Milestone 2: Gaussian Processes

Data set :

In this milestone project, we implement Gaussian Processes using two types of Kernels—RBF kernel and Matérn Kernel—to classify the Wine quality(score between 0 and 10) , given its chemical properties:

Features (based on physicochemical tests):

1 - fixed acidity

2 - volatile acidity

3 - citric acid

4 - residual sugar

5 - chlorides

6 - free sulfur dioxide

7 - total sulfur dioxide

8 - density

9 - pH

10 - sulphates

11 - alcohol

We used a single dataset ‘winequality-red.csv’ to split it into train:test =7:3 to produce this report.

Some properties of Matérn Kernel:

* Matérn kernel is a generalization of the RBF and the absolute exponential kernel parameterized by an additional parameter nu.
* The smaller nu, the less smooth the approximated function is. For nu=inf, the kernel becomes equivalent to the RBF kernel and for nu=0.5 to the absolute exponential kernel.
* Important intermediate values are nu=1.5 (once differentiable functions) and nu=2.5 (twice differentiable functions).

**Reference**: Rasmussen and Williams 2006, pp84 was referred, for details regarding the different variants of the Matern kernel.

We can compare various Kernels’ *Negative log marginal likelihood* to check how well the model performs. Likelihood is proportional to the probability of observing the data given the parameter estimates and our models. Two Kernel models’ likelihoods can be compared since they are nested(setting *nu ()* to infinite in Matérn kernel will result in an RBF kernel). In other words, model with lower negative log marginal likelihood is a more probable model.

In **“1-vs-1” classification of multivariate GP,** one binary Gaussian Process classifier is fitted for each pair of classes, which is trained to separate these two classes. Basic steps are:

1. Create all possible pairs of classes, which is binary classification problems, addressing each pair of classes.
2. Use weighted majority vote to assign each final label

GP with **RBF Kernel**

Multiclass (One-vs-One) Gaussian Process Train Accuracy :: 0.965

Multiclass (One-vs-One) Gaussian Process Test Accuracy :: 0.546

Negative Log Likelihood: -716.260

GP with **Matern Kernel**

Multiclass (One-vs-All) Gaussian Process Train Accuracy :: 0.716

Multiclass (One-vs-All) Gaussian Process Test Accuracy :: 0.550

Negative Log Likelihood: -704.592

In “**1-vs-All**” **classification of multivariate GP**, one binary Gaussian Process classifier is fitted for each class, which is trained to separate this class from the rest. Basic steps are:

1. Train K binary classifiers
2. This will get k classifiers. For each point xi, assign it the maximum possible label (i.e. the highest value returnable by any of the classifiers) using majority vote.

Thought on **1-vs-all** model: we can probably give negative examples smaller weights, since there will be excessively more negative values than positive values.

We trained/tested our data using two popular kernels, with

GP with **RBF Kernel**

Multiclass (One-vs-All) Gaussian Process Train Accuracy :: 0.968

Multiclass (One-vs-All) Gaussian Process Test Accuracy :: 0.546

Negative Log Likelihood: -716.260

GP with **Matern Kernel**

Multiclass (One-vs-All) Gaussian Process Train Accuracy :: 0.770

Multiclass (One-vs-All) Gaussian Process Test Accuracy :: 0.627

Negative Log Likelihood: -704.592

We now can increase the length scale of the RBF kernel to check the change in negative log marginal likelihood.

GP with **RBF Kernel**

Multiclass (One-vs-All) Gaussian Process Train Accuracy :: 0.702

Multiclass (One-vs-All) Gaussian Process Test Accuracy :: 0.577

Negative Log Likelihood: -593.513

By observing the accuracy and negative log likelihood values, we conclude that we see a better fit to the data can be achieved by parameterizing the kernel with small length scale with the RBF kernel. We see a low negative log marginal likelihood value, and we hypothesize that it depends on the complexity of our dataset and noise level, and carefully confirm that the value is in a feasible range.

By observing the performance of both **‘1-vs-1’** model and ‘**1-vs-all’** classification models, we see that they both produce very similar accuracy and log likelihood results. However, when we measure the computation time using *time* library:

GP with **RBF Kernel**

Multiclass (1-vs-All) computation time :: 7.149 (sec)

Multiclass (1-vs-1) computation time :: 2.337 (sec)

GP with **Matern Kernel**

Multiclass (1-vs-All) computation time :: 82.009 (sec)

Multiclass (1-vs-1) computation time :: 24.320 (sec)

We can easily see that **1-vs-1** classification is almost 3 times faster than **1-vs-all** method, while both classification methods yields similar numerical accuracy and log likelihood values.

Additionally, by changing the *nu ()* parameter in Matérn Kernel, we easily can apply different kernels for our model.

GP with **Matern Kernel**

Multiclass (One-vs-All) Gaussian Process Train Accuracy :: 0.755

Multiclass (One-vs-All) Gaussian Process Test Accuracy :: 0.569

Negative Log Likelihood: -559.653

Here, by controlling the smoothness parameter *nu ()*, we can fit the training data little bit better, but our negative log likelihood value increases as a tradeoff.

When performing 10-fold Cross Validation on all 4 classification models:

RBF(1-vs-all, ) Accuracy(Mean CV): **0.46 (+/- 0.07)**

RBF(1-vs-1, ) Accuracy(Mean CV): **0.47 (+/- 0.06)**

Matérn(1-vs-all, , ) Accuracy(Mean CV): **0.58 (+/- 0.13)**

Matérn(1-vs-1, , ) Accuracy(Mean CV): **0.54 (+/- 0.09)**

The CV computation time was 354.296
seconds for Matérn(1-vs-all, , ). In the case of Matérn Kernel(1-vs-all), it took 1611.358 seconds to run.

By comparing Test Accuracy and Cross Validation results, we can conclude that Matérn kernel performs slightly better than RBF kernel in multiclass classification using Gaussian Processes.