Final Project

CM50268 Bayesian Machine Learning 2021-2022

April 2022

1 Overview

- **Marks**: 70% of overall unit
- **Group work**: This year you will work in groups. You will be randomly allocated to a group, which may contain 3 - 4 members. Members in the same group can submit a same Jupyter Notebook file, but should submit an individual report
- **Individual contribution**: Upon completion of your group work, please have a discussion on each member's percentage of contribution to each of the following three items:
  - Coding (100% in total)
  - Theoretical / mathematical derivation (100% in total)
  - Useful thoughts/opinions leading to group solutions (100% in total)

  Please indicate your assigned percentage (between $0\%-100\%$) on each of the above items at the beginning of your individual report. Note that your grade of the final project is mainly based on report + code, the contribution percentage will not affect the marks awarded to most of the students. It is mainly to penalise members that have zero (or almost zero) contribution to ALL three items, and award extra bonus to members that have significant contribution to any of the three items (given your group solutions are correct)
- **Release date**: April 8, 2022
- **Submission due**: 8pm, May 10, 2022
- **Submission type**: project report (no more than 1500 words) + Jupyter Notebook code
- **Plagiarism**: both the code and the report will be cross checked by the plagiarism detectors. Submissions with high similarity score will be manually checked, and the confirmed pairs will be reported

The final piece of assessed coursework involves the evaluation of Bayesian modelling methods on a real multivariate regression task. The guiding objectives are to derive a good predictor for data derived from an "energy efficiency" data set, and to estimate which of the input variables are relevant for prediction.

In particular, the exercise focuses on approximating (and averaging over) posterior distributions using the Hamiltonian Monte Carlo (HMC), Variational Inference (VI), and Gaussian Processes

(GP) methods (example implementations of HMC and GP are supplied). Your experiments will be based mainly on existing (or supplied) analytic code and techniques you have already learnt. You will of course need to write all the relevant code in the relevant cells of the Jupyter Notebook to process the data, apply the methods appropriately, extend them in places, and ultimately calculate and output the necessary results.

A key part of the assessment is to compile, present and critique all those results effectively within an "individual project report" document. For this exercise, your code in the Jupyter Notebook will also be auto-marked. Marks will be awarded based on the report, code, and (potentially) group contributions.

2 Data

You will be analysing the "Energy efficiency" data set, originally from the University of Oxford, and now made available at the UCI Machine Learning Repository.

This multivariate data set contains 768 examples and comprises 1 constant bias and 8 input variables as $x_0, x_1, x_2, \cdots, x_8$, where x_0 is the constant bias and others presenting some basic architectural parameters for buildings (e.g. "Roof Area" and "Glazing Area") with the intention of predicting a tenth target variable y, the required "Heating Load". This can be considered a real-value variable, suitable for standard regression modelling (with the usual Gaussian noise model).

The data has been pre-processed and equally split into two specific data sets for your use:

- Training set: ee-train.csv
- Test set: ee-test.csv

Both csv files have a header row (labelling each variable), with the first 9 columns as the input X, which contains 1 constant bias (first column as x_0) and 8 input variables ($x_1, x_2, \cdots, x_8$) and the final column being the target variable (y "Heating Load"). For the purposes of modelling, you may find it useful to standardise the other inputs to mean zero and standard deviation one.

For model training, only ee-train.csv should be used, and ee-test.csv should be reserved purely for assessing model performance.

3 Summary of objectives and mark scheme

The principal objectives are focused on the predictive modelling of the energy efficiency data and take the form of a series of (sequential) sub-tasks. These are, in summary:

1. Undertake an initial exploratory analysis of the training data and summarise. [4 marks]
2. Apply the standard Bayesian linear regression model (Lecture 3), then:
   (a) Using Type-II maximum likelihood (Lecture 4) to estimate "most probable" values for hyper-parameters [4 marks]
   (b) Using Variational Inference (Lecture 9) with simple 'Mean-Field Theory' factorisation (Lecture 10) to estimate "most probable" values for the hyper-parameters [6 marks]

