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, which will only be briefly covered here[1]. The implicit filtering algorithm is designed for optimising problems where the objective function $f(\cdot)$ can only be evaluated up to an arbitary degree of accuracy, but the exact value is unavailable. The parameter h controls the degree of the desired 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 a Monte Carlo simulation for example, it would be reasonable to set $h = 1/\sqrt{N}$ where N is the number of samples used. If the objective function relies on numerically solving an ODE, h would be set to the step size.

Let $f(\mathbf{x}; h)$ denote the result of approximately evauluating $f(\cdot)$ at $\mathbf{x}$ with precision level h. To generate a search direction, Implicit Filtering uses a finite difference approximation to the gradient $\nabla f(\mathbf{x})$. The simplest such approximation is forward differencing, though other choices are available:

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

The approximate gradient is 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 [2].

If such a point cannot be found, or or the the approximate gradient is $\mathcal{O}(h)$ then the value of h is shrunk so that $h \leftarrow \delta h$ with $0 < \delta < 1$. The algorithm then proceeds again with a 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.

There are many disadvantages with Implicit Filtering compared to Brent's Method. First, it is much more complex to code, and is thus more difficult to maintain and debug[3]. Second, it's slow. Third, the results of the fitting are sensitive to the value of the shrink factor δ choosen. Fourth, it can be necessary to impose a penalty term to ensure convergence.

To test the implicit filtering method, the following quasi-linear fourth order ODE was fitted to the melanoma data[4].

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

The objective function used is 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 Equation 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 [25]. The precision factor h determines the stepsize used.

As can be seen in Table 1, the algorithm takes a long time to run. In addition, it can be seen in both the table and Figure 1 that changing the value of δ can introduce qualitative changes in behaviour.

[1] Interested readers are pointed towards [19, 11, 10].

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

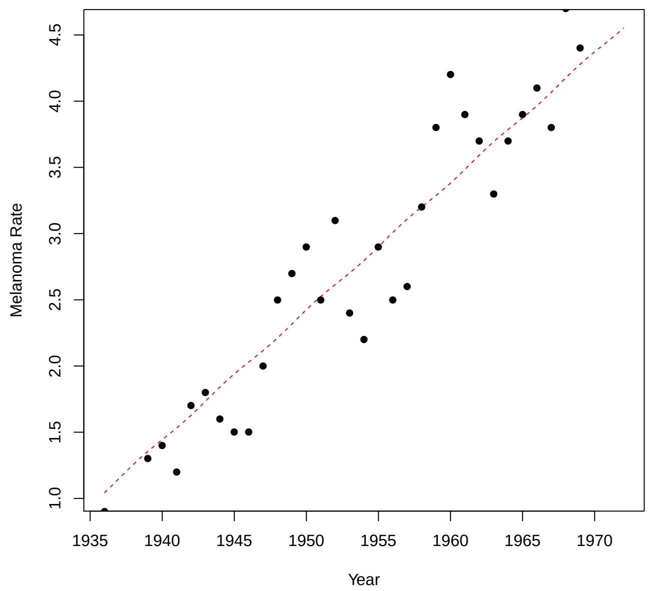
[3] The R code used to fit the ODE (2) came out at a little over 300 lines long. The code in the Data2LD package that performs optimisation is over 600 lines long. Code that uses Brent's Method tends to be much shorter.

[4] 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).

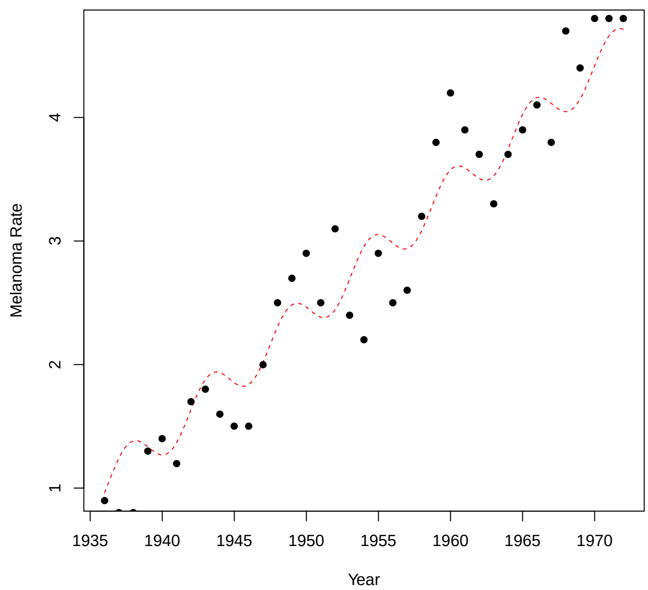
The algorithm is much quicker for $\delta = 0.9$, presumably because the algorithm is converging to a different fit than for the other cases.

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 for Implicit Filtering to converge for various values of δ.



(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 fit in (a) resembles a straight line, but $\delta = 0.9$ results in a sinusodial plus linear trend as can be seen in (b).

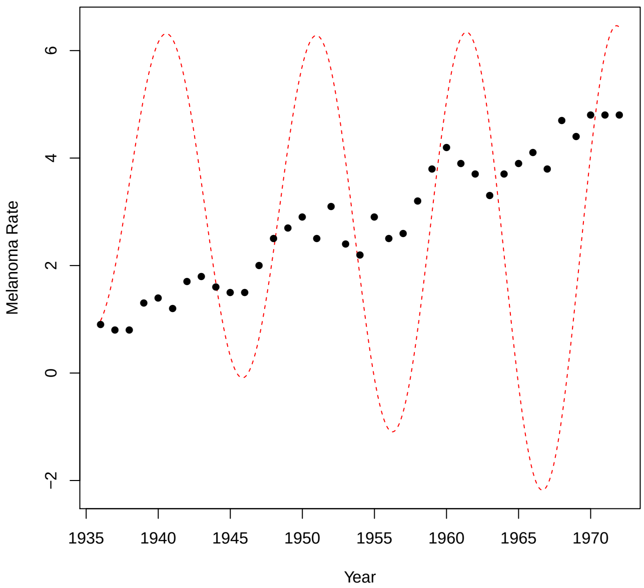


Figure 2: Without a penalty term, Implicit Filtering fails to fit the ODE (2) to the melanoma data.

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] 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.

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

[7] 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.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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