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 $g_{Z^*} \mid Y \sim \mathcal{N}(\mu(Z^*), \Sigma(Z^*))$:

$$\begin{cases} \mu(Z^*) &= \mathscr{K}_{Z^*,Z} (\mathscr{K}_{Z,Z} + \sigma_w^2 I_N)^{-1} Y \\ \Sigma(Z^*) &= \mathscr{K}_{Z^*,Z^*} - \mathscr{K}_{Z^*,Z} (\mathscr{K}_{Z,Z} + \sigma_w^2 I_N)^{-1} \mathscr{K}_{Z,Z^*} \end{cases} \tag{1.1}$$

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:

$$\operatorname{argmax}_\xi \; p(Y_h \mid \xi) = \operatorname{argmax}_\xi \int_{\mathbb{R}^N} p(Y_h \mid g_{Z_h}, \xi) p(g_{Z_h} \mid \xi) dg_{Z_h} \tag{1.2}$$

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:

$$\operatorname{argmax}_\xi \; Y_h^\top (\mathscr{K}_{Z_h,Z_h} + \sigma_w^2)^{-1} Y_h + \log\det(\mathscr{K}_{Z_h,Z_h} + \sigma_w^2) \tag{1.3}$$

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 1.2.

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:

$$\begin{cases} x_{i+1} &= f(x_i, u_i) + v_i \\ y_i &= g(x_i) + w_i \end{cases} \tag{1.4}$$

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:

$$x_{i+1} = g_{nom}(x_i, u_i) + B_d g(x_i, u_i) + v_i \tag{1.5}$$

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:

$$\begin{aligned}
\text{minimize}_{\{\pi_i\}} \quad & \mathbb{E}\left[\bar{\mathcal{L}}_{\bar{T}}(x_{\bar{T}}) + \sum_{i=0}^{\bar{T}-1} \bar{\mathcal{L}}_{\rangle}(x_i, u_i)\right] && (1.6)\\
\text{subject to} \quad & x_{i+1} = g_{\text{nom}}(x_i, u_i) + B_d g(x_i, u_i) + v_i && (1.7)\\
& u_i = \pi_i(x_i) && (1.8)\\
& \mathbb{P}(h_j(x_i, u_i) \leq 0, \forall i \geq 0) \geq p_j \quad \forall j = 1, \ldots, n_h && (1.9)\\
& x_0 = \bar{x}_0 && (1.10)
\end{aligned}$$

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

$$\begin{aligned}
\text{minimize}_{\{\pi_i|k\}} \quad & \mathbb{E}\left[\mathcal{L}_T(x_{T|k}) + \sum_{i=0}^{T-1} \mathcal{L}_{\rangle}(x_{i|k}, u_{i|k})\right] && (1.11)\\
\text{subject to} \quad & x_{i+1|k} = g_{\text{nom}}(x_{i|k}, u_{i|k}) + B_d g(x_{i|k}, u_{i|k}) + v_{i|k} && (1.12)\\
& u_{i|k} = \pi_{i|k}(x_{i|k}) && (1.13)\\
& \mathbb{P}(h_j(x_{i|k}, u_{i|k}) \leq 0, \forall i \geq 0) \geq p_j \quad \forall j = 1, \ldots, n_h && (1.14)\\
& x_{0|k} = x_k && (1.15)
\end{aligned}$$

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	$\mathcal{O}(N^2)$	$\mathcal{O}(M^2)$	$\mathcal{O}(M_e^2)$	$\mathcal{O}(M^2)$	$\mathcal{O}(p^2)$	$\mathcal{O}(M\log M)$	$\mathcal{O}(M^2)$

1. 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*.

