Active Filtering for Robot Tactile Learning

Hannes P Saal[1,2], Jo-Anne Ting[2], and Sethu Vijayakumar[2]

[1] *Neuroinformatics Doctoral Training Centre,* [2] *Institute of Perception, Action and Behaviour, School of Informatics, University of Edinburgh, UK*

Background

The problem in active learning is generally to determine controls x such that parameters $\boldsymbol{\theta}$ can be estimated as accurately as possible given observations y and an observation model $\mathbf{y} = \mathbf{f}(\boldsymbol{\theta}, \mathbf{x})$. In the sequential setting, a distribution over $\boldsymbol{\theta}$ is updated after each observation, and new controls x are selected to maximize the mutual information:

$$argmax_x I(\boldsymbol{\theta}; \mathbf{y}|\mathbf{x}) = argmax_x \iint p(\boldsymbol{\theta}, \mathbf{y}|\mathbf{x})\, log \frac{p(\boldsymbol{\theta}, \mathbf{y}|\mathbf{x})}{p(\boldsymbol{\theta})p(\mathbf{y}|\mathbf{x})}\, d\boldsymbol{\theta}\, d\mathbf{y}$$

We focus on problems with robotics applications, where we are faced with some additional constraints:

- **nonlinear** forward model that needs to be **learned from data**
- parameters, controls and observations are **continuous**
- decisions have to be taken **quickly**, sometimes in real-time
- controls and observations are typically **high-dimensional**

Under these constraints, solving these types of problems is often analytically intractable, and sampling methods quickly become untenable.

Our contribution: We incorporate an **active update** into two well-known classes of filters, namely **linear filters (A)**, which are fast but may be inaccurate, and **particle filters (B)**, which allow for multi-modal distributions. Both methods are used with a Gaussian process observation model.

