Qualitative learning of object pushing by a robot

machine learning, qualitative reasoning, learning qualitative models

Abstract

In this paper, we demonstrate advantages of qualitative learning from experiments in a complex dynamic domain, in comparison with quantitative learning. Our learning domain is block pushing by a mobile robot. Induced qualitative models are intended for the planning of robot tasks of moving a block to a given position by point-contact pushing. Quantitative mathematical models of block pushing are very complex and therefore hard to apply. We used the QUIN method for learning tree-structured qualitative models in this domain. We showed that incomparably simpler and correct qualitative models can be induced from no more than a few hundreds of examples. Yet these models suffice for successful planning and plan execution for block moving tasks.

1 Introduction

In this paper we present a new approach to learning and task planning in continuous domains. The approach includes: learning of a qualitative model of the problem domain through experimentation in the domain, planning with a qualitative model to solve a given task, and plan execution in the real world. We carry out a detailed experimental case study in applying this approach to the task of moving a block to a goal position by robot's pushing the block. A typical task is illustrated in Figure 1. The results were obtained by a real robot (Figure 2) using an overhead vision system.

We use the QUIN method for learning tree-structured qualitative models. Usual, quantitative mathematical models of block pushing are very complex compared to induced qualitative models. In section 4 we describe the application of QUIN to inducing a qualitative model from the measured data. We validate the induced qualitative model in terms of its consistency with a known mathematical model of the same dynamic system. The obtained qualitative model is much simpler and more intuitive than the corresponding mathematical model.

A demonstration video of our work is available at http://www.youtube.com/watch?v=3xwwoIEDoQo. In addition to learning a qualitative model, the video also illustrates planning and plan execution, which is not the topic of this paper.

2 Related work

The use of machine learning to automatically induce models of a planning domain to be used for task planning is a traditional topic in AI. [Zimmerman and Kambhampati, 2003] give an overview of this research. [Garcia-Martinez and Borrajo, 2000; Veloso et al., 1995] are representative papers in this field. Our work is different from typical work in this field because we learn qualitative models of *continuous* dynamics of planning domain, as opposed to typically discrete models used in large majority of that research.

There is a considerable amount of work in the learning of qualitative models from data. [Bratko and Šuc, 2003] is a review til then. In this respect, our work is closest to [Šuc, 2003]. Our work is an application of Šuc's tree structured qualitative learning to a dynamic domain.

The problem of pushing objects by a robot has been studied in various settings. In [Lynch and Mason, 1996; Bajd and Balorda, 1994] a two point or a line contact between a robot and an object is assumed. More challenging, one point contact has also been discussed in [Behrens et al., 2010; Lau et al., 2011]. Problem of modeling the object-pushing domain has been addressed from two aspects. One assumed a predefined model for predicting consequences of robot's actions, as in [Lynch and Mason, 1996; Behrens, 2010]. The other used machine learning techniques to induce models from learning examples collected by robot's experimentation [Kopicki et al., 2009; Lau et al. 2011].

Predefined models can be based on human intuition [Emery and Balch, 2001] or more traditional, based on the laws of physics [Behrens 2010; Lynch and Myson; 1996]. In [Lau et al. 2011] a simple learning problem using only robot position and direction of pushing as independent variables was examined. In [Kopicky et al. 2009] a more extensive approach using Gaussian machine learning methods was proposed. Complex models are, due to high computational complexity, found difficult to use for the planning purposes. They only allow planning a few actions ahead [Walker and Salisbury, 2008]. An approach to plan six actions ahead was proposed in [Lau et. al 2011]. Discussed approaches have problems to capture the necessary details to perform more complicated tasks (e. g. model robot's angular velocity as an independent variable, which is needed for balancing the object while pushing it under single point of contact) and to enable time efficient

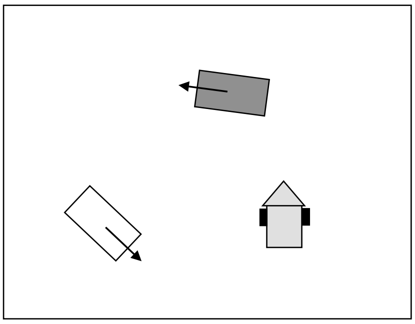

Figure 1: The robot's task is to move the object (dark rectangle) to the goal position (white rectangle) with the block in the indicated orientation (arrows). The robot may push the object by the tip of its triangular bumper. Dark rectangles of the robot's shape are the robot's wheels.

