Advanced Data Analysis: Projection Pursuit (2)

Masashi Sugiyama (Computer Science)

W8E-406, sugi@cs.titech.ac.jp

http://sugiyama-www.cs.titech.ac.jp/~sugi

Projection Pursuit

- Find the most non-Gaussian direction.
- Original formulation: maximize distance of kurtosis from 3

$$\boldsymbol{\psi} = \mathop{\mathrm{argmax}}_{\boldsymbol{b} \in \mathbb{R}^d} \left(\frac{1}{n} \sum_{i=1}^{n} \langle \boldsymbol{b}, \widetilde{\boldsymbol{x}}_i \rangle^4 - 3 \right)^2 \quad \text{subject to } \|\boldsymbol{b}\| = 1$$

- Gradient ascent algorithm
  - $\boldsymbol{b} \longleftarrow \boldsymbol{b} + \varepsilon \left(\frac{1}{n} \sum_{i=1}^{n} \langle \boldsymbol{b}, \widetilde{\boldsymbol{x}}_i \rangle^4 - 3 \right) \frac{1}{n} \sum_{i=1}^{n} \widetilde{\boldsymbol{x}}_i \langle \boldsymbol{b}, \widetilde{\boldsymbol{x}}_i \rangle^3$
  - $\boldsymbol{b} \longleftarrow \boldsymbol{b} / \|\boldsymbol{b}\|$

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

- Maximize or minimize kurtosis
  - $$\boldsymbol{\psi}_{max} = \underset{\boldsymbol{b}\in\mathbb{R}^d}{\mathrm{argmax}} \left[\frac{1}{n} \sum_{i=1}^{n} \langle \boldsymbol{b}, \widetilde{\boldsymbol{x}}_i \rangle^4 \right]$$ $$\text{subject to } \|\boldsymbol{b}\|^2 = 1$$
  - $$\boldsymbol{\psi}_{min} = \underset{\boldsymbol{b}\in\mathbb{R}^d}{\mathrm{argmin}} \left[\frac{1}{n} \sum_{i=1}^{n} \langle \boldsymbol{b}, \widetilde{\boldsymbol{x}}_i \rangle^4 \right]$$ $$\text{subject to } \|\boldsymbol{b}\|^2 = 1$$
- $\boldsymbol{\psi}$ is given by $\boldsymbol{\psi}_{max}$ or $\boldsymbol{\psi}_{min}$.

Lagrangian

- In either minimization or maximization case, Lagrangian is given by

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

- Stationary (necessary) condition:

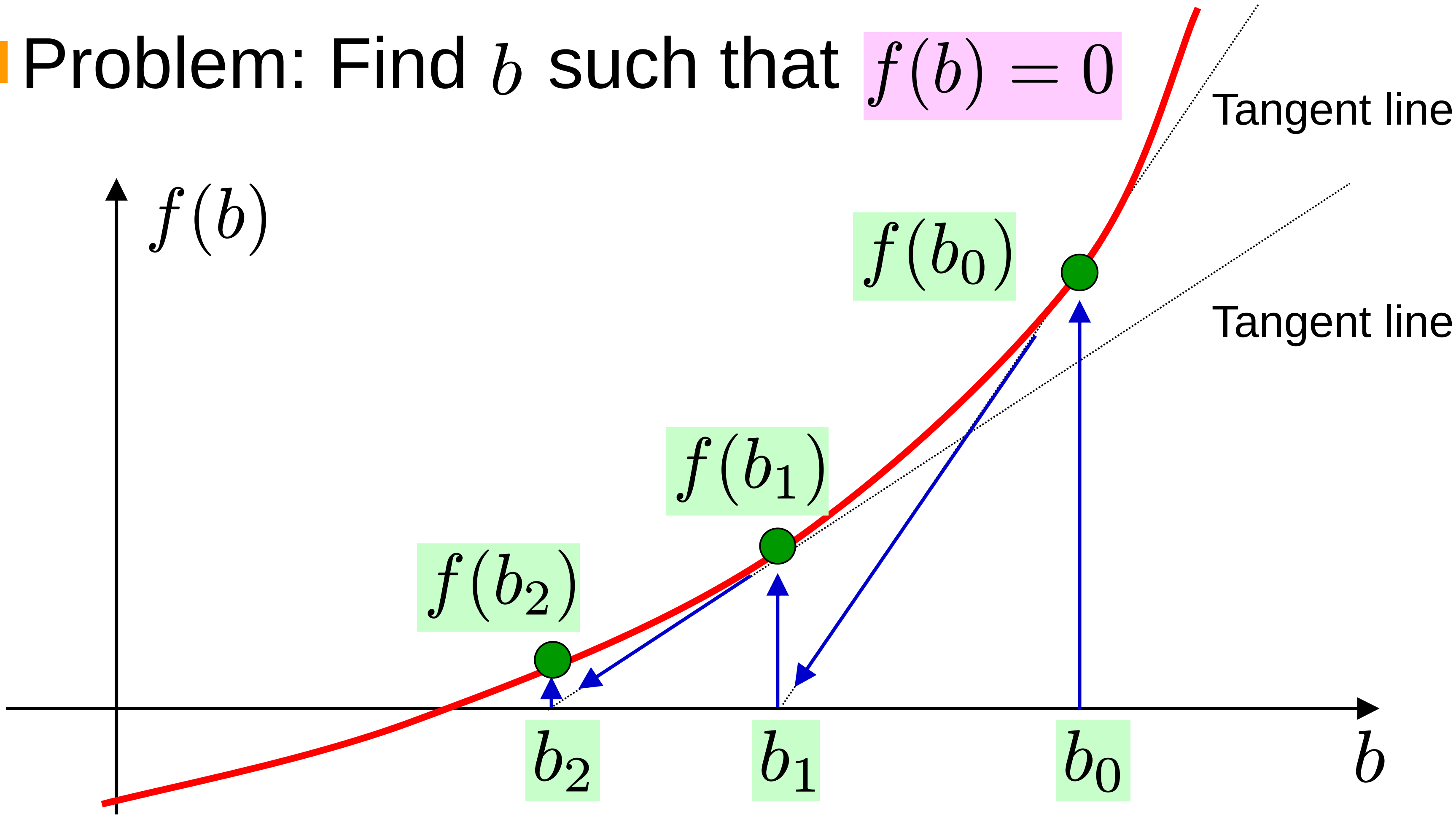
$$\frac{\partial L}{\partial \boldsymbol{b}} = \frac{4}{n} \sum_{i=1}^{n} \widetilde{\boldsymbol{x}}_i \langle \boldsymbol{b}, \widetilde{\boldsymbol{x}}_i \rangle^3 + 2\lambda \boldsymbol{b} = \boldsymbol{0}$$

- We want to find $\boldsymbol{b}$ such that

$$\frac{\partial L}{\partial \boldsymbol{b}} = \boldsymbol{0}$$

Newton Method (1-Dim.)

- Problem: Find b such that $f(b)=0$



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

Newton Method (Multi-Dim.)

- **Problem**: Find $\boldsymbol{b}$ such that $f(\boldsymbol{b}) = \boldsymbol{0}$

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

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

Newton-Based PP Method

- In the current setting,

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

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

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

Approximation

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

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

- Then

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

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

- Calculating inverse is easy!

Approximation (cont.)

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

$$\frac{\partial f}{\partial \boldsymbol{b}} \approx (12+2\lambda)\,\boldsymbol{I}_d$$

- Approximate updating rule is given by

$$\boldsymbol{b} \longleftarrow \frac{2}{6+\lambda}\left(3\boldsymbol{b} - \frac{1}{n}\sum_{i=1}^{n}\widetilde{\boldsymbol{x}}_i\langle\boldsymbol{b},\widetilde{\boldsymbol{x}}_i\rangle^3\right)$$

- $\boldsymbol{b}$ is later normalized, so the scaling factor can be dropped:

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

- The update rule does not depend on λ !

Approximate Newton-Based PP Method

- Problem to be solved:

$$f(\boldsymbol{b}) = \boldsymbol{0} \quad \text{subject to } \|\boldsymbol{b}\|^2 = 1$$

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

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

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

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

Examples

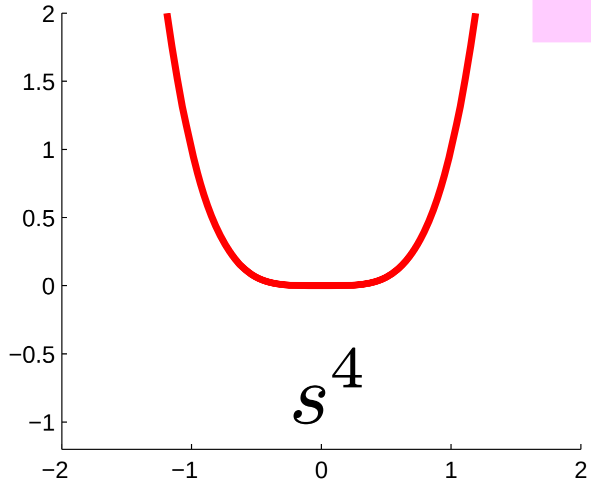
- Demonstrations:
  - demo(1): Gradient ascent with appropriate ε
  - demo(4): Approximate Newton

