185.A83 Machine Learning for Health Informatics
2017S, VU, 2.0 h, 3.0 ECTS
Tutorial 02 - 04.04.2017

Tutorial on Probabilistic Programming with PyMC3

florian.endel@tuwien.ac.at
http://hci-kdd.org/machine-learning-for-health-informatics-course

Schedule

- 01. Introduction to Probabilistic Programming
- 02. PyMC3
- 03. linear regression – the Bayesian way
- 04. generalized linear models with PyMC3

- Probabilistic Programming (PP)
  - allows automatic Bayesian inference
  - on complex, user-defined probabilistic models
  - utilizing “Markov chain Monte Carlo” (MCMC) sampling
- PyMC3
  - a PP framework
  - compiles probabilistic programs on-the-fly to C
  - allows model specification in Python code

Salvatier J, Wiecki TV, Fonnesbeck C. (2016) Probabilistic programming in Python using PyMC3. PeerJ Computer Science 2:e55 https://doi.org/10.7717/peerj-cs.55

Properties of Probabilistic Programs

- IS NOT
  - Software that behaves probabilistically
  - General programming language
- IS
  - Toolset for statistical / Bayesian modeling
  - Framework to describe probabilistic models
  - Tool to perform (automatic) inference
  - Closely related to
    graphical models and Bayesian networks
  - Extension to basic language (e.g. PyMC3 for Python)

“does in 50 lines of code what used to take thousands”

Kulkarni, T. D., Kohli, P., Tenenbaum, J. B. & Mansinghka, V. Picture: A probabilistic programming language for scene perception. in Proceedings of the ieee conference on computer vision and pattern recognition 4390–4399 (2015).

- Machine learning algorithms / models often a black box

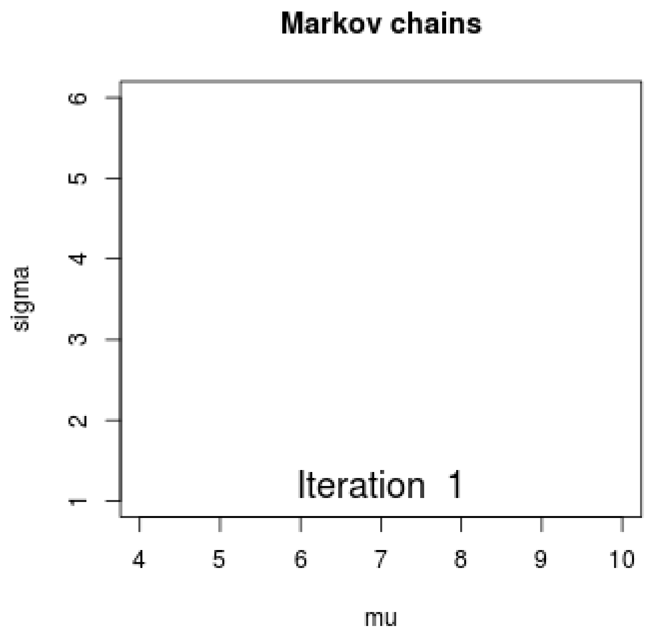
  → PP “open box”
- Simple approach
  1. Define and build model
  2. Automatic inference
  3. Interpretation of results

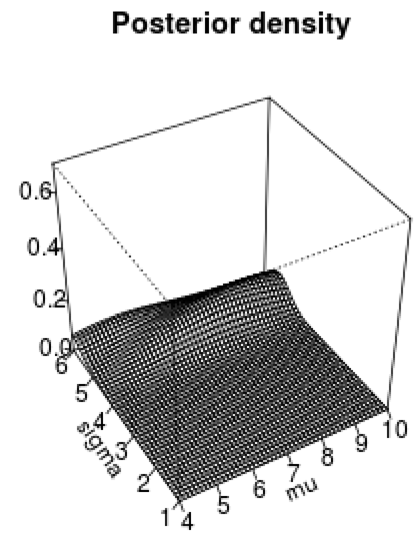
  → not much equations anymore!
- “inference”: guess latent variables based on observations, using e.g. MCMC

Markov chain Monte Carlo (MCMC)

- Markov chain
  - Stochastic process
  - “memoryless” (Markov property)
  - Conditional probability distribution of future states depends only upon the present state
- Sampling from probability distributions
  - State of chain → sample of distribution
  - Quality improves with number of steps
- Class of algorithms / methods
- Numerical approximation of complex integrals

Markov chain Monte Carlo (MCMC)





(animated)

- Metropolis-Hastings: random walk
- Gibbs-sampling: popular, complex, no tuning
- PyMC3
  - No-U-Turn Sampler (NUTS)
  - Hamiltonian Monte Carlo (HMC)
  - Metropolis
  - Slice
  - BinaryMetropolis

- Quantity of interest: θ (theta)
- Prior = probability distribution
  - Uncertainty **before** observation: $\mathrm{p}(\theta)$
  - Belief in absence of data
- Posterior = probability distribution
  - Uncertainty **after** observation X: $\mathrm{p}(\theta|\mathrm{X})$
- Likelihood: $\mathrm{p}(X|\theta)$

$$Posterior \propto Likelihood \times Prior$$

$$p(\theta|x) = \frac{p(x|\theta)p(\theta)}{p(x)}$$

- Calculating posterior from observation and prior = updating beliefs

Coin toss example

- Python 3 package / framework
- Probabilistic machine learning
- **specification** and **fitting** of Bayesian models
- Inference by MCMC & variational fitting algorithms
- Performance enhancements:
  cross-compilation to C
  (Python numerical computation package “Theano”)
- Accessible, natural syntax
- Various capabilities: GPU computing, sampling backends, object-oriented, extendable design

Live presentation

- PyMC3 syntax introduction
- linear regression – the Bayesian way
- generalized linear models with PyMC3

Thank you!

- What is probabilistic programming? What type of problems can be solved?
- What are the typical steps in a probabilistic program?
- What is inference?
- What is PyMC3?
- What is the posterior distribution?

- Main sources
  - Salvatier J, Wiecki TV, Fonnesbeck C. (2016) Probabilistic programming in Python using PyMC3. PeerJ Computer Science 2:e55 https://doi.org/10.7717/peerj-cs.55
  - Davidson-Pilon, C. Bayesian Methods for Hackers: Probabilistic Programming and Bayesian Inference. (Addison-Wesley Professional, 2015).
  - PyMC3's documentation http://pymc-devs.github.io/pymc3/index.html