2. Inducing Variables, given inducing points (pseudo-inputs) $\bar{Z}$ and $g_{\bar{Z}} \sim \mathcal{N}(0, \mathscr{K}_{\bar{Z},\bar{Z}})$, 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:

$$\begin{aligned} p(g_Z, g_{Z^*}) &= \int p(g_Z, g_{Z^*} | g_{\bar{Z}}) p(g_{\bar{Z}}) dg_{\bar{Z}} \\ &\approx \int q(g_Z | g_{\bar{Z}}) q(g_{Z^*} | g_{\bar{Z}}) p(g_{\bar{Z}}) dg_{\bar{Z}} \\ &= q(g_Z, g_{Z^*}) \end{aligned} \tag{1.16}$$

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

- ***Subset of Regressors*** (SoR), $\mathscr{K}_{Z,Z} \approx \mathscr{K}_{Z,\bar{Z}} \mathscr{K}_{\bar{Z},\bar{Z}}^{-1} \mathscr{K}_{\bar{Z},Z} = Q_{Z,Z}$, so covariance of $q(g_Z|g_{\bar{Z}})$ and $q(g_{Z^*}|g_{\bar{Z}})$ is $\tilde{Q}_{Z,Z} = Q_{Z,Z} - \mathscr{K}_{Z,\bar{Z}} \mathscr{K}_{\bar{Z},\bar{Z}}^{-1} \mathscr{K}_{\bar{Z},Z} = 0$, possibly leading to overconfident predictions. $g_{Z^*} = \mathscr{K}_{Z^*,\bar{Z}} W_{\bar{Z}}, W_{\bar{Z}} \sim \mathcal{N}(0, \mathscr{K}_{\bar{Z},\bar{Z}}^{-1})$, $W_{\bar{Z}}$ can also be written as $\mathscr{K}_{\bar{Z},\bar{Z}}^{-1} \bar{Z}$.
- ***Deterministic Training Conditional*** (DTC), same mean of SoR, covariance is more sensible but the result is an inconsistent GP (train with low-rank approximation, test with full variance).
- ***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**. $\mathscr{K}_{FITC} = \mathscr{K}_{Z,\bar{Z}} \mathscr{K}_{\bar{Z},\bar{Z}}^{-1} \mathscr{K}_{\bar{Z},Z} + \text{diag}(\mathscr{K}_{Z,Z} - Q_{Z,Z})$
- ***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. $\mathscr{K}_{PITC} = Q_{Z,Z} - \text{block.diag}(\mathscr{K}_{Z,Z} - Q_{Z,Z})$.

These methods lead to an approximation marginal likelihood:

$$q(Y) = \mathcal{N}(0, \tilde{Q}_{Z,Z} + \mathscr{K}_{Z,\bar{Z}} \mathscr{K}_{\bar{Z},\bar{Z}}^{-1} \mathscr{K}_{\bar{Z},Z} + \sigma_w^2 I_N) \tag{1.17}$$

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 1.17, 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:

$$\begin{aligned} p(g_{Z^*}|Y) &= \int\int p(g_{Z^*}|g_Z, g_{\bar{Z}})p(g_Z|g_{\bar{Z}}, Y)p(g_{\bar{Z}}|Y)dg_Z g_{\bar{Z}} \\ &\approx q(g_{Z^*}) \\ &= \int\int p(g_{Z^*}|g_{\bar{Z}})p(g_Z|g_{\bar{Z}})p(g_{\bar{Z}}|Y)dg_Z g_{\bar{Z}} \\ &= \int p(g_{Z^*}|g_{\bar{Z}})p(g_{\bar{Z}}|Y)dg_{\bar{Z}} \\ &\approx \int p(g_{Z^*}|g_{\bar{Z}})q(g_{\bar{Z}})dg_{\bar{Z}} \end{aligned} \tag{1.18}$$

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:

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

and the hyperparameters can be found by optimizing:

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

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*** (structured kernel interpolation, kernel with structure good for efficient calculation), or from training inputs.

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

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

The first one is ***Sparse Spectrum Gaussian processes*** (SSGP), introduced in [234], using sum of Fourier features. It is suitable for stationary kernels:

$$\mathcal{K}(z_a, z_b) = \int \exp\{2\pi i s^\top (z_a - z_b)\}S(s)ds$$

where $S(s)$ is proportional to $p_S(s)$, for Gaussian kernel:

$$p_S(s) = \frac{1}{\lambda}\int \exp\{-2\pi i s^\intercal (z_a - z_b)\}\mathcal{K}(z_a, z_b)dz = \sqrt{2\pi\eta^{n_z}}\exp\{-2\pi^2\eta\|s\|^2\}$$

which is a multivariate Gaussian. Using MC method, sample $\{s_r, -s_r\}_{r=1}^{M/2}$ to respect symmetry and exploiting trigonometric identities:

$$\mathcal{K}(z_a, z_b) \approx \frac{\lambda}{M}\sum_{r=1}^{M/2}\left(\cos(2\pi s_r^\top z_a)\cos(2\pi s_r^\top z_b) + \sin(2\pi s_r^\top z_a)\sin(2\pi s_r^\top z_b)\right)$$

