Sequential kernel herding: Frank-Wolfe for particle filtering

Simon Lacoste-Julien, Fredrik Lindsten, Francis Bach (Lacoste-Julien et al., 2015)

Geelon So, agso@eng.ucsd.edu
Sampling/optimization reading group — June 2, 2021

Localization problem

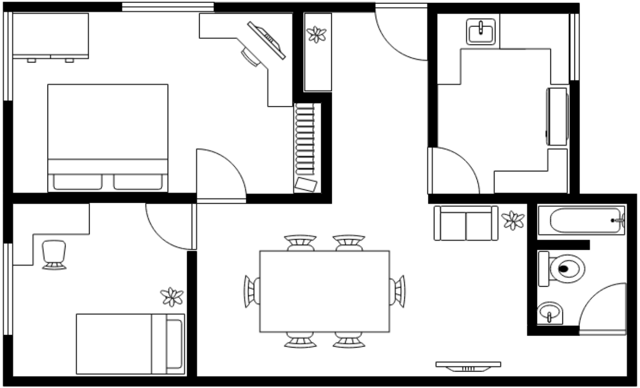

Figure 1: Suppose you're dropped into a new location. Given a map, can you explore your surroundings to figure out where on the map you are?

Localization example

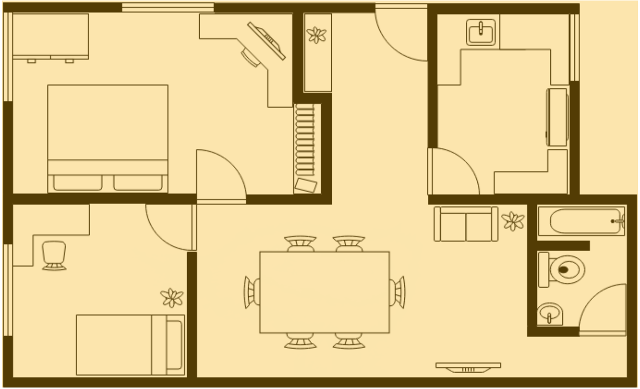

Figure 2: Estimate your initial position X_0 with a uniform prior $\pi(x_0)$.

Localization example

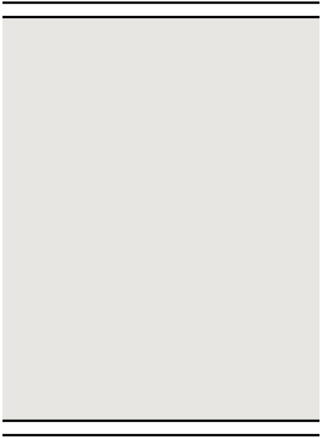

Figure 3: Opening your eyes, you see an image Y_0, which is a blank wall.

Localization example

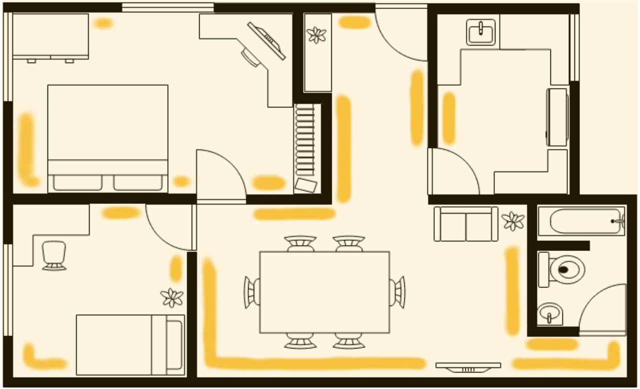

Figure 4: Having seen Y_0, you can update your posterior $p(x_0 \mid Y_0)$. A darker color corresponds to greater probability.

Localization example

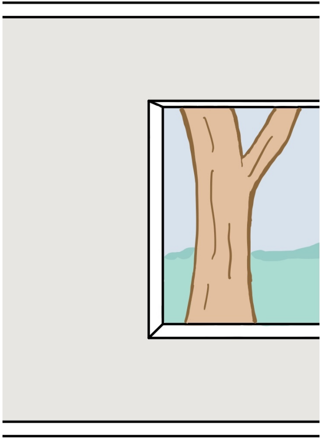

Figure 5: You move to the right, and now see a new image Y_1, a part of a window.

