5

Generalised MAP inference

Concept. To largely generalise the procedure of statistical inference for any model using an algorithm which builds from techniques we developed in the previous chapter. When we say 'statistical inference' here; we specifically mean computing the maximum a posteriori (MAP) estimate for any arbitrary stochastic model which has been defined in the stochadex simulator. In order for our algorithm to evaluate the MAP of a model, we show that the user must specify the prior distribution over model parameters, and the model must itself be defined within the stochadex. In this chapter, we will discuss some concepts which are commonplace within the field of Bayesian inference and provide a few simple examples of how our algorithm might work in various instances. For the mathematically-inclined, this chapter will give a very brief exposition for Bayesian statistical inference metholodology — in particular, how it relates to the evaluation of MAP estimates. For the programmers, the software described in this chapter lives in the public Git repository: https://github.com/umbralcalc/learnadex.

5.1 Inference formalism

In Bayesian inference, one applies Bayes' rule to the problem of statistically inferring a model from some dataset. This typically involves the following formula for a posterior distribution

$$\mathcal{P}_{\mathsf{t}+1}(z|Y) \propto \mathcal{L}_{\mathsf{t}+1}(Y|z)\mathcal{P}(z)\,. \tag{5.1}$$

In the formula above, one relates the prior probability distribution over a parameter set $\mathcal{P}(z)$ and the likelihood $\mathcal{L}_{\mathsf{t}+1}(Y|z)$ of some data matrix Y up to timestep $\mathsf{t}+1$ given the parameters z of a model to the posterior probability distribution of parameters given the data $\mathcal{P}_{\mathsf{t}+1}(z|Y)$ up to some proportionality constant. All this may sound a bit technical in statistical language, so it can also be helpful to summarise what the formula above states verbally as follows: the initial (prior) state of knowledge about the parameters z we want to learn can be updated by some likelihood function of the data to give a new state of knowledge about the values for z (the 'posterior' probability).

From the point of view of statistical inference, if we seek to maximise $\mathcal{P}_{\mathsf{t}+1}(z|Y)$ — or its logarithm — in Eq. (5.1) with respect to z, we will obtain what is known as a maximum posteriori

(MAP) estimate of the parameters. In fact, we have already encountered this metholodology in the previous chapter when discussing the algorithm which obtains the best fit parameters for the empirical probability reweighting. In this case; while it appears that we optimised the log-likelihood directly as our objective function, one can easily show that this is also technically equivalent obtaining a MAP estimate where one chooses a specfic prior $\mathcal{P}(z) \propto 1$ (typically known as a 'flat prior').

How might we calulate the posterior in practice with some arbitrary stochastic process model that has been defined in the stochadex? In order to make the comparison to a real dataset, any stochadex model of interest will always need to be able to generate observations which can be directly compared to the data. To formalise this a little; a stochadex model could be represented as a map from z to a set of stochastic outputs $\mathsf{Y}_{\mathsf{t}+1}(z), \mathsf{Y}_{\mathsf{t}}(z), \dots$ that are directly comparable to the rows in the real data matrix Y. The values in Y may only represent a noisy or partial measurement of the latent state of the simulation x, so a more complete picture can be provided by the following probabilistic relation

$$P_{\mathsf{t}+1}(\mathsf{y}|z) = \int_{\omega_{\mathsf{t}+1}} \mathrm{d}x \, P_{\mathsf{t}+1}(\mathsf{y}|x) P_{\mathsf{t}+1}(x|z) \,, \tag{5.2}$$

where, in practical terms, the measurement probability $P_{\mathsf{t}+1}(\mathsf{y}|x)$ of $\mathsf{y} = \mathsf{Y}_{\mathsf{t}+1}$ given $x = X_{\mathsf{t}+1}$ can be represented by sampling from another stochastic process which takes the state of the stochadex simulation as input. Given that we have this capability to compare like-for-like between the data and the simulation; the next problem is to figure out how this comparison between two sequences of vectors can be done in a way which ensures the the statistics of the posterior are ultimately respected.

