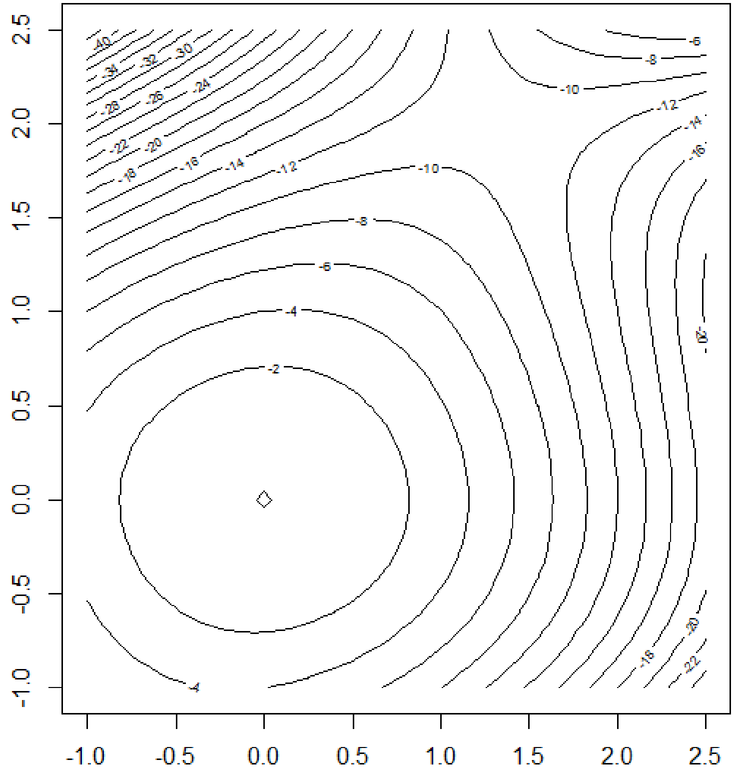
Figures can be drawn using R-core-functions **`contour`** and **`persp`**

Multivariate optimisation – saddle points



Multivariate optimisation – analytical optimisation

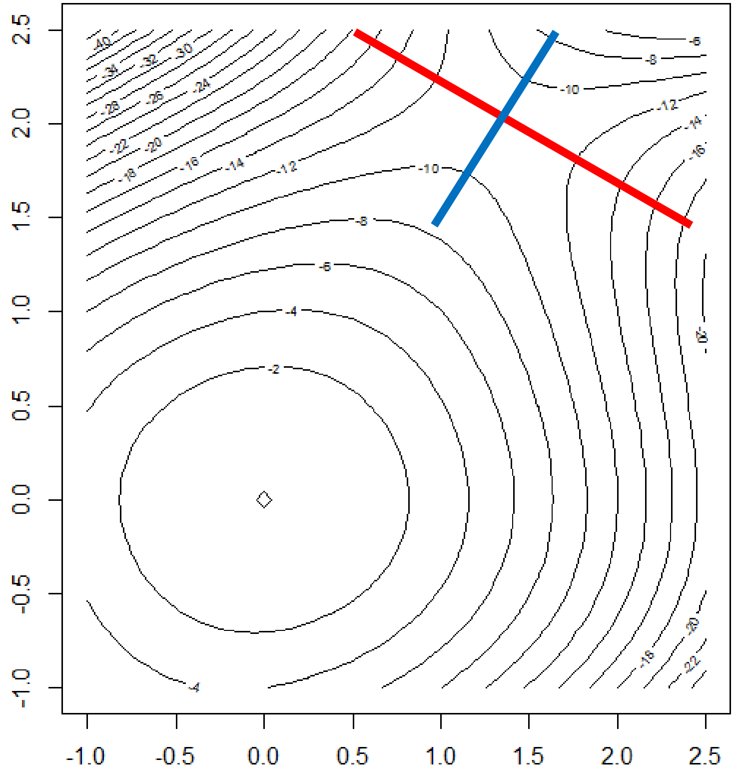
- $\boldsymbol{g}\begin{pmatrix} x \\ y \end{pmatrix} = -3x^2 - 4y^2 + xy^3$
- $\boldsymbol{g}'\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -6x + y^3 \\ -8y + 3xy^2 \end{pmatrix}$
- $\boldsymbol{g}''\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -6 & 3y^2 \\ 3y^2 & -8 + 6xy \end{pmatrix}$



- See calculation in following document: `AdvCompStat_AnalytOpt.pdf`
- Maximum at (0,0), saddle point at (4/3,2)

Saddle point and eigenvectors of the Hessian

- $\boldsymbol{g}\begin{pmatrix} x \\ y \end{pmatrix} = -3x^2 - 4y^2 + xy^3$
- Saddle point at (4/3,2)



- $\boldsymbol{g}'\begin{pmatrix} 4/3 \\ 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$
- $\boldsymbol{g}''\begin{pmatrix} 4/3 \\ 2 \end{pmatrix} = \begin{pmatrix} -6 & 12 \\ 12 & 8 \end{pmatrix}$

- Eigenvalues 14.89, -12.89; eigenvectors $\begin{pmatrix} 0.498 \\ 0.867 \end{pmatrix}$, $\begin{pmatrix} -0.867 \\ 0.498 \end{pmatrix}$

Multivariate Newton

- $\boldsymbol{x}$ p–dimensional vector, g: $\mathbb{R}^p \to \mathbb{R}$ function
- We search $\boldsymbol{x}^*$ with $g(\boldsymbol{x}^*) = \max g(\boldsymbol{x})$
- Now, $\boldsymbol{g}'$ is p-dim. vector and $\boldsymbol{g}''$ is pxp-matrix (“Hessian”)

- The multivariate version of the Newton method is motivated by the multivariate Taylor expansion

$$0 = \boldsymbol{g}'(\boldsymbol{x}^*) \approx \boldsymbol{g}'\left(\boldsymbol{x}^{(t)}\right) + \boldsymbol{g}''\left(\boldsymbol{x}^{(t)}\right)\left(\boldsymbol{x}^* - \boldsymbol{x}^{(t)}\right)$$

- The Newton-iteration works as:

$$\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} - \left(\boldsymbol{g}''\left(\boldsymbol{x}^{(t)}\right)\right)^{-1} \boldsymbol{g}'\left(\boldsymbol{x}^{(t)}\right)$$

Multivariate Newton

- $\boldsymbol{x}^{(\boldsymbol{t}+\mathbf{1})} = \boldsymbol{x}^{(\boldsymbol{t})} - \left(\boldsymbol{g}''\left(\boldsymbol{x}^{(\boldsymbol{t})}\right)\right)^{-1} \boldsymbol{g}'\left(\boldsymbol{x}^{(\boldsymbol{t})}\right)$
- Example:

  Let g_1 be the density of $N\left(\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}0.6 & 0\\0 & 0.6\end{pmatrix}\right)$, g_2 be density of $N\left(\begin{pmatrix}1.5\\1.2\end{pmatrix}, \begin{pmatrix}0.5 & 0\\0 & 0.5\end{pmatrix}\right)$,

  and $g = \frac{g_1+g_2}{2}$, i.e.

