UiO : Matematisk institutt

Det matematisk-naturvitenskapelige fakultet

STK-4051/9051 Computational Statistics Spring 2022
Chaper 2

Instructor: Odd Kolbjørnsen, oddkol@math.uio.no

Course originally made by
Geir Storvik

Optimization

- Focus maximum likelihood $\boxed{\max_{\theta} L(\theta|\boldsymbol{y})}$
  - But methods are general
- Different settings
  - Continuous vs discrete
  - One vs multi-dimensional
  - Unconstrained vs constrained
    - Common: $y \sim \mathrm{N}(\mu, \sigma^2)$, $\mu -$ unconstrained, $\sigma^2 > 0$
- Can we compute the derivative analytically?

One dimensional ML, Newton's method

So common that people usually do not mention that this is what they use

- Common to consider log likelihood:
  - $\underset{\theta}{\operatorname{argmax}} L(\theta|\boldsymbol{y}) = \underset{\theta}{\operatorname{argmax}} \underbrace{\log(L(\theta|\boldsymbol{y}))}_{\ell(\theta|\boldsymbol{y}) \text{ or just } \ell(\theta)}$

$$\boxed{\begin{aligned}\ell(\theta) &\approx \ell(\theta^*) + (\theta-\theta^*)\ell'(\theta^*) + \tfrac{1}{2}(\theta-\theta^*)^2\ell''(\theta^*) \\ &\ \ell(\theta^*) + (\theta-\theta^*)s(\theta^*) \quad - \tfrac{1}{2}(\theta-\theta^*)^2 J(\theta^*)\end{aligned}}$$

Taylor expansion around θ^*

Score function: $s(\theta) = \ell'(\theta)$
Observed information: $J(\theta) = -\ell''(\theta)$

  - Solving the maximum of the approximation:

$$\theta = \theta^* + \frac{s(\theta^*)}{J(\theta^*)} = \theta^* - \frac{\ell'(\theta^*)}{\ell''(\theta^*)}$$

Example $\mathbb{R}$

$$L(\mu) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\{-\frac{1}{2}\left(\frac{x_i - \mu}{\sigma}\right)^2\}$$

σ known

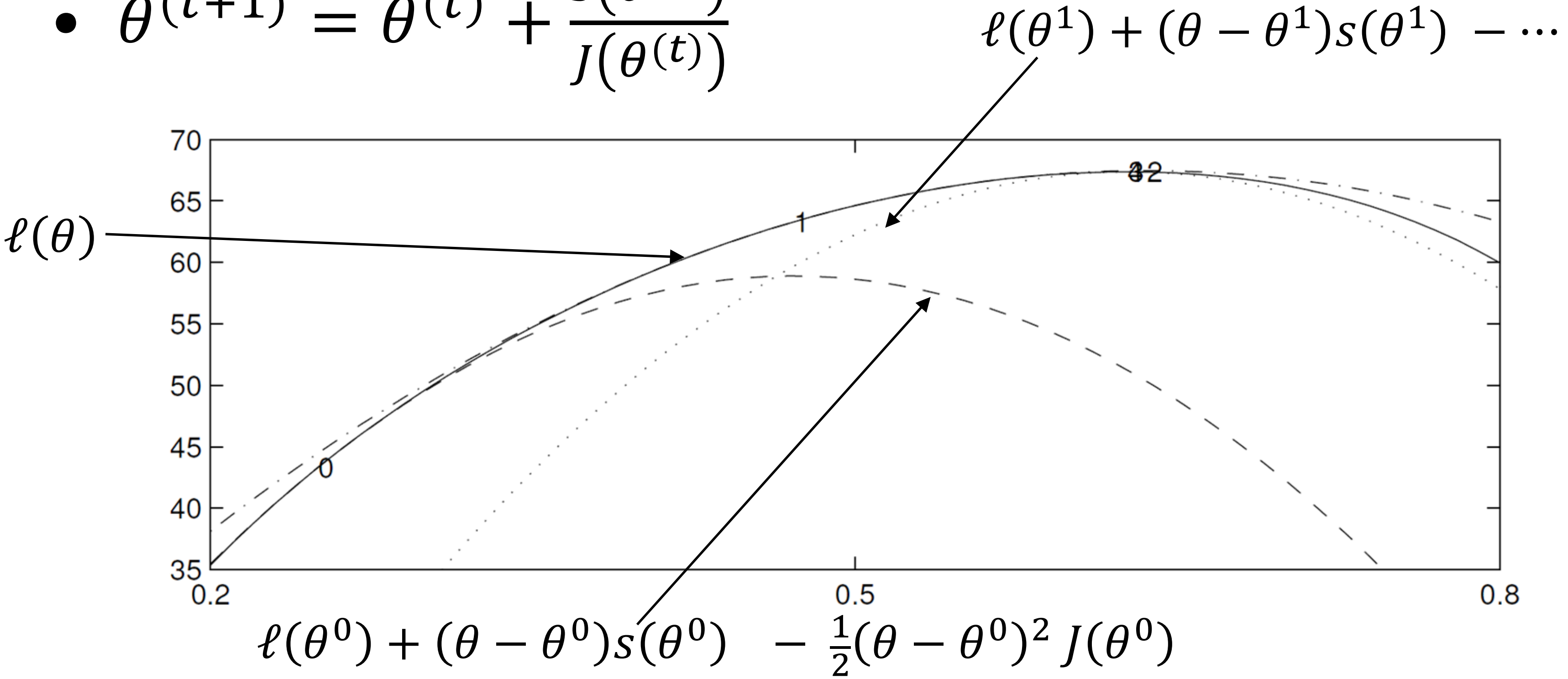
$$l(\mu) = \sum_{i=1}^{n} -\frac{1}{2}\log 2\pi - \frac{1}{2}\log\sigma^2 - \frac{1}{2}\left(\frac{x_i - \mu}{\sigma}\right)^2$$

