

## 8. APPENDIX (SUPPLEMENTARY)

### 8.1. Partial Derivatives and Gradient

The expression of  $\mathbf{g}^{(\eta)}(\boldsymbol{\theta})$  used for updating the GP hyper-parameters,  $\boldsymbol{\theta}$ , in (12) is obtained as:

$$\begin{aligned}\mathbf{g}^{(\eta)}(\boldsymbol{\theta}) = & -2 \cdot \mathbf{y}_V^T \mathbf{K}_{VT}(\boldsymbol{\theta}_h) \mathbf{z}_T^\eta \\ & + (\mathbf{z}_T^\eta)^T \mathbf{K}_{VT}(\boldsymbol{\theta}_h)^T \mathbf{K}_{VT}(\boldsymbol{\theta}_h) \mathbf{z}_T^\eta \\ & + (\boldsymbol{\lambda}^\eta)^T \mathbf{K}_{TT}(\boldsymbol{\theta}_h) \mathbf{z}_T^\eta \\ & + \rho(\sigma_e^2 \mathbf{z}_T^\eta - \mathbf{y}_T)^T \mathbf{K}_{TT}(\boldsymbol{\theta}_h) \mathbf{z}_T^\eta \\ & + \frac{\rho}{2} (\mathbf{z}_T^\eta)^T \mathbf{K}_{TT}(\boldsymbol{\theta}_h) \mathbf{K}_{TT}(\boldsymbol{\theta}_h) \mathbf{z}_T^\eta.\end{aligned}\quad (15)$$

For each element of  $\boldsymbol{\theta}$  (denoted as  $\theta_i$ ), its partial derivative is computed as:

$$\begin{aligned}\frac{\partial \mathbf{g}^{(\eta)}(\boldsymbol{\theta})}{\partial \theta_i} = & -2 \cdot \mathbf{y}_V^T \frac{\partial \mathbf{K}_{VT}(\boldsymbol{\theta}_h)}{\partial \theta_i} \mathbf{z}_T^\eta \\ & + (\mathbf{z}_T^\eta)^T \frac{\partial \mathbf{K}_{VT}^T(\boldsymbol{\theta}_h)}{\partial \theta_i} \mathbf{K}_{VT}(\boldsymbol{\theta}_h) \mathbf{z}_T^\eta \\ & + (\mathbf{z}_T^\eta)^T \mathbf{K}_{VT}^T(\boldsymbol{\theta}_h) \frac{\partial \mathbf{K}_{VT}(\boldsymbol{\theta}_h)}{\partial \theta_i} \mathbf{z}_T^\eta \\ & + (\boldsymbol{\lambda}^\eta)^T \frac{\partial \mathbf{K}_{TT}(\boldsymbol{\theta}_h)}{\partial \theta_i} \mathbf{z}_T^\eta \\ & + \rho(\sigma_e^2 \mathbf{z}_T^\eta - \mathbf{y}_T)^T \frac{\partial \mathbf{K}_{TT}(\boldsymbol{\theta}_h)}{\partial \theta_i} \mathbf{z}_T^\eta \\ & + \frac{\rho}{2} (\mathbf{z}_T^\eta)^T \mathbf{K}_{TT}(\boldsymbol{\theta}_h) \frac{\partial \mathbf{K}_{TT}(\boldsymbol{\theta}_h)}{\partial \theta_i} \mathbf{z}_T^\eta \\ & + \frac{\rho}{2} (\mathbf{z}_T^\eta)^T \frac{\partial \mathbf{K}_{TT}(\boldsymbol{\theta}_h)}{\partial \theta_i} \mathbf{K}_{TT}(\boldsymbol{\theta}_h) \mathbf{z}_T^\eta.\end{aligned}\quad (16)$$

### 8.2. Explicit Form of Kernel Functions

The expressions for the selected kernels that we use for the synthetic data are listed below.