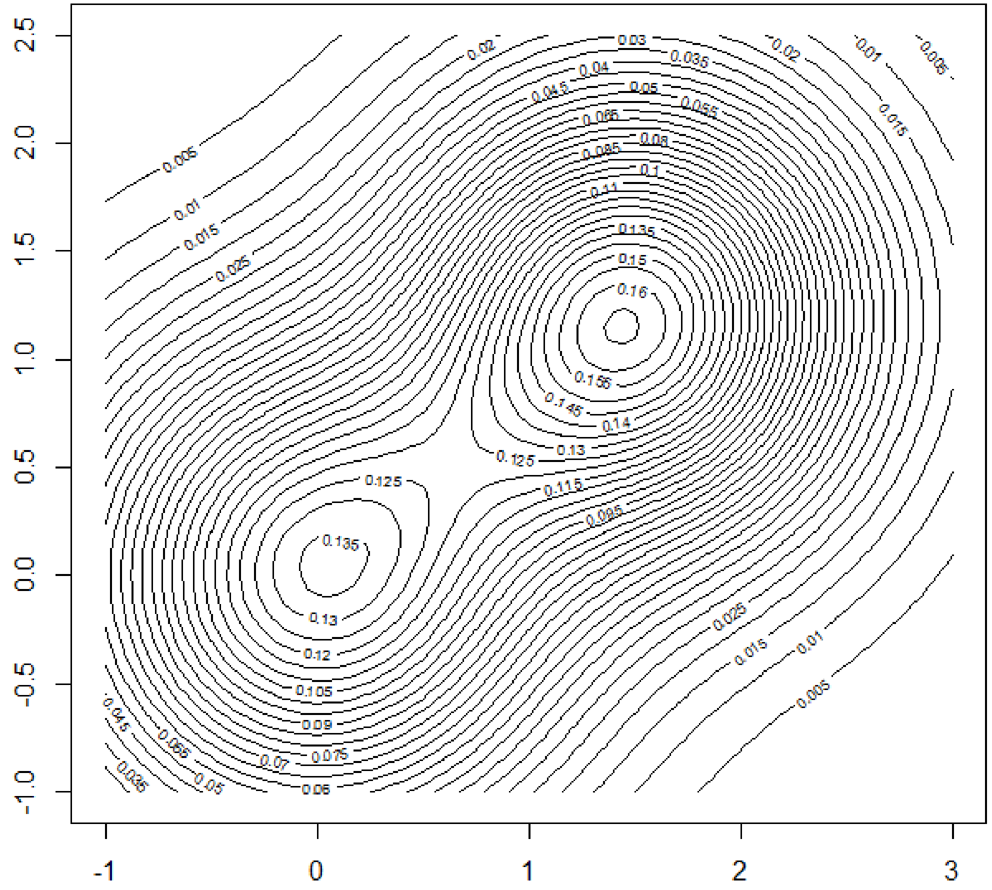
  $$g(x_1, x_2) = \frac{1}{4\pi}\left(\frac{1}{0.6}e^{-({x_1}^2+{x_2}^2)/1.2} + \frac{1}{0.5}e^{-\left((x_1-1.5)^2+(x_2-1.2)^2\right)}\right)$$

  (g is density of a normal mixture distribution).
- Compute point $\mathbf{x}=(x_1,x_2)$ where density $g(\mathbf{x})$ maximal.

- Do you have a guess?

Multivariate Newton

- $g(x_1, x_2) = \frac{1}{4\pi}\left(\frac{1}{0.6}e^{-({x_1}^2+{x_2}^2)/1.2} + \frac{1}{0.5}e^{-\left((x_1-1.5)^2+(x_2-1.2)^2\right)}\right)$



Multivariate Newton

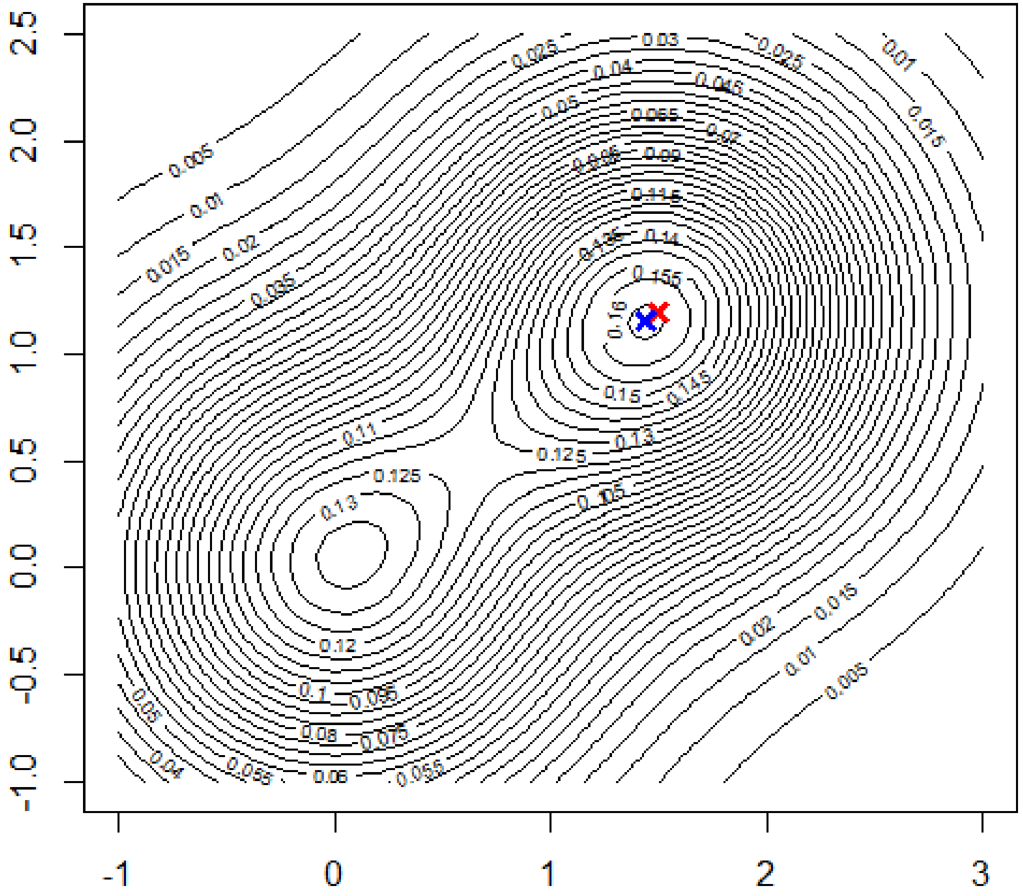
- $\boldsymbol{x}^{(\boldsymbol{t}+\mathbf{1})} = \boldsymbol{x}^{(\boldsymbol{t})} - \left(\boldsymbol{g}''\left(\boldsymbol{x}^{(\boldsymbol{t})}\right)\right)^{-1} \boldsymbol{g}'\left(\boldsymbol{x}^{(\boldsymbol{t})}\right)$
- We need ***g'*** and ***g''*** of

$$g(x_1, x_2) = \frac{1}{4\pi}\left(\frac{1}{0.6} e^{-\left({x_1}^2 + {x_2}^2\right)/(2\cdot 0.6)} + \frac{1}{0.5} e^{-\left((x_1-1.5)^2+(x_2-1.2)^2\right)}\right)$$

- $\frac{\partial g}{\partial x_1}(x_1, x_2) = \frac{1}{4\pi}\left(\frac{-2x_1}{1.2\cdot 0.6} e^{-\left({x_1}^2+{x_2}^2\right)/1.2} + \frac{-2(x_1-1.5)}{0.5} e^{-\left((x_1-1.5)^2+(x_2-1.2)^2\right)}\right)$
- $\frac{\partial g}{\partial x_2}(x_1, x_2) = \frac{1}{4\pi}\left(\frac{-2x_2}{1.2\cdot 0.6} e^{-\left({x_1}^2+{x_2}^2\right)/1.2} + \frac{-2(x_2-1.2)}{0.5} e^{-\left((x_1-1.5)^2+(x_2-1.2)^2\right)}\right)$
- $\boldsymbol{g}'(x_1, x_2) = \begin{pmatrix} \frac{\partial g}{\partial x_1}(x_1, x_2) \\ \frac{\partial g}{\partial x_2}(x_1, x_2) \end{pmatrix}$
- $\frac{\partial^2 g}{\partial^2 x_1}(x_1, x_2) = \ldots$; $\frac{\partial^2 g}{\partial x_1 \partial x_2}(x_1, x_2) = \ldots$; $\frac{\partial^2 g}{\partial^2 x_2}(x_1, x_2) = \ldots$ give ***g''***

