Chapter 1

Dissecting the Data2LD Package

First some notation, for the sake of brevity and legibility we will let $\mathbf{g}_n$ denote the gradient of the objective function evaluated at the nth iterate $\mathbf{x}_n$.

$$\mathbf{g_n} = \nabla f(x_n)$$

Likewise, we will let $\mathbf{H}_n$ denote the Hessian matrix at the nth iteration

$$\mathbf{H}_n = \nabla\nabla^\top f(x_n)$$

1.1 Line Search Methods

A common methodology to minimise $f(x)$ is as follows: given a point x_0, an approximate function $\tilde{f}(x|x_0)$ is constructed, the point that minimises $\tilde{f}(x|x_0)$ becomes the new candidate point, the process is then repeated until convergence.

The Newton Raphson Method employed for Maximum Likelihood Estimation is a well-known example of this approach. The log-likelihood $\ell(\theta)$ is approximated by a quadratic in the neighbourhood of some point θ_0 :

$$\ell(\theta) \approx \ell(\theta_0) + S(\theta_0)(\theta - \theta_0) + \frac{1}{2}I(\theta_0)(\theta - \theta_0)^2$$

It can be shown easily that this approximation is minimised at $\theta_1 = \theta_0 - I^{-1}(\theta_0)S(\theta_0)$

There is a serious weakness with this approach, there is no guarantee that the model will be valid over a sufficiently large radius. It could be the case that the next point advocated by the model could be so far away that the model's opinion on what it regards as the best possible point is completely wrong.

For a one dimesional problem, one can often plot the function at hand if one suspects irregular behaviour. For a function with many input variables the problem is more difficult. Usually, one must conduct post hoc tests to check if the approximation is acting as a good guide as to how the objective function is actually behaving.

The situation needs to be approached more systemically.

The Data2LD package employs Line Search Methods, which work by picking a direction, and then deciding how far in that direction to go. For comparison, Trust-Region Methods set a maximum allowable distance, and then decide on the direction.

For a smooth function $f(x)$ and a search direction p, the following holds:

$$f(x_0 + p) \approx f(x_0) + \nabla f(x_0)'p$$

Two important results follow from this.

The first is that a point x_0 can only be a local minimum or maximum of $f(x)$ if the gradient is zero at that point, otherwise one could find a point nearby where $f(x)$ were higher or lower than $f(x_0)$ by taking a small step in the appropriate direction.

The second, is that if one is at some point x_0 and wishes to reduce $f(x)$ from its current value, one should ensure that the direction p satisfies $\nabla f(x_0)'p < 0$.

The naive choice would be to set $p = -\nabla f(x_0)$, this is known as the gradient descent method.

Gradient descent can be very slow. In particular, the Steepest Descent Method, which produces the point exactly minimise $f(x)$ along the line through x_0 in the direction of $\nabla f(x_0)$, produces steps that are perpendicular to each other, so that the path take by the algorithm forms a zig-zag pattern.

1.1.1 Chord Methods

Chord Methods attempt to approximate the Hessian matrix by using a constant matrix $\mathbf{Q}$. The next iterate is defined by:

$$\mathbf{x}_{n+1} = \mathbf{x}_k + \mathbf{Q}\mathbf{g}_k$$

It can be shown that the Chord Method converges linearly[13], a very informal sketch will be provided here.[1] Let $\mathbf{g}(\mathbf{x})$ denote the mapping $\mathbf{g}(\mathbf{x}) = \mathbf{x} + \mathbf{Q}\nabla f(\mathbf{x})$. Note that $\mathbf{x}_{n+1} = \mathbf{g}(\mathbf{x}_n)$. Suppose the sequence $\mathbf{x}_n$ converges, to determine how quickly the converge occurs, perform a Taylor expansion about the limit $\mathbf{x}^*$:

$$\begin{aligned}\mathbf{x}_{n+1} - \mathbf{x}^* &= \mathbf{g}'(\mathbf{x}^*)(\mathbf{x}_n - \mathbf{x}^*) \\ &= (\mathbf{I} + \mathbf{Q}\mathbf{H})(\mathbf{x}_n - \mathbf{x}^*) \\ &= \mathbf{K}(\mathbf{x}_n - \mathbf{x}^*)\end{aligned}$$

