Optimisation algorithms in Statistics I, lecture 2

Frank Miller, Department of Statistics; Stockholm University

October 13, 2020

Stockholm University

Course schedule

- Topic 1: **Gradient based algorithms**
  Lectures: October 2; Time 10-12, 13-15 (online, Zoom)
- Topic 2: **Stochastic gradient based algorithms**
  Lecture: October 13; Time: 9-12 (online, Zoom)
- Topic 3: **Gradient free algorithms**
  Lecture: October 23; Time 9-12 (online, Zoom)
- Topic 4: **Optimisation with restrictions**
  Lecture: November 6, Time 9-12 (online, Zoom)

Course homepage: http://gauss.stat.su.se/phd/oasi/

Includes reading material, lecture notes, assignments

Today's schedule

- Accelerated steepest ascent algorithms
  - Polyak's momentum
  - Nesterov's momentum
  - Optimal choice of hyperparameters

- Stochastic steepest ascent algorithms
  - Idea and issues
  - Choice of step size
  - Mini-batches

Multivariate optimisation – Steepest ascent method

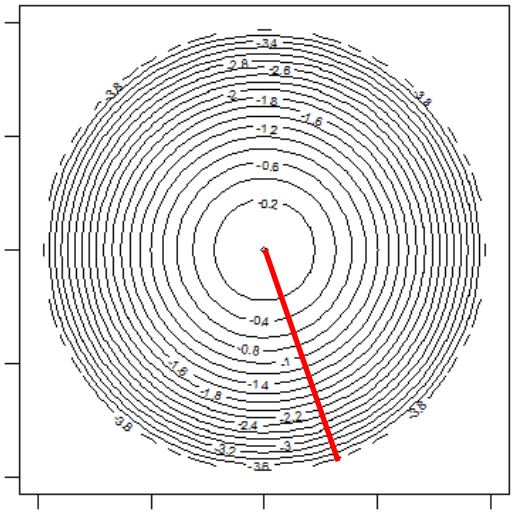
- Optimisation problem:
  - $\boldsymbol{x}$ p-dimensional vector, g: $\mathbb{R}^p \to \mathbb{R}$ function
  - We search $\boldsymbol{x}^*$ with $g(\boldsymbol{x}^*) = \max g(\boldsymbol{x})$

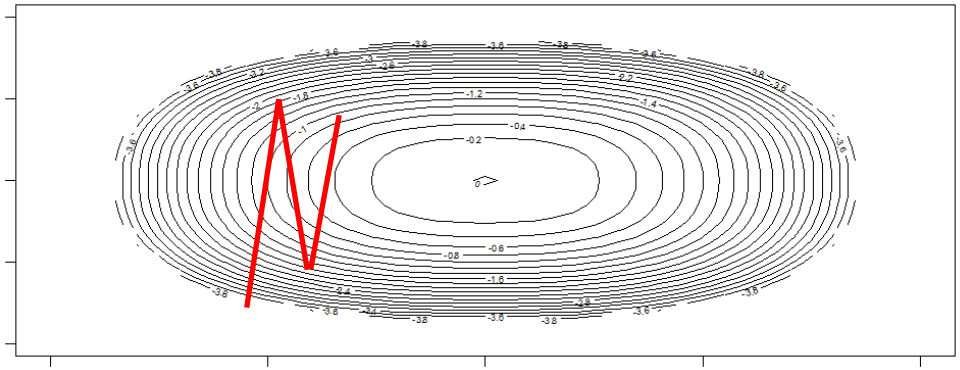
- Steepest ascent:
  - Iteration: $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \alpha^{(t)}\, \boldsymbol{g}'\left(\boldsymbol{x}^{(t)}\right)$