Multivariate Newton

- $g(x_1, x_2) = \frac{1}{4\pi}\left(\frac{1}{0.6}e^{-({x_1}^2+{x_2}^2)/1.2} + \frac{1}{0.5}e^{-\left((x_1-1.5)^2+(x_2-1.2)^2\right)}\right)$

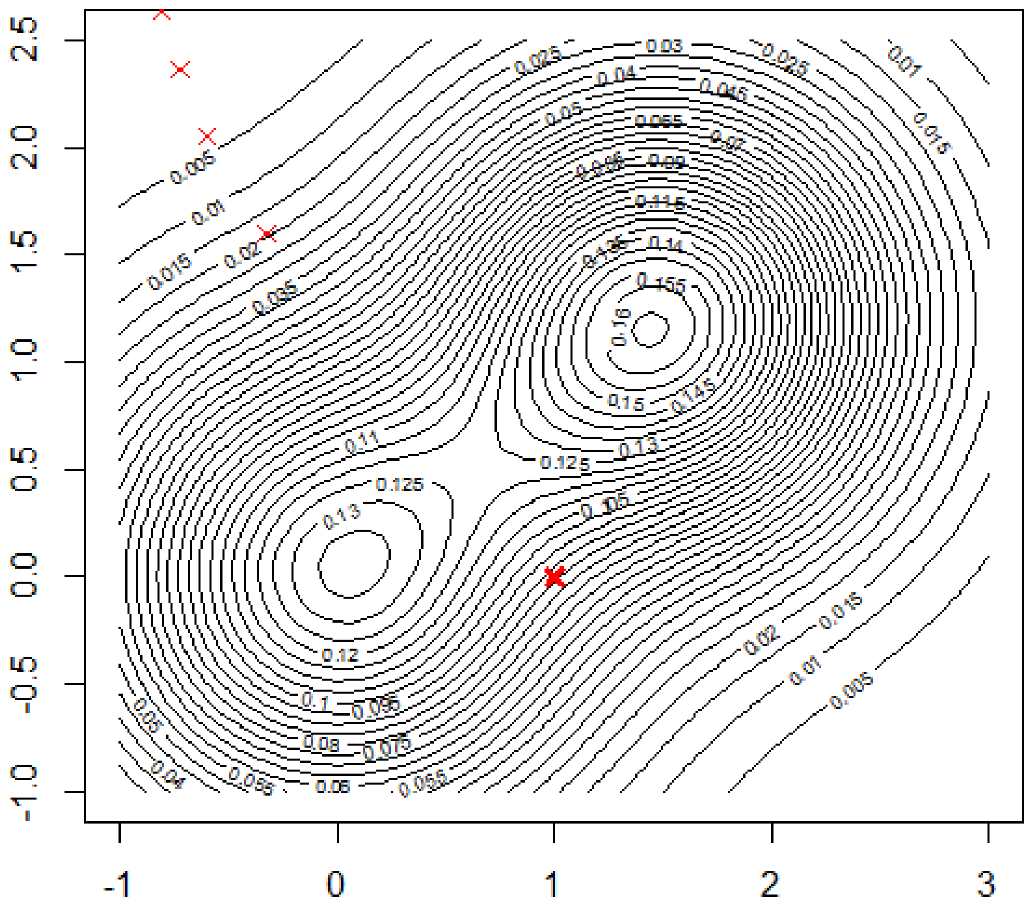


- Start with $\boldsymbol{x}^{(0)} = \begin{pmatrix} \mathbf{1.5} \\ \mathbf{1.2} \end{pmatrix}$
- $\boldsymbol{g}'(\boldsymbol{x}^{(0)}) = \begin{pmatrix} -0.0153 \\ -0.0123 \end{pmatrix}$
- $\boldsymbol{g}''(\boldsymbol{x}^{(0)}) = \begin{pmatrix} -0.2902 & 0.0306 \\ 0.0306 & -0.3040 \end{pmatrix}$
- $\left(\boldsymbol{g}''(\boldsymbol{x}^{(0)})\right)^{-1}\boldsymbol{g}'(\boldsymbol{x}^{(0)}) = \begin{pmatrix} \mathbf{0.058} \\ \mathbf{0.046} \end{pmatrix}$
- $\boldsymbol{x}^{(1)} = \begin{pmatrix} \mathbf{1.5} \\ \mathbf{1.2} \end{pmatrix} - \begin{pmatrix} \mathbf{0.058} \\ \mathbf{0.046} \end{pmatrix} = \begin{pmatrix} \mathbf{1.442} \\ \mathbf{1.154} \end{pmatrix}$

$\boldsymbol{x}^{(2)} = \boldsymbol{x}^{*} = \begin{pmatrix} \mathbf{1.441} \\ \mathbf{1.153} \end{pmatrix}$

Multivariate Newton

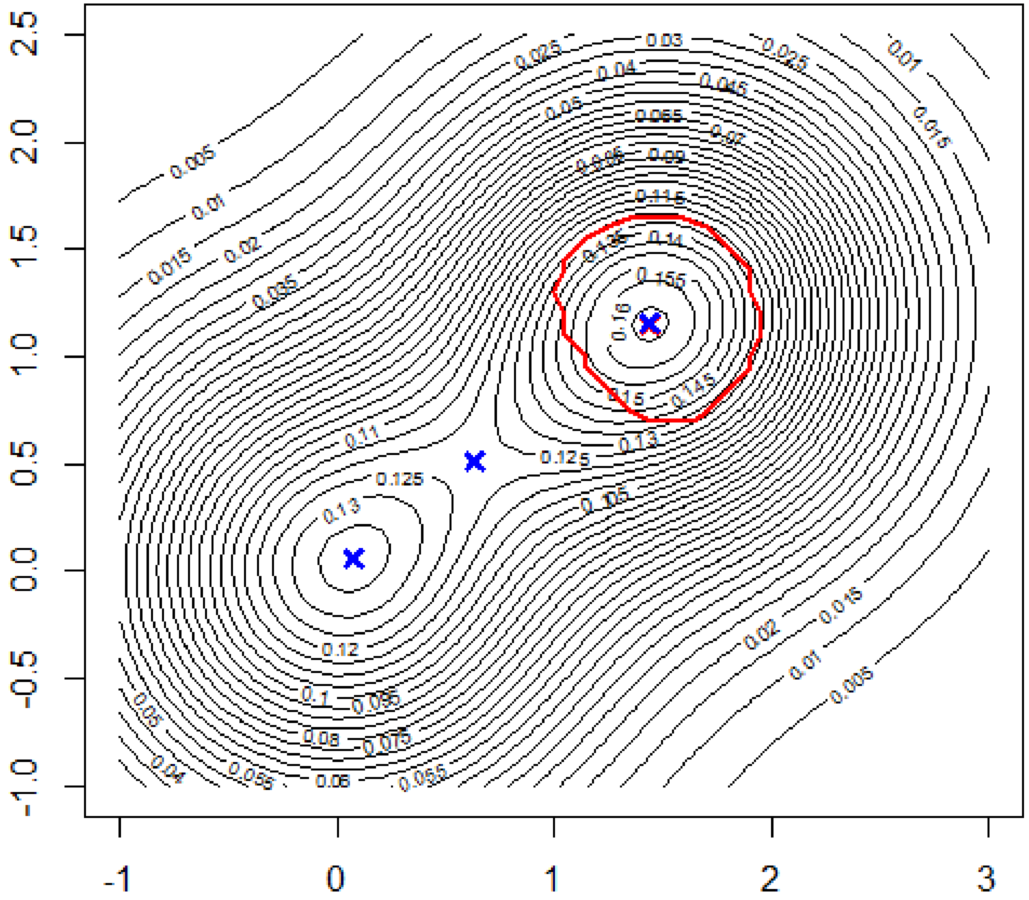
- $g(x_1, x_2) = \frac{1}{4\pi}\left(\frac{1}{0.6}e^{-({x_1}^2+{x_2}^2)/1.2} + \frac{1}{0.5}e^{-\left((x_1-1.5)^2+(x_2-1.2)^2\right)}\right)$



