Advanced computational statistics, lecture 2

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

Department of Statistics; Stockholm University

March 17, 2023

LINKÖPING UNIVERSITY

Course schedule

- 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**

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

Includes schedule, reading material, lecture notes, assignments

Today's schedule

- Stochastic steepest descent (SSD; Stochastic gradient descent; SGD)
  - Idea and issues
  - Choice of step size
  - Mini-batches
  - Convergence analysis
- Exercise session

Note: Changed to descent and minimisation problem here to correspond to most literature, but this is no essential change.

Steepest descent

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

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

Steepest descent

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

- Optimisation problem (finite sum case):
  - $\boldsymbol{x}$ p-dimensional vector, g_i: $\mathbb{R}^p \to \mathbb{R}$ functions
  - We search $\boldsymbol{x}^*$ with $g(\boldsymbol{x}^*) = \min g(\boldsymbol{x})$ 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 descent

- Iteration:
  - Choose $i \in \{1, \dots, n\}$ randomly
  - $\boldsymbol{x}^{(\boldsymbol{t}+\mathbf{1})} = \boldsymbol{x}^{(\boldsymbol{t})} - \alpha^{(t)} \boldsymbol{g}'_i\left(\boldsymbol{x}^{(\boldsymbol{t})}\right)$
- $\alpha^{(t)}$ 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 minimum) 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 descent

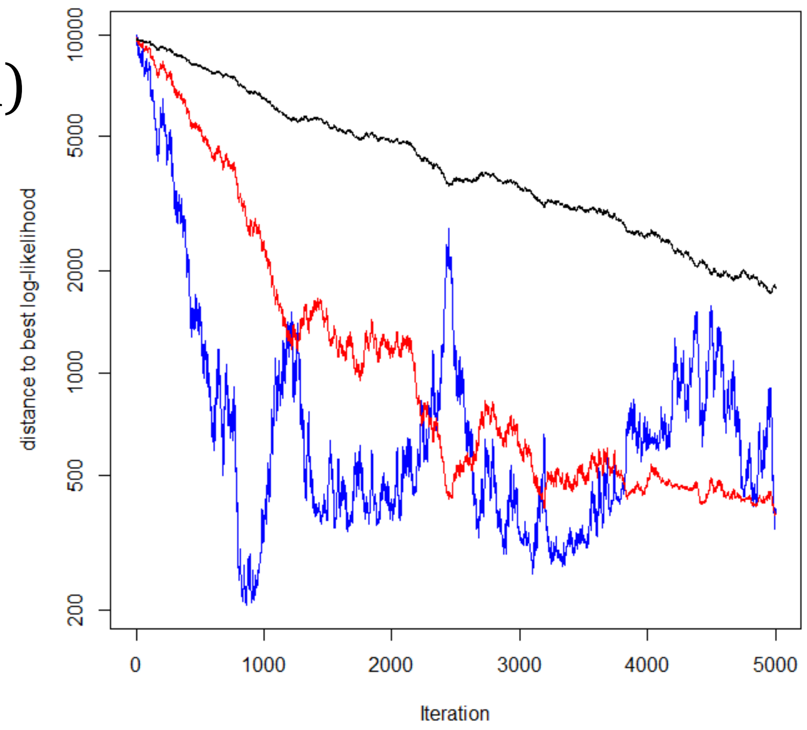
- Constant step size $\alpha^{(t)} = \alpha$ can still make sense if
  - Another algorithm is run afterwards, or
  - If good but not necessarily best solution desired

- Choice of step size is critical

- Example: Two-parameter MLE computation (large n)
  Computation of MLE for a model with two parameters and n = 1 000 000.
  Starting value is not too good (has some distance to correct MLE). We monitor:
  - distance of current log likelihood to maximal log likelihood,
  - search path in 2d parameter space.

Stochastic steepest descent: choice of step size

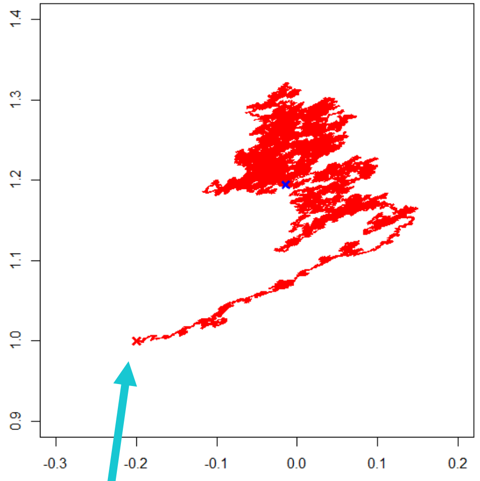
- Example: Two-parameter MLE computation (large n)
- Constant step size $\alpha^{(t)} = \alpha$
- Choice of step size is critical
- Step size here:
  - $\alpha = 0.0006$ (black)
  - $\alpha = 0.002$ (red)
  - $\alpha = 0.006$ (blue)



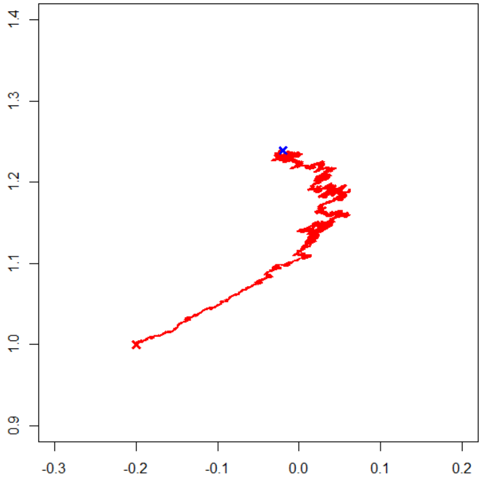
- If you have time for 5000 iterations: which step size is best?
- If you have only time for 500 iterations?
- If you have time for 50000 iterations?