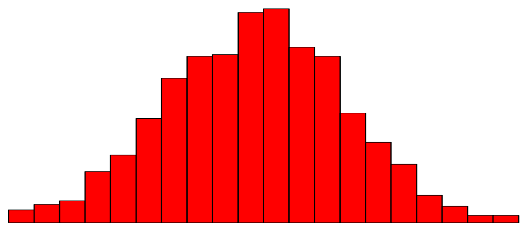
- Approximate Newton
  - is much faster than gradient ascent.
  - does not include any tuning parameter!

Outliers

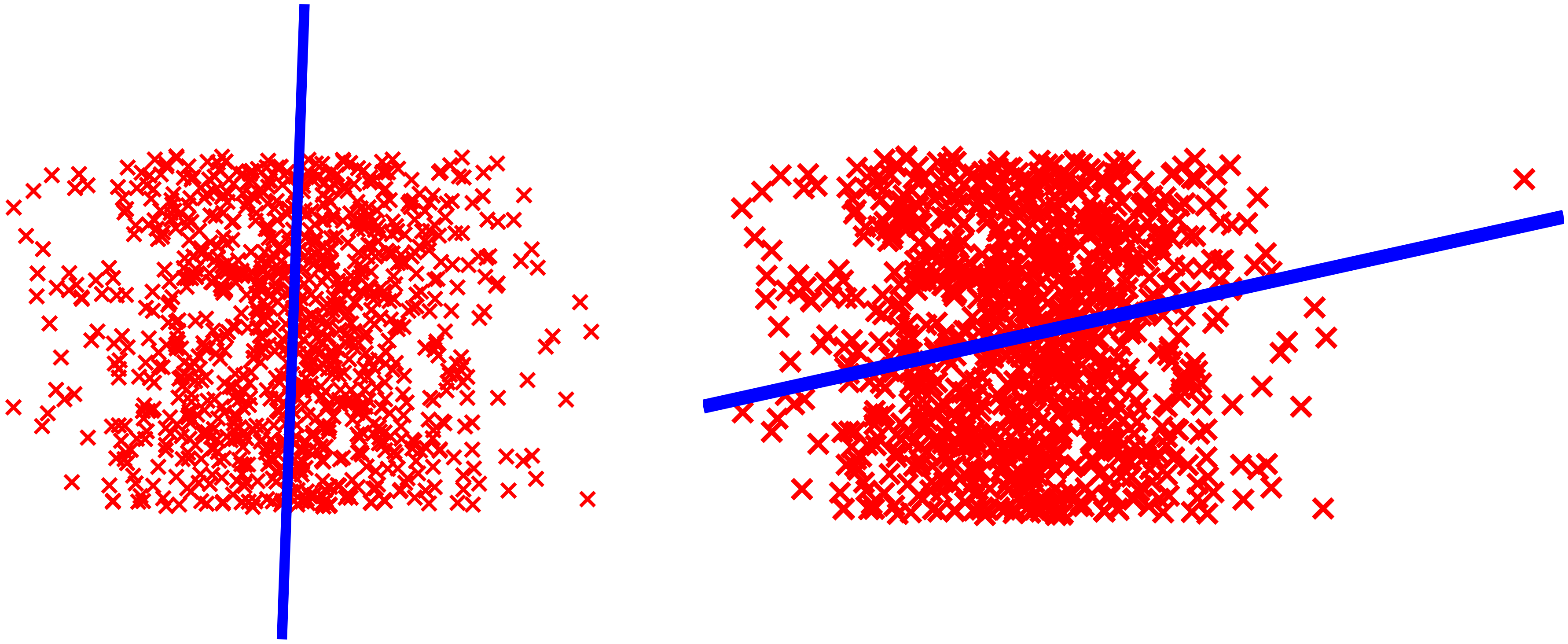
- Outliers: Irregular large values
- If a Gaussian component contains outliers, its non-Gaussianity becomes very large since kurtosis contains 4th power.

