Predicting Next-Step Loss Distributions in PILCO

Rowan McAllister

February 18, 2016

This document is still under construction.

Contents

1 Introduction

1.1 Active Learning for PILCO

We wish to increase PILCO's data-efficiency by 'upgrading' PILCO from being a *passive* reinforcement learning (RL) algorithm to an *active* RL algorithm. Currently, PILCO greedily optimises an expected loss function (a sum of state-costs over time) using a probabilistic dynamics model to predict distributions of trajectories, evaluated by the loss function. In RL this is known as pure-exploitation. But using a *probabilistic* dynamics model allows us to not only predict a loss-mean for a given policy, but a full loss *distribution* (approximated as Gaussian). I.e. we can predict the loss-mean *and* a loss-variance of a particular policy given the dynamics data we currently have. PILCO currently optimises the loss-mean only, and ignores the loss-variance, which can otherwise help guide exploration and evaluate potential information gains from testing a particular policy.

1.2 Using Loss Distributions for Active Learning

We can use loss-distributions to help balance exploitation and exploration. For example, let the loss distribution for a policy with parameterisation θ be:

$$L_\theta \sim \mathcal{N}(\mu_\theta, \sigma_\theta^2). \tag{1}$$

What policy parameterisation $\theta \in \Theta$ should we next test? We might use some type of UCB algorithm to select $\theta \leftarrow \underset{\theta}{\text{argmin}}[\mu_\theta - \beta\sigma_\theta]$, for some exploitation-exploration trade-off scalar parameter β.

Unfortunately, such a loss distribution is not sufficient for our active learning problem, and might mislead our UCB type algorithm. This is because the total loss-variance σ_θ^2 is the result of multiple sources of randomness: 1) permanent system stochasticity, 2) uncertainties due to our temporarily ignorance of the system because of the limited data we have. We only care about the type of variance we can reduce, i.e. our ignorance of the system. **Q:** How can we get a handle on system-ignorance, and disregard system-stochasticity? **A:** We can do so by fantasising about potential future data, to predict if how extra data generated from policy parameterisation θ might reduce uncertainty in the loss in the next timestep. If there is not predicted effect, then we predict all loss uncertainty is due to inherent system uncertainty and no more data can help reduce that. This is (possibly?) equivalent to understanding how the loss-mean might change in the next timestep.

1.3 Predicting Next-Step Loss Distributions

So how can we implement an analytic solution for the question: '*how would adding (uncertain and correlated) fantasy dynamics data affect the loss distribution?*'. Even if one only cares about the loss-mean, they would still be interested in the variance of the loss-mean at some future point in time. I.e. if we have some data at the current point in time t then our predictive loss-mean at time t is certain. However, if we consider adding new fantasy data in the future at time $t+1$, then our prediction conducted at time t of the loss-mean at future time $t+1$ has a variance, because the future loss is dependent on uncertain data we are yet to collect.

Even without uncertain fantasy-data, the system trajectories are uncertain due to inherent stochasticity and ignorance. Yet, the uncertain fantasy data adds a new type of uncertainty to these trajectories. To distinguish the uncertainty caused by the fantasy data when simulating system trajectories, we can formulate the system-state as a hierarchically-uncertain, and composed of two uncertainty types we wish to distinguish.

The latter uncertainty type results from how the fantasy data would affect the GP posterior mean and variance We approximate using the expected affect on the GP posterior variance w.r.t. the fantasy data, i.e. assuming the variance is affected the same way across all possible fantasy data independent on the sampled fantasy data. Doing so, we get access to all the nice identities in the GPH document. From there, `simulate` would call the hierarchical version of `propagate` H many times as before.

2 Problem Statement

We wish to minimise the loss L_θ of our control system, w.r.t. the controller parameterisation $\theta \in \Theta$. The loss is a cumulative-cost over from time $t = 0$ to horizon $t = H$, where individual costs covary. Using a probabilistic dynamics model, we compute loss *distributions*:

$$L_\theta \sim \mathcal{N}(\mu_\theta, \sigma_\theta^2) \quad \leftarrow \quad \texttt{simulate}(\theta) \tag{2}$$

The variance σ_θ^2 is the result of:

1. process noise
2. observation noise
3. ignorance (a lack of infinite data everywhere)

To explore the parameterisation space Θ effectively using gradient-based optimisation methods, we wish to make use of our uncertainty information σ_θ^2, to explore areas which have both a low mean μ_θ and high variance σ_θ^2.

However, we only care about variance is because of the ability to gain information. Both the process noise and the observation noise are inherent to the system, and no amount of learning will reduce their contributions to variance σ_θ^2. We therefore wish to distinguish the variance contribution solely due to our ignorance of the system (we'll denote this ν_θ). How can we get a handle on this variance-because-of-ignorance ν_θ?

