Beyond scipy.optimize.curve_fit(): nonlinear regression with MCMC

I. Introduction

Weak localization [1] is an interesting phenomenon often shown in metals and heavily-doped semiconductors. Because magnetic field breaks the time reversal symmetry (TRS) of the materials system, conductivity increases due to the broken phase correlation of electron wavefunction as magnetic field increases. One important aspect is that the magnitude of weak localization is only determined by the number of layers in the materials system. Described as:

$$\delta(B) = G(B) - G(0) = 2N\frac{e^2}{4\pi^2\hbar}\left[\psi\left(\frac{1}{2}+\frac{B_{d\phi}}{2B}\right) - \psi\left(\frac{1}{2}+\frac{B_d}{2B}\right) + \ln\left(\frac{B_d}{B_{d\phi}}\right)\right]$$

Where G is the conductivity, $e, \hbar$ are known constants. B is the variable. $B_d, B_{d\phi}, N$ are fitting parameters, where N describe the number of layers in this system.

We did the experiment for a material where 24 layers of conductors are separated by 3.5 nm-thick insulators in between each other. Since the insulators are thin enough, there is a chance that there might be cross-talk such that the effective number-of-layer will be less than 24.

II. Baseline: scipy.optimize.curve_fit()

A. Problem with curve_fit()

Normally, scipy packages are standard tools to solve such nonlinear regression problem. The function used most of the time is scipy.optimize.curve_fit() with Levenberg–Marquardt algorithm [2]. We first tested the performance of curve_fit(). Most of the time curve_fit() could not converge in 10^6 steps, while sometimes the return fitting function is a constant. This represent the least-square fitting is trapped in local minima of the cost function.

Here the cost function is defined as:

$$J\left(N, B_d, B_{d\phi}\right) = \sum_i \left(\delta_i - \hat{\delta}\left(B_i;\ N, B_d, B_{d\phi}\right)\right)^2$$

Note that $J\left(N, B_d, B_{d\phi}\right)/\sigma^2 = \chi^2$ is the goodness-of-fit parameter. Since the standard deviation of the measurement equipment is not carefully characterized, we will use cost function J to compare the goodness-of-fit within one set of data

B. Naive solution: multiple curve_fit()

Note that we only have one set of data per sample, but the data is fairly large (number of data points > 3000), we use bootstrap method to create multiple dataset at the size of 150 data points.

The naive solution is to use the curve_fit() multiple times. The algorithm is to randomly select initial trial parameter, feed curve_fit(), and retrieve the cost function. If for a certain parameter, the cost function is smaller than the subsequent N_{trial} number of trials, then we claim this set of parameter is the solution.

The result for this method (for a specific run) is shown as Fig. 1, and the calculated cost function is 4.208×10^{-6}.

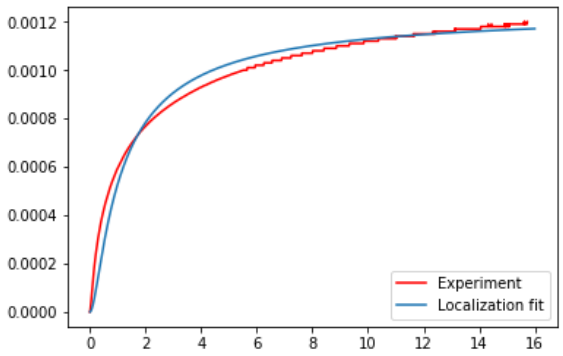

Fig. 1 Fitting result with multiple curve_fit()

III. Global minimization with MCMC

In order to avoid trapping in local minimum, we use a Markov chain Monte Carlo (MCMC) simulated annealing [3] method. In the parameter space $(N, B_d, B_{d\phi})$, initial parameter is generated using to exponential distribution ($\lambda = 100$ for N and $\lambda = 16$ for B's due to the physics background). Each movement of parameter space is multiplying the parameter by a factor drawn from a normal distribution centered at 1. Specifically, for each parameter p in $(N, B_d, B_{d\phi})$

$$p_{n+1} = (1 + \epsilon)p_n, \qquad \epsilon \sim N(\mu = 1, \sigma)$$

The annealing "temperature" cools down exponentially. In N_{step} steps, $T = 1 \to 10^{-4}$. The temperature determines if a certain uphill movement is accepted. Specifically, step-$(n+1)$ is accepted if:

$$\frac{J_{n+1}}{J_n} < e^{\alpha T}, \qquad \alpha \sim N(\mu = 0, \sigma = 1)$$

Here with temperature cools, the probability of accepting an uphill movement is harder. The system should have an increasing probability of ending in the global minimum.

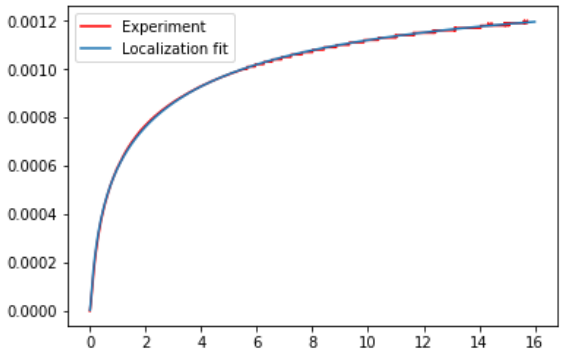

Fig. 2 Fitting result with MCMC simulated annealing.

There are also by chance times where the parameters are trapped at local minima. We perform the whole process N_{test} times to rule out the local trappings.

In order to compare the fitting parameters, it is necessary to move the parameters downhill to the bottom of the global minimum using gradient descent. So that there is no additional variance of fitting parameters due to the randomness of MCMC method.

Below is a table of cost function for the best 5 runs with and without the final gradient descent [4] step.

Table 1 Improvement with a final gradient descent step

Rank	J (without final minimization)	J (with final minimization)
1	2.71E-09	2.28E-09
2	2.72E-09	2.43E-09
3	2.82E-09	2.47E-09
4	2.94E-09	2.49E-09
5	2.97E-09	2.60E-09
Mean	2.83E-09	2.45E-09

Finally we conclude that for the dataset we have been testing, one way to search for global minimum inside lots of local minimum is the MCMC simulated annealing method. Finally we use the mean of the best 5 runs as our conclusion from this regression.

For another set of data which is measured with the same sample but at a different temperature, we get a different set of fitting parameters:

$T = 3.5\ \mathrm{K}$: $B_d = 2.574\ \mathrm{T},\ B_{d\phi} = 0.0283\ \mathrm{T},\ N = 39.785$
$T = 7\ \mathrm{K}$: $B_d = 5.205\ \mathrm{T},\ B_{d\phi} = 0.0468\ \mathrm{T},\ N = 42.342$

Here we discover that with increasing temperature, the effective number of layers changes slightly, and the magnetic diffusion field B_d and $B_{d\phi}$ both increase.

References:

[1] Hikami, S., Larkin, A.I. and Nagaoka, Y. *Spin-orbit interaction and magnetoresistance in the two dimensional random system*. Progress of Theoretical Physics, 63, 2 (1980).
[2] Marquardt, D.W. *An algorithm for least-squares estimation of nonlinear parameters*. Journal of the society for Industrial and Applied Mathematics, 431, 2 (1963).
[3] Rosario, F. and Thangadurai, K. *Simulated Annealing Algorithm For Feature Selection*. International Journal of Computers & Technology, 6471, 15 (2016).
[4] Snyman, J.A. and Wilke, D.N. *Practical Mathematical Optimization: Basic Optimization Theory and Gradient-Based Algorithms*. Springer 133 (2018).