Classifying Movement Articulation for Robotic Arms via Machine Learning

Asha Vijayan, Chaitanya Nutakki, Chaitanya Medini, Hareesh Singanamala, Bipin Nair
Krishnashree Achuthan, Shyam Diwakar
Amrita School of Biotechnology
Amrita Vishwa Vidyapeetham
Clappana (P.O.), Kollam
Kerala, India
(ashavijayan, chaithanyakumar, bipin, hareeshsinganamala}@am.amrita.edu
{krishnachaitanya, krishna, shyam}@amrita.edu

ABSTRACT: *Articulation via target-oriented approaches have been commonly used in robotics. Movement of a robotic arm can involve targeting via a forward or inverse kinematics approach to reach the target. We attempted to transform the task of controlling the motor articulation to a machine learning approach. Towards this goal, we built an online robotic arm to extract articulation datasets and have used SVM and Naïve Bayes techniques to predict multi-joint articulation. For controlling the preciseness and efficiency, we developed pick and place tasks based on pre-marked positions and extracted training datasets which were then used for learning. We have used classification as a scheme to replace prediction-correction approach as usually attempted in traditional robotics. This study reports significant classification accuracy and efficiency on real and synthetic datasets generated by the device. The study also suggests SVM and Naïve Bayes algorithms as alternatives for computational intensive prediction-correction learning schemes for articulator movement in laboratory environments.*

Keywords: Robotic Articulator, Movement, Machine Learning, experimental dataset, Naïve Bayes, SVM

Received: 10 June 2013, Revised 19 July 2013, Accepted 24 July 2013

1. Introduction

Robotic articulation and manipulation in the real world environment has improved in recent years in various fields and applications [1]. Modern robotic manipulators are constructed using a set of joints which are combined together using rigid links [2]. These manipulators have been developed to do specialized tasks like examining different parameters of an object [3], reaching the target with an optimal feedback control [4], etc. A main area of research in humanoid robotics is on the identification of objects based on spatial relationship [5]. Different robotic tasks related to object detection require large datasets to build good object classifiers [6]. Previous studies [7] have shown that the classifiers like support vector machine(SVM), Decision tree (C4.5 algorithm) and Naïve Bayes classifiers can be used to predict surface textures with the helpof opto-tactile sensors. Though we do not use any sensors in thispresent work, usage of the above mentioned algorithms has beenreported on predicting multi-joint articulation. Classification studies have been employed for different purposes in different areas. Electromyography (EMG)-based classification has been used to classify finger movements [25], which has been used in several learning techniques such as perceptron linear separation and a back-propagation type neural network [26]. K-nearest neighbor (KNN) classifier and genetic algorithm (GA) showed less misclassification [26] compared to perceptron linear separation or back-propagation neural network. Here, in this paper, we have used linear SVM, nonlinear SVM and Naïve Bayes classifiers on datasets which were generated using an online virtual lab simulator [8, 9] for predicting movement of the articulator. The robotic arm is part of remotely-triggered experiments available online (http://amrita.edu/virtuallabs) and was used to generate the dataset on which these algorithms were tested.

High dimensional problems like interacting with humanoid robots is very complex and would be inappropriate for some machine learning approaches [27]. For simple robotic hand articulation with lesser dimensionality, machine learning methods can be effectively used to classify the accuracy of its movement. Many sophisticated prosthetic devices have been developed for amputees and paralyzed individuals due to recent advancements in bioengineering. For these type of devices real-time classification of bio-signals is required. Real time control of robotic arm has been shown using EMG signals [28]. Certain ongoing studies on controlling robotic prosthetics have shown that EMG could be used as a useful non-invasive method. Studies shown that amputees and partially paralyzed individuals typically have intact muscles that they can exercise varying degrees of control over. The signals reaching the muscles could be used to control robotic devices such as prosthetic hands and limbs with multiple degrees of freedom [28].

The objective in this study was to substitute a trajectory targeting approach (e.g. Kalman filter (KF) trained Multilayer Perceptron (MLP), unpublished data) with a classifier. The drawback of predictor-corrector methods is that the computational cost was high when the dataset was nonlinear [21]. To substitute and complement such predictor- corrector models with simpler methods, we suggest a linear hyperplane or probability-based stochastic classifier through this study.

In order to test and validate our hypothesis of substituting with simpler classification schemes, we used two types of classifiers on generated datasets: SVM with linear and non-linear kernels and Naïve Bayes classifier. Linear SVMs are popular for their ability to classify large datasets in simple and fast way [10, 11]. Naïve Bayes (*NB*) classifier has been widely used as a supervised learning technique and is well-known for its ability to perform better than non-probabilistic classification approaches such as neural networks and decision trees [18, 19, 20]. NB classification assumes conditional independence and Bayes theorem to estimate probability values of success/failure of an event [12, 13].