Then the kernel is approximate to $\mathcal{K}(z_a, z_b) = \langle\phi(z_a), \phi(z_b)\rangle$, where

$$\phi(z) = [\cos(2\pi s_1^\intercal z), \cdots, \cos(2\pi s_{M/2}^\intercal z), \cdots, \sin(2\pi s_{M/2}^\intercal z)]^\top$$

then $g(z) \approx \phi(z)^\top \alpha$.

Another one aims for a series expansion of the type (see Mercer's Theorem):

$$\mathcal{K}(z_a, z_b) = \sum_{k=1}^{+\infty}\gamma_k\varphi_k(z_a)\varphi_k(z_b)$$

such that $\int \mathcal{K}(z_a, z_b)\varphi_k(z_a)p_z(z_a)dz_a = \gamma_k\varphi_k(z_b)$, where $\{\varphi_k\}_{k=1}^{+\infty}$ is a family of orthonormal functions with respect to the measure induced by $p_z(\cdot)$, and $\{\gamma_k\}_{k=1}^{+\infty}$ is a set of decreasing, non-negative values.

4. Computational techniques is the last topic in this section.

First we talk about ***Nystrom approximation***, $\mathcal{K}_{Z,Z} \approx \mathcal{K}_{Z,\bar{Z}}\mathcal{K}_{\bar{Z},\bar{Z}}^{-1}\mathcal{K}_{\bar{Z},Z}$, can be efficiently obtained from an incomplete (block) column-based Cholesky decomposition when the same (block) columns are selected.

Secondly, ***Conjugate gradient*** (CG) method is for optimization problem $\min_x \frac{1}{2}x^\top Ax - x^\top b$, iteratively refines its solution estimate by performing the minimization successively in a growing subspace of orthogonal search directions. Computational complexity is $\mathcal{O}(PN^2)$, where P is the iteration, and CG can guarantee to when $P = N$, it will recover the exact solution. But we will make $P \ll N$ to get efficience while get a good approximation. With efficient MVMs (matrix-vector multiplications) of SKI, CG can speed up sparse GP.

Thirdly, the ***parallelization of computations***, for MoE, inducing point method (PITC has the block structure, using low-rank-cum-Markov approximations), Nystrom approximation (row-based Cholesky decomposition), CG (batch version).

5. online learning:

- ***Active-data selection***, using a variety of methods to add or delete data points in dataset.
- Expert-based methods, compute the distance of new data and each local model, and update corresponding local model.
- Incuding point methods, [358] for FITC, [366] for VFE, a lot of papers mentioned.
- Finite-dimensional approximations of kernel operator, [245] for SSGP and other.
- Computational methods, leverage SKI [382]

1.1.5 Uncertainty Propagation

Directly predict n-step, [387-389]. Indirect method, iteratively do one-step prediction, neglect non-consecutive states relation, the uncertainty prediction can be deteriorate [391].

1. Independent one-step-ahead predictions, with ramdom Z^*:

$$p(g_{Z^*}|Y) = \int p(g_{Z^*}|Z^*, Y)p(Z^*)dZ^* \tag{1.19}$$

the integral is the product of two Gaussian (if we assume $p(Z^*) = \mathcal{N}(\mu^*, \Sigma^*)$), it is not analytic, note that $p(g_{Z^*}|Z^*, Y) = \mathcal{N}(\mu(Z^*), \Sigma(Z^*))$. We actually only need first and second moments, which are given as:

$$\begin{aligned}
\mathbb{E}_{g_{Z^*}}[g_{Z^*} \mid \mathbf{Y}] &= \mathbb{E}_{Z^*}[\mathbb{E}_{g_{Z^*}}[g_{Z^*} \mid Z^*, \mathbf{Y}]] \\
&\overset{1.1}{=} \mathbb{E}_{Z^*}[\mu(Z^*)], \\
\mathrm{Var}_{g_{Z^*}}[g_{Z^*} \mid \mathbf{Y}] &= \mathbb{E}_{Z^*}[\mathrm{Var}_{g_{Z^*}}[g_{Z^*} \mid Z^*, \mathbf{Y}]] + \mathrm{Var}_{Z^*}[\mathbb{E}_{g_{Z^*}}[g_{Z^*} \mid Z^*, \mathbf{Y}]] \\
&\overset{1.1}{=} \mathbb{E}_{Z^*}[\Sigma(Z^*)] + \mathrm{Var}_{Z^*}[\mu(Z^*)].
\end{aligned} \tag{1.20}$$

