Machine Learning - Winter 2015/16

Bernstein Focus: Neurotechnology Frankfurt

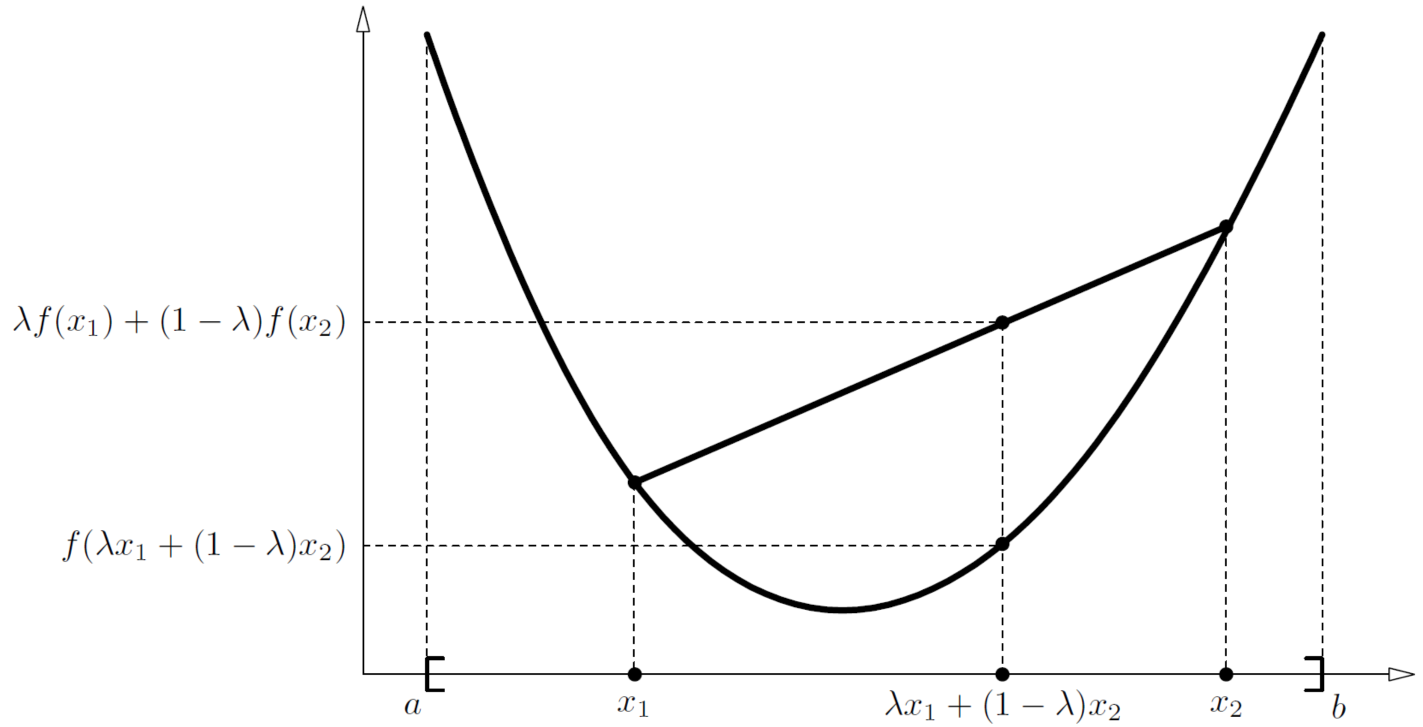
LECTURE : EM ALGORITHM

PREPARED BY
PROF. DR. VISVANATHAN RAMESH

(VERSION 1.0)

Outline

Convex Functions



$\lambda f(x_1) + (1-\lambda) f(x_2)$
$f(\lambda x_1 + (1-\lambda) x_2)$
a
x_1
$\lambda x_1 + (1-\lambda) x_2$
x_2
b

Figure 1: f is *convex* on $[a, b]$ if $f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2)$ $\forall x_1, x_2 \in [a, b], \quad \lambda \in [0, 1]$.

Definitions

Definition 1 *Let f be a real valued function defined on an interval $I = [a, b]$. f is said to be* convex *on I if $\forall x_1, x_2 \in I, \lambda \in [0, 1]$,*

$$f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2).$$

f is said to be strictly convex *if the inequality is strict. Intuitively, this definition states that the function falls below (strictly convex) or is never above (convex) the straight line (the secant) from points $(x_1, f(x_1))$ to $(x_2, f(x_2))$. See Figure (1).*

Definition 2 *f is concave (strictly concave) if $-f$ is convex (strictly convex).*

Theorem 1 *If $f(x)$ is twice differentiable on $[a, b]$ and $f''(x) \geq 0$ on $[a, b]$ then $f(x)$ is convex on $[a, b]$.*