Localization example

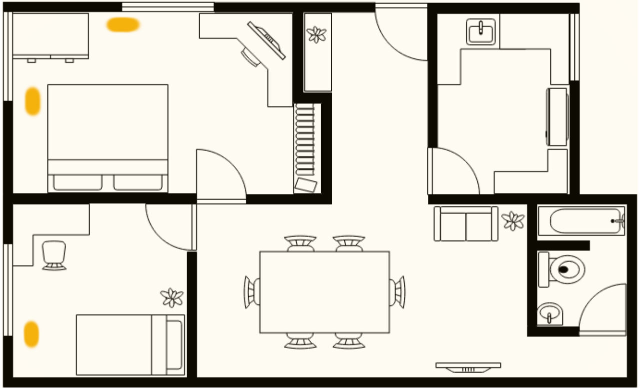

Figure 6: Having seen Y_1, update your posterior $p(x_0, x_1 \mid Y_0, Y_1)$.

Localization problem

Setting

- $\mathcal{X}$ is set of possible positions and orientations (unknown without GPS/compass)
- $\mathcal{Y}$ is the set of possible images you can see in the house

Problem

- After seeing $Y_1, \ldots, Y_t$, estimate posterior distribution on $X_1, \ldots, X_t$,

$$p(x_{0:t} \mid Y_{0:t}).$$

State-space models (SSM)

Definition

*A **state-space model**, or a general state-space hidden Markov model, is a probabilistic model on the state space $\mathcal{X}$ and observation space $\mathcal{Y}$ satisfying:*

$$
\begin{aligned}
X_t \mid X_{0:(t-1)} &\sim p(X_t \mid X_{t-1}) \\
Y_t \mid X_{0:t} &\sim p(Y_t \mid X_t),
\end{aligned}
$$

where $X_t \in \mathcal{X}$ and $Y_t \in \mathcal{Y}$ are the latent state variable and observation at time t.

$$
\begin{array}{ccccccc}
X_0 & \longrightarrow & X_1 & \longrightarrow & X_2 & \longrightarrow & \cdots \\
\downarrow & & \downarrow & & \downarrow & & \\
Y_0 & & Y_1 & & Y_2 & & \cdots
\end{array}
$$

The filtering problem

Filtering problem: given observations $Y_{0:t}$ from an SSM, estimate the posterior:

$$p(x_{0:t} \mid Y_{0:t}).$$

Computational issue

- The normalization term for computing Bayes' rule is often intractable:

$$p(x_{0:t+1} \mid Y_{0:t+1}) = \frac{p(Y_{t+1} \mid x_{t+1}) p(x_{t+1} \mid x_t) p(x_{0:t} \mid Y_{0:t})}{\int_{\mathcal{X}} p(Y_{t+1} \mid x'_{t+1}) p(x'_{t+1} \mid x_t) p(x_{0:t} \mid Y_{0:t}) \, dx'_{t+1}}.$$

$$p(x_{0:t+1} \mid Y_{0:t+1}) = \frac{p(Y_{t+1} \mid x_{t+1}) p(x_{t+1} \mid x_t) p(x_{0:t} \mid Y_{0:t})}{\int_{\mathcal{X}} p(Y_{t+1} \mid x'_{t+1}) p(x'_{t+1} \mid x_t) p(x_{0:t} \mid Y_{0:t}) \, dx'_{t+1}},$$

$$r_{t+1}(x_{0:t+1}) = \frac{p(Y_{t+1} \mid x_{t+1}) p(x_{t+1} \mid x_t) r_t(x_{0:t})}{\int_{\mathcal{X}} p(Y_{t+1} \mid x'_{t+1}) p(x'_{t+1} \mid x_t) r_t(x_{0:t}) \, dx'_{t+1}},$$

$$r_{t+1}(x_{0:t+1}) = \frac{p(Y_{t+1} \mid x_{t+1}) p(x_{t+1} \mid x_t) r_t(x_{0:t})}{\cdots},$$

Particle filter

Main idea: approximate the joint predictive distribution

$$p_{t+1}(x_{0:t+1}) = p(x_{t+1} \mid x_t)p(x_{0:t} \mid Y_{0:t})$$

using a finite i.i.d. sample $X_{0:t}^{(i)}, \ldots X_{0:t+1}^{(N)}$,

$$\hat{p}_{t+1}(x_{0:t+1}) = \frac{1}{N}\sum_{i=1}^{N} \delta_{X_{0:t+1}^{(i)}}(x_{0:t+1}).$$

- To obtain closer approximation, increase sample size N.