Stochastic steepest descent: choice of step size

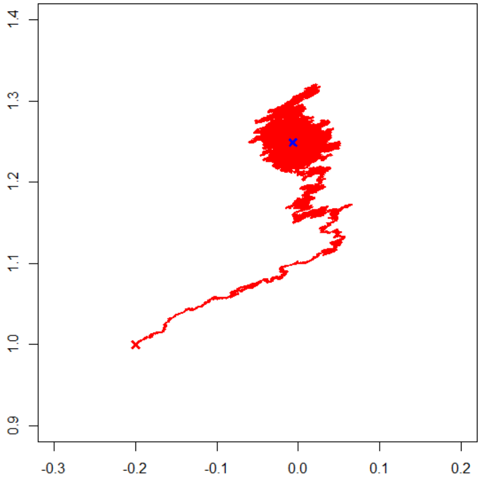
- Example: Two-parameter MLE computation (large n)
  Search path in the 2d parameter space
- 100 000 iterations,
  $\alpha = 0.002,$



$\alpha = 0.0006$



1 000 000 iterations,
$\alpha = 0.0006$



Starting value

Stochastic steepest descent

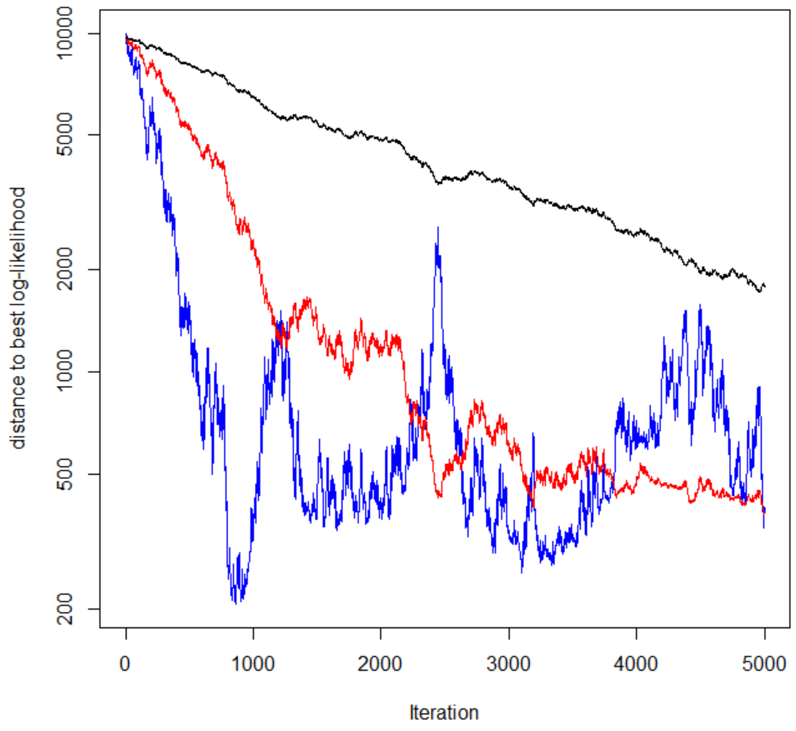
- Influence of step size can be investigated at http://fa.bianp.net/teaching/2018/COMP-652/stochastic_gradient.html (Fabia Pedregosa, Nov 2018)

Stochastic steepest descent: 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

(Stochastic) steepest descent: running time

- Example: Two-parameter MLE computation (large n)
- Stochastic steepest ascent: 50000 iterations took 7 s
- Steepest ascent with alpha-halving: 112 iterations took 52 s
- Stochastic steepest ascent could run 3320 iterations when steepest ascent could run 1 iteration



Stochastic steepest descent: 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
  - $\boldsymbol{x}^{(\boldsymbol{t}+\mathbf{1})} = \boldsymbol{x}^{(\boldsymbol{t})} - \alpha^{(t)} \sum_{j=1}^{m} \boldsymbol{g}'_{\boldsymbol{i_j}}\left(\boldsymbol{x}^{(\boldsymbol{t})}\right)$
- Decreases risk of large random oscillations
- Especially interesting when algorithm performed on a parallel computer

Accelerated stochastic steepest descent (adding momentum)

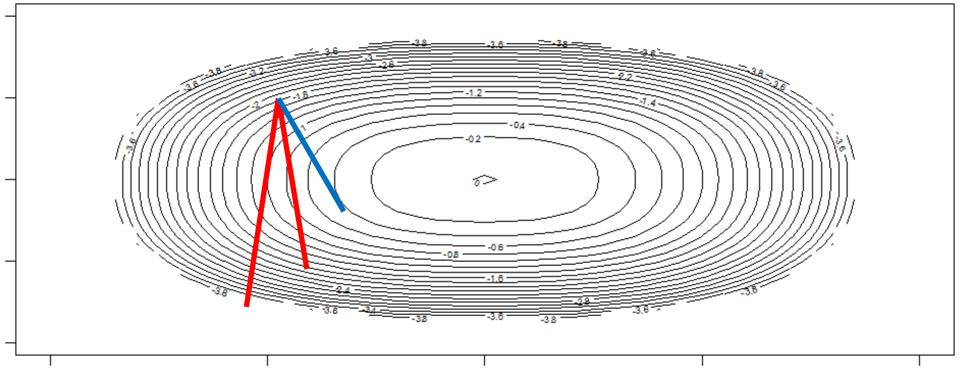
- Stochastic steepest descent $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} - \alpha \boldsymbol{g'}_i\left(\boldsymbol{x}^{(t)}\right)$ can be combined with momentum method (see Goodfellow, Bengio, Courville, 2016, Chapter 8.3.2)
- Iteration:
  - Choose $i \in \{1, \ldots, n\}$ randomly
  - $\boldsymbol{v}^{(t+1)} = \beta \boldsymbol{v}^{(t)} - \boldsymbol{g'}_i\left(\boldsymbol{x}^{(t)}\right)$
  - $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \alpha \boldsymbol{v}^{(t+1)}$
- Advantages:
  - Momentum advantages (handling ill-conditioning, accelerating)
  - Information from previous gradients contribute (variance of stochastic gradient reduced)

