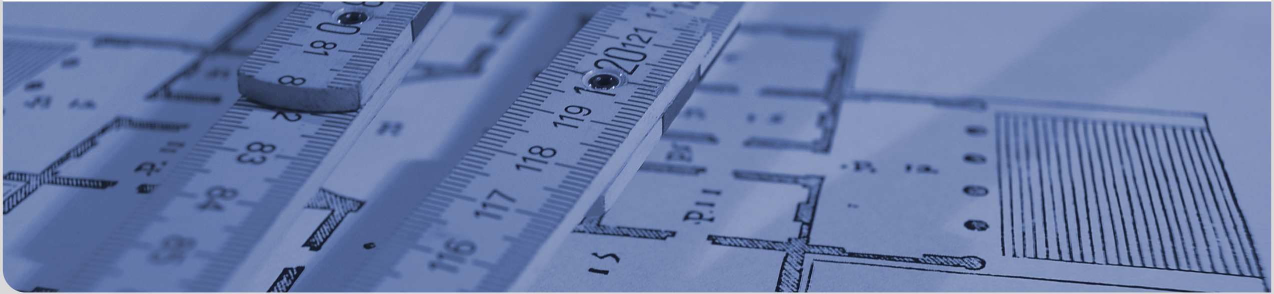
Karlsruhe Institute of Technology

Recursive Surrogate-Modeling for Stochastic Search

Intermediate Presentation - Bachelor Thesis

Philipp Theyssen | January 29, 2021

AUTONOMOUS LEARNING ROBOTS, INSTITUTE FOR ANTHROPOMATICS AND ROBOTICS , KIT DEPARTMENT OF INFORMATICS



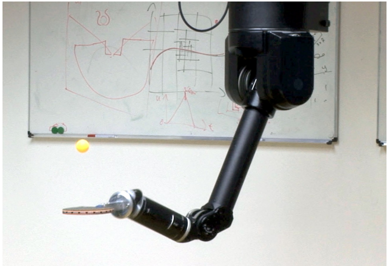
Outline

- Motivation
- Introduction
- Recursive Surrogate-Modeling for MORE
- Preliminary Results
- Outlook

Autonomous Robots: Key Challenges

Main tasks for autonomous robots:

- Modeling
- Predicting
- Decision making



Challenges

- fully autonomous (no human in the loop)
- uncertainty, sensor noise
- data efficiency

Sample Efficiency Problem

- **sample efficiency** for robot learning
- real-world samples are expensive in time, labor and finances
- robot hardware is expensive, needs careful maintenance
- working with non-industrial robots

Stochastic search

Stochastic search

- optimize objective function $f(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$
- learn policy $\pi(\mathbf{x})$ (search distribution) over parameter space of objective function
- black-box optimizer: only evaluate objective function (no gradients)
- iteratively update search distribution (policy)

Examples

- learn motor skills with Dynamic Movement Primitives

MORE Algorithm for Policy Search

MORE-Framework for Constraint Optimization Problem

$$\max_{\pi} \int \pi(\mathbf{x}) f(\mathbf{x}) dx \qquad \rightarrow \text{Maximize objective}$$

$$\text{s.t. KL}(\pi(\mathbf{x}) || \pi_{t-1}(\mathbf{x})) \leq \epsilon \qquad \rightarrow \text{bound between old and new policy}$$

$$H(\pi) \geq \beta \qquad \rightarrow \text{lower entropy bound}$$

$$1 = \int \pi(\mathbf{x}) dx \qquad \rightarrow \text{distribution requirement}$$

→ cannot evaluate integral in optimization problem

→ approximate objective with quadratic surrogate model

Solve Dual Function

Surrogate Model

$$f(\mathbf{x}) \approx \hat{f}(\mathbf{x}) = \mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{x}^T \mathbf{a} + a$$

New policy:

$$\pi_{t+1} = \mathcal{N}(\mu_{t+1}, \Sigma_{t+1})$$

$$\mu_{t+1} = B_t^{-1} b_t \qquad \Sigma_{t+1} = B_t^{-1}$$

with

$$B_t = (\eta \Sigma_t^{-1} + \mathbf{A})/(\eta + \omega)$$

$$b_t = (\eta \Sigma_t^{-1} \mu_t + \mathbf{a})/(\eta + \omega)$$

and η, ω Lagrangian multipliers

MORE Algorithm

Karlsruhe Institute of Technology

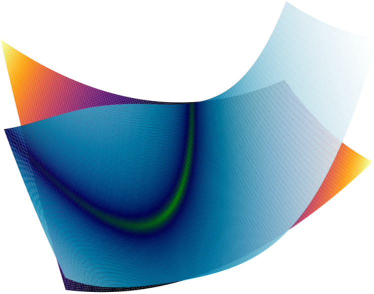
Iteration Step

1. Draw samples from search distribution + evaluate objective function
2. Estimate surrogate model using samples + values
3. Use surrogate model to solve optimization problem analytically
4. update search distribution

Surrogate Model

Karlsruhe Institute of Technology

- local approximation of objective function
- $\mathcal{O}(n^2)$ parameters to estimate
- KL-bound avoids over-exploitation of surrogate model



Surrogate Model

- current approach: learn new model in each iteration
- subsequent models are locally correlated

Central Question

Can we improve sample efficiency by recursively estimating the surrogate model?

