**Methods of Bayesian ML**

**MAP**

While MAP is the first step towards fully Bayesian machine learning, it’s still only computing what statisticians call a *point estimate*, that is the estimate for the value of a parameter at a single point, calculated from data. The downside of point estimates is that they don’t tell you much about a parameter other than its optimal setting. In reality, we often want to know other information, like how certain we are that a parameter’s value should fall within this predefined range.

To that end, the true power of Bayesian ML lies in the computation of the entire posterior distribution. This is a tricky business though. Distributions are not nicely packaged mathematical objects that can be manipulated at will. Often they come defined to us as tricky, intractable integrals over continuous parameter spaces that are infeasible to analytically compute. Therefore, a number of fascinating Bayesian methods have been devised that can be used to *sample*(i.e. draw sample values) from the posterior distribution.

**MCMC**

Probably the most famous of these is an algorithm called *Markov Chain Monte Carlo*, an umbrella which contains a number of subsidiary methods such as *Gibbs*and *Slice Sampling.* The math behind MCMC is difficult but intriguing. In essence, these methods work by constructing a known Markov chain which settles into a distribution that’s equivalent to the posterior. A number of successor algorithms improve on the MCMC methodology by using gradient information to allow the sampler to more efficiently navigate the parameter space.

MCMC and its relatives are often used as a computational cog in a broader Bayesian model. Their downside is that they are often very computationally inefficient, although this drawback has been improved tremendously in recent years. That said, it’s often preferable to use the simplest tool possible for any given job.

To that end, there exist many simpler methods which can often get the job done. For example, there exist Bayesian linear and logistic regression equivalents in which something called the *Laplace Approximation* is used. This algorithm provides an analytical approximation to the posterior distribution by computing a second-order Taylor expansion around the log-posterior and centered at the MAP estimate.

**Gaussian process**

One popular Bayesian method capable of performing both classification and regression is the *Gaussian process*. A GP is a stochastic process with strict Gaussian conditions imposed upon its constituent random variables. GPs have a rather profound theoretical underpinning, and much effort has been devoted to their study. In effect, these processes provide the ability to perform regression in function space.

That is, instead of choosing a single line to best fit your data, you can determine a probability distribution over the *space of all possible lines* and then select the line that is most likely given the data as the actual predictor. This is Bayesian estimation in the truest sense in that the full posterior distribution is analytically computed. The ability to actually work out the method in this instance is due to the suitability of *conjugate functions*.

In the case of classification using GPs, the posterior is once again no longer conjugate to the likelihood, and the ability to do analytic computations breaks down. In this case, it’s necessary to once again resort to approximate solvers like the Laplace Approximation in order to suitably train the model to a desired level of accuracy.

**The presence of Bayesian models in ML**

While Bayesian models are not nearly as widespread in industry as their counterparts, they are beginning to experience a new resurgence due to the recent development of computationally tractable sampling algorithms, greater access to CPU/GPU processing power, and their dissemination in arenas outside academia.

They are especially useful in low-data domains where deep learning methods often fail and in regimes in which the ability to reason about a model is essential. In particular, they see wide use in Bioinformatics and Healthcare, as in these fields taking a point estimate at face value can often have catastrophic effects and more insight into the underlying model is necessary.

For example, one would not want to naively trust the outputs of an MRI cancer prediction model without at least first having some knowledge about how that model was operating. Similarly, within bioinformatics, variant callers such as Mutect2 and Strelka rely heavily on Bayesian methods underneath the hood.

These software programs take in DNA reads from a person’s genome and label “variant” alleles which differ from those in the reference. In this domain, accuracy and statistical soundness are paramount, so Bayesian methods make a lot of sense despite the complexity that they add in implementation.

Overall, Bayesian ML is a fast growing subfield of machine learning and looks to develop even more rapidly in the coming years as advancements in computer hardware and statistical methodologies continue to make their way into the established canon.