Coursework **Fall 2021**
Machine Learning (B555) **Roni Khardon**
Department of Computer Science, IU Bloomington

Programming Project 4

This assignment is due by Wednesday 11/17, 11:59pm via Canvas.

You can write your code in any programming language so long as we are able to test it on SICE servers. We plan to run some or all or submitted code for further testing and validation.

Overview: Gaussian Processes

In this programming project we will be working with Gaussian processes using the RBF kernel and the polynomial kernel with degree 1 (i.e., linear) kernel.

Data

Data for this assignment is provided in a zip file `pp4data.zip` on Canvas. We have 4 datasets, `crime, artsmall, housing, 1D` whose files per train/test and features/labels are organized as in project 2.

The first 2 datasets are drawn from project 2 but we normalized the features and labels and reduced the size of the train and test sets to reduce the run time for the project. `1D` is a 1 dimensional artificial dataset where we can visualize the predictions.

Implementing GP

Mean and Covariance functions: We will use a zero mean function. The covariance functions we use are exactly as in lecture slides. $C(x_1, x_2) = \delta(x_1, x_2)\frac{1}{\beta} + \frac{1}{\alpha}k(x_1, x_2)$, where $k(x_1, x_2) = (x_1^\top x_2) + 1$ for the linear kernel and $k(x_1, x_2) = e^{-0.5\|x_1 - x_2\|^2/s^2}$ for the square exponential (a.k.a. RBF) kernel. In matrix form, this looks like $C(X, X) = \frac{1}{\beta}I + \frac{1}{\alpha}K(X, X)$ and when examples in X are distinct from examples in Z we get $C(X, Z) = \frac{1}{\alpha}K(X, Z)$.

Predictive distribution, derivatives, and model selection: The equations for the log evidence, predictive distribution and derivatives w.r.t. α, β, s are given in the slides. As suggested in the slides, for model selection in this project you should perform gradient ascent on $\ln \alpha, \ln \beta, \ln s$.

Note: implementing the GP equations normally requires some care to avoid numerical instability in calculations (for example using the Cholesky decomposition to avoid computing inverses). However, for this project a direct implementation of the equations works fine.

Optimization: Initialize $\alpha = \beta = 1$ and $s = 5$, except for dataset `1D` where s should be initialized to 0.1 (of course s is only relevant when using the RBF kernel). For model selection use gradient ascent on the log values of hyperparameters with learning rate of $\eta = 0.01$ for up to 100 iterations. You should track the value of the log evidence during optimization, and can stop the optimization early if the log evidence does not change "too much". Specifically let a be the log evidence before the update, b be the value after the update and $c = (b - a)/|a|$ be the relative change. Then we can stop the optimization if $c \leq 10^{-5}$.

Notes: It is a good idea for debugging to make sure that the log evidence is increasing, i.e., $b > a$, with each gradient step. Setting the learning rate is tricky in general but using the value specified works well for our datasets. Similarly, the results are somewhat sensitive to the initial value of s but the values above yield good results for our datasets.

Evaluation

We will use MNLL (mean negative log likelihood) on the test set. The negative log likelihood on example i is NLL $= -\ln \mathcal{N}(t_i|m_i, v_i)$ where m_i, v_i are the mean and variance of the predictive distribution. Then we have MNLL $= \frac{1}{N}\sum_i \ln \mathcal{N}(t_i|m_i, v_i)$. In addition, to compare to results with Bayesian Linear Regression we calculate the test set MSE: MSE $= \frac{1}{N}\sum_i (m_i - t_i)^2$ (where in project 2 we had $m_i = m_N^\top \phi(x_i)$).

Visualizing performance on the 1D dataset: For dataset `1D` run the algorithm until it stops and record the final learned function. Then visualize the results as follows. The true function for this dataset is: if $x > 1.5$ then: $f(x) = -1$; if $x < -1.5$ then: $f(x) = 1$; otherwise $f(x) = sin(6 * x)$. Plot the true function, and the mean of the predictive distribution with $\pm$ 2 standard deviations around the mean, where x is in the range $[-3, 3]$. This should be done with both the RBF kernel and the linear kernel, i.e., provide 2 such plots.

Performance as a function of iterations: Run the algorithm, on each of the 4 datasets, with both the RBF and linear kernels, and record the test set MNLL performance as a function of the number of training iterations. To save in compute time (since evaluation of GPs is time consuming) evaluate the prediction every 10 iterations, and after the last iteration (for example, if the algorithm stops at iteration 33 you will have evaluations at 0,10,20,30,33 iterations). Then plot the MNLL as a function of the number of training iterations. Thus we expect 8 plots from 4 datasets $*$ 2 kernels.

Comparison to Bayesian Linear Regression: In addition to the above, record the values of α, β and test set MSE after the last iteration (i.e., when the algorithms stops). Tabulate these results and compare them to the results of Bayesian Linear Regression, either by running your code from project 2, or taken from the table on the right.

dataset	MSE	α	β
crime	0.5	357.5	2.6
artsmall	0.716	141.4	4.23
housing	0.288	20.4	4.0
1D	0.39	7.5	1.9

Discuss the results: With all these results recorded, what can you observe w.r.t. the performance of the algorithms? Are the BLR and GP with linear kernel behaving similarly w.r.t. α, β, MSE as expected?[1] How does the performance of GP compare when changing RBF vs. linear kernel? What are potential advantages or disadvantages of each method?

[1] Note that there is a small difference in feature space because of the addition of +1 in the kernel version so they may not be identical. But this does not affect the results significantly.

Submission

Please submit two separate items via Canvas:

(1) A zip file `pp4.zip` with all your work and the report. The zip file should include: (1a) Please write a report on the experiments, include all plots and results, and your conclusions as requested above. Prepare a PDF file with this report. (1b) Your code for the assignment, including a README file that explains how to run it. When run your code should produce all the results and plots as requested above. Your code should assume that the data files will have names as specified above and will reside in sub-directory pp4data/ of the directory where the code is executed. We will read your code as part of the grading – please make sure the code is well structured and easy to follow (i.e., document it as needed). This portion can be a single file or multiple files.

(2) One PDF "printout" of all contents in 1a,1b: call this `YourName-pp4-everything.pdf`. One PDF file which includes the report, a printout of the code and the README file. We will use this file as a primary point for reading your submission and providing feedback so please include anything pertinent here.

Grading

Your assignment will be graded based on (1) the clarity of the code, (2) its correctness, (3) the presentation and discussion of the results, (4) our ability to run the code on SICE servers.