Robot Control via Local Gaussian Process Regression

Yash Bagla

February 26, 2018

1 Problem Description

In this work I propose to present an approximation inspired by locally linear regression techniques to the standard GPR using local Gaussian processes model. [1] I will follow this paper by Duy Nguyen-Tuong, Jan Peters, Matthias Seeger and implement some of the more successful approaches in Support Vector Regression (SVR). I will evaluate these procedures in terms of Model-based control, e.g. torque control, locally weighted projection regression (LWPR).

Robot control requires accurate dynamics models which cannot be obtained analytically for sufficiently complex robot systems to get high performance and compliant. In such cases, machine learning offers a promising alternative for approximating the robot dynamics using measured data. A natural framework to incorporate unknown nonlinearities as well as to continually adapt online for changes in the robot dynamics is something this approach offers. However, the most accurate regression methods, e.g. Gaussian processes regression (GPR) and support vector regression (SVR), suffer from exceptional high computational complexity which prevents their usage for large numbers of samples or online learning to date.[2]

2 Related Work

There is considerable literature in robot control using learning through GPR. Earlier work has touched upon Locally weighted projection regression (LWPR) which uses a small number of univariate regressions in selected directions in input space in the spirit of partial least squares regression,[1] control algorithms for learning and compensating for the dynamics of manipulators during the motion are presented and tested experimentally the resulting control performances are better than with measured parameters for any trajectory in the workspace [3] When using unscented Kalman filter (UKF) for control applications, learning to place the sigma points correctly from data can make sigma point collapse much less likely. Learning can result in a significant increase in predictive performance over default settings of the parameters in the UKF and other filters designed to avoid the problems of the UKF, such as the GP-ADF. At the same time, we maintain a lower computational complexity than the other methods. [4]

3 Project Milestones

This section will detail milestones so that I can roughly organize my work toward the completion of this project.

- Prepare a proposal and survey important literature
- Dig deeper into the literature and search for implementational details
- Clean, organize and explore the data. Include or remove clinical covariates as necessary.
- Prepare the code, run and test it extensively on example data. Decide if simulations will be necessary.
- Prepare a second draft with a detailed introduction and some results based on exploration.
- Run the alogrithms on the Data allocation and Model Learning.
- Run the alogrithms on Weighted Prediction, and collect the results. Prepare summaries and visualizations.
- Prepare Final Report

References

[1] S. Vijayakumar, A. D'Souza, and S. Schaal, "Incremental online learning in high dimensions," Neural Computation, 2005.

[2] Duy Nguyen-Tuong, Jan Peters, "Local Gaussian process regression for real-time model-based robot control," Intelligent Robots and Systems, 2008.

[3] E. Burdet, B. Sprenger, and A. Codourey, "Experiments in nonlinear adaptive control," International Conference on Robotics and Automation (ICRA), vol.1, pp. 537-542, 1997.

[4] Ryan Turner, Carl Edward Rasmussen, "Model based learning of sigma points in unscented Kalman filtering," Machine Learning for Signal Processing (MLSP), 2010.