Newton’s method vs Gradient descent & SVD in PCA

Evert Jan Karman & John Stegink

Contents

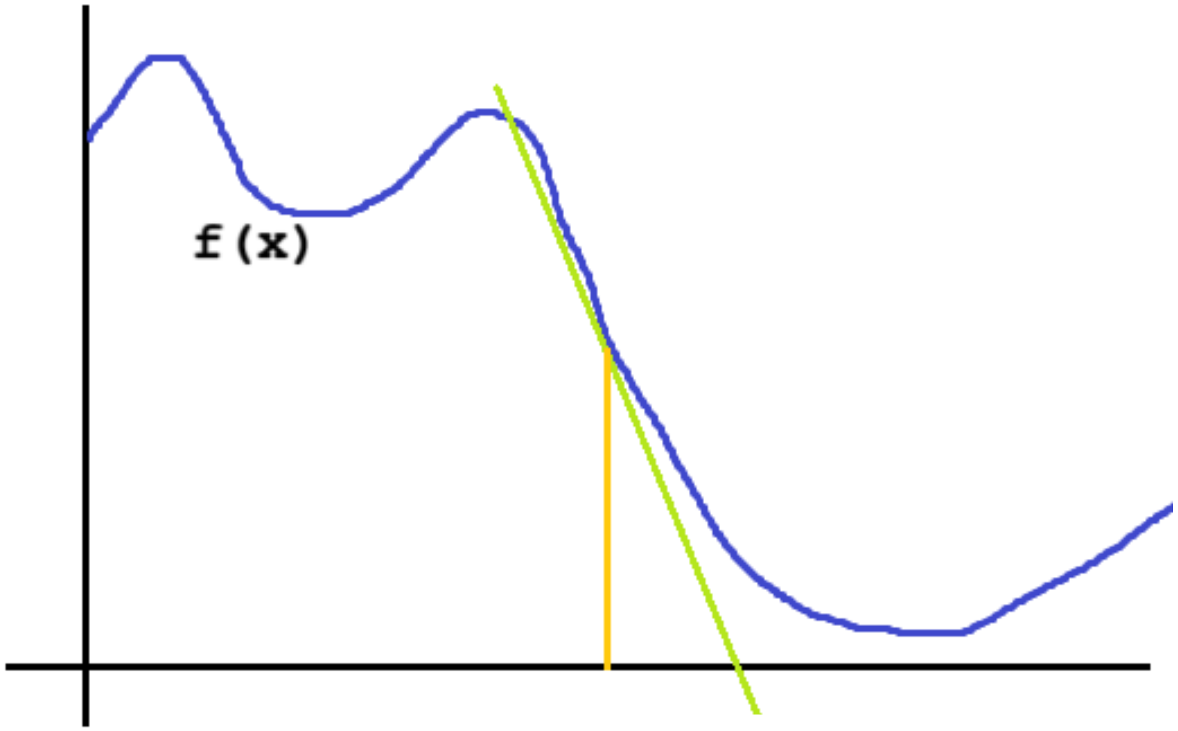
Introduction

- Cost function machine learning
- Optimizing cost function
- Gradient Descent & Newton’s method
- SVD <-> PCA
- SVD <-> EVD

Gradient descent with one variable

- Metaphore descending a hill in the mist
- Continous function and differentiable
- α is the learning rate

