BAYESIAN LEARNING FOR NEURAL NETWORK

Abstract:

Artificial Neural Networks are now widely used as flexible models for regression and classification applications. But questions remain what these models mean, and how can they safely be used when training data is limited. Overfitting occurs when a model is excessively complex (e.g. neural network), such as having too many parameters relative to the number of observations. Bayesian methods allow complex neural network models to be used without fear of the overfitting. Use of these models in practice is made possible using Markov chain Monte Carlo (MCMC) techniques.

Markov Chain Monte Carlo

Markov Chain Monte Carlo refers to a class of algorithms for sampling from probability distributions in a special way. It does this by constructing a **Markov Chain** which converges after a certain number of steps to the desired probability distribution. A Markov Chain is a process in which the next step or iteration of the process only depends upon the current step and not upon any previous steps in the process. If a Markov Chain has certain properties then the process will evolve in a random fashion until it reaches a certain state in which it remains thereafter, called **equilibrium**. Within the context of Bayesian Statistics it is desirable to sample from a posterior distribution. An MCMC algorithm here constructs a Markov Chain in which the random evolution of the chain is probabilistic, each step in the chain constructs an empirical distribution which is a Monte Carlo approximation of the posterior, and the chain converges to an equilibrium distribution which is the posterior distribution. It is this fact regarding the guaranteed convergence to the desired equilibrium distribution which enables the empirical distribution generated to serve as a Monte Carlo simulation of the posterior distribution.

The simplest MCMC algorithms are **random walks** which means that each step in the algorithm takes small, random steps around the target equilibrium (or posterior in a Bayesian context) distribution. After a sufficient number of steps have been taken then the full equilibrium distribution has been explored. A key issue is how these steps are defined and how to minimize the number of steps required exploring the entire target equilibrium distribution. There are no specific correct answers to these questions, but a range of different algorithms exist each offering a different way of sampling from the desired probability distribution. Some MCMC algorithms are tuned in order to fit the context of the problem at hand, since poor tuning can lead to poor Monte Carlo simulations.

BAYESIAN LEARNING FOR NEURAL NETWORK

MCMC Simulation

- A dependent sequence (a chain) of random variables, $\{\theta(i)\}$ i=1 to M, with approximate distribution, $p(\theta(i)) \approx p(\theta|y)$
- The chain is initialized with a user defined starting value, $\theta(0)$
- The Markov property then specifies that the distribution of $\theta(i+1) \mid \theta(0), \theta(1), \ldots, \theta(i)$ depends only on the current state of the chain $\theta(i)$
- A Markov chain is specified by its transition kernel defined as,

  $$P(\theta(i), A) =_{def} P(\theta(i+1) \in A|\theta(i))$$

  for any set $A \in \Theta$, which specifies the conditional distribution of $\theta(i+1)$ given the current state $\theta(i)$. If the chain is independent of *i* it is termed homogeneous. We shall always consider homogeneous chains.
- The n-step transition kernel is, $P^n(\theta(0), A) =_{def} P(\theta(n) \in A|\theta(0))$
- MCMC works by constructing the Markov chain in such a way that,

  $$P^n(\theta(0), A) \approx P(\theta \in A|y)$$

  for some n, irrespective of $\theta(0)$

  Transition kernel – mapping of the transition matrix
- Moreover the approximation improves at each step in that,

  $$\sup_{A \in \Theta} |P^n(\theta(0), A) - P(\theta \in A|y)| \rightarrow 0, \text{ as } n \rightarrow \infty$$

  That is the distribution of the state of the chain, $p(\theta(i))$, converges to the target density, $p(\theta|y)$ as *i* gets "large"

References

- Radford M. Neal, "Bayesian Learning for Neural Networks" https://books.google.com/books?id=LHHrBwAAQBAJ&lpg=PR3&ots=K3CbUN8xZ9&dq=BAYESIAN%20LEARNING%20FOR%20NEURAL%20NETWORK&lr&pg=PP1#v=onepage&q&f=false (Google Scholar Book)
- University of Oxford http://www.stats.ox.ac.uk/~cholmes/Courses/BDA/bda_mcmc.pdf
- University of Lancaster https://www.lancaster.ac.uk/pg/jamest/Group/index.html
- Wikipedia https://en.wikipedia.org/wiki/Markov_chain_Monte_Carlo