- Start with $\boldsymbol{x}^{(0)} = \begin{pmatrix} \mathbf{1} \\ \mathbf{0} \end{pmatrix}$
- $\boldsymbol{g}'(\boldsymbol{x}^{(0)}) = \begin{pmatrix} -0.0667 \\ +0.0705 \end{pmatrix}$
- $\boldsymbol{g}''(\boldsymbol{x}^{(0)}) = \begin{pmatrix} 0.0347 & 0.0705 \\ 0.0705 & 0.0144 \end{pmatrix}$
- $\left(\boldsymbol{g}''(\boldsymbol{x}^{(0)})\right)^{-1} \boldsymbol{g}'(\boldsymbol{x}^{(0)}) = \begin{pmatrix} \mathbf{1.33} \\ \mathbf{-1.60} \end{pmatrix}$
- $\boldsymbol{x}^{(1)} = \begin{pmatrix} \mathbf{1} \\ \mathbf{0} \end{pmatrix} - \begin{pmatrix} \mathbf{1.33} \\ \mathbf{-1.6} \end{pmatrix} = \begin{pmatrix} \mathbf{-0.33} \\ \mathbf{1.6} \end{pmatrix}$

Multivariate Newton

- $g(x_1, x_2) = \frac{1}{4\pi}\left(\frac{1}{0.6} e^{-({x_1}^2+{x_2}^2)/1.2} + \frac{1}{0.5} e^{-\left((x_1-1.5)^2+(x_2-1.2)^2\right)}\right)$



- Only starting values within the red-marked area converge to the right global maximum
- Convergence very quick
- Other starting values converge to the local maximum or saddle point (both blue-marked) or diverge while searching for a minimum

Stopping criteria

- Stopping criterion e.g. $\left(\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^{(t)}\right)^T \left(\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^{(t)}\right) < \epsilon$
- Other stopping criteria:
  - Absolut stopping criterion, $\left\|\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^{(t)}\right\| < \epsilon$,
  - Relative stopping criterion, $\left\|\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^{(t)}\right\| / \left\|\boldsymbol{x}^{(t+1)}\right\| < \epsilon$,
  - Modified rel. stopping crit., $\frac{\left\|\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^{(t)}\right\|}{\left\|\boldsymbol{x}^{(t+1)}\right\| + \varepsilon} < \varepsilon$
  - Different norms $\|\cdot\|$ can be used

Today's schedule

- Univariate Optimisation (bi-section, Newton, secant)
- Multivariate Optimisation
  - Analytical opt.
  - Newton
  - Steepest ascent
  - Accelerated steepest ascent
  - Quasi-Newton

Steepest ascent method

- When using Newton method, it is not guaranteed that *g(x)* increases in each step
- To compute the Hessian ***g''*** can be difficult
- A method forcing improvements in each step is the steepest ascent method

$$\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} - \left(\boldsymbol{g}''\left(\boldsymbol{x}^{(t)}\right)\right)^{-1} \boldsymbol{g}'\left(\boldsymbol{x}^{(t)}\right)$$

$$\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \alpha^{(t)} \boldsymbol{I}\, \boldsymbol{g}'\left(\boldsymbol{x}^{(t)}\right)$$

- Other choices instead ***I*** in formula above possible
- We know that *g* will increase for small α

Backtracking line search (for steepest ascent)

$$\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \alpha^{(t)}\boldsymbol{I}\,\boldsymbol{g}'\left(\boldsymbol{x}^{(t)}\right)$$

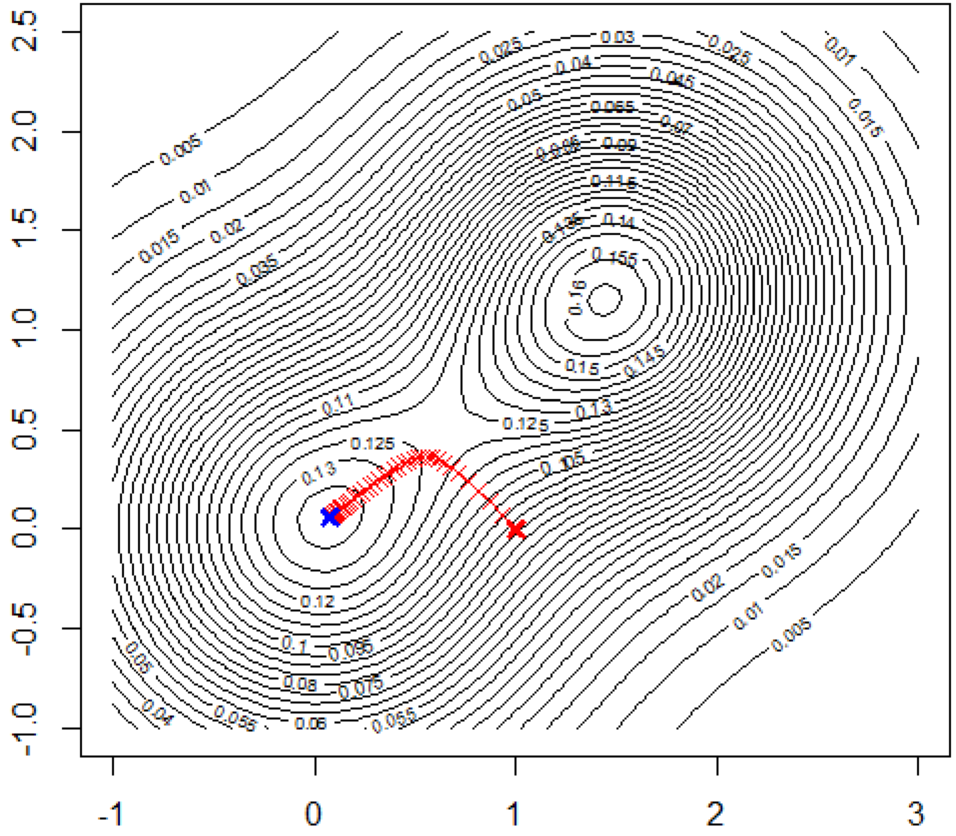
- We know that g will increase for small α

- Try $\alpha^{(t)} = 1$ first
- If g decreases, half $\alpha^{(t)}$ until $g(\boldsymbol{x}^{(t+1)})$ increases