Gradient descent with one variable

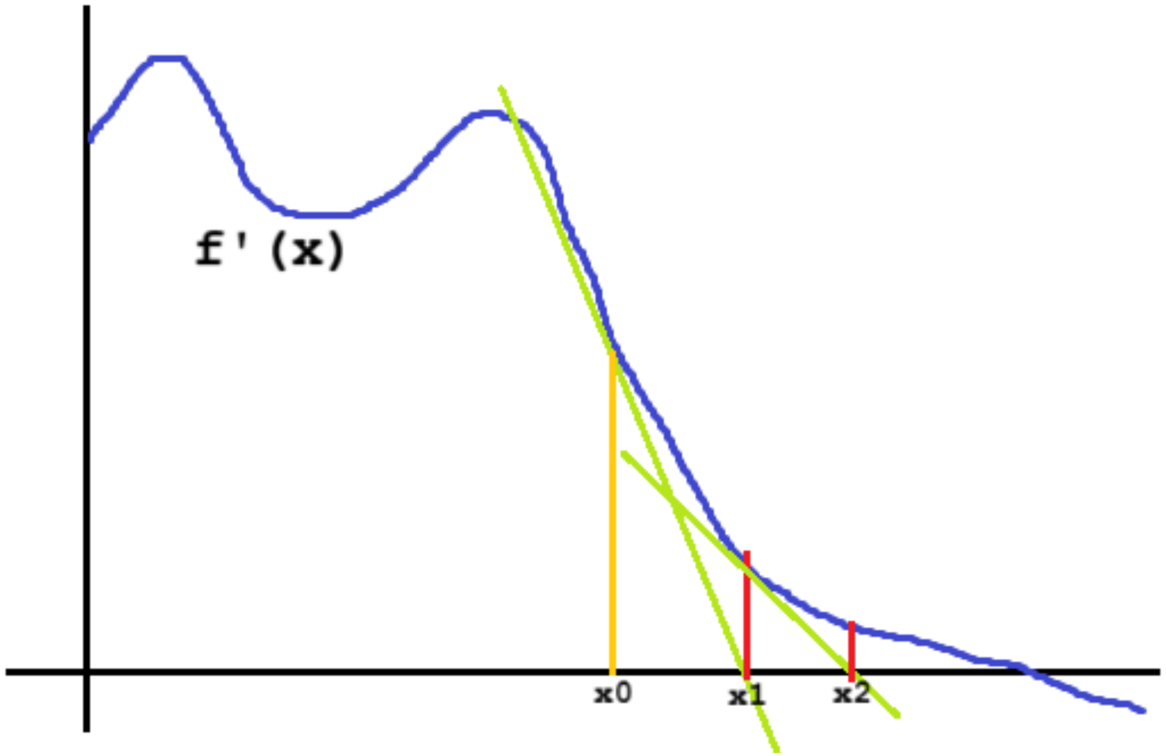


$$x_{k+1} = x_k - f'(x_k) \cdot \alpha$$

Newton’s method with one variable

- Finds zero’s of function
- Continous function and differentiable
- Method:
  - Start at a random point on the curve
  - Determine the tangent line at that point (using the derivative)
  - Determine the point where the tangent line hits the X-axis
  - From the point found in previous step, continue with step, 2 until the value does not change significantly anymore