$$s(\mu) = l'(\mu) = \sum_{i=1}^{n} -0 - 0 - \frac{x_i - \mu}{\sigma^2} = \frac{1}{\sigma^2}\sum_{i=1}^{n}(x_i - \mu)$$

$$J(\mu) = -l''(\mu) = -s'(\mu) = \frac{-1}{\sigma^2}\sum_{i=1}^{n} -1 = \frac{n}{\sigma^2}$$

Iterations in Newton's method

- $\theta^{(t+1)} = \theta^{(t)} + \frac{s(\theta^{(t)})}{J(\theta^{(t)})}$



- $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} + \boldsymbol{J}(\boldsymbol{\theta}^{(t)})^{-1}\boldsymbol{s}(\boldsymbol{\theta}^{(t)})$

Multidimensional extension

- Common to consider log likelihood:

$$\underset{\theta}{\operatorname{argmax}}\, \ell(\boldsymbol{\theta}) = \underset{\theta}{\operatorname{argmax}}\, L(\boldsymbol{\theta}|\boldsymbol{y}) = \underset{\theta}{\operatorname{argmax}}\, \log(L(\boldsymbol{\theta}|\boldsymbol{y}))$$

$$\boxed{\begin{aligned}\ell(\boldsymbol{\theta}) \approx \ell(\boldsymbol{\theta}^*) + (\boldsymbol{\theta}-\boldsymbol{\theta}^*)\ell'^{(\boldsymbol{\theta}^*)} + \tfrac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\theta}^*)^T \boldsymbol{H}(\theta^*)(\boldsymbol{\theta}-\boldsymbol{\theta}^*)\\ \ell(\boldsymbol{\theta}^*) + (\boldsymbol{\theta}-\boldsymbol{\theta}^*)^T \boldsymbol{s}(\boldsymbol{\theta}^*) \quad - \tfrac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\theta}^*)^T \boldsymbol{J}(\boldsymbol{\theta}^*)(\boldsymbol{\theta}-\boldsymbol{\theta}^*)\end{aligned}}$$

Score function: $\boldsymbol{s}(\boldsymbol{\theta}) = \nabla\ell(\boldsymbol{\theta}) = \frac{\partial}{\partial\boldsymbol{\theta}}\ell(\boldsymbol{\theta})$ p - vector

Observed information: $\boldsymbol{J}(\boldsymbol{\theta}) = -\nabla^2\ell(\boldsymbol{\theta}) = \frac{\partial^2}{\partial\boldsymbol{\theta}^2}\ell(\boldsymbol{\theta})$ $p\times p$ - matrix

  - Solving the maximum of the approximation:

$$\boldsymbol{\theta} = \boldsymbol{\theta}^* + \boldsymbol{J}(\boldsymbol{\theta}^*)^{-1}\boldsymbol{s}(\boldsymbol{\theta}^*) = \boldsymbol{\theta}^* - \boldsymbol{H}(\boldsymbol{\theta}^*)^{-1}\nabla\ell(\boldsymbol{\theta}^*)$$

