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 methodology

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

$$\mathcal{P}_{\mathsf{t}+1}(z|y,y',y'',\dots) \propto \mathcal{L}_{\mathsf{t}+1}(y,y',y'',\dots|z)\mathcal{P}(z)\,. \tag{5.1}$$

In the formula above, one relates the prior probability distribution over a parameter set $\mathcal{P}(z)$ and the likelihood of the of some data measurements $(y,y',y'',\dots)$ up to timestep $\mathsf{t}+1$ given this parameter set of a model $\mathcal{L}_{\mathsf{t}+1}(y,y',y'',\dots|z)$ to the posterior probability distribution of parameters given the data $\mathcal{P}_{\mathsf{t}+1}(z|y,y',y'',\dots)$ up to a 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 $\mathcal{P}_{\mathsf{t}+1}(z|y,y',y'',\dots)$ — or its logarithm — in Eq. (5.1) with respect to a given set of data and parameters, we will obtain what is known as a maximum posteriori (MAP) estimate of the parameters which fit the data given a specified model. 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 filter. 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').

So we need to now specify in a little more detail how Eq. (5.1) translates into a practical calculation with some arbitrary stochastic process model that has been defined in the stochadex. In the general case, this stochadex model of interest must be able to use z to generate a set of outputs from a particular realisation of the stochastic process $(y_z,y'_z,y''_z,\dots)$ that are directly comparable to the measurements made in the real data $(y,y',y'',\dots)$. Since that we have this capability; the first problem is to figure out how we compare these two sequences of vectors in a way which ensures the the statistics of the likelihood are respected.

If we now apply Bayes' rule to the likelihood itself in Eq. (5.1), we can easily find the following proportionality relationship

$$\mathcal{L}_{\mathsf{t}+1}(y,y',y'',\dots|z) \propto P(z|y,y',y'',\dots)P(y,y',y'',\dots)\,. \tag{5.2}$$

Every model one can specify in the stochadex can be represented by some process which stochastically maps z onto a set of comparable measurements $(y_z,y'_z,y''_z,\dots)$. Given this identification, we may generally assert that

$$P(z|y,y',y'',\dots) = P(y_z|y,y',y'',\dots)P(y'_z|y',y'',\dots)P(y''_z|y'',\dots)\,. \tag{5.3}$$

As we demonstrated in the previous chapter, we can also optimise a probability distribution $P_{\mathsf{t}+1}(y;M_{\mathsf{t}+1},C_{\mathsf{t}+1},\dots)$ for each step in time to match the statistics of the measurements $(y,y',y'',\dots)$ as well as possible, given some statistics $M_{\mathsf{t}+1}$ and $C_{\mathsf{t}+1}$. We do not necessarily need to obtain these statistics from the probability filter estimation method, but could instead try to fit them via some other objective function. If we apply both of these changes, our likelihood in Eq. (5.2) can be rewritten as

$$\ln \mathcal{L}_{\mathsf{t}+1}(y,y',y'',\dots|z) = \mathrm{const.} + \ln P_{\mathsf{t}+1}(y_z;M_{\mathsf{t}+1},C_{\mathsf{t}+1},\dots) + \ln P_{\mathsf{t}}(y'_z;M_{\mathsf{t}},C_{\mathsf{t}},\dots) + \dots\,. \tag{5.4}$$

This relationship demonstrates how one can construct a meaningful way to compare the sequence of real data measurements $(y,y',y'',\dots)$ to their modelled equivalents $(y_z,y'_z,y''_z,\dots)$.

Let's consider a few concrete examples of Eq. (5.4). If the data measurements were well-described by a multivariate normal distribution, then one would insert terms like

$$P_{\mathsf{t}+1}(y_z;M_{\mathsf{t}+1},C_{\mathsf{t}+1},\dots) \propto \mathsf{MultivariateNormalPDF}(y_z;M_{\mathsf{t}+1},C_{\mathsf{t}+1})\,, \tag{5.5}$$

into Eq. (5.4) where $M_{\mathsf{t}+1}$ and $C_{\mathsf{t}+1}$ would be estimated from the data measurements $(y,y',y'',\dots)$. Similarly, if the data measurements were instead better described by a Poisson distribution, we might disregard the need for a covariance matrix statistic $C_{\mathsf{t}+1}$ and instead use

$$P_{\mathsf{t}+1}(y_z;M_{\mathsf{t}+1},C_{\mathsf{t}+1},\dots) \propto \mathsf{PoissonPMF}(y_z;M_{\mathsf{t}+1})\,, \tag{5.6}$$

in Eq. (5.4). Once again, here, we would need to determine $M_{\mathrm{t+1}}$ from $(y, y', y'', \dots)$. The more statistically-inclined readers may also note here that we can get away with inserting either a probability mass function or density function in this context because Eq. (5.1) allows for an arbitrary normalisation constant.

We now have an objective function to optimise values of z with respect to 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 $(y_z, y'_z, y''_z, \dots)$, 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. To solve this issue in a way which maintains the generality of our approach, it turns out that we can rely on another application of our probability filter!

Let's now consider a process which represents the progress made by an optimiser towards the optimum value, whose state is defined by the parameters z. If we assume that all the elements of z are continuous[1] in their domain υ, one representation of this process we might consider is

$$\mathcal{P}^{n+1}_{\mathrm{t+1}}(z|y, y', y'', \dots) = \frac{1}{n}\sum_{n'=0}^{n}\int_{\upsilon} \mathrm{d}z'\, P^{(n+1)n'}(z|z')\, \mathcal{P}^{n'}_{\mathrm{t+1}}(z'|y, y', y'', \dots)\,, \tag{5.7}$$

where n is the iteration number of the optimisation, and corresponds to a different candidate posterior distribution $\mathcal{P}^{n}_{\mathrm{t+1}}$ to one at any of the previous steps n'. Eq. (5.7) can represent the convergence of a probabilistic optimisation algorithm — represented in particular by $P^{(n+1)n'}(z|z')$ — towards the true actual posterior distribution of z.

At this point, readers may recognize that Eq. (5.7) has the same structure as the generalised probabilistic description we introduced previously. This is a deliberate construction, since it enables us to reuse all of the exposition we provided for the probabilistic filter to recognize that the filter itself could serve as an algorithm to optimise the posterior.

- Need to point out here that because we are only interested in the MAP, the algorithm can stop converging to the posterior and can exploit the function it creates to get to the optimum.
- consider a tunable conditional probability that is a Gaussian with mean and variance estimated directly from the history (don't optimise it like in Bayesian optimisation!) with a tunable exponential timestep kernel (timestep being the optimiser step here) and the resulting function can be optimised (or draw monte-carlo samples from) in an EM algorithm approach. The resulting Gaussian function could also exploit gradients for SGD.
- Illustrate the algorithm structure and describe the maths for it step-by-step as well: 1. choose random sampling for $P^{(n+1)n'}(z|z')$ 2. alternate between using the probability filter for $P^{(n+1)n'}(z|z')$ to sample and optimisation of the current posterior iteration with respect to z
- Reference the contrast to ABC methods here, which involve approximating the data likelihood with a simple proximity function with a tolerance ϵ.
- Also talk about the BOLFI method which does indeed use the full Bayesian optimisation as it goes.

[1] We can modify this expression to work for discrete variables too.

- Programming steps for this chapter: need to write a new OptimisationAlgorithm which implements this algorithm — the rest should just be config

Bibliography