Newton's method with one variable



$$x_{k+1} = x_k - (f'(x_k)/f''(x_k))$$

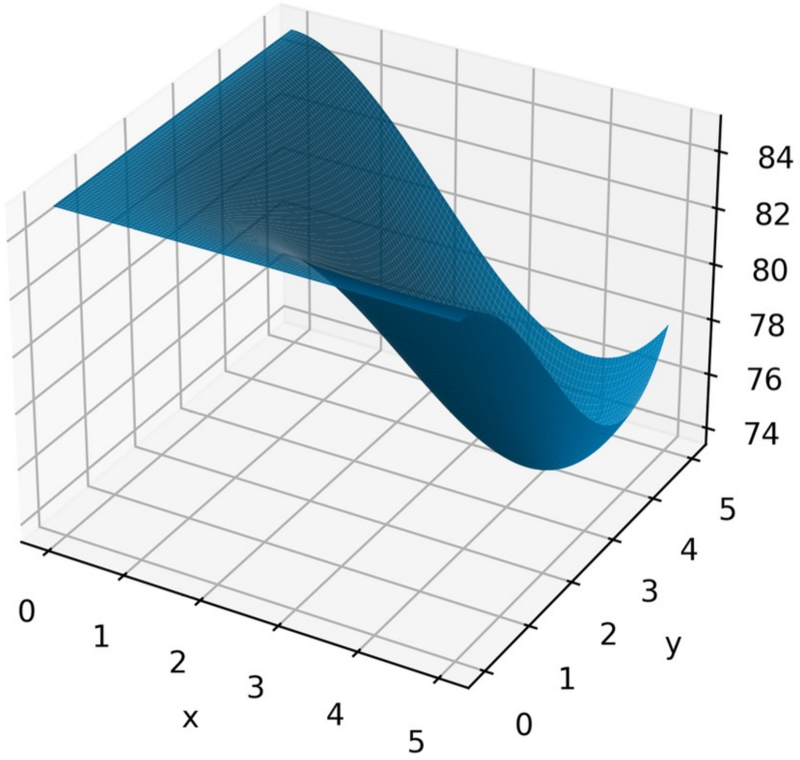
Newton's method with one variable

- Use second derivative if it is minimum or maximum
  - $f''(x) > 0 \Rightarrow f$ has a minimum at x
  - $f''(x) < 0 \Rightarrow f$ has a maximum at x
  - $f''(x) = 0 \Rightarrow$ inconclusive, perhaps an inflection point

Functions with two or more variables and their derivatives

- Multiple variables in function
- Common in machinelearning
- Example find the price of the house
  - Floor area
  - Number of shops in the neighbourhood

Functions with two or more variables and their derivatives



$$f(x, y) = 85 - \frac{1}{90}x^2(x-6)y^2(y-6)$$

Functions with two or more variables and their derivatives

- Derivatives are with respect to **one** variable

$$\frac{\partial f}{\partial x} = f_x(x, y) = \lim_{\Delta x \to 0} \frac{f(x + \Delta x, y) - f(x, y)}{\Delta x}$$

$$\frac{\partial f}{\partial y} = f_y(x, y) = \lim_{\Delta y \to 0} \frac{f(x, y + \Delta y) - f(x, y)}{\Delta y}$$

Gradient descent with two or more variables

- Same metaphore walk to west en north
- Gradient:

$$\nabla f(x,y) = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right)$$

- Definition

$$(x_{k+1}, y_{k+1}) = (x_k, y_k) - \nabla f(x,y) \cdot \alpha$$

Hessian matrix

- Two variables:

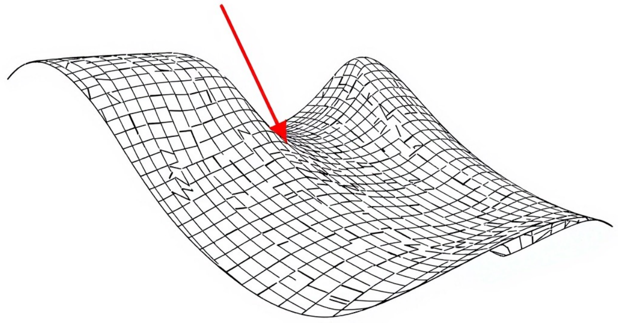
$$H_f(x,y) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix}$$

- General:

$$\mathbf{H}_f = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

Hessian matrix

- Symmetric
  - Clairaut’s theorem
  - Only smooth functions
- Can become very large (>1.000.000.000 parameters) n^2
- Other methods like BFGS
- Local minima, maxima and saddle points (neither max nor min)



Newton's method with two or more variables

- Principles identical
- Minima, maxima and inconclusive
  - If all the eigenvalues of Hf at (x,y) are positive, it's a minimum
  - If all the eigenvalues of Hf at (x,y) are negative, it's a maximum
  - If one of the eigenvalues of Hf at (x,y) are zero, or we have mixed signs, it's inconclusive

$$(x_{k+1}, y_{k+1}) = (x_k, y_k) - (H_f^{-1}(x_k, y_k) \cdot \nabla f(x_k, y_k))$$