Example $\mathbb{R}^{\boldsymbol{p}}$

$$L(\mu) = \prod_{i=1}^{n} \frac{1}{\sqrt{(2\pi)^p |\Sigma|}} \exp\{-\frac{1}{2}(x_i - \mu)^T \Sigma^{-1} (x_i - \mu)\}$$ Σ known

$$l(\mu) = \sum_{i=1}^{n} -\frac{p}{2}\log 2\pi - \frac{1}{2}\log|\Sigma| - \frac{1}{2}(x_i - \mu)^T \Sigma^{-1}(x_i - \mu)$$

$$s(\mu) = \nabla\, l(\mu) = \sum_{i=1}^{n} \Sigma^{-1}(x_i - \mu) = \Sigma^{-1}\sum_{i=1}^{n}(x_i - \mu)$$

$$J(\mu) = -\nabla^2 l(\mu) = -\sum_{i=1}^{n} -\Sigma^{-1} = n\Sigma^{-1} = \left(\frac{1}{n}\Sigma\right)^{-1}$$

Stopping criteria

- Absolute convergence
  - $\left|x^{(t+1)} - x^{(t)}\right| < \epsilon$ or $\left\|\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^{(t)}\right\| < \epsilon$
  - If x is large this might iterate too long
- Relative convergence
  - $\frac{\left|x^{(t+1)} - x^{(t)}\right|}{\left|x^{(t)}\right|} < \epsilon$ or $\frac{\left\|\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^{(t)}\right\|}{\left\|\boldsymbol{x}^{(t)}\right\|} < \epsilon$
  - Unstable if $\left|x^{(t)}\right|$ is small
    - usually not a problem in a multivariate setting
- After N iterations (use as additional criteria)

- If not converged do not trust result
- There is in general no theorem that tells you in advance how many iterations you need
- Try different methods and starting points

Fisher scoring and ascent algorithms

- Newton's method require $\ell''(\theta) < 0$ or $J(\theta) > 0$
  Multivariate: $\mathbf{J}(\boldsymbol{\theta})$ need to be positive definite
- Note: $\mathbf{J}(\boldsymbol{\theta})$ is stochastic (depend on data)
- $\mathbf{I}(\boldsymbol{\theta}) = E[\mathbf{J}(\boldsymbol{\theta})]$ is the expected information matrix
- Can show: $\mathbf{I}(\boldsymbol{\theta}) = \text{Var}[\mathbf{s}(\boldsymbol{\theta})]$, always positive (semi-)definite
- Fisher scoring algorithm:

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} + [\mathbf{I}(\boldsymbol{\theta}^{(t)})]^{-1}\mathbf{s}(\boldsymbol{\theta}^{(t)})$$

- Will typically be more stable than Newton's method
- Can be both computationally and analytically easier
- Generalized linear models (STK3100/4100): $\mathbf{I}(\boldsymbol{\theta}) = \mathbf{J}(\boldsymbol{\theta})$.
- Alternative: Ascent algorithms

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} + \alpha^{(t)}\mathbf{s}(\boldsymbol{\theta}^{(t)})$$

  By choosing $\alpha^{(t)}$ small enough, decrease in likelihood value can be avoided.

Example: $I(\mu) = E\big(J(\mu)\big) = \mathrm{Var}(\ s(\mu\ )\ )$

$$s(\mu) = \Sigma^{-1}\sum_{i=1}^{n}(x_i - \mu) \qquad\qquad J(\mu) = \left(\frac{1}{n}\Sigma\right)^{-1}$$

1 $$E\big(J(\mu)\big) = E\left(\left(\frac{1}{n}\Sigma\right)^{-1}\right) = \left(\frac{1}{n}\Sigma\right)^{-1}$$

Independent observations

2 $$\mathrm{Var}(\ s(\mu\ )\ ) = \mathrm{Var}\left(\Sigma^{-1}\sum_{i=1}^{n}(x_i - \mu)\right) = \sum_{i=1}^{n}\Sigma^{-1}\mathrm{Var}(x_i - \mu)\Sigma^{-1}$$

$$= \sum_{i=1}^{n}\Sigma^{-1}\Sigma\,\Sigma^{-1} = n\Sigma^{-1}$$

$$= \left(\frac{1}{n}\Sigma\right)^{-1}$$

Gauss-Newton method

- Assume we have a model

$$Y_i = f(\boldsymbol{z}_i; \theta) + \varepsilon_i$$

and want to maximize $g(\theta) = -\sum_{i=1}^n (y_i - f(\boldsymbol{z}_i; \theta))^2$
- Newton's method: Approximate $g(\theta)$
- Gauss-Newton: Approximate $f(\boldsymbol{z}_i; \theta)$:

$$\tilde{f}(\boldsymbol{z}_i; \theta; \theta^{(t)}) \approx f(\boldsymbol{z}_i; \theta^{(t)}) + (\theta - \theta^{(t)}) \nabla_\theta f(\boldsymbol{z}_i, \theta^{(t)})$$

$$\boldsymbol{f}(\boldsymbol{z}; \boldsymbol{\theta}) = \begin{bmatrix} f(\boldsymbol{z}_1; \boldsymbol{\theta}) \\ \vdots \\ f(\boldsymbol{z}_n; \boldsymbol{\theta}) \end{bmatrix}$$