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





Examples



Without outlier With single outlier

- A single outlier can totally corrupt the result.
- Influence of outliers needs to be reduced!

General Non-Gaussian Measures

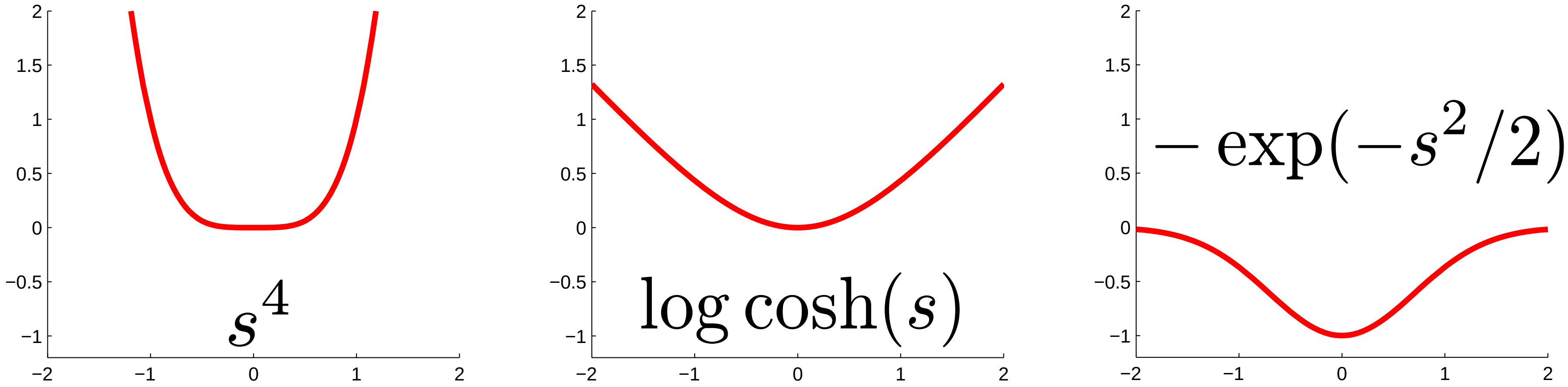
- For some function $G(s)$, we define a general non-Gaussian measure by

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

- $G(s) = s^4$ 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:

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

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

(Homework)

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

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

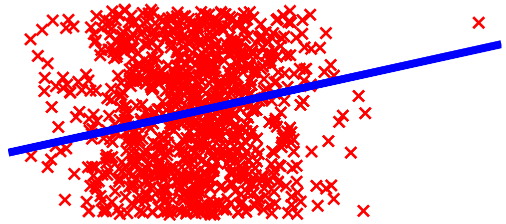
Derivatives

- 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

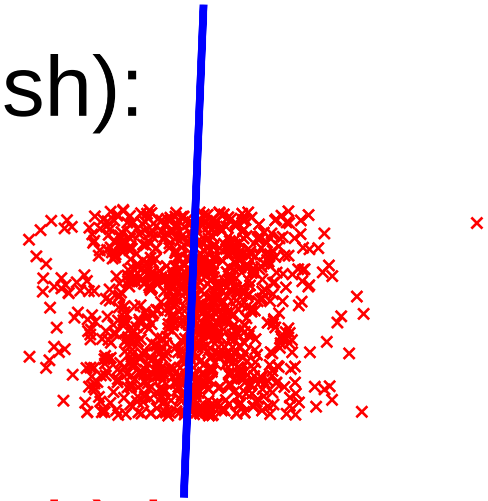
- Approximate Newton with Kurtosis:

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



- Approximate Newton with log(cosh):

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

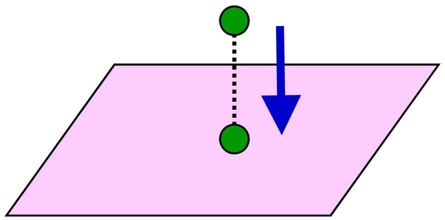


- Approximate Newton with log(cosh) is robust against outliers!

