Bayesian Analysis Toolkit: 1.0 and Beyond

Frederik.Beaujean@lmu.de
Excellence cluster universe, LMU Munich

CHEP 2015, Okinawa

Given data from LHC, what are likely values of masses, cross sections...?
Limits including systematic uncertainties?
Bayes’ theorem

Learning rule

\[ P(\theta|D, M) \propto P(D|\theta, M)P_0(\theta|M) \]

posterior $\propto$ likelihood $\times$ prior
Applying Bayes’ theorem

Integration

- marginalization \( P(\theta_i|D, M) = \int \prod_{j \neq i} d\theta_j P(\theta|D, M) \)
- evidence \( P(D|M) = \int d\theta P(D|\theta, M)P_0(\theta|M) \)
- quadrature \(\rightarrow\) curse of dimensionality
Applying Bayes’ Theorem

Integration

- marginalization: \( P(\theta_i | D, M) = \int \prod_{j \neq i} d\theta_j P(\theta | D, M) \)
- evidence: \( P(D | M) = \int d\theta P(D | \theta, M) P_0(\theta | M) \)
- quadrature \( \rightarrow \) curse of dimensionality

\( \Rightarrow \) need samples from posterior
**Markov Chain Monte Carlo**

**Metropolis Hastings Algorithm**

- One sample per step
- **1** propose move
- **2** accept or stay

\[ P(\theta_1 | D, M) \quad P(\theta_2 | D, M) \]

\[ f(\theta) \rightarrow P(f | D, M) \]
**Markov Chain Monte Carlo**

**Metropolis Hastings Algorithm**

- one sample per step
- propose move
- accept or stay

- marginals
- sample near mode ⇒ seed for optimization
- uncertainty propagation

\[ f(\theta) \rightarrow P(f|D, M) \]
**Motivation**

- reinventing the wheel time waster, error prone
- C++ toolkit to supply algorithms/models \(\Rightarrow\) user can focus on problem
Bayesian Analysis Toolkit

- home page http://mpp.mpg.de/bat
- fork me on https://github.com/bat/bat

Features

- implemented: MCMC (multithreaded), simulated annealing ...
- depends on ROOT: I/O, plots, optimization (Minuit) ...
- optional: roostats, CUBA (integration)
- docs, tutorials, examples ... on web page
\[ P(\theta|D, M) \propto P(D|\theta, M)P_0(\theta|M) \]

**USER DEFINED**
- create model
- read data
\[ P(\theta|D, M) \propto P(D|\theta, M)P_0(\theta|M) \]

**USER DEFINED**
- create model
- read data

**DEFINE**

`MyModel : BCMModel`

- `AddParameter("mu", 0, 1)`
- `LogLikelihood()`
- `LogAPrioriProbability()`
\[ P(\theta|D, M) \propto P(D|\theta, M)P_0(\theta|M) \]

**USER DEFINED**
- create model
- read data

**COMMON TOOLS**
- Normalize()
- FindMode()
- MarginalizeAll()
- PrintAllMarginalized()
- PrintKnowledgeUpdatePlots()

**DEFINE MyModel : BCMModel**
- AddParameter("mu", 0, 1)
- LogLikelihood()
- LogAPrioriProbability()
// define the model
BCMTF m("SingleChannelMTF");
m.AddChannel("channel1");
m.SetData("channel1", hist_data);
m.AddProcess("background", 200., 400.);
m.SetTemplate("channel1", "background",
    hist_background, 1.0);
m.SetPriorGauss("background", 300., 10.);
m.AddProcess("signal", 0., 200.);
m.SetTemplate("channel1", "signal", hist_signal, 1.0);
m.SetPriorConstant("signal");
// run MCMC, find mode, then plot
m.MarginalizeAll();
m.FindMode(m.GetBestFitParameters());
m.PrintKnowledgeUpdatePlots("upd.pdf");
m.PrintCorrelationPlot("corr.pdf");
ATLAS: $Z'$


Belle: dark photon
arXiv:1502.00084

CMS: quantum black hole
arXiv:1501.04198v2

UTFIT: D meson mixing
arXiv:1402.1664

110 citations@inspire, ∼ 100 downloads of v0.9.4.1 since Jan 20, 2015
## History

- first release 2008
- subversion

- one of two main developers left physics
### History
- first release 2008
- subversion
- one of two main developers left physics

### Present
1. better code with git: distributed, code review
**History**

- first release 2008
- subversion

- one of two main developers left physics

---

**Present**

1. better code with git: distributed, code review
2. benefit from github: discuss issues, fork, pull requests...
Lessons in Software Engineering

History

- first release 2008
- subversion
- one of two main developers left physics

Present

1. better code with git: distributed, code review
2. benefit from github: discuss issues, fork, pull requests...
3. write unit tests: refactor code, add features w/o worrying, automatic tests on different platforms

⇒ time investments pay off
**Improvements Under Development**

- ease of use: streamline option setting, building ...
- factorized priors $P(\theta|M) = \prod_i P(\theta_i|M)$
  - ⇒ community extensible
- sharing samples as ROOT files (even w/o the model)
  - ⇒ uncertainty propagation, replotting
- multivariate proposal ⇒ big speed-up in high dimensions
- evidence from MCMC arXiv:1410.7149
  - ⇒ release in summer 2015
Wishlist for the Future

- threads + MPI for tough problems \(\Rightarrow\) rewrite
- interface to script languages: python, mathematica, R . . .
- sampling algorithms: MCMC, Hamiltonian MC, nested sampling, variational Bayes + importance sampling . . .
1 Bayes: random numbers
2 BAT well established
3 more powerful sampling algorithms in BAT 2.0