Jensen's Inequality

Theorem 2 (Jensen's inequality) *Let f be a convex function defined on an interval I. If $x_1, x_2, \ldots, x_n \in I$ and $\lambda_1, \lambda_2, \ldots, \lambda_n \geq 0$ with $\sum_{i=1}^n \lambda_i = 1$,*

$$f\left(\sum_{i=1}^n \lambda_i x_i\right) \leq \sum_{i=1}^n \lambda_i f(x_i)$$

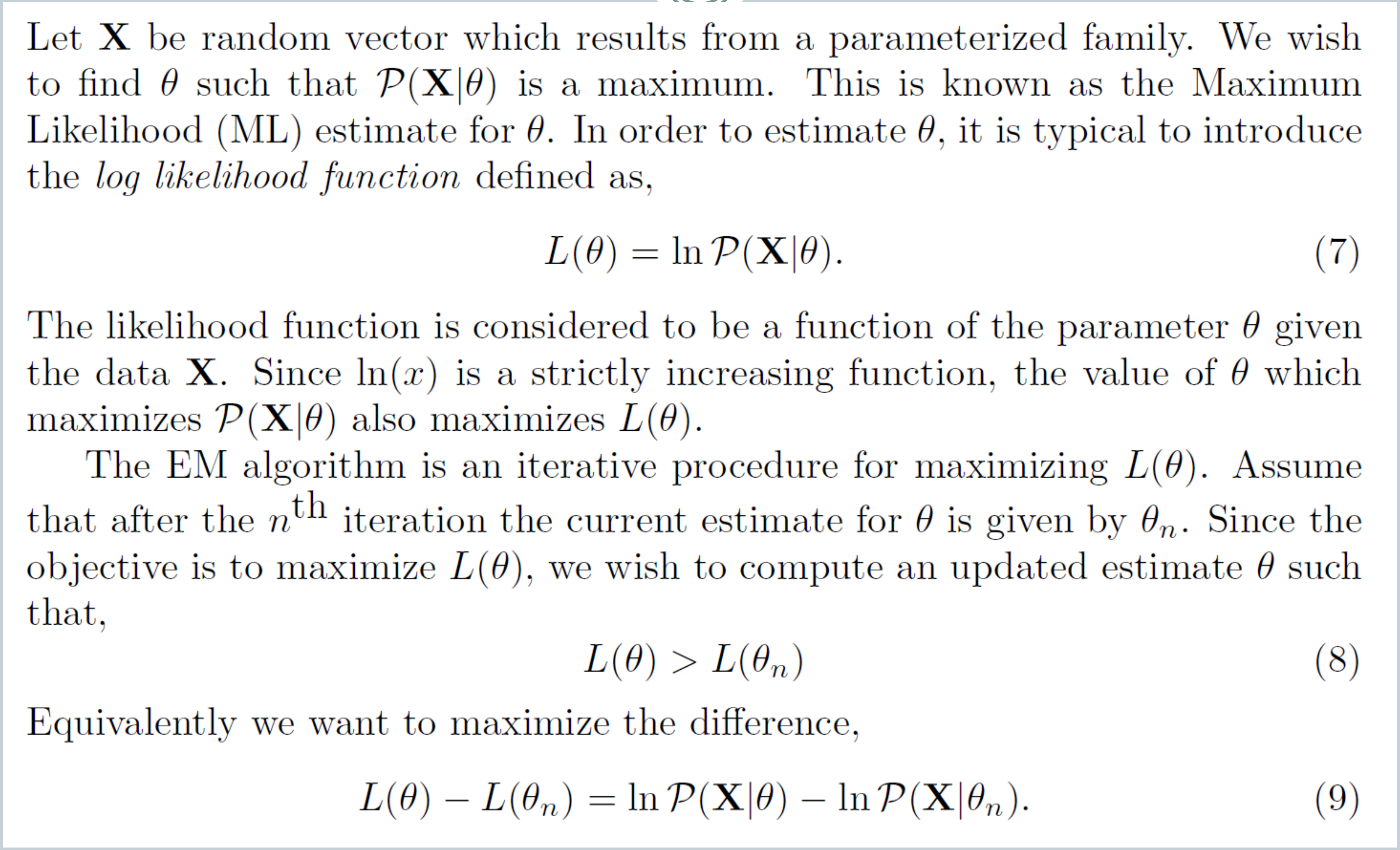
- Proof follows by Induction (Trivial for n = 1, n=2 → follows from convexity, demonstrate for n+1 assuming theorem true for n).

Since $\ln(x)$ is concave, we may apply Jensen's inequality to obtain the useful result,

$$\ln \sum_{i=1}^n \lambda_i x_i \geq \sum_{i=1}^n \lambda_i \ln(x_i). \tag{6}$$

This allows us to lower-bound a logarithm of a sum, a result that is used in the derivation of the EM algorithm.

EM Algorithm Overview



Let $\mathbf{X}$ be random vector which results from a parameterized family. We wish to find θ such that $\mathcal{P}(\mathbf{X}|\theta)$ is a maximum. This is known as the Maximum Likelihood (ML) estimate for θ. In order to estimate θ, it is typical to introduce the *log likelihood function* defined as,

$$L(\theta) = \ln \mathcal{P}(\mathbf{X}|\theta). \tag{7}$$

The likelihood function is considered to be a function of the parameter θ given the data $\mathbf{X}$. Since $\ln(x)$ is a strictly increasing function, the value of θ which maximizes $\mathcal{P}(\mathbf{X}|\theta)$ also maximizes $L(\theta)$.