Extracting Several Non-Gaussian Directions

- Running the algorithm many times from different initial points may give different non-Gaussian directions.
- However, this is not computationally efficient.
- Another idea: Find orthogonal directions
- This is achieved by modifying the direction as

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



Full Algorithm

- Center and sphere samples: $\widetilde{\boldsymbol{X}} = (\boldsymbol{X}\boldsymbol{H}^2\boldsymbol{X})^{-\frac{1}{2}}\boldsymbol{X}\boldsymbol{H}$
- For $k = 1, 2, \ldots, m$
  - Repeat until convergence:
    - $\boldsymbol{b} \longleftarrow \frac{1}{n}\boldsymbol{b}\sum_{i=1}^{n} g'(\langle \boldsymbol{b}, \widetilde{\boldsymbol{x}}_i\rangle) - \frac{1}{n}\sum_{i=1}^{n}\widetilde{\boldsymbol{x}}_i g(\langle \boldsymbol{b}, \widetilde{\boldsymbol{x}}_i\rangle)$
    - $\boldsymbol{b} \longleftarrow \boldsymbol{b} - \sum_{i=1}^{k-1}\langle \boldsymbol{b}, \boldsymbol{\psi}_i\rangle \boldsymbol{\psi}_i$
    - $\boldsymbol{b} \longleftarrow \boldsymbol{b}/\|\boldsymbol{b}\|$
  - $\boldsymbol{\psi}_k = \boldsymbol{b}$
- Embed the data $\boldsymbol{x}$ by

$$\boldsymbol{z} = \boldsymbol{B}_{PP}(\boldsymbol{x} - \frac{1}{n}\boldsymbol{X}\boldsymbol{1}_n)$$

$$\widetilde{\boldsymbol{X}} = (\widetilde{\boldsymbol{x}}_1|\widetilde{\boldsymbol{x}}_2|\cdots|\widetilde{\boldsymbol{x}}_n)$$

$$\boldsymbol{X} = (\boldsymbol{x}_1|\boldsymbol{x}_2|\cdots|\boldsymbol{x}_n)$$

$$\boldsymbol{H} = \boldsymbol{I}_n - \frac{1}{n}\boldsymbol{1}_{n\times n}$$

$\boldsymbol{I}_n$: n-dimensional identity matrix

$\boldsymbol{1}_{n\times n}$: $n\times n$ matrix with all ones

$\boldsymbol{1}_n$: n-dimensional vector with all ones

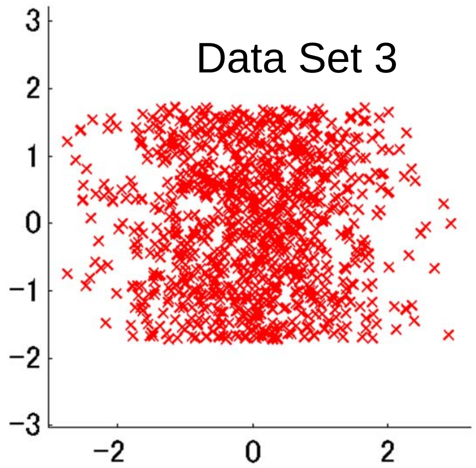
$$\boldsymbol{B}_{PP} = (\boldsymbol{\psi}_1|\boldsymbol{\psi}_2|\cdots|\boldsymbol{\psi}_m)^\top$$

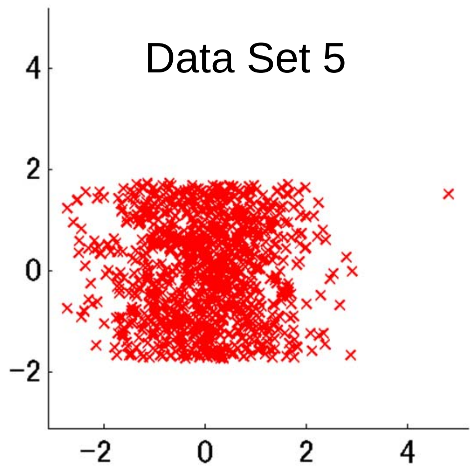
Homework

1. Implement approximate Newton-based PP method with general non-Gaussianity measure and reproduce the 2-dimensional examples with an outlier shown in the class.

   You may create similar (or more interesting) data sets by yourself.

http://sugiyama-www.cs.titech.ac.jp/~sugi/data/DataAnalysis





Homework (cont.)

2. Prove that the approximate Newton updating rule is given by

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

under the following approximation:

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

Schedule

- June 25th: Projection Pursuit (2)
  - Application Deadline to Mini-Conference
- July 2nd: Independent Component Analysis
- July 9th: Preparation for Mini-Conference
- July 16th: Mini-Conference Day 1
- July 23rd: Mini-Conference Day 2