Loss-Variance from Ignorance: One approximation of ν_θ follows the line of thought: if I just have enough data, then ν_θ would be driven to zero. Whether or not we can drive ν_θ to zero in the next timestep, let us *assume* that what we learn in the next rollout is all we will ever learn.

We being by generating fantasy intra-correlated data u, using `simulate`. We should be able to add in uncertain fantasy data, and marginalise it out again to revert back to our current loss distribution. I.e.:

$$\sigma_{\theta,t}^2 \approx \mathbb{E}_u[\sigma_{\theta,t+1}^2] + \mathbb{V}_u[\mu_{\theta,t+1}], \tag{3}$$

where the approximation is that $\mathbb{C}_u[\sigma^2_{\theta,t+1}, \mu_{\theta,t+1}] = 0$. OK we now estimate how much our new data u would reduce uncertainty. (noting that any reduction of uncertainty must be the result of improving system ignorance, it cannot be the result of system stochasticity changing which is assumed fixed). The reduced uncertainty is:

$$\Delta\sigma^2_\theta = \sigma^2_{\theta,t} - \mathbb{E}_u[\sigma^2_{\theta,t+1}]. \tag{4}$$

Note, according to equation 3 this is equivalent to asking how would the uncertain fantasy data affect the variance of the future loss-mean? i.e.

$$\Delta\sigma^2_\theta = \mathbb{V}_u[\mu_{\theta,t+1}]. \tag{5}$$

So this tells us we can still be greedy about optimising the loss-mean, except now we can incorporate the value of information gained in one timestep, and how it might affect the loss-mean at the next timestep which we wish to optimise. This is a myopic belief lookahead RL algorithm, that assumes we can learn from one more timestep worth of datum, but then learn nothing further at time $t+2$ onwards etc.

3 Solutions to Incorporate Fantasy Data

Using `simulate`, we generate predictive state distributions x_t from time $t = 0$ to time $t = H$, from which we can fantasise about what the future data might look like. We will denote the expectation and variance of each state distributions along as $\mathbb{E}[x_t]$ and $\mathbb{V}[x_t]$. Note states can also covary: $\mathbb{C}[x_t, x_{t+\tau}]$.

3.1 Solution A:
Certain Fantasy Data with Certain Location at State-Mean

The current solution (currently implemented in the code) generates fantasy data using `simulate` to generate state distributions at each time step, and then assumes the fantasy data it will generate is non-noisy data points located at the state-mean of each state distribution:

$$\text{Solution A:} \qquad u_{0:H} \sim \mathcal{N}(\mathbb{E}[x_{0:H}], 0) \tag{6}$$

3.2 Solution B:
Uncertain Fantasy Data with Distribution equal to State Distribution

We can make use of simulate's uncertainty information $\mathbb{V}[x_{0:H}]$. This solution assumes the fantasy data points were observed noisily, with noise of each datum u_t equal to the predictive variance that simulate provides at each time step: $u_t \sim \mathcal{N}(\mathbb{E}[x_t], \mathbb{V}[x_t])$. After incorporating uncertain data into our GP dynamics model, we then make predictions based on our fantasy-dynamics-model which uses combinations of the regular (real) data combined with the very noisy fantasy data:

$$\text{Solution B:} \qquad u_{0:H} \sim \mathcal{N}(\mathbb{E}[x_{0:H}], \mathbb{V}[x_{0:H}]) \tag{7}$$

3.3 Solution C:
Certain Fantasy Data with Uncertain Locations, MC-Sampled from State-Distribution

The problem with Proposed Solution B, is that it does not reflect what really happens. The data we are about to receive is not a single set of very-noisy datum (one per time step), but rather a sample from a distribution over plausible sets of non-noisy data points. Proposed Solution B is making a pessimistic assuming about the expected information gain, but in fact we know the expected information gain will be much higher, we simply do not know what the information is yet!

What would a 'more correct' solution look like? Consider taking N MC-samples from the predicted state distribution, i.e. N sets of plausible fantasy data $u_{0:H}^{(n)}$ for $n \in [1, N]$. With each sample $u_{0:H}^{(n)}$, we retrain a corresponding fantasy dynamics models (where the n'th fantasy-dynamics-model is trained from using the current real data $\mathbf{x}$ *and* the n'th fantasy data $u_{0:H}^{(n)}$). We can then make mean and variance predictions using each of the N models, which gives then allows us to estimate variance of the loss-means and expectation of the loss-variances etc. So each MC-sample follows:

$$\text{Solution C:} \qquad u_{0:H}^{(n)} \sim \mathcal{N}(\bar{u}_{0:H}^{(n)}, 0), \qquad \bar{u}_{0:H}^{(n)} \stackrel{\text{iid}}{\sim} \mathcal{N}(\mathbb{E}[x_{0:H}], \mathbb{V}[x_{0:H}]), \tag{8}$$

4 Solution D:
Certain Fantasy Data with Uncertain Location in State-Distribution

How can we do even better than proposed solution C? Here we attempt to formulate the analytically analogue of solution C.

