

Evaluating Pinch Quality of Underactuated Robotic Hands

Shiquan Wang
Mechanical Engineering
Stanford University
Email: shiquan@stanford.edu

Hao Jiang
Mechanical Engineering
Stanford University
Email: jianghao@stanford.edu

Abstract—Underactuated robotic hands are favorable for some applications due to its simplicity in mechanism and control compared with fully actuated hand. Incorporating sensors makes it possible for a hand to accomplish low level task autonomously. This project aims at analyzing pinch quality of an underactuated hand that was developed for underwater teleoperation with 8 sensor channels on each finger, exploring a machine learning method that is able to predict pinch quality reliably for autonomous pinch. The importance of each sensor feature was also evaluated. An average of test error of 4.6% was achieved with feature selection, SVR algorithm and a modified predicting strategy.

Keywords—*machine learning, underactuated hand, sensor fusion, pinch quality.*

I. INTRODUCTION

The application for the hand described in this paper is a new underwater robot intended for exploration and biological research in coral reef zones in the Red Sea which will allow marine biologists to remain above water while obtaining specimens, positioning equipment, and performing other monitoring and maintenance tasks via tele-operation system down to 100 m below the surface, at pressures to 11 bar[1]. The hand is compliant, underactuated, and backdrivable and uses flexures instead of pin joints, subsequently reducing mechanical complexity.

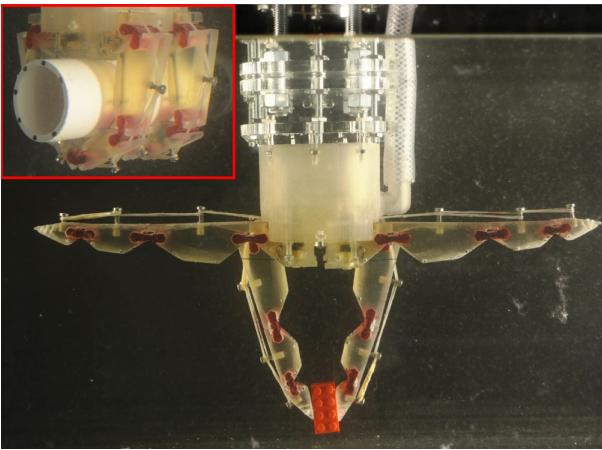


Fig. 1. Examples of the underwater underactuated hand pinching and grasping objects.

Underactuated hand can automatically adapt to the object with easy control but still cannot solve all control problems.

Buoyancy makes object with density similar to sea water subject to disturbance such as unsuccessful grasp or pinch and could be floating around. Re-grasping or re-pinchng the object could be very slow in tele-operation tasks. For this specific application, it would be helpful to sense the pinch or grasp quality to prevent slip when exerting force, which could typically be solved by using static force sensor. However, most of static force sensors cannot be used under water due to the change of water pressure. Other problems include limited space on the finger tip, difficulty of accurate force sensing over different contact positions.

To solve this problem, we propose an active sensing method to evaluate pinch quality based on non-static-force sensors that have already been installed in the fingers for other sensing purposes.

The idea is to vibrate the finger in certain frequency when pinching the object to obtain information that might indicate the pinch quality. Similar active sensing idea has been done in [2], where a vibration motor was used to resonate with the finger so as to detect the contact position. However, it is difficult to vibrate the finger fast enough without extra actuator due to the slow mechanical response of the finger compliant joint. To solve this problem, we propose a different active sensing method. Instead of fast vibration, slow squeezing motion was used while sensing. The sensing information were then converted into pinch quality indicator using machine learning methods. The following sections talk about the details of sensing, experiments, machine learning methods and the optimization.

