Machine Learning Notes

All in the data.

Author: occupymars

Date: June. 23, 2025

Version: 0.1

Contents

Chapter 1 Gaussian Process

1.1 Gaussian processes for dynamics learning in model predictive control (2025)

1.1.1 Overview of static Gaussian process regression

GPR was introduce in the statistics community by *Curve Fitting and Optimal Design for Prediction*, and gained attention after *Bayesian Learning for Neural Networks* proved that they can be regarded as neural networks of infinite width.

Given two input data $Z = \{z_1, \cdots, z_N\}, Z^* = \{z_1^*, \cdots, z_N^*\}$, using the GP prior, we get:

$$\begin{bmatrix} g_Z \\ g_{Z^*} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \mathscr{K}_{Z,Z} & \mathscr{K}_{Z,Z^*} \\ \mathscr{K}_{Z^*,Z} & \mathscr{K}_{Z^*,Z^*} \end{bmatrix} \right)$$

Now given observation for Z as $Y = [y_1, \cdots, y_N]$, the posterior can be written as:

$$p(g_Z, g_{Z^*} \mid Y) = \frac{p(Y \mid g_Z) p(g_Z, g_{Z^*})}{p(Y)}$$

The kernel is often taken as Gaussian one:

$$\mathscr{K}_{Z,Z^*} = \lambda \exp\left\{ -\frac{\|Z - Z^*\|^2}{2\eta} \right\}$$

If we are using the measurement model $y_i = g(z_i) + w_i$ and assume noise term is independent from prior g:

$$Y \mid g_Z \sim \mathcal{N}(g_Z, \sigma_w^2 I_N)$$

so the posterior is also Gaussian, we can get the posterior of interest:

$$p(g_{Z^*} \mid Y) = \int_{\mathcal{Z}} \frac{p(Y \mid g_Z) p(g_Z, g_{Z^*})}{p(Y)} dg_Z$$

then compute the first and second order moments, we get $g_{Z^*} \mid Y \sim \mathcal{N}(\mu(Z^*), \Sigma(Z^*))$:

$$\begin{cases} \mu(Z^*) &= \mathscr{K}_{Z^*,Z}(\mathscr{K}_{Z,Z} + \sigma_w^2 I_N)^{-1} Y \\ \Sigma(Z^*) &= \mathscr{K}_{Z^*,Z^*} - \mathscr{K}_{Z^*,Z}(\mathscr{K}_{Z,Z} + \sigma_w^2 I_N)^{-1} \mathscr{K}_{Z,Z^*} \end{cases}$$

Remark 1, Alternative paradigms for uncertainty quantification, from RKHS to multi-arm bandits, frequency methods...

The hyperparameters are estimated from a subset of data (Z_h, Y_h) by optimizing the marginal likelihood:

$$\text{argmax}_\xi \, p(Y_h \mid \xi) = \text{argmax}_\xi \int_{\mathbb{R}^N} p(Y_h \mid g_{Z_h}, \xi) p(g_{Z_h} \mid \xi) dg_{Z_h} \tag{1.1}$$

if the measurement is i.i.d., then $Y_h \mid \xi \sim \mathcal{N}(0, \mathscr{K}_{Z_h,Z_h} + \sigma_w^2)$, so the above optimization problem can be written as a negative-log-likelihood minimization problem:

$$\text{argmax}_\xi \, Y_h^\top (\mathscr{K}_{Z_h,Z_h} + \sigma_w^2)^{-1} Y_h + \log\det(\mathscr{K}_{Z_h,Z_h} + \sigma_w^2) \tag{1.2}$$

we can use a gradient based method to optimize this one, however this cost is not convex, so its result maybe not global minimum and thus unreliable. An alternative way is ***Markov Chain Monte Carlo*** approaches, perform numerical integration on 1.1.

1.1.2 Gaussian processes for dynamical systems

A first option to describe a dynamical system is the Nonlinear, Auto-Regressive with eXogenous input (***NARX***) model:

$$y_i = g_{NARX}(y_{i-1}, \cdots, y_{\tau_y}, u_{i-1}, \cdots, u_{\tau_u}) + w_i$$

We can write it as the state-space model:

$$\begin{cases} x_{i+1} &= f(x_i, u_i) + v_i \\ y_i &= g(x_i) + w_i \end{cases} \tag{1.3}$$

where f and g denotes transition and emission maps, typically g is known (even if it is not, we can augment it into transition maps). There are two challenges, learning two maps and state inference (from y get x), that is tackled by two different approaches in academic:

- Optimizing latent state variables: treate the state variables as optimization variables, jointly optimize it with model parameters to get maximum likelihood.
- Alternating function learning and state inference: this method try to extend Bayesian techniques such as ***Extended Kalman Filter***, ***Unscented Kalman Filter***, Assumed Density Filter, and Particle Filter to non-parametric models, the approximation in these studys are Taylor expansions, exact moment matching and particle representations. But when the state measurement are not available, we have to iteratively alternate between inferring the posterior and updating ξ to maximize the marginal likelihood, using algorithm like ***Expectation Maximization***. The approximation to decrease the computational complexity are truncated orthogonal basis functions expansions (see [132, 133, 134]) and variational inference.