For brevity, we let $\mathbf{H}$ denote the Hessian of f at $\mathbf{x}^*$. The convergence of the Chord Method around $\mathbf{x}^*$ is governed by the matrix $\mathbf{K} = \mathbf{I} + \mathbf{Q}\mathbf{H}$. If $\mathbf{K} = 0$, then $\mathbf{Q} = \mathbf{H}^{-1}$ and the method converges superlinearly. It is very rarely the case that the Hessian at the limit point is available though. Usually the matrix $\mathbf{Q}$ is only an approximation to $\mathbf{H}^{-1}$. The better the approximation, the smaller the matrix $\mathbf{K}$ will be, and the faster the rate of convergence.

Using the Chord Method For Maximum Likelihood Estimation With The Poisson Distribution

For the problem of maximum likelihood estimation in one variable, the mapping associated with the Chord Method is given by by $g(\theta) = \theta - mS(\theta_n)$. It is readily apparent that $g'(\theta) = 1 + mI(\theta)$ The Chord Method will converge if to the MLE if $|1 + mI(\hat{\theta})| < 1|$. The closer m is to $I(\hat{\theta})$, the faster the rate of convergence.

Consider the Poisson Distribution. The MLE[2] is given by $\hat{\lambda} = \bar{x}$. The usual Newton-Raphson iteration is given by:

$$\bar{\lambda}_{n+1} = \bar{\lambda}_n - \frac{S(\bar{\lambda}_n)}{I(\bar{\lambda}_n)},$$

A Chord Method is of the form:

$$\tilde{\lambda}_{n+1} = \tilde{\lambda}_n + mS(\tilde{\lambda}_n).$$

To find a value of m, rely on the fact that the variance of the MLE is asympotically equal to $1/I(\theta)$. Since the MLE is the sample mean, the Fisher Information can thus be approximated by estimating

[1] This is a straightforward generalisation of results presented in [12] to the multivariate case

[2] The Score function is given by $S(\lambda) = n(\frac{\bar{x}}{\lambda} - 1)$ and the information is given by $I(\lambda) = -\frac{n\bar{x}}{\lambda^2}$.

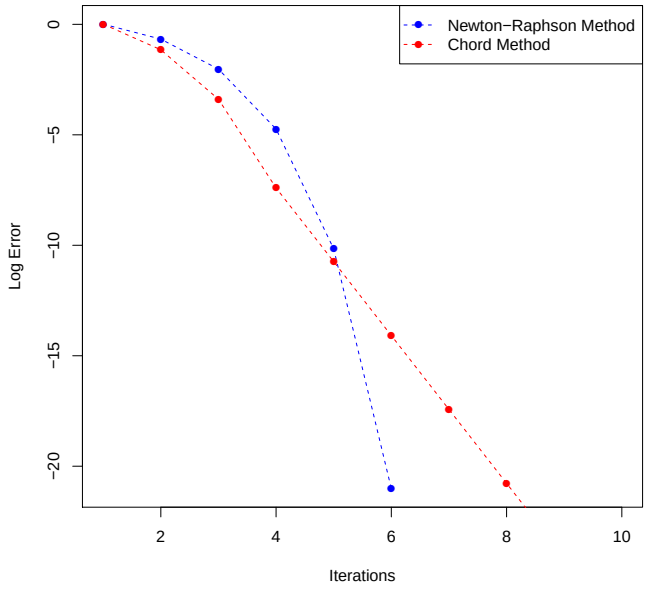

Figure 1.1: Plot comparing the convergence of the Chord Method and the Newton Raphson Method for finding the MLE of a Poisson Distribution

the variance of the sample mean, which is straightforwardly yields $m = \hat{\sigma}^2/n$,where $\hat{\sigma}^2$ is the sample variance.