Steepest ascent method

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

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



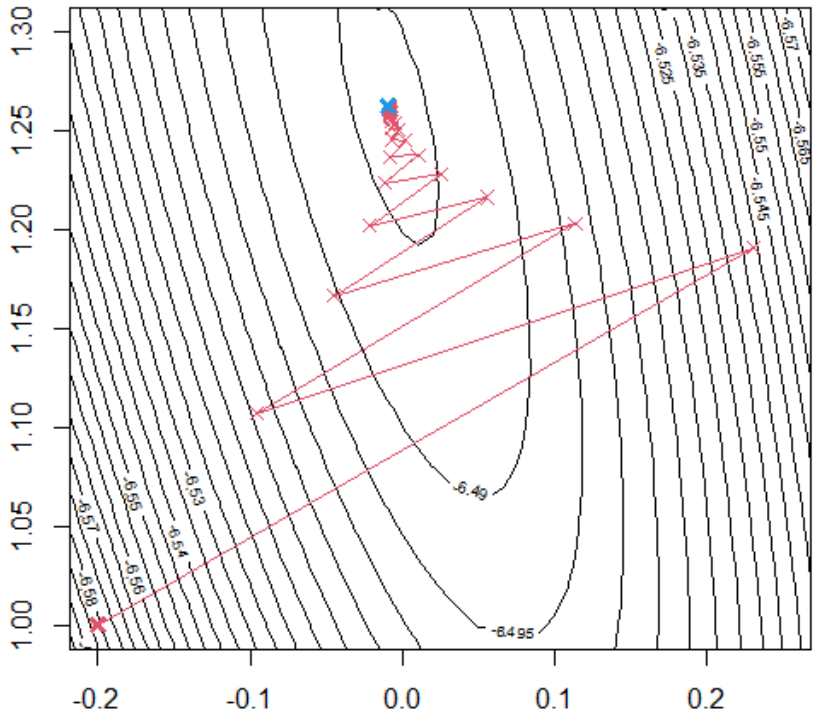


Stockholm University

Steepest ascent method

Problem 1.3:

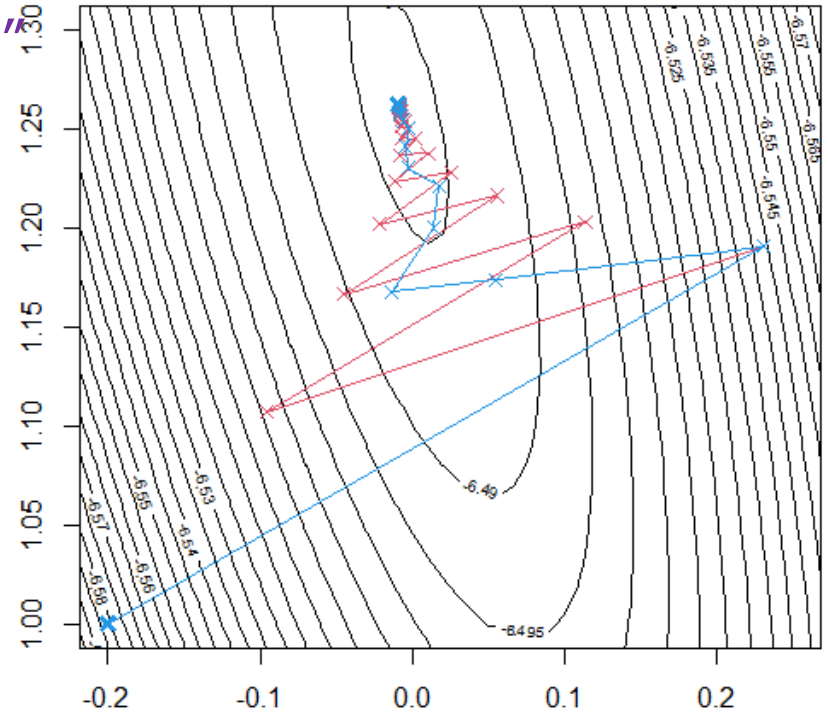
- ML computation with steepest ascent, $\alpha^{(t)} = 0.667$; logistic regression, n=10
- Zick-zack path is common and slows down convergence
- Idea to reduce/avoid this issue: use information from last iteration about ”momentum” of search path



Stockholm University

Steepest ascent method – Polyak's momentum method

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



Stockholm University

Steepest ascent method – Nesterov's momentum method

- $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \alpha^{(t)}\, \boldsymbol{g}'\left(\boldsymbol{x}^{(t)} + \beta(\boldsymbol{x}^{(t)} - \boldsymbol{x}^{(t-1)})\right) + \beta(\boldsymbol{x}^{(t)} - \boldsymbol{x}^{(t-1)})$
- 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 momentum methods

- Polyak's momentum method

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

  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 momentum method

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

  can be written also as

  $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \alpha \boldsymbol{v}^{(t+1)}$

