Linear Shrinkage Learner for GP Noise Variance

Xingchen Wan
28 Feburary 2019

1 Methods

Various ways to tune the hyperparameters for a GP regression were implemented. We can use 1) Maximum Likelihood Estimate (MLE), 2) Maximum-a-posteriori (MAP) estimate, 3) Hybrid Monte Carlo (HMC) or 4) the linear shrinkage estimator.

`param_maximum_likelihood` implements both the MAP and MLE methods. If the "prior" of various hyperparameters are set to None, the function will do a MLE estimate. Otherwise MAP estimate will be done. in the `__main__` function a number of priors were written, and since the hyperparameters are strictly non-negative, a few Log-Gaussian priors were given.

It is also possible to fix the noise variance value and optimise only on the kernel hyperparameters. This will be useful for later, since `param_lin_shrinkage` requires an input model.

`param_hmc` implements the Hybrid Monte Carlo method. It first samples on the parameter posterior surface until the budget is exhausted, and then select the hyperparameters by computing the mean of the samples. Sampling on the posterior is more computationally expensive than the previous point estimates, so if the budget is set to a large number this can take quite a long time to complete.

`param_lin_shrinkage` is the linear shrinkage estimator for **Gaussian noise only**. Unlike the previous two functions which compute estimates for every hyperparameter, this estimator only concerns the Gaussian noise variance (or jitter, or Gaussian likelihood variance) for the GP. Thus, to function properly this function takes a `gpy_gp` input, which is a GPy GP object. It will set the kernel hyperparameters (for RBF kernel, they are the variance and lengthscales) to the values of `gpy_gp`, and set the noise variance to the value estimate from linear shrinkage. Since for now a small dataset is used, it first computes an eigenvalue decomposition of the covariance matrix $\mathbf{K}$, and computes the threshold, λ_c, eigenvalues above which will be considered outliers depending on the threshold argument `c`. It will then compute the bulk mean:

$$\lambda_b = \frac{1}{n}\sum_{i=1}^{n}\lambda_i, \forall\lambda_i \leq \lambda_c$$

For GP, this will be taken as the noise variance and $K \leftarrow K + \lambda_b I$ (Todo: test shrinkage instead $K \leftarrow (1-\lambda_b)K + \lambda_b I$).

2 Experiments

2.1 Hydrodynamics Dataset

Experiments were performed on the hydrodynamics dataset, a 6-dimensional regression problem. Firstly the dataset is randomised, and was split into two halves for training and testing, then GP regression with RBF kernel was performed. RMSE is the root mean standard error over the test set:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{*}(\hat{y}_i - y_i)^2}{N}}$$

LL is the Log-likelihood of the model on the test set, since the GP returns a variance of the prediction too:

$$LL = \sum_{i=1}^{*} \log \mathcal{N}(y_i; \hat{y}_i, \sigma_i^2)$$

For the experiments, a maximum evaluation of 2,000 iterations was budgeted for MAP and MLE. For MAP, the prior is specified as:

$$\log(\theta) \sim \mathcal{N}(0, \sigma^2 I)$$

where $\sigma = 2$. 2,000 samples were budgeted for HMC method.

	MLE	MAP	LinShrink	HMC
RMSE	0.973	0.616	**0.579**	0.649
LL	-229	-173	**-122**	-169

Table 1: Results for the Hydrodynamics Dataset

2.2 Boston Housing

A second experiment was performed on the Boston Housing Dataset, and the results are summarised in the table:

	MLE	MAP	LinShrink	HMC
RMSE	3.72	3.67	**3.41**	3.76
LL	**-694.3**	-700.3	-701.8	-702.0

Table 2: Results for the Boston Housing Dataset