Real-time Model-based Robot Control using Local gaussian Process Regression

Yash Bagla

Intermediate Report

Abstract. For model-based robot control online, approximation of the inverse dynamics model requires fast online regression techniques. To speed up standard Gaussian process regression (GPR) with local GP models (LGP) the training data is partitioned in local regions, for each an individual GP model is trained. Weighted estimation using nearby local models is used for the prediction of a query point. The proposed method achieves online learning and prediction in real-time. Comparisons with other non-parametric regression methods show that LGP has higher accuracy than LWPR and close to the performance of standard GPR and v-SVR.

1 Introduction

It is often hard to model the system sufficiently well for a complex light-weight arm robot and, thus, modern regression methods offer a viable alternative [1, 2]. For most real-time applications, online model learning poses a difficult regression problem due to three constraints, i.e., firstly, the learning and prediction process should be very fast (e.g., learning needs to take place at a speed of 20-200Hz and prediction at 200Hz to a 1000Hz). Secondly, the learning system needs to be capable at dealing with large amounts of data (i.e., with data arriving at 200Hz, less than ten minutes of runtime will result in more than a million data points). And, thirdly, the data arrives as a continuous stream, thus, the model has to be continuously adapted to new training examples over time [3].

These problems have been addressed by real-time learning methods such as locally weighted projection regression (LWPR) [1, 2]. The required manual tuning of many highly data-dependent metaparameters is the major drawback of LWPR [4]. Furthermore, for complex data, large numbers of linear models are necessary in order to achieve a competitive approximation.

A powerful alternative for accurate function approximation in high-dimensional space is Gaussian process regression (GPR). Since the hyperparameters of a GP model can be adjusted by maximizing the marginal likelihood, GPR requires little effort and is easy and flexible to use. However, the main limitation of GPR is that the computational complexity scales cubically with the training examples n. This drawback prevents GPR from applications which need large amounts of training data and require fast computation, e.g., online learning of inverse dynamics model for model-based robot control. Many attempts have been made to alleviate this problem, for example, (i) sparse Gaussian process (SGP) [5], and (ii) mixture of experts (ME) [6, 7]. In SGP, the training data is approximated by a smaller set of so-called inducing inputs [5]. Here, the difficulty is to choose an appropriate set of inducing inputs, essentially replacing the full data set [5]. In contrast to SGP, ME divide the input space in smaller subspaces by a gating network, within which a Gaussian process expert, i.e., Gaussian local model, is trained. The computational cost is then significantly reduced due to much smaller number of training examples within a local model. The ME performance depends largely on the way of partitioning the training data and the choice of an optimal number of local models for a particular data set [4].

In my project we combine the basic idea behind both LWPR and GPR. This will result in accuracy comparable with GPR while the speed will be close to local learning. The proposed paper used disturbance based measures for partitioning of data, where the corresponding hyperparameters are optimized by maximizing the marginal likelihood. I'll present results on a version of the Barrett WAM showing that with the online learned model using LGP the tracking accuracy is superior compared to state-of-the art model-based methods [8] while remaining fully compliant.

I'll be using data for Barrett WAM, a robot with 7 degree of freedoms. The training and testing data will be 2-dimensional (X, Y) with 12000 training and 3000 testing data points.

2 Project Milestones

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

Done so far

- Dig deeper into the literature and search for implementational details
- Clean, organize and explore the data. Include or remove clinical covariates as necessary
- Prepare a second draft with a detailed introduction and some results based on exploration

Remains to be Done

- Prepare the code, run and test it extensively on example data. Decide if simulations will be necessary.
- 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. Schaal, C. G. Atkeson, and S. Vijayakumar, Scalable techniques from nonparameteric statistics for real-time robot learning, Applied Intelligence, pp. 4960, 2002.
2. S. Vijayakumar, A. DSouza, and S. Schaal, Incremental online learning in high dimensions, Neural Computation, 2005.
3. Duy Nguyen-Tuong, Jan Peters, "Local Gaussian process regression for real-time model-based robot control," Intelligent Robots and Systems, 2008.
4. D. Nguyen-Tuong, J. Peters, and M. Seeger, Computed torque control with nonparametric regression models, Proceedings of the 2008 American Control Conference (ACC 2008), 2008.
5. J. Q. Candela and C. E. Rasmussen, A unifying view of sparse approximate gaussian process regression, Journal of Machine Learning Research, 2005.
6. V. Treps, Mixtures of gaussian process, Advances in Neural Information Processing Systems,2001.
7. C. E. Rasmussen and Z. Ghahramani, Infinite mixtures of gaussian process experts, Advances in Neural Information Processing Systems, 2002.
8. J. J. Craig, Introduction to Robotics: Mechanics and Control, 3rd ed. Prentice Hall, 2004.