In this study, the robotic articulator was developed using a low-cost approach and calibrations were done to implement a simpler version of a prosthetic device. The classificationalgorithms were tested on normalized datasets to classify a set of given attributes (motor values and end-effector coordinates) based on their ability to train the articulator to reach a '*known*' target. Normalized dataset contains both training and test data among which test data was a subset of the training data whose class label need to be predicted. We have used a polynomial kernel in the non-linear SVM algorithm. Alongside with our datasets, we also pre-tested the classifiers on standard datasets such as weather dataset [22] and compared it to other algorithmic methods.

After making preliminary observations on the robotic hand movement data, we tested its robustness on pre-fixed positions using a chess board as a reference. Autonomic chess games has attracted attention and robot chess definitely serves as test bed for understanding the constraints as well as an entertainment application [24]. Our 'chess-board' dataset was generated based on the ability of the robotic hand to pick up an object from one square and place it in another square depending on the type of movement. Main goal here was to test the robustness of the robotic hand but not on the identification of the object or on precise movement.

2. Methods

2.1 Robotic Articulator

A robotic arm was constructed using rigid links which have 5 degrees of freedom (DOF) and a grasper with 6-DOF (see Figure 1A, online experiment freely available at http://amrita.vlab.co.in/index.php?sub=3&brch=257&sim=1458&cnt=3171). Each link constitutes a servo motor with a torque of 17 kg-cm at 6 Volts. A microcontroller (Figure 1B) was programmed to generate PWM (Pulse Width Modulation) signals with a time period of 20ms and a duty cycle varying from 1ms to 2ms for controlling the motors of the articulator. Microcontroller was interfaced through a serial port (RS-232) to the PC from which the desired motor angles will be transferred to the controller (Figure 2). User may perform the experiment in two modes: firstly, forward kinematics: where the user provided motor angle and obtained the end-effector and secondly, inverse kinematics: where the user provided the end-effector and obtained the individual motor angles which were given as inputs to the microcontroller. With these motor angles, the microcontroller generated control signal for each motor.

This setup (with a gripper instead of a grasper) can be remotely-triggered via Amrita virtual labs and is part of the National mission project on Education through ICT of the Government of India [17].

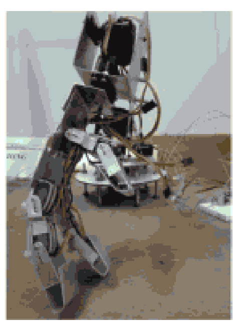
2.2 Chess Game

The board was designed with 64 squares, modeling a typical chess board. Each square was of size 2.3×2.5 cm. Since there were no visual tracking sensors, we identified the object initial positions joint angles through random adjustment. Suppose eight pieces are present in the first row, we identified their related joint angles, so that the hand comes and picks the object. These motor values were taken as source coordinates. The object was moved to a different square depending on the pattern of movement. In typical chess game, each object has a definite pattern to move on chess board that can be straight/diagonal/horizontal/vertical movements (see Figure 3). According to these movements, the object's target coordinates (motor values) were identified and allowed movement.

2.3 Generating Data

The datasets were obtained by two methods, simulation (synthetic dataset) using a kinematics approach [23] while the other was from the online virtual lab robotic tool using Cartesian coordinate measurements (experimental dataset). Datasets containing the motor angles of the 6 motors and their corresponding end-effector co-ordinates were generated using the experimental setup shown in Figure 1. The class label (y) determined whether supplied values indicated if the arm reached the target object. If the target was reached, class label was assigned as +1, otherwise the target was assigned as -1. For classification purposes, the motor values of the arm alone (5+1 motor values) were taken and values from the grasper were ignored. The mid-point of the grasper was taken as the end-effector coordinates (generated by both methods) used for classification.

A.



B.

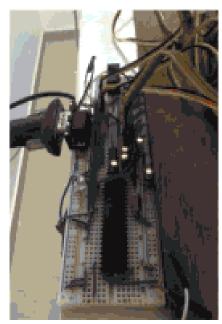


Figure 1. Setup. A. Robotic Articulator with 11 DOF. B. Microcontroller Setup

Min-Max method [15] was used to normalize the dataset (see Equation1).

$$Norm\_data = \left(\frac{data - \min(data)}{\max(data) - \min(data)} \right) \quad (1)$$

In Equation 1, “*data*” refers to the element of an attribute being normalized. “*min* (*data*)” refers to the element which has least value in an attribute (motor values and end effector coordinates) for all instances and “*max* (*data*)” refers to the element which has high value of the same attribute. “*Norm_data*” contains the normalized value.