3. Familiarise yourself with the use of the Hamiltonian Monte Carlo (HMC) algorithm (Lecture 8), initially verifying the HMC implementation on a **standard 2D Gaussian example** [4 marks]

4. Apply HMC to sample weights and the hyper-parameters of the standard Bayesian regression model [6 marks]

5. Apply GP (Lecture 11 & 12) to sample weights (but not hyper-parameters) of the standard Bayesian regression model [6 marks]

6. Coding: fill all required cells in the provided Jupyter Notebook file `BML_Final_Project.ipynb`. See the notebook for breakdown marks of the coding exercises [25 marks]

7. Document all results in a coherent and structured individual project report, assessment will be based on

   - Overall quality of the report [6 marks]
   - Clear and informative presentation of the figures and/or numerical tables [4 marks]
   - Some degree of critical review and analysis of the project [5 marks]

4 Tasks in detail

4.1 Exploratory analysis [4 marks]

Undertake an initial exploratory analysis of the given data variables and summarise appropriately. This need not be particularly extensive, but it should demonstrate that you have undertaken some degree of "due diligence" with respect to the data set. In particular, you should focus on identifying which of the input variables (if any) might be expected to be useful for predicting heating load, and which might be irrelevant. You may also wish to comment on the apparent linearity, or otherwise, of the problem.

As the final part of the exploratory analysis, establish a predictive "baseline" by fitting a linear model to the training set by least-squares, and assessing its prediction accuracy on both train and test sets.

This section of your report should include:

- Initial observations as to the difficulty of the task, its linearity etc.
- Your comment on the likely relevance of the variables for predicting "Heating Load"
- Appropriate graphs/charts as evidence to support the above
- Detail of the accuracy of the least-squares linear model, on both train and test sets, in terms of mean absolute error (MAE).

4.2 Bayesian linear regression [10 marks]

Consider a standard linear regression model with unknown coefficient set $\mathbf{w}$. Some modelling guideline/hints:

- $\mathbf{w}$ is assumed to have a Gaussian prior $\mathcal{N}(0, \sigma_{\mathbf{w}}^2)$, and we define the precision of prior as $\alpha = 1/\sigma_{\mathbf{w}}^2$
- The problem can be modelled using an additive Gaussian noise $\mathcal{N}(0, \sigma_\epsilon^2)$, and we define the precision of noise as $\beta = 1/\sigma_\epsilon^2$
- Now we have the unknown hyper-parameter set $\theta = (\sigma_\epsilon^2, \sigma_{\mathbf{w}}^2) = (\alpha, \beta)$
- If we denote the observation data as D, the posterior we want to estimate can be written as $p(\mathbf{w}, \theta|D)$.

(a) Type-II maximum likelihood [4 marks]. Please follow the Type-II maximum likelihood methodology outlined in Lecture 4 (and maybe also Coursework 2, task 1b) to estimate "most probable" values for hyper-parameters θ

(b) Variational Inference [6 marks]. Please use Variational Inference (Lecture 9) with simple 'Mean-Field Theory' factorisation (Lecture 10) to estimate "most probable" values for the hyper-parameters θ. Note that, although looks similar, the problem described here is not exactly the same as the one described in Example 2 of Lecture 10. You should only follow the VI methodology (but not that example) described in Lecture 10 to solve the problem

This section (both task (a) and (b)) of your report should at least include:

- A visualisation (e.g. using plt.contourf) of the posterior distribution
- The 'most probable' values of the parameters of interest, ideally marked on the visualisation
- MAE of your model predictions on the test set.

4.3 Verify HMC on a standard 2D Gaussian example [4 marks]

Code to implement HMC is supplied in the module `hmc_Lab.py` and there is a simple demonstration of its usage in the notebook `demo_hmc.ipynb` along with a one page instruction `hmc_Spec.pdf`. You should try and complete this exercise before this exercise.

Apply the `hmc_Lab.sample` function to generate samples from a simple standard Gaussian in two dimensions. This is an artificial challenge, of course, as we can sample directly from such a Gaussian. However, it enables us to check our code is working and visualise its performance.