II. DATA ACQUISITION

The basic experiment setup includes an underactuated hand developed for an underwater robot, an external motor driver, an Arduino DUE and a PC. The Arduino DUE collects data from the finger sensors via I2C communication and does position PID control of the motors that drive the fingers. A piece of C++ code runs on the PC which communicates with the Arduino DUE to give high level position commands to the hand and acquire sensing data. On each finger there are 4 sensors with 8 sensor channels, known as 3 axes of an accelerometer on the fingertip and 5 joint angle sensor channels, detecting the motion of each finger with 100Hz sampling rate. The experiment setup is shown in Fig. 2 (left).

In each pinch test, two opposed fingers were driven to target positions to pinch an object, then started to move slightly



Fig. 2. *Left:* The experiment setup of the underactuated hand pinch wooden block. *Middle and right:* Two target objects used in the experiment: a piece of pink foam which is stiff and a piece of black foam which is soft.

back and forth to squeeze the object for three and a half cycles. After that, the object was slowly pulled out, and a Mark-10 Series 4 digital scale was used to measure the pull-out force. This pull-out force evaluates the pinch quality; i.e. larger force corresponds to stronger pinch. The pinch position for each pinch test was randomized to achieve varied pinch quality. Two objects were used in the experiment, known as a piece of stiff pink foam and a piece of soft black foam, which are shown in Fig. 2 (middle and right). For each object 150 sets of sensor data were recorded.

III. FEATURE EXTRACTION

Joint angle sensor data were unfiltered because the noise is not significant. A typical joint angle sensor reading throughout the pinch test is shown in Fig. 3. Because of the periodic squeeze, features were extracted into means and standard deviations of the upper and lower peaks of the periodic signals. Since accelerometer is sensitive to mechanical and electrical noise, the data was low-pass filtered at around 25Hz using equi-ripple window for extracting the upper and lower peak during the squeezing cycles. A standard deviation of the noise (the difference between filtered and unfiltered signal) was also computed as the inhibition of noise might reveal information about the pinch. A one-axis accelerometer data in a typical experiment of pinching a piece of pink foam is shown in Fig. 4. All the features are listed in Tab. I.

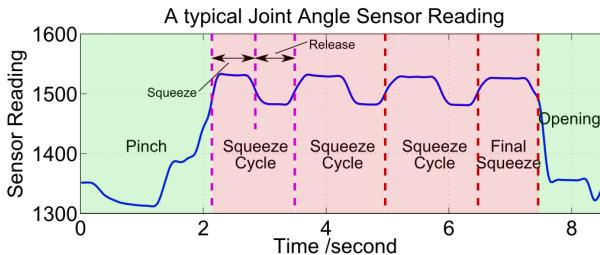


Fig. 3. Sensor reading of a typical joint angle sensor on a finger joint.

Feature	Description	Feature	Description
1-8	Sensor upper peak mean	17-24	Sensor lower peak mean
9-16	Sensor upper peak standard deviation	25-32	Sensor lower peak standard deviation
33-35	Standard deviation of high frequency components of the 3 axis accelerometer data		

TABLE I. MEAN AND STANDARD DEVIATION OF 6 SENSORS ALONG THE FINGER WERE CHOSEN AS FEATURES.

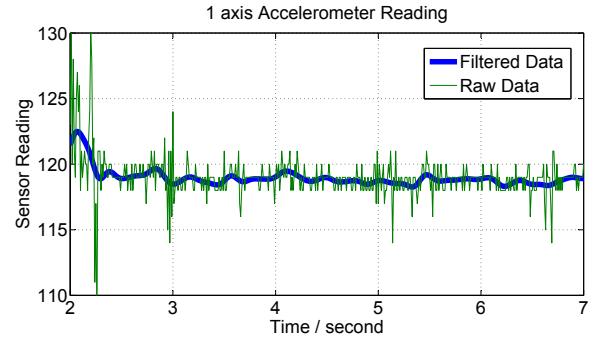


Fig. 4. Sensor reading of a one-axis accelerometer data.

IV. METHOD

A. Labeling Optimization

The original label is the pull-off force of each pinch test which is continuous and need to be discretized. K-means was used to categorized the pull-off force into multiple levels (clusters). By sorting the cluster centroid, the labels become indicators of the pinch quality, where larger label corresponds to better pinch. The number of labels was optimized with Softmax logistic regression to achieve the minimum training and test error.