For training and test purposes, both datasets were independently divided into training set and testing set using percentage-split method [16] with a split ratio of 66% (34% was taken as test data). Training data included the class label. While test data does not include a class label and the class label was predicted by the algorithms. Both datasets consisted of 6 motor values and 3 end-effector coordinate values (9 attributes, see Figure 4). A sample 3D plot with the attributes of 4 instances (in red) are shown in Table 1. We used linear SVM (LSVM) and Naïve Bayes implementations on both datasets.

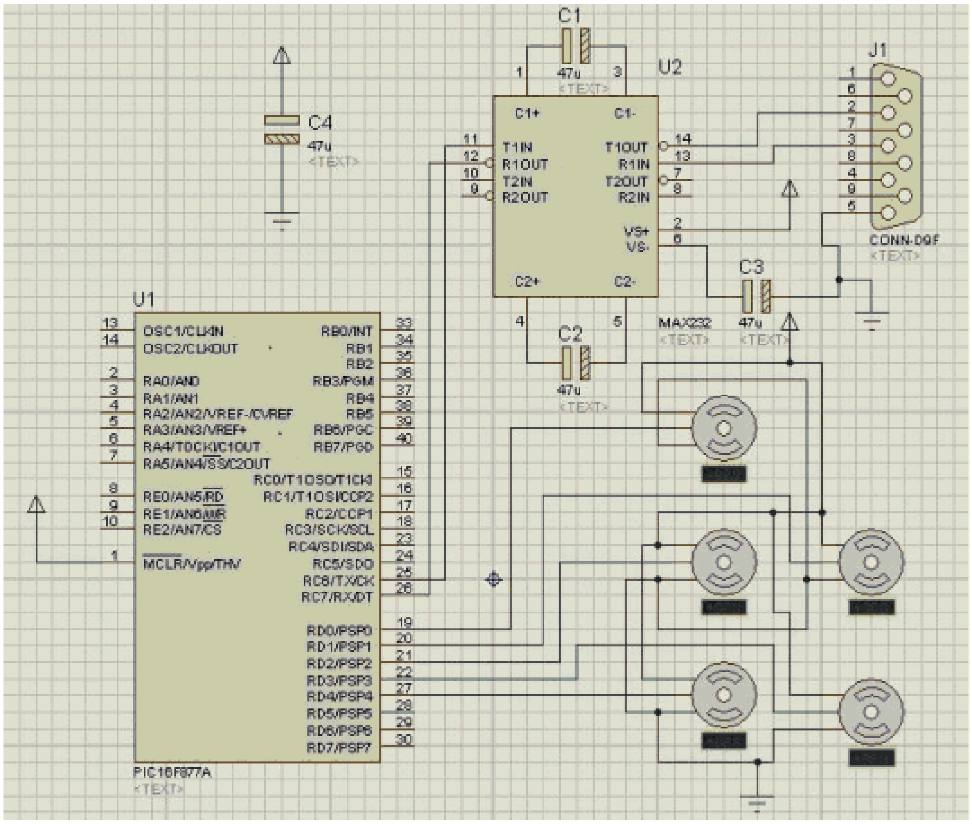


Figure 2. Schematics of the control circuit

A.

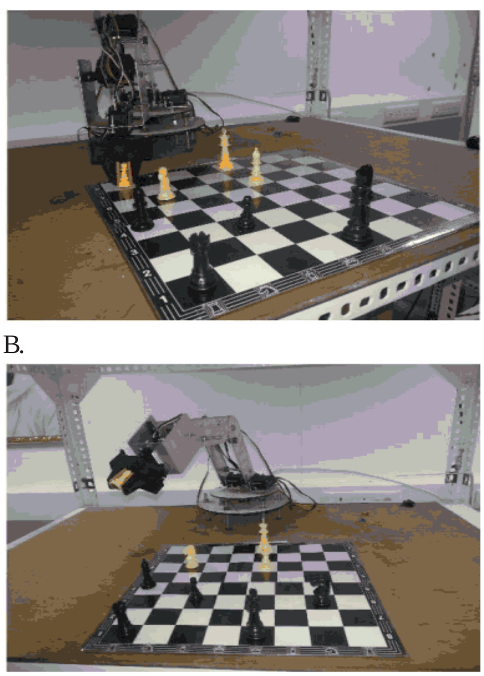


Figure 3. Chess Game. A. Robotic Arm grasping the object. B. Robotic Arm placing object at its target

2.4 Classification

Along with linear kernel in SVM, polynomial kernel was used as non-linear kernel for the dataset. Our formulation is as follows. A is normalized data obtained through Equation 1, polynomial kernel (polyA) is represented as in Equation 2.