- More sophisticated is to search α such that g becomes maximal

Steepest ascent

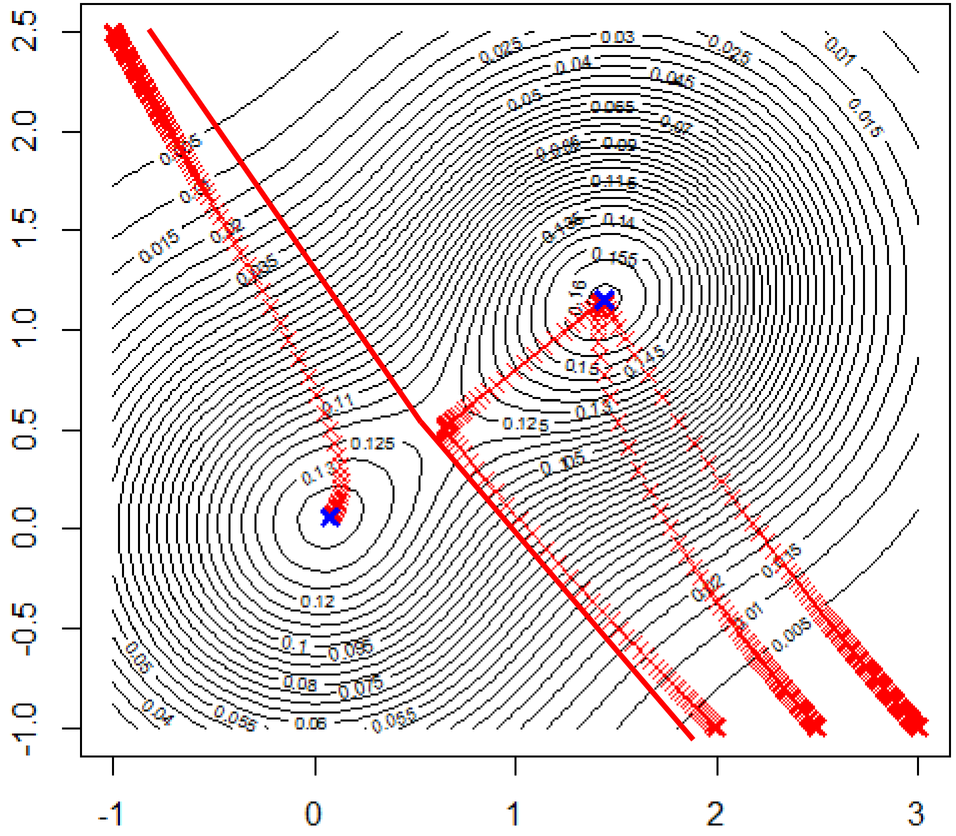
- $g(x_1, x_2) = \frac{1}{4\pi}\left(\frac{1}{0.6} e^{-({x_1}^2+{x_2}^2)/1.2} + \frac{1}{0.5} e^{-\left((x_1-1.5)^2+(x_2-1.2)^2\right)}\right)$



- Start with $\boldsymbol{x}^{(\mathbf{0})} = \begin{pmatrix}\mathbf{1}\\ \mathbf{0}\end{pmatrix}$
- $\boldsymbol{g}'\left(\boldsymbol{x}^{(\mathbf{0})}\right) = \begin{pmatrix}-0.0667\\ +0.0705\end{pmatrix}$
- $\boldsymbol{x}^{(1)} = \begin{pmatrix}\mathbf{1}\\ \mathbf{0}\end{pmatrix} + \alpha^{(0)}\begin{pmatrix}-0.0667\\ +0.0705\end{pmatrix} = \begin{pmatrix}\mathbf{0.9333}\\ \mathbf{0.0705}\end{pmatrix}$

Steepest ascent

- $g(x_1, x_2) = \frac{1}{4\pi}\left(\frac{1}{0.6}e^{-({x_1}^2+{x_2}^2)/1.2} + \frac{1}{0.5}e^{-\left((x_1-1.5)^2+(x_2-1.2)^2\right)}\right)$

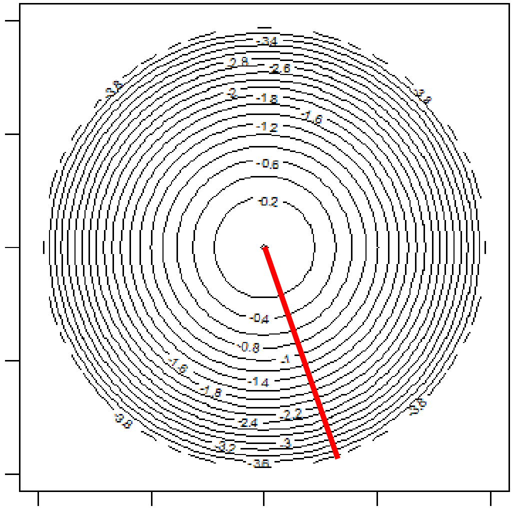


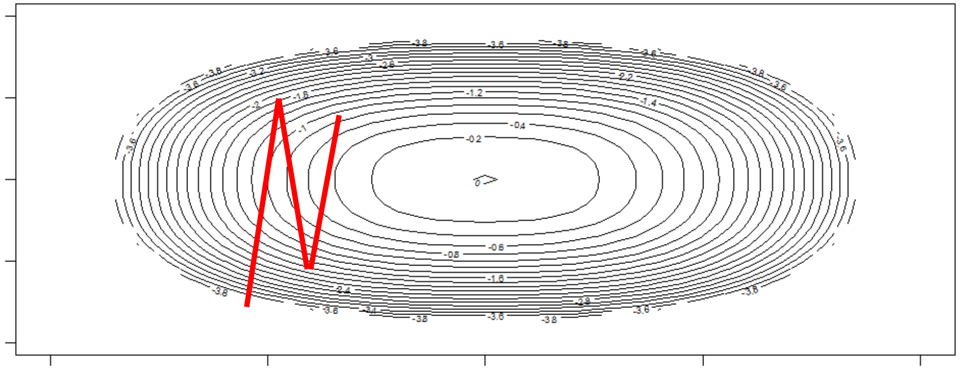
- Start with $\boldsymbol{x}^{(0)} = \begin{pmatrix}-1\\2.5\end{pmatrix}, \begin{pmatrix}2\\-1\end{pmatrix}, \begin{pmatrix}2.5\\-1\end{pmatrix}, \begin{pmatrix}3\\-1\end{pmatrix}$
- All these paths converge to either the global or local maximum
- Convergence is much slower than for Newton
- Depending on convergence criterion and alpha-rule, convergence not always guaranteed

Steepest ascent

Globen, Stockholm – by Arild Vågen, CC BY-SA 4.0, https://commons.wikimedia.org/wiki/File:Globen_September_2014_02.jpg

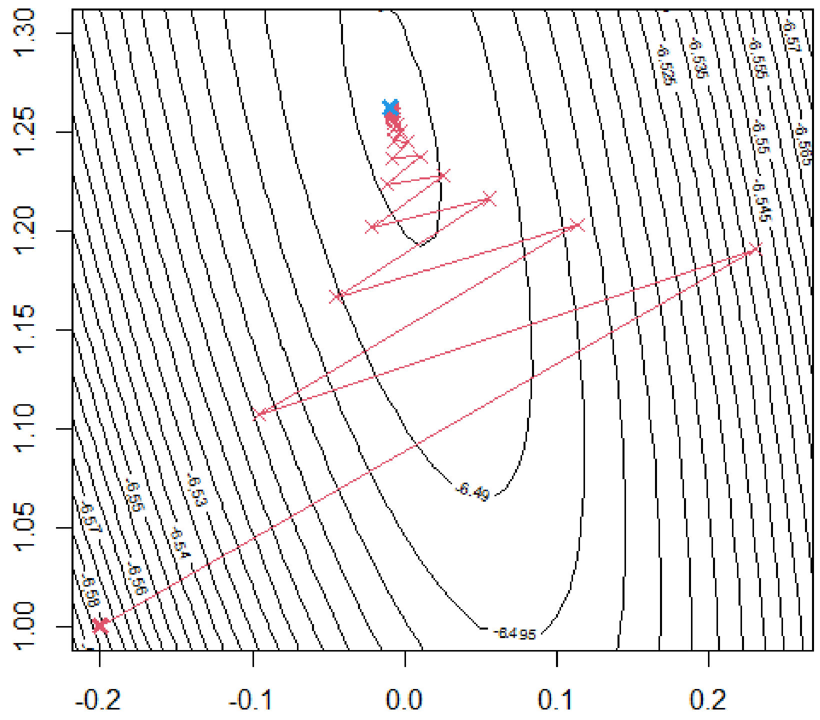
Uluru, Australia – by Stuart Edwards, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=1650537