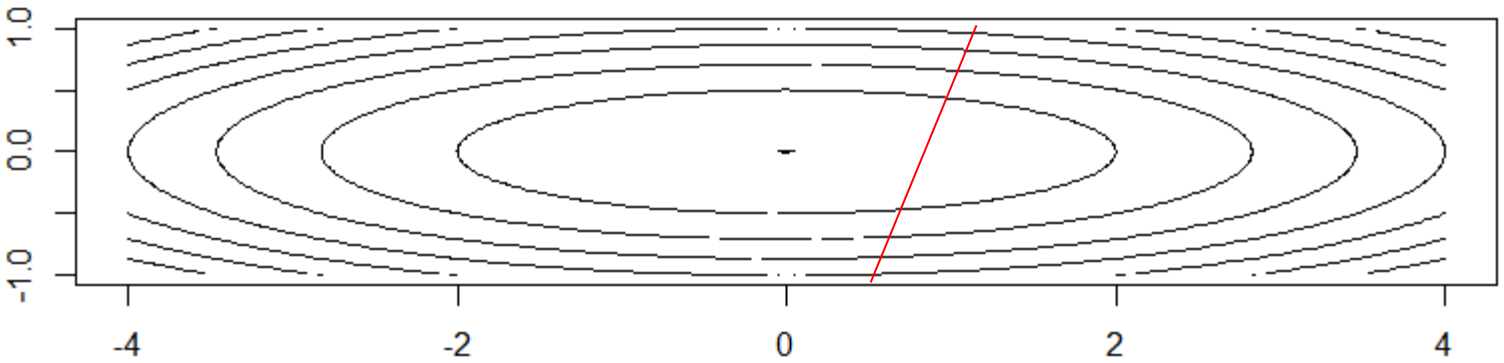
  $\boldsymbol{v}^{(t+1)} = \beta \boldsymbol{v}^{(t)} + \boldsymbol{g}'\left(\boldsymbol{x}^{(t)} + \alpha\beta \boldsymbol{v}^{(t)}\right)$

Steepest ascent method – optimal choice of step size

- $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \alpha \boldsymbol{g}'\left(\boldsymbol{x}^{(t)}\right)$
- Example:
  $g(\boldsymbol{x}) = -\frac{1}{2}\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{x}$, $\boldsymbol{A}$ symmetric *p*x*p* 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 method – optimal choice of step size

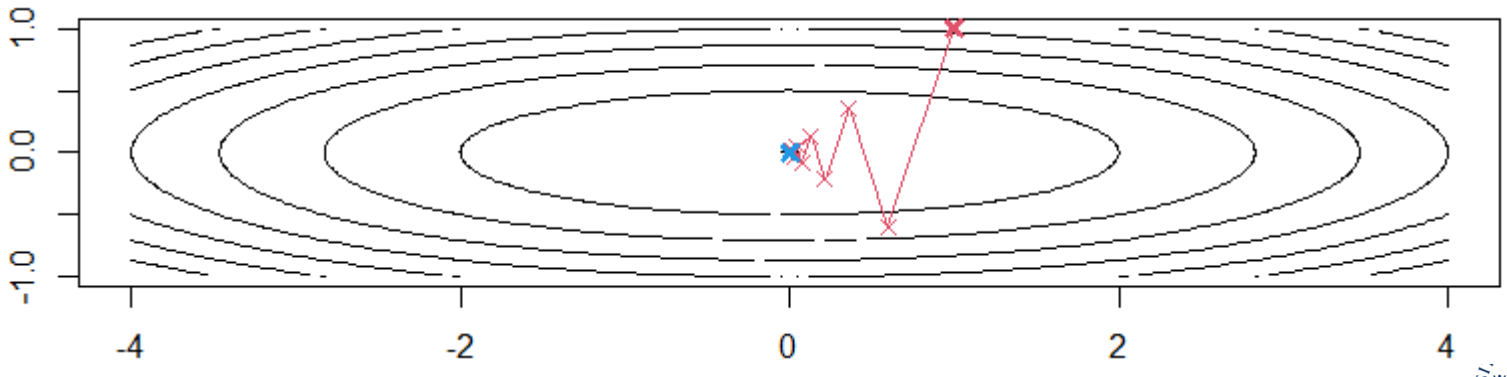
- $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \alpha \boldsymbol{g}'\left(\boldsymbol{x}^{(t)}\right)$
- Example:
  $g(\boldsymbol{x}) = -\frac{1}{2}\boldsymbol{x}^T \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \boldsymbol{x}, \ \boldsymbol{g}'(\boldsymbol{x}) = \begin{pmatrix} -\lambda_1 x_1 \\ -\lambda_2 x_2 \end{pmatrix}, \ \boldsymbol{x}^{(\boldsymbol{0})} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$
- 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 method – optimal choice of step size

