Anurag Chaudhury

997298160

Assignment 1

NOTE: TR1 means training set 1 (train1x and train1y) and TR2 means (train2x and train2y)

1. **Linear Model**

Covariates	Test case square error (TR1)	Test case sq. error (TR2)
Given	0.2876439	0.2890586
Square Basis functions (16)	0.2832273	0.2577993
Cube basis functions (24)	0.3106012	0.2514223

For training set 1, as we increase the number of basis functions there is first a dip and then an increase in the square error on the test set. This indicates that maybe the model is over fitting the training set when we use cubic basis functions as well.

On the other hand, for the second training set, the square error decreases as we increase the number of basis functions. This indicates that maybe the second training set is a better reflection of the test set.

From the above observations, it is clear that the square error is minimized when using the second training set (train2x) with 24 predictors (normal covariates, squares and cubes)

2. Linear covariance function

For a linear covariance function we see that the square error **is 0.2876462 when training on the first training set and 0.2890609 when training on the second training set**. This is very close to the results obtained from just using a linear model with the given covariates (as expected).

3. Other covariance functions

Tested the square covariance function on 2 training sets and noted the maximum log likelihood obtained in each case by using different hyperparameters.

Different hyperparameters were obtained by using a uniform distribution for each of them.

The table below illustrates the different hyperparameter starting values and their final values after running the gp_log_likelihood function.

Initial hyperparameters (σ, γ, ρ)	Final hyperparameters (σ, γ, ρ)
(3.836155,1.883457,9.242932)	0.4576301 4.9960569 0.1490625

(5.934841,8.763616,3.042249)	0.4576305 4.9959589 0.1490645
(6.223477 , 6.022438, 7.394077)	0.4576311 4.9959133 0.1490662
(1.486374 , 2.255353 ,4.19862)	0.4576312 4.9960275 0.1490628
(5.386058 , 1.525368 ,0.3269338)	0.4576295 4.9958098 0.1490667

Log likelihood values and Sq.Error corress. To different initial value settings

σ	γ	ρ	TR1 log likelihood	TR2 log likelihood	Sq. Error TR1	Sq. Error TR2
3.836155	1.883457	9.242932	-197.8135	-209.0035	0.2930736	0.256647
5.934841	8.763616	3.042249	-197.8135	-209.0035	0.2930737	0.2566457
6.223477	6.022438	7.394077	-197.8135	-209.0035	0.293074	0.2566453
1.486374	2.255353	4.19862	-197.8135	-209.0035	0.293075	0.256645
5.386058	1.525368	0.3269338	-197.8135	-209.0035	0.2930738	0.256645

As can be seen, the difference in hyperparameter values does not alter the log likelihood obtained in either training set. In both cases, it seems to always be converging to the same local maxima. This is probably because the hyperparameter values tested do not differ by a large value in each of the tests performed.

2. Covariance function vs Test error

Using any of the above hyperparameters (since the average square error across the different hyperparameters was not significant) –

Noise	Gamma	Rho
3.836155	1.883457	9.242932

Final values of hyperparams for different covariance functions

Cov. Func.	TR 1 Final hp values	TR 2 final value hp values
Square(σ, γ, ρ)	(0.4007879, 1.1273876, 1.946)	(0.4485, 1.678, 1.114)
Absolute(σ, γ, ρ)	(0.35, 1.39, 0.239)	(0.424, 1.32, 0.187)
Combined(σ, γ1, ρ γ2)	(0.346, 1.161, 0.3287,1.59)	(0.4301, 0.709, 0.398,2.66)

Covariance func.	Test Error TR1	Test Error TR2
Square	0.2930736	0.256647
Square(scaled data)	0.2435766	0.245429
Absolute(scaled data)	0.240903	0.2164909
Combined(scaled data)	0.237124	0.2207385

Covariance function and test errors in different training sets

The test error for both training sets is smaller n the case of the square covariance, after we scale the first and 7th covariates of the training and test data. This is most likely because when summing across the square differences in the covariate vectors between two training/test instances, the difference in the case of the 1st and 7th covariates carries larger weight since it shall be a bigger number. Thus, training instances that are considerably similar in terms of the values of the other covariates might end up being considered dissimilar – more possible error. However, after scaling the 1st and 7th covariates to the same range as the other ones, this is taken care of and correspondingly the observed square error decreases.