- **Squared Exponential (SE) Kernel**

SE kernel is usually regarded as the default kernel for GP models, due to its great universality as well as many good properties. The length scale  $l$  in an SE kernel specifies the width of the kernel and thereby determines the smoothness of the regression function.

$$k_{se}(x, x') = \sigma^2 \exp\left(-\frac{(x - x')^2}{2l^2}\right)$$

- **Locally Periodic (LP) Kernel**

Periodicity is another important pattern that people always get interested, especially in modeling time series data. Most of the real data do not repeat themselves exactly. Therefore combining a local kernel together with

a periodic kernel into a locally periodic kernel, is considered to allow the shape of the repeating patterns to evolve over time:

$$k_{lp}(x, x') = \sigma^2 \exp\left(-\frac{2\sin^2(\pi|x - x'|/p)}{l^2}\right) \exp\left(-\frac{(x - x')^2}{2l^2}\right)$$

- **Composite SE + LP Kernel**

One good thing about using kernel function is its flexibility in combining various kernel components, which allows multiplications and/or additions over different kernels to capture different features of the data. In our experiments, we added up one SE kernel and one LP kernel to model local periodicity with trend.

$$k_{se+lp}(x, x') = \sigma^2 \exp\left(-\frac{(x - x')^2}{2l_1^2}\right) + \sigma^2 \exp\left(-\frac{2\sin^2(\pi|x - x'|/l_3)}{l_2^2}\right) \exp\left(-\frac{(x - x')^2}{2l_2^2}\right)$$

### 8.3. Implementation Details

The practical implementation of GPCV-ADMM requires special attentions to the following aspects.

**Initialization.** A good starting point for both the hyper-parameters  $\boldsymbol{\theta}$  and the auxiliary variable  $\mathbf{z}_T$ , will lead to faster and smoother convergence of GPCV-ADMM as observed in Figure 2. Random restarts could be adopted to alleviate the adverse impact of bad initializations.

**Numerical Search.** We follow (10) to update the GP hyper-parameters numerically. Coordinate descent [20] is adopted when  $\boldsymbol{\theta}$  has more than one element. New GD type of methods such as the Adam algorithm. and other variants could be used for faster and more stable numerical search.

**Choice of the regularization parameter  $\rho$ .** The magnitude of  $\rho$  controls both the descent speed and the convexity of the ADMM objective function. A large  $\rho$  endows a strong convexity of the ADMM objective function, yet often requiring more iterations to converge. A smaller  $\rho$  endows faster descent, but the training procedure may get stuck at a bad local minimum more easily. When a suitable  $\rho$  value is difficult to determine, one possible remedy, as suggested in [19], is to use a different and smaller  $\rho'$  in (9c) for updating the dual variable.

**Simulation Platform.** Our GPCV-ADMM is implemented in R (version 3.5.2), and compared with the GPML toolbox executed in MATLAB 2018b. All the experiments were conducted on a MacBook Pro with 2.2 GHz Intel Core i7.

---

**Algorithm 1** HOCV Based GP Hyper-Parameter Optimization

---

**Input:** Complete data set  $\mathcal{D}$  divided into  $\mathcal{D}_T$  and  $\mathcal{D}_V$

**Output:** Optimal GP hyper-parameters  $\theta^*$

**Initialization:**  $\eta = 0, \lambda^0, z_T^0, \theta^0$

- 1: **while**  $\|\theta^{\eta+1} - \theta^\eta\|_2 \geq \epsilon$  and  $\eta \leq \text{maxItr}$  **do**
  - 2:   Update  $\theta^{\eta+1}$  according to (10)
  - 3:   Update  $z_T^{\eta+1}$  according to (12)
  - 4:   Update  $\lambda^{\eta+1}$  according to (9c)
  - 5:   Set  $\eta = \eta + 1$ .
  - 6: **end while**
  - 7: **return**  $\theta^* = \theta^\eta$
- 

#### 8.4. Algorithm Setup

For both fairness and clarity of comparisons, all the hyper-parameters are initialized with same values for both GPCV-ADMM and GPML when testing on the synthetic data sets under a fixed kernel configuration. However, random restarts are recommended for initialization in practice, and was also adopted in our experiment for the real  $CO_2$  concentration data set. In the Algorithm 1 above, the auxiliary variable  $z_T$  of GPCV-ADMM is initialized according to (8) with a perturbed  $\theta^0$ , and the dual variable  $\lambda$  initialized to be a vector of all ones. The regularization parameter is pre-selected to be  $\rho = 5$ . The error tolerance for ADMM is set to be  $\epsilon = 10^{-2}$  and the maximum number of iterations is set to be  $\text{maxItr} = 100$ .