Hands on Introduction to Bayesian optimization Metropopolis-Hastings Algorithm

Patrice M. OKOUMA

April 9, 2018

In Brief

The general aim is to to fit a model with N adjustable parameters $\Theta = (\Theta_1, \Theta_2, ..., \Theta_N)$ to a set of M data points (our observations) $\{(x_i, y_i, \sigma_i)\}_{i=1,...,M}$, where σ_i is the measurement error for datapoint y_i.

In the present case, we want to fit a straight line (our model) $y = mx + b$ to a set of data. Our model here is described by two parameters (Θ_1, Θ_2), i.e. $N = 2$ where $\Theta_1 = m$ is the slope and $\Theta_2 = b$ is the $y - intercept$ of he straight line. In other words:

$$\begin{aligned} y_i &= y(x_i) + \epsilon_i \\ &= mx_i + b + \epsilon_i \end{aligned} \tag{1}$$

The optimization problem is: " Let us determine the BEST straight line that describes our provided data set". "BEST" in either the Frequentist (hence Frequentist optimization) or the Bayesian (hence Bayesian optimization) sense.

Frequentist Optimization

The Frequentist optimization framework had us introducing a special function to maximize, namely the *likelihood*, $\mathcal{L} \equiv \mathcal{P}(D|\Theta)$, i.e. the probability of getting our dataset given our chosen model (defined by Θ). A common (well-motivated) assumption is that $\forall\ i = 1, ..., N,\ \epsilon_i$ is a realization of a random variable $\mathcal{R}_i$ *distributed as a Gaussian* with mean 0 and variance σ_i^2. If in addition, one assumes that each measurement $\{y_i\rangle_{i=1,...,N}$ are outcomes of N random variables **i.i.d**, then it can be shown that the *likelihood* of the data can be written as:

$$\mathcal{L} \propto \prod_{i=1}^{i=M} \left\{ \exp\left(\left[-\frac{1}{2} \left(\frac{y_i - y(x)}{\sigma_i} \right)^2 \right] \right) \right\} = \exp\left(-\frac{1}{2} \sum_{i=1}^{i=M} \left(\frac{y_i - y(x_i; \Theta_1; ...; \Theta_N)}{\sigma_i} \right)^2 \right) = \exp\left(-\frac{1}{2} \chi^2 \right). \tag{2}$$

Maximizing Eq.(2) is equivalent to maximizing its logarithm, or minimizing the negative of its logarithm. **Under the Gaussian likelihood assumption, the maximum likelihood estimate of the parameters is equivalent to minimizing the quantity**:

$$\chi^2 = \sum_{i=1}^{i=M} \left(\frac{y_i - y(x_i; \Theta_1; ...; \Theta_N)}{\sigma_i} \right)^2. \tag{3}$$

In other words, a General Linear Least Squares minimization is nothing but a version of Frequentist Optimization. Starting from our simple 2-parameters model ($y(x) = mx + b$), A generalization of least square is to fit a set of data points (x_i, y_i) to a model that is a linear combination of any M specified functions of x. For example, the functions could be $1 = x^0, x^1, x^2, ..., x^{M-1}$, in which case their general linear combination,

$$y(x) = \Theta_1 x^0 + \Theta_2 x^1 + \Theta_3 x^2 + ... + \Theta_M x^{M-1}, \tag{4}$$

is a polynomial of degree $M - 1$. Or, the functions could be sines and cosines, in which case their general linear combination is a harmonic series. The general form of this kind of **linear** model is:

$$y(x) = \sum_{k=1}^{M} \Theta_x X_k(x), \tag{5}$$

where $X_1(x), ..., X_M(x)$ are arbitrary fixed functions of x, called the **basis functions**. Note that the functions $X_k(x)$ can be wildly nonlinear functions of x. Here, linear refers only to the models linear dependence on its parameters $\Theta_k,\ k = 1, ..., M$.

