Drawbacks of Gradient Method

- Choice of ε affects speed of convergence.
  - If ε is small: Slow convergence
  - If ε is large: Fast but less accurate
- Appropriately choosing ε is not easy in practice.
- Demonstrations:
  - demo(1): appropriate ε
  - demo(2): small ε
  - demo(3): large ε

Alternative Formulation

$$\boldsymbol{\psi} = \underset{\boldsymbol{b} \perp \{\boldsymbol{\psi}_i\}_{i=1}^{k-1}}{\operatorname{argmax}} \left(\frac{1}{n} \sum_{i=1}^{n} \langle \boldsymbol{b}, \boldsymbol{x}_i \rangle^4 - 3 \right)^2 \text{ subject to } \|\boldsymbol{b}\| = 1$$

- $\boldsymbol{\psi}$ is given by $\boldsymbol{\psi}_{max}$ or $\boldsymbol{\psi}_{min}$:

$$\boldsymbol{\psi}_{max} = \underset{\boldsymbol{b} \perp \{\boldsymbol{\psi}_i\}_{i=1}^{k-1}}{\operatorname{argmax}} \frac{1}{n} \sum_{i=1}^{n} \langle \boldsymbol{b}, \boldsymbol{x}_i \rangle^4 \text{ subject to } \|\boldsymbol{b}\|^2 = 1$$

$$\boldsymbol{\psi}_{min} = \underset{\boldsymbol{b} \perp \{\boldsymbol{\psi}_i\}_{i=1}^{k-1}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \langle \boldsymbol{b}, \boldsymbol{x}_i \rangle^4 \text{ subject to } \|\boldsymbol{b}\|^2 = 1$$

Lagrangian

- For the moment, ignore $\boldsymbol{b} \perp \{\boldsymbol{\psi}_i\}_{i=1}^{k-1}$.
- In either minimization or maximization case, Lagrangian is given by

$$L(\boldsymbol{b}, \lambda) = \frac{1}{n} \sum_{i=1}^{n} \langle \boldsymbol{b}, \boldsymbol{x}_i \rangle^4 + \lambda(\|\boldsymbol{b}\|^2 - 1)$$

- Stationary point (necessary condition):