- Gauss-Newton step: Maximize

$$\tilde{g}(\theta) = -\sum_{i=1}^n (y_i - \tilde{f}(\boldsymbol{z}_i; \theta; \theta^{(t)}))^2$$

$$= -\sum_{i=1}^n [y_i - f(\boldsymbol{z}_i; \theta^{(t)}) + (\theta - \theta^{(t)})^T \nabla_\theta f(\boldsymbol{z}_i, \theta^{(t)})]^2$$

- Solution

$$\theta^{(t+1)} = \theta^{(t)} + [(\boldsymbol{A}^{(t)})^T \boldsymbol{A}^{(t)}]^{-1} (\boldsymbol{A}^{(t)})^T [\boldsymbol{y} - \boldsymbol{f}(\boldsymbol{z}; \theta^{(t)})]$$

$$A^{(t)} = \begin{bmatrix} \nabla_\theta f(\boldsymbol{z}_1, \theta) \\ \vdots \\ \nabla_\theta f(\boldsymbol{z}_n, \theta) \end{bmatrix}$$

$$n \times p$$

- Advantage: Only need first derivatives!

Other optimization methods

- Newton-type methods require derivatives
- Secant methods: Replace $J(\theta) = -\ell''(\theta)$ by finite difference approximation
- Fixed-point methods ($\max_x g(x)$)
  - Find function $G(x)$ such that $G(x) = x \Leftrightarrow g'(x) = 0$
  - Use updating scheme $x^{(t+1)} = G(x^{(t)})$
  - Obvious choice: $G(x) = \alpha g'(x) + x \Rightarrow x^{(t+1)} = x^{(t)} + \alpha g'(x^{(t)})$
  - Requirements for convergence:
    1. $x \in [a, b] \Rightarrow G(x) \in [a, b]$
    2. $|G(x_1) - G(x_2)| \leq \lambda |x_1 - x_2|$ for all $x_1, x_2 \in [a, b]$ for some $\lambda \in (0, 1)$.
  - Newton-type methods can be seen as special cases of fixed point methods

Example fixed point

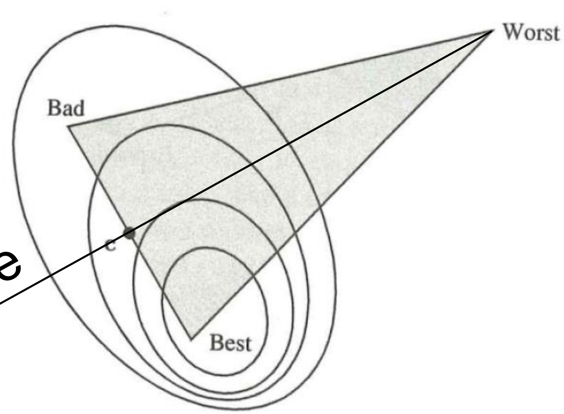
- Maximize $g(x) = x\log(x) - x + 0.5x^2$, $g'(x) = \log(x) + x$
- Possible choices of G:

$$
\begin{aligned}
G_1(x) =& g'(x) + x = \log(x) + 2x\\
G_2(x) =& -\log(x)\\
G_3(x) =& \exp(-x)\\
G_4(x) =& (x + \exp(-x))/2
\end{aligned}
$$

- `fixed_point_example.R`

Nelder - Mead

- Starts with $p+1$ distinct points $\mathbf{x}_1, \ldots, \mathbf{x}_{p+1}$
- Points ranked through $g(\mathbf{x}_1), \ldots, g(\mathbf{x}_{p+1})$
- $\mathbf{x}_{best}$ and $\mathbf{x}_{worst}$ best and worst points
- Calculate $\mathbf{c} = \frac{1}{p}\left[\sum_{i=1}^{p+1} \mathbf{x}_i - \mathbf{x}_{worst}\right]$
- Find new value $\mathbf{x}_r = \mathbf{c} + \alpha(\mathbf{c} - \mathbf{x}_{worst})$, replace with $\mathbf{x}_{worst}$
- Require no derivatives