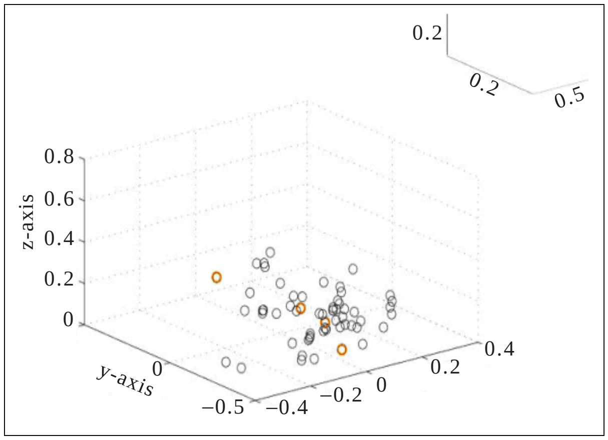


Figure 4. 3D plot of the synthetic dataset

$$poly\,A = (AA^T + 1)^4 \tag{2}$$

We substituted the obtained polynomial matrix into the algorithm and thus predicted the class labels for the test data. Consider Bayesian approach, when joint probability distribution is considered for 3 random variables case (A_1, A_2 and V)

$$p\,(A_1, A_2 \,|\, V) = p\,(A_1 \,|\, V) * \, p\,(A_2 \,|\, V) \tag{3}$$

$$p\,(A_1, A_2, ..., A_k \,|\, V) = \pi_{i=1}^{k}\, p\,(A_i \,|\, V) \tag{4}$$

In Naïve Bayes, each instance x is expressed as a conjugation of attribute values ($A_1, A_2 \ldots A_n$); $x = < A_1, A_2 \ldots A_n) >$. The target function $f(x)$ € V like $V = \{+, -\}$. In our implementation, $A_1, A_2 \ldots A_6$ are motor values, A_7, A_8, A_9 are the XYZ coordinates of the end effector and V is the class label with +1 or –1. The Naïve Bayes classifier is of the form.

$$V_{NB} = \underset{V_j \in V}{\arg max} \; [\![\Pi_{i=1}^{n} P(a_i | \in V_j)]\!] \, P(v_j) \tag{5}$$

Mot or1	Mot or2	Mot or3	Mot or4	Mot or5	Mot or6	X	Y	Z	Class label (y)
120	100	60	120	48	91	–0.03621	0.318427	0.168556	–1
128	67	67	158	112	74	–0.00332	–0.36114	0.168556	–1
99	100	40	160	98	80	–0.10043	–0.27861	0.259654	1
120	100	60	140	20	67	–0.03717	–0.31851	0.183259	–1

Table 1. Attribute values of the selected instances from synthetic dataset

For sake of reproducibility, the algorithmic implementation of SVM and NB classifiers is listed below.

Algorithm for linear SVM [11]:

1. Training data $X = \{x_1, x_2 \ldots x_n\}$ with class label y € $\{-1, 1\}$ was provided as input.

2. X along with a column vector (e) containing ones stored in a matrix V. $V = [X, -e]$

3. V was then multiplied with diagonal matrix (D) containing class labels. $H = D * V$

4. H was multiplied with its diagonal matrix (HTLet this matrix be $tranH = H * H^T$

5. Identity matrix (I) was generated with size of $tranH$ and divided with v which has value of 0.1, let this matrix be M. $[I / v]$

6. M was added with $tranH$ matrix and stored in N. Now we take inverse of N. $N = (I / v + tranH)^{-1}$

7. Obtain the final matrix using the below equation.

$$u = v * \left(I - H * \left(\frac{1}{v} + H^T H \right)^{-1} * H^T \right) * e$$

8. Finally, w and γ values were obtained. And used them in the equation $w^T * x - \gamma$ to classify the test dataset.

Algorithm for Naïve Bayes [12]:

1. Setting the data
   a) Each instance consists of n attributes $X = \{x_1, x_2 \ldots x_n\}$.
   b) Each of the instance was classified into different classes, C_j where $j = 1, 2 \ldots m$.

2. Train the dataset
   a) Find the probabilities $p(C_j)$.
   b) Using conditional independence, calculate

$$p(X \mid C_j) = \pi_{i=1}^{k} \, p(x_k \mid C_j)$$

3. Predict test data

$$\text{a) } p(C_j \mid X) = \frac{p(X \mid C_j) * p(C_j)}{p(X)}$$

$$\text{b)Maximize } p(C_j \mid X) = p(X \mid C_j) * p(C_j)$$

4. The error is estimated as follows
a) error = (sum of misclassified data)) / Total number of data_points $*100$

2.5 Chess Game Dataset
Our classification dataset was made up of source coordinates, target coordinates and class label. Class label was assigned depending on the source coordinates. A sample of the dataset is shown in Table 2. Class labels range from 1 to 5 depending on the pattern of movement. For classification purpose, we used WEKA interface directly called from MATLAB. Figure 5 shows the flowchart explaining the extraction of the individualrecords from dataset and assigning class label to it.

