pipeline: TODO

John Letey[1], Tony E. Wong[1]

ABSTRACT

TODO!!!

[1] University of Colorado at Boulder

Note: If you want to get started immediately with the pipeline *package, start at Appendix A on page 5 or visit the online documentation at* `https://mussles.github.io/pipeline`*. If you are sampling with* pipeline *and having low-acceptance-rate or other issues, there is some advice in Section 4 starting on page 3.*

1. Introduction

Probabilistic data analysis—including Bayesian inference—has transformed scientific research in the past decade. Many of the most significant gains have come from numerical methods for approximate inference, especially Markov chain Monte Carlo (MCMC). For example, many problems in cosmology and astrophysics[1] have directly benefited from MCMC because the models are often expensive to compute, there are many free parameters, and the observations are usually low in signal-to-noise.

2. The Algorithm

The general goal of MCMC algorithms is to draw M samples $\{\Theta_i\}$ from the posterior probability density

$$p(\Theta,\alpha|D) = \frac{1}{Z}\,p(\Theta,\alpha)\,p(D|\Theta,\alpha) \quad , \tag{1}$$

where the prior distribution $p(\Theta,\alpha)$ and the likelihood function $p(D|\Theta,\alpha)$ can be relatively easily (but not necessarily quickly) computed for any particular value of (Θ_i,α_i). The normalization $Z = p(D)$ is independent of Θ and α once we have chosen the form of the generative model. This means that it is possible to sample from $p(\Theta,\alpha|D)$ without computing Z — unless one would like to compare the validity of two different generative models. This is important because Z is generally very expensive to compute.

Once the samples produced by MCMC are available, the marginalized constraints on Θ can be approximated by the histogram of the samples projected into the parameter subspace spanned by Θ. In particular, this implies that the expectation value of a function of the model parameters $f(\Theta)$ is

$$\langle f(\Theta)\rangle = \int p(\Theta|D)\,f(\Theta)\,\mathrm{d}\Theta \approx \frac{1}{M}\sum_{i=1}^{M} f(\Theta_i) \quad . \tag{2}$$

[1] The methods and discussion in this document have general applicability, but we will mostly present examples from astrophysics and cosmology, the fields in which we have most experience

Generating the samples Θ_i is a non-trivial process unless $p(\Theta, \alpha, D)$ is a very specific analytic distribution (for example, a Gaussian). MCMC is a procedure for generating a random walk in the parameter space that, over time, draws a representative set of samples from the distribution. Each point in a Markov chain $X(t_i) = [\Theta_i, \alpha_i]$ depends only on the position of the previous step $X(t_{i-1})$.

The Metropolis-Hastings (M–H) Algorithm The simplest and most commonly used MCMC algorithm is the M–H method (Algorithm 1; MacKay 2003; Gregory 2005; Press *et al.* 2007; Hogg, Bovy & Lang 2010). The iterative procedure is as follows: (1) given a position $X(t)$ sample a proposal position Y from the transition distribution $Q(Y; X(t))$, (2) accept this proposal with probability

$$\min\left(1, \frac{p(Y|D)}{p(X(t)|D)}\,\frac{Q(X(t);Y)}{Q(Y;X(t))}\right) \quad . \tag{3}$$

The transition distribution $Q(Y; X(t))$ is an easy-to-sample probability distribution for the proposal Y given a position $X(t)$. A common parameterization of $Q(Y; X(t))$ is a multivariate Gaussian distribution centered on $X(t)$ with a general covariance tensor that has been tuned for performance. It is worth emphasizing that if this step is accepted $X(t + 1) = Y$; Otherwise, the new position is set to the previous one $X(t + 1) = X(t)$ (in other words, the position $X(t)$ is *repeated in the chain*).

The M–H algorithm converges (as $t \to \infty$) to a stationary set of samples from the distribution but there are many algorithms with faster convergence and varying levels of implementation difficulty. Faster convergence is preferred because of the reduction of computational cost due to the smaller number of likelihood computations necessary to obtain the equivalent level of accuracy. The inverse convergence rate can be measured by the autocorrelation function and more specifically, the integrated autocorrelation time (see Section 3). This quantity is an estimate of the number of steps needed in the chain in order to draw independent samples from the target density. A more efficient chain has a shorter autocorrelation time.

3. Tests

4. Discussion & Tips

REFERENCES

MacKay, D., *Information Theory, Inference, and Learning Algorithms*, Cambridge

Algorithm 1 The procedure for a single Metropolis-Hastings MCMC step.

```
Draw a proposal $Y \sim Q(Y; X(t))$
$q \leftarrow [p(Y)\, Q(X(t); Y)] / [p(X(t))\, Q(Y; X(t))]$        // This line is generally expensive
$r \leftarrow R \sim [0, 1]$
if $r \leq q$ then
    $X(t+1) \leftarrow Y$
else
    $X(t+1) \leftarrow X(t)$
end if
```

University Press, 2003

Gregory, P. C., *Bayesian Logical Data Analysis for the Physical Sciences*, Cambridge University Press, 2005

Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P., *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, 2007

Hogg, D. W., Bovy, J., & Lang, D., 2010, arXiv:1008.4686 [astro-ph.IM]

This preprint was prepared with the AAS LaTeX macros v5.2.

A. Installation

TODO

B. Issues & Contributions

The development of pipeline is being coordinated on GitHub at http://github.com/mussles/pipeline and contributions are welcome. If you encounter any problems with the code, please report them at http://github.com/mussles/pipeline/issues and consider contributing a patch.

C. Online Documentation

To learn more about how to use pipeline in practice, it is best to check out the documentation on the website https://mussles.github.io/pipeline. This page includes the API documentation and many examples of possible work flows.