For our linear models, we saw above that we have to minimize the **Chi-squared**:

$$\chi^2 = \sum_{k=1}^{N} \left[\frac{y_i - \sum_{k=1}^{M} \Theta_k X_k(x_i)}{\sigma_i} \right]^2, \tag{6}$$

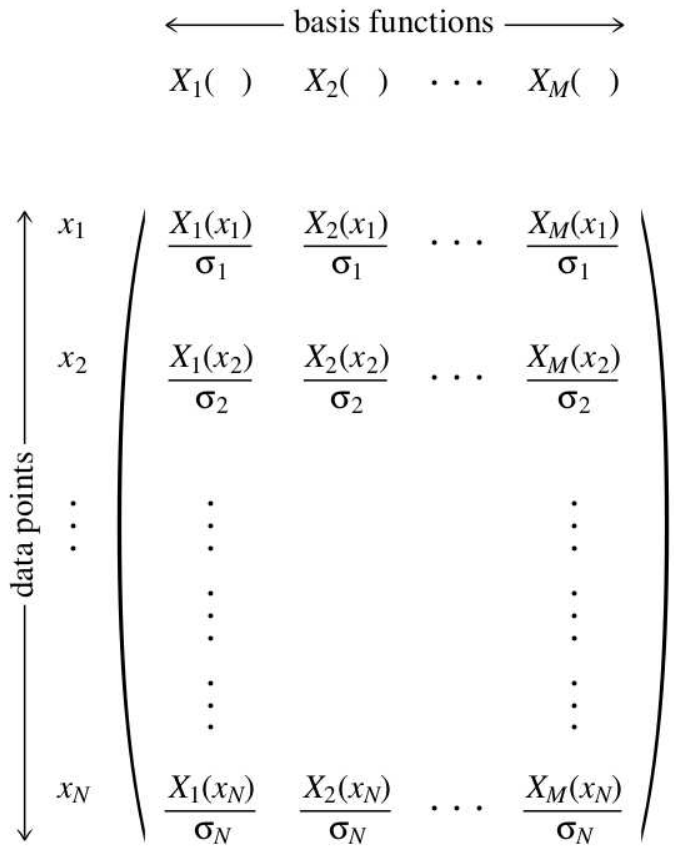

Figure 1: Design matrix for the least-squares fit of a linear combination of M basis functions to N data points. The matrix elements involve the basis functions evaluated at the values of the independent variable at which measurements are made, and the standard deviations of the measured dependent variable. The measured values of the dependent variable do not enter the design matrix.

where σ_i is the measurement error (standard deviation) of the ith data point and is presumed to be known. If the measurement errors are not known, they may all be set to the constant value $\sigma_i = \sigma = 1$. For optimization, we will determine as best parameters those that minimize χ^2. To do so, let A be a matrix whose $N \times M$ components are constructed from the M basis functions evaluated at the N abscissas x_i, and from the N measurement errors σ_i, by the prescription

$$A_{ij} = \frac{X_j(x_i)}{\sigma_i}. \tag{7}$$

The matrix **A** is often referred to as the **Design matrix** of the fitting problem. The design matrix is shown schematically in Figure 1. One also define a vector b of length N by

$$b_i = \frac{y_i}{\sigma_i}, \tag{8}$$

and denote $\overrightarrow{\Theta}$, the M vector whose components are the parameters to be fitted, $\Theta_1, ..., \Theta_M$

Solution

The minimum of Eq. 6 occurs where the derivative of χ^2 with respect to all M parameters Θ_k vanishes. In other words, one needs to solve the M equations

$$\begin{aligned}
\frac{\partial}{\partial\Theta_k}\left(\chi^2\right)=0 \quad &\Leftrightarrow \quad 0=\frac{\partial}{\partial\Theta_k}\left\{\sum_{i=1}^{N}\left[\frac{y_i-\sum_{k=1}^{M}\Theta_k X_k(x_i)}{\sigma_i}\right]^2\right\},\ k=1,...,M \qquad (9)\\
&\Leftrightarrow \quad 0=\sum_{i=1}^{N}\frac{1}{\sigma_i^2}\left[y_i-\sum_{j=1}^{M}\Theta_j X_j(x_i)\right]X_k(x_i),\ k=1,...,M\\
&\Leftrightarrow \quad 0=\sum_{i=1}^{N}\frac{y_i}{\sigma_i^2}X_k(x_i)-\sum_{i=1}^{N}\sum_{j=1}^{M}\frac{\Theta_j X_j(x_i)X_k(x_i)}{\sigma_i^2},\ k=1,...,M\\
&\Leftrightarrow \quad \sum_{i=1}^{N}\sum_{j=1}^{M}\frac{X_j(x_i)X_k(x_i)}{\sigma_i^2}\Theta_j=\sum_{i=1}^{N}\frac{y_i}{\sigma_i^2}X_k(x_i),\ k=1,...,M\\
&\Leftrightarrow \quad \sum_{j=1}^{N}\sum_{i=1}^{M}\frac{X_j(x_i)X_k(x_i)}{\sigma_i^2}\Theta_j=\sum_{i=1}^{N}\frac{X_k(x_i)}{\sigma_i}\frac{y_i}{\sigma_i},\ k=1,...,M \text{ since we can swap order of summations on RHS}\\
&\Leftrightarrow \quad \sum_{j=1}^{N}\alpha_{kj}\Theta_j=\beta_k,\ k=1,...,M
\end{aligned}$$