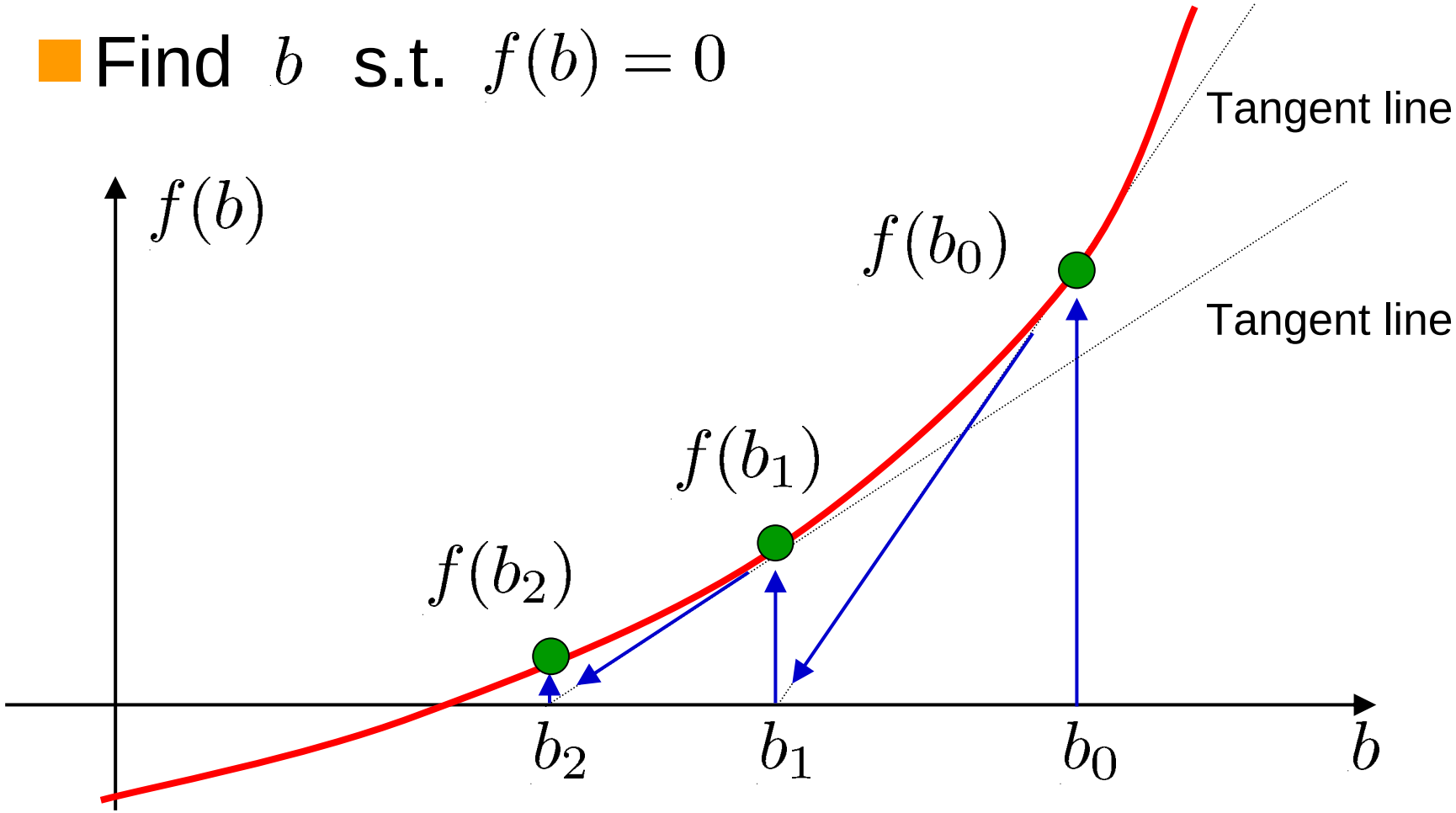
$$\frac{\partial L}{\partial \boldsymbol{b}} = \frac{4}{n} \sum_{i=1}^{n} \boldsymbol{x}_i \langle \boldsymbol{b}, \boldsymbol{x}_i \rangle^3 + 2\lambda \boldsymbol{b} = \boldsymbol{0}$$

- We want to find $\boldsymbol{b}$ which satisfies

$$f(\boldsymbol{b}) = \frac{\partial L}{\partial \boldsymbol{b}} = \boldsymbol{0}$$

Newton Method (1-Dim.)

- Find b s.t. $f(b) = 0$



$$b_{k+1} \longleftarrow b_k - \frac{f(b_k)}{f'(b_k)}$$

Newton Method (Multi-Dim.)

- Find $\boldsymbol{b}$ s.t. $f(\boldsymbol{b}) = \mathbf{0}$

$$\boldsymbol{b}_{k+1} \longleftarrow \boldsymbol{b}_k - \left(\frac{\partial f}{\partial \boldsymbol{b}} \bigg|_{\boldsymbol{b}=\boldsymbol{b}_k} \right)^{-1} f(\boldsymbol{b}_k)$$

- Note:
  - $f(\boldsymbol{b})$ is a d -dimensional vector.
  - $\frac{\partial f}{\partial \boldsymbol{b}}$ is a d -dimensional matrix.