Linear regression model

Karlsruhe Institute of Technology

Regression problem

$$y_k = \mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{x}^T \mathbf{a} + a + \epsilon_k$$
$$\theta = (\mathbf{A}, \mathbf{a}, a)$$

$\rightarrow$ estimate parameters θ from samples + objective values
$\rightarrow$ measurement noise ϵ_k is zero mean Gaussian $\epsilon_k \sim N(0, \sigma^2)$

Linear regression

Karlsruhe Institute of Technology

Bayesian Estimation

$$
\begin{aligned}
p(\theta) &= \mathsf{N}(\theta|\mathbf{m}_0, \mathbf{P}_0) && \rightarrow \text{prior} \\
p(y_k|\theta) &= \mathsf{N}(y_k|\mathbf{H}_k\theta, \sigma^2) && \rightarrow \text{likelihood} \\
p(\theta|y_{1:T}) &= \mathsf{N}(\theta|y_{1:T}) && \rightarrow \text{posterior}
\end{aligned}
$$

$\mathbf{H}_k$ matrix with samples (design matrix)

Least Squares

Least Squares solution

batch solution with application of Bayes' rule

$$p(\theta|y_{1:T}) \propto p(\theta) \prod_{k=1}^{T} p(y_k|\theta)$$

$$= \mathsf{N}(\theta|\mathbf{m}_0, \mathbf{P}_0) \prod_{k=1}^{T} \mathsf{N}(y_k|\mathbf{H}_k\theta, \sigma^2)$$

results in typical least squares solution for regression problem

$$\mathbf{m}_T = [\mathbf{P}_0^{-1} + \frac{1}{\sigma^2}\mathbf{H}^T\mathbf{H}]^{-1}[\frac{1}{\sigma^2}\mathbf{H}^T\mathbf{y} + \mathbf{P}_0^{-1}\mathbf{m}_0]$$

Recursive Least Squares

- dynamic model has to be Markov sequence
- use *previous posterior* as *prior*

$$p(\theta|y_1) = \frac{1}{Z_1} p(y_1|\theta) p(\theta)$$
$$p(\theta|y_{1:2}) = \frac{1}{Z_2} p(y_2|\theta) p(\theta|y_1)$$
$$\vdots$$
$$p(\theta|y_{1:T}) = \frac{1}{Z_T} p(y_T|\theta) p(\theta|y_{1:T-1})$$

Normalization term: $Z_k = \int p(\theta) p(y_k|\theta) d\theta$

Recursive Least Squares Equations

Karlsruhe Institute of Technology

matrix inversion lemma and introducing
temporary variables S_k and $\mathbf{K}_k$ yields:

$$\begin{aligned} S_k &= \mathbf{H}_k \mathbf{P}_{k-1} \mathbf{H}_k^T + \sigma^2 \\ \mathbf{K}_k &= \mathbf{P}_{k-1} \mathbf{H}_k^T S_k^{-1} \\ \mathbf{m}_k &= \mathbf{m}_{k-1} + \mathbf{K}_k [y_k - \mathbf{H}_k \mathbf{m}_{k-1}] \\ \mathbf{P}_k &= \mathbf{P}_{k-1} - \mathbf{K}_k S_k \mathbf{K}_k^T \end{aligned}$$

$\rightarrow$ equivalent to Kalman Filter without prediction step

Drift model

Assume parameters change over time:

$$p(\theta_k|\theta_{k-1}) = \mathsf{N}(\theta_k|\theta_{k-1}, \mathbf{Q})$$

before performing update step:

$$\mathbf{P}_k^- = \mathbf{P}_{k-1} + \mathbf{Q}$$

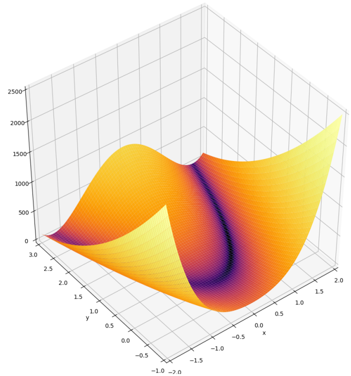
Setup

- try to use minimal amount of samples
- CMA-ES heuristics: $s = 4 + 3\lfloor \log(n) \rfloor$
- clusterwork 2.0 framework for hyper parameter tuning
- 20 repetitions