where

$$\alpha_{kj}=\sum_{i=1}^{N}\frac{X_j(x_i)X_k(x_i)}{\sigma_i^2},\ \text{Or equivalently } [\alpha]=\mathbf{A}^T.\ \mathbf{A}, \qquad (10)$$

is an $M\times M$ matrix, and

$$\beta_k=\sum_{i=1}^{N}\frac{y_i X_k(x_i)}{\sigma_i^2},\ \text{Or equivalently } [\beta]=\mathbf{A}^T.\ \mathbf{b}\ , \qquad (11)$$

is a vector of length M. The equations 9 or 10 are called the **normal equations** of the least-squares problem. In matrix form, the normal equations can be written as either

$$[\alpha].\ \vec{\Theta}=[\alpha]\ \text{Or as}\ \left(\mathbf{A}^T.\ \mathbf{A}\right).\ \vec{\Theta}=\mathbf{A}^T.\ \vec{b}. \qquad (12)$$

It can be solved using standard algebra techniques. That is what is done in the backbone code provided at:
https://aimshacksbayes.weebly.com/

IF INVERTIBLE, the inverse matrix

$$C_{jk}=[\alpha]^{-1}_{jk}$$

is related to the standard uncertainties of the estimated parameters $\vec{\Theta}$. One can establish that

$$\sigma^2(a_j)=C_{jj}.$$

In other words, the diagonal elements of the matrix $[C]$ are the variances (squared uncertainties) of the fitted parameters in $\vec{\Theta}$. It should not surprise you to learn that the off-diagonal elements C_{jk} are the covariances between Θ_j and Θ_k.

Bayesian Optimization

Thanks to Bayes' Theorem, one can introduce the posterior distribution, $\mathcal{P}\left(\Theta|D\right)$ given by

$$\mathcal{P}\left(\Theta|D\right)\propto\mathcal{P}(D|\Theta)\times\mathcal{P}(\Theta)$$

where $\mathcal{P}(\Theta)$ is the prior[1]:

$$\text{Posterior}\propto\text{Likelihood}\times\text{Prior}.$$

The Bayesian optimization framework essentially consists in maximizing the posterior distribution.
Born out of powerful Monte Carlo Statistical Techniques [2], the Metropopolis-Hastings algorithm below as in (Doran & Muller, 2004)[3] is a computationally cheap way to maximise BOTH the likelihood AND the posterior.

Here, we work under the Gaussian Likelihood assumption, i.e the likelihood $\mathcal{P}(D|\Theta)\propto\exp(-\frac{1}{2}\chi^2)$.

[1] It encapsulates our a-priori knowledge about the chosen model or equivalently about the parameters
[2] See C. Robert & G. Casella. Monte Carlo Statistical Methods. Springer-Verlag (2004).
[3] M. Doran & C. Muller. *Analyse this! A cosmological constraint package for cmbeasy.* astro-ph/0311311 (2004)

Algorithm

1. Choose a starting parameter vector Θ_0.
2. Using Eqs. 2 & 3, compute the Likelihood $\mathcal{L}_0 = \mathcal{P}(D|\Theta_0)$ of observing the experimental data D given the (vector of) parameters Θ_0.
3. Obtain a new parameter vector by sampling the jump from (in our case) a Gaussian Distribution with mean 0 and standard deviation vector σ: How big the characteristic jump for each parameter is controlled by a vector $\Delta\Theta = (\textit{step size for slope}, \textit{step size for Y-intercept})$.
4. Construct $U_i = \Theta_{i-1} + \Delta\Theta_{i-1}$ i.e in our case:
$$U_i \equiv (slope_i, intercept_i) = (slope_{i-1}, intercept_{i-1}) + (\text{step size for slope}, \text{step size for Y-intercept})\,. \tag{13}$$
5. Compute the Likelihood $\mathcal{P}(D|U_i) = \mathcal{L}_i$.
6. If $\mathcal{L}_i \geq \mathcal{L}_{i-1}$ then save u_i as a new "point" Θ_i in the chain ("take the step") and go to (3).
7. If $\mathcal{L}_i < \mathcal{L}_{i-1}$ then generate a random variable u from]0,1]. If $u \leq \frac{\mathcal{L}_i}{\mathcal{L}_{i-1}}$ take the step as in (6). If $u > \frac{\mathcal{L}_i}{\mathcal{L}_{i-1}}$ then reject u_i, save Θ_{i-1} as a new "point" Θ_i in the chain and go to (3).