Accelerated stochastic steepest ascent (adding momentum)

- Both hyperparameters α, β may depend on iteration number
- Iteration:
  - Choose $i \in \{1, \dots, n\}$ randomly
  - $\boldsymbol{v}^{(t+1)} = \beta^{(t)}\boldsymbol{v}^{(t)} - \boldsymbol{g}'_i\left(\boldsymbol{x}^{(t)}\right)$
  - $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \alpha^{(t)}\boldsymbol{v}^{(t+1)}$
- Changing hyperparameters:
  - $\beta^{(t)}$ usually increased with t, common values 0.5 to 0.99
  - $\alpha^{(t)}$ is decreased with t
  - Decreasing $\alpha^{(t)}$ more important than changing $\beta^{(t)}$

Stochastic steepest descent: adaptive step sizes

- Stochastic steepest descent:
  - Choose $i \in \{1, \dots, n\}$ randomly
  - $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} - \alpha^{(t)} \boldsymbol{g}'_i(\boldsymbol{x}^{(t)})$
- $\alpha^{(t)}$ is now adapted automatically based on previous iterations and separately for each dimension
- If previous gradients in a dimension were large, we want to reduce step size more



Stochastic steepest descent: adaptive step sizes

- **AdaGrad:**

  $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} - \text{diag}(\boldsymbol{\alpha}^{(t)})\boldsymbol{g}'_i\left(\boldsymbol{x}^{(t)}\right)$ with vector $\boldsymbol{\alpha}^{(t)}$
- $\alpha_j^{(t)} = \alpha / \sqrt{\epsilon + \sum_{k=1}^{t} (g'_{j,k})^2}$
  - $g'_{j,k}$ is j^{th} partial derivative of gradient in iteration k
  - ϵ is small constant (e.g. 1e-8)
- A default value $\alpha = 0.01$ is a popular choice

Stochastic steepest descent: adaptive step sizes

- AdaGrad:
  - Choose $i \in \{1, \dots, n\}$ randomly
  - $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} - \mathrm{diag}(\boldsymbol{\alpha}^{(t)})\boldsymbol{g}'_i\left(\boldsymbol{x}^{(t)}\right)$
  - $\alpha_j^{(t)} = \alpha / \sqrt{\epsilon + \sum_{k=1}^{t}(g'_{j,k})^2}$
- Disadvantage: $\alpha_j^{(t)}$ can only decrease
- **AdaDelta:**
  - $\alpha_j^{(t)} = \alpha / \sqrt{\epsilon + h_j^{(t)}}$
  - $h_j^{(t)} = \gamma h_j^{(t-1)} + (1-\gamma)(g'_{j,t})^2$
    (exponential smoothing of earlier partial derivatives; popular choice of γ is around 0.9)

Stochastic steepest descent: adaptive step sizes

- AdaDelta:
  - Random $i \in \{1, \ldots, n\}$; $\boldsymbol{x}^{(\boldsymbol{t}+\mathbf{1})} = \boldsymbol{x}^{(\boldsymbol{t})} - \mathrm{diag}(\boldsymbol{\alpha}^{(t)})\boldsymbol{g'}_i\left(\boldsymbol{x}^{(\boldsymbol{t})}\right)$
  - $\alpha_j^{(t)} = \alpha / \sqrt{\epsilon + h_j^{(t)}}$; $h_j^{(t)} = \gamma h_j^{(t-1)} + (1-\gamma)(g'_{j,t})^2$

- **Adam** ("Adaptive moment estimation")
  - Random $i \in \{1, \ldots, n\}$; $\boldsymbol{x}^{(\boldsymbol{t}+\mathbf{1})} = \boldsymbol{x}^{(\boldsymbol{t})} - \mathrm{diag}(\boldsymbol{\alpha}^{(t)})\widehat{\boldsymbol{m}}_t$;
  - $\boldsymbol{m}_t = \beta_1 \boldsymbol{m}_{t-1} + (1-\beta_1)\boldsymbol{g'}_{i(t)}\left(\boldsymbol{x}^{(\boldsymbol{t})}\right)$
  - $\boldsymbol{v}_t = \beta_2 \boldsymbol{v}_{t-1} + (1-\beta_2)\left((g'_{j,t})^2\right)_{j=1,\ldots,p}$
  - $\alpha_j^{(t)} = \alpha / \sqrt{\epsilon + \hat{v}_{j,t}}$

$\widehat{\boldsymbol{m}}_t = \boldsymbol{m}_t / (1-\beta_1^t)$

$\widehat{\boldsymbol{v}}_t = \boldsymbol{v}_t / (1-\beta_2^t)$

- Default values $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$

Stochastic steepest descent: adaptive step sizes

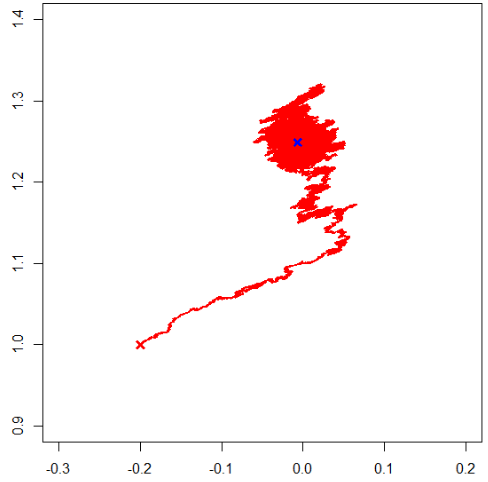
- Momentum method can be added to AdaGrad and AdaDelta
- AdaGrad works well for convex functions
- AdaDelta handles non-convex functions better
- In Adam, momentum method already included

Steepest descent – comparisons of methods

- Animated comparisons:
  - https://imgur.com/a/Hqolp

Stochastic steepest descent

- Going back to the stochastic steepest descent (with non-adaptive step sizes)
- Iteration:
  - Choose $i \in \{1, \dots, n\}$ randomly
  - $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} - \alpha_t \boldsymbol{g'}_i\left(\boldsymbol{x}^{(t)}\right)$
- α_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 fulfils $\sum_{t=1}^{\infty} \alpha_t = \infty$ and $\sum_{t=1}^{\infty} \alpha_t^2 < \infty$ (example: $\alpha_t = \alpha / t$)