To apply HMC, you will need to write appropriate functions `energy_func` and `energy_grad` to pass to sample. The former is the negative log probability of the 2 dimensional variables under the Gaussian (easily obtained directly from `scipy.stats`). The latter is the gradient of that function, returning an array containing the partial derivatives with respect to two dimensional variables (see the notebook demo for an example). You will need to work those derivatives out and code the `energy_grad` function explicitly in the provided Jupyter Notebook file.

In `demo_hmc.ipynb`, a value of `L` of 25 should be fine for this simple distribution, but you will need to adjust the `eps` parameter appropriately following the guideline in `hmc_Spec.pdf`.

It is highly recommended that you always test the consistency of your functions by setting `checkgrad=True` when calling `sample`. This will compute an estimate of the gradient using numerical techniques, and compare it with your analytical calculation. It is a numerical approximation, so there will always be small differences, but anything large (one part in 10^6 or greater perhaps) may suggest an error in your working or your code.

This section of your report, you should:

- Write down the mathematical formula of the standard 2D Gaussian example, please make sure the variables are consistent to the rest of your report
- Verify and demonstrate (with appropriate figures or numerical tables) that your HMC works as expected
- Report the values of `R`, `L` and `eps` that you used to obtain your presented results
- Report your designed functions `energy_func` and `energy_grad`.

4.4 Apply HMC to the Linear Regression Model [6 marks]

Apply HMC to obtain samples from the joint posterior over linear regression coefficients (weights) $\mathbf{w}$, hyperparameter set θ for the linear regression model on the energy efficiency training data.

This section of your report, you should:

- Demonstrate (with appropriate figures or numerical tables) that your HMC works as expected
- Report the values of `R`, `L` and `eps` that you used to obtain your presented results
- Report the optimal values of the unknown terms
- Evaluate MAE of prediction in test set
- Provide your insight and analysis supported by figures and/or numerical tables. You could also compare the different algorithms you have used so far in this project.

4.5 Apply GP to the Linear Regression Model [6 marks]

Example implementation of GP is supplied in the modules `demo_GP_task.ipynb` and `demo_GP_solution.ipynb`, one is the exercise and the other is the solution. You should try and complete this exercise before this task.

In this task, you need to apply GP to obtain joint posterior over linear regression coefficients (weights) $\mathbf{w}$. You don't need to estimate the hyperparameter set θ in this task.

This section of your report, you should:

- Demonstrate (with appropriate figures or numerical tables) that your GP works as expected

- Report the optimal kernel function(s) and their hyper-parameters you use to achieve the results results
- Evaluate MAE of prediction in test set
- Provide your insight and analysis supported by figures and/or numerical tables. You could also compare the different algorithms you have used so far in this project.

4.6 Coding exercises [25 marks]

The corresponding coding exercises can be found in the module `BML_Final_Project.ipynb`. You need to discuss with your group peers and fill all required cells. See the notebook for breakdown marks of the coding tasks.

4.7 Your overall report quality [15 marks]

The mark scheme of your overall report quality has been listed in the last item of section 3. In your report summary, please write a generous paragraph (or two) summarising what you have learned in respect of both:

- The specific data set under study, the suitability of linear models for prediction and, in particular, the relevance of the different variables
- Your observations on the effectiveness of the various methods applied in this project.

You should submit both your report and the code (ideally jupyter notebook) via Moodle by the specified due date. The report should be self-contained with sufficient information to allow any reader to replicate the experiments, though be concise and remember the 1500 word limit. (Note that graphs and listings of code do not contribute to the limit. Use of multiple explanatory graphs is indeed encouraged!)

- Please follow the instructed order of objectives in Section 3 to write your report
- You should make use of additional text to make clear what is being presented. Do not assume that the reader has the task specification (i.e. this document) available to cross-reference.
- Please be as clear as possible in your presentation! With the best will in the world, marks cannot be awarded for content that cannot be understood :).