The EM algorithm is an iterative procedure for maximizing $L(\theta)$. Assume that after the n^{th} iteration the current estimate for θ is given by θ_n. Since the objective is to maximize $L(\theta)$, we wish to compute an updated estimate θ such that,

$$L(\theta) > L(\theta_n) \tag{8}$$

Equivalently we want to maximize the difference,

$$L(\theta) - L(\theta_n) = \ln \mathcal{P}(\mathbf{X}|\theta) - \ln \mathcal{P}(\mathbf{X}|\theta_n). \tag{9}$$

EM Algorithm (Derivation)

$$L(\theta) - L(\theta_n) = \ln\left(\sum_{\mathbf{z}} \mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta)\right) - \ln \mathcal{P}(\mathbf{X}|\theta_n). \tag{11}$$

$$\begin{aligned}
L(\theta) - L(\theta_n) &= \ln\left(\sum_{\mathbf{z}} \mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta)\right) - \ln \mathcal{P}(\mathbf{X}|\theta_n) \\
&= \ln\left(\sum_{\mathbf{z}} \mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta) \cdot \frac{\mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n)}{\mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n)}\right) - \ln \mathcal{P}(\mathbf{X}|\theta_n) \\
&= \ln\left(\sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n)\frac{\mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta)}{\mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n)}\right) - \ln \mathcal{P}(\mathbf{X}|\theta_n) \\
&\geq \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n) \ln\left(\frac{\mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta)}{\mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n)}\right) - \ln \mathcal{P}(\mathbf{X}|\theta_n) \qquad (12) \\
&= \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n) \ln\left(\frac{\mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta)}{\mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n)\mathcal{P}(\mathbf{X}|\theta_n)}\right) \qquad (13) \\
&\triangleq \Delta(\theta|\theta_n). \qquad (14)
\end{aligned}$$

$$l(\theta|\theta_n) \triangleq L(\theta_n) + \Delta(\theta|\theta_n)$$

EMAlgorithm (Continued)

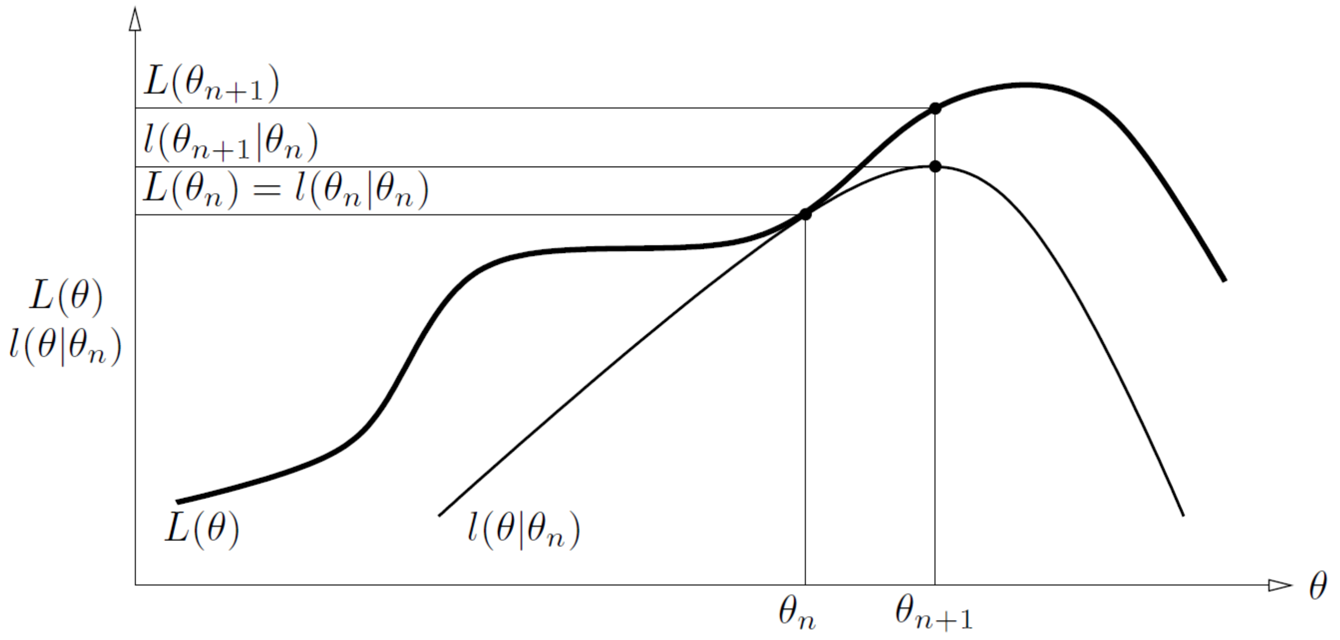


Figure 2: Graphical interpretation of a single iteration of the EM algorithm: The function $l(\theta|\theta_n)$ is bounded above by the likelihood function $L(\theta)$. The functions are equal at $\theta = \theta_n$. The EM algorithm chooses θ_{n+1} as the value of θ for which $l(\theta|\theta_n)$ is a maximum. Since $L(\theta) \geq l(\theta|\theta_n)$ increasing $l(\theta|\theta_n)$ ensures that the value of the likelihood function $L(\theta)$ is increased at each step.

