1 Implicit Filtering

There are other methods for fitting without derivatives besides Brent's method and Parabolic Interpolation. One method that was investigated is known as the Implicit Filtering algorithm. Implicit Filtering will only be briefly covered here because it proved to be inferior to other optimisation methods.[1]

1.1 Description Of Implicit Filtering

The implicit filtering algorithm is designed for optimising problems where the exact value of the objective function $f(\cdot)$ is unavailable. Instead, $f(\cdot)$ can only be evaluated up to an arbitrary degree of accuracy. It is assumed that there is parameter h which controls the degree of the accuarcy - the lower h, the lower the error. It is usually the case that getting a higher degree of accuaracy means a higher run time.

For example, if the objective function is an expectation being approximated using a Monte Carlo method for example, it would be reasonable to set $h = 1/\sqrt{N}$ where N is the number of samples used so that the standard deviation is proportional to h. On the other hand if the expectation were being approximated by numerical integration, h would be set to the step size.

Let $f(\mathbf{x}; h)$ denote the result of approximately evauluating $f(\cdot)$ at the point $\mathbf{x}$ with precision level h. To generate a search direction, Implicit Filtering uses an approximation $\nabla_h f(\mathbf{x})$ to the gradient $\nabla f(\mathbf{x})$ that depends on h. The simplest such approximation employs forward differencing to approximate the gradient:

$$[\nabla_h f(\mathbf{x})]_i = \frac{f(\mathbf{x} + h\mathbf{e}_i; h) - f(\mathbf{x}; h)}{h} \tag{1}$$

Here $[\nabla_h f(\mathbf{x})]_i$ denotes the ith component of $\nabla_h f(\mathbf{x})$ and $\mathbf{e}_i$ is the ith basis vector. This approximate gradient is then used to define a search direction. The algorithm proceeds to conduct a line search along this direction until a point that achieves a sufficient reduction is found, which then becomes the latest iterate, and a new search direction is computed.[2]

In the event of any of the following occuring, it is deemed that more precision is needed:

- A point achieving sufficient reduction cannot be found after a maximum number of iterations.
- A clear descent direction cannot be identified.
- The approximate gradient is of a similar order of magnitude to h, so that one can't be confident that true gradient isn't in fact zero.

If any of these conditions hold, the value of h is shrunk so that $h \leftarrow \delta h$ with $0 < \delta < 1$. The algorithm then proceeds again with this higher level of precision.

The algorithm terminates when the change in the value of the objective function produced by reducing the value of h and running again is within a chosen tolerance.

The sequence of approximate values returned by Implicit Filtering is monotone decreasing so that if $m < n$ then $f(\mathbf{x}_m; h_1) \geq f(\mathbf{x}_n; h_2)$, where h_1 is the precision used with $\mathbf{x}_m$ and h_2 is the precision used with $\mathbf{x}_n$. Bear in mind however that since Implicit Filtering only ever approximately evaluates the objective function, it is not necessarily the case that $m < n$ implies $f(\mathbf{x}_m) \geq f(\mathbf{x}_n)$.

[1] Interested readers are pointed towards [20, 12, 11].

[2] More precisely, the next point $\mathbf{x}_{k+1}$ is required to satisfy a condition of the form $f(\mathbf{x}_{k+1}; h) \leq f(\mathbf{x}_k; h) - c\mathbf{d}_k^\top [\nabla_h f(\mathbf{x}_k)]$, where $\mathbf{d}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$ and $0 < c < 1$. Note that there is no requirement to decrease the norm of the approximate gradient $\nabla_h f(x)$.

1.2 Implicit Filtering and Brent's Method

It was determined that there are many disadvantages with Implicit Filtering compared to Brent's Method:

- Implicit Filtering is much more complex to code, and is thus more difficult to maintain and debug. The R code used to fit the ODE (2) by Implicit Filtering came out at a little over 300 lines long. The code in the Data2LD package that performs optimisation is over 600 lines long. Code that uses Brent's Method tends to be much shorter.
- Implicit Filtering proved to be very slow when applied to the test problem in Section 1.3 below.
- The results of the fitting are sensitive to the value of the shrink factor δ choosen.
- It can be necessary to add a penalty term to the objective function to ensure convergence.

1.3 Using Implicit Filtering to Fit an ODE to the Melanoma Data

To test Implicit Filtering, the following quasi-linear fourth order ODE was fitted to the melanoma data:[3]

$$y^{(4)} = \mu^2[1 - \sin(\pi y'')^2]y''' - \omega^2 y'' \tag{2}$$

The objective function used was a penalised sum of squared errors of the form:

$$PENSSE(f(t), \omega, \mu) = \rho \sum [y_i - f(t_i)]^2 + (1 - \rho) \int |f''(t)|^2 dt$$

The value of $PENSSE$ is influenced by ω and μ because $f(t)$ is required to be a solution of (2) with given values of ω and μ. The Implicit Filtering algorithm will not converge correctly without the penalty term as illustrated in Figure 2.