We have three ways to approximate this. First is ***Linearlization***, using Taylor expansion:

$$\begin{aligned}
\mu(z^*) &\approx \mu(\mu^*) + \nabla_z \mu(z)|_{z=\mu^*}^\top (z^* - \mu^*) \\
\Sigma(z^*) &\approx \Sigma(\mu^*) + \nabla_z \Sigma(z)|_{z=\mu^*}^\top (z^* - \mu^*) + \tfrac{1}{2}(z^* - \mu^*)^\top \nabla_z^2 \Sigma(z)|_{z=\mu^*}^* (z^* - \mu^*)
\end{aligned} \tag{1.21}$$

which then:

$$\begin{aligned}
\mathbb{E}_{Z^*}[\mu(Z^*)] &\approx \mathbb{E}_{Z^*}[\mu(\mu^*) + \nabla_z \mu(z)^\top\big|_{z=\mu^*}(Z^* - \mu^*)] = \mu(\mu^*), \\
\mathbb{E}_{Z^*}[\Sigma(Z^*)] &\approx \mathbb{E}_{Z^*}[\Sigma(\mu^*) + \nabla_z \Sigma(z)\big|_{z=\mu^*}(Z^* - \mu^*) + \tfrac{1}{2}(Z^* - \mu^*)^\top \nabla_z^2 \Sigma(z)\big|_{z=\mu^*}(Z^* - \mu^*)] \\
&= \Sigma(\mu^*) + \tfrac{1}{2}\mathrm{Tr}\left\{\nabla_z^2 \Sigma(z)\big|_{z=\mu^*}\Sigma^*\right\}, \\
\mathrm{Var}_{Z^*}[\mu(Z^*)] &\approx \mathrm{Var}_{Z^*}[\mu(\mu^*) + \nabla_z \mu(z)^\top\big|_{z=\mu^*}(Z^* - \mu^*)] = \nabla_z \mu(z)^\top\big|_{z=\mu^*}\Sigma^* \nabla_z \mu(z)\big|_{z=\mu^*}.
\end{aligned} \tag{1.22}$$

Secondly, we have ***Exact Moment Matching***, if we using Gaussian kernel and assume Gaussian input, then the integral of 1.20 can be derived:

$$\mathbb{E}_{z^*}[\mu(z^*)] = \int \mu(z^*)p(z^*)dz^* = \beta^\top \mathbf{l}$$

where $\beta = (\mathscr{K}_{Z,Z} + \sigma_w^2 I_N)^{-1} Y$ and $\mathbf{l} = [l_1, \ldots, l_N]$ is function of z_i, μ^*, Σ^* and hyperparameters.

$$\begin{aligned}
\mathrm{Var}_{g_{Z^*}}[g_{Z^*} \mid \mathbf{Y}] &= \mathbb{E}_{Z^*}[\Sigma(Z^*)] + \mathbb{E}_{Z^*}[\mu^2(Z^*)] - (\mathbb{E}_{Z^*}[\mu(Z^*)])^2 \\
&= \beta^\top L \beta - \mathrm{Tr}\left\{\left(\mathscr{K}_{Z,Z} + \sigma_w^2 I_N\right)^{-1} L\right\} - \left(\beta^\top \mathbf{l}\right)^2,
\end{aligned}$$

Thirdly, we have ***Sigma-point propagation***, we compute sigma-point for $z^* \sim (\mu^*, \Sigma^*)$, we denote by $\{\bar{z}_j^*\}_{j=0}^{2n_z}$, where n_z is the dimensionality of z.