Newton Approach

$$f(\boldsymbol{b}) = \mathbf{0} \quad \text{subject to } \|\boldsymbol{b}\|^2 = 1 \text{ and } \boldsymbol{b} \perp \{\boldsymbol{\psi}_i\}_{i=1}^{k-1}$$

- Repeat the following until convergence:
  - Update $\boldsymbol{b}$ by Newton method to satisfy the stationary point condition $\frac{\partial L}{\partial \boldsymbol{b}} = \mathbf{0}$:

$$\boldsymbol{b} \longleftarrow \boldsymbol{b} - \left(\frac{\partial f}{\partial \boldsymbol{b}}\right)^{-1} f(\boldsymbol{b})$$

  - Modify $\boldsymbol{b}$ to satisfy $\boldsymbol{b} \perp \{\boldsymbol{\psi}_i\}_{i=1}^{k-1}$:

$$\boldsymbol{b} \longleftarrow \boldsymbol{b} - \sum_{i=1}^{k-1} \langle \boldsymbol{b}, \boldsymbol{\psi}_i \rangle \boldsymbol{\psi}_i$$

  - Modify $\boldsymbol{b}$ to satisfy $\|\boldsymbol{b}\| = 1$:

$$\boldsymbol{b} \longleftarrow \boldsymbol{b} / \|\boldsymbol{b}\|$$

Newton Approach (cont.)

$$f(\boldsymbol{b}) = \frac{4}{n}\sum_{i=1}^{n} \boldsymbol{x}_i \langle \boldsymbol{b}, \boldsymbol{x}_i \rangle^3 + 2\lambda \boldsymbol{b}$$

$$\frac{\partial f}{\partial \boldsymbol{b}} = \frac{12}{n}\sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^\top \langle \boldsymbol{b}, \boldsymbol{x}_i \rangle^2 + 2\lambda \boldsymbol{I}_d$$

- However,
  - Calculating inverse $\left(\frac{\partial f}{\partial \boldsymbol{b}}\right)^{-1}$ in each step is computationally demanding.
  - λ is unknown.

Approximation

- $$\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i\boldsymbol{x}_i^\top\langle\boldsymbol{b},\boldsymbol{x}_i\rangle^2 \approx \left(\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i\boldsymbol{x}_i^\top\right)\left(\frac{1}{n}\sum_{i=1}^{n}\langle\boldsymbol{b},\boldsymbol{x}_i\rangle^2\right) = \boldsymbol{I}_d$$

$$\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i\boldsymbol{x}_i^\top = \boldsymbol{I}_d$$

$$\|\boldsymbol{b}\| = 1$$

- Then

$$\frac{\partial f}{\partial \boldsymbol{b}} = \frac{12}{n}\sum_{i=1}^{n}\boldsymbol{x}_i\boldsymbol{x}_i^\top\langle\boldsymbol{b},\boldsymbol{x}_i\rangle^2 + 2\lambda\boldsymbol{I}_d$$

$$\approx (12+2\lambda)\,\boldsymbol{I}_d$$

- Calculating inverse is easy!

Approximation (cont.)

$$f(\boldsymbol{b}) = \frac{4}{n}\sum_{i=1}^{n} \boldsymbol{x}_i \langle \boldsymbol{b}, \boldsymbol{x}_i \rangle^3 + 2\lambda \boldsymbol{b} \qquad \frac{\partial f}{\partial \boldsymbol{b}} \approx (12 + 2\lambda)\, \boldsymbol{I}_d$$

- Approximate updating rule is

$$\boldsymbol{b} \longleftarrow \frac{1}{12 + 2\lambda}\left(12\boldsymbol{b} - \frac{4}{n}\sum_{i=1}^{n} \boldsymbol{x}_i \langle \boldsymbol{b}, \boldsymbol{x}_i \rangle^3\right)$$

- $\boldsymbol{b}$ is normalized so we can ignore constant:

$$\boldsymbol{b} \longleftarrow 3\boldsymbol{b} - \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i \langle \boldsymbol{b}, \boldsymbol{x}_i \rangle^3$$

- λ can be removed!