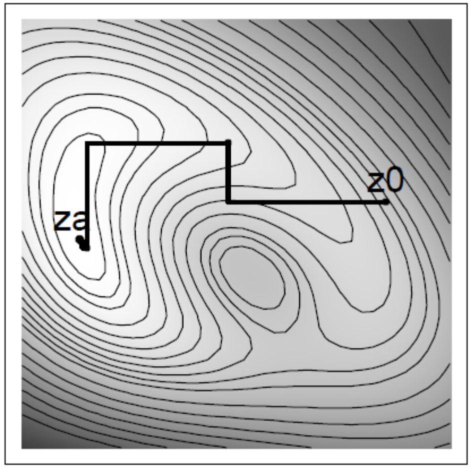


$\mathbf{x}_r$ somewhere along this line

If you do not find improvement shrink all points towards best

Gauss- Seidel

- Aim: maximize $g(\boldsymbol{\theta})$, $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_p)$
- Procedure: For $j = 1, \ldots, p$,
  - Maximize $g(\boldsymbol{\theta})$ with respect to θ_j keeping the other θ_k's fixed
- Reduce the multivariate problem to many univariate problems



Pick your favorite 1D optimization

BFGS-algorithm
Broyden–Fletcher–Goldfarb–Shanno

- Quasi-Newton (variable metric) method (argmax $g(\boldsymbol{x})$)

$$x_{k+1} = x_k - \alpha_k M_k^{-1} \nabla g(x_k)$$

- M_k is an approximation to the Hessian
- α_k obtained by line-search
- Do a rank 1 update of M_k to M_{k+1} using quantities computed during iterations (see book)

- Note: even though x_k converges,

  M_k may not converge to Hessian in optimum

`optim` in R

```
optim(par, fn, gr = NULL, ...,
      method = c("Nelder-Mead", "BFGS", "CG", "L-BFGS-B", "SANN",
                 "Brent"),
      lower = -Inf, upper = Inf,
      control = list(), hessian = FALSE)
```

- Nelder-Mead: Default. Robust, but can be slow.
- BFGS:
  - $\mathbf{x}^{t+1)} = \mathbf{x}^{(t)} - (\mathbf{M}^{(t)})^{-1}\mathbf{g}'(\mathbf{x}^{(t)})$, $\mathbf{M}^{(t)}$ approximation of $\mathbf{g}''(\mathbf{x}^{(t)})$
  - $\mathbf{M}^{(t)}$ updated by a low-rank operation
- CG (Conjugate gradient): Optimize along gradient direction (iteratively).
- L-BFGS-B: Modification of BFGS to allow for constraints
- SANN: Simulated annealing (to be covered later)
- Brent: One-dimensional method

Recursive approaches

- Optimisation of $g(\mathbf{x})$
- Iterative approach: $\mathbf{x}^{(t+1)} = T(\mathbf{x}^{(t)})$
- Stochastic iterative approach: $\mathbf{x}^{(t+1)} = T(\mathbf{x}^{(t)}, \boldsymbol{\varepsilon}^{(t+1)})$
  - $\mathbf{x}^{(t+1)}$ only depend on $\mathbf{x}^{(t)}$ and not the previous values
  - This is called a Markov process
  - If $\mathbf{x}^{(t)}$ is discrete: Markov chain (STK2030)

Brief review of Markov chains

- Consider a stochastic sequence $X^{(t)}, t = 0, 1, \ldots$
- $X^{(t)} \in S$, a finite (or countable) set
- In general:

$$
\begin{aligned}
&P(X^{(0)}, X^{(1)}, X^{(2)}, \ldots, X^{(n)}) \\
&\qquad = P(X^{(0)})P(X^{(1)}, |X^{(0)})P(X^{(2)}|X^{(0)}, X^{(1)}) \cdots P(X^{(n)}|X^{(0)}, X^{(1)}, \ldots X^{(n-1)})
\end{aligned}
$$

- Markov assumption:

$$P(X^{(t)}|X^{(0)}, X^{(1)}, \ldots X^{(t-1)}) = P(X^{(t)}|X^{(t-1)})$$

- Denote $P_{ij}^t = P(X^{(t)} = j|X^{(t-1)} = i)$, defines a transition matrix
- Time-homogeneous Markov chain: $P_{ij}^t = P_{ij}^1$ for all t
- A Markov chain is irreducible if any state $j \in S$ can be reached from any state $i \in S$ in a finite number of transitions.

Next time:

- Iterative re-weighted least square
- ADMM
  – Lasso example
- Combinatorial optimization (chapter 3)

- Exercise