EM Algorithm (Derivation)

$$
\begin{aligned}
\theta_{n+1} &= \arg\max_{\theta} \{l(\theta|\theta_n)\} \\
&= \arg\max_{\theta} \left\{ L(\theta_n) + \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n) \ln \frac{\mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta)}{\mathcal{P}(\mathbf{X}|\theta_n)\mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n)} \right\} \\
& \quad \text{Now drop terms which are constant w.r.t. } \theta \\
&= \arg\max_{\theta} \left\{ \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n) \ln \mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta) \right\} \\
&= \arg\max_{\theta} \left\{ \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n) \ln \frac{\mathcal{P}(\mathbf{X},\mathbf{z},\theta)}{\mathcal{P}(\mathbf{z},\theta)} \frac{\mathcal{P}(\mathbf{z},\theta)}{\mathcal{P}(\theta)} \right\} \\
&= \arg\max_{\theta} \left\{ \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n) \ln \mathcal{P}(\mathbf{X},\mathbf{z}|\theta) \right\} \\
&= \arg\max_{\theta} \left\{ \mathrm{E}_{\mathbf{Z}|\mathbf{X},\theta_n} \{ \ln \mathcal{P}(\mathbf{X},\mathbf{z}|\theta) \} \right\} \qquad (17)
\end{aligned}
$$

EM Algorithm - Summary

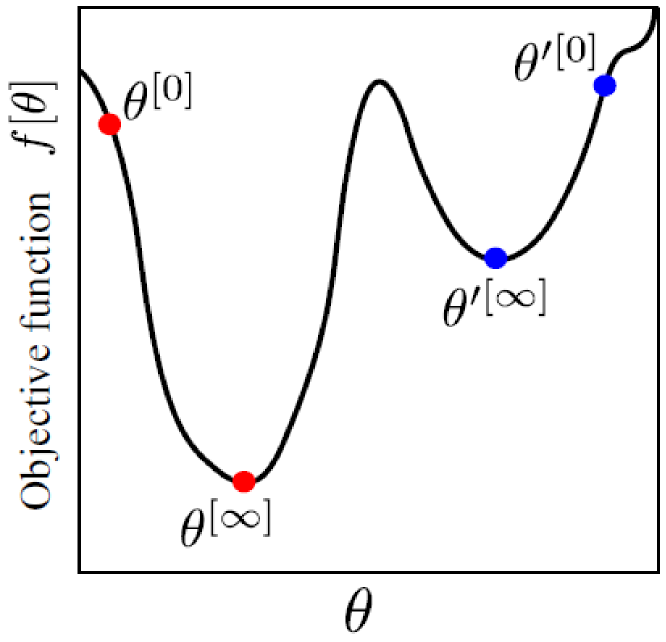
1. *E-step*: Determine the conditional expectation $\mathrm{E}_{\mathbf{Z}|\mathbf{X},\theta_n}\{\ln \mathcal{P}(\mathbf{X},\mathbf{z}|\theta)\}$
2. *M-step*: Maximize this expression with respect to θ.

Key Points:
- Iteratively converges to a local maximum
- Detailed Proof done later demonstrates convergence may not be only to Maxima (e.g. saddle points)
- Method is a unified principle for a number of estimation problems with Hidden variables and/or missing data.
- Several methods followed addressing computational speedups of algorithm

