Online Model Learning and Control using Sparse Local Gaussian Processes for Teleoperation

Brian Wilcox and Michael C. Yip
Department of Electrical and Computer Engineering, University of California, San Diego
{bpwilcox,yip}@ucsd.edu, Webpage: www.ucsdarclab.com

Introduction

In the task of teleoperation, commands and sensor data are received in sequential streams, thus, in a data-driven controller, the robot model should learn and adapt to these streams of data, while also making accurate predictions about the desired motor joint inputs at pace with the operator. *This work aims to achieve this task by using Sparse Online Local Gaussian Process Regression (SOLGPR) to infer a local function mapping of robot sensor states to joint states and perform a prediction of the teleoperation command for joint control.* Local Gaussian process models are learned and sparsified via Variational Inference with user-tuned bounds on model size and complexity. An optimization scheme involving periodic optimization of a drifting Gaussian process helps to reduce computation time in conjunction with the sparse local models. This framework provides a basis for a tradeoff between model complexity and performance

Background

Gaussian Processes (GP) represent a distribution over functions, characterized by a mean function and a covariance function (kernel).

$$f(x) \sim GP(m(x), k(x, x'))$$

Learning in GPs requires optimizing the hyperparameters of a kernel which in our case, is the squared exponential kernel:

$$k(x, x') = \sigma_s^2 exp(-\frac{1}{2}(x - x')^T W(x - x'))$$

Local GP Models are separate Gaussian models defined by local partitions of the training data into clusters, each defined by a model center location. Data is separated into the closest local model until a point exceeds a threshold distance, at which point a new model is created. Prediction of a query point becomes a weighted average of the local models according to the distance of the query point to each model center.

$$w_k = exp(-\frac{1}{2}(x - c_k)^T W(x - c_k))$$

Sparse Variational Inference reduces the computational complexity of a GP by selecting M << N support points, carried out by choosing the M points which minimize the Kullback-Liebler divergence of the induced posterior distribution of the GP and the original posterior distribution: $KL(q(f, f_m)||p(f, f_m|y))$

Approach

Sparsification strategy: tradeoff sparse representation, M, of L local models, with K as an upper bound on number of local model training points

Optimization strategy: Use a "drifting" GP such that the last d experienced training points are used for optimizing hyperparameters, with training scheduled intermittently

Prediction strategy: Use weighted average of combined variance of local model predictions & distance metric from model center. Choose up to 3 best models

Algorithm 1 Sparse Online Local Gaussian Process Regression

procedure SOLGPR
  initialize: jitter robot to create initial local GP Model
loop:
  Teleoperate: receive current command signal x_{tel}
  Control: predict joint command y_{pred} using latest GP models and use as control input. Predictions of each local model weighted by query distance to local models and variance of predictions. $y_{pred} \leftarrow \sum_{k=1}^{M} w_k y_k e^{-V_k} / \sum_{k=1}^{M} w_k e^{-V_k}$
  Experience: receive resultant robot state x_{exp}
  Partition: partition new experience into L local regions bounded by K points. New local model created when $max(w_k) > w_{gen}$
  Train: Update and train sparse local GP models of M support points via Variational Inference
  Learn: Optimize drifting GP of last D experienced points at a periodic schedule
until finished

Experimental Setup

The SOLGPR method was evaluated using a simulated teleoperation on a planar two-link robot manipulaton in ROS. The GPy library (Python) by Sheffield ML Group was used for the creation and optimization of Gaussian Process Models. Teleoperation paths (2D desired positions) were generated offline and fed to the SOLGPR controller sequentially. In every loop, the training data consists of the sensed 2D position of the robot end effector paired with the joint positions of the robot