Particle filter algorithm

Algorithm *Particle filter*
(∗ sequential Monte Carlo algorithm ∗)
Input: SSM $p(x_{t+1} \mid x_t)$ and observation likelihood $o_t(x_t) := p(Y_t \mid x_t)$
Initialize: prior distribution $\tilde{p}_0(x_0) := \pi(x_0)$
1. **for** $t = 0, 1, 2, \ldots,$
2. **do** sample data point $X_{0:t}^{(1)}, \ldots, X_{0:t}^{(N)} \sim \tilde{p}_t$ and set:

$$\hat{p}_t(x_{0:t}) = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_{0:t}^{(i)}}(x_{0:t}).$$

3. estimate joint filtering distribution $\hat{r}_t(x_{0:t}) = \dfrac{o_t(x_t)\hat{p}_t(x_{1:t})}{\mathbb{E}_{\hat{p}_t}[o_t]}$ to compute:

$$\tilde{p}_{t+1}(x_{t+1}, x_{0:t}) := p(x_{t+1} \mid x_t)\hat{r}_t(x_{0:t}).$$

Particle filter visualization

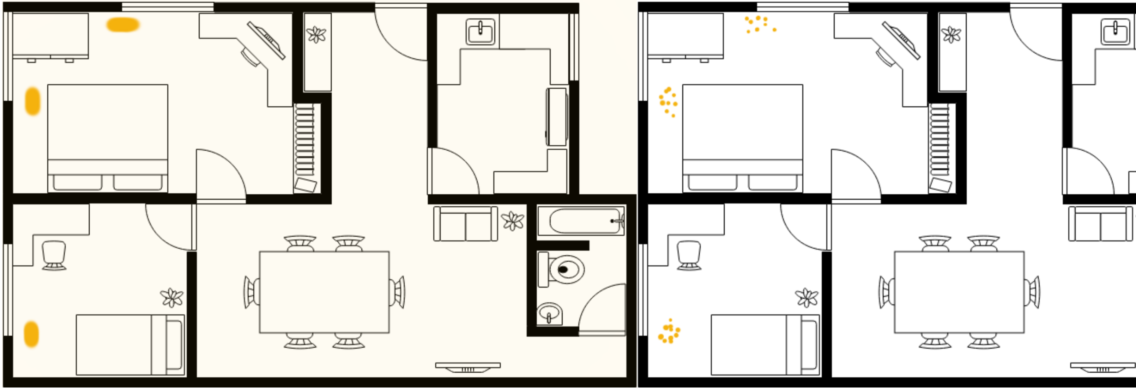

Figure 7: The particle filter algorithm approximates $p_{t+1}(x_{t+1}, x_{0:t})$ using samples $\hat{p}_{t+1}(x_{t+1}, x_{0:t})$; we need compute $p(Y_{t+1} \mid x_{t+1})$ only on those N particles.

Quality of approximator

Question: how large does sample size N have to be to get a good estimate?

- For our purposes, we want to get a good estimate:

$$\hat{r}_t(x_{0:t}) = \frac{p(Y_t \mid x_t)\hat{p}_t(x_{1:t})}{\mathbb{E}_{\hat{p}_t}[p(Y_t \mid x_t)]}.$$

- We need $\hat{p}_t$ close to p_t in the sense $\underset{\hat{p}_t}{\mathbb{E}}[f] \approx \underset{p_t}{\mathbb{E}}[f]$, for functions $p(Y_t \mid x_t)$ and $\mathbf{1}_{x_{1:t}}$.
- Hoeffding's implies approximation error decreases rate:

$$\left| \underset{\hat{p}_t}{\mathbb{E}}[f] - \underset{p_t}{\mathbb{E}}[f] \right| = O\left(\frac{1}{\sqrt{N}} \right).$$

Improving the estimator

Remark: estimating p_t using N i.i.d. draws may be wasteful

- instead of uniformly weighting N i.i.d. draws to construct $\hat{p}_t$,

  $$\hat{p}_t = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_{1:t}^{(i)}},$$

  can cleverly select and weight a handful of representatives? Intuitively, we might want to select points from the modes of the distribution.