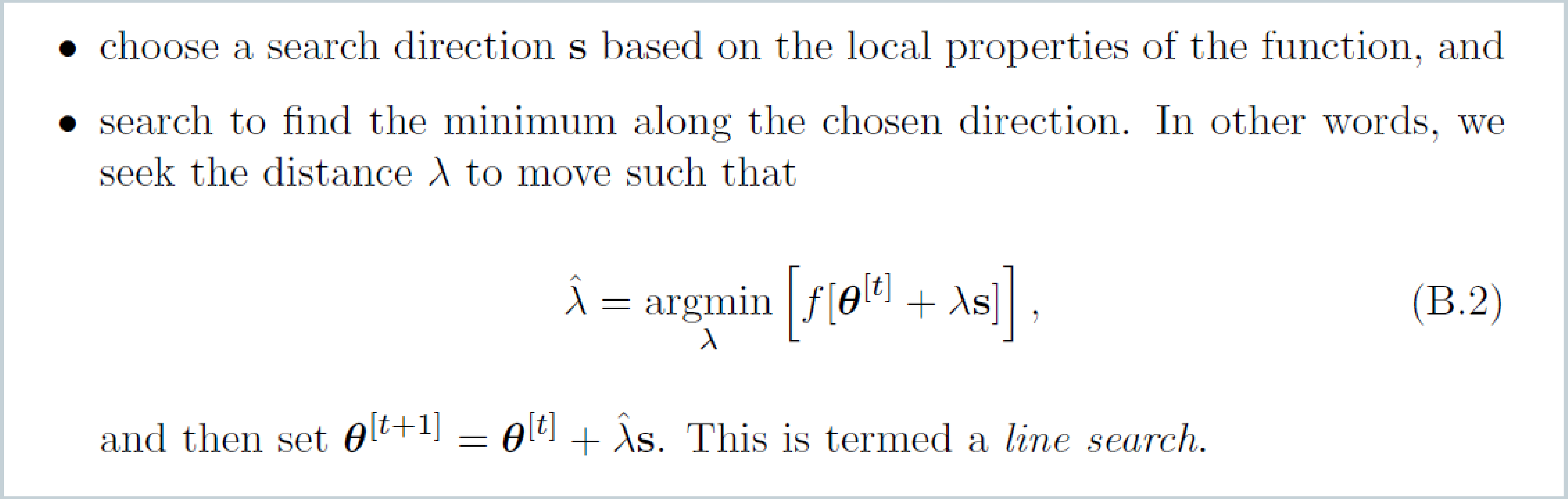
Basics of Optimization

- From Appendix B – Prince's Book (2012)

Figure B.1 Local minima. Optimization methods aim to find the minimum of the objective function $f[\theta]$ with respect to parameters θ. Roughly, they work by starting with an initial estimate $\theta^{[0]}$ and moving iteratively downhill until no more progress can be made (final position represented by $\theta^{[\infty]}$). Unfortunately, it is possible to terminate in a local minimum. For example, if we start at $\theta'^{[0]}$ and move downhill, we wind up in position $\theta'^{[\infty]}$.



Optimization: Line search



- choose a search direction $\mathbf{s}$ based on the local properties of the function, and
- search to find the minimum along the chosen direction. In other words, we seek the distance λ to move such that

$$\hat{\lambda} = \underset{\lambda}{\operatorname{argmin}} \left[f[\boldsymbol{\theta}^{[t]} + \lambda \mathbf{s}] \right], \tag{B.2}$$

and then set $\boldsymbol{\theta}^{[t+1]} = \boldsymbol{\theta}^{[t]} + \hat{\lambda}\mathbf{s}$. This is termed a *line search*.

Optimization – via Steepest Descent

$$\frac{\partial f}{\partial \theta_j} \approx \frac{f\left[\boldsymbol{\theta} + a\mathbf{e}_j\right] - f\left[\boldsymbol{\theta}\right]}{a}$$

$$\boldsymbol{\theta}^{[t+1]} = \boldsymbol{\theta}^{[t]} - \lambda \left.\frac{\partial f}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}^{[t]}}$$

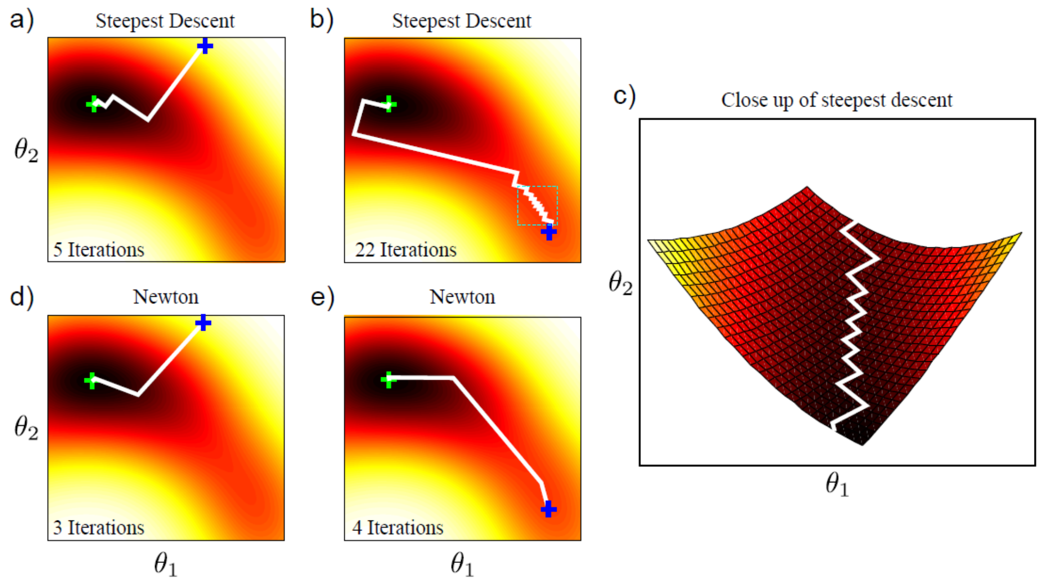


Figure B.3 Optimization on a two dimensional function (color represents height of function). We wish to find the parameters that minimize the function (green cross). Given an initial starting point $\boldsymbol{\theta}^0$ (blue cross), we choose a direction and then perform a local search to find the optimal point in that direction. a) One way to chose the direction is steepest descent: at each iteration, we head in the direction where the function changes the fastest. b) When we initialize from a different position, the steepest descent method takes many iterations to converge due to oscillatory behavior. c) Close-up of oscillatory region (see main text). d) Setting the direction using Newton's method results in faster convergence. e) Newton's method does not undergo oscillatory behavior when we initialize from the second position.