2.6 Validation
For validation of implemented algorithms, we used the weather dataset [16] and obtained 78% accuracy for linear SVM, 65% for nonlinear SVM (NLSVM) and 60% for Naïve Bayes. These results were crosschecked with other implementations ofsame algorithms in WEKA (Waikato Environment for Knowledge Analysis) [16].

3. Results

The arm, developed with 5 + 6 joints, was able to grasp objects of around 100g (input voltage is 7.5V for each motor) at a given position and has been tested repeatedly to check if there were errors in '*reaching*' a given position. If the object weight exceeded beyond 120g (given 7.5V as input voltage to motors), the arm started to stagger. Forward kinematics implementations allowed generating data which was similar to the experimental data recorded from the device.

We tested different algorithms on training and test datasets using different percentages of split (see Figure 6A, 6B and 6C). We observed that with increase in the percentage of split the training data efficiency decreases whereas test data efficiency

mo	mo	mo	mo	mo	mo	mo	mo	mo	mo	mo	mo	Class
1	2	3	4	5	6	1	2	3	4	5	6	
90	111	116	36	138	5	90	100	105	47	138	5	1
90	111	116	36	138	5	90	89	94	58	138	5	1
90	111	116	36	138	5	90	78	83	69	138	5	1
90	111	116	36	138	5	90	67	72	80	138	5	1
90	111	116	36	138	5	90	56	61	91	138	5	1
100	111	116	36	128	5	115	89	94	58	118	5	2
100	111	116	36	128	5	90	89	94	58	138	5	2
100	111	116	36	128	5	90	78	83	69	138	5	2
115	111	116	36	128	5	90	89	94	58	138	5	3
115	111	116	36	128	5	140	89	94	58	98	5	3

Table 2. Sample Dataset showing the attributes and respective class label

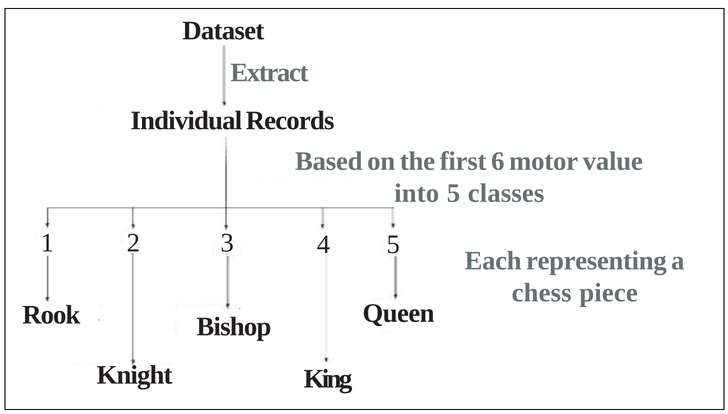


Figure 5. Flowchart showing class label assignment to individual records extracted from dataset

increases. Nonlinear SVM showed significantly higher test data efficiency when compared to linear SVM and NB classifiers. We have used 66% split to compare the training and test data efficiency as well as error since the results for different percentages were almost similar (Figure 6). Table 3 shows the different efficiencies observed based on the methods used (LSVM, NLSVM or NB classifier) with 66% split. Figure 7A shows the comparison of training and test efficiency with different methods. Figure 7B shows the comparison between training and test error. In both cases (training and test), SVM with polynomial kernel of degree 4 showed very less error (Figure 7B) and gave reasonably higher correct classification (Figure 7A). Linear SVM showed a close accuracy to nonlinear SVM in both cases (training and test), NB classifier showed relatively less percentage of efficiency with training data, however the test data showed improved efficiency suggesting that it can be used as a good classifier.

We compared test data efficiency between synthetic and experimental datasets. Classifiers on experimetal dataset showed higher performance in case of linear and nonlinear SVM when compared to the same on synthetic dataset (Figure 8). However the efficiency of NB classifier showed higher perfomance in case of synthetic dataset than on experimental dataset (Figure 8).

The results suggest that linear SVM and NB classifiers may be used to classify small articulator movement datasets with reasonable efficiency with reduced computational cost unlike nonlinear kernels. The simulation time required for the training of synthetic dataset was as follows, 0.741883s for linear SVM, 0.766669s for nonlinear SVM and 0.136662s for NB classifier.