Test Functions

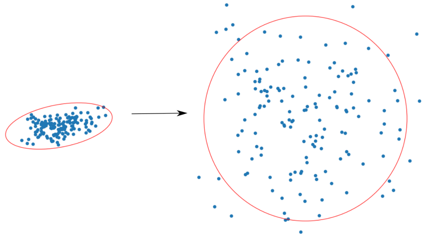
Rosenbrock function
(uni-modal):

$$f(\mathbf{x}) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$$



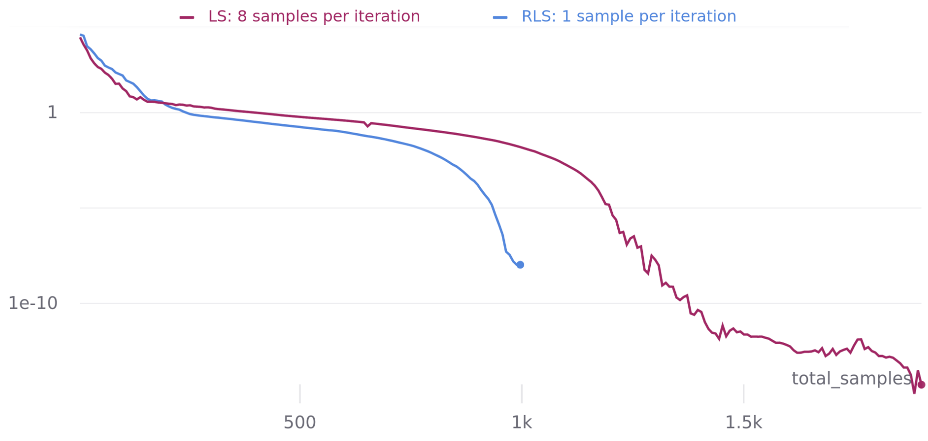
Data Preprocessing Techniques

- whitening transformation
  $\rightarrow$ high range of rewards problematic
- normalization of samples and reward

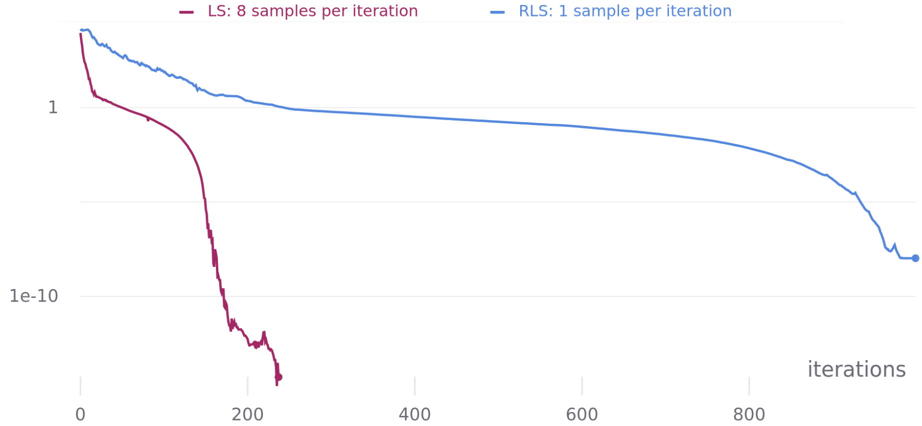


100 samples 2 dim Rosenbrock

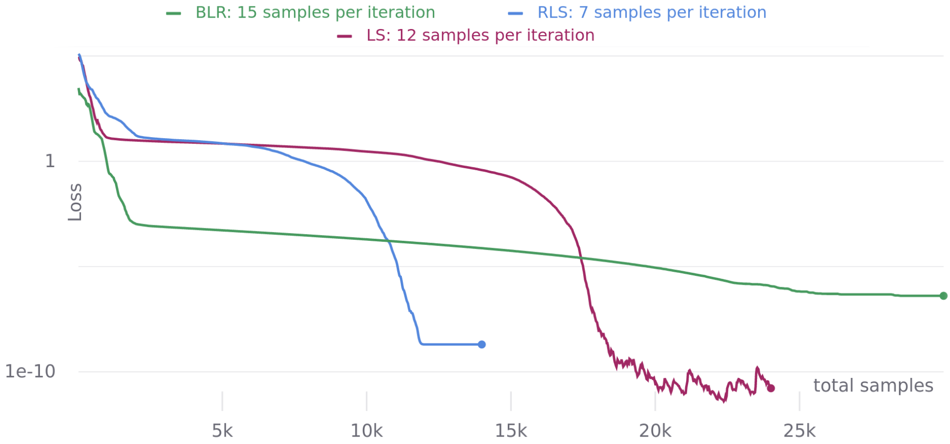
5 Dimensional Rosenbrock



5 Dimensional Rosenbrock



15 Dimensional Rosenbrock



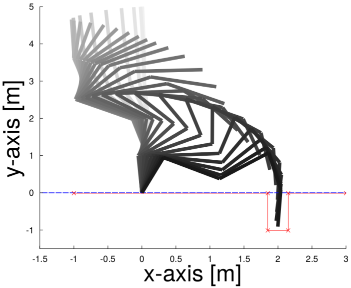
Outlook

Problems:

- constant parameter drift
- sample pool for RLS

Next steps:

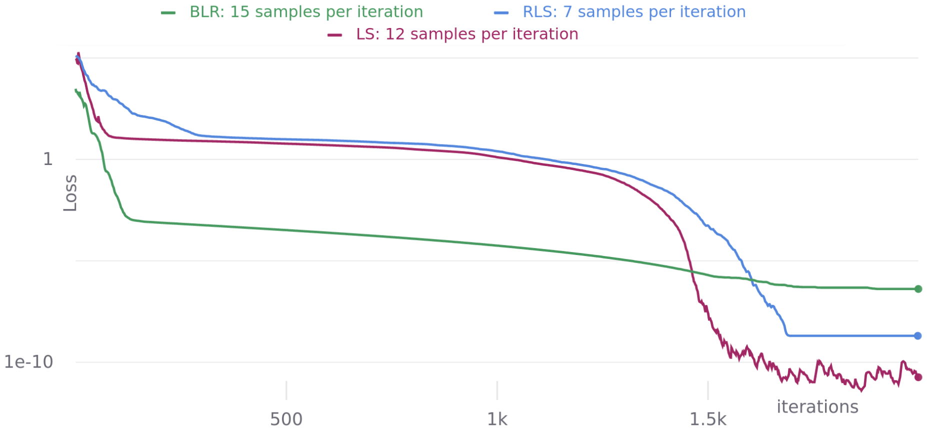
- performance on other higher dimensional/noisy objective functions?
- try to learn model parameter noise
- try planar reaching task



Thank you!

Discussion & Questions?

15 Dimensional Rosenbrock



Runtimes: LS $\sim$ 16 s, RLS $\sim$ 34s, BLR $\sim$ 11 min

Drift model

Parameters perform Gaussian random walk:

$$p(\theta_k|\theta_{k-1}) = \mathsf{N}(\theta_k|\theta_{k-1}, \mathbf{Q})$$

Given:

$$p(\theta_{k-1}|y_{1:k-1}) = N(\theta_{k-1}|m_{k-1}, P_{k-1})$$

the joint distribution is (Markov assumption):

$$p(\theta_k, \theta_{k-1}|y_{1:k-1}) = p(\theta_k|\theta_{k-1})p(\theta_{k-1}|y_{1:k-1})$$

Use *Chapman-Kolmogorov equation*:

$$p(\theta_k|y_{1:k-1}) = \int p(\theta_k|\theta_{k-1})p(\theta_{k-1}|y_{1:k-1})d\theta_{k-1}$$

MORE Algorithm details

- Kullback Leibler Divergence

$$D_{KL}(P||Q) = \int p(x) \log \frac{p(x)}{q(x)} dx$$

MORE Theory

$$\mathcal{L}(\pi, \eta, \omega) = \int \pi(\theta)\mathcal{R}_\theta d\theta \; + \; \eta\left(\epsilon - \int \pi(\theta) \log\frac{\pi(\theta)}{q(\theta)} d\theta\right) - \; \omega\left(\beta + \int \pi(\theta) \log(\pi(\theta)) d\theta\right)$$

Optimizing the Lagrangian

$$\pi^* = l(\eta, \omega) = \text{argmax}_\pi \mathcal{L}(\pi, \eta, \omega)$$

Least Squares Equations

Karlsruhe Institute of Technology

$$p(\theta|y_{1:T}) \propto p(\theta) \prod_{k=1}^{T} p(y_k|\theta)$$

$$= N(\theta|\mathbf{m}_0, \mathbf{P}_0) \prod_{k=1}^{T} N(y_k|\mathbf{H}_k\theta, \sigma^2)$$

Because prior and likelihood are Gaussian, the *posterior distribution* will also be Gaussian:

$$p(\theta|y_{1:T}) = N(\theta|\mathbf{m}_T, \mathbf{P}_T)$$

$$\mathbf{m}_T = [\mathbf{P}_0^{-1} + \frac{1}{\sigma^2}\mathbf{H}^T\mathbf{H}]^{-1}[\frac{1}{\sigma^2}\mathbf{H}^T\mathbf{y} + \mathbf{P}_0^{-1}\mathbf{m}_0]$$

$$\mathbf{P}_T = [\mathbf{P}_0^{-1} + \frac{1}{\sigma^2}\mathbf{H}^T\mathbf{H}]^{-1}$$