Setting: if computing $p(Y_t \mid x_t)$ is expensive, we can try to get the approximation error to decrease faster than $\frac{1}{\sqrt{N}}$ by spending some additional compute to construct a clever:

$$\hat{p}_t = \sum_{i=1}^{N} w^{(i)} \delta_{X_{1:t}^{(i)}}.$$

Interlude: the kernel herding algorithm

Herding algorithm

> "Herding can be used to subselect a small collections of 'super-samples' from a much larger set of MCMC samples...In theory we would only need $\sqrt{N}$ samples to obtain the same order of error as N i.i.d. samples.
>
> *Chen et al. (2012)*

Herding visualization

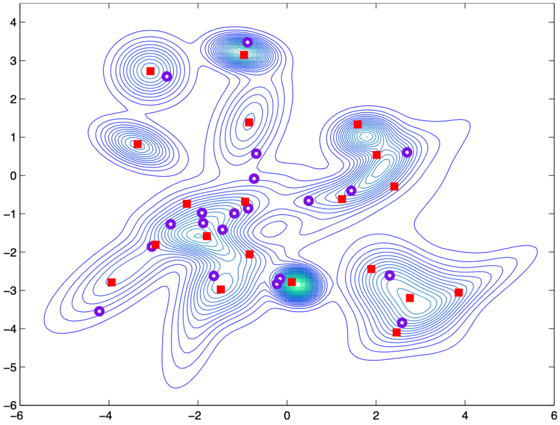

Figure 8: First 20 samples (i) from herding (red squares), and (ii) from i.i.d. draws (purple circles), Chen et al. (2012).

Approximating integrals

Problem: let $\mathcal{X}$ be a probability space with distribution p. We would like to approximate for a class of real-valued functions $f \in \mathcal{H}$,

$$\mathop{\mathbb{E}}_{p}[f] \approx \sum_{i=1}^{N} w^{(i)} f(X^{(i)}) = \mathop{\mathbb{E}}_{\hat{p}}[f],$$

for a set of points $X^{(1)}, \ldots, X^{(N)} \in \mathcal{X}$ and nonnegative weights $\sum w^{(i)} = 1$. That is,

$$\hat{p} = \sum_{i=1}^{N} w^{(i)} \delta_{X^{(i)}}.$$

- One choice is to let $X^{(i)}$'s be i.i.d. draws and $w^{(i)} = \frac{1}{N}$.

Reproducing kernel Hilbert space (RKHS)

Components of an RKHS

- Let $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a positive definite kernel,

$$\kappa(x, x') = \text{similarity between } x \text{ and } x'.$$

- Then, there exists an inner product space $\mathcal{H}$ and feature map $\phi : \mathcal{X} \to \mathcal{H}$ such that:

$$\kappa(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}}.$$

- $\mathcal{H}$ can be viewed as a family of real-valued functions f on $\mathcal{X}$,

$$f(x) \equiv \langle f, \phi(x) \rangle.$$

Approximating integrals on an RKHS

Problem: let $\mathcal{H}$ be an RKHS with respect to a fixed distribution p on $\mathcal{X}$. Construct $\hat{p}$,

$$\hat{p} = \sum_{i=1}^{N} w^{(i)} \delta_{X^{(i)}}$$

such that the *maximum mean discrepancy* is minimized:

$$\mathrm{MMD}(p, \hat{p}) := \sup_{\substack{f \in \mathcal{H} \\ \|f\|_{\mathcal{H}} \leq 1}} \left| \mathop{\mathbb{E}}_{p}[f] - \mathop{\mathbb{E}}_{\hat{p}}[f] \right|$$

Mean element

Notice that $\mathbb{E}_p[f] = \mathbb{E}_p\left[\langle f, \phi(x)\rangle\right] = \left\langle f,\, \mathbb{E}_p[\phi(x)]\,\right\rangle$.

- Define the **mean element** by:

$$\mu(p) = \mathop{\mathbb{E}}_{p}[\phi(x)].$$

- The maximum mean discrepancy is precisely:

$$\mathrm{MMD}(p, \hat{p}) = \sup_{\substack{f\in\mathcal{H}\\ \|f\|_{\mathcal{H}}\le 1}} \left|\left\langle f,\, \mu(p) - \mu(\hat{p})\,\right\rangle\right| = \|\mu(p) - \mu(\hat{p})\|_{\mathcal{H}},$$

where $\|\cdot\|_{\mathcal{H}}$ is the operator norm.