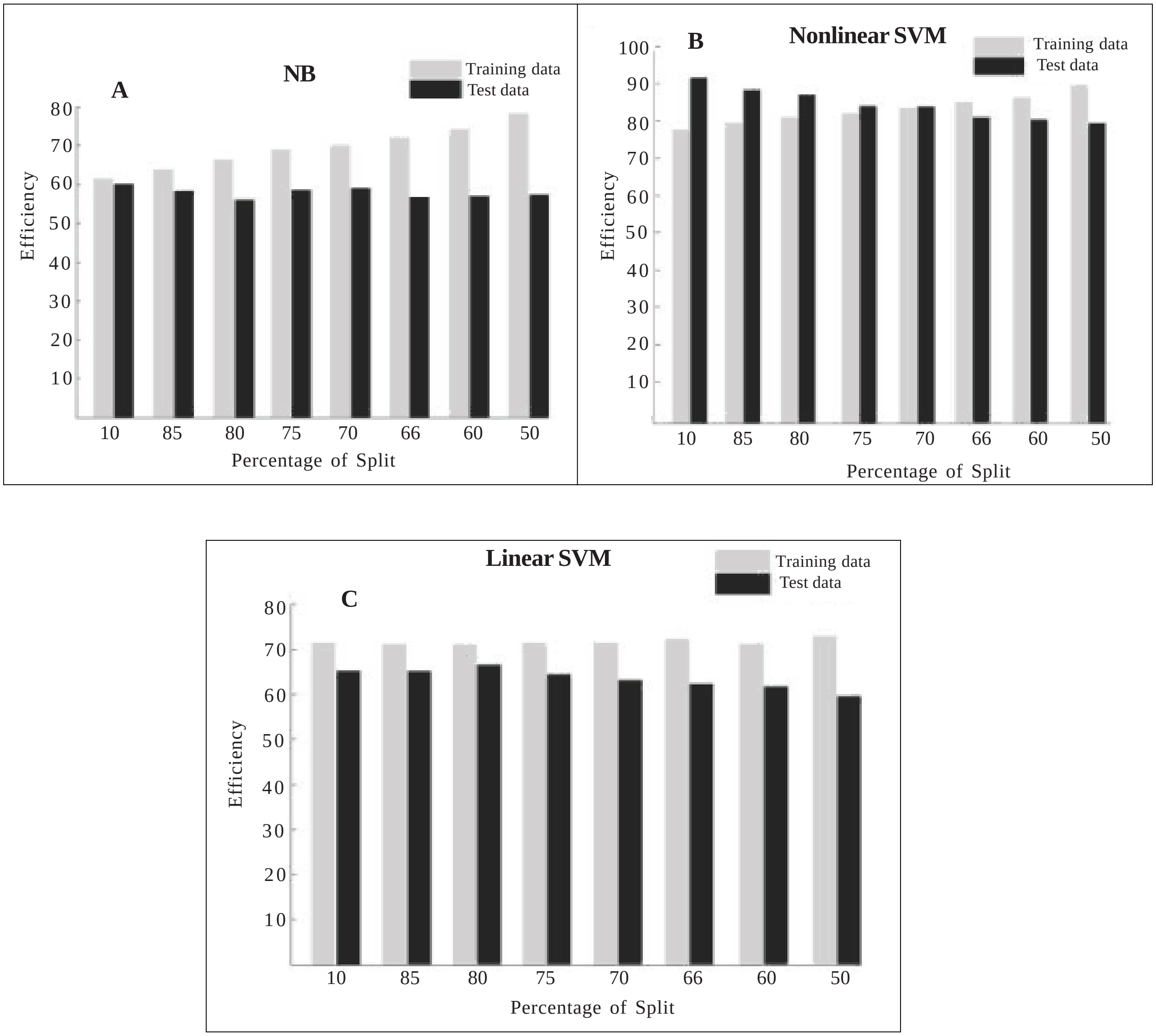
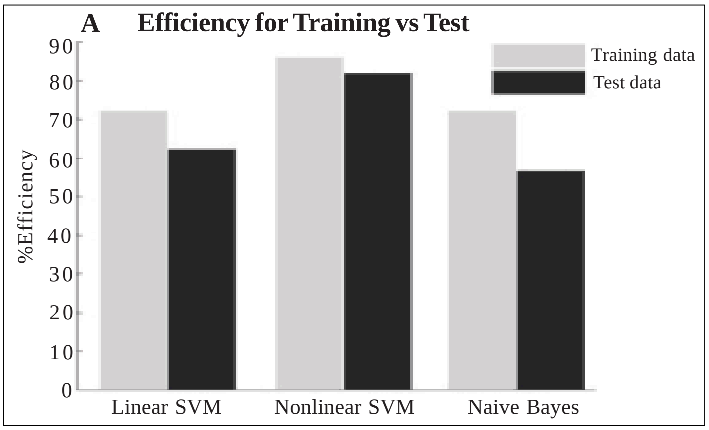


Figure 6. Classification of robotic datasets. A. Naïve Bayes (NB) classifier efficiency over different percentages of split. B.Nonlinear SVM efficiency. C. Linear SVM efficiency

Experimental dataset training showed the simulation time as follows, 0.880s for linear SVM, 1.13s for nonlinear SVM and 0.17 s for NB classifier. All simulations were done on Dell T3500 workstation with Ubuntu Linux 13.04 (SMP x86_64 kernel 3.8.0-19-generic) operating system on a Intel XeonW3670 CPU (frequency of 3.20 GHz) with 8 GB RAM. These observations indicate linear and NB classifiers take less computational time when compared to Kalman filter trained MLP (KF-MLP, data not shown).