B. Learning Algorithm

Four different methods known as k-means, logistic regression (softmax regression), SVM (Support Vector Machine), and SVR (Support Vector Regression [3]) were implemented to classify the data sets. The explanations of above algorithms are in [4]. The gradient ascent method of softmax regression algorithm with multiple labels is shown as follows:

$$\nabla_{\theta_j} J(\theta) = -\frac{1}{m} \sum_{i=1}^m [x^{(i)}(1\{y^{(i)} = j\} - p(1\{y(i) = j\}|x^{(i)}; \theta))] \quad (1)$$

Data were normalized so that each feature in the training examples has a mean of 0 and a standard deviation of 1. The test data were normalized in the same way that the training examples do with the normalization parameter derived from the training examples. All the above algorithms use both normalized and un-normalized data except logistic regression (softmax regression) because of the difficulty of finding a good initial condition to start without crashing the code. The normalization algorithm is shown as follows:

$$\mu = \frac{1}{m} \sum_{i=1}^m x^{(i)} \quad (2)$$

$$x^{(i)} := x^{(i)} - \mu \quad (3)$$

$$\sigma_j^2 = \sum_{i=1}^m (x_j^{(i)})^2 \quad (4)$$

$$x^{(i)} := \frac{x^{(i)}}{\sigma_j} \quad (5)$$

The mean and standard deviation of training and test errors and the average computation time were acquired for each method by 50 runs of learning with randomly chosen training and testing sets, based on which the performance of the methods can be compared. SVR was chosen for further optimization due to its better prediction accuracy and computational efficiency.

C. Feature Selection

Feature selection and evaluation based on SVR method was done to optimize the feature set. In feature selection, the contribution of each feature was prioritized with backward search: in each iteration, searching for the feature that makes least effect on the test error when being removed and then remove it for next iteration. Test errors and training errors were recorded during the iterations. To evaluate the contribution of each feature quantitatively, the effect caused by each feature was computed from the difference of test error of each feature before and after being removed, which is regarded as the feature score. Thus, for a feature that causes the test error increase when being removed, its score is negative. Since a single score result has to be computed based on the same training and testing data sets for prioritization consistency, which is not statistically accurate, 50 rounds of feature selection and evaluation were done with randomly chosen training and test set to get the feature scores.

D. Prediction Strategy

Another reason for choosing SVR is that it provides a non-binary result indicating the confidence of the prediction which gives more information and more flexibility. Since in real application (tele-operation task) one can always choose to redo the pinch if the confidence of good pinch is low, so one strategy is to ignore any prediction that is within a predefined confidence margin which can further improve the performance.

V. RESULT

A. Number of Labels

As shown in Fig.5, the training and test error of softmax regression increases dramatically as the number of labels increases. This is because for each label there is not enough data points and the data is not very separable. Therefore, the number of labels is set to 2 for all the rest of learning algorithms.

B. Machine Learning Algorithms

The size of training data set and testing is 105 and 45. For each method the training and testing data sets were generated randomly for 50 times and computed correspondingly. The result is shown in Tab.II. SVR outperformed the rest of the algorithms due to its comparatively low training and test error, efficient computation and non-binary output.

C. Feature Scores

The score of each feature is given in Fig.6. As shown in the plot, Feature 1, 2 and 4 has zero or negative effect on the performance which called as "bad features". Feature 15, 21 and 31 (good features) are the three most important features for

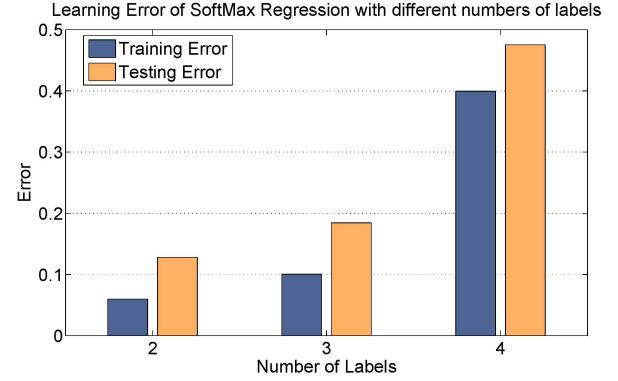


Fig. 5. Training and test error of logistic regression (Softmax) with different number of labels.