Figure 2: The robot.

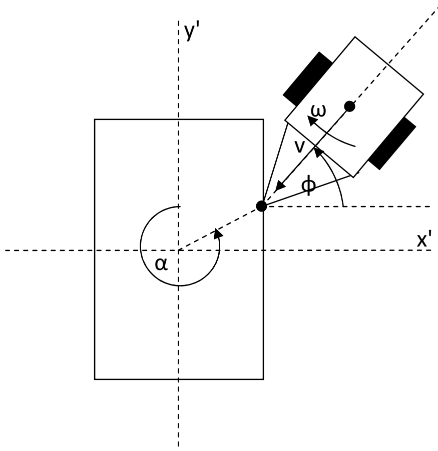

Figure 3: Variables defining the position and orientation of the robot relative to the block.

planning at the same time.

3 The pushing task

Figure 1 illustrates our robot's task of moving a block from its start position to the given goal position. The object is very light compared to the robot. The robot may push the object by making a point contact between the object and the tip point of the triangular bumper part of the robot. Note that the task involves both the translation and rotation of the object. The difficulty of the task comes from the complex translational and rotational movement of the object when the robot bumps into it. Models of point-contact pushing are very complicated.

Our robot, shown in Figure 2, was constructed from the Lego Mindstorms kit. The dimensions of the block were 20x11 cm, and the robot was of comparable size. The current position and orientation of the robot and the object were observed by an overview camera. Figure 3 shows the variables chosen to represent the measured data, where v is the velocity of the robot's center of gravity, ω is the robot's rotational velocity, α is a central angle used to define contact point between the robot and the object and ϕ is an angle under which the robot touches the object. The object's position is (x_o, y_o), and orientation is θ_o. The situation in Figure 3 is shown in adjusted coordinate system (x', y', θ') chosen so that the block's coordinates in this coordinate system are 0 which is convenient as it simplifies the presentation of the learned model.

4 Learning a qualitative model of pushing

4.1 The learning problem

The robot's actions last for a fixed time interval empirically chosen in our case to be 0.5 sec. The learning problem to induce a suitable dynamical model of pushing was formulated as follows. A robot's action is defined by the variables α, ϕ, v and ω. So these are the independent variables. The results of an action are the changes in the object's position $(\Delta x_o, \Delta y_o)$ and orientation $\Delta\theta_o$, and in the robot's position relative to the object given by $(\Delta\alpha, \Delta\phi)$. Thus a model consists of five functions: $\Delta x_o = f_x(\alpha, \phi, v, \omega)$, $\Delta y_o = f_y(\alpha, \phi, v, \omega)$, $\Delta\theta_o = f_\theta(\alpha, \phi, v, \omega)$, $\Delta\alpha = f_\alpha(\alpha, \phi, v, \omega)$, and $\Delta\phi = f_\phi(\alpha, \phi, v, \omega)$.

The problem of learning a qualitative model in our case is to find a qualitative statement of these five dependences.

The learning data were collected by the robot performing a number of example actions of pushing, and measuring the actions' effects (5-tuples) with the camera. The example actions were chosen so as to systematically cover the problem space with uniform sampling, without any sophisticated sampling strategy. The chosen number of examples for learning was 432. Due to the symmetries with respect to left-right and above-below, the actual number of measurements taken by the robot was 108. The robot was able to physically collect such a sample in about 15 minutes.

4.2 The qualitative learning tool

We used the QUIN program [Šuc, 2003] for learning tree-structured qualitative models from data. Induction of qualitative trees is similar to the induction of decision trees. The difference is that in decision trees, the leaves are labeled with values of the dependent variable, whereas in qualitative trees the leaves are labeled with what we call monotonic qualitative constraints (abbreviated as MQC).