1.1.3 Porblem formulation

The discrete model dynamic:

$$x_{i+1} = g_{nom}(x_i, u_i) + B_d g(x_i, u_i) + v_i \tag{1.4}$$

where $g_{nom} : \mathbb{R}^{n_x \times n_u} \to \mathbb{R}^{n_x}, g : \mathbb{R}^{n_x \times n_u} \to \mathbb{R}^{n_d}$, if we do not have nominal model, then $B_d = I_{n_x}$.

And we use $z_i = [x_i^\top \quad u_i^\top]^\top$, we will train n_d GPs for each dimension separately.

The optimal control problem is then:

$$\begin{aligned}
\text{minimize}_{\{\pi_i\}} \quad & \mathbb{E}\left[\bar{\mathcal{L}}_{\bar{T}}(x_{\bar{T}}) + \sum_{i=0}^{\bar{T}-1} \bar{\mathcal{L}}_{\rangle}(x_i, u_i)\right] && (1.5)\\
\text{subject to} \quad & x_{i+1} = g_{\text{nom}}(x_i, u_i) + B_d g(x_i, u_i) + v_i && (1.6)\\
& u_i = \pi_i(x_i) && (1.7)\\
& \mathbb{P}(h_j(x_i, u_i) \le 0, \forall i \ge 0) \ge p_j \quad \forall j = 1, \dots, n_h && (1.8)\\
& x_0 = \bar{x}_0 && (1.9)
\end{aligned}$$

This problem is hard to solve, so we transform it to a MPC problem at time step k:

$$\begin{aligned}
\text{minimize}_{\{\pi_i|k\}} \quad & \mathbb{E}\left[\mathcal{L}_T(x_{T|k}) + \sum_{i=0}^{T-1} \mathcal{L}_{\rangle}(x_{i|k}, u_{i|k})\right] && (1.10)\\
\text{subject to} \quad & x_{i+1|k} = g_{\text{nom}}(x_{i|k}, u_{i|k}) + B_d g(x_{i|k}, u_{i|k}) + v_{i|k} && (1.11)\\
& u_{i|k} = \pi_{i|k}(x_{i|k}) && (1.12)\\
& \mathbb{P}(h_j(x_{i|k}, u_{i|k}) \le 0, \forall i \ge 0) \ge p_j \quad \forall j = 1, \dots, n_h && (1.13)\\
& x_{0|k} = x_k && (1.14)
\end{aligned}$$

1.1.4 Scalable methods for GPR

More detailed survey please refer to *When Gaussian Process Meets Big Data: A Review of Scalable GPs*.

Table 1.1: Computational Complexity of Gaussian Process Methods.

	GP Full	Subset of Data	Expert-based	FTC	SSGP	SKI	SVGP
Training	$\mathcal{O}(N^3)$	$\mathcal{O}(M^3)$	$\mathcal{O}(NM_e^2)$	$\mathcal{O}(NM^2)$	$\mathcal{O}(Np^2)$	$\mathcal{O}(N + M \log M)$	$\mathcal{O}(M^3)$
Inference	$\mathcal{O}(N^2)$	$\mathcal{O}(M^2)$	$\mathcal{O}(M_e^2)$	$\mathcal{O}(M^2)$	$\mathcal{O}(p^2)$	$\mathcal{O}(M \log M)$	$\mathcal{O}(M^2)$

Subset of Data, sample data using some criterion (refer to *Gaussian Process Models: PAC-Bayesian Generalisation Error Bounds and Sparse Approximations*, chapter 4) and clustering. This will overestimate uncertainty, but new study leveraging graphons complementss rigorous bounds for it.

We can also use multiple models for different regions for non-Stationarity or scalability. One of them is called ***Mixture-of-Experts*** (MoE), given N_{exp} GPs, denoting with $\{s_k(\cdot)\}_{k=1}^{N_{exp}}$ a set of gating functions, the overall likelihood is

$$p_{MoE}(y \mid g_z^1, \cdots, g_z^{N_{exp}}) = \sum_{k=1}^{N_{exp}} s_k(g_z^k) p_k(y \mid g_z^k)$$

to scale well, we need to use infinite MoE or one of the approximation methods. We can also pre-allocate experts but this will lose connection between experts. See [166], [167] for online updates. And there is another method called "bagging".