Figure 1.1 compares the convergence of the Chord Method and the standard Newton-Raphson Method. The linear convergence of the Chord Method is readily apparent on the log plot, and the Chord Method performs reasonably well compared to the Newton-Raphson Method.

The asymptotic method used to justify the choice of m suggests that as the sample size gets bigger, the sample variance and observed Fisher Information might get closer and closer together, so that the Chord Method should converge more quickly. A simulation was conducted and the results are presented in Figure 1.2. Each curve plotted is the avearge over many error curves generated by generating samples from a poisson distribution and running the Chord Method on them. It can be seen that the method tends to converge more quickly as the sample size increases.

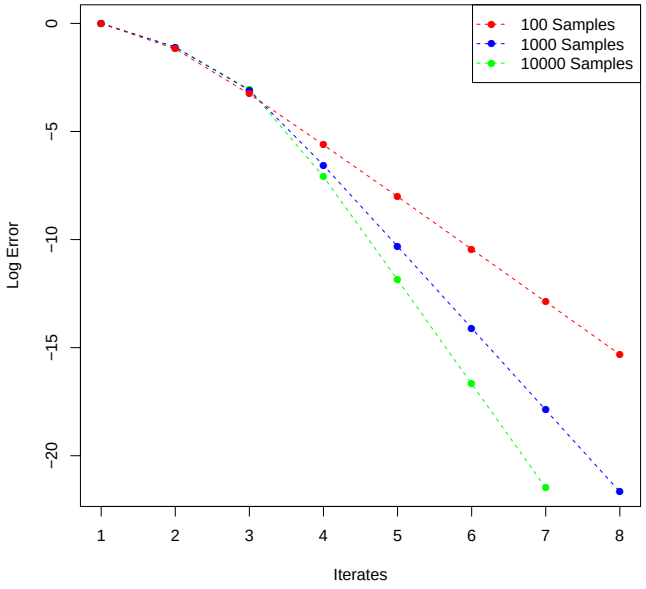

Figure 1.2: The Chord Method converges more quickly as the sample size increases.

1.1.2 Higher Order Methods and Quasi-Newton Methods

Instead of using a fixed matrix on each iteration as with the Chord Method, more advanced methods allow the matrix $\mathbf{Q}$ to vary on each iteration:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \mathbf{Q}_n \mathbf{g}_n$$

The choice $\mathbf{Q}_n = \mathbf{H}_n^{-1}$ corresponds to Newton's Method. The discussion in Section 1.1.1 suggests that to ensure faster than linear convergence, it is necessary to ensure that $\mathbf{I} + \mathbf{Q}_n \mathbf{H} \to \mathbf{0}$ as n goes to infinity.[3] Not every method in use has this property. Consider for example Fisher's Method of Scoring, which uses the expected information matrix $\mathcal{I}(\theta)$ to approximate the observed information $\mathbf{I}(\theta)$. It is not the case that $\mathcal{I}(\hat{\theta}) = \mathbf{I}(\hat{\theta})$, so one should not expect $\mathbf{I} + \mathbf{Q}_n \mathbf{H} \to \mathbf{0}$ as the algorithm converges to the MLE $\hat{\theta}$. As a result, Fisher's Method of Scoring will only converge linearly.[4]

Quasi-Newton Methods use the computed gradients to construct approximations to the true Hessians as the algorithm progesses. These methods produce a sequence of psuedo-Hessians $\mathbf{B}_n$ that satisfy the Secant Condition:

$$\mathbf{B}_n(\mathbf{x}_n - \mathbf{x}_{n-1}) = \mathbf{g}_n - \mathbf{g}_n$$

In one dimension, finding a B_n that satisfies the Secant Condition is equivalent to computing a finite difference approximation to the second deriviative:

$$\begin{aligned} B_n(x_n - x_{n-1}) &= f'(x_n) - f'(x_{n-1}) \\ B_n &= \tfrac{f'(x_n) - f'(x_{n-1})}{x_n - x_{n-1}} \end{aligned}$$

For multivariate problems, the second derivative is in the form of a matrix, so there is not enough information to construct a full approximation afresh on each iteration. Rather the approximate Hessian is partially updated using one of several approaches.