4.1 Assumptions

The following approximations concern how the GP posterior changes w.r.t. additional fantasy data $u_{0:H}$. We also use the shorthand of input $u = u_{0:H-1}$ and corresponding targets $v = u_{1:H}$.

1. We assume the hyper-parameters remain fixed.

4.2 How Additional Fantasy Data Affects GP Posterior

Let $\mathbf{x}$ represent the training points comprising the entire history of real-observations, x_* represent a test-point, $f_* = f(x_*)$ represent a test-target, and random fantasy data points u with targets v we would combine with the rest of our training data. The posterior distribution is

$$\mathbb{E}[f_*] \quad = \quad k_{*x} K^{-1} y \; = \; k_{*x} \beta, \tag{9}$$

$$\mathbb{V}[f_*] \quad = \quad k_{**} - k_{*x} K^{-1} k_{*x}, \tag{10}$$

and $k(\cdot, \cdot)$ is the kernel, K and K^{-1} precomputed at the root node, where

- $K = k(\mathbf{x}, \mathbf{x}) + \sigma_n^2 I$, be the current covariance matrix, size $n \times n$
- K' be the covariance matrix with additional points u, size $(n+H) \times (n+H)$

- $k_{ux} = k(u, \mathbf{x})$, and $k_{*x} = k(x_*, \mathbf{x})$, etc

We can represent our updated-belief covariance matrix K' after observing additional datum $\{u, v\}$ as:

$$K' = \begin{bmatrix} K & k_{xu} \\ k_{ux} & k_{uu} \end{bmatrix} \tag{11}$$

$$(K')^{-1} = \begin{bmatrix} \hat{P} & \hat{Q} \\ \hat{Q}^\top & z_{uu}^{-1} \end{bmatrix} \tag{12}$$

where

$$z_{uu} = k_{uu} - k_{ux}K^{-1}k_{xu} = \mathbb{V}[f_u], \tag{13}$$

$$\hat{Q} = -K^{-1}k_{xu}z_{uu}^{-1} \tag{14}$$

$$\hat{Q}^\top = -z_{uu}^{-1}k_{ux}K^{-1}, \tag{15}$$

$$\hat{P} = K^{-1} + K^{-1}k_{xu}z_{uu}^{-1}k_{ux}K^{-1}, \tag{16}$$

using results from GPML page 201. Now let:

- $\mathbb{E}[f_*]'$ be the updated $\mathbb{E}[f_*]$
- $\mathbb{V}[f_*]'$ be the updated $\mathbb{V}[f_*]$
- $y' = \begin{bmatrix} y \\ v \end{bmatrix}$
- $k'_{*x} = [k_{*x}\ ,\ k_{*u}]$
- $z_{*u} = k_{*u} - k_{*x}K^{-1}k_{xu}$

4.2.1 Affect on GP Posterior Mean

The updated GP mean at test point x_* with additional observations $\{u, v\}$ is:

$$\begin{aligned}
\mathbb{E}[f_*]' &= k'_{*x}(K')^{-1}y' \\
&= [k_{*x}, k_{*u}] \begin{bmatrix} \hat{P} & \hat{Q} \\ \hat{Q}^\top & z_{uu}^{-1} \end{bmatrix} \begin{bmatrix} y \\ v \end{bmatrix} \\
&= (k_{*x}\hat{P} + k_{*u}\hat{Q}^\top)y + (k_{*x}\hat{Q} + k_{*u}z_{uu}^{-1})v \\
&= (k_{*x}K^{-1} + k_{*x}K^{-1}k_{xu}z_{uu}^{-1}k_{ux}K^{-1} - k_{*u}z_{uu}^{-1}k_{ux}K^{-1})y + (-k_{*x}K^{-1}k_{xu}z_{uu}^{-1} + k_{*u}z_{uu}^{-1})v \\
&= k_{*x}K^{-1}y + (k_{*x}K^{-1}k_{xu} - k_{*u})(z_{uu}^{-1}k_{ux}K^{-1})y + (-k_{*x}K^{-1}k_{xu} + k_{*u})z_{uu}^{-1}v \\
&= \mathbb{E}[f_*] - z_{*u}z_{uu}^{-1}k_{ux}K^{-1}y + z_{*u}z_{uu}^{-1}v \\
&= \mathbb{E}[f_*] + z_{*u}z_{uu}^{-1}(v - k_{ux}K^{-1}y) \\
&= \mathbb{E}[f_*] + z_{*u}z_{uu}^{-1}(v - \mathbb{E}[f_u])
\end{aligned} \tag{17}$$