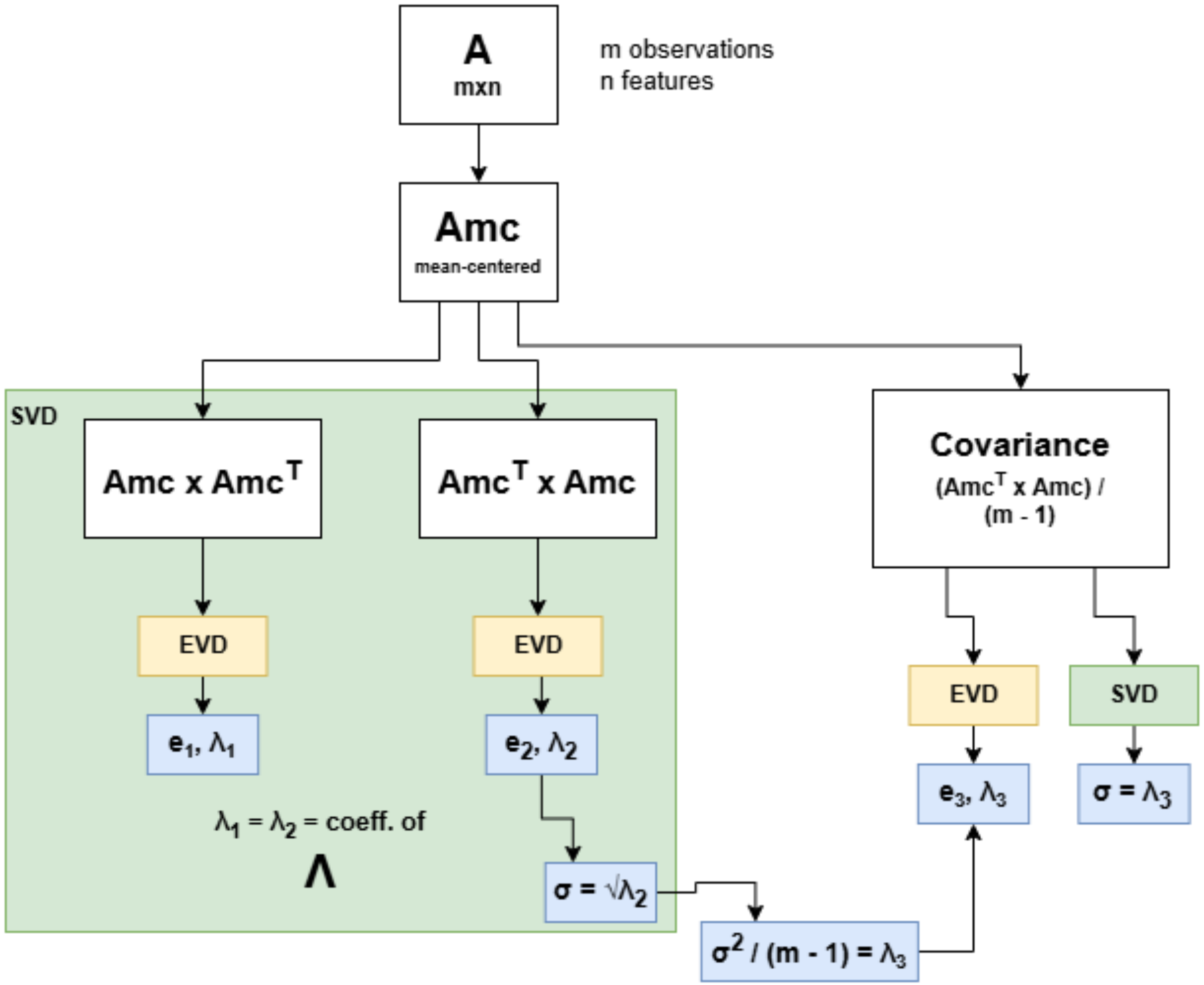
SVD in PCA

- Application of PCA in SVD instead of EVD
- AMC – Mean centered

$$A_{Cov} = \frac{Amc^T \cdot Amc}{m-1}$$

- EVD on AMC -> λ_3

SVD in PCA


A
mxn
m observations
n features
Amc
mean-centered
SVD
Amc x Amc^T
Amc^T x Amc
Covariance
(Amc^T x Amc) /
(m - 1)
EVD
EVD
EVD
SVD
e1, λ1
e2, λ2
e3, λ3
σ = λ3
λ1 = λ2 = coeff. of
Λ
σ = √λ2
σ^2 / (m - 1) = λ3