One output will be a Chain, i.e in our case a matrix of two columns (because two parameters) and as many lines as steps taken. The post-processing of that Chain will help us to quantify our degree of belief in our model. Ideally, if one histogram any column of the Chain and take the mean, one directly gets the mean of the parameter associated to that column. The variance of that distribution leads to your standard deviation (error-bar) for that same parameter.

A few more clarifications

Let's set $\vec{\theta} = (\theta_1, \theta_2, ..., \theta_n)$, where $\{\theta_j\}_{j=1,...,m}$ are the parameters making up our model, **Bayes Theorem** states that

$$P(\vec{\theta}|D) = \frac{P(D|\vec{\theta})P(\vec{\theta})}{P(D)}$$

where the left-hand side is the *posterior probability function* which depends on the *prior* $P(\vec{\theta})$ (our a-priori knowledge about the parameters) and the likelihood $P(D|\vec{\theta})$. The denominator $P(D)$ is a normalization factor also known as the Bayesian evidence, and because we are only interested in the relative probability density functions of the different models, one can omitted for now. Priors can take the form of information obtained from previous experiments which cannot readily be incorporated into the current experiment or simply consist of feasible ranges of values for the parameters considered. However, the absence of prior information is not a restriction for the use of a Bayesian inference and estimation can still be regarded as valid [4].

Marginalization

Let's consider continuous random variables. Given a model defined by a set of parameters $\vec{\theta} = (\theta_1, \theta_2, ..., \theta_n)$ with a posterior probability density function $P(\vec{\theta}|D)$, marginalization is the procedure for obtaining the posterior probability function, $P(\vec{\theta_*}|D)$, of a sub-set of parameters $\vec{\theta_*} = \{\theta_j\}_{j=1,...,m}$ with $m < n$. The parameters other than those listed in the sub-set are considered as mere nuisance parameters that nevertheless contribute to the full posterior probability density function. Formally and in the case of continuous random variables, the posterior probability function $P(\vec{\theta_*}|D)$ is:

$$P(A|D) = \int_{\mathcal{D}} P(A, B|D)d\vec{B}, \tag{14}$$

where $A = \vec{\theta_*}$ are the parameters we are interested in and $\vec{B}$, the remaining set of parameters we are marginalizing over.

Monte Carlo Markov Chain-based parameter estimation

Let's write

$$\textit{posterior} \equiv \pi = P(\vec{\Theta}|D) \propto P(D|\vec{\Theta})P(\vec{\Theta}).$$

It is generally difficult to obtain the posterior distribution, due to (a) the dimension of the parameter vector $\vec{\Theta}$, and (b) an analytical formulation of the likelihood ($P(D|\vec{\Theta})$) functions is not always possible. A practical solution to this difficulty is to replace the analytical study of the posterior distribution with a simulation from this distribution, since producing a sample from π allows for a straightforward approximation of all integrals related with π, due to Monte Carlo principle [5]. In

[4] See C. Robert & G. Casella. Monte Carlo Statistical Methods. Springer-Verlag (2004).
[5] See also R. Trotta. arXiv:0803.4089 (2008)

a simplified version, the latter principle states that if $x_1, \ldots, x_N$ is a sample drawn from the distribution π and $\mathcal{F}$ denotes a function (with finite expectation under π), the empirical average[6]

$$\frac{1}{N} \sum_{n=1}^{N} \mathcal{F}(x_n) \tag{15}$$

is a convergent estimator of the integral

$$\Pi(\mathcal{F}) = \int \mathcal{F}(x)\pi(x)dx, \tag{16}$$

in the sense that the relation in Eq. 15 converges towards $\Pi(\mathcal{F})$ when $N \longrightarrow +\infty$. In a typical Bayesian analysis, quantities of interest usually include the posterior mean, for which $\mathcal{F}(x) = x$; the posterior covariance matrix corresponding to $\mathcal{F}(x) = xx^T$; and confidence intervals, given by $\mathcal{F}(x) = \mathbf{1}_{\mathcal{D}}(x)$, where $\mathcal{D}$ is a domain of interest, and $\mathbf{1}_{\mathcal{D}}(x)$ denotes the indicator function which is equal to one if $x \in \mathcal{D}$ and zero otherwise.

[6] D. Wraith, M. Kilbinger et al. arXiv:0903.0837 (2009)