Optimization (Newton's method)

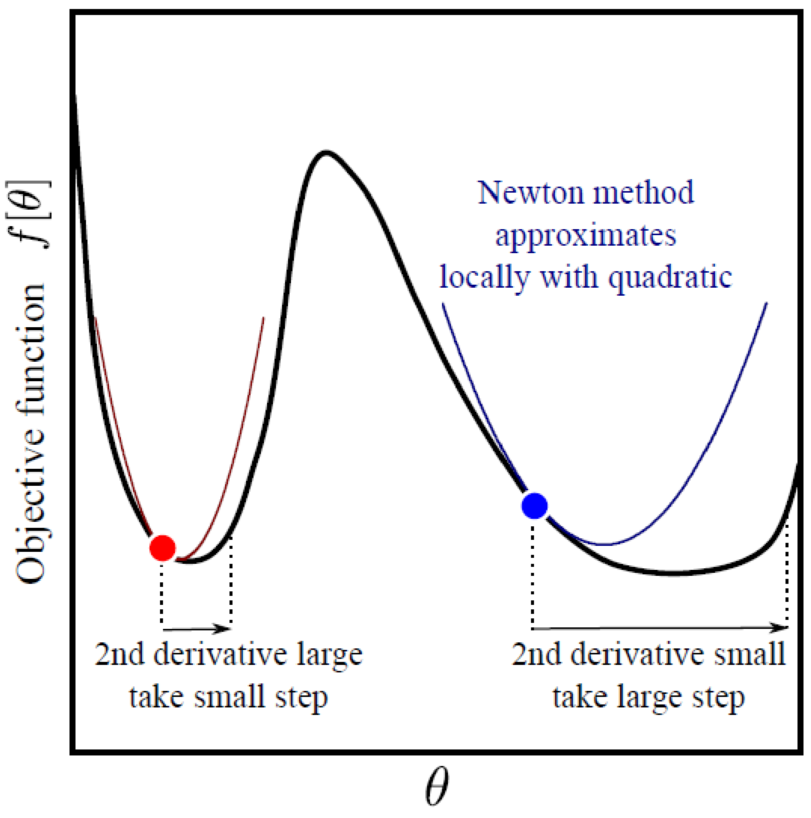
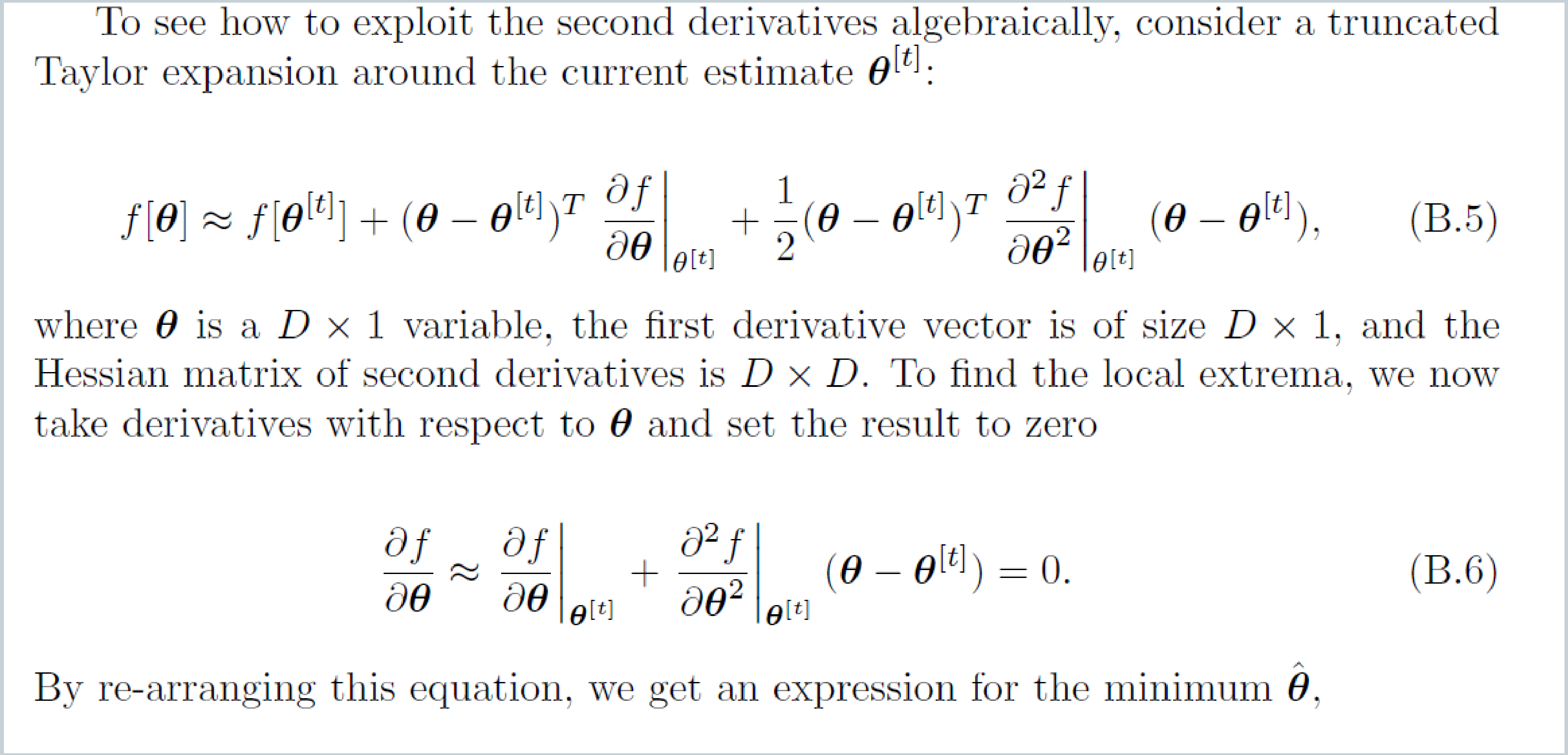