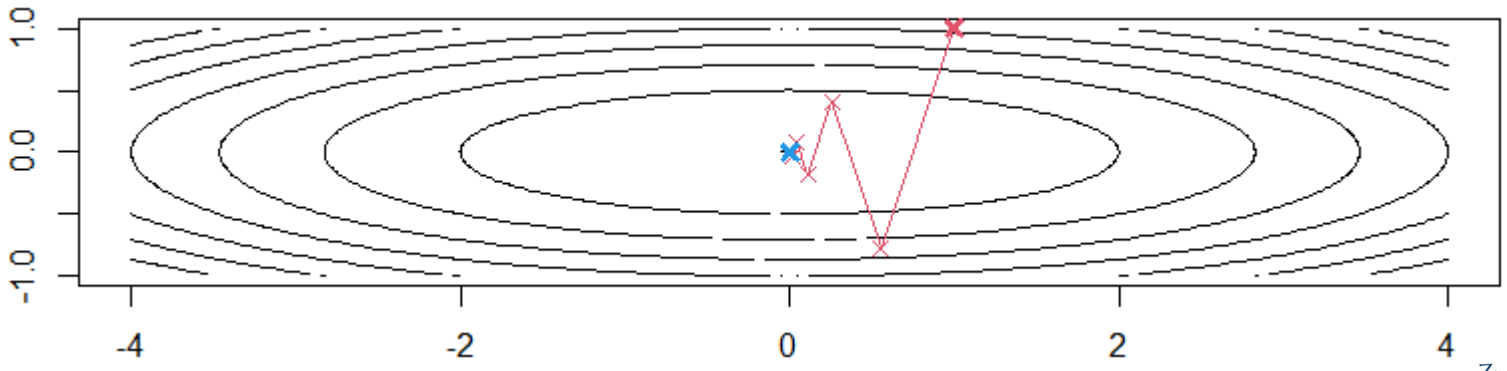
- Steepest ascent: $x_1^{(t+1)} = (1-\alpha\lambda_1)^{t+1}, x_2^{(t+1)} = (1-\alpha\lambda_2)^{t+1}$
- Fastest convergence attained if α such that $\rho = \max\{|1-\alpha\lambda_1|, |1-\alpha\lambda_2|\}$ is as small as possible
- Fulfilled for $\alpha = \frac{2}{\lambda_1+\lambda_2}$ and then $\rho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



Stockholm University

Steepest ascent method (with/without momentum) – 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-length: $\alpha = \frac{(1+\rho)^2}{\lambda_{max}} = \frac{(1-\rho)^2}{\lambda_{min}}$
  - Optimal momentum: $\beta = \rho^2$
- For example with $\lambda_1 = \frac{1}{2}, \lambda_2 = 2$: $\rho = \frac{1}{3}, \alpha = \frac{8}{9}, \beta = \frac{1}{9}$.



Stockholm University

Steepest ascent method (with/without momentum) – 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 $\beta = 1$

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

convergence rate

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

Multivariate optimisation – Steepest ascent method

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

- Optimisation problem (finite sum case):
  - $\boldsymbol{x}$ p-dimensional vector, g_i: $\mathbb{R}^p \to \mathbb{R}$ functions
  - We search $\boldsymbol{x}^*$ with $g(\boldsymbol{x}^*) = \max g(\boldsymbol{x})$ where $g = \sum_{i=1}^{n} g_i$

- If n large: Takes time to evaluate gradient $\boldsymbol{g}' = \sum_{i=1}^{n} \boldsymbol{g}'_i$

Stochastic steepest ascent method

- Iteration:
  - Choose $i \in \{1, \ldots, n\}$ randomly
  - $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \alpha^{(t)} \boldsymbol{g'}_i\left(\boldsymbol{x}^{(t)}\right)$
- $\alpha^{(t)}$ is a predefined sequence, either
  - constant step size $\alpha^{(t)} = \alpha$ or
  - decreasing step size e.g. $\alpha^{(t)} = \alpha/t$

- Convergence (to a local maximum) can be shown if step size fullfills $\sum_{t=1}^{\infty} \alpha^{(t)} = \infty$ and $\sum_{t=1}^{\infty} (\alpha^{(t)})^2 < \infty$ (example: $\alpha^{(t)} = \alpha/t$)