R's `optim` routine uses the BFGS method to compute the next approximate Hessian[26]. BFGS finds the symmetric matrix $\mathbf{B}_{n+1}$ satisfying the secant condition such that the inverse $\mathbf{B}_{n+1}^{-1}$ minimises a weighted Frobenius distance between itself and the previous inverse $\mathbf{B}_n^{-1}$. A low memory variant of BFGS known as L-BFGS is also available in R's standard libary[26, 23].

Quasi-Newton Methods are slower than Newton's Method, but not overwhelmingly so.

1.1.3 Wolfe Conditions

In order to ensure the line search method converges, the steps are required to satisfy the Wolfe Conditions. There are other conditions, but these are standard. Data2LD actually makes use of the Strong Wolfe Conditions instead of the standard ones:

$$\begin{aligned} f(\mathbf{x}_k + t_k \mathbf{p}_k) &\leq f(\mathbf{x}_k) + c_1 t_k \mathbf{p}_k' \nabla f(\mathbf{x}_k) \\ |\mathbf{p}_k' \nabla f(\mathbf{x}_k + t_k \mathbf{p}_k)| &\leq c_2 |\mathbf{p}_k' \nabla f(\mathbf{x}_k)| \end{aligned}$$

The first condition ensures sufficient decrease in the objective function, the second ensures a sufficent decrease in the gradient between steps.

[3] As noted in **??**, it is actually only required that the $\mathbf{Q}_n$ approximate $\mathbf{H}_n$ along the directions which the alogorithm is searching. For the sake of simplicity, this consideration will be neglected

[4] As the sample size grows larger, the expected Fisher Information gets increasingly good at approximating $\mathbf{I}(\hat{\theta})^{-1}$, so that Fisher's Method of Scoring tends to converge faster and faster as the sample size gets bigger in a similar fashion to Figure 1.2. But that doesn not mean that Fisher's Method of Scoring achieves superlinear convergence when applied to one specific sample.

1.1.4 Extrapolation Failure

The line search method employs an approximation to the objective function at each iteration instead of working on the true function. As illustrated in Figure 1.3, the model can fail if one takes too big a step. For complex estimation problems, the objective funtion often has multiple peaks and troughs, so one must be careful that one has not wandered out of the range of validity of the locally constructed approximation.

For example, Data2LD checks whether the slope at a candidate point is positive - the line search method's approximation would regard this as an impossibility. Should this be the case, it attempts cubic interpolation at that point to find a minimum, and if that is impossible, it falls back on linear interpolation.

Data2LD is thus required to to verify that the objective function is actually behaving as predicted, this is should not be necessary according to standard implementations of line searches [23]

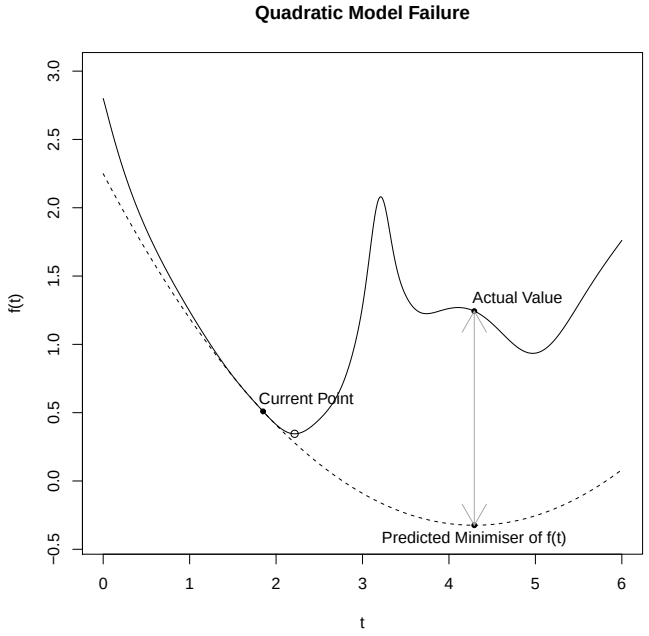

Figure 1.3: Extrapolating too far out can lead to disaster

The code in Data2LD hardcodes unnamed constants into the code. For example putting the number `3.14159` into code without context instead of `PI`. Allowing such 'Magic Numbers' is strongly discouraged because it makes code more error prone and difficult to understand [21].