Examples of qualitative trees will be given later. MQCs define qualitative constraints on the dependent variable. The dependent variable is also called class variable, and the other, independent variables are also called attributes. MQCs are a version of monotonicity constraints that are often used in qualitative reasoning [Kuipers, 1996; Forbus, 1997] give overviews and discuss various abstractions of mathematical relations used in qualitative reasoning. MQCs have the form $M^{Signs}(x_1, x_2, ...)$ where $Signs$ is a sequence of + or – signs, each of them corresponding to a variable x_i. For example, formula $p = M^{+,-}(T, V)$ says that p is "positively related" to T and "negatively related" to V. That means: if T increases and V stays unchanged then p also increases, and if V increases and T stays unchanged then p decreases.

4.3 Induced qualitative models

Learned models depend on measured data which may slightly differ between experiments at least due to measurement errors. An induced model consists of five qualitative trees, one for each dependent variable. We here present a typical such model. Figure 4 shows the qualitative tree for the change $\Delta\theta_o$ in the orientation of the object.

This tree applies to the cases when the robot's bumper is in contact with the left side of the rectangle (see Figure 3). When the contact is in the middle of this side, $\alpha = 90°$. When the point of contact is above the middle of the side, $\alpha < 90°$. This tree is intuitively quite sensible and corresponds to common sense human models of block pushing. Let us consider the leftmost leaf of the tree. This leaf applies to the cases when $\alpha < 76.82°$. The formula in the leaf states the qualitative relation $\Delta\theta_o = M^-(v)$. That is, the greater the robot's velocity, the faster the block will rotate in the clockwise direction (that is θ_o will decrease). The tree tends towards symmetry between the contact above or below the middle of the left side of the block. The tree is however not completely symmetrical. First, the thresholds for α are not symmetrical w.r.t. 90° as expected. So, for example, the threshold $87.19°$ should theoretically be $90°$. Also, the MQCs in the middle two leaves should mention the same variables. The reason for this asymmetry lies in the differences between measurement data that fall into these leaves. The monotonicity constraint with respect to α is very weak and was not detected in the data in one of these two leaves. The rest of the induced qualitative models are presented on Figure 5.

Again, most of the relations in these trees can be explained by common sense physics, although the symmetries in the thresholds are not ideal.

4.4 Validation against mathematical model

In addition to studying the intuitive plausibility of induced qualitative models, we also made an attempt to validate these models against known mathematical models. We started with the models by Yia and Erdmann [1996, 1998, 1999] that describe the point-contact pushing an object of any shape in the

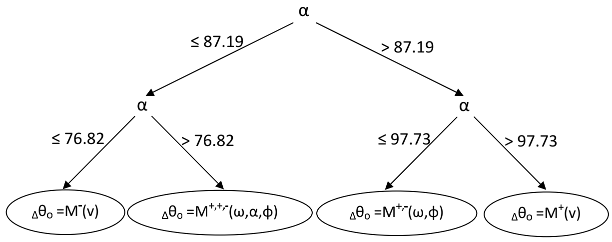

Figure 4: Qualitative dependence of the change $\Delta\theta_o$ of the orientation θ_o on the parameters of the action. mesured between positive y-axis and the line through the point of contact and the origin.

