**Week 4 Notes**

Basically, this week’s content is all about sampling. This sampling will allow us to be able to estimate expected values. We can see how this will allow us to perform Bayesian inference by going back to that all important equation that we started with in week 1.

This brings us to the Monte Carlo estimate. An analogy I make is with surveying. A survey can be used to approximate the value of an attribute of a population, such as height. Surveying works for a large and notionally infinite population. The intuition is that a survey tends to exhibit the same properties as the population from which it is drawn.

For example, the average height, f, of people, p, in London is:

There are important properties of this, namely that the estimator is unbiased, which essentially means that it will converge to the true value given enough samples.

and the second is that the variance shrinks as the sample size increases, meaning that our uncertainty with the approximation shrinks:

The next question becomes, how do we get the samples that the above equations needs. There are a number of methods which were covered in the notes, the first being rejection sampling. The premise to rejection sampling is as follows. You start with an intractable distribution, p, that you know up to a constant. You then choose a distribution q such that every point on q is larger than p.

The first step is to sample uniformly a value of x. The next step is to uniformly sample a point in the range (0, q(x)). Finally, we reject the sample with probability p(x) / q(x)

We can see that this ratio will increase in high dimensionality cases and as we reject according to this ratio, it quickly becomes very inefficient.

We now move on to Markov Chain Monte Carlo techniques that allow us to sample from high dimensionality distributions. To start with we need to talk about Markov chains. Markov chains are a stochastic model describing a sequence of possible events in which the probability of each event depends only on the state attained in the previous event. This assumes that one can make predictions for the future based solely on its present state. More formally:

A Markov chain is uniquely defined by its transition probabilities.

A Markov chain has the following properties:

* A Markov chain converges if it is stationary. If you start at distribution pi, if you make one transition or one time step, the distribution you get is again pi. This means if we encounter this pi, then we will stay with it.
* If it is reducible – i.e. if you can transfer from a state to any other with non-zero probability.
* A Markov chain is said to be reversible if it satisfies the detailed balance condition. This is defined as there being a probability distribution over its states such that

There is a theorem that if these two conditions hold then the chain converges.

Essentially a biased random walk exploring a distribution which gives approximate, correlated samples. How can we build these chains – one method is Gibbs sampling. This is an algorithm for obtaining a sequence of observations that are approximated from a specified multivariate probability distribution where direct sampling is difficult.

**Metropolis-Hastings**

A Markov process has a unique stationary distribution when the following two conditions are met: Must be ergodic, which states that every state be aperiodic – meaning that the system does not return to the same state at fixed intervals and positive recurrent – the expected number of steps for returning to the same state is finite. MH designs such a Markov process by:

Satisfies detailed balance by rejecting moves. A transition operator to propose a move from the current state to a new state, e.g. a Gaussian. At each iteration of the algorithm, the new candidate is accepted or rejected with:

EQUATION – rewrite detailed balance and separate into two sub-steps:

Diffusion time.

Combine transition operators.

Notes on MCMC: the samples are not independent so we should thin the results, i.e. only keep every kth sample or even use the Monte Carlo method on MCMC samples.. Use a burn-in period.

**LDA**

LDA is a model to find topics in a corpus of documents. The idea is that each document is a distribution over topics and that each topic is a distribution over words.

MCMC for LDA: In variational approach we made phi – the distribution over words for each topic – parameters, as opposed to distributions (i.e. full Bayesian).

Collapsed Gibbs for LDA: If we can integrate some parts out of model it will make it easier (which we can – by choosing Dirilecht). We can marginalise out theta (see screenshot).

**Bayesian Neural Networks**

Bayesian NNs have distributions over weights instead of weights. We treat the weights as latent variables and for predictions we marginalise w out. EQUATION 1 for every possible value for w. i.e. pass image through NN with every value of w. Then weight these by the posterior. This allows us to get the benefits of Bayesian inference in NNs. i.e we can tune hyperparameters, get uncertainty scores, etc.

**Any interesting examples of the above methods.**