- Now: Looking closer into the convergence properties, ”Convergence analysis”

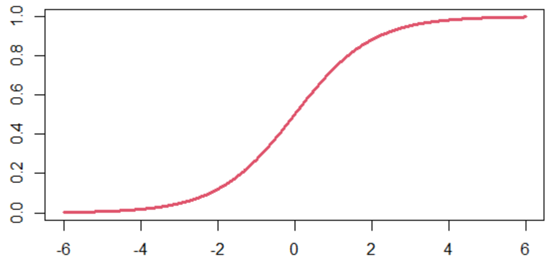
Stochastic steepest descent (SSD)

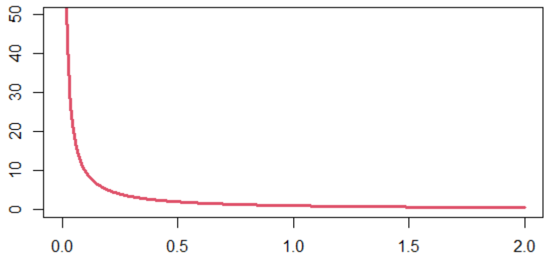
- Function to be **minimised**: $g = \frac{1}{n}\sum_{i=1}^{n} g_i$
- Predefined sequence of step sizes: α_t, $t = 1,2,\ldots$
- Starting value: $\boldsymbol{x}^{(\mathbf{0})}$
- Sequence of random numbers: $R^{(t)} \in \{1, \ldots, n\}$, $t = 1,2,\ldots$
- Iteration: $\boldsymbol{x}^{(\boldsymbol{t}+\mathbf{1})} = \boldsymbol{x}^{(\boldsymbol{t})} - \alpha_t \boldsymbol{g}'_{R(t)}\left(\boldsymbol{x}^{(\boldsymbol{t})}\right)$

- We assume in the lecture:
  $R^{(t)}$ uniformly distributed on {1, …, n}, all $R^{(t)}$ independent
- We note that $\boldsymbol{E}\boldsymbol{g}'_{R(t)}\left(\boldsymbol{x}^{(\boldsymbol{t})}\right) = \boldsymbol{g}'\left(\boldsymbol{x}^{(\boldsymbol{t})}\right)$

Lipschitz continuous functions

- A function f is called *Lipschitz continuous* with Lipschitz constant L>0, if for all $\mathbf{x}_1$, $\mathbf{x}_2$,
$$\|f(\boldsymbol{x}_1) - f(\boldsymbol{x}_2)\|_2 \leq L \cdot \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2.$$
- If $f: (a, b) \to \mathbb{R}$ is differentiable, the following is true:
f Lipschitz continuous with constant L if and only if |f'(x)|≤L for all x
- Example: 1/(1+exp(-x)) is Lipschitz continuous with L=0.25
- Example: 1/x is not Lipschitz continuous on (0,∞)





- If f has a derivative (gradient) f' which is Lipschitz continuous with L>0, then f itself is called *L-smooth*. Further,
$$f(\boldsymbol{x}_1) - f(\boldsymbol{x}_2) \leq \boldsymbol{f}'(\boldsymbol{x}_2)^T(\boldsymbol{x}_1 - \boldsymbol{x}_2) + \frac{L}{2} \cdot \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2^2.$$

SSD's expected decrease per iteration

- Minimisation of $g = \frac{1}{n}\sum_{i=1}^{n} g_i$ with SSD

$$\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} - \alpha_t \boldsymbol{g}'_{R(t)}\left(\boldsymbol{x}^{(t)}\right) \qquad \text{(SSD)}$$

- <u>Lemma 1 (Bottou et al):</u> Let g be L-smooth with L>0. Given $\mathbf{x}^{(t)}$, the expected decrease in an SSD iteration is bounded:

$$E\left[g\left(\boldsymbol{x}^{(t+1)}\right)\right] - g\left(\boldsymbol{x}^{(t)}\right) \leq -\alpha_t \left\|\boldsymbol{g}'\left(\boldsymbol{x}^{(t)}\right)\right\|_2^2 + \alpha_t^2 \frac{L}{2} E\left\|\boldsymbol{g}'_{R(t)}\left(\boldsymbol{x}^{(t)}\right)\right\|_2^2$$