1.2 How Data2LD Works in Practice

The search directions used by Data2LD are the gradient descent direction:

$$\mathbf{p}_k = -\mathbf{g}_k \tag{S1}$$

and the Newton Direction:

$$\mathbf{p}_k = -\mathbf{H}_k^{-1}\mathbf{g}_k \tag{1.1}$$

The objective function $f(\cdot)$ is then probed along the line segment $\{\mathbf{x}_k + \alpha_k\mathbf{p}_k | \alpha \geq 0\}$ to find the next point. Let $\phi(\alpha) = f(\mathbf{x}_k + \alpha\mathbf{p}_k)$ denote the value of $f(\cdot)$ restricted to the search direction $\mathbf{p}_k$. Note that $\phi(0) = f(\mathbf{x}_k)$ and $\phi'(\alpha) = \mathbf{p}_k^\top \nabla f(\mathbf{x}_k + \alpha\mathbf{p}_k)$. This in turn implies that:

$$\begin{aligned}
\phi'(0) &= \mathbf{p}_k^\top \nabla f(\mathbf{x}_k) \\
&= \left[-\frac{\nabla f(\mathbf{x}_k)}{\|\nabla f(\mathbf{x}_k)\|} \right]^\top [\nabla f(\mathbf{x}_k)] \\
&= -\frac{\nabla f(\mathbf{x}_k)^\top \nabla f(\mathbf{x}_k)}{\|\nabla f(\mathbf{x}_k)\|} \\
&= -\frac{\|\nabla f(\mathbf{x}_k)\|^2}{\|\nabla f(\mathbf{x}_k)\|} \\
&= -\|\nabla f(\mathbf{x}_k)\|
\end{aligned}$$

Since $\phi(\alpha) = \phi(0) + \phi'(0)\alpha + \mathcal{O}(\alpha^2)$ and $\phi'(0) < 0$, if α is small enough, then $\phi(\alpha) < \phi(0)$. In other words, it is always possible to to reduce $\phi(\alpha)$ so long as the step taken is small enough.

Data2LD uses four tests to determine how good a step is:[5]

- First Wolfe Condition - compares the decrease in the value of the objective function to an estimate of what it should be.

$$f(\mathbf{x}_k + \alpha_k \mathbf{p}_k) \leq f(\mathbf{x}_k) + c_1 \alpha_k \mathbf{p}_k^\top \nabla f(\mathbf{x}_k) \tag{T1}$$

- Second Wolfe Condition (Strong Version) - tests whether the gradient has decreased sufficiently relative to the previous value

$$|\mathbf{p}_k^\top \nabla f(\mathbf{x}_k + \alpha_k \mathbf{p}_k)| \leq c_2 |\mathbf{p}_k^\top \nabla f(\mathbf{x}_k)| \tag{T2}$$

- Has the function even decreased compared to the previous iteration?

$$f(\mathbf{x}_k + \alpha_k \mathbf{p}_k) \leq f(\mathbf{x}_k) \tag{T3}$$

- Has the slope along the search direction remained nonnegative?

$$\mathbf{p}_k^\top \nabla f(\mathbf{x}_k + \alpha_k \mathbf{p}_k) \leq 0 \tag{T4}$$

If **T1** and **T2** are satisfied, then the line search has converged completely. If **T3** has failed, this represents a total failure because it means the line search has failed to actually produce any improvement in the objective function. A failure in **T4** means the function has overshot a critical point.[6]

Depending on the outcome of the tests, Data2LD chooses the stepsize as follows:

- If **T1**, **T2**, and **T3** are passed, the algorithm terminates.
- If **T1** and **T1** are passed, or **T4** is passed; but **T3** is failed, it means that the slope is satisfactory, but the function has increased rather than decreased. Data2LD reduces the step size
- If all four tests are failed, then the newest point is unsuitable entirely. Data2LD falls back on interpolation to try to find a critical point of $\phi(\alpha)_k = f(\mathbf{x}_k + \alpha \mathbf{p}_k)$, falling back on the secant method if necessary.