4.2.2 Affect on GP Posterior Variance

The updated GP variance at test point x_* with additional observations u is:

$$
\begin{aligned}
\mathbb{V}[f_*]' &= k_{**} - k'_{*x}(K')^{-1}k'_{x*} \\
&= k_{**} - [k_{*x}, k_{*u}] \begin{bmatrix} \hat{P} & \hat{Q} \\ \hat{Q}^\top & z_{uu}^{-1} \end{bmatrix} \begin{bmatrix} k_{x*} \\ k_{u*} \end{bmatrix} \\
&= k_{**} - (k_{*x}\hat{P} + k_{*u}\hat{Q}^\top)k_{x*} - (k_{*x}\hat{Q} + k_{*u}z_{uu}^{-1})k_{u*} \\
&= k_{**} - (k_{*x}K^{-1} + k_{*x}K^{-1}k_{xu}z_{uu}^{-1}k_{ux}K^{-1} - k_{*u}z_{uu}^{-1}k_{ux}K^{-1})k_{x*} - (-k_{*x}K^{-1}k_{xu}z_{uu}^{-1} + k_{*u}z_{uu}^{-1})k_{u*} \\
&= (k_{**} - k_{*x}K^{-1}k_{x*}) - (k_{*x}K^{-1}k_{xu} - k_{*u})(z_{uu}^{-1}k_{ux}K^{-1})k_{x*} - (-k_{*x}K^{-1}k_{xu} + k_{*u})z_{uu}^{-1}k_{u*} \\
&= \mathbb{V}[f_*] + z_{*u}z_{uu}^{-1}k_{ux}K^{-1}k_{x*} - z_{*u}z_{uu}^{-1}k_{u*} \\
&= \mathbb{V}[f_*] - z_{*u}z_{uu}^{-1}(k_{u*} - k_{ux}K^{-1}k_{x*}) \\
&= \mathbb{V}[f_*] - z_{*u}z_{uu}^{-1}z_{u*} \qquad (18)
\end{aligned}
$$

We may just wish to make life simpler by using the expectation (under random $\{u, v\}$) of the new predictive variance, rather than the variance-of-variance.

4.3 How does this fit together?

OK so we need to understand how:

1. additional fantasy data $u_{0:H}$ affects our predictions (see Sec 4.2),
2. changes in the mean and variance of the simulated states effect a distribution in L-mean at time $t + 1$, i.e. a one-step lookahead.

Previously in Sec 3 we talked of `simulate` generating predictive state distributions $p(x_t)$ from time $t = 0$ to time $t = H$. Now we talk about the altered predictive state distributions $p(x'_t)$, which have been altered because of (random) additional fantasy data $u_{0:H}$.

- **Time t=0:**

  We begin by noting $p(x'_0) = p(x_0)$, since any additional data has no effect on the initial distribution (the distribution over possible system starting positions).

- **Time t=1:**

  Before considering fantasy data, let us discuss the normal case without.

  Without fantasy data:

  To predict forwards one timestep, we input distribution $p(x_0)$ into the GP dynamics model. Even with a certain input of $p(x_0) = \delta(x_0)$, the output $p(x_1) \sim \mathcal{N}(\mu_1, \Sigma_1)$ will be uncertain. So for uncertain input $p(x_0) \sim \mathcal{N}(\mu_0, \Sigma_0)$, the original non-filtered PILCO computes uncertain $p(x_1)$ by grouping both sources of uncertainty: $\mu_1 = \mathbb{E}_{x_0}[\mathbb{E}_f[f(x_0)]]$, and $\Sigma_1 = \mathbb{V}_{x_0}[\mathbb{E}_f[f(x_0)]] + \mathbb{E}_{x_0}[\mathbb{V}_f[f(x_0)]]$.

  With fantasy data, and certain input $p(x'_0) = \delta(x'_0)$:

  To introduce (uncertain) fantasy data is to introduce a new form of uncertainty. Note: we should be able to add fantasy data in, and marginalise it out again to return to aforementioned case distribution. So the marginal state probabilities of $p(x'_t)$ are not expected(?) to change for any $p(x_t)$. So in the case of certain input $p(x'_0) = \delta(x'_0)$, we have $p(x'_1) = \mathcal{N}(\mu_1, \Sigma_1)$, where $\mu_1 = \mathbb{E}_u[\mathbb{E}_f[f(x'_0)]]$, and $\Sigma_1 = \mathbb{V}_f[f(x'_0)] = \mathbb{V}_u[\mathbb{E}_f[f(x'_0)]] + \mathbb{E}_u[\mathbb{V}_f[f(x'_0)]]$.

However, we can *decompose* uncertainty in $p(x'_1)$ and track two separate sources of variance $p(x'_1)$, to understand the affect each has in simulation. For time $t = 1$, we are simply concerned with how to output a binary-decomposition of $p(x'_1)$. We will worry about how to input it into a GP later at time $t + 2$.

Let us distinguish uncertainty in $p(x'_1) \sim \mathcal{N}(\mu_1, \Sigma_1)$ caused by fantasy data, by denoting it Σ_1^u, such that $\Sigma_1 = \Sigma_1^u + \Sigma_1^x$. OK, how can we compute Σ_1^u? Using results from Section 4.2, we have an expression of GP output $p(x'_1)$ as a relative change from $p(x_1)$, given uncertain data $u_{0:H}$. The (uncertain) fantasy data affects our GP mean and our GP variance (in an uncertain way).

