10/30/19

Physics 8805: Learning from Data: Lecture 19

Notebooks for today:
- mini-projects/model-selection_mini-project-IIb_How_many_lines.ipynb
  - Mini-project IIb due Sunday 11/10.
- mcmc-sampling/PyMC3_intro.ipynb and PyMC3_docs_getting_started.ipynb

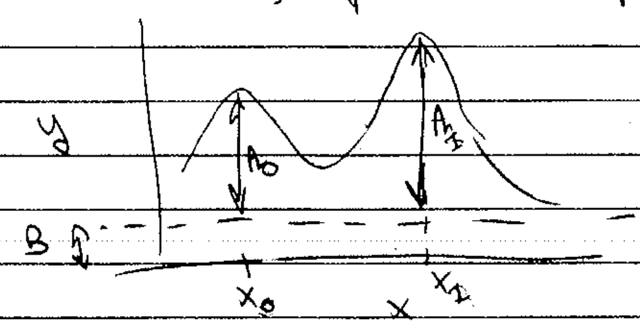
- demo-Gaussian Processes.ipynb
- GaussianProcesses.ipynb

Quick comment on project: A notebook relevant to your research interests using Bayesian tools.

Overview of Mini-Project IIb: How many lines?

- The problem is adapted from Sivia, section 4.2, which is available on Carmen.
- Basic problem: we are given a noisy spectrum with some number of lines. We want to use parameter estimation and model selection (with parallel tempering) to determine what we can about the peaks (model selection: how many peaks; parameter estimation: what are the positions and amplitudes of the peaks).

- The model is a sum of Gaussians with known width but unknown number, amplitudes, and positions, plus a constant background.



If M lines, then 2M+1 model parameters $\vec{\alpha} = (A_0, x_0, A_1, x_1, \ldots, B)$

- We add Gaussian noise with known σ_{exp}.
- Formulas are in the notebook.

10/30/19

There are 5 required subtasks, plus a bonus subtask and one to get a plus.

1. Formulate the problem of how many lines and what are the model parameters in Bayesian language.
   - This amounts to working through Sivia 4.2.

2. Derive an approximation for the posterior probability.
   - Again, Sivia 4.2 has intermediate steps, but try doing it yourself. Be careful of the M!
   - Where is the Ockham factor?
   - Does this assume the background B is known?

3. Optional: Numerical implementation.

4. Generate data ⇒ just need to look at fluctuations and impact of width and noise.

5. Parameter estimation with emcee.
   - Show what happens with numpeaks = 2.
   - Your job: explain!

6. Main part: parallel tempering results for evidence.
   - Explain the behavior (clear maximum or saturation). Connect to toy model results.

7. Repeat for another situation (different # peaks, width, noise.

10/30/19

PyMC3 Take-aways

- Clearly set-up on Windows machines can be non-trivial. A helpful blog entry has been added to Carmen (under 6. MCMC Sampling II), with particular details about the compiler.

- We'll just give an overview of some features here, because we need to move along. You are encouraged to follow-up on your own.

- Take-away features from PyMC3_intro.ipynb ← use shift-tab-tab
  - many samplers available — sample step can mix different samplers
  - traceplot ← use shift-tab-tab ⇒ histogram and sampled values
  - built-in distributions like pm.Normal and pm.HalfNormal
  - Gelman-Rubin diagnostic available (recall want this close to 1 (≪1.2)
    - Try with trace_2_samplers (also plot_posterior)
  - Example from Rob Hicks
    - find_MAP is discouraged for NUTS (but Metropolis used)
    - autocorrplot
    - Geweke plot — test for stationarity
    - two-parameter model

- Looking at PyMC3_docs_getting_started.ipynb (may be good or bad!)
  - note comment about Theano
  - consider the example: make data, encode the model ≈ statistical notation
  - line-by-line explanation ⇒ cf. $\alpha \sim N(0,100)$, $\beta_i \sim N(0,100)$, $\sigma \sim |N(0,1)|$ then $Y \sim N(\mu, \sigma^2)$, $\mu = \alpha + \beta_1 X_1 + \beta_2 X_2$ (linear model)
    This is just saying $y_{exp} = y_{th} + \delta y_{exp}$ with $y_{exp} \to Y$, $y_{th} \to \mu$ and δy_{exp} Gaussian noise with σ
  - Note the background on the context model
  - summary function
  - Other examples as prototypes.

10/30/19

Gaussian Processes

- Let's step through demo-GaussianProcesses.ipynb

- A stochastic process is a collection of random variables (RVs) indexed by time or space. E.g. at each time there is a random variable or at each space point.

- A Gaussian process is a stochastic process with particular relationships (correlations!) between the RVs. In particular any finite subset (say at x_1, x_2, x_3) has a multivariate normal distribution.
  - cf the definitions at the beginning of the notebook
  - A GP is the natural generalization of multivariate random variables to infinite (countably or continuous) index sets.
  - They look like random functions, but with characteristic degrees of smoothness, correlation lengths, and range.

- Multivariate Gaussian distribution in general is

$$p(\vec{x}\mid\vec{\mu},\Sigma) = \frac{1}{\sqrt{\det(2\pi\Sigma)}}\, e^{-\frac{1}{2}(\vec{x}-\vec{\mu})^T\Sigma^{-1}(\vec{x}-\vec{\mu})}$$

For bivariate case:

$$\vec{\mu} = \begin{pmatrix}\mu_x \\ \mu_y\end{pmatrix} \quad \text{and} \quad \Sigma = \begin{pmatrix}\sigma_x^2 & \rho\sigma_x\sigma_y \\ \rho\sigma_x\sigma_y & \sigma_y^2\end{pmatrix} \quad \text{with } 0<\rho^2<1 \ (\Sigma \text{ positive definite})$$

- In notebook the case $\sigma_x = \sigma_y = \sigma$ is seen to be an ellipse.

- Think of the bivariate case with strong correlations as belonging to two points close together $\Rightarrow$ smoothness of function tells us that the lines in the plot in the notebook should be closer to flat (small slope).

10/30/19

- Kernels: These are the covariance functions that, given two points in the N-dimensional space: $\vec{x}_1$ and $\vec{x}_2$, return the covariance between $\vec{x}_1$ and $\vec{x}_2$.

- Consider these to be one-dimensional, so we have x_1 and x_2. Then

$$K_{RBF}(x_1, x_2) = \sigma^2 e^{-(x_1-x_2)^2/2l^2}$$

Compare to $\Sigma = \sigma^2 \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$

The diagonals are $x_1 = x_2$, while $\rho = e^{-(x_1-x_2)^2/2l^2}$. So when x_1 and x_2 are close compared to l then the values of the sample at x_1 and x_2 are highly correlated (so they will be very close). When x_1 and x_2 are far apart, $\rho \to 0$ and they become independent $\Rightarrow$ l plays the role of a correlation length.

so $\rho \simeq 1 \Rightarrow$

- Look at the examples for different l. What does the RBF Cov. Matrix plot show? This is the covariance matrix!

- GP models for regression
  - First example of using GPs $\Rightarrow$ train on some points, predict elsewhere with error bands.

- No-core shell model $\hbar\omega$ dependence. More details next time!