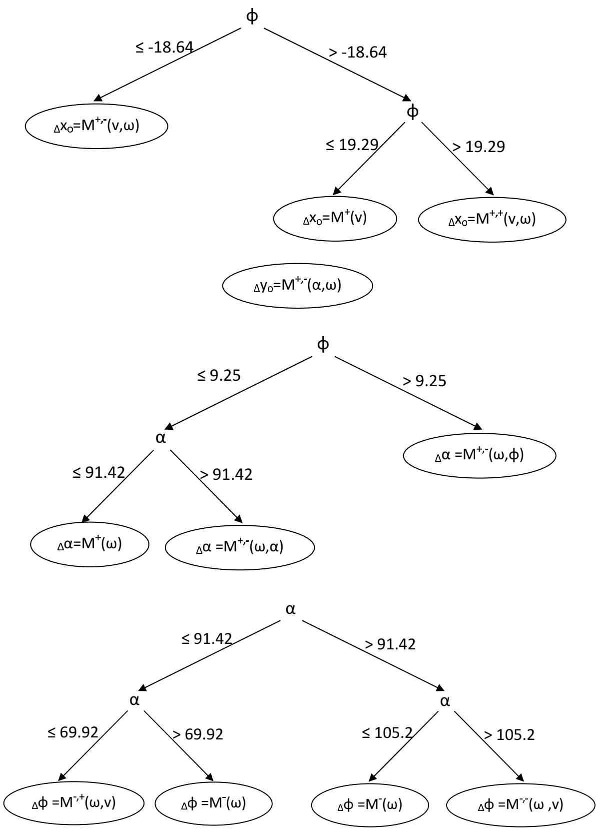

Figure 5: Qualitative trees for variables Δx_o, Δy_o, $\Delta\alpha$, and $\Delta\phi$.

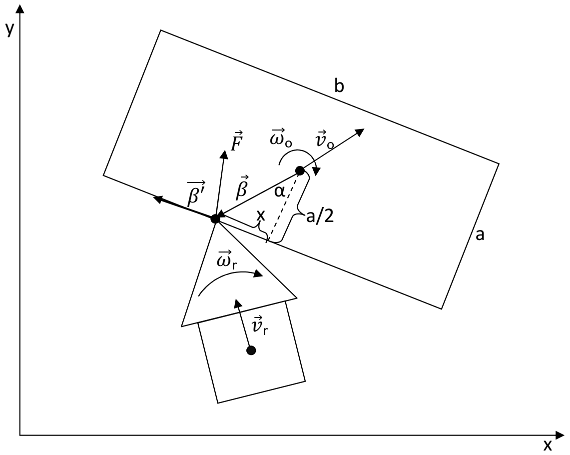

Figure 6: Model of pushing rectangular objects.

$x - y$ plane. We extended the model with robot's angular velocity ω_r and then applied the model to rectangular objects (see Figure 6). Object is described with a piecewise smooth closed curve β. Notation '′' means differentiation with respect to a position change along curve.

Pushing was described by Newton's (Equation 1) and Euler's (Equation 2) formula. Newton's formula describes translation and Eulers's formula rotation of the object. μ represents coefficient of friction between the robot and the object, η is the object's area density, R rotation matrix, and $J = \frac{1}{12}m(a^2 + b^2)$ moment inertia of a rectangular object. Vector $\overrightarrow{A_2} = 4(\int_0^{\frac{a}{2}} \int_0^{\frac{b}{2}} \sqrt{x^2 + y^2} dxdy)\hat{\omega_o}$ is used to describe dynamic friction due to rotation of the object around its center of mass.

$$\overrightarrow{F} - \mu\eta gab\hat{v_o} = m\dot{\overrightarrow{v_o}} \tag{1}$$

$$R\overrightarrow{\beta} \times \overrightarrow{F} - \mu\eta g\overrightarrow{A_2} = J\dot{\overrightarrow{\omega_o}} \tag{2}$$

Another equation was needed to describe the movement of point of contact between robot and object. For this purpose two cases were found.

If static friction between object and robot is the interval described with Equation 3, where $\gamma = arctan(\omega_r * r * d/v_r)$ and $\psi = arctan(\mu)$, then point of contact does not move.

$$R(\frac{\pi}{2} + \gamma + \psi)R\overrightarrow{\beta'} \times \overrightarrow{F} < 0 < R(\frac{\pi}{2} + \gamma - \psi)R\overrightarrow{\beta'} \times \overrightarrow{F} \tag{3}$$

If static friction is not this interval, then contact point moves along one of the sides as defined with Equation 4, where $\overrightarrow{d}$ represents a vector from robot's center of mass to the end of the bumper.

$$\overrightarrow{v_r} + \overrightarrow{\omega_r} \times \overrightarrow{d} = \overrightarrow{v_o} + \overrightarrow{\omega_o} \times R\overrightarrow{\beta} + R\frac{\overrightarrow{\alpha}}{2\cos^2\dot{\alpha}} \tag{4}$$