If u was nonexistent, or placed far away, then we would expect $\Sigma_1^u = 0$, i.e. no variance in Σ_1 is attributed to u. In such a case, we would expect two things 1) the mean to be unaffected: $\mathbb{V}_u[\mathbb{E}_f[f(x'_0)]] = 0$, and 2) the variance remains unreduced: $\mathbb{V}_u[\mathbb{E}_f[f(x'_0)]] = \Sigma_1$. Now if placing u instead completely determined the new posterior to something certain, then we expect $\Sigma_1^u = \Sigma_1$ (i.e. all variance in Σ_1 attributed to u), where $\mathbb{V}_u[\mathbb{E}_f[f(x'_0)]] = \Sigma_1$ and variance collapses $\mathbb{V}_u[\mathbb{E}_f[f(x'_0)]] = 0$.

So perhaps we can attribute $\Sigma_1^u = \mathbb{V}_u[\mathbb{E}_f[f(x'_0)]] = \Sigma_1 - \mathbb{E}_u[\mathbb{V}_f[f(x'_0)]]$.

With fantasy data, and uncertain input $p(x'_0) \sim \mathcal{N}(\mu_0, \Sigma_0)$:

With uncertain input $p(x'_0) \sim \mathcal{N}(\mu_0, \Sigma_0)$, we get output $p(x'_1) \sim \mathcal{N}(\mu_1, \Sigma_1)$ where $\mu_1 = \mathbb{E}_{x'_0}[\mathbb{E}_f[f(x'_0)]]$, and $\Sigma_1 = \mathbb{V}_{x'_0}[\mathbb{E}_f[f(x'_0)]] + \mathbb{E}_{x'_0}[\mathbb{V}_f[f(x'_0)]]$. OK, so to be consistent with the delta input case, we want a similar effect when $p(x'_0)$ is tight.

So perhaps we can instead attribute $\Sigma_1^u = \mathbb{V}_u[\mathbb{E}_{x'_0}[\mathbb{E}_f[f(x'_0)]]] = \Sigma_1 - \mathbb{E}_u[\mathbb{V}_{x'_0}[\mathbb{E}_f[f(x'_0)]] + \mathbb{E}_{x'_0}[\mathbb{V}_f[f(x'_0)]]]$. This has the desired behaviour with small Σ_0. What about large Σ_0?

- **Time t=2+:**

  OK so we have a form for sub-components $\Sigma_1 = \Sigma_1^u + \Sigma_1^x$. Now, how do we continue to recurse? Well $p(x'_1)$, which was the output at time $t = 0$, is also the input to the GP at time $t = 1$. So now, not only are the GP outputs have dual-variances, but so too the GP inputs.

  How can we deal with this? Well the GPH framework handles GP prediction of hierarchically-uncertain inputs. In GPH, inputs have distribution over distributions, which takes the input mean as random but approximates the input variance as fixed (for simplicity). It turns out the variance and mean-variance are quite similar, the marginal always adds up to the same, even if the variances are switched, so we can perhaps use GPH to help us out with our dual-variance task.

  So we'll need to do a full simulate up to horizon H using GPH instead of the normal non-hierarchical noisy-GP prediction.

So, with input $p(x_1') = \mathcal{N}(\mu_1, \Sigma_1^u + \Sigma_1^x)$, We treat it hierarchically for GPH as $p(x_1') = \mathcal{N}(\mathcal{N}(\mu_1, \Sigma_1^u), \Sigma_1^x)$.

OK so we can input this into the GPH to output: $p(x_2') = \mathcal{N}(\mathcal{N}(\mu_2, \hat{\Sigma}_2^u), \hat{\Sigma}_2^x)$. But is this the end of the story? We still need to inform the GP at this next stage, how much the new fantasy data affected this transition. It is simply propagated the affects of the fantasy data from the previous step forwards as if the fantasy data only existed in the first timestep.

So we can propagate forward and then 'take from $\hat{\Sigma}_2^x$' and 'give to $\hat{\Sigma}_2^u$'. The equal take-give is because of the conserved quantity in Σ_t, i.e. marginalising out any u should result in the original variance Σ_t. OK so how much do we take from $\hat{\Sigma}_2^x$ then?