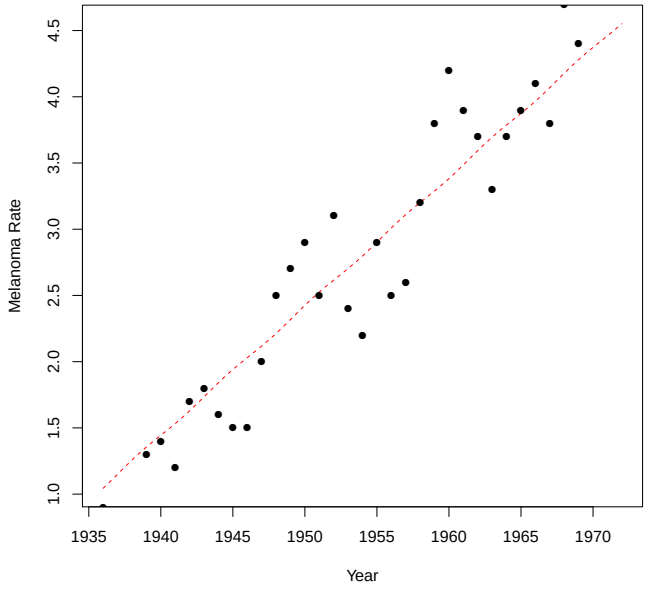
To compute $PENSSE$, the package `deSolve` was used to numerically solve (2) with the appropriate values of ω and μ. The precision factor h determined the stepsize used. The $\int |f''(t)|^2 dt$ term was approximated by taking the vector of computed values of $f''(t)$ returned by `deSolve`, and then finding the sum of squared values. As $h \to 0$, this approximation becomes more and more accurate.

As can be seen in Table 1, the algorithm takes a long time to run. It can be seen in both the table and Figure 1 that changing the value of δ can introduce qualitative changes in behaviour. The algorithm is much quicker for $\delta = 0.9$, presumably because the algorithm is converging to a different fit than for the other cases. For the fastest case where $\delta = 0.9$, 200 values of the $PENSSE$ sequence are generated before the sequence converges to within a tolerance of 10^{-4}. This statistic substantially underestimates the actual amount of work done since Implicit Filtering rejects many evaluations as being inadaquate in the course of its execuction and further evaluations are needed to compute the approximate gradients $\nabla_h f(\cdot)$. For case where $\delta = 0.9, PENSSE$ was computed over 3700 times with various values of h used.

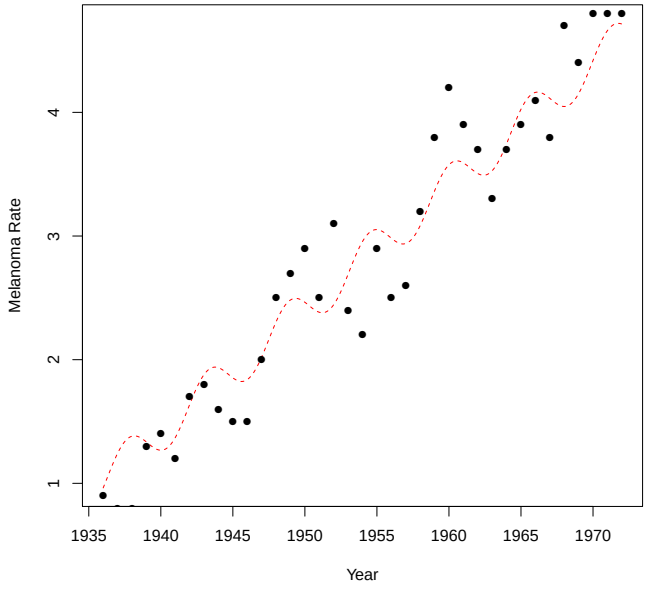
δ	Running Time (Seconds)	Running Time (Minutes)
0.7	1717.807	28.63
0.8	1611.459	26.85
0.9	1013.165	16.88

Table 1: Time taken for Implicit Filtering to fit (2) to the melanoma data for various values of δ.

[3]The version of Implicit Filtering used is actually a modified version of that described above. A Quasi-Newton algorithm was used instead of naive gradient descent to compute search directions, and central differences were used to estimate the gradient instead of forward differences as in (1).



(a)



(b)

Figure 1: Fitting the ODE (2) to the Melanoma data. The exact value of the shrink value δ effects the fit the Implicit Filtering algorithm converges to. For $\delta = 0.7$ the computed fit in (a) resembles a straight line, but $\delta = 0.9$ results in a sinusodial plus linear trend as can be seen in (b).

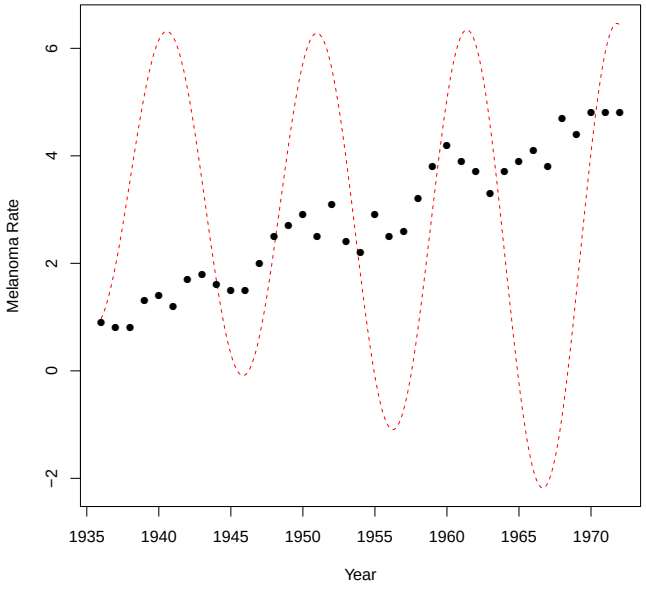


Figure 2: Without a penalty term, Implicit Filtering entirely fails to fit the ODE (2) to the melanoma data.

References

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

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

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

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

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

[6] 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.

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

[8] 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.

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

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

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

[12] 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].

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[29] 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.

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