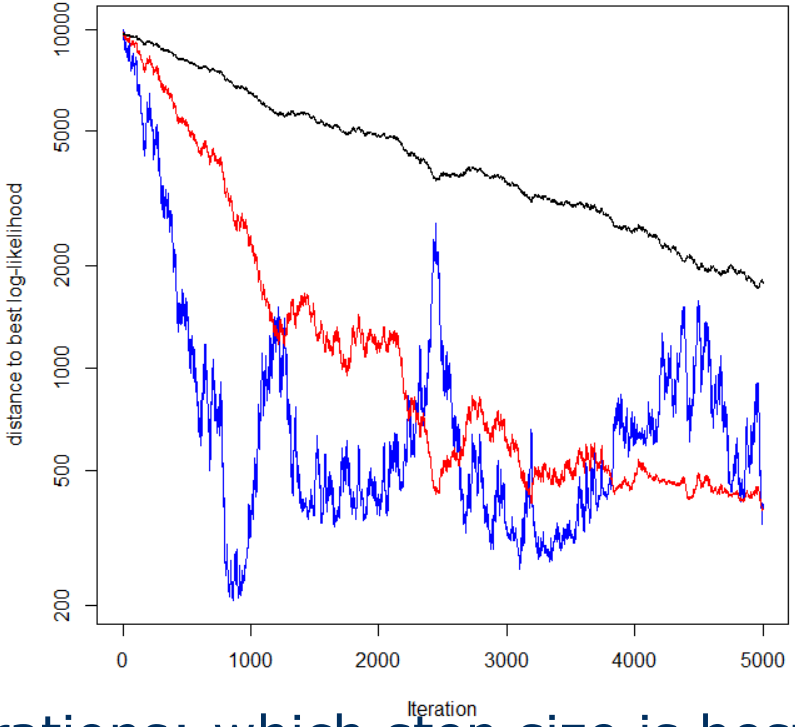
Stochastic steepest ascent method

- Constant step size $\alpha^{(t)} = \alpha$ can still make sense if
  - Another algorithm is run afterwards, or
  - If good but not necessarily best solution desired

- Choice of step size is critical

Stochastic steepest ascent method – choice of step size

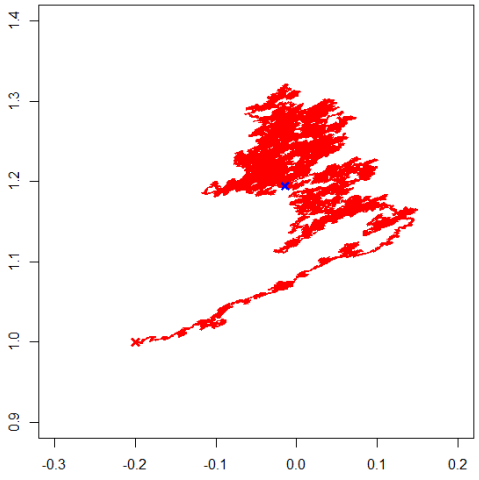
- Constant step size $\alpha^{(t)} = \alpha$
- Choice of step size is critical
- Influence of step size
- Step size
  - $\alpha = 0.0006$ (black)
  - $\alpha = 0.002$ (red)
  - $\alpha = 0.006$ (blue)

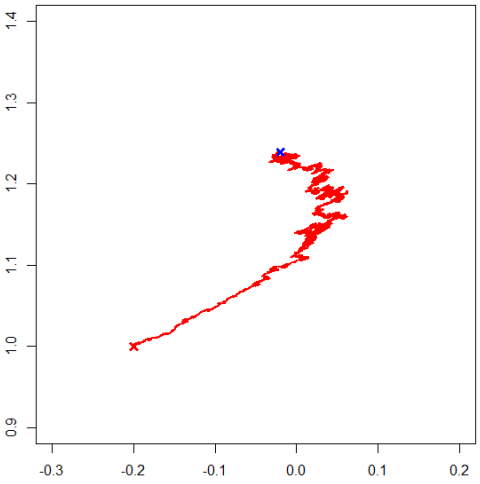


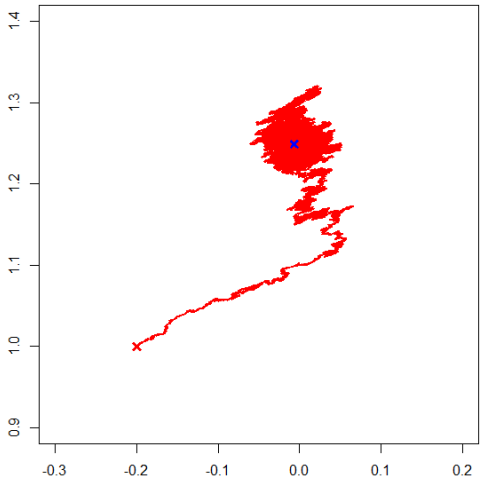
- If you have time for 5000 iterations: which step size is best?
- If you have only time for 500 iterations?
- If you have time for 50000 iterations?