[5] Data2LD actually tests for the negative of **T3** and **T4**, but they are presented so here so that passing a test consistetly good good and failing is consistently bad.

[6] If **T4** fails, this implies that $\mathbf{p}_k^\top \nabla f(\mathbf{x}_k + \alpha_k \mathbf{p}_k)$ and $\mathbf{p}_k^\top \nabla f(\mathbf{x}_k)$ are of opposite sign since $\mathbf{p}_k$ is choosen so that $\mathbf{p}_k^\top \nabla f(\mathbf{x}_k) < 0$. The Intermediate Value Theorem means there is an $\bar{\alpha}$ between 0 and α_k such that $\mathbf{p}_k^\top \nabla f(\mathbf{x}_k + \bar{\alpha} \mathbf{p}_k) = 0$, so that there is a crtical point on the line segment between $\mathbf{x}_k$ and $\mathbf{x}_k + \alpha_k \mathbf{p}_k$.

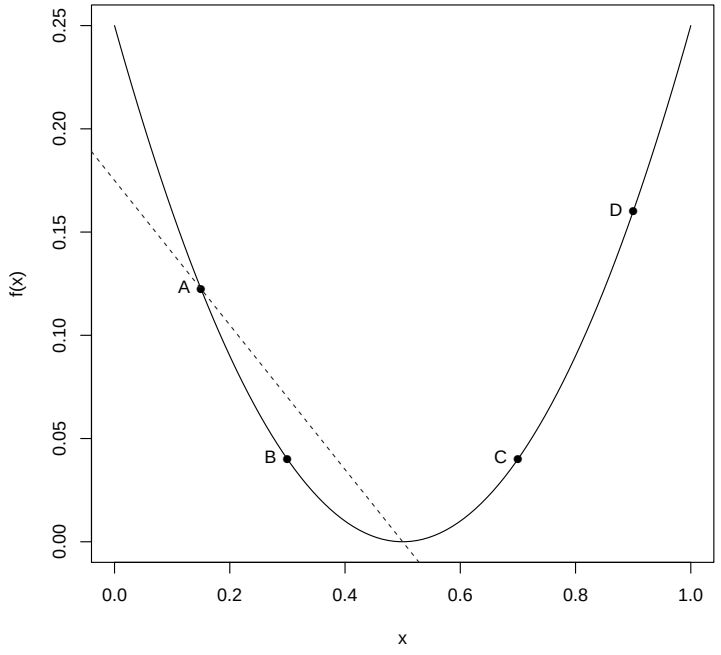

Figure 1.4: Point A is the initial point. Point B passes **T1** with $c_1 = 0.5$ and passes **T2** with $c_2 = 0.9$. Point C fails **T1** with $c_1 = 0.5$ and also fails **T3**, but passes **T4**, and passes **T2** with $c_2 = 0.9$. Point D fails all four tests.

Bibliography

[1] John P Boyd. *Chebyshev and Fourier spectral methods*. Courier Corporation, 2001.

[2] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[3] Richard P Brent. *Algorithms for minimization without derivatives*. Courier Corporation, 2013.

[4] Jiguo Cao and James O Ramsay. Parameter cascades and profiling in functional data analysis. *Computational Statistics*, 22(3):335–351, 2007.

[5] Kwun Chuen Gary Chan. Acceleration of expectation-maximization algorithm for length-biased right-censored data. *Lifetime data analysis*, 23(1):102–112, 2017.