For an arbitrary simulation model which is defined by the stochadex, the likelihood in Eq. (5.1) is typically not describable as a simple function or distribution. While we could train the probability reweighting we derived in the previous chapter to match the simulation; to do this well would require having an exact formula for the conditional probability, and this is not always easy to derive in the general case. Instead, there is a class of Bayesian inference methods which we shall lean on to help us compute the posterior distribution (and hence the MAP), which are known as 'Likelihood-Free' methods [1, 2, 3, 4].

'Likelihood-Free' methods work by separating out the components of the posterior which relate to the closeness of rows in Y (the matrix which comprises $\mathsf{Y}_{\mathsf{t}+1}(z), \mathsf{Y}_{\mathsf{t}}(z), \dots$) to the rows in Y from the components which relate the state x and parameters z of the simulation stochastically to Y. To achieve this separation, we can make use of chaining conditional probability like this

$$\mathcal{P}_{\mathsf{t}+1}(x, z|Y) = \prod_{\mathsf{t}'=0}^{\mathsf{t}+1} \left[\int_{\varpi_{\mathsf{t}'}} \mathrm{d}\mathsf{y}' \, \mathcal{P}_{\mathsf{t}'}(\mathsf{y}'|Y) \right] P_{\mathsf{t}+1}(x, z|\mathsf{Y}) \,, \tag{5.3}$$

where $\varpi_{\mathsf{t}'}$ here corresponds to the domain of y' at time t'.

As we demonstrated in the previous chapter, it's possible for us to also optimise a probability distribution $\mathcal{P}_{\mathsf{t}+1}(\mathsf{y}|Y) = P_{\mathsf{t}+1}[\mathsf{y}; M_{\mathsf{t}+1}(Y), C_{\mathsf{t}+1}(Y), \dots]$ for each step in time to match the statistics of the measurements in Y as well as possible, given some statistics $M_{\mathsf{t}+1}(Y)$ and $C_{\mathsf{t}+1}(Y)$. We do not necessarily need to obtain these statistics from the probability reweighting method, but could instead try to fit them via some other objective function. Either way, this represents a lossy *compression* of the data we want to fit the simulation to, and so the best possible fit is desirable; regardless of overfitting. This choice to summarise the data with statistics means we are using what is known as a Bayesian Synthetic Likelihood (BSL) method [2, 3] instead of another class of methods which approximate an objective function directly using a proximity kernel — known as Approximate Bayesian Computation (ABC) methods [1].

Let's consider a few concrete examples of $P_{\mathsf{t}+1}[\,\mathsf{y}; M_{\mathsf{t}+1}(Y), C_{\mathsf{t}+1}(Y), \dots]$. If the data measurements were well-described by a multivariate normal distribution, then one would use terms like

$$P_{\mathsf{t}+1}[\,\mathsf{y}; M_{\mathsf{t}+1}(Y), C_{\mathsf{t}+1}(Y), \dots] = \mathsf{MultivariateNormalPDF}[\,\mathsf{y}; M_{\mathsf{t}+1}(Y), C_{\mathsf{t}+1}(Y)]\,, \tag{5.4}$$

where $M_{\mathsf{t}+1}(Y)$ and $C_{\mathsf{t}+1}(Y)$ would be estimated from the data measurements in Y. Similarly, if the data measurements were instead better described by a Poisson distribution, we might disregard the need for a covariance matrix statistic $C_{\mathsf{t}+1}(Y)$ and instead use

$$P_{\mathsf{t}+1}[\,\mathsf{y}; M_{\mathsf{t}+1}(Y), C_{\mathsf{t}+1}(Y), \dots] = \mathsf{PoissonPMF}[\,\mathsf{y}; M_{\mathsf{t}+1}(Y)]\,. \tag{5.5}$$

Once again, here, we would need to determine $M_{\mathsf{t}+1}(Y)$ from Y. The more statistically-inclined readers may notice that the probability mass function here would require the integrals in Eq. (5.3) to be replaced with summations over the relevant domains.