Figure B.4 Use of second derivatives. The gradient at the red and blue points is the same, but the magnitude of the second derivative is larger at the red point than the blue point: the gradient is changing faster at the red point than the blue point. The distance we move should be moderated by the second derivative: if the gradient is changing fast, then the minimum may be nearby and we should move a small distance. If it is changing slowly, then it is safe to move further. Newton's method takes into account the second derivative: it uses a Taylor expansion to create a quadratic approximation to the function and then moves toward the minimum.

Newton's method: Derivation



To see how to exploit the second derivatives algebraically, consider a truncated Taylor expansion around the current estimate $\boldsymbol{\theta}^{[t]}$:

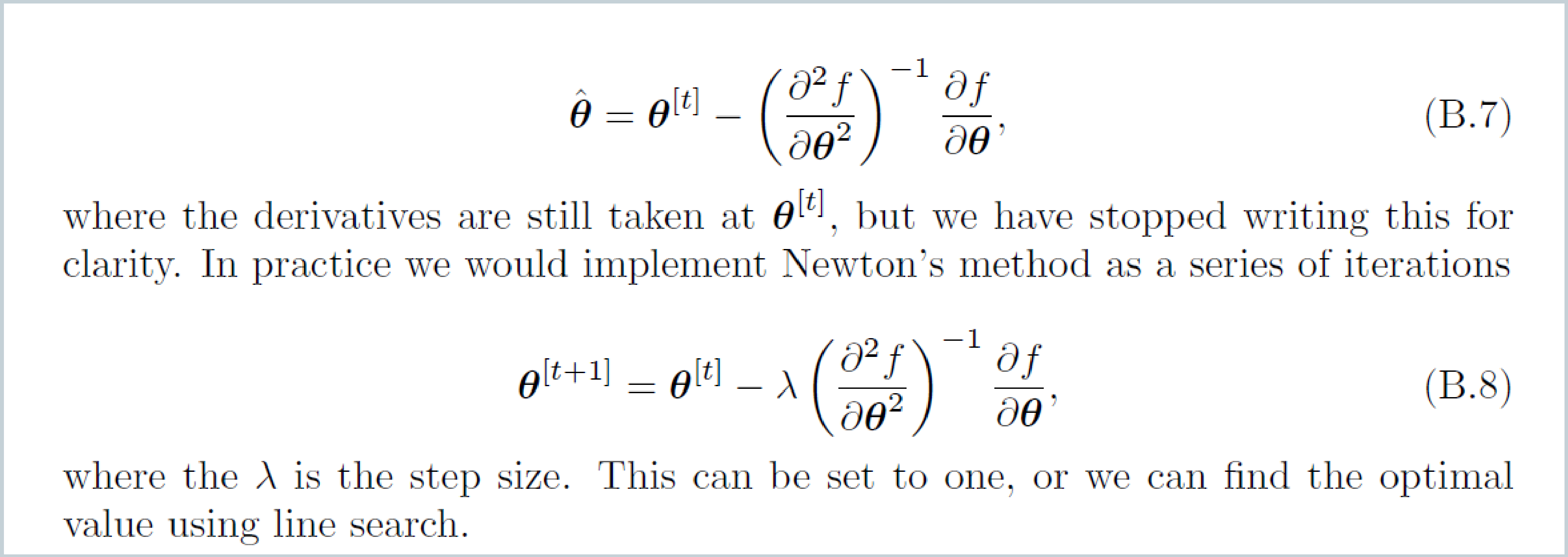
$$f[\boldsymbol{\theta}] \approx f[\boldsymbol{\theta}^{[t]}] + (\boldsymbol{\theta} - \boldsymbol{\theta}^{[t]})^T \left.\frac{\partial f}{\partial \boldsymbol{\theta}}\right|_{\theta^{[t]}} + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}^{[t]})^T \left.\frac{\partial^2 f}{\partial \boldsymbol{\theta}^2}\right|_{\theta^{[t]}} (\boldsymbol{\theta} - \boldsymbol{\theta}^{[t]}), \qquad \text{(B.5)}$$