Stochastic steepest ascent method – choice of step size

- Problem 1.3 with n= 1 000 000 (data duplicated 100 000 x); start at $(\beta_0, \beta_1) = (-0.2, 1)$; best $(\hat{\beta}_0, \hat{\beta}_1) = (-0.009, 1.263)$
- 100 000 iterations,
  $\alpha = 0.002$,
  $\alpha = 0.0006$
  1 000 000 iterations,
  $\alpha = 0.0006$







- (-0.014, 1.195) (-0.020, 1.239) (-0.007, 1.248)

Stochastic steepest ascent method

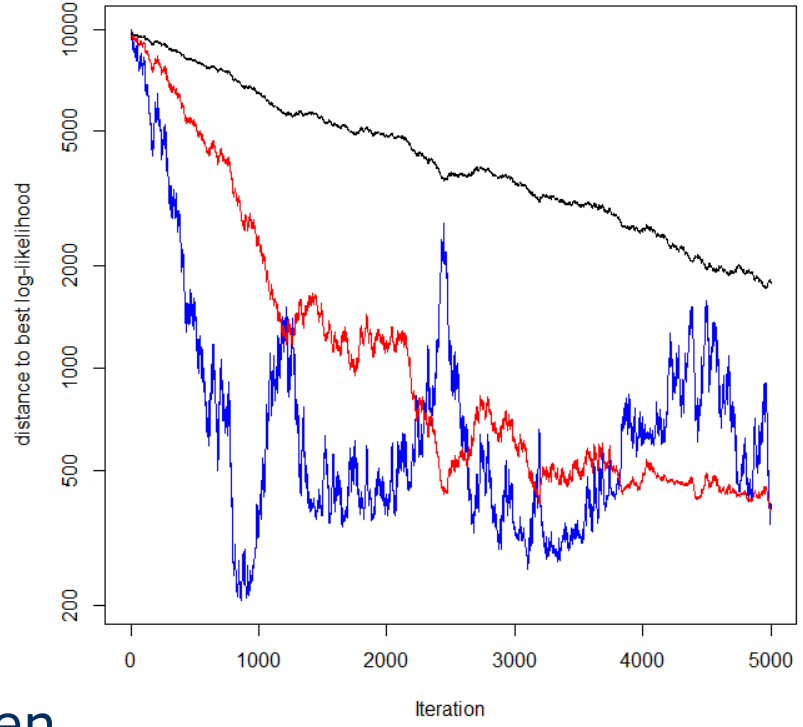
- Influence of step size can be investigated at
  - http://fa.bianp.net/teaching/2018/COMP-652/stochastic_gradient.html (Fabia Pedregosa, Nov 2018)

Stochastic steepest ascent method – choice of step size

- Goodfellow et al., 2016, Chapter 8.3.1 (notation adjusted):
- "In practice, it is common to decay the learning rate [=step size $\alpha^{(t)}$] linearly until iteration τ: $\alpha^{(t)} = (1-\gamma)\alpha_0 + \gamma\alpha_\tau$ with $\gamma = \frac{t}{\tau}$. After iteration τ, it is common to leave α constant."
- Choice of step size "is more of an art than a science, and most guidance on this subject should be regarded with some skepticism."
- Choose τ "to make a few hundred passes through the training set."
- $\alpha_\tau \approx \alpha_0/100$
- Choose α_0 avoiding violent oscillations and too low learning rate

Comparison Steepest ascent and Stochastic steepest ascent

- Example: Problem 1.3 with n = 1 000 000
- Stochastic steepest ascent: 50000 iterations took 7s
- Steepest ascent with alpha-halving: 112 iterations took 52s
- Stochastic steepest ascent could run 3320 iterations when Steepest ascent could run 1 iteration



Stockholm University

Stochastic steepest ascent method – mini-batches

- Instead of sampling a single *i*, a batch of size *m* can be sampled in each iteration
- Iteration:
  - Choose $\{i_1, \ldots, i_m\} \subseteq \{1, \ldots, n\}$ randomly
  - $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \alpha^{(t)} \sum_{j=1}^{m} \boldsymbol{g'}_{i_j}\left(\boldsymbol{x}^{(t)}\right)$
- Decreases risk of large random oscillations
- Especially interesting when algorithm performed on a parallel computer