	Testing			Testing		
	LSVM	NLSM	NB	LSVM	NLSVM	NB
Efficiency of the data (%)	62.2	82	56.8	72.1	86	72.1

Table 3. Efficiency in percentages for the dataset with different algorithms



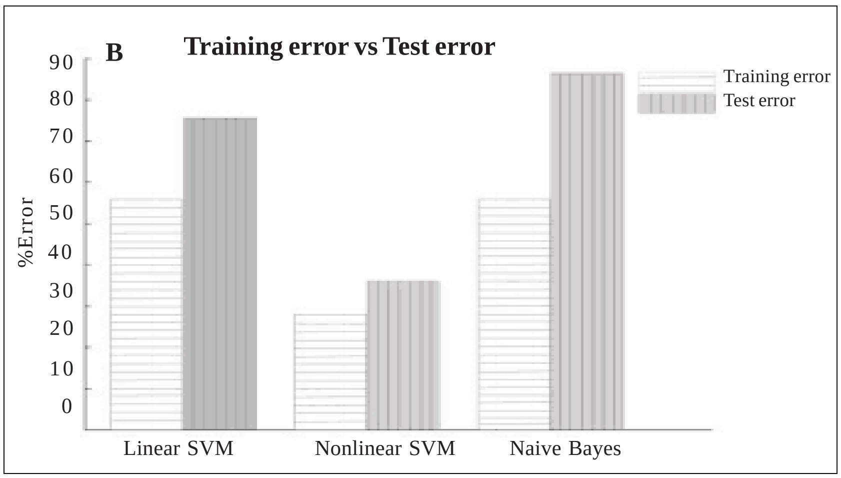


Figure 7. A. Comparison of Training (gray) vs Test (black) efficiencies. B. Comparison of Training vs Test error

With the chess dataset, we were able to see the trajectory of the hand movement follow what was predicted. We had taken one-third of the whole dataset as test dataset and observed that the arm was able to mimic similar movement.

Classification accuracy has been obtained for training as well as testing. We tested with Naive Bayes, linear SVM, nonlinear SVM (kernel 3) and nonlinear SVM (kernel 4). We observed that nonlinear SVM (kernel 3) showed highest classification accuracy (85%) followed by linear SVM with classification accuracy of 76%. Naïve Bayes showed leastaccuracy with 72% while nonlinear SVM (kernel 4) showed 73% accuracy. Classification results are tabulated in Table 4.

Algorithm	Classification Accuracy (%)
Naïve Bayes	72
SVM (linear)	76
SVM (nonlinear kernel-3)	85
SVM (nonlinear kernel-4)	73

Table 4. Classification accuracy by taking average of 10 trials

4. Discussion

We have attempted using a machine learning approach rather than detailed bio-inspired approach wherein we used a linear

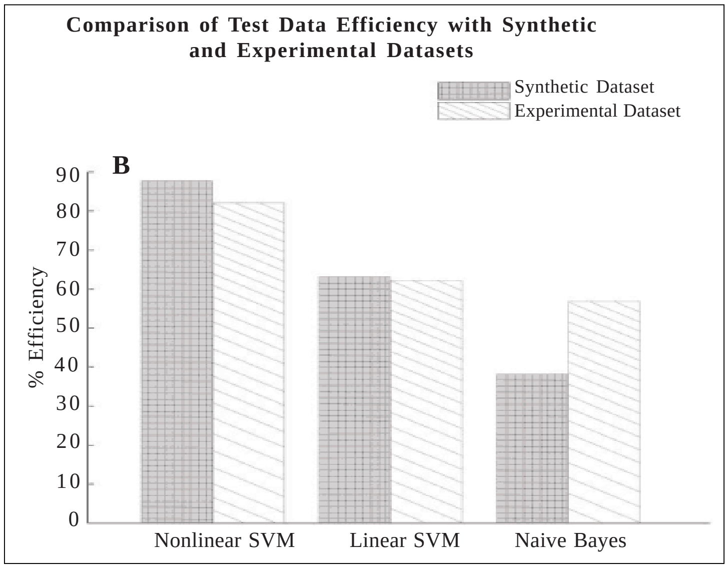


Figure 8. Comparison of test data efficiency between synthetic and experimental datasets