Essentially, the fact that there is a larger margin between the values for the 1st and 7th covariates negatively affects the result but by scaling them down to the same range as the other covariates, we get a better measure of the covariance between the training/test examples and hence less square error.

Furthermore, the average error of the combined cov. Function across the 2 training sets is 0.2289 whereas the average error of the absolute cov. Function across the 2 training sets is 0.2286. This would indicate that either covariance function is fine, however to say this with more certainty we would need to evaluate their performances by training across more different training sets.

3. <u>Cross Validation Error</u>

Starting values – σ= 3.836155, γ1 = 1.883457, ρ =9.242932, γ2 = 2.5

Cov.Func	TR1 CV hp (σ, γ, ρ)	TR2 CV hp (σ, γ, ρ)	TR1 test error	TR2 test error	TR1 LL Test error	TR2 LL Test error	Comp. time(secs)
Square	9.754036 246.839307 1.045909	0.1714397 4.1931471 1.2101390	0.3522	0.347	0.293	0.256	229.5
Absolute	1.330321e+07 8.304367e+07 9.212394e-01	2.508952e+07 1.190191e+08 6.937254e-01	0.44	0.33	0.2409	0.216	243.1
Combined (σ, γ1, ρ, γ2)	1.9150656 30.7864598 34.2912078 0.3690727	6.367201e+07 2.985351e+08 9.563068e+06 1.025747e+00	0.299	0.509	0.237	0.220	426.8

The square error is higher for cross validation as opposed to maximum likelihood. This is probably because of the fact that that cross validation is trying to find the hyper parameters to minimize square error on a small amount of data whereas log likelihood method is trying to find the maximum probability across the entire dataset. This would likely cause the log likelihood method to perform better on the test set.

I experimented further with cross validation by trying another starting point and observed a slightly different set of results from before. This would seem to indicate that with the nlm method trying to minimize the square error , the cross validation is sensitive to the starting point.

Starting values – σ= 5.386058, γ1 =1.525368, ρ = 0.3269338, γ2 =2.5

Cov.Func	TR1 CV hp (σ, γ, ρ)	TR2 CV hp (σ, γ, ρ)	TR1 test error	TR2 test error	TR1 LL Test error	TR2 LL Test error	Comp. time(secs)
Square	9.7961402 249.6623676 0.9992021	8.685710e+05 5.758014e+07 1.206972e+00	0.339	0.612	0.293	0.256	228.3
Absolute	1.707166e+07 8.653325e+07 8.658169e-01	1.521791e+07 8.810405e+07 6.934313e-01	0.41	0.335	0.2409	0.216	242.8
Combined (σ, γ1, ρ, γ2)	3.865751e+06 1.947925e+07 9.006227e-01 3.977880e+07	6.367201e+07 2.985351e+08 9.563068e+06 1.025747e+00	0.287	0.429	0.237	0.220	431.2

5. Sampling

Training set 1

Num.Samples	Square Cov. Error	Absolute Cov. Error	Combined Cov. Error
10	0.2788892	0.2329533	0.230884
100	0.2392045	0.2435177	0.2296235
1000	0.2394504	0.2387822	0.2311403

TR1 - Computation time

Num.Samples	Square Cov.	Absolute Cov.	Combined Cov.
10	8.29	8.16	11.44
100	84.09	83.07	115.66
1000	413.39	406.2	562.59

Training data 2

Num.Samples	Square Cov. Error	Absolute Cov. Error	Combined Cov. Error
10	0.2670786	0.2181359	0.22031
100	0.2421811	0.2156638	0.22017
1000	0.2454633	0.2159531	0.2213

TR2 -Computation time

Num.Samples	Square Cov.	Absolute Cov.	Combined Cov.
10	8.52	8.13	11.41
100	83.62	82.78	114.82
1000	415.86	408.09	562.5

From the above observations it is apparent that as we increase the number of sample points, there is somewhat improvement in the average square error from both training sets across all the covariance functions. However, the improvement is very little if any in the different cases. Also, the time taken increases significantly as we increase the number of sampling points. This is because for each test case it now has to compute the expected value based on a larger number of points.

The results are quite similar to that of marginal likelihood and better than cross validation. This is because the prior is similar with the posterior distribution and as such sampling from the prior produces good results, a prior which did not reflect this similarity would probably give bad results.