LINKÖPING UNIVERSITY

SSD's expected decrease per iteration

- Detailed proof of Lemma 1 (Bottou et al): We have:
- $$\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} - \alpha_t \boldsymbol{g}'_{R(t)}\left(\boldsymbol{x}^{(t)}\right) \qquad \text{(SSD)}$$
- $$g(\boldsymbol{x}_1) - g(\boldsymbol{x}_2) \leq \boldsymbol{g}'(\boldsymbol{x}_2)^T(\boldsymbol{x}_1 - \boldsymbol{x}_2) + \frac{L}{2}\cdot \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2^2 \text{ for all } \boldsymbol{x}_1,\ \boldsymbol{x}_2 \qquad \text{(Lsmooth)}$$
- R(t) uniformly distributed on {1, …, n} (R)
- Using (Lsmooth) for $\boldsymbol{x}_1 = \boldsymbol{x}^{(t+1)}$ and $\boldsymbol{x}_2 = \boldsymbol{x}^{(t)}$ (conditional on $R^{(t)}$ and $\boldsymbol{x}^{(t)}$),
  $$g\left(\boldsymbol{x}^{(t+1)}\right) - g\left(\boldsymbol{x}^{(t)}\right) \leq \boldsymbol{g}'\left(\boldsymbol{x}^{(t)}\right)^T\left(\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^{(t)}\right) + \frac{L}{2}\cdot\left\|\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^{(t)}\right\|_2^2$$
- Using (SSD),
  $$= \boldsymbol{g}'\left(\boldsymbol{x}^{(t)}\right)^T\left(-\alpha_t \boldsymbol{g}'_{R(t)}\left(\boldsymbol{x}^{(t)}\right)\right) + \frac{L}{2}\cdot\left\|\alpha_t \boldsymbol{g}'_{R(t)}\left(\boldsymbol{x}^{(t)}\right)\right\|_2^2$$
- $$= -\alpha_t \boldsymbol{g}'\left(\boldsymbol{x}^{(t)}\right)^T \boldsymbol{g}'_{R(t)}\left(\boldsymbol{x}^{(t)}\right) + \alpha_t^2\frac{L}{2}\cdot\left\|\boldsymbol{g}'_{R(t)}\left(\boldsymbol{x}^{(t)}\right)\right\|_2^2$$
- Take expectation over $R^{(t)}$ given $\mathbf{x}^{(t)}$
  $$E\left[g\left(\boldsymbol{x}^{(t+1)}\right)\right] - g\left(\boldsymbol{x}^{(t)}\right) \leq -\alpha_t \boldsymbol{g}'\left(\boldsymbol{x}^{(t)}\right)^T E\left[\boldsymbol{g}'_{R(t)}\left(\boldsymbol{x}^{(t)}\right)\right] + \alpha_t^2\frac{L}{2}E\left[\left\|\boldsymbol{g}'_{R(t)}\left(\boldsymbol{x}^{(t)}\right)\right\|_2^2\right]$$
- $$= -\alpha_t\left\|\boldsymbol{g}'\left(\boldsymbol{x}^{(t)}\right)\right\|_2^2 + \alpha_t^2\frac{L}{2}E\left\|\boldsymbol{g}'_{R(t)}\left(\boldsymbol{x}^{(t)}\right)\right\|_2^2 \text{ since}$$
- $E\left[\boldsymbol{g}'_{R(t)}\left(\boldsymbol{x}^{(t)}\right)\right] = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{g}'_i\left(\boldsymbol{x}^{(t)}\right) = \boldsymbol{g}'\left(\boldsymbol{x}^{(t)}\right)$ due to (R). □

SSD's expected decrease per iteration

- Minimisation of $g = \frac{1}{n}\sum_{i=1}^{n} g_i$ with SSD
$$\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} - \alpha_t \boldsymbol{g'}_{R(t)}\left(\boldsymbol{x}^{(t)}\right) \qquad \text{(SSD)}$$
- <u>Lemma 1 (Bottou et al):</u> Let g be L-smooth with L>0. Given $\mathbf{x}^{(t)}$, the expected decrease in an SGD iteration is bounded:
$$E\left[g\left(\boldsymbol{x}^{(t+1)}\right)\right] - g\left(\boldsymbol{x}^{(t)}\right) \le -\alpha_t \left\|\boldsymbol{g}'\left(\boldsymbol{x}^{(t)}\right)\right\|_2^2 + \alpha_t^2 \frac{L}{2} E\left\|\boldsymbol{g}'_{R(t)}\left(\boldsymbol{x}^{(t)}\right)\right\|_2^2$$
- Proof idea: We apply the **consequence of L-smoothness** for $\boldsymbol{x}_1 = \boldsymbol{x}^{(t+1)}$ and $\boldsymbol{x}_2 = \boldsymbol{x}^{(t)}$ (conditional on $R^{(t)}$ and $\boldsymbol{x}^{(t)}$),
$$g\left(\boldsymbol{x}^{(t+1)}\right) - g\left(\boldsymbol{x}^{(t)}\right) \le \boldsymbol{g}'\left(\boldsymbol{x}^{(t)}\right)^T \left(\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^{(t)}\right) + \frac{L}{2} \cdot \left\|\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^{(t)}\right\|_2^2$$
We **use Equation (SSD)** above, **take expectation over $\mathbf{R}^{(t)}$** (given $\mathbf{x}^{(t)}$ or the history $R^{(t-1)}$, $R^{(t-2)}$, …) and **replace $\boldsymbol{E}\boldsymbol{g'}_{R(t)}\left(\boldsymbol{x}^{(t)}\right)$ by $\boldsymbol{g}'\left(\boldsymbol{x}^{(t)}\right)$**. This shows the claim. □

SSD's expected decrease per iteration

- Minimisation of $g = \frac{1}{n}\sum_{i=1}^{n} g_i$ with SSD

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

- Lemma 2 (Bottou et al): Let g be L-smooth with L>0 and we have following second moment condition:

$$E\left\|\boldsymbol{g}'_{R(t)}\left(\boldsymbol{x}^{(t)}\right)\right\|_2^2 \le s + w\left\|\boldsymbol{g}'\left(\boldsymbol{x}^{(t)}\right)\right\|_2^2 \text{ for all } t.$$

Given $\mathbf{x}^{(t)}$, the expected decrease in an SSD iteration is bounded:

$$E\left[g\left(\boldsymbol{x}^{(t+1)}\right)\right] - g\left(\boldsymbol{x}^{(t)}\right) \le -\alpha_t(1-\alpha_t Lw/2)\left\|\boldsymbol{g}'\left(\boldsymbol{x}^{(t)}\right)\right\|_2^2 + \alpha_t^2\frac{Ls}{2}$$

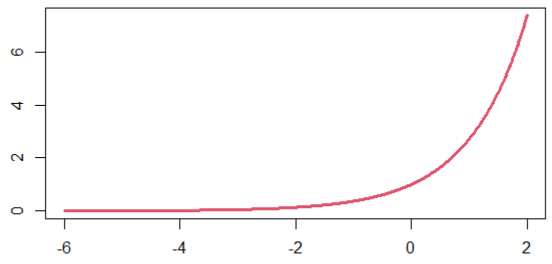
- Proof: Follows directly from Lemma 1:

$$\begin{aligned} E\left[g\left(\boldsymbol{x}^{(t+1)}\right)\right] - g\left(\boldsymbol{x}^{(t)}\right) &\le -\alpha_t\left\|\boldsymbol{g}'\left(\boldsymbol{x}^{(t)}\right)\right\|_2^2 + \alpha_t^2\frac{L}{2}E\left\|\boldsymbol{g}'_{R(t)}\left(\boldsymbol{x}^{(t)}\right)\right\|_2^2 \\ &\le -\alpha_t\left\|\boldsymbol{g}'\left(\boldsymbol{x}^{(t)}\right)\right\|_2^2 + \alpha_t^2\frac{L}{2}\left(s + w\left\|\boldsymbol{g}'\left(\boldsymbol{x}^{(t)}\right)\right\|_2^2\right) \qquad \square \end{aligned}$$