Takeaway: on an RKHS, to bound the discrepancy $\left|\mathbb{E}_p[f] - \mathbb{E}_{\hat{p}}[f]\right|$, just optimize:

$$\frac{1}{2}\|\mu(p) - \mu(\hat{p})\|^2_{\mathcal{H}}.$$

Frank-Wolfe optimization (or, kernel herding)

Optimization problem

- Let $\mathcal{M} \subset \mathcal{H}$ be the *marginal polytope*, $\mathcal{M} := \mathrm{cl}\left(\mathrm{conv}(\phi(\mathcal{X}))\right)$.
- Optimization objective: $\mathcal{J}(g) := \frac{1}{2}\|\mu(p) - g\|^2_{\mathcal{H}}$.

$$\min_{g \in \mathcal{M}} \mathcal{J}(g).$$

Frank-Wolfe algorithm: a gradient descent algorithm

- At each iteration k, optimize *linearization*:

$$\overline{g}_{k+1} \leftarrow \arg\min_{g \in \mathcal{M}} \langle \mathcal{J}'(g_k), g \rangle.$$

- Then update with learning rate γ_{k+1},

$$g_{k+1} \leftarrow (1 - \gamma_{k+1}) g_k + \gamma_{k+1} \overline{g}_{k+1}.$$

Properties of Frank-Wolfe

- When a linear function is optimized over a polytope, one of the vertices is an optimum. Thus, one of $\phi(\mathcal{X})$ optimizes:

$$\min_{x \in \mathcal{X}} \langle \mathcal{J}'(g_k), \phi(x) \rangle = \min_{g \in \mathcal{M}} \langle \mathcal{J}'(g_k), g \rangle.$$

- The Frank-Wolfe algorithm can just maintain set $\left\{ w^{(i)}, X^{(i)} \right\}$ for:

$$g_{k+1} = \sum_{i=1}^{k+1} w^{(i)} \phi(X^{(i)}) = \mathop{\mathbb{E}}_{\hat{p}}[\phi(x)].$$

N.B. *$\mathcal{H}$ can be very high or even infinite dimensional.*

Kernel herding algorithm

Algorithm *Kernel herding*
(∗ super-sampling algorithm ∗)
Input: distribution p, setting $\mathcal{J}(g) = \frac{1}{2}\|\mu(p) - g\|_{\mathcal{H}}^2$
Initialize: $g_0 = 0$

1. **for** $k = 0, 1, 2, \ldots, N-1$
2. **do** solve $X^{(k+1)} \leftarrow \underset{x \in \mathcal{X}}{\arg\min} \left\langle \mathcal{J}'(g_k), \phi(x) \right\rangle$
3. update $g_{k+1} \leftarrow (1 - \gamma_{k+1}) g_k + \gamma_{k+1} \phi(X^{(k+1)})$
4. set $w^{(k+1)} \leftarrow \gamma_{k+1}$ and $w^{(i)} \leftarrow (1 - \gamma_k) w^{(i)}$ for $i = 1, \ldots, t$
5. **return** estimator $\hat{p} \leftarrow \sum_{i \in [N]} w^{(i)} \delta_{X^{(i)}}$

Approximate vertex search

Algorithm *Kernel herding*

(∗ super-sampling algorithm ∗)

Input: distribution p, setting $\mathcal{J}(g) = \frac{1}{2}\|\mu(p) - g\|_{\mathcal{H}}^2$

Initialize: $g_0 = 0$

1. **for** $k = 0, 1, 2, \ldots, N-1$
2. **do** solve $X^{(k+1)} \leftarrow \underset{x \in \mathcal{X}}{\arg\min} \left\langle \mathcal{J}'(g_k), \phi(x) \right\rangle$
3. update $g_{k+1} \leftarrow (1 - \gamma_{k+1}) g_k + \gamma_{k+1} \phi(X^{(k+1)})$
4. set $w^{(k+1)} \leftarrow \gamma_{k+1}$ and $w^{(i)} \leftarrow (1 - \gamma_k) w^{(i)}$ for $i = 1, \ldots, t$
5. **return** estimator $\hat{p} \leftarrow \sum_{i \in [N]} w^{(i)} \delta_{X^{(i)}}$

Question: how do we solve $X^{(k+1)} \leftarrow \underset{x \in \mathcal{X}}{\arg\min} \left\langle \mathcal{J}'(g_k), \phi(x) \right\rangle$?