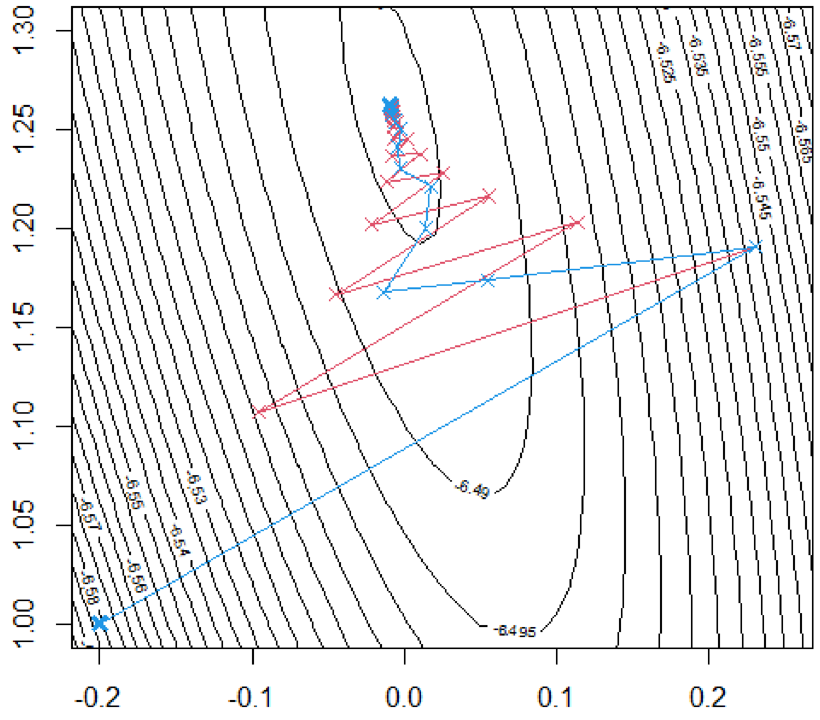
Steepest ascent: idea for acceleration

- Example: ML computation for a two-parameter model with steepest ascent, with fixed $\alpha^{(t)} = 0.667$ (no backtracking)
- Zick-zack path is common and slows down convergence
- Idea to reduce/avoid this issue: use information from last iteration about "momentum" of search path
- Called: **Accelerated steepest ascent** (or steepest ascent with momentum)



Accelerated steepest ascent: Polyak's momentum

- $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \alpha^{(t)}\, \boldsymbol{g}'\left(\boldsymbol{x}^{(t)}\right) + \beta\left(\boldsymbol{x}^{(t)} - \boldsymbol{x}^{(t-1)}\right)$
- Polyak="gradient+momentum"
- Steepest ascent ($\alpha^{(t)} = 0.667$)
- with momentum ($\beta = 0.35$)
- Called also *heavy-ball method*
- Adding momentum reduces number of iterations from 31 to 21 in this example
- Works well in many situations
- Examples exist where Polyak's method fails to converge



Accelerated steepest ascent: Nesterov's momentum

- $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \alpha^{(t)}\, \boldsymbol{g}'\left(\boldsymbol{x}^{(t)} + \beta(\boldsymbol{x}^{(t)} - \boldsymbol{x}^{(t-1)})\right) + \beta(\boldsymbol{x}^{(t)} - \boldsymbol{x}^{(t-1)})$
- Nesterov = “lookahead gradient + momentum”
- Ideally, this method has the capacity
  - to dampen oscillations and
  - to accelerate if the search path is in right direction
- Nesterov’s accelerated ascent has better convergence rate as steepest ascent

Parametrisation of accelerated methods

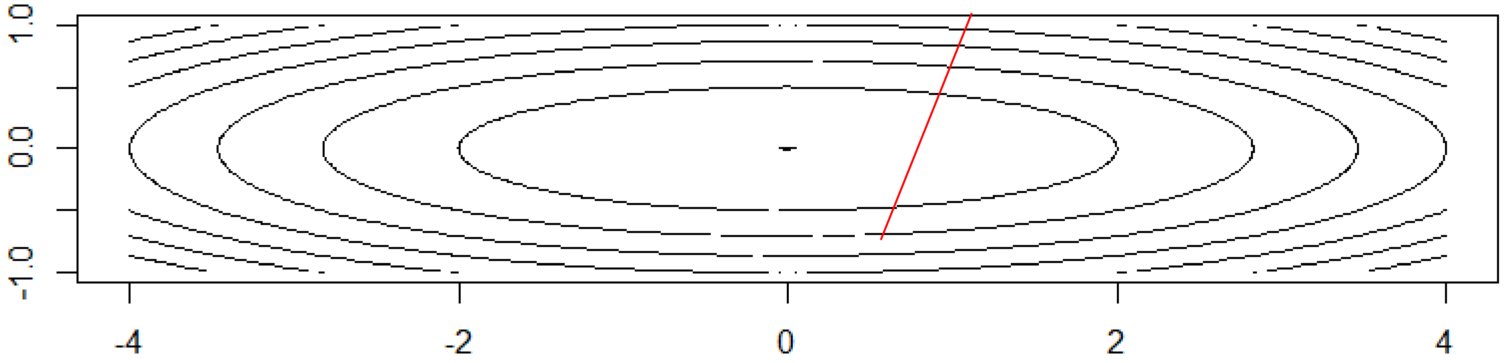
- Polyak's accelerated steepest ascent
  $$\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \alpha \boldsymbol{g}'\left(\boldsymbol{x}^{(t)}\right) + \beta\left(\boldsymbol{x}^{(t)} - \boldsymbol{x}^{(t-1)}\right)$$
  can be written also as
  $$\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \alpha \boldsymbol{v}^{(t+1)}$$
  $$\boldsymbol{v}^{(t+1)} = \beta \boldsymbol{v}^{(t)} + \boldsymbol{g}'\left(\boldsymbol{x}^{(t)}\right)$$
- Nesterov's accelerated steepest ascent
  $$\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \alpha \boldsymbol{g}'\left(\boldsymbol{x}^{(t)} + \beta\left(\boldsymbol{x}^{(t)} - \boldsymbol{x}^{(t-1)}\right)\right) + \beta\left(\boldsymbol{x}^{(t)} - \boldsymbol{x}^{(t-1)}\right)$$
  can be written also as
  $$\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \alpha \boldsymbol{v}^{(t+1)}$$
  $$\boldsymbol{v}^{(t+1)} = \beta \boldsymbol{v}^{(t)} + \boldsymbol{g}'\left(\boldsymbol{x}^{(t)} + \alpha\beta \boldsymbol{v}^{(t)}\right)$$

Steepest ascent: optimal choice of step size

- $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(\boldsymbol{t})} + \alpha \boldsymbol{g}'\left(\boldsymbol{x}^{(\boldsymbol{t})}\right)$
- Example:
  $g(\boldsymbol{x}) = -\frac{1}{2}\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{x}$, $\boldsymbol{A}$ symmetric pxp and of full rank
- $\boldsymbol{g}'(\boldsymbol{x}) = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}$
- To keep things simple (and to avoid a change of basis and some more linear algebra…), we use $\boldsymbol{b} = \boldsymbol{0}$, $\boldsymbol{A}$=diagonal (i.e. eigenvalues in diagonal), p=2
- $\boldsymbol{A} = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$, $\boldsymbol{g}'(\boldsymbol{x}) = \begin{pmatrix} -\lambda_1 x_1 \\ -\lambda_2 x_2 \end{pmatrix}$, $\lambda_1, \lambda_2 > 0$
- Then, steepest ascent is:
- $x_i^{(t+1)} = (1 - \alpha\lambda_i)x_i^{(t)} = (1 - \alpha\lambda_i)^{t+1}x_i^{(0)}$

