Lab 8: Bayesian Analysis using pyjags

+ Reinforcement Learning using gym

Prepared by Paul Tylkin and Vivek HV

CS109B Advanced Topics in Data Science

Pavlos Protopapas and Mark Glickman

Myself!

Vivek HV

Masters in Design Engineering, SEAS & GSD

Bachelors in Aerospace Engineering, IIT Madras

Email: vivekhv@mde.harvard.edu
Feel free to say hi or provide feedback!

Conversation Starters:
Cats, Waffles, GANs, Art, CS, Math, Aerodynamics, Philosophy

Today's agenda

A brief overview of Bayesian Analysis

Introduction to pyjags

Reinforcement Learning using gym

Coding and Q&A!

Bayesian Statistics

You **don’t (and should not) ignore your knowledge** about the state of the world in summarizing conclusions based on data.

Given your **belief about the state of the world, you observe new data** which could possibly update that state.

Bayesian Statistics (and Analysis) lets you encode your prior information which informs your final results

Founded on the **subjective definition of probability** - which is based on your degree of belief that an event will occur - can consider probabilities (and hence uncertainities) of values of unknown parameters

A brief overview of Bayesian Analysis

1. Formulate a **model**

2. Define **prior** distributions of unknown parameters

3. Construct **likelihood function** based on observed data

4. Determine the **posterior distribution**

5. **Summarize** from posterior distribution

An Example

Lets assume we have data collected from a 100 coin flips

What is the probability that it is a fair coin
/How fair is this coin?

An Example

Model:
All coin flips return heads with a probability **`theta`** (and tails with **`1-theta`**)

Prior:
`theta` has a uniformly distributed probability between 0 and 1. Initialize with 0.5

Likelihood:
Construct a likelihood based on observed data (HTHTHT -> theta * (1-theta) * ...)

Posterior:
Posterior is proportional to prior x likelihood or Posterior = c x prior x likelihood

Summarize:
Find mean value for **theta**

How to calculate the posterior distribution?

In a few cases, there is a closed form solutions to the summaries of a posterior distribution.

In most cases (real world models), high dimensionality and complex likelihood functions mean that it is not possible to analytically summarize your posterior distribution.

Thats where Monte Carlo simulations come in. Take a very large sample from the posterior distribution and use sample summaries as approximate actual summaries.

How to calculate the posterior distribution?

What about Markov Chain Monte Carlo?
Sometimes, the posterior densities are too complex/non-standard that even Monte Carlo simulations become hard. Markov chain has a stationary distribution which is the same as the target distribution. Running a markov chain long enough will converge it to the target distribition - in this case the posterior distribution

How to run a MCMC?
Lot of options. We will be using Gibbs Sampler to run multiple markov chains
Run the markov chains for a burn-in period where the different chains start to converge
Sample after burn-in period and summarize from sample

Introduction to pyjags

pyjags in a python interface to **JAGS (Just Another Gibbs Sampler). Gibbs is just one of many different MCMC samplers**

You should have it **already installed on Jupyter Hub!**

If you have been to Lab 1 (or used the config to create your conda environment, you should have pyjags installed)

pyjags does not support Windows :(

Introduction to pyjags

If you are installing it today on your local computer:

Download and install JAGS (Use its default installation location to avoid changing configuration)

```
pip install pyjags
```

If you have a mac you might run into a gcc error, export an env variable required by the installation using:

```
export MACOSX_DEPLOYMENT_TARGET=10.9
```

Let's try doing some Bayesian Analysis!

Reinforcement Learning using gym

If you have not done so already

```
pip install gym
```

That’s all folks!