where $\boldsymbol{\theta}$ is a $D \times 1$ variable, the first derivative vector is of size $D \times 1$, and the Hessian matrix of second derivatives is $D \times D$. To find the local extrema, we now take derivatives with respect to $\boldsymbol{\theta}$ and set the result to zero

$$\frac{\partial f}{\partial \boldsymbol{\theta}} \approx \left.\frac{\partial f}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}^{[t]}} + \left.\frac{\partial^2 f}{\partial \boldsymbol{\theta}^2}\right|_{\boldsymbol{\theta}^{[t]}} (\boldsymbol{\theta} - \boldsymbol{\theta}^{[t]}) = 0. \qquad \text{(B.6)}$$

By re-arranging this equation, we get an expression for the minimum $\hat{\boldsymbol{\theta}}$,

Newton's Method



$$\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}^{[t]} - \left(\frac{\partial^2 f}{\partial \boldsymbol{\theta}^2}\right)^{-1} \frac{\partial f}{\partial \boldsymbol{\theta}}, \tag{B.7}$$

where the derivatives are still taken at $\boldsymbol{\theta}^{[t]}$, but we have stopped writing this for clarity. In practice we would implement Newton's method as a series of iterations

$$\boldsymbol{\theta}^{[t+1]} = \boldsymbol{\theta}^{[t]} - \lambda \left(\frac{\partial^2 f}{\partial \boldsymbol{\theta}^2}\right)^{-1} \frac{\partial f}{\partial \boldsymbol{\theta}}, \tag{B.8}$$

where the λ is the step size. This can be set to one, or we can find the optimal value using line search.

Line Search Using Bracketing

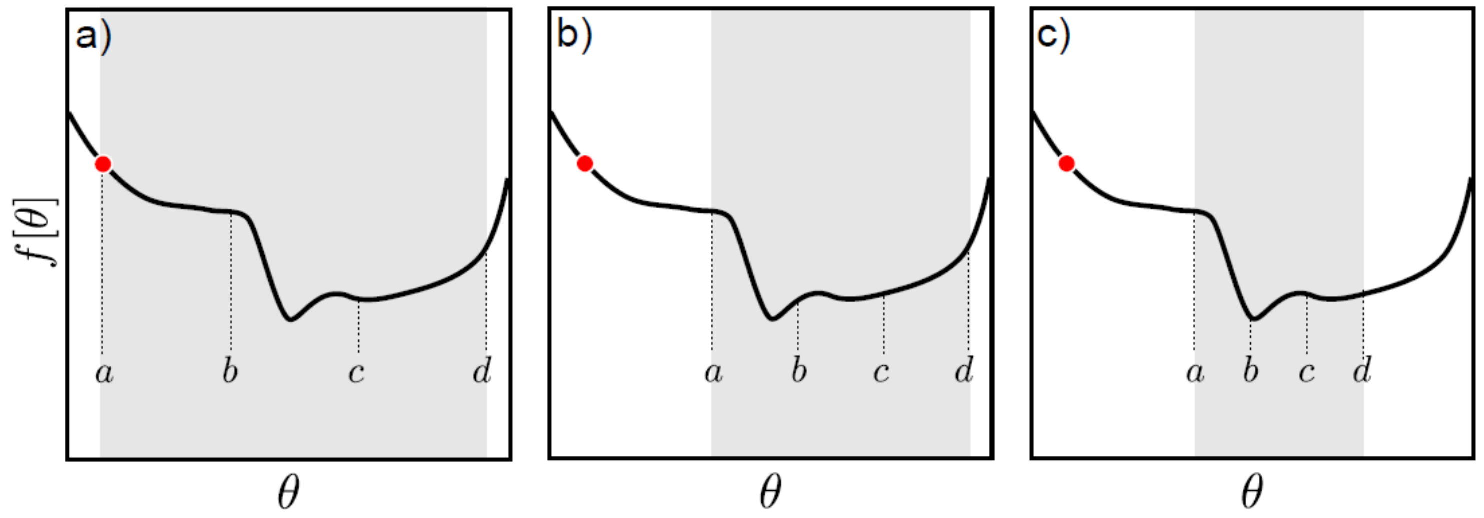


Figure B.5 Line search over region $[a, d]$ using bracketing approach. Gray region indicates current search region. a) We define two points b, c that are interior to the search region and evaluate the function at these points. Here $f[b] > f[c]$ so we eliminate the range $[a, b]$. b) We evaluate two points $[b, c]$ interior to the new range and compare their values. This time we find that $f[b] < f[c]$ so we eliminate the range $[c, d]$. c) We repeat this process until the minimum is closely bracketed.