Steepest ascent: optimal choice of step size

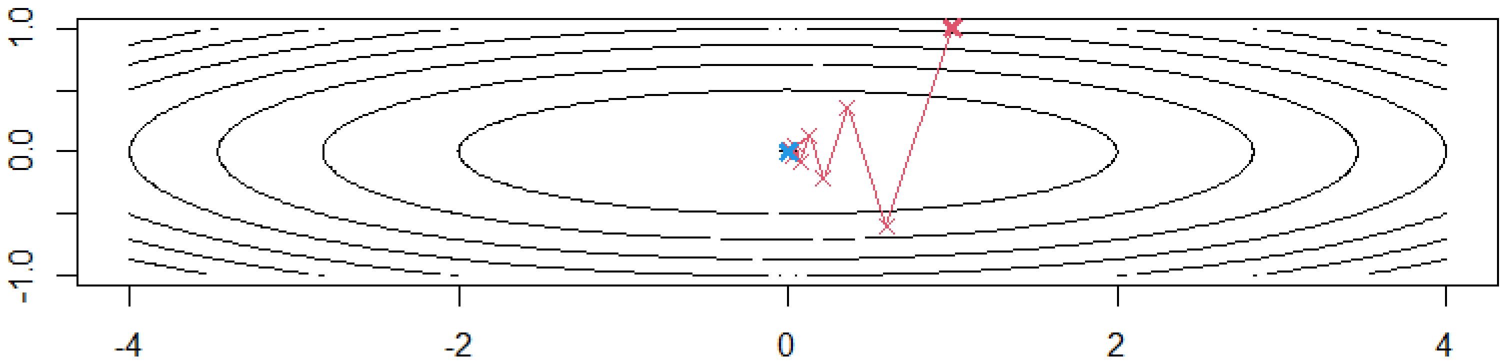
- $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \alpha \boldsymbol{g}'(\boldsymbol{x}^{(t)})$
- Example: $g(\boldsymbol{x}) = -\frac{1}{2}\boldsymbol{x}^T \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \boldsymbol{x}$, $\boldsymbol{g}'(\boldsymbol{x}) = \begin{pmatrix} -\lambda_1 x_1 \\ -\lambda_2 x_2 \end{pmatrix}$, $\boldsymbol{x}^{(0)} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$
- Steepest ascent: $x_1^{(t+1)} = (1 - \alpha\lambda_1)^{t+1}, x_2^{(t+1)} = (1 - \alpha\lambda_2)^{t+1}$
- For $\lambda_1 = \frac{1}{2}, \lambda_2 = 2$:



- Fastest convergence attained if α such that $\rho = \max\{|1 - \alpha\lambda_1|, |1 - \alpha\lambda_2|\}$ is as small as possible

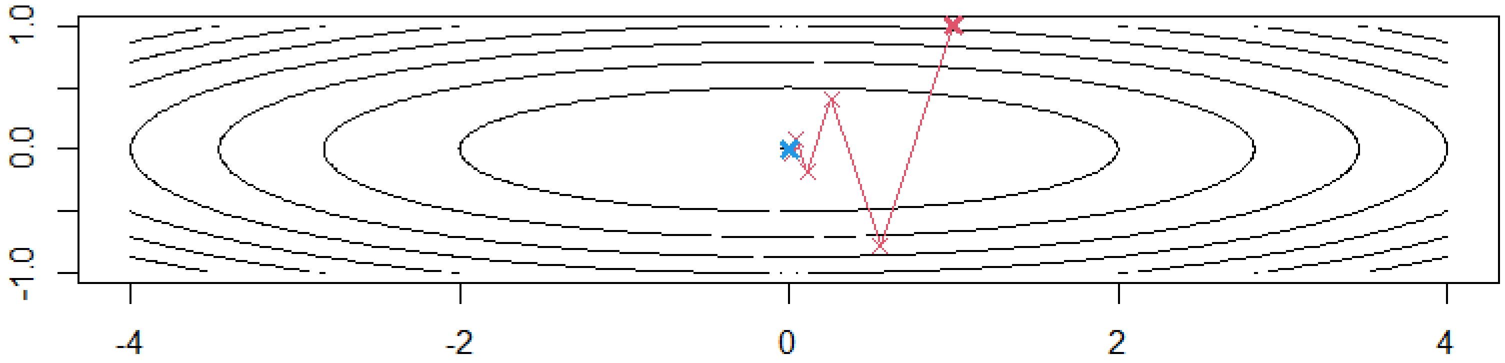
Steepest ascent: optimal choice of step size

- Steepest ascent: $x_1^{(t+1)} = (1 - \alpha\lambda_1)^{t+1}, x_2^{(t+1)} = (1 - \alpha\lambda_2)^{t+1}$
- Fastest convergence attained if α such that $\rho = \max\{|1 - \alpha\lambda_1|, |1 - \alpha\lambda_2|\}$ is as small as possible
- Fulfilled for $\alpha = \frac{2}{\lambda_1 + \lambda_2}$ and then $\rho = \frac{\kappa - 1}{\kappa + 1}$ with $\kappa = \lambda_2/\lambda_1$
- ρ is convergence rate; κ is condition number
- For example, with $\lambda_1 = \frac{1}{2}, \lambda_2 = 2$: $\rho = \frac{3}{5}, \alpha = \frac{4}{5}$.



Accelerated steepest ascent: choice of hyperparameters

- Steepest ascent: convergence rate $\rho = \frac{\kappa-1}{\kappa+1}$ with $\kappa = \frac{\lambda_{max}}{\lambda_{min}}$
- Accelerated steepest ascent:
  - Best convergence rate: $\rho = \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)$
  - Optimal step size: $\alpha = \frac{(1+\rho)^2}{\lambda_{max}} = \frac{(1-\rho)^2}{\lambda_{min}}$
  - Optimal momentum: $\beta = \rho^2$
- For example, with $\lambda_1 = \frac{1}{2}, \lambda_2 = 2$: $\rho = \frac{1}{3}, \alpha = \frac{8}{9}, \beta = \frac{1}{9}$.



(Accelerated) steepest ascent: convergence

- Convergence rate for $\kappa = \frac{\lambda_{max}}{\lambda_{min}}$:
  - Steepest ascent: $\rho = \frac{\kappa-1}{\kappa+1}$
  - Accelerated steepest ascent: $\rho = \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)$

convergence order; here $q = 1$

- $\lim_{t\to\infty} \|\boldsymbol{x}^{(t+1)}-\boldsymbol{x}^*\|/\|\boldsymbol{x}^{(t)}-\boldsymbol{x}^*\|^q = \rho$

convergence rate

- Example $\kappa = 100$ ("ill-conditioned"):
  - $\frac{\kappa-1}{\kappa+1} = \frac{99}{101}; (\frac{\kappa-1}{\kappa+1})^t = 1, 0.98, \dots, \underset{t=10}{0.82}, \dots, \underset{t=100}{0.14}, \dots$
  - $\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} = \frac{9}{11}; (\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1})^t = 1, 0.82, \dots, 0.13, \dots, 1.9 \cdot 10^{-9}, \dots$

Today's schedule

- Univariate Optimisation (bi-section, Newton, secant)
- Multivariate Optimisation
  - Analytical opt.
  - Newton
  - Steepest ascent
  - Accelerated steepest ascent
  - Quasi-Newton