[6] Edwin KP Chong and Stanislaw H Zak. *An introduction to optimization*, volume 76. John Wiley & Sons, 2013.

[7] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[8] Ronald Aylmer Fisher. The wave of advance of advantageous genes. *Annals of eugenics*, 7(4):355–369, 1937.

[9] Bengt Fornberg. Generation of finite difference formulas on arbitrarily spaced grids. *Mathematics of computation*, 51(184):699–706, 1988.

[10] PR Graves-Morris, DE Roberts, and A Salam. The epsilon algorithm and related topics. *Journal of Computational and Applied Mathematics*, 122(1-2):51–80, 2000.

[11] David R Hunter and Kenneth Lange. A tutorial on MM algorithms. *The American Statistician*, 58(1):30–37, 2004.

[12] Eugene Isaacson and Herbert Bishop Keller. *Analysis of numerical methods*. Courier Corporation, 2012.

[13] Carl T Kelley. *Iterative methods for linear and nonlinear equations*, volume 16. Siam, 1995.

[14] Carl T Kelley. *Implicit filtering*, volume 23. SIAM, 2011.

[15] C.T. Kelley. A brief introduction to implicit filtering. `https://projects.ncsu.edu/crsc/reports/ftp/pdf/crsc-tr02-28.pdf`, 2002. [Online; accessed 12-October-2019].

[16] Jack Kiefer. Sequential minimax search for a maximum. *Proceedings of the American mathematical society*, 4(3):502–506, 1953.

[17] Kenneth Lange. *Optimization*. Springer, 2004.

[18] Kenneth Lange. *Numerical analysis for statisticians*. Springer Science & Business Media, 2010.

[19] Kenneth Lange. The MM algorithm. `https://www.stat.berkeley.edu/~aldous/Colloq/lange-talk.pdf`, April 2007. [Online, accessed 18-September-2019].

[20] Randall J LeVeque. *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems*, volume 98. Siam, 2007.

[21] Steve McConnell. *Code complete*. Pearson Education, 2004.

[22] Geoffrey McLachlan and Thriyambakam Krishnan. *The EM algorithm and extensions*, volume 382. John Wiley & Sons, 2007.

[23] J Nocedal and SJ Wright. *Numerical Optimisation*. Springer verlag, 1999.

[24] Naoki Osada. *Acceleration methods for slowly convergent sequences and their applications*. PhD thesis, PhD thesis, Nagoya University, 1993.

[25] Yudi Pawitan. *In all likelihood: statistical modelling and inference using likelihood*. Oxford University Press, 2001.

[26] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013. ISBN 3-900051-07-0.

[27] James Ramsay. Functional data analysis. *Encyclopedia of Statistics in Behavioral Science*, 2005.

[28] Walter Rudin et al. *Principles of mathematical analysis*, volume 3. McGraw-hill New York, 1964.

[29] Christopher G Small. A survey of multidimensional medians. *International Statistical Review/Revue Internationale de Statistique*, pages 263–277, 1990.

[30] Karline Soetaert, Thomas Petzoldt, and R. Woodrow Setzer. Solving differential equations in R: Package deSolve. *Journal of Statistical Software*, 33(9):1–25, 2010.

[31] Gerald Teschl. *Ordinary differential equations and dynamical systems*, volume 140. American Mathematical Soc., 2012.

[32] Keller Vandebogart. Method of quadratic interpolation. `http://people.math.sc.edu/kellerlv/Quadratic_Interpolation.pdf`, September 2017. [Online; accessed 13-September-2019].

[33] Jet Wimp. *Sequence transformations and their applications*. Elsevier, 1981.

[34] Stephen Wright. Optimization for data analysis. In Michael W. Mahoney, John C. Duchi, and John C. Duchi, editors, *The Mathematics of Data*, chapter 2, pages 49–98. American Mathematical Society and IAS/Park City Mathematics Institute and Society for Industrial and Applied Mathematics, 2018.

[35] Tong Tong Wu, Kenneth Lange, et al. The MM alternative to EM. *Statistical Science*, 25(4):492–505, 2010.