Strongly convex functions

- A differentiable function f is called *m-strongly convex* with m>0, if for all $\mathbf{x}_1$, $\mathbf{x}_2$,
$$(\boldsymbol{f}'(\boldsymbol{x}_1) - \boldsymbol{f}'(\boldsymbol{x}_2))^T(\boldsymbol{x}_1 - \boldsymbol{x}_2) \geq m \cdot \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2^2.$$
- For one-dimensional functions:
$(f'(x_1) - f'(x_2))/(x_1 - x_2) \geq m$ for all x_1, x_2.
- The function $f(x)=x^2$ is m-strongly convex with m=2

- The function f(x)=exp(x) is convex but not m-strongly convex since for $x \to -\infty$, smaller and smaller m would be necessary; no m>0 can be found to fulfil condition above



Strongly convex functions

- A differentiable function f is called *m-strongly convex* with m>0, if for all $\mathbf{x}_1$, $\mathbf{x}_2$,
$$(\boldsymbol{f}'(\boldsymbol{x}_1) - \boldsymbol{f}'(\boldsymbol{x}_2))^T(\boldsymbol{x}_1 - \boldsymbol{x}_2) \geq m \cdot \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2^2.$$

- An equivalent condition is
$$(f(\boldsymbol{x}_1) - f(\boldsymbol{x}_2)) \geq \boldsymbol{f}'(\boldsymbol{x}_2)^T(\boldsymbol{x}_1 - \boldsymbol{x}_2) + \frac{m}{2} \cdot \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2^2.$$

- An m-strongly convex function f has a unique minimum $\mathbf{x}^*$ and the following holds true:
$$2m\big(f(\boldsymbol{x}) - f(\boldsymbol{x}^*)\big) \leq \|\boldsymbol{f}'(\boldsymbol{x})\|_2^2.$$

Assumptions

- Assumptions (A):
  - g is differentiable and L-smooth with L>0,
  - g is m-strongly convex with m>0
  - For all $\mathbf{x}$: $\mathrm{E}\left\|\boldsymbol{g'}_{R(t)}(\boldsymbol{x})\right\|_2^2 \leq s + w\|\boldsymbol{g}'(\boldsymbol{x})\|_2^2$
- Assumptions (B):
  - g_i are differentiable and L-smooth with $L_i > 0$,
  - g is m-strongly convex with m>0
  - $\mathrm{E}\left\|\boldsymbol{g'}_{R(t)}(\boldsymbol{x}^*)\right\|_2^2 = s$

Convergence analysis for fixed step size

- Theorem 1 (Bottou et al): Consider the finite sum case of the optimization problem, assume Assumptions (A) and that the step size is constant, $\alpha_t = \alpha \leq 1/\{L \max(w, 1)\}$. Then, we have the following convergence result:

$$\mathrm{E}\left[g\left(\boldsymbol{x}^{(t)}\right)\right] - g(\boldsymbol{x}^*) \leq \frac{\alpha L s}{2m} + (1 - \alpha m)^t \left\{g\left(\boldsymbol{x}^{(0)}\right) - g(\boldsymbol{x}^*) - \frac{\alpha L s}{2m}\right\}$$

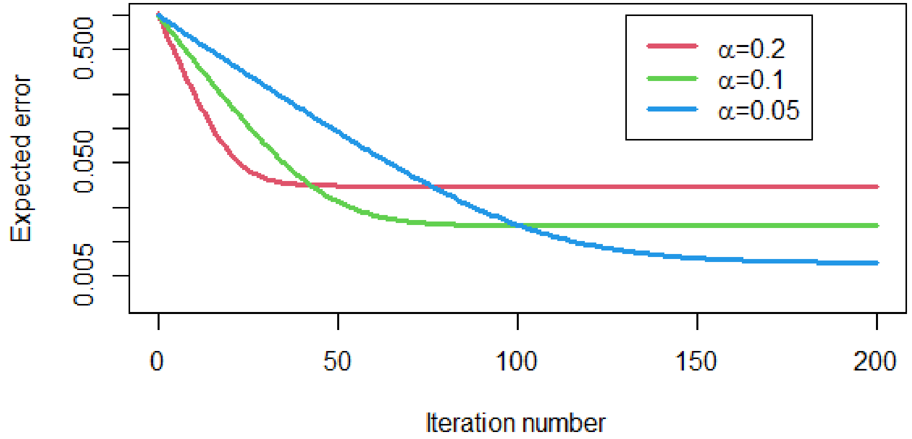
- Proof: Based on Lemma 2, see Bottou et al (2018), https://arxiv.org/pdf/1606.04838.pdf

Convergence analysis for fixed step size

- Theorem 2 (Needell et al): Consider the finite sum case of the optimization problem, assume Assumptions (B) and that the step size is constant, $\alpha_t = \alpha \leq \frac{1}{\max(L_i)}$. Then, we have the following convergence result:

$$\mathrm{E}\left\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^*\right\|_2^2 \leq \frac{\alpha s}{m\{1-\alpha \max(L_i)\}} + (1 - \alpha m\{1 - \alpha \max(L_i)\})^t \left\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^*\right\|_2^2$$

