Project 3 CompSci program, deadline May 31

CompSci program

Department of Physics, University of Oslo, Norway

Bayesian analysis / Gaussian processes regression

For this project you can choose whether to focus on (1) *Bayesian parameter estimation and model comparison*, or (2) *Gaussian process (GP) regression.* If you want you are of course welcome to study both topics — since GP regression is a Bayesian approach there is clearly much overlap here.

We encourage you to find (or generate/simulate!) a dataset that is relevant to your own work, or that you simply find interesting, and use this for your project. However, we will also provide some example datasets below.

1) Bayesian parameter estimation and model comparison

Here the main goal is to perform a Bayesian data analysis (both parameter estimation and model comparison) using different parameterised models, and study how different modelling assumptions affect the final result. Some key questions you should investigate and/or comment on are:

- What is the likelihood function you use, and why?
- Prior dependence: how sensitive are your analysis results to changes in your priors?
- Which (if any) of the models you consider is favored in Bayesian model comparison?[1] How are your model comparison results related to model complexity?
- Are your posteriors multimodal?
- Would a max-likelihood approach (non-Bayesian) to parameter estimation yield similar or different results in your case?
- How can the results be best presented and summarised?

[1] Also, note that you can of course use the model comparison framework to compare any sorts of hypotheses you can evaluate posterior beliefs for — it doesn't have to be a comparison of posterior probabilities on the form $P(\text{model}|\text{data})$. For instance, you can consider comparing separate regions within the parameter space of a single model, if this seems interesting for your problem.

- Any numerical/computational challenges?

The parameterised models you consider should have at least a few (≥ 3) free parameters, to make sure the problem isn't completely trivial. You can use any numerical tool(s) you prefer, but we recommend trying out the pymultinest package (Python interface to the MultiNest package):

- Documentation: `http://johannesbuchner.github.io/PyMultiNest/` and `https://github.com/JohannesBuchner/MultiNest`
- Tutorial: `http://johannesbuchner.github.io/pymultinest-tutorial/example2.html`
- We recommend not relying on the scripts that pymultinest provide for the final analysis of your posterior results, but to rather use the set of generated posterior samples toegether with your own analysis/visualisation code.
- The documentation above contains information on the content of all the output files generated by MultiNest. The most important is the post_equal_weights.dat file.
- *Note:* The function referred to in pymultinest examples as `prior` is not a function representing the actual prior pdf, but rather a function where you must *transform* samples drawn uniformly from a unit hypercube, to samples from your chosen prior.
- If needed, we can provide an annotated simple example of how to use pymultinest.

Possible example dataset

We recommend finding a dataset that interests you. But as an alternative, we provide a simple example dataset here:
`https://www.dropbox.com/s/tdhb7bwjfjkuha0/example_data.txt?dl=0`
We assume that these are data from an experiment where the *background* component of the data is expected to follow a falling spectrum, while any interesting *signals* in the data can appear as peaks on top of this falling background. Common question when faced with such a dataset are: Is there evidence for any signal peaks in the data? If so, where are these signal peaks and what are the parameters describing them? How many such peaks are there?

2) Gaussian process regression

Here the main goal is to explore various aspects of GP regression. Here are some questions you can investigate and/or comment on:

- How does your regression results depend on your choice of covariance function (kernel), i.e. your choice of prior on function space?

- Is a “standard” few-parameter covariance function sufficient? Can you get better / more reasonable results by constructing a more proble-specific covariance function?

- Are your GP models “overconfident”, i.e. are there cases where the regression uncertainty is severely underestimated?

- Is there noise in the data, and what impact does this have on your results? And what exactly is your GP giving predictions for — a true (noiseless) function or noisy data points?

- What approach is used for training the GP(s)? Why this approach?

- How do your GPs tackle the limit of large number of data points (if available)? What approaches could you take to overcome problems with large datasets?

- Can your GPs extrapolate beyond the region of training data? Should they be able to?

We again advice you to find/generate datasets that are relevant for your own field of research. An alternative is to use some of the test functions given here `https://www.sfu.ca/~ssurjano` (e.g. some of the functions from the *emulation/prediction* category), and generate your own test and training datasets from these functions. While one-dimensional test functions are useful to build intuition and provide easy-to-visualise examples, we advice you to explore test functions with many-dimensional input spaces to make the problem more challenging/realistic.

You are free to use any numerical GP software you prefer, or write your own GP code. A fairly user-friendly option is the scikit-learn package:
`https://scikit-learn.org/stable/modules/gaussian_process.html`