Rates using approximate vertex search

In the following theorem, assume:

- $\mu(p)$ is in the interior of $\mathcal{M}$, so there is some $r > 0$ such that $B\big(\mu(p), r\big) \subset \mathcal{M}$.
- $\mathcal{M}$ is bounded, so there is some $R > 0$ such that $\|g\| \leq R$ for all $g \in \mathcal{M}$.

Theorem (Lacoste-Julien et al. (2015))

Consider the kernel herding algorithm where an approximate vertex search is used:

$$\left\langle \mathcal{J}'(g_k), \bar{g}_{k+1} \right\rangle \leq \min_{g \in \mathcal{M}} \left\langle \mathcal{J}'(g_k), g \right\rangle + \delta,$$

where $\bar{g}_{k+1} = \phi(X^{(k+1)})$ *and* $\delta \geq 0$. *If* g_{k+1} *is updated with learning rate* $\gamma_{k+1} = \frac{1}{k+1}$, *then we obtain fast rates of convergence,* $\mathcal{J}(g_k) = O\left(\frac{1}{k^2}\right)$. *Specifically,*

$$\|g_k - \mu(p)\| \leq \frac{1}{k} \frac{2R^2}{r} + \frac{\delta}{r}.$$

Proof of theorem

0. Notation: let $\mu_p = \mu(p)$.
1. Note that since $\mathcal{J}(g) = \frac{1}{2}\|\mu_p - g\|^2$, minimizing the linear approximation at g_k is equivalent to minimizing:

$$\min_{g \in \mathcal{M}} \langle g_k - \mu_p, g - \mu_p \rangle.$$

   This objective is linear in the displacement from μ_p.

Proof of theorem (cont.)

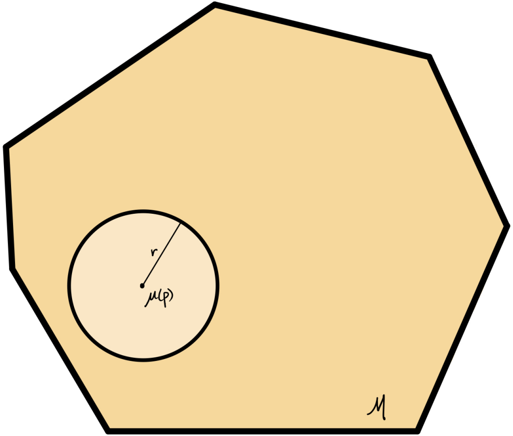

Figure 9: The polytope $\mathcal{M}$ contains the ball $B(\mu_p, r)$.

Proof of theorem (cont.)

2. It follows that optimum is upper bounded:

$$\min_{g \in \mathcal{M}} \left\langle g_k - \mu_p, g - \mu_p \right\rangle \leq \min_{g \in B\left(\mu_p, r\right)} \left\langle g_k - \mu_p, g - \mu_p \right\rangle$$
$$= -r \| g_k - \mu_p \|.$$

3. If we are guaranteed δ-accuracy, then:

$$\left\langle g_k - \mu_p, \overline{g}_{k+1} - \mu_p \right\rangle \leq -r \| g_k - \mu_p \| + \delta.$$

Proof of theorem (cont.)

4. Using the learning rate $\gamma_{k+1} = \frac{1}{k+1}$ in update:

$$g_{k+1} \leftarrow (1 - \gamma_{k+1}) g_k + \gamma_{k+1} \overline{g}_{k+1},$$

   we can compute explicitly:

$$\|g_{k+1} - \mu_p\|^2 = \frac{1}{(k+1)^2} \|(\overline{g}_{k+1} - \mu_p) + k(g_k - \mu_p)\|^2.$$

5. Rearrange the terms:

$$\left\|(k+1)(g_{k+1} - \mu_p)\right\|^2 = \left\|(\overline{g}_{k+1} - \mu_p) + k(g_k - \mu_p)\right\|^2.$$

$$\left\|(k+1)(g_{k+1} - \mu_p)\right\|^2 = \left\|(\overline{g}_{k+1} - \mu_p) + k(g_k - \mu_p)\right\|^2,$$

   where we let $v_k = k(g_k - \mu_p)$.

Proof of theorem (cont.)