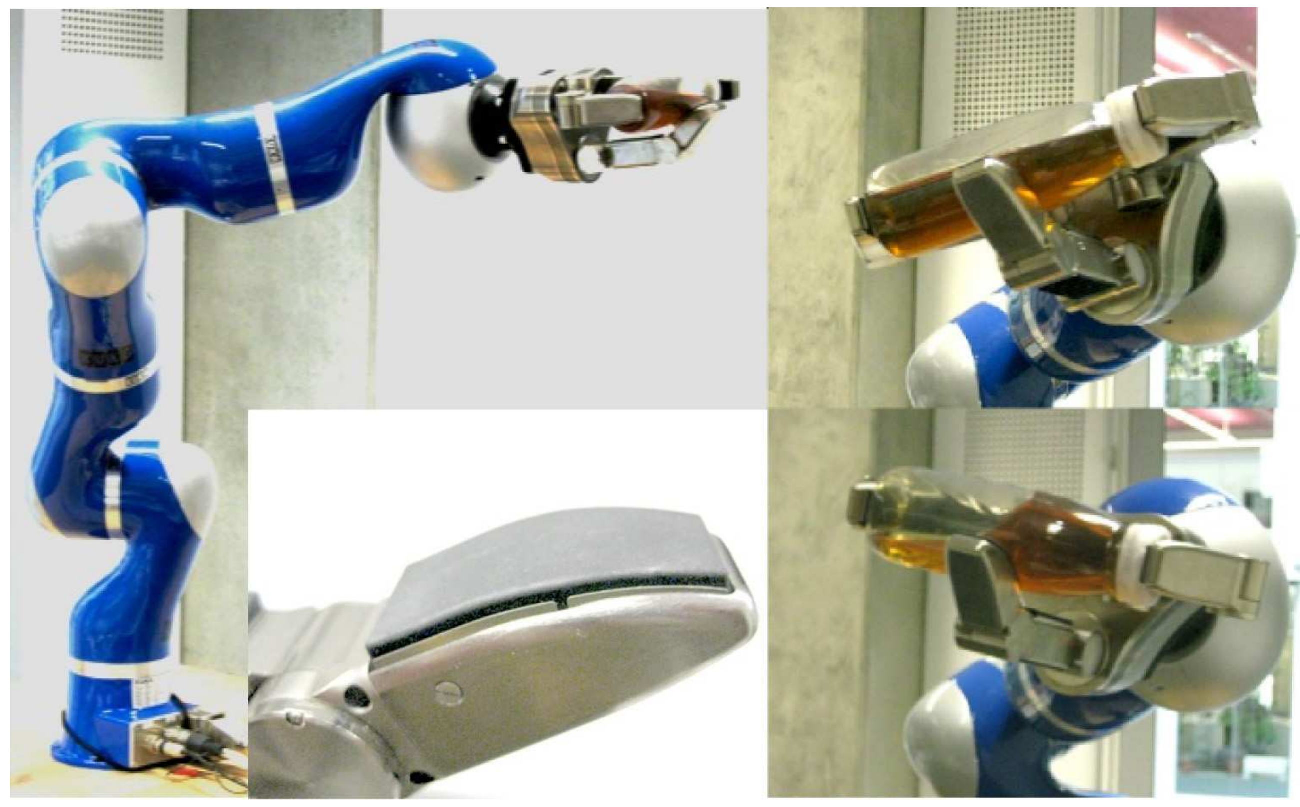

Figure 1: The robotic setup: The viscosity of the liquid in the bottle is to be estimated from tactile reponses of touch sensors while the bottle is being shaken.

Gaussian process forward model

We place a Gaussian Process (GP) prior over the observation model $\mathbf{f}(\boldsymbol{\theta}, \mathbf{x})$, using squared exponential kernels:

$$k_m(\mathbf{z}_p, \mathbf{z}_q) = \alpha_m^2 \exp\left(\frac{1}{2}(\mathbf{z}_p - \mathbf{z}_q)^T \mathbf{H}_m^{-1}(\mathbf{z}_p - \mathbf{z}_q)\right) + \sigma_m^2 \delta(\mathbf{z}_p, \mathbf{z}_q)$$

where z is the vector $[\boldsymbol{\theta} \;\; \mathbf{x}]^T$ $\mathbf{H}_m = \begin{pmatrix} \mathbf{H}_m^{\theta} & 0 \\ 0 & \mathbf{H}_m^{x} \end{pmatrix}$; and $\mathbf{H}_m^{\boldsymbol{\theta}}$ and $\mathbf{H}_m^{\mathbf{x}}$ are diagonal matrices.

A: Analytical Gaussian update

Here we assume that $p(\boldsymbol{\theta}_t)$ is represented as a unimodal Gaussian $\mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$, leading to the standard linear filter update equations:

$$\boldsymbol{\mu}_t = \boldsymbol{\mu}_{t-1} + \mathbf{C}_t^T \mathbf{S}_t^{-1}(\mathbf{y}_t^{obs} - \mathbf{m}_{t-1}) \quad \text{and} \quad \boldsymbol{\Sigma}_t = \boldsymbol{\Sigma}_{t-1} - \mathbf{C}_t^T \mathbf{S}_t^{-1} \mathbf{C}_t$$

where m is the marginal mean of y, S is the marginal covariance, and C is the cross-covariance between $\boldsymbol{\theta}$ and y.
Information $I(\boldsymbol{\theta}; \mathbf{y}|\mathbf{x})$ becomes maximal at $\mathbf{x}_t$:

$$argmax_x |\mathbf{C}_t^T \mathbf{S}_t^{-1} \mathbf{C}_t|$$

C and S can be calculated analytically [1,2]:

$$\begin{aligned} m_m &= \mathbf{q}_m(\mathbf{x})^T \boldsymbol{\beta}_m \\ S_{mn} &= \boldsymbol{\beta}_m^T \mathbf{Q}_{mn}(\mathbf{x}) \boldsymbol{\beta}_n - m_m m_n + \delta(m-n)\left(\alpha_m^2 - tr(\mathbf{K}_m^{-1}\boldsymbol{\Gamma}_m)\right) \\ C_{mn} &= \mathbf{Z}_m^T(\mathbf{x})\boldsymbol{\beta}_m - \mu_n m_m \end{aligned}$$

where the expressions for $\boldsymbol{\beta}_m$ and $\boldsymbol{\Gamma}_m$ are independent of x.
We note that $\mathbf{q}_m$ and $\mathbf{Q}_{mn}$ can be written as the Schur product of two factors:

$$\mathbf{q}_m = \mathbf{q}_m^{\theta} \circ \mathbf{q}_m^{x} \quad \text{and} \quad \mathbf{Q}_{mn} = \mathbf{Q}_{mn}^{\theta} \circ \mathbf{q}_m^{x}(\mathbf{q}_n^{x})^T$$

resulting in a quick update during the optimization of x, which can be done by standard gradient based methods at each timestep t.