Since the point of contact is now moving, we have a dynamic friction instead of static one. Dynamic friction is captured with equation 5. Decision on using $+$ or $-$ depends on a direction of contact point movement.

$$\overrightarrow{F} \cdot R(\theta_o + \gamma \pm \psi)\overrightarrow{\beta'} = 0 \tag{5}$$

We will not discuss details of this mathematical model as complete understanding is not needed for the rest of the paper. The point to be noted is the striking difference in the complexity of this model in comparison with the induced qualitative model. The mathematical model above cannot be solved analytically, so the correctness of the induced qualitative model cannot be directly verified by considering the formulas above. It is, however, possible to detect inconsistencies between the two models by considering results of numerical simulation based on the mathematical model. We ran numerical simulations targeted at checking various parts of the qualitative model. Inconsistencies between the qualitative constraints in the qualitative model can thus be caught in numerical simulation traces. We will present an example of finding inconsistences for qualitative tree for $\Delta\theta_o$. For the rest of the trees, we will just describe the results.

Qualitative tree for $\Delta\theta_o$ This tree was consistent with the numerical simulations up to the asymmetry between the two middle leaves. Figure 7 (above) depicts results of the simulation with the initial conditions $\alpha = 96°$, $v = 10cm/s$, and for Case 1 $\omega = -9°s^{-1}$ and $\phi = -15°$, for Case 2 $\omega = -9°s^{-1}$ and $\phi = 15°$, for Case 3 $\omega = 9°s^{-1}$ and $\phi = -15°$, and for Case 4 $\omega = 9°s^{-1}$ and $\phi = 15°$. The results are consistent with MQC $M^{+,-}(\omega, \phi)$. To test if this QMC should also include dependency on α a series of experiments with initial conditions $v = 10\ cm/s$, $\omega = 0$, $\phi = 0$ and for Case 1 $\alpha = 102°$, for Case 2 $\alpha = 98°$, and for Case 3 $\alpha = 94°$ were carried out. Results are shown on Figure 7 (below). Numerical simulation showed that the correct QMC includes α. However, it is obvious that this influence is weak, being in the order of measurement errors. This explains why dependence on α was only detected in one of the two leaves.

Qualitative tree for Δx_o This tree was completely consistent with numerical simulations.

Qualitative (single node) tree for Δy_o This tree was completely consistent with numerical simulations.

Qualitative tree for Δα Induced threshold $\phi = 9.25°$ corresponds well to the transition point between static friction and dynamic friction, which is explained in the description of the mathematical model. However, due to the symmetry, there should be a similar threshold also for negative ϕ. In the region of static friction $\Delta\alpha = 0$, which cannot be described with a QMC. Thus QUIN induced a QMC in this region that corresponds only to the noise in the learning examples.

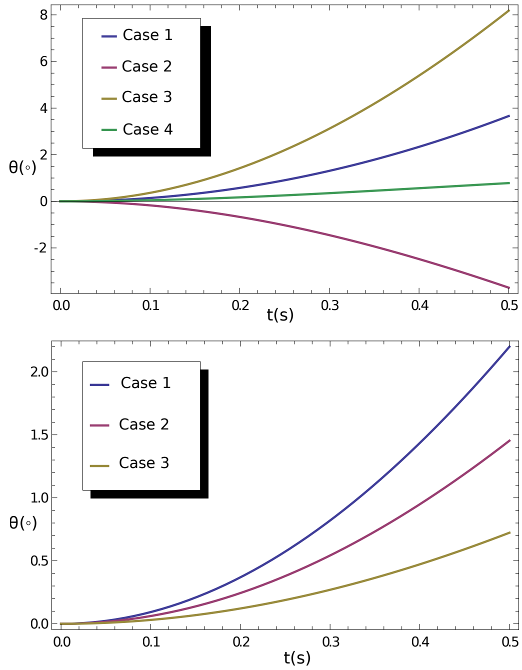

Figure 7: Results of numerical simulation for $\Delta\theta_o$.

Qualitative tree for $\Delta\phi$ This tree was consistent with all numerical simulations.

