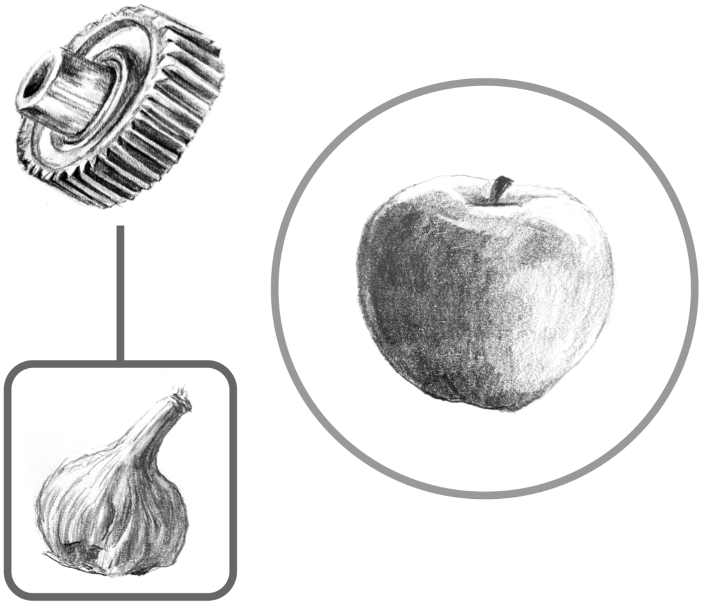

5

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 involves the following formula for a posterior distribution

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

In the formula above, one relates the prior probability distribution over a parameter set $\mathcal{P}(\theta)$ 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 $\mathcal{L}_{\mathsf{t}+1}(y, y', y'', \dots \vert \theta)$ to the posterior probability distribution of parameters given the data $\mathcal{P}_{\mathsf{t}+1}(\theta \vert y, y', y'', \dots)$ 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 θ 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 θ (the ‘posterior’ probability).

From the point of view of statistical inference, if we seek to maximise $\mathcal{P}_{\mathsf{t}+1}(\theta|y,y',y'',\dots)$ — 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 $\mathcal{P}(\theta) \propto 1$ (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 generate a set of outputs $(y_{\rm mod}, y'_{\rm mod}, y''_{\rm mod}, \dots)$ that are directly comparable to the measurements made in the real data $(y, y', y'', \dots)$. Given 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.

As we demonstrated in the previous chapter, we can 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 estimated statistics. We do not necessarily need to obtain these statistics from the probability filter method, but could instead construct some other method to obtain them. The key point here is that, because the measurements are all assumed to be independent to one another, we can construct a cumulative log-likelihood which compares model outputs to the measurements for the whole dataset like this

$$\ln \mathcal{L}_{\mathsf{t}+1} = \ln P_{\mathsf{t}+1}(y_{\rm mod}; M_{\mathsf{t}+1}, C_{\mathsf{t}+1}, \dots) + \ln P_{\mathsf{t}}(y'_{\rm mod}; M_{\mathsf{t}}, C_{\mathsf{t}}, \dots) + \dots \,. \tag{5.2}$$

Once you have a data likelihood, then inference should proceed on the sampling domain side of things where the filtering algorithm is used again! This time with 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.

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.

Bibliography