B: MC-sampling based update

This approach uses a particle filter to represent $p(\boldsymbol{\theta})$. We use a quadratic divergence measure [3] instead of standard mutual information, to speed up computation:

$$\begin{aligned} \mathbf{I}_Q(\boldsymbol{\theta}; \mathbf{y}) &= \iint (p(\boldsymbol{\theta}, \mathbf{y}) - p(\boldsymbol{\theta})p(\mathbf{y}))^2\, d\mathbf{y}\, d\boldsymbol{\theta} \\ &= \iint \left[(p(\boldsymbol{\theta})p(\mathbf{y}|\boldsymbol{\theta}))^2 + p(\boldsymbol{\theta})^2 p(\mathbf{y})^2 - 2p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})^2 p(\mathbf{y})\right] d\mathbf{y}\, d\boldsymbol{\theta} \\ &:= V_1 + V_2 - 2V_3 \end{aligned}$$

This measure allows us to solve integrals over y analytically and evaluates to:

$$\begin{aligned} V_1 + V_2 &= \sum_{p=1}^{P} w_p^2 (4\pi)^{-\frac{d_y}{2}} |\boldsymbol{\Phi}(\boldsymbol{\theta}_p, \mathbf{x})|^{-\frac{1}{2}} + \left[\sum_{p=1}^{P} w_p^2\right] \sum_{a=1}^{P}\sum_{b=1}^{P} w_a w_b G(\boldsymbol{\nu}_{ab}, \boldsymbol{\Phi}_{ab}) \\ V_3 &= \sum_{a=1}^{P}\sum_{b=1}^{P} w_a^2 w_b G(\boldsymbol{\nu}_{ab}, \boldsymbol{\Phi}_{ab}) \end{aligned}$$

where $\boldsymbol{\nu}_{ab} = \boldsymbol{\nu}(\boldsymbol{\theta}_a, \mathbf{x}) - \boldsymbol{\nu}(\boldsymbol{\theta}_b, \mathbf{x})$, $\boldsymbol{\Phi}_{ab} = \boldsymbol{\Phi}(\boldsymbol{\theta}_a, \mathbf{x}) + \boldsymbol{\Phi}(\boldsymbol{\theta}_b, \mathbf{x})$, $G(\mathbf{r}, \mathbf{R}) = (2\pi)^{-0.5 d_y} |\mathbf{R}|^{-0.5} \exp(-0.5\mathbf{r}^T\mathbf{R}^{-1}\mathbf{r})$, and $\boldsymbol{\nu}(\boldsymbol{\theta}, \mathbf{x})$ and $\boldsymbol{\Phi}(\boldsymbol{\theta}, \mathbf{x})$ are the GP predictive mean and variance respectively. We use gradient-based methods for optimisation at each timestep.

Evaluation in simulation

We tested both analytical Gaussian and MC sampling updates on simulated data and found improved speed of convergence using the active strategy as opposed to the passive one.