classifier to define the movement prediction in a robotic articulator. The robotic arm was constructed with an idea of achieving maximum accuracy to reach the given target. We used 70 point dataset in this study and showed significant (82%) classification accuracy. In the experimental dataset, the Denavit-Hartenberg (DH) parameters (D) [23] were generated from the kinematics model on which the forward kinematics was applied to obtain transformation matrix (a combination of rotational and translation matrix (end effector co-ordinates)). In our approach of classification, we replaced this matrix with a weight matrix (W) of dimension 9×1 in the linear hyper-plane classifier. The generated datasets were validated through repeatability allowing the robotic online articulator to be used for extracting the datasets related to movement classification studies. The intentional lack of precision in some data points was due to design restrictions of allowing the consideration of the roboticarm as a low cost prosthetic device. For our classificationpurpose, linear SVM proved to be a promising approach rather than nonlinear SVM since nonlinear SVM has higher computational cost. While on the other hand, NB showed higher percentage of error, however the test dataset after training showed relatively less error. We prevented over-learning by decreasing number of training instances and found our classification levels did not vary drastically (test efficiency was 87.5% with 40 data points, probably due to overlearning).

Our classification results indicated that certain linear classifiers perform with comparable efficiency as nonlinear classifiers thereby allowing us to use simpler implementations without any other kernel methods. Alternative implementation in MATLAB provided similar results indicating that computational cost may be overcome by introducing the appropriate modelling techniques. Our intention in keeping the dataset small was to verify whether the classifier could identify the generalization properties exhibited by the kinematics model. Efficiency of the classifier is strongly dependent on the number of training vectors since our training data size was significantly small ($n = 70$), there will be no overlearning.

5. Conclusion and Future Work

With this efficiency, we find that typical predictive-correction model such as KF-trained MLP algorithms which are computationally expensive may be replaced using simpler linear SVM classifiers. We are currently extending this work with an alternative approach of using a spiking neuron network model (CIS-NN, unpublished data) substituting firing rate as a metric to measure trajectory patterns.

6. Acknowledgment

This work derives direction and ideas from the chancellor of Amrita University, Sri Mata Amritanandamayi Devi. This work is partially supported by grants SR/CSI/49/2010 and SR/CSI/60/2011 from DST, BT/PR5142/MED/30/764/2012 from DBT, Sakshat Virtual labs, NMEICT, Government of India and also by Indo-Italy POC 2012-2013. Authors would like to thank our colleague Dhanush Kumar, Sandeep Bodda, Harilal Parasuram, Manjusha Nair, Nidheesh Melethadathil, Latha AR, Rakhi Radhamani,

Gopika Sujatha, Akhila Sekhar and Hemalatha Sasidharankurup for support with this study.

References

[1] Lee, J., Knox, W. B., Stone, P. (2008). Inter-Classifier Feedback for Human-Robot Interaction in a Domestic Setting, *Journal of Physical Agents*, 2, p. 41-50, June.

[2] Tejomurtula, S., Kak, S. (1999). Inverse kinematics in robotics using neural networks, *Information Sciences*, 116, p. 147-164.

[3] Saal, H. P., Ting, J. A., Vijayakumar, S. (2010). Active Estimation of Object Dynamics Parameters with Tactile Sensors, *In*: Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS 2010), Taiwan.

[4] Mitrovic, D., Nagashima, S., Klanke, S., Matsubara, T., Vijayakumar, S. (2010). Optimal Feedback Control for Anthropomorphic Manipulators, *In*: Proc. IEEE International Conference on Robotics and Automation (ICRA 2010), Anchorage, Alaska, USA.

[5] Ho, E., Komura, T., Ramamoorthy, S., Vijayakumar, S. (2010). Controlling Humanoid Robots in Topology Coordinates, *In*: Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS 2010), Taiwan.

[6] Jia, Z., Saxena, A., Chen, T. (2011). Robotic Object Detection: Learning to Improve the Classifiers Using Sparse Graphs for Path Planning, *In*: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence.

[7] Mazid, A. M., Shawkat Ali, A. B. M. (2008). Opto-tactile Sensor for Surface Texture Pattern Identification using Support Vector Machine, 10th Intl. Conf. on Control, Automation, Robotics and Vision Hanoi, Vietnam, p. 17–20, December.

[8] Nair, B., Krishnan, R., Nizar, N., Radhamani, R., Rajan, K., Yoosef, A., Sujatha, G., Radhamony, V., Achuthan, K., Diwakar, S. (2012). Role of ICT-enabled visualization-oriented virtual laboratories in Universities for enhancing biotechnology education – VALUE initiative: Case study and impacts, *FormaMente*, 7 (1-2).

[9] Diwakar, S., Achuthan, K., Nedungadi, P., Nair, B. (20101). Enhanced Facilitation of Biotechnology Education in Developing Nations via Virtual Labs: Analysis, Implementation and Case-studies, *International Journal of Computer Theory and Engineering*, 3 (1) 1-8.

[10] Ladicky, L., Torr, P. H. S. (2011). Locally Linear Support Vector Machines, *In*: Proceedings of the Twenty-Eighth International Conference on Machine Learning.