SVD in PCA

- SVD: calculate $Amc \cdot Amc^T$

  $Amc^T \cdot Amc$

- We get λ_1 and λ_2

$$\sigma = \sqrt{\lambda_2}$$

$$\frac{\sigma^2}{m-1} = \lambda_3$$

SVD in PCA

- Python program for validation
  - Define test matrix A
  - Calculate PCA values
  - Show that EVD on the covariance matrix and SVD on the covariance matrix produces the
    same values
  - Show that SVD on the mean-centered data produces the same values apart from scaling

SVD in PCA

- Advantages:
  - It is more memory intensive
  - On ill-condioned dataset SVD is more stable

Connection EVD and SVD

- EVD:

$$A = Q\Lambda Q^{-1}.$$

- SVD:

$$A = U\Sigma V^T$$

- Calculate U and V:

$$AA^T = \left(U\Sigma V^T\right)\left(U\Sigma V^T\right)^T = U\Sigma V^T V\Sigma^T U^T = U\Sigma^2 U^T$$

$$A^T A = \left(U\Sigma V^T\right)^T\left(U\Sigma V^T\right) = V\Sigma^T U^T U\Sigma V^T = V\Sigma^2 V^T$$

- Also because $\Sigma^2 = \Lambda$

$$AA^T = U\Lambda U^T, \qquad A^T A = V\Lambda V^T,$$

Connection EVD and SVD

- S is square symmetric, and positive semi-definite ($S = S^T$)
- Consider:
  - $U = V$
  - The singular values of S are equal to its eigenvalues ,
  - U is orthogonal, so $U^{-1} = U^T$

$$S = U \Lambda U^T$$

Numerical examples

- Function:

$$f(x,y) = 85 - \frac{1}{90}x^2(x-6)y^2(y-6)$$

$$\nabla f(x,y) = \left(-\frac{1}{90}(3x^2-12x)y^2(y-6), -\frac{1}{90}x^2(x-6)(3y^2-12y)\right)$$

$$\frac{\partial^2 f}{\partial x^2} = -\frac{1}{90}(6x-12)y^2(y-6)$$

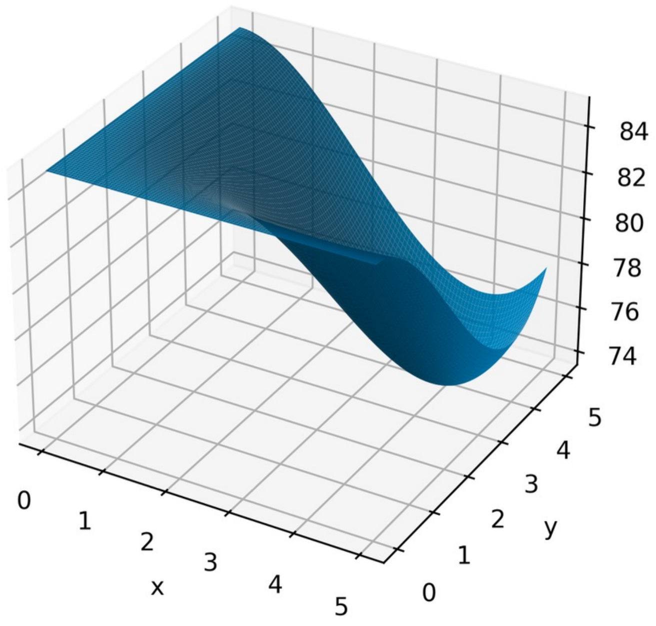
$$\frac{\partial^2 f}{\partial x \partial y} = -\frac{1}{90}(3x^2-12x)(3y^2-12y)$$

$$\frac{\partial^2 f}{\partial y \partial x} = -\frac{1}{90}(3x^2-12x)(3y^2-12y)$$

$$\frac{\partial^2 f}{\partial y^2} = -\frac{1}{90}x^2(x-6)(6y-12)$$

Numerical examples

- Hessian matrix: 5 iterations
- Gradient descent when α = 0.01: 109 iterations
- Gradient descent when α = 0.05: 27 iterations



Colaboration & Reflection