Using Crowdsourcing to Generate Surrogate Training Data for Robotic Grasp Prediction

Matthew Unrath*, **Zhifei Zhang**†, **Alex Goins**†, **Ryan Carpenter**†,
Weng-Keen Wong* and **Ravi Balasubramanian**†
*School of EECS, †School of MIME
Oregon State University
Corvallis, Oregon 97331
Email: unrathm@onid.orst.edu, zzf861011@gmail.com, alexgoins301@gmail.com, rcarpenter05@gmail.com,
wong@eecs.oregonstate.edu, ravi.balasubramanian@oregonstate.edu

Abstract

As an alternative to the laborious process of collecting training data from physical robotic platforms for learning robotic grasp quality prediction, we explore the use of surrogate training data from crowd-sourced evaluations of images of robotic grasps. We show that in certain regions of the grasp feature space, grasp predictors trained with this surrogate data were almost as accurate as predictors built using data from physical testing with robots.

Introduction

Automated grasping of everyday objects outside of a controlled laboratory setting is a challenging problem in robotics. A key aspect of the automated grasping process involves evaluating the quality of the grasp. Machine learning is a promising approach to predicting grasp quality (Goins et al. 2014) but its success depends on acquiring accurate training data. Usually, such data is produced from a physical robotic platform. However, obtaining this *physical testing* training data is labor intensive and time consuming.

In this work, we explored using more easily obtained surrogate training data consisting of crowd-sourced evaluations of robotic grasp images. We compared the performance of two grasp predictors: a classifier trained on physical testing data, which we refer to as the *PTPredictor*, and a classifier trained on crowd-sourced human evaluations, which we refer to as the *CSPredictor*. These two training data sets contained identical feature vectors, but differed only in the class labels. We also identified high-competence regions of the grasp feature space for the CSPredictor.

Methodology

Grasp Generation and Testing

Using a virtual BarrettHand[1] in a simulation environment developed in OpenRAVE (Diankov and Kuffner 2008), 22 human subjects generated 522 grasps across nine everyday objects. For each grasp, the object's location and the robot's configuration were recorded in order to be able to replicate the grasp physically and in a simulator.

After grasp generation, the performance of each grasp was evaluated by reproducing the same grasp on a physical robot and picking up the grasped object over ten trials. Then, the grasp was subjected to a random rapid movement to shake the object, with a success being a grasp that retained the object in the hand. Grasps with higher than 80% average success rate were considered good grasps, while the rest were considered bad grasps. The threshold of 80% was determined from past studies (Balasubramanian et al. 2012).

Human Assessment of Grasps

Pictures of the grasps were uploaded to Mechanical Turk[2] and Survey Monkey[3] for online users to vote on for small monetary gain. Each survey consisted of 32 tasks, where each task was a randomly selected grasp from the full set of 522 grasp examples. To eliminate voter bias, a Latin-squares random sequence generator was used to create each arrangement of the 32 pictures.

Each task consisted of two pictures of the robot grasping the object from different angles followed by the question ”Will the robot be able to securely pick up the object?”. The humans were asked to rate the success of the grasp on a 1 to 5 scale with 5 being the most successful. For each example grasp, the scores from all the human voters were averaged. For quality control, we made one grasp for each object to be blatantly unsuccessful. Data from voters who scored poorly on this control grasp were discarded.

Grasp Prediction

We converted each grasp into a feature vector and a class label of *good* or *bad* grasp. For our feature representation, we used 11 grasp metrics (more details in (Goins et al. 2014)) that have been used in the robotics literature to capture properties of a grasp that make it secure and robust. Each feature was standardized by subtracting off the mean and dividing by the standard deviation.

We divided the data into three partitions (*train1*, *train2* and *test*). Before training, we used Principal Component Analysis (Jolliffe 2002) to reduce the dimensionality of the *train1* feature space from 11 down to 2. We then trained the grasp predictor, which was a Gaussian Process (Rasmussen

[1] http://www.barrett.com/robot/products-hand.htm

[2] https://www.mturk.com/mturk/welcome

[3] https://www.surveymonkey.com

2006), on the *train1* data. The class labels for the *train1* partition were from physical testing data for the PTPredictor and from crowd-sourced evaluations for the CSPredictor. The *train2* and *test* partitions contained ground truth physical testing class labels. The *train2* partition was used to discover the high-competence regions of the feature space. The *test* partition was used to evaluate the grasp predictor. We repeated this partitioning for 30 iterations and reported results averaged over these iterations.