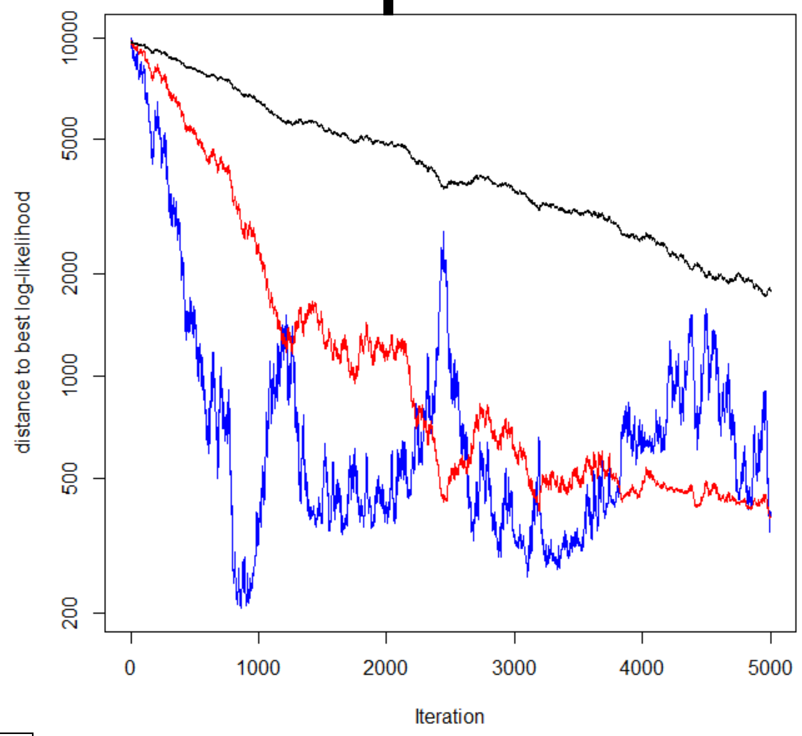
- Proof: See Needell et al (2016), here an arXiv version: https://arxiv.org/abs/1310.5715



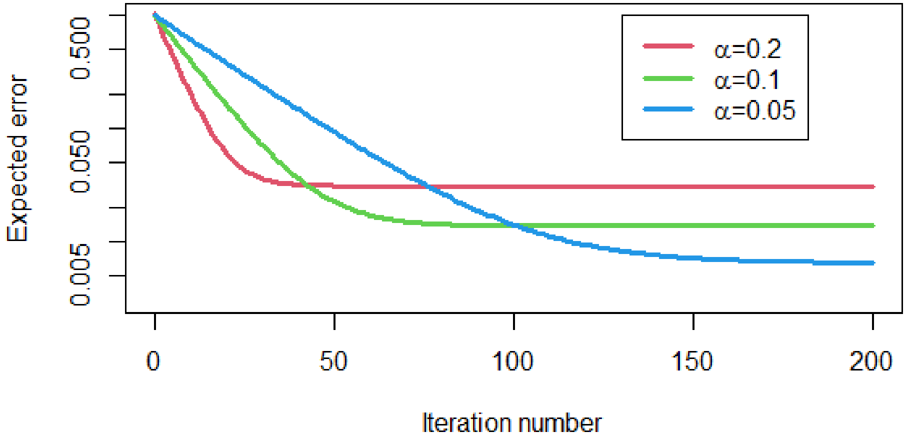
Theoretical behaviour of above bound for expected distance to optimum for s=0.5, m=2, $\max(L_i)$=2, $\varepsilon_0 = \left\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^*\right\|_2^2 = 1$

Stochastic steepest descent: empirical examples

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



- Compare with theoretical result:



Theoretical behaviour of above bound for expected distance to optimum for s=0.5, m=2, $\max(L_i)=2$, $\varepsilon_0 = \left\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^*\right\|_2^2 = 1$

Convergence analysis for decreasing step size

- Theorem 3 (Bottou et al): Consider the finite sum case of the optimization problem, assume Assumptions (A) and that the step size is decreasing as $\alpha_t = \frac{\beta}{t+\gamma}$ with $\beta > \frac{1}{m}$, $\gamma > 0$, $\alpha_0 \leq 1/\{L \max(w, 1)\}$. Then, we have the following convergence result:

$$\mathrm{E}\left[g\left(\boldsymbol{x}^{(t)}\right)\right] - g(\boldsymbol{x}^*) \leq \nu/(\gamma + t),$$

where

$$\nu = \max\left\{\frac{\beta^2 Ls}{2(\beta m - 1)}, (\gamma + 1)\left(g\left(\boldsymbol{x}^{(0)}\right) - g(\boldsymbol{x}^*)\right)\right\}.$$

- Proof: See Bottou et al (2018), https://arxiv.org/pdf/1606.04838.pdf

Convergence analysis for decreasing step size

- Note that
$$\mathrm{E}\left[g\left(\boldsymbol{x}^{(t)}\right)\right] - g(\boldsymbol{x}^*) \approx \nu/(\gamma + t),$$
means sublinear convergence since
$$\frac{\left\{\mathrm{E}\left[g\left(\boldsymbol{x}^{(t+1)}\right)\right] - g(\boldsymbol{x}^*)\right\}}{\left\{\mathrm{E}\left[g\left(\boldsymbol{x}^{(t)}\right)\right] - g(\boldsymbol{x}^*)\right\}} \approx \frac{\gamma+t}{\gamma+t+1} \to 1 \ \ (\text{for } t \to \infty)$$

- A bound like
$$\mathrm{E}\left[g\left(\boldsymbol{x}^{(t)}\right)\right] - g(\boldsymbol{x}^*) \leq \nu^t \ \text{ with } 0 < \nu < 1$$
would lead to linear convergence

- So, SSD with $\alpha_t = \frac{\beta}{t+\gamma}$ gives only sublinear convergence

Lipschitz continuous functions and matrix norms

- If f has a Hessian matrix **f''** with a bounded spectral norm (by L), the gradient **f'** is Lipschitz continuous with L:

$$\|\boldsymbol{f}''(\boldsymbol{x})\|_{Spectral} \leq L \text{ for all } \boldsymbol{x} \Rightarrow \boldsymbol{f}' \text{Lipschitz continuous with } L$$