Examples

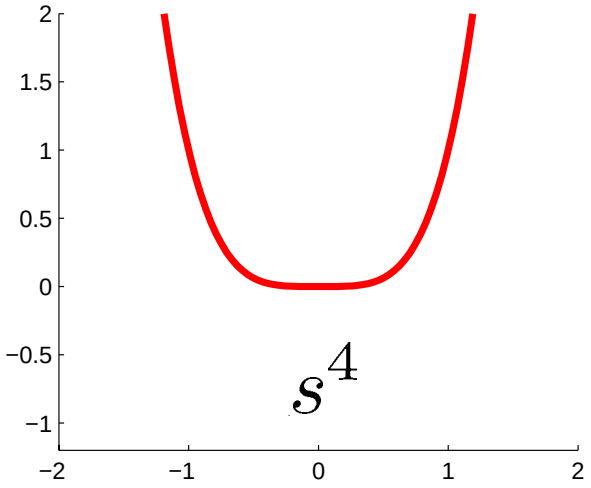
- Demonstrations:
  - demo(1): Gradient ascent with appropriate ε
  - demo(4): Approximate Newton

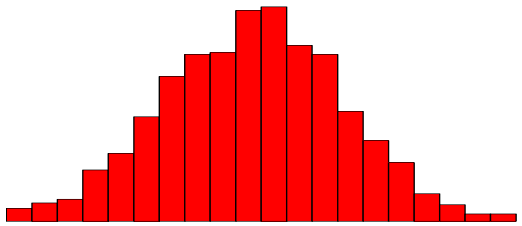
- Approximate Newton
  - is much faster than gradient ascent.
  - does not include any tuning parameter!

Outliers

- Outliers: Irregular large values
- If a Gaussian component contains outliers, its non-Gaussianity becomes very large since kurtosis contains 4th power.

$$\frac{1}{n}\sum_{i=1}^{n}\langle \boldsymbol{b}, \boldsymbol{x}_i\rangle^4$$





Examples

- Demonstrations:
  - demo(4): Approximate Newton (without outlier)
  - demo(5): Approximate Newton (with single outlier)

- A single outlier can totally corrupt the result.
- Influence of outliers should be deemphasized!

General Non-Gaussian Measures

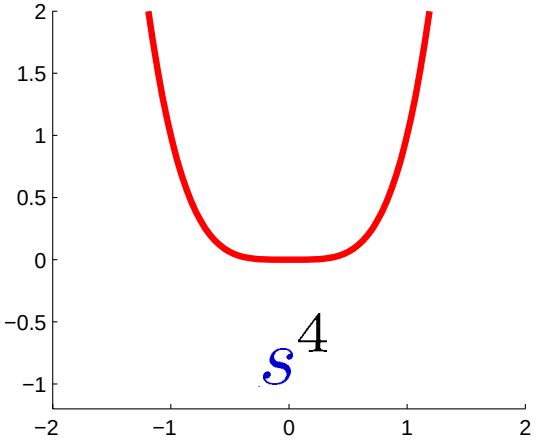
- For some function $G(s)$, we define a general non-Gaussian measure by

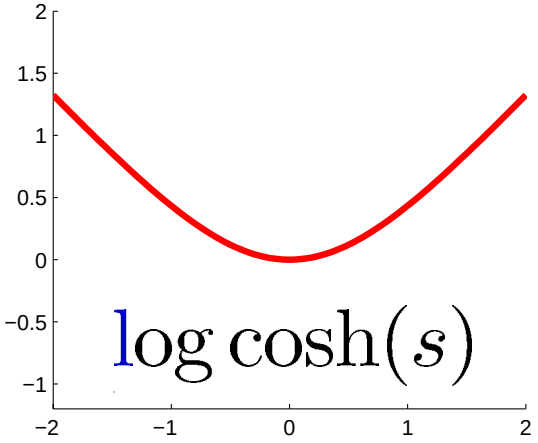
$$\frac{1}{n}\sum_{i=1}^{n} G(\langle \boldsymbol{b}, \boldsymbol{x}_i \rangle)$$