Ignoring the current binary-decomposition, we would take $\mathbb{V}_u[\mathbb{E}_{x_1'}[\mathbb{E}_f[f(x_1')]]]$ equal to $\Sigma_2 - \mathbb{E}_u[\mathbb{V}_{x_1'}[\mathbb{E}_f[f(x_1')]] + \mathbb{E}_{x_1'}[\mathbb{V}_f[f(x_1')]]]$. Indeed this is what we would take if $\Sigma_1^u = 0$. How about if $\Sigma_1^u = \Sigma_1$, i.e. if the fantasy data was responsible for the entire input variance? Then surely this explains all subsequent variances: $\Sigma_t^u = \Sigma_t$ for $t \geqslant 2$. OK so, what about in-between, i.e. the input variance was explained 'half' of the fantasy data last timestep, and another half this round? Do we expect the fantasy data to explain 3/4 of the current variance? So if it explains proportion $\rho_{1:t-1}$ before (resulting from fantasy-data affects accumulated from all timesteps before from 1 to $t-1$) and ρ_t now (resulting from fantasy-data affects on just this timestep), then we expect total affect at this timestep, $\rho_{1:t}$, to satisfy the following (for any $a \in [0, 1]$ and $b \in [0, 1]$):

$\rho_{1:t-1}$	ρ_t	$\rho_{1:t}$
0	a	a
1	a	1
a	0	a
a	1	1
a	b	$1-(1-a)(1-b)$

So the rule (i.e. the function f_ρ) must be: $\rho_{1:t} = f_\rho(\rho_{1:t-1}, \rho_t) = \rho_{1:t-1} + \rho_t - \rho_{1:t-1}\rho_t$. Note this still satisfies $\rho_{1:t} \in [0, 1]$. OK so note $1-(1-a)(1-b) = a + b - ab \geqslant \max(a, b)$. So the total proportion $\rho_{1:t}$ is non-decreasing as time t progresses from 0 to H.

OK so our update is: $\Sigma_2^u = \hat{\Sigma}_2^u + \tilde{\Sigma}_2^u - \hat{\Sigma}_2^u\tilde{\Sigma}_2^u$, where $\tilde{\Sigma}_2^u = \mathbb{V}_u[\mathbb{E}_{x_1'}[\mathbb{E}_f[f(x_1')]]] = \Sigma_2 - \mathbb{E}_u[\mathbb{V}_{x_1'}[\mathbb{E}_f[f(x_1')]] + \mathbb{E}_{x_1'}[\mathbb{V}_f[f(x_1')]]]$. I'm not quite sure how the multivariate multiplication (i.e. $\hat{\Sigma}_2^u\tilde{\Sigma}_2^u$) should work, perhaps point-wise? Note a point-wise multiplication of two PSD matrices is also PSD (Schur product theorem).

OK so the $\hat{\Sigma}_2^u$ part is easy to compute, it is the GPH output. But what about $\tilde{\Sigma}_2^u$? We need to compute $\tilde{\Sigma}_2^u$ using some new math, which we do below.

4.4 Key Identities

4.4.1 The Main Equation

OK so we need a way of computing

$$\tilde{\Sigma}_2^{\mu} = \underbrace{\mathbb{V}_u[\underbrace{\mathbb{E}_{x_1'}[\mathbb{E}_f[f(x_1')]]}_{\mu_2}]}_{\text{part 1}} = \underbrace{\Sigma_2 - \mathbb{E}_u[\underbrace{\mathbb{V}_{x_1'}[\mathbb{E}_f[f(x_1')]] + \mathbb{E}_{x_1'}[\mathbb{V}_f[f(x_1')]]}_{\Sigma_2}]}_{\text{part 2}}\,. \qquad (19)$$

I.e. we can compute either part 1 or part 2, whichever is easiest. We'll use the GPH document to get the forms of μ_2 (Eq. 24) and Σ_2 (Eq. 27).

4.4.2 Previous Identities

Let's start with some identities already known. The GPH document has identities concerning kernel integration and GP-output with uncertain inputs:

$$\int q(x,t,\Lambda,V)\mathcal{N}(t|\mu,\Sigma)dt = q(x,\mu,\Lambda,\Sigma+V), \qquad (20)$$

$$\int q(x,t,\Lambda_a,V)\, q(t,x',\Lambda_b,V)\,\mathcal{N}(t|\mu,\Sigma)dt = Q(x,x',\Lambda_a,\Lambda_b,V,\mu,\Sigma), \qquad (21)$$

$$\int Q(x,x',\Lambda_a,\Lambda_b,0,\mu,V)\mathcal{N}(\mu|m,\Sigma)d\mu = Q(x,x',\Lambda_a,\Lambda_b,0,m,\Sigma+V).$$

$$k_a(x,x') = s_a^2 q(x,x',\Lambda_a,0). \qquad (22)$$

and
Consider making predictions from $a = 1,\ldots,E$ GPs at x^* with specification

$$p(x^*|m,\Sigma) \sim \mathcal{N}(m,\Sigma). \qquad (23)$$

We have the following expressions for the predictive mean, variances and input output covariances