	Training Error Mean	Training Error STD	Testing Error Mean	Testing Error STD	Computation Time
K-Means (UN)	0.2437	0.0500	0.2454	0.0624	0.6708
K-Means (N)	0.3514	0.0400	0.3566	0.0520	0.6781
Softmax (N)	0.0663	0.0141	0.1164	0.0283	1.4798
SVM(UN)	0.1020	0.0200	0.1081	0.0283	0.1890
SVM(N)	0.1201	0.0693	0.1807	0.0768	0.1911
SVR (UN)	0.0921	0.0173	0.1135	0.0316	0.1872
SVR(N)	0.1332	0.0412	0.1813	0.0458	0.1907

TABLE II. MEAN AND STANDARD DEVIATION OF TRAINING AND TEST ERRORS ACQUIRED FROM 50 LEARNING TEST USING DIFFERENT METHODS. N AND UN DENOTE NORMALIZED AND UNNORMALIZED DATA.

the prediction. Features with scores less than 0.05 are called as "weak features".

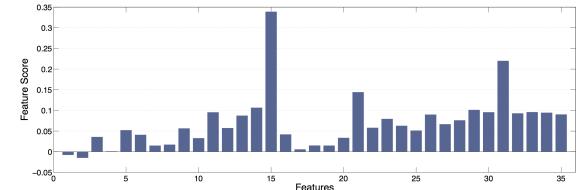


Fig. 6. Score of each feature which indicates its contribution to the classification.

D. Feature Set and Prediction Margin

Performance of SVR was optimized by selecting different feature sets and a prediction strategy with different margin. The feature sets include all features, features without bad features, features without bad and weak features, and only good features. Two different prediction margin (0.1 and 0.2) was tested which resulted in 8.6% and 16.4% ignoring rates. As shown in Tab.III, the feature set without bad features gives the best performance. The 3 good features can already guarantee the accuracy to be around 87%. With 0.2 prediction margin, the accuracy can be up to 96%.

E. Learning Curve

The learning curve (Fig.7) was computed based on SVR algorithm with zero prediction margin, which runs 50 times

	Training Error Mean	Training Error STD	Testing Error Mean	Testing Error STD
All Features	0.0936	0.0157	0.1163	0.0317
32 Features (Score > 0)	0.0951	0.0181	0.1106	0.0324
22 Features (Score > 0.2)	0.0954	0.0156	0.1115	0.0298
3 Features (Top 3 scores)	0.1173	0.0200	0.1255	0.0335
Modified Prediction (0.1 margin, 8.6% data ignored)	0.0298	0.0160	0.0772	0.0280
Modified Prediction (0.2 margin, 16.4% data ignored)	0.0509	0.0149	0.0585	0.0268

TABLE III. MEAN AND STANDARD DEVIATION OF TRAINING AND TEST ERRORS ACQUIRED FROM 50 LEARNING TEST USING DIFFERENT FEATURE SETS AND PREDICTION MARGIN. THE AMOUNT OF DATA THAT WAS IGNORED GIVEN DIFFERENT PREDICTION MARGIN IS ALSO PROVIDED

for each data size. The training error and test error tend to converge at the data size of around 130, which indicates the data that we collected is enough for the learning.