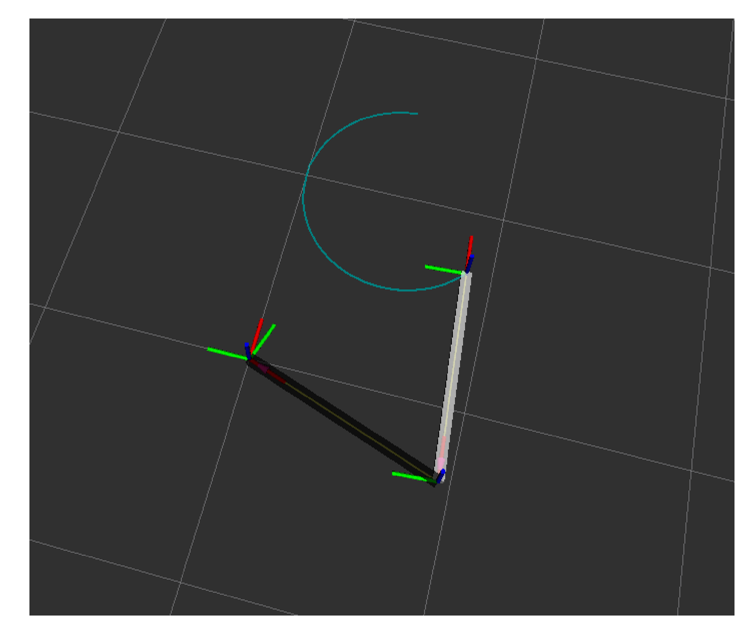

Figure 1: ROS simulator tracing a circle from scratch

Results

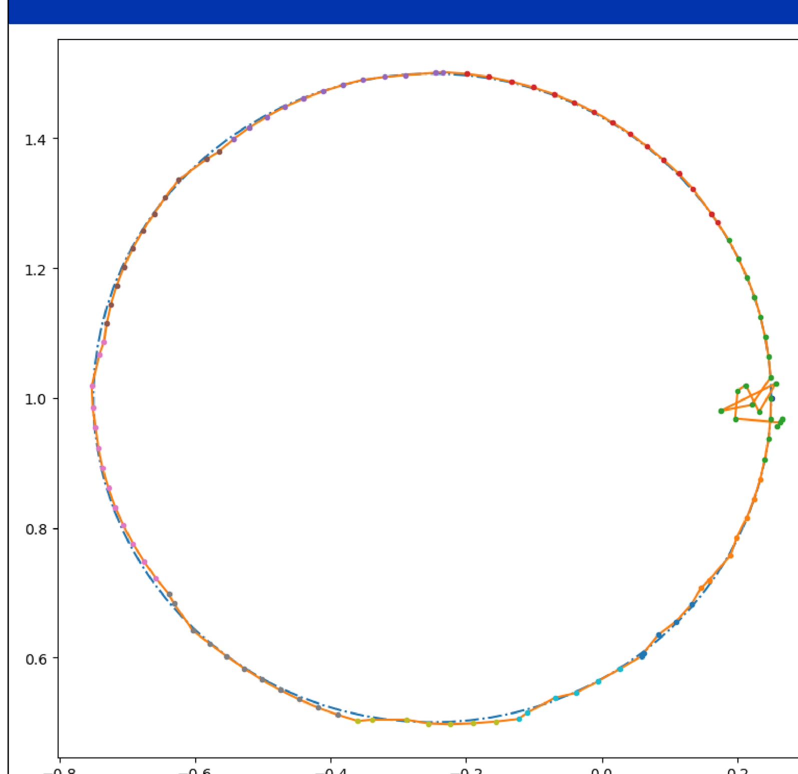

Figure 2: Trace of end effector path. Different color dots represent a separate local GP model

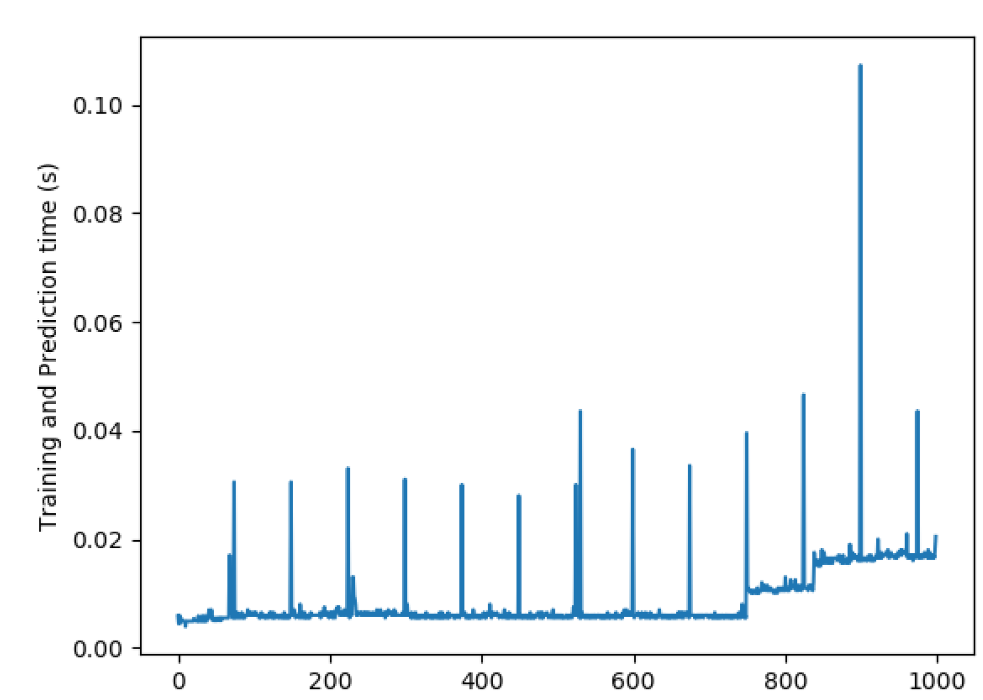

Figure 3: Time for algorithm to perform training and prediction per iteration of test input

Conclusion and Future Work

The Sparse Online Local Gaussian Process Regression method utilized in this work realized online model learning and control from sequential data. Computation time was kept low by implementing the strategies to tradeoff local model size and sparsity and optimizing a drifting GP periodically. However, the local model partitioning requires tuning of a distance threshold, thus some arbitrary paths do not lead to accurate predictions. Future work will focus on 1) using the distance between a query point and the convex hull of the neighboring local regions as a metric for partitioning, 2) optimization of model complexity parameters (K local points, L local models, M support points, and D drifting points, 3) adding additional features (i.e. end effector velocity) to training inputs

References

[1] M. K. Titsias, "Variational learning of inducing variables in sparse Gaussian processes," in International Conference on Artificial Intelligence and Statistics, 2009, pp. 567–574.
[2] D. Nguyen-Tuong, J. R. Peters, and M. Seeger, "Local Gaussian process regression for real time online model learning," in Advances in Neural Information Processing Systems, 2008, pp. 1193–1200.
[3] C. E. Rasmussen and C. K. Williams, Gaussian Processes for Machine Learning. MIT Press, 2006.
[4] C. Cheng and B. Boots, "Incremental Variational Sparse Gaussian Process Regression", in NIPS, 2016.
[5] F. Meier and S. Schaal, "Drifting Gaussian processes with varying neighborhood sizes for online model learning", International Conference on Robotics and Automation, Stockholm, 2016.