- Most often when writing $\|\cdot\|$, we have a norm for a vector inside the norm-signs (and $\|\boldsymbol{x}\|_2$ can be interpreted as length of vector $\boldsymbol{x}$)
- There are also matrix-norms, and the spectral norm is one example: $\|\boldsymbol{A}\|_{Spectral} = \sqrt{\lambda_{max}(\boldsymbol{A}^T\boldsymbol{A})}$ where $\lambda_{max}(\cdot)$ is the largest eigenvalue of the matrix inside
- Spectral norm and Euclidian norm are *compatible* in the sense that for any $\boldsymbol{A} \in \mathbb{R}^{\boldsymbol{n}\times\boldsymbol{n}}$ and $\boldsymbol{x} \in \mathbb{R}^{\boldsymbol{n}}$, we have

$$\|\boldsymbol{A}\boldsymbol{x}\|_2 \leq \|\boldsymbol{A}\|_{Spectral}\|\boldsymbol{x}\|_2$$

SSD convergence analysis - exercise

Optimisation in a least squares situation,

- $g(\boldsymbol{b}) = \frac{1}{n}\|\boldsymbol{X}\boldsymbol{b}-\boldsymbol{y}\|_2^2 = \frac{1}{n}\sum_{i=1}^{n} g_i(\boldsymbol{b})$ with $g_i(\boldsymbol{b}) = \left(\boldsymbol{x}_i^T\boldsymbol{b}-y_i\right)^2$
- $\boldsymbol{g}'(\boldsymbol{b}) = \frac{2}{n}\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{b}-\boldsymbol{y}) = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{g}_i{}'(\boldsymbol{b})$ with $\boldsymbol{g}_i'(\boldsymbol{b}) = 2(\boldsymbol{x}_i^T\boldsymbol{b}-y_i)\boldsymbol{x}_i$
- $\boldsymbol{g}''(\boldsymbol{b}) = \frac{2}{n}\boldsymbol{X}^T\boldsymbol{X}$

R uniformly distributed on {1, ..., n}

Compute for (i) general $\boldsymbol{x}_i$, (ii) $\boldsymbol{x}_i = \begin{pmatrix}1\\ w_i\end{pmatrix}$ (straight line regression):

a) $\|\boldsymbol{g}_i{}'(\boldsymbol{b})\|_2^2 =$

b) $\mathrm{E}\|\boldsymbol{g}_R{}'(\boldsymbol{b})\|_2^2 =$

Compute for general $\boldsymbol{x}_i, \boldsymbol{X}$:

c) $\mathrm{E}[\boldsymbol{g}_{\mathrm{R}}'(\mathbf{b})] =$

d) $\|\boldsymbol{g}''(\mathbf{b})\|_{\text{Spectral}} =$

SSD convergence analysis - exercise

Optimisation in a least squares situation,

- $g(\boldsymbol{b}) = \frac{1}{n}\|\boldsymbol{X}\boldsymbol{b} - \boldsymbol{y}\|_2^2 = \frac{1}{n}\sum_{i=1}^{n} g_i(\boldsymbol{b})$ with $g_i(\boldsymbol{b}) = \left(\boldsymbol{x}_i^T\boldsymbol{b} - y_i\right)^2$
- $\boldsymbol{g}'(\boldsymbol{b}) = \frac{2}{n}\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{b} - \boldsymbol{y}) = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{g}_i{}'(\boldsymbol{b})$ with $\boldsymbol{g}_i'(\boldsymbol{b}) = 2(\boldsymbol{x}_i^T\boldsymbol{b} - y_i)\boldsymbol{x}_i$
- $\boldsymbol{g}''(\boldsymbol{b}) = \frac{2}{n}\boldsymbol{X}^T\boldsymbol{X}$

R uniformly distributed on {1, ..., n}

Compute for (i) general $\boldsymbol{x}_i$, (ii) $\boldsymbol{x}_i = \begin{pmatrix} 1 \\ w_i \end{pmatrix}$ (straight line regression):

a) $\|\boldsymbol{g}_i{}'(\boldsymbol{b})\|_2^2 = 4\left(\mathbf{x}_\mathrm{i}^\mathrm{T}\mathbf{b} - \mathrm{y_i}\right)^2 \mathrm{x}_i^T \mathrm{x_i} = 4(b_1 + b_2 w_i - y_i)^2(1 + \mathrm{w_i^2})$

b) $\mathrm{E}\|\boldsymbol{g}_R{}'(\boldsymbol{b})\|_2^2 = \frac{1}{n}\sum_i \|\boldsymbol{g}_i{}'(\boldsymbol{b})\|_2^2 = \ldots$

Compute for general $\boldsymbol{x}_i, \boldsymbol{X}$:

c) $\mathrm{E}[\boldsymbol{g}_\mathrm{R}'(\mathbf{b})] = \frac{1}{\mathrm{n}}\sum_i g_i'(\boldsymbol{b}) = g'(\boldsymbol{b}) = \ldots$

d) $\|\boldsymbol{g}''(\mathbf{b})\|_{\text{Spectral}} = \frac{2}{n}\sqrt{\lambda_{\max}\left((X^TX)^T(X^TX)\right)} = \frac{2}{n}\lambda_{max}(X^TX)$

Maximum likelihood estimator (MLE)

- The MLE is solution of $g(\widehat{\boldsymbol{\beta}}) = \max g(\boldsymbol{b})$ with $g(\widehat{\boldsymbol{\beta}}) = \text{log–likelihood}(\widehat{\boldsymbol{\beta}}, \boldsymbol{X}, \boldsymbol{y}) = \sum_{i=1}^{n} \text{log–likelihood}(\widehat{\boldsymbol{\beta}}, \boldsymbol{x_i}, y_i)$ (the latter equation requires independence of observations)
- In the simple case of normally distributed observations, MLE=LSE and we have an algebraic solution
- Otherwise, we need usually iterative methods to compute the MLE
- If the data is from an exponential family, the function g is concave ($-g$ is convex)