Instead of resorting to a linear combination of GPs, we can use ***Product-of-Experts*** (PoE), where

$$p_{PoE}(y \mid g_z^1, \cdots, g_z^{N_{exp}}) \propto \prod_{k=1}^{N_{exp}} p_k(y \mid g_z^k)$$

this will make weak expert plays which is not good, so we can use weighted product and Bayesian Committee Machine, combined we have [174]. MoE and PoE combine in *Deep Structured Mixtures of Gaussian Processes*. Analysis of theory in *An asymptotic analysis of distributed nonparametric methods*.

Inducing Variables, given inducing points (pseudo-inputs) $\bar{Z}$ and $g_{\bar{Z}} \sim \mathcal{N}(0, \mathcal{K}_{\bar{Z},\bar{Z}})$, g_Z and g_{Z^*} are conditionally independent, they can only communicate through $g_{\bar{Z}}$. There are two main groups, one is approximate prior $p(g_Z, g_{Z^*})$ and do exact inference (reviewed in *A Unifying View of Sparse Approximate Gaussian Process Regression*), one is from original prior and approximate $p(g_{Z^*} \mid Y)$ (reviewed in *A unifying framework for Gaussian process pseudo-point approximations using power expectation propagation*), they are compared in *Understanding Probabilistic Sparse Gaussian Process Approximations*.

First we talk about method approximating prior, with:

$$\begin{aligned} p(g_Z, g_{Z^*}) &= \int p(g_Z, g_{Z^*} | g_{\bar{Z}}) p(g_{\bar{Z}}) dg_{\bar{Z}} \\ &\approx \int q(g_Z | g_{\bar{Z}}) q(g_{Z^*} | g_{\bar{Z}}) p(g_{\bar{Z}}) dg_{\bar{Z}} \\ &= q(g_Z, g_{Z^*}) \end{aligned} \tag{1.15}$$

the choice of $q(g_Z|g_{\bar{Z}}) = \mathcal{N}(\mathcal{K}_{Z,\bar{Z}}\mathcal{K}_{\bar{Z},\bar{Z}}^{-1} g_{\bar{Z}}, \tilde{Q}_{Z,Z})$ and $q(g_{Z^*}|g_{\bar{Z}}) = \mathcal{N}(\mathcal{K}_{Z^*,\bar{Z}}\mathcal{K}_{\bar{Z},\bar{Z}}^{-1} g_{\bar{Z}}, \tilde{Q}_{Z^*,Z^*})$ will differ between different methods below. The conditional distribution is actully $q(g_Z|g_{\bar{Z}}) = \mathcal{N}(\mathcal{K}_{Z,\bar{Z}}\mathcal{K}_{\bar{Z},\bar{Z}}^{-1} g_{\bar{Z}}, \mathcal{K}_{Z,Z} - \mathcal{K}_{Z,\bar{Z}}\mathcal{K}_{\bar{Z},\bar{Z}}^{-1}\mathcal{K}_{\bar{Z},Z})$, refer to proof, $\tilde{Q}$ is a low rank matrix.

- ***Subset of Regressors*** (SoR), $\mathcal{K}_{Z,Z} \approx \mathcal{K}_{Z,\bar{Z}}\mathcal{K}_{\bar{Z},\bar{Z}}^{-1}\mathcal{K}_{\bar{Z},Z} = Q_{Z,Z}$, so covariance of $q(g_Z|g_{\bar{Z}})$ and $q(g_{Z^*}|g_{\bar{Z}})$ is $\tilde{Q}_{Z,Z} = Q_{Z,Z} - \mathcal{K}_{Z,\bar{Z}}\mathcal{K}_{\bar{Z},\bar{Z}}^{-1}\mathcal{K}_{\bar{Z},Z} = 0$, possibly leading to overconfident predictions. $g_{Z^*} = \mathcal{K}_{Z^*,\bar{Z}} W_{\bar{Z}}, W_{\bar{Z}} \sim \mathcal{N}(0, \mathcal{K}_{\bar{Z},\bar{Z}}^{-1})$, $W_{\bar{Z}}$ can also be written as $\mathcal{K}_{\bar{Z},\bar{Z}}^{-1}\bar{Z}$.
- ***Deterministic Training Conditional*** (DTC), same mean of SoR, covariance is more sensible but the result is an inconsistent GP.
- ***Fully Independent Conditional*** (FIC), assume g_Z and g_{Z^*} are independent of $g_{\bar{Z}}$, and ***Fully Independent Training Conditional*** (FITC) admits the factorization on the training conditional only, at the price of having again an inconsistent GP. **If the prediction is to be performed on a single point, this two method coincide**.
- ***Partially Independent (Training) Conditional*** (PI(T)C) generalizes FI(T)C by introducing a block structure in the covariance. There maybe no significant improve with respect to FI(T)C.