Eq. (5.3) demonstrates how one can construct a statistically meaningful way to compare the sequence of real data measurements $Y_{\mathsf{t}+1}, Y_{\mathsf{t}}, \dots$ to their modelled equivalents $\mathsf{Y}_{\mathsf{t}+1}(z), \mathsf{Y}_{\mathsf{t}}(z), \dots$. But we still haven't shown how to compute $P_{\mathsf{t}+1}(x, z|\mathsf{Y})$ for a given simulation, and this can be the most challenging part. To begin with, we can reapply Bayes' rule and the chaining of conditional probability to find

$$P_{\mathsf{t}+1}(x, z|\mathsf{Y}) \propto P_{\mathsf{t}+1}(\mathsf{y}|x) P_{\mathsf{t}+1}(x|z) P_{\mathsf{t}}(z|\mathsf{Y}')\,, \tag{5.6}$$

where, to keep the expression simpler, we are leveraging our notation in previous chapters to use Y' as indicating all of the past rows of simulation measurements up to Y_{t}.

The relationship between $P_{\mathsf{t}+1}(\mathsf{y}|x)$ and previous timesteps can be directly inferred from the probabilistic iteration formula that we introduced in the previous chapter. So we can map probabilities of x throughout time and learned information about the state of the system can be applied from previous values, given z. But is there a similar relationship we might consider for $P_{\mathsf{t}+1}(z|\mathsf{Y})$? Yes there is! The marginalisation

$$P_{\mathsf{t}+1}(z|\mathsf{Y}) = \int_{\omega_{\mathsf{t}+1}} \mathrm{d}x\, P_{\mathsf{t}+1}(x, z|\mathsf{Y}) \propto \bigg[\int_{\omega_{\mathsf{t}+1}} \mathrm{d}x\, P_{\mathsf{t}+1}(\mathsf{y}|x) P_{\mathsf{t}+1}(x|z) \bigg] P_{\mathsf{t}}(z|\mathsf{Y}')\,, \tag{5.7}$$

shows how the z updates can occur in an iterative fashion. The reader may also recognize the factor above in brackets as Eq. (5.2). To complete the picture, one can combine the x and z updates into a joint distribution update which takes the following form

$$P_{\mathsf{t}+1}(x, z|\mathsf{Y}_{\mathsf{t}+1}, \mathsf{Y}_{\mathsf{t}}, \dots) \propto \frac{1}{\mathsf{t}} \sum_{\mathsf{t}'=0}^{\mathsf{t}} \int_{\omega_{\mathsf{t}'}} \mathrm{d}x'\, P_{\mathsf{t}+1}(\mathsf{y}|x) P_{(\mathsf{t}+1)\mathsf{t}'}(x|x', z) P_{\mathsf{t}'}(x', z|\mathsf{Y}_{\mathsf{t}'}, \mathsf{Y}_{\mathsf{t}'-1}, \dots)\,, \tag{5.8}$$

where we have expanded out the conditional arguments for more clarity on which rows of the simulation measurements matrix are being used at each point in time, e.g., $P_{\mathsf{t}+1}(x, z|\mathsf{Y}) \equiv P_{\mathsf{t}+1}(x, z|\mathsf{Y}_{\mathsf{t}+1}, \mathsf{Y}_{\mathsf{t}}, \dots)$.