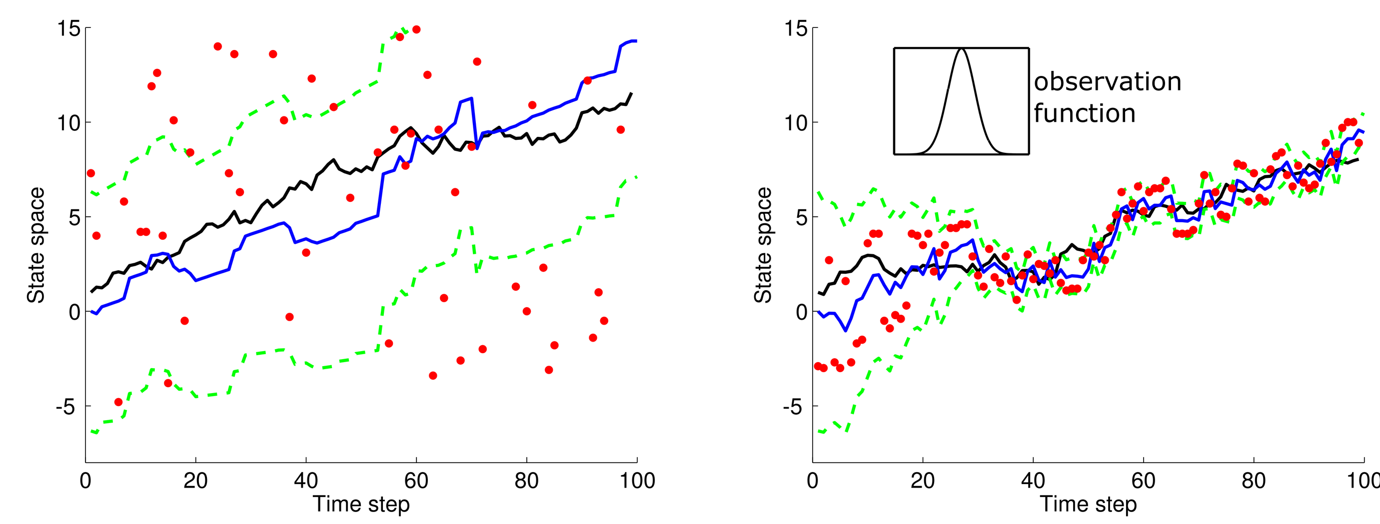

Figure 2: Tracking an object with linear dynamics and nonlinear (bump-shaped) observation function. Black: Target, Blue: mean of estimation, Green: confidence intervals, Red dots: x. Left: Passive filter using random controls. Right: Active filter maxiziming information in each step

Evaluation on robotic system

Task: Estimate viscosity of liquids by shaking bottles
Goal of active filtering: determine best shaking directions, angles and frequency sequentially so as to converge to the correct viscosity as quickly as possible.

- using DLR light-weight 7DOF arm with attached Schunk SDH2 7DOF hand, with mounted touch sensors on the fingers (486 texels in total)
- 16–23D outputs (y): tactile response in frequency spectrum
- 1–5D controls (x): e.g. shaking frequency, 3D direction and length, orientation of bottle
- 300–3500 training points for forward model (using standard or sparse GPs)
- updates within 0.5s including information maximization.

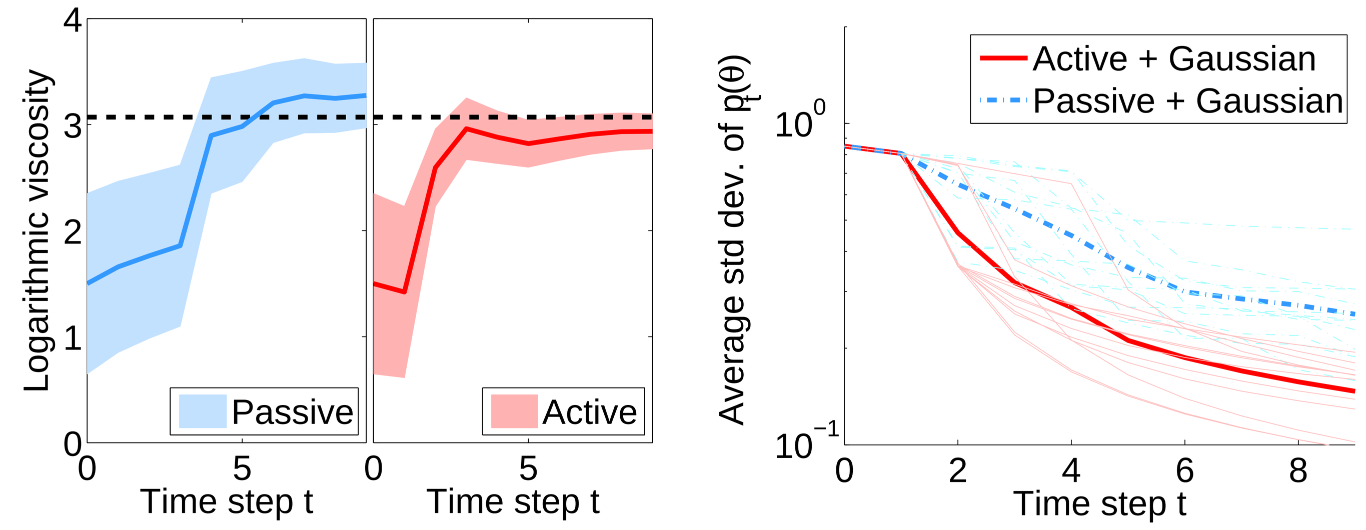

Figure 3: Left: Trial runs for passive (blue) and active (red) updates. Right: Reduction of uncertainty (i.e. standard deviation) over time for both passive and active updates.

References
[1] Girard et al., *NIPS*, 2003. [2] Deisenroth et al., *ICML*, 2009. [3] Torkkola, *JMLR*, 2005.

Acknowledgements: This study is supported by the EPSRC/MRC and the EU-funded Sensopac project. Contact: Hannes P Saal (h.saal@sms.ed.ac.uk).