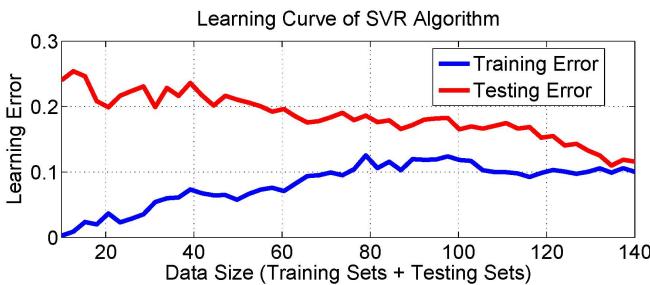


Fig. 7. Learning curve. Plots of training and test errors with different size of training and testing sets.

VI. DISCUSSION

All the supervised learning algorithms discussed in Section IV demonstrated decent performance because the data is intrinsically very separable according to the 2 labels. K-means had poor performance because the clusters calculated by this algorithm does not necessarily reflect the real labeling and is very sensitive to data stretch. As a result, k-means method has very different learning results on normalized and un-normalized data.

As shown in the result, in terms of learning performance the best choice of the number of labels is 2, which corresponds to a classification of "good" pinch and "bad" pinch. Higher number of labels results in much higher learning errors. However, in terms of real pinch application, it is desirable if the algorithm predicts continuous pull-out force. Thus, the more the number of labels is, the closer the prediction is to continuous case. SVR and Softmax regression show better classification results than SVM when the ambiguous classification region with equal or approximately equal probability is neglected. In real application, such neglect corresponds to abandoning the current pinch and redoing a firmer pinch to guarantee the pinch quality.

As shown by the feature score and performance with different feature sets, deleting the features that have negative impact can slightly improve the performance. Most of the features related with standard deviation are more useful than absolute values, since the latter tend to vary over different

pinched objects. According to the actual implementation, the good features are upper peak and lower peak standard deviation of proximal joint angle and lower peak mean of middle joint angle. In the robotic hand, the proximal joints are quite flexible and twisting of which sometimes can cause unstable pinch as the two finger tips are not faced each other. This fact might explain the reason that the proximal joint angle is such a good indicator of the pinch quality. The bad features include upper peak mean of two of the three axes of the accelerometer reading and the distal joint angle. All these sensors are very sensitive to noises. Most of the weak features are peak mean of the sensor readings. Understanding the importance of each feature for evaluating the pinch quality can help to choose which sensor to implement if a more simplified hand is needed.

VII. CONCLUSION AND FUTURE WORK

In conclusion, with the data and features acquired from pinch experiments with an underactuated robotic hand, logistic regression (Softmax regression) and SVR both demonstrate a training error and test error less than 15%. Labeling optimization shows the best choice of the number of labels is 2, yet the real application suggests larger number of labels. Feature selection shows there are 3 most influential features that captures the major properties of the 33 features, which lowers down the computation cost significantly. In the future, more data sets will be acquired from experiment to lower down the variance of the above learning algorithms. Learning algorithms to detect the properties of different objects will be developed.

APPENDIX A PROOF OF THE FIRST ZONKLAR EQUATION

ACKNOWLEDGMENT

Acknowledgments: The work was supported by The KAUST Red Sea Robotics Research Exploratorium. The assistance of Hannah Stuart, Heather Barnard, Matteo Bagheri, Merritt Jenkins, and Audrey Sheng is gratefully acknowledged.

REFERENCES

- [1] Stuart, H.S.; Shiquan Wang; Gardineer, B.; Christensen, D.L.; Aukes, D.M.; Cutkosky, M., "A compliant underactuated hand with suction flow for underwater mobile manipulation," Robotics and Automation (ICRA), 2014 IEEE International Conference on , vol., no., pp.6691,6697, May 31 2014-June 7 2014
- [2] Backus, S.B.; Dollar, A.M., "Robust, inexpensive resonant frequency based contact detection for robotic manipulators," Robotics and Automation (ICRA), 2012 IEEE International Conference on , vol., no., pp.1514,1519, 14-18 May 2012
- [3] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification Journal of Machine Learning Research 9(2008), 1871-1874.
- [4] <http://cs229.stanford.edu/materials.html>