5.2 Online/offline inference algorithms

We now have a way of probabilistically translating the current state of knowledge about (x, z) forward through time as new data is received, because Eq. (5.8) tells us how this should work. We also know how to connect these simulated distributions to the real data because Eq. (5.3) essentially gives us an objective function to maximise for each step in time. This is all great in theory; but in practice, this optimisation problem typically has several layers of difficulty to it. Since the model has been defined by its stochastically generated samples $\mathsf{Y}_{\mathsf{t}+1}(z), \mathsf{Y}_{\mathsf{t}}(z), \dots$, the objective function will manifestly be stochastic too. Another layer of difficulty is that gradients of the objective function are not immediately computable and so navigation around the optimisation domain could be difficult, especially in high-dimensional problems. Lastly, given that the simulation model in the stochadex needs to be running multiple times for each timestep, we need a way of mitigating computational expense.

Talk about stochastic resetting in the broader context from the literature, stating that in this paper [5] the First Passage Time (FPT) distribution for a particle undergoing Brownian motion between two fixed points in space has an infinite mean FPT, its mean FPT with stochastic resetting becomes finite — so in practice, sampling towards a target can be infinitely faster. And, in this paper [6], they show this works well for molecular dynamics simulations in practice. The algorithm is specifically: 1. sample new values for (x, z) for the current resetters from the current (x, z) distribution 2. run the iterations for the resetters from the back of the window all the way up to the new step and for the non-resetters just take another step 3. determine the new MAP (x, z) sample and store it for this point in time 4. either optimise using a posterior shape or update the statistics of the posterior over (x, z) using Eq. (5.3) as additional weightings for each member of the ensemble (consider using annealing to reduce the dependence on history and make this even more like reinforcement learning) 5. draw the resetters randomly with a random reset rate, iterate time forward, stream new datapoints in go to 1.

As such an algorithm converges, we can recompute (and hence iteratively improve) the MAP estimate with respect to each iteration of the posterior.

Readers with some machine learning experience may be familiar with the classic exploration vs exploitation tradeoffs. It's clear that these tradeoffs will manifest in our case here when trying to strike a balance between iterating the posterior distribution and optimizing the current posterior with respect to (x, z) to compute the MAP.

Readers of the previous section may also have recognized that Eq. (5.8) contains the same conditional probability $P_{(\mathsf{t}+1)\mathsf{t}'}(x|x, z)$ as the reweighting algorithm. This structure enables us to reuse all of the exposition we provided for the probabilistic reweighting and highlights how the reweighting itself can be used in the algorithm to optimise the posterior.

If we now synthesize both of these observations together, we can see how a stochastic variant of the well-known Expectation-Maximisation Algorithm [7, 8, 9] naturally emerges.

Bibliography

[1] S. A. Sisson, Y. Fan and M. Beaumont, *Handbook of approximate Bayesian computation.* CRC Press, 2018.

[2] L. F. Price, C. C. Drovandi, A. Lee and D. J. Nott, *Bayesian synthetic likelihood*, *Journal of Computational and Graphical Statistics* **27** (2018) 1–11.

[3] S. N. Wood, *Statistical inference for noisy nonlinear ecological dynamic systems*, *Nature* **466** (2010) 1102–1104.

[4] C. Drovandi and D. T. Frazier, *A comparison of likelihood-free methods with and without summary statistics*, *Statistics and Computing* **32** (2022) 42.

[5] M. R. Evans and S. N. Majumdar, *Diffusion with stochastic resetting*, *Physical review letters* **106** (2011) 160601.

[6] O. Blumer, S. Reuveni and B. Hirshberg, *Stochastic resetting for enhanced sampling*, *The journal of physical chemistry letters* **13** (2022) 11230–11236.

[7] H. O. Hartley, *Maximum likelihood estimation from incomplete data*, *Biometrics* **14** (1958) 174–194.

[8] A. P. Dempster, N. M. Laird and D. B. Rubin, *Maximum likelihood from incomplete data via the em algorithm*, *Journal of the royal statistical society: series B (methodological)* **39** (1977) 1–22.

[9] K. P. Murphy, *Machine learning: a probabilistic perspective.* MIT press, 2012.