To conclude this comparison between the induced qualitative model and the mathematical model, we observe that the qualitative model is remarkably consistent with the mathematical model. The qualitative model has strong advantage in respect of simplicity and interpretability. The mathematical model, on the other hand, enables accurate numerical predications whereas the qualitative model can only make qualitative predictions about the direction of change of dependent variables.

5 Conclusions

The main contribution of this paper is the learning and validation of a qualitative model of block pushing by a robot with one point contact. This model is incomparably simpler than known mathematical models of pushing. It is intuitive and thus gives human-understandable insight into the dynamics of pushing an object. Even though it cannot predict numerical values precisely, it can still be successfully used in task planning (for initial results see video in the Introduction).

Future work mainly address utilization of qualitative models in planning tasks. We believe that the learning of (qualitative) models from examples in dynamic domains may be in some applications more practical than the use of mathematical models, even in cases when mathematical models are already known beforehand. They may, however, due to their complexity be too demanding and impractical to use in comparison with the learning of simpler, qualitative models.

References

[Balorda and Bajd, 1994] Z. Balorda, and T. Bajd, “Reducing Positioning Uncertainty”, ICRA, Aug. 1994.

[Behrens et al., 2010] M. Behrens, H. Shoudong, G. Dissanayake, "Models for pushing objects with a mobile robot using single point contact," in IEEE/RSJ International Conference Intelligent Robots and Systems, Oct. 2010.

[Bratko and Suc, 2003] I. Bratko, D. Šuc. Learning qualitative models. AI Magazine, 24(4) : 107-119.

[Emery and Balch, 2001] R. Emery and T. Balch, “Behavior-based control of a non-holonomic robot in pushing tasks,” ICRA 2001, 2001.

[Forbus, 1997] Ken Forbus, Qualitative reasoning. in: A. Tucker (Ed.), CRC Computer Science and Engineering Handbook, CRC Press, 715–733.

[Garcia-Martinez and Borrajo, 2000] An integrated approach of learning, planning and execution. Journal of Intelligent and Robotic Systems, 29:47-78.

[Kopicki et al., 2009] M. Kopicki, J. Wyatt, R. Stolkin, “Prediction learning in robotic pushing manipulation,” ICAR, 2009.

[Kuipers, 1994] Ben Kuipers, Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge, MIT Press, Cambridge, Massachusetts.

[Lau et al. 2011] M. Lau, J. Mitani, and T. Igarashi, “Automatic Learning of Pushing Strategy for Delivery of Irregular-Shaped Objects,” in IEEE International Conference on Robotics and Automation (ICRA), 2011.

[Lynch and Mason, 1996] K. Lynch and M. T. Mason, “Stable pushing: Mechanics, controllability, and planning,” in International Journal of Robotics Research, Dec. 1996.

[Šuc, 2003] Dorian Šuc, Machine Reconstruction of Human Control Strategies, Vol. 99 of Frontiers in Artificial Intelligence and Applications, IOS Press, Amsterdam, The Netherlands, 2003.

[Veloso et al., 1995] M. Veloso, J. Carbonell, A. Perez, D. Borrajo, E. Fink, J. Blythe. Integrating planning and learning. J. of Experimental and Theoretical AI, 7(1).

[Walker and Salisbury, 2008] S. Walker and J. K. Salisbury, “Pushing using learned manipulation maps,” ICRA 2008, pp. 3808–3813, 2008.

[Jia and Erdmann 1996] Y. Jia and M. Erdmann, “Pose from pushing”, Proc. IEEE International Conference on Robotics and Automation, April 1996, 165-171.

[Jia and Erdmann 1998] Y. Jia in M. Erdmann, Observing Pose and Motion through Contact, Proc. IEEE International Conference on Robotics and Automation, May 1998.

[Jia and Erdmann 1999] Y. Jia and M. Erdmann, “Pose and motion from contactInternational”, Journal of Robotics Research , 18(5), 466-490.

[Zimmermann and Kambhampati, 2003] . Zimmerman and S. Kambhampati, Learning-assisted automated planning: Looking back, taking stock, going forward. AI Magazine, 24(2) 73-96.