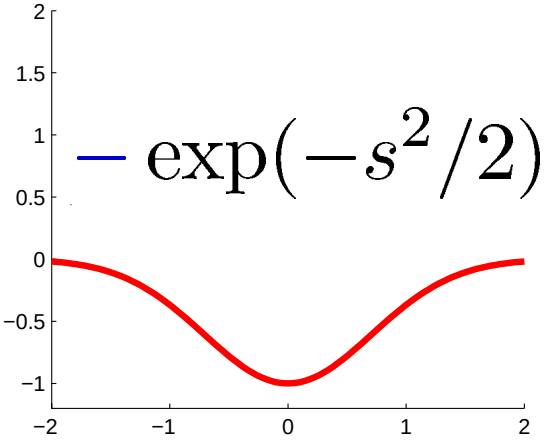
- $G(s) = s^4$ corresponds to Kurtosis.
- To suppress the effect of outliers, using a "gentler" function would be appropriate.

General Non-Gaussian Measures

- Examples of smooth functions:
  - $G(s) = \log\cosh(s)$
  - $G(s) = -\exp(-s^2/2)$







Approximate Newton Procedure

- Approximate Newton procedure for centered and sphered data:
  - Update $\boldsymbol{b}$ to satisfy the stationary-point condition:

$$\boldsymbol{b} \longleftarrow \frac{1}{n}\boldsymbol{b}\sum_{i=1}^{n} g'(\langle \boldsymbol{b}, \boldsymbol{x}_i \rangle) - \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i g(\langle \boldsymbol{b}, \boldsymbol{x}_i \rangle)$$

$$g(s) = G'(s)$$

  - Modify $\boldsymbol{b}$ to satisfy $\boldsymbol{b} \perp \{\boldsymbol{\psi}_i\}_{i=1}^{k-1}$:

$$\boldsymbol{b} \longleftarrow \boldsymbol{b} - \sum_{i=1}^{k-1} \langle \boldsymbol{b}, \boldsymbol{\psi}_i \rangle \boldsymbol{\psi}_i$$

  - Modify $\boldsymbol{b}$ to satisfy $||\boldsymbol{b}|| = 1$:

$$\boldsymbol{b} \longleftarrow \boldsymbol{b}/||\boldsymbol{b}||$$

Derivatives

- Derivatives:
  - $(s^4)' = 4s^3$
    $(4s^3)' = 12s^2$
  - $(\log\cosh(s))' = \tanh(s)$
    $(\tanh(s))' = 1 - \tanh^2(s)$
  - $(-\exp(-s^2/2))' = s\exp(-s^2/2)$
    $(s\exp(-s^2/2))' = (1 - s^2)\exp(-s^2/2)$

Examples

- Demonstrations:
  - demo(5): Approximate Newton with Kurtosis

$$g(s) = 4s^3$$

  - demo(6): Approximate Newton with log(cosh)

$$g(s) = \tanh(s)$$

- Approximate Newton with log(cosh) is robust against outliers!

Homework

- Prove that approximate Newton updating rule is given by

$$\boldsymbol{b} \longleftarrow \frac{1}{n}\boldsymbol{b}\sum_{i=1}^{n} g'(\langle \boldsymbol{b}, \boldsymbol{x}_i \rangle) - \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i g(\langle \boldsymbol{b}, \boldsymbol{x}_i \rangle)$$

under the following approximation:

$$\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^\top g'(\langle \boldsymbol{b}, \boldsymbol{x}_i \rangle) \approx \frac{1}{n}\sum_{i=1}^{n} g'(\langle \boldsymbol{b}, \boldsymbol{x}_i \rangle) \boldsymbol{I}_d$$

Homework (cont.)

- Implement projection pursuit algorithms.
- Apply the algorithms to your data set and extract two non-Gaussian directions in your data.
- Project your data onto the two-dimensional subspace and find something interesting (data mining!).