To identify high competence regions, we partitioned the 11 dimensional feature space and evaluated the performance of the grasp predictor in each region. Due to the data being non-uniformly distributed, we applied a multi-resolution space-partitioning data structure called a k-d tree (Bentley 1975). Each leaf in the k-d tree contained a subset of the data instances that are "near" each other. For each leaf node, we computed the F1 score using the ground truth class labels from *train2* and the predicted class labels from a GP trained on *train1*. K-d tree leaf nodes with F1 score above a specified threshold were considered as high-competence regions. During testing, we first evaluated if a test data point fell within a high-competence region. If it did, the GP made a prediction; otherwise, the GP abstained.

Results and Discussion

A commonly used metric for classification performance is area under the ROC curve (abbreviated as AUC). Figure 1 illustrates the AUC of the PTPredictor and the CSPredictor over a variety of competence thresholds. Note that as we increased the competence threshold, the percentage of test data points in the high-competence regions (shown by the percentages above each group of bars) decreased. In the leftmost pair of bars, the AUC of the two classifiers was computed over the entire test set; in this case, the PTPredictor (AUC 0.766) outperformed the CSPredictor (AUC 0.659) by a margin of only 0.107, indicating that crowd-sourced evaluations were a viable surrogate dataset. In addition, we identified and leveraged the high-competence regions of the CSPredictor as described in our Methodology section. We compared the AUCs of both classifiers on only the data points that fell within the high-competence regions. As shown in Figure 1, the CSPredictor's AUC increased to the 0.79-0.89 range and the gap between the two classifiers narrowed substantially. In fact, at higher competence thresholds, the 95% confidence intervals of the classifiers overlapped considerably.

In order to visualize the data, we projected the data instances onto their first two Principal Components. The left plot in Figure 2 shows the grasps that were often correctly predicted by the PTPredictor during the 30 iterations and not by the CSPredictor, and vice versa on the right plot. These plots clearly show that the two grasp predictors were accurate in different regions of the grasp feature space.

Conclusion

In high-competence regions of the feature space, a grasp predictor trained on crowd-sourced evaluations of robotic grasps images was shown to be almost as accurate as a predictor trained on data from physical testing with robots. For future work, we will investigate the grasps in Figure 2 that were more accurately predicted by one classifier over the other. We will also explore other surrogate datasets, such as data generated by a grasp simulator.

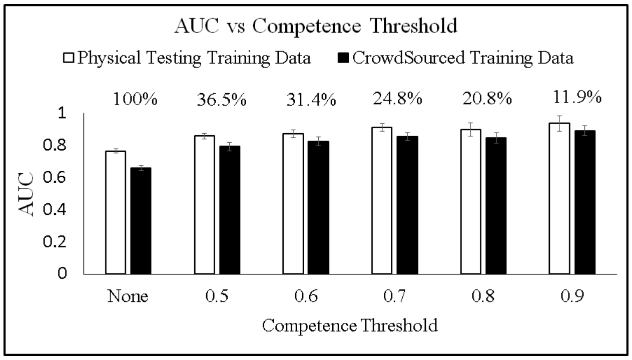

Figure 1: AUCs for the PTPredictor (white) and the CSPredictor (black) at various competence thresholds. The 95% confidence intervals are also shown. The number above each pair of bars represents the percentage of test data points in the high-competence regions.

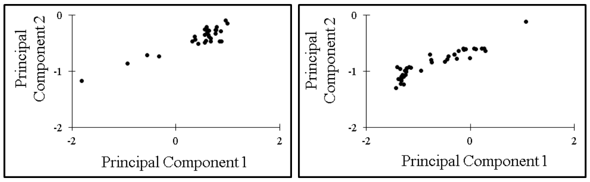

Figure 2: A 2D visualization of the data instances, shown as black dots, that were correctly predicted by the PTPredictor but not by the CSPredictor (left) and vice versa (right).

References

Balasubramanian, R.; Xu, L.; Brook, P. D.; Smith, J. R.; and Matsuoka, Y. 2012. Physical human interactive guidance: Identifying grasping principles from human-planned grasps. *IEEE Transactions on Robotics* 28(4):899–910.

Bentley, J. L. 1975. Multidimensional binary search trees used for associative searching. *Communications of the ACM* 18(9):509–517.

Diankov, R., and Kuffner, J. 2008. OpenRAVE: A planning architecture for autonomous robotics. Technical Report CMU-RI-TR-08-34, The Robotics Institute, Pittsburgh, PA.

Goins, A. K.; Carpenter, R.; Wong, W.-K.; and Balasubramanian, R. 2014. Evaluating the efficacy of grasp metrics for utilization in a gaussian process-based grasp predictor. In *Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Jolliffe, I. T. 2002. *Principal Component Analysis*. New York: Springer, 2nd edition.

Rasmussen, C. E. 2006. *Gaussian processes for machine learning*. MIT Press.