

Charlie meeting 7th July

Meeting

This week I showed Mark and Andrew the results I produced, and there are some concerns regarding the performance of the invariance kernel. Mark has suggested me to start writing up the thesis and answer each question one by one and slowly expanding bits by bits. Try to answer a sequence of questions and separate them out and build up one by one; for example, learning invariance can be different from data efficiency problem although they are ultimately interlinked. In terms of comparing the performance, Mark has suggested me to try with fewer data to test data efficiency. We will then accordingly describe the benefits of the invariance kernel. He also said marginal likelihood is not a suitable metrics to measure the performance of the algorithm since even it's a better model that better explains the training data, it is not helpful if it cannot predict better. For local optim problem, we can try to initialise the polynomial around the true solution and see if it recognise the invariance; if it does then perhaps we need to try different optimisation techniques, such as different parameterisation or different optimiser. For the prior on the polynomial coefficients, Andrew suggested me to use Laplace prior instead of Gaussian to encourage sparse solution (similar to LASSO), I can also set the variance of the prior to be a hyperparameter. We also talked about the fact that in our algorithm, the Euler discretisation is baked in, and we might need to do something about that in the future, but for now we will just decrease the time step to obtain smaller error. We can also compare the RMSE of the ground truth to learnt function so that we do not have to rely on Euler. Mark also suggests me to run on CPU instead of GPU for large memory requirements.

Does it make sense that if I fix all the parameters, the invariance kernel does not perform better. I had some ideas, such as dropping out adding l1 loss to marginal likelihood, adding invariance criteria directly to the loss function? Different optimiser. How do I make the regularisation not so strong on real coefficients but strong on the zeroed ones? not pruning etc.? too much data naive GP learn but too few data the coefficients don't learn especially for double pendulum where a lot of data is needed.