$$\begin{aligned}
\mathbb{E}[f^*|m,\Sigma] &= \int \big(s_a^2\beta_a^\top q(x_i,x^*,\Lambda_a,0) + \theta_a^\top x^*\big)\mathcal{N}(x^*|m,\Sigma)dx^* \\
&= s_a^2\beta_a^\top q^a + \theta_a^\top m, \\
\mathbb{C}[x^*,f_a^*|m,\Sigma] &= \int (x^*-m)\big(s_a^2\beta_a^\top q(x,x^*,\Lambda_a,0) + \theta_a^\top x^*\big)\mathcal{N}(x^*|m,\Sigma)dx^* \\
&= s_a^2\Sigma(\Lambda_a+\Sigma)^{-1}(x-m)\beta_a q^a + \Sigma\theta_a \;=\; \Sigma C_a + \Sigma\theta_a, \\
\mathbb{V}[f_a^*|m,\Sigma] &= \mathbb{V}[\mathbb{E}[f_a^*|x^*]|m,\Sigma] + \mathbb{E}[\mathbb{V}[f_a^*|x^*]|m,\Sigma] \\
&= \mathbb{V}[s_a^2\beta_a^\top q(x,x^*,\Lambda_a,0) + \theta_a^\top x^*] + \delta_{ab}\mathbb{E}[s_a^2 - k_a(x^*,x)(K_a+\Sigma_\varepsilon^a)^{-1}k_a(x,x^*)] \\
&= s_a^2 s_b^2\big[\beta_a^\top(Q^{ab} - q^a q^{b\top})\beta_b + \delta_{ab}\big(s_a^{-2} - \mathrm{tr}((K_a+\Sigma_\varepsilon^a)^{-1}Q^{aa})\big)\big] + C_a^\top\Sigma\theta_b + \theta_a^\top\Sigma C_b \\
\text{where}\quad q_i^a &= q(x_i,m,\Lambda_a,\Sigma), \;\;\text{and}\;\; Q_{ij}^{ab} \;=\; Q\big(x_i,x_j,\Lambda_a,\Lambda_b,0,m,\Sigma\big).
\end{aligned}$$

And Section 4.2 has identities concerning relative changes to the GP posterior (mean and variance):

$$
\begin{aligned}
z_{uu} &= k_{uu} - k_{ux}K^{-1}k_{xu} = \mathbb{V}[f(u)], && (29)\\
z_{*u} &= k_{*u} - k_{*x}K^{-1}k_{xu}, && (30)
\end{aligned}
$$

so if $|u| = 1$ (with outer δ_{ab} around the expression?), then

$$
\begin{aligned}
\mathbb{E}_u[z_{uu}] &= \mathbb{E}[k_{uu} - k_{ux}K^{-1}k_{xu}],\\
&= s^2 - s^4(K^{-1} \odot Q),\\
&= s^2 - s^4 \operatorname{tr}(K^{-1}Q). && (31)
\end{aligned}
$$

4.4.3 New Basic Identities

Let's start with some basic identities. Let any pairwise fantasy points

$$
\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \sim \mathcal{N}\Big(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \sigma_1^2 & c \\ c & \sigma_2^2 \end{bmatrix} \Big) \tag{32}
$$

To integrate out an uncertain fantasy datum against a certain datum inside a kernel we use Eq. 20:

$$
\begin{aligned}
\mathbb{E}_u[k(u_1, x)] &= s_a^2 \mathbb{E}_u[q(u_1, x, \Lambda_a, 0)],\\
&= s_a^2 q(\mu_1, x, \Lambda_a, \sigma_1^2) && (33)
\end{aligned}
$$

Now what about when both inputs are uncertain and correlated? We may wish to answer this question to compute the mean of k_{uu} (Eq. 29).

$$
\begin{aligned}
\mathbb{E}_u[k(u_1, u_2)] &= s_a^2 \mathbb{E}_u[q(u_1, u_2, \Lambda_a, 0)]\\
&= s_a^2 \mathbb{E}_u[q(u_1 - u_2, 0, \Lambda_a, 0)]\\
&= s_a^2 q(\mu_1 - \mu_2, 0, \Lambda_a, \sigma_1^2 + \sigma_2^2 - 2c)\\
&= s_a^2 q(\mu_1, \mu_2, \Lambda_a, \sigma_1^2 + \sigma_2^2 - 2c), && (34)
\end{aligned}
$$

noting $(u_1 - u_2) \sim \mathcal{N}(\mu_1 - \mu_2, \sigma_1^2 + \sigma_2^2 - 2c)$. Now how computing the mean of the other term in Eq. 29, i.e. $k_{ux}K^{-1}k_{xu}$? Well using Eq. 42:

$$
\begin{aligned}
\mathbb{E}_u[k_{u_1x}K^{-1}k_{xu_2}] &= s^4 \operatorname{tr}\left(K^{-1}\mathbb{E}_u\left[q(x, u_1, \Lambda_a, 0)q(x', u_2, \Lambda_b, 0)\right]\right)\\
&= s^4 \operatorname{tr}\left(K^{-1} q\left(\begin{bmatrix} x \\ x' \end{bmatrix}, \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Lambda_a & 0 \\ 0 & \Lambda_b \end{bmatrix}, \begin{bmatrix} \sigma_1^2 & c \\ c & \sigma_2^2 \end{bmatrix}\right)\right), && (35)
\end{aligned}
$$

with similar derivation for Eq. 30. Thus we can now compute expectations for both Eq. 29 and Eq. 30.