5. Rearrange the terms (from previous page):

$$\left\|v_{k+1}\right\|^2 = \left\|(\bar{g}_{k+1} - \mu_p) + v_k\right\|^2. \qquad (\dagger)$$

6. Note that it suffices to prove an upper bound:

$$\|v_k\| = k\|g_k - \mu_p\| \leq \frac{2R^2 + k\delta}{r}$$

   to obtain result, $\|g_k - \mu_p\| \leq \frac{1}{k}\frac{2R^2+k\delta}{r}$.

7. Expanding $(\dagger)$, we get upper bound:

$$\|v_{k+1}\|^2 \leq \|v_k\|^2 + 2r\left[\frac{2R^2 + k\delta}{r} - \|v_k\|\right].$$

8. Follows by induction and noting that $\frac{R}{r} \leq 1$. □

Sequential kernel herding

Brief recap

1. We would like to solve the filtering problem:

$$r_t(x_{0:t}) = p(x_{0:t} \mid Y_{0:t}).$$

2. Using the particle filter algorithm, we maintain estimator $\hat{p}_{t+1}$ of predictive distribution:

$$p_{t+1}(x_{0:t+1}) = p(x_{t+1} \mid x_t)p(x_{0:t} \mid Y_{0:t})$$

using N i.i.d. draws to get $\hat{p}_{t+1} = \frac{1}{N} \sum_{i \in [N]} \delta_{X^{(i)}}$.

3. Kernel herding suggests that we may be able to use much fewer points to get a good estimator,

$$\hat{p}_{t+1} = \sum_{i \in [N]} w^{(i)} \delta_{X^{(i)}}.$$

Sequential kernel herding algorithm

Algorithm *Particle filter**Sequential kernel herding*

(∗ sequential Monte Carlo algorithm ∗)

Input: SSM $p(x_{t+1} \mid x_t)$ and observation likelihood $o_t(x_t) := p(Y_t \mid x_t)$

Initialize: prior distribution $\tilde{p}_0(x_0) := \pi(x_0)$

1. **for** $t = 0, 1, 2, \ldots,$
2. **do** sample data point $X_{0:t}^{(1)}, \ldots, X_{0:t}^{(N)} \sim \tilde{p}_t$ and set:

$$\hat{p}_t(x_{0:t}) = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_{0:t}^{(i)}}(x_{0:t}).$$

obtain estimator $\hat{p}_t$ from kernel herding algorithm on $\tilde{p}_t$ using N samples

3. estimate joint filtering distribution $\hat{r}_t(x_{0:t}) = \dfrac{o_t(x_t)\hat{p}_t(x_{1:t})}{\mathbb{E}_{\hat{p}_t}[o_t]}$ to compute:

$$\tilde{p}_{t+1}(x_{t+1}, x_{0:t}) := p(x_{t+1} \mid x_t)\hat{r}_t(x_{0:t}).$$

Consistency of SKH

Theorem (Informal, Lacoste-Julien et al. (2015))

Suppose the following is bounded:

$$\left\| \frac{p(x_{t+1} \mid x_t) p(Y_t \mid x_t)}{\mathbb{E}_p[p(Y_t \mid x_t)]} \right\|_{\mathcal{F}_t \otimes \mathcal{H}_t} \leq \rho,$$

where $\mathcal{F}_t : \mathcal{X}_{t+1} \to \mathbb{R}$ *and* $\mathcal{H}_t : \mathcal{X}_t \to \mathbb{R}$ *are function classes. If in the algorithm,*

$$\|\mu(\hat{p}_t) - \mu(\tilde{p}_t)\|_{\mathcal{H}_t} \leq \varepsilon,$$

then we have after T *iterations:*

$$\|\mu(\hat{p}_T) - \mu(p_T)\|_{\mathcal{H}_T} = \begin{cases} O(\rho^T \varepsilon) & \rho > 1 \\ O(T\varepsilon) & \rho = 1 \\ O(\varepsilon) & \rho < 1. \end{cases}$$

References

Yutian Chen, Max Welling, and Alex Smola. Super-samples from kernel herding. *arXiv preprint arXiv:1203.3472*, 2012.

Simon Lacoste-Julien, Fredrik Lindsten, and Francis Bach. Sequential kernel herding: Frank-wolfe optimization for particle filtering. In *Artificial Intelligence and Statistics*, pages 544–552. PMLR, 2015.