These methods lead to an approximation marginal likelihood:

$$q(Y) = \mathcal{N}(0, \tilde{Q}_{Z,Z} + \mathcal{K}_{Z,\bar{Z}}\mathcal{K}_{\bar{Z},\bar{Z}}^{-1}\mathcal{K}_{\bar{Z},Z} + \sigma_w^2 I_N) \tag{1.16}$$

the choice of inducing points can be same as subset of data method, using information gain, online learning and greedy posterior maximization. Or we can treat them as hyperparameters, and maximized by 1.16, which is complicated and may lead to local optimal and over-fitting. Other methods can be seen in MCMC schemes, which will taker longer traning times.

Second we talk about approximate the posterior. We can use so-called ***Variational Free Energy*** (VFE) to get:

$$\begin{aligned} p(g_{Z^*}|Y) &= \int\int p(g_{Z^*}|g_Z, g_{\bar{Z}}) p(g_Z|g_{\bar{Z}}, Y) p(g_{\bar{Z}}|Y) dg_Z g_{\bar{Z}} \\ &\approx q(g_{Z^*}) \\ &= \int\int p(g_{Z^*}|g_{\bar{Z}}) p(g_Z|g_{\bar{Z}}) p(g_{\bar{Z}}|Y) dg_Z g_{\bar{Z}} \\ &= \int p(g_{Z^*}|g_{\bar{Z}}) p(g_{\bar{Z}}|Y) dg_{\bar{Z}} \\ &\approx \int p(g_{Z^*}|g_{\bar{Z}}) q(g_{\bar{Z}}) dg_{\bar{Z}} \end{aligned} \tag{1.17}$$

where $p(g_{Z^*}|g_Z, g_{\bar{Z}}, Y) = p(g_{Z^*}|g_Z, g_{\bar{Z}})$ because Y is just a noisy version of g_Z and $g_{\bar{Z}}$ is sufficient.

We then use variational inference to choose $g_{\bar{Z}}$ and $\bar{Z}$, by minimizing Kullback-Leibler (KL) divergence:

$$\mathcal{KL}\left(q(g_{Z^*}, g_Z) \| p(g_{Z^*}, g_Z|Y)\right) = \log p(Y) - \mathbb{E}_{q(g_{\bar{Z}}, g_Z)}\left[\frac{p(Y, g_{\bar{Z}}, g_Z)}{q(g_{\bar{Z}}, g_Z)}\right]$$

I think here log is for both term, and I think left side Z^* should be $\bar{Z}$, or the $=$ should be $\approx$.

In [204] we get $g(g_{\bar{Z}}) = \mathcal{N}(\mu_q, \Sigma_q)$, where:

$$\begin{aligned} \mu_q &= \sigma_w^{-2} \mathscr{K}_{\bar{Z},\bar{Z}} (\mathscr{K}_{\bar{Z},\bar{Z}} + \sigma_w^{-2} \mathscr{K}_{\bar{Z},Z} \mathscr{K}_{Z,\bar{Z}})^{-1} \mathscr{K}_{\bar{Z},Z} Y \\ \Sigma_q &= \mathscr{K}_{\bar{Z},\bar{Z}} (\mathscr{K}_{\bar{Z},\bar{Z}} + \sigma_w^{-2} \mathscr{K}_{\bar{Z},Z} \mathscr{K}_{Z,\bar{Z}})^{-1} \mathscr{K}_{\bar{Z},\bar{Z}} \end{aligned}$$

and the hyperparameters can be found by optimizing:

$$\log p(Y) - \log[\mathcal{N}(0, \sigma_w^2 + \mathscr{K}_{Z,\bar{Z}} \mathscr{K}_{\bar{Z},\bar{Z}}^{-1} \mathscr{K}_{\bar{Z},Z}] + \frac{1}{\sigma_w^2} \mathrm{Tr}(\mathscr{K}_{Z,Z} - \mathscr{K}_{Z,\bar{Z}} \mathscr{K}_{\bar{Z},\bar{Z}}^{-1} \mathscr{K}_{\bar{Z},Z})$$

the predictive distribution is the same as the one obtained in the DTC approach, but the optimization problem yielding hyperparameters and pseudo-inputs differs by the last addendum, which plays the role of a regularizer and acts against over-fitting. And the relation with FITC in [220], [221] for mini-batch training. Pseudo-inputs can be set arbitraily, or use SKI, or from training inputs.

Sparse GP error estimation in [230], [231].

Finite-dimensional representations of the kernel operator is next method, as above sparse GP revolve around the concept of eigen-decomposition of Gram matrices $\mathscr{K}_{Z,Z}$, this one consider eigen-decomposation of the kernel operator $\mathscr{K} : Z \times Z \to \mathbb{R}$.