4.4.4 Other New Identities

Expectation-of-expectation: (Eq.17) is (note I haven't split up the u-v indices in z_{uu} yet):

$$
\begin{aligned}
\mathbb{E}_u[\mathbb{E}[f_*]'] &= \mathbb{E}[f_*] + \int_u z_{*u} z_{uu}^{-1} \left[\int_v (v - \mathbb{E}[f_u])p(v|u)\right] p(u)\\
&= \mathbb{E}[f_*] + \int_u z_{*u} z_{uu}^{-1} [0]\, p(u)\\
&= \mathbb{E}[f_*] && (36)
\end{aligned}
$$

Variance-of-expectation: (Eq.17):

$$\begin{aligned}\mathbb{V}_u[\mathbb{E}[f_*]'] &= \mathbb{V}[z_{*u}z_{uu}^{-1}(\nu - \mathbb{E}[f_u])] \\ &= \mathbb{E}[z_{*u}z_{uu}^{-1}(\nu - k_{ux}\beta)(\nu - k_{ux}\beta)^\top z_{uu}^{-1}z_{u*}] - \mathbb{E}_u[\mathbb{E}[f_*]']^2 \end{aligned} \quad (37)$$

Expectation-of-variance: (Eq.18):

$$\mathbb{E}_u[\mathbb{V}[f_*]'] = \mathbb{V}[f_*] - \mathbb{E}_u[z_{*u}z_{uu}^{-1}z_{u*}] \quad (38)$$

4.5 Notes and Open Questions

1. We can use covariance information from each x_t to x_{t+1} etc to inform on correlations of u_t to u_{t+1} etc.

2. Looking at affect on expectation (Sec. 4.2.1), I assume the expectation (under u) of the updated mean does not change from the previous mean value because of the symmetric $(\nu - \mathbb{E}[f_u])$ term in Eq.17, i.e. $\mathbb{E}_u[\mathbb{E}[f_*]'] = \mathbb{E}[f_*]$? I'll just concentrate on the variance of the expectation.

3. Still need to incorporate additive linear models in GP posterior mean, e.g. $\theta^\top x_*$.

4. Note, dealing with the inverted term z_{uu}^{-1} might get tricky, but since $z_{uu} \leqslant \sigma_n^2$, we can perhaps approximate: $z_{uu} \approx \sigma_n^2$. Not sure - one some small datasets I tested with this can be out by a factor of 4. Also the above approximation is maybe OK in one dimension, not sure about more.

A More Detailed Derivations

From the gph.pdf document we have:

$$\int q(x, t, \Lambda, V)\mathcal{N}(t|\mu, \Sigma)dt = q(x, \mu, \Lambda, \Sigma + V), \quad (39)$$

$$\int q(x, t, \Lambda_a, V)\, q(t, x', \Lambda_b, V)\, \mathcal{N}(t|\mu, \Sigma)dt = Q(x, x', \Lambda_a, \Lambda_b, V, \mu, \Sigma), \quad (40)$$

$$\int Q(x, x', \Lambda_a, \Lambda_b, 0, \mu, V)\mathcal{N}(\mu|m, \Sigma)d\mu = Q(x, x', \Lambda_a, \Lambda_b, 0, m, \Sigma + V) \quad (41)$$

And note another way to express the left hand side of Eq. 40 is:

$$\begin{aligned}&\int q(x, t, \Lambda_a, V)\, q(t, x', \Lambda_b, V)\, \mathcal{N}(t|\mu, \Sigma)dt \\ &= \int q\left(\begin{bmatrix} x \\ x' \end{bmatrix}, \begin{bmatrix} t \\ t \end{bmatrix}, \begin{bmatrix} \Lambda_a & 0 \\ 0 & \Lambda_b \end{bmatrix}, 0\right) \mathcal{N}\left(\begin{bmatrix} t \\ t \end{bmatrix} | \begin{bmatrix} \mu \\ \mu \end{bmatrix}, \begin{bmatrix} \Sigma & \Sigma \\ \Sigma & \Sigma \end{bmatrix}\right) dt \\ &= q\left(\begin{bmatrix} x \\ x' \end{bmatrix}, \begin{bmatrix} \mu \\ \mu \end{bmatrix}, \begin{bmatrix} \Lambda_a & 0 \\ 0 & \Lambda_b \end{bmatrix}, \begin{bmatrix} \Sigma & \Sigma \\ \Sigma & \Sigma \end{bmatrix}\right)\end{aligned} \quad (42)$$