$$\begin{aligned}\bar{z}_0^* &= \mu^* \\ \bar{z}_j^* &= \mu^* + \sqrt{n_z + \lambda_{mm}}[\text{chol}(\Sigma^*)]_j, \quad \text{for } j = 1, \dots, n_z \\ \bar{z}_j^* &= \mu^* - \sqrt{n_z + \lambda_{mm}}[\text{chol}(\Sigma^*)]_j, \quad \text{for } j = n_z + 1, \dots, 2n_z\end{aligned} \tag{1.23}$$

where $[\text{chol}(\Sigma^*)]_j$ is the $j - th$ column of the Cholesky factorization of matrix Σ^*, and λ_{mm} is a user-defined parameter representing how far the sigmapoints are spread from the mean. By viewing the g as $g = \mu + \tilde{w}$, where $\tilde{w} \sim \mathcal{N}(0, \Sigma)$ is treated as ”process noise”, we can approximate $\mathbb{E}_{g_{Z^*}}[g_{Z^*}|Y]$ by evaluating μ on the sigma-points:

$$\mathbb{E}_{g_{z^*}}[g_{z^*}|Y] \approx \mu_{\text{sp}} = \sum_{j=0}^{2n_z} W_j^m \mu(\bar{z}_j^*).$$

and the variance is:

$$\text{Var}_{g_{z^*}}[g_{z^*}|Y] \approx \sum_{j=0}^{2n_z} W_j^v (\mu(\bar{z}_j^*) - \mu_{\text{sp}})(\mu(\bar{z}_j^*) - \mu_{\text{sp}})^T + \Sigma(\mu^*).$$

the weights are choose to sum to 1, how to tune them can be found in [117]. This method can be compared to MC method, first one choose the points deterministicly, second one randomly (but it does not need the assumption of GP prior).

Fourth is the numerical approximation, 1.19 can be computed using MC. [403] tackles the computational issue, where the sampling process is guided by a measure of correlation to the previous evaluations.

The above methods handle uncertainty in inference, does not exploit them in training. So here we talk about them, and they are connected to treate latent variables as optimization variables in the second section [108]. Or we can use Gaussian mixture models [397].

2. Robust uncertainty propagation, try to give a deterministic bound rather than stochastic one for the propagation. Model the $z \in \mathcal{E}(z_p, Q_p)$ which is an ellipsoidal confidence region, and it can maintain its shape throgh the $\tilde{g} = g_{\text{nom}}(\bar{z}) + \nabla_z g_{\text{nom}}(z)|_{z=\bar{z}}^{\mathsf{T}}(z - \bar{z}) + \mu(\bar{z})$. This method maybe too conservative. More detail please refer to [407].

3. Overcomming the independece assumption, iteratively recondition the GP model, enabling a scenario-based [411] control design. while this may face the curse of dimensionality, other method try to approximate it with finite horizon [415], [391].

1.1.6 Closed-loop Safty Guarantees

- Bounded support assumption: ***compact set*** is bounded and the bound is inside it (for Euler space, other general case need to use Open coverage to define). In this assumption, the error is bounded with $C_\delta = \{\varepsilon : \mathbb{P}(\varepsilon \in C_\delta) \geq \mathbf{1} - \delta\}$, where δ is a small probability.
- Robust-in-probability, define $\triangle = \{g(x, u) \in \mathcal{G}(x, u) \text{ for all } (x, u) \in \mathcal{Z}\}$, this method guarantee that $\mathbb{P}(\triangle) \geq p$. Here $\mathcal{G}$ is high probability region. Ensuring that

$$\mathbb{P}(h_j(x_i, u_i) \leq 0,\ \forall i \geq 0,\ \forall \{j\}_{j=1}^{n_h} \mid \triangle) = 1$$

  we can then make sure:

$$\begin{aligned}&\mathbb{P}(h_j(x_i, u_i) \leq 0,\ \forall i \geq 0,\ \forall \{j\}_{j=1}^{n_h}) \\ &\geq \mathbb{P}(h_j(x_i, u_i) \leq 0,\ \forall i \geq 0,\ \forall \{j\}_{j=1}^{n_h} \mid \triangle) P(\triangle) \\ &\geq p\end{aligned}$$

  such a region construction can be found in [62], a complete survey can be found in [64].
- Sampling-based approach, [413], [414], [427] samples GP to get tightened constraints. This method is not conservative, but it is computational heavy.
- other results, generally consider point-wise-in-time chance constraints, can be seen in [432]...

1.1.7 Discussion

1. MPC with Scalable GP,