Quasi-Newton

- Steepest ascent and Newton method have iteration
$$\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} - \left(\boldsymbol{M}^{(t)}\right)^{-1} \boldsymbol{g}'\left(\boldsymbol{x}^{(t)}\right)$$
with $\boldsymbol{M}^{(t)} = \boldsymbol{g}''\left(\boldsymbol{x}^{(t)}\right)$ for the Newton method and
with $\left(\boldsymbol{M}^{(t)}\right)^{-1} = -\alpha_t \boldsymbol{I}$ for the steepest ascent method
- A disadvantage of Newton is the need to calculate the Hessian $\boldsymbol{g}''\left(\boldsymbol{x}^{(t)}\right)$ in each iteration
- A disadvantage of steepest ascent is that no information about the curvature is used
- We can monitor the computed gradients $\boldsymbol{g}'\left(\boldsymbol{x}^{(t)}\right)$ and their change gives information about the curvature of g

Quasi-Newton

- Steepest ascent and Newton method have iteration

$$\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} - \left(\boldsymbol{M}^{(t)}\right)^{-1} \boldsymbol{g}'\left(\boldsymbol{x}^{(t)}\right)$$

- Newton ($\boldsymbol{M}^{(t)} = \boldsymbol{g}''\left(\boldsymbol{x}^{(t)}\right)$) was motivated with the multidimensional Taylor expansion

$$\boldsymbol{g}'(\boldsymbol{x}^{*}) \approx \boldsymbol{g}'\left(\boldsymbol{x}^{(t)}\right) + \boldsymbol{g}''\left(\boldsymbol{x}^{(t)}\right)\left(\boldsymbol{x}^{*} - \boldsymbol{x}^{(t)}\right)$$

or

$$\boldsymbol{g}'(\boldsymbol{x}^{*}) - \boldsymbol{g}'\left(\boldsymbol{x}^{(t)}\right) \approx \boldsymbol{g}''\left(\boldsymbol{x}^{(t)}\right)\left(\boldsymbol{x}^{*} - \boldsymbol{x}^{(t)}\right)$$

- We want to use approximations $\boldsymbol{M}^{(t+1)}$ to $\boldsymbol{g}''\left(\boldsymbol{x}^{(t)}\right)$ which fulfil this relation when $\boldsymbol{x}^{*}$ is replaced by $\boldsymbol{x}^{(t+1)}$:

$$\boldsymbol{g}'\left(\boldsymbol{x}^{(t+1)}\right) - \boldsymbol{g}'\left(\boldsymbol{x}^{(t)}\right) = \boldsymbol{M}^{(t+1)}\left(\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^{(t)}\right)$$

- This condition is called secant condition
- There are multiple solutions to the secant condition

Quasi-Newton

- Steepest ascent and Newton method have iteration

$$\boldsymbol{x}^{(\boldsymbol{t}+\mathbf{1})} = \boldsymbol{x}^{(\boldsymbol{t})} - \left(\boldsymbol{M}^{(t)}\right)^{-1} \boldsymbol{g}'\left(\boldsymbol{x}^{(\boldsymbol{t})}\right)$$

- Secant condition:

$$\boldsymbol{g}'\left(\boldsymbol{x}^{(t+1)}\right) - \boldsymbol{g}'\left(\boldsymbol{x}^{(t)}\right) = \boldsymbol{M}^{(t+1)}\left(\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^{(t)}\right)$$

- Or, with $\boldsymbol{y}^{(t)} = \boldsymbol{g}'\left(\boldsymbol{x}^{(t+1)}\right) - \boldsymbol{g}'\left(\boldsymbol{x}^{(t)}\right)$ and $\boldsymbol{z}^{(t)} = \boldsymbol{x}^{(t+1)} - \boldsymbol{x}^{(t)}$:

$$\boldsymbol{y}^{(t)} = \boldsymbol{M}^{(t+1)} \boldsymbol{z}^{(t)}$$

- Suggestion from Broyden, Fletcher, Goldfarb, and Shanno (BFGS; 4 publications in 1970) fulfilling secant condition:

$$\boldsymbol{M}^{(t+1)} = \boldsymbol{M}^{(t)} - \frac{\boldsymbol{M}^{(t)} \boldsymbol{z}^{(t)} \left(\boldsymbol{M}^{(t)} \boldsymbol{z}^{(t)}\right)^{T}}{\boldsymbol{z}^{(t)^{T}} \boldsymbol{M}^{(t)} \boldsymbol{z}^{(t)}} + \frac{\boldsymbol{y}^{(t)} \boldsymbol{y}^{(t)^{T}}}{\boldsymbol{y}^{(t)^{T}} \boldsymbol{z}^{(t)}}$$

Quasi-Newton

- The BFGS (quasi-Newton) method has iteration

$$\boldsymbol{x}^{(\boldsymbol{t}+\mathbf{1})} = \boldsymbol{x}^{(\boldsymbol{t})} - \left(\boldsymbol{M}^{(t)}\right)^{-1} \boldsymbol{g}'\left(\boldsymbol{x}^{(\boldsymbol{t})}\right)$$

  and

$$\boldsymbol{M}^{(t+1)} = \boldsymbol{M}^{(t)} - \frac{\boldsymbol{M}^{(t)}\boldsymbol{z}^{(t)}\left(\boldsymbol{M}^{(t)}\boldsymbol{z}^{(t)}\right)^{T}}{\boldsymbol{z}^{(t)^{T}}\boldsymbol{M}^{(t)}\boldsymbol{z}^{(t)}} + \frac{\boldsymbol{y}^{(t)}\boldsymbol{y}^{(t)^{T}}}{\boldsymbol{y}^{(t)^{T}}\boldsymbol{z}^{(t)}}$$

- Ascent is not ensured but backtracking (stepsize-halving) can be used as for steepest ascent to ensure it:

$$\boldsymbol{x}^{(\boldsymbol{t}+\mathbf{1})} = \boldsymbol{x}^{(\boldsymbol{t})} - \alpha_t \left(\boldsymbol{M}^{(t)}\right)^{-1} \boldsymbol{g}'\left(\boldsymbol{x}^{(\boldsymbol{t})}\right)$$

- The **R** function **`optim`** includes the quasi-Newton BFGS
- Convergence of quasi-Newton methods are faster than linear but slower than quadratic (some assumptions necessary; see e.g. Nocedal and Wright, 2006, Theorem 3.7)

Convergence order for deterministic algorithms

- Recall: Convergence order and convergence rate

$$\frac{\{g(\boldsymbol{x}^{(t+1)}) - g(\boldsymbol{x}^*)\}}{\{g(\boldsymbol{x}^{(t)}) - g(\boldsymbol{x}^*)\}^q} \to c \quad (\text{for } t \to \infty)$$

- q is convergence order (q=1, 0<c<1 linear; q=2, 0<c<1 quadratic)
- c is convergence rate
- Under certain assumption, we have following orders:

Uni-dimensional	**Bisection** order = roughly 1*		**Secant** order = $(1+\sqrt{5})/2$	**Newton** order = 2
Multi-dimensional		**Steepest ascent** order = 1	**Quasi-Newton** order > 1**	**Newton** order = 2

*strictly, the above criterion cannot be proven for bisection
**criterion above fulfilled for q=1 and c=0; “superlinear”

Convergence speed for an example function

- The convergence of BFGS and Newton can be extremely fast in praxis compared to steepest ascent/descent
- Example from Nocedal and Wright (2006), chapter 6: Rosenbrock function $g(\boldsymbol{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$, starting point (-1.2, 1), optimum at (1,1).
  #iterations until error < 10^{-5}:
  - Steepest descent 5264
  - BFGS 34
  - Newton 21

Assignments

- Topic 1+2: March 17 until April 3
- Topic 3: April 4 until April 17
- Topic 4: April 18 until May 1
- Topic 5: May 2 until May 15
- Topic 6+7: May 17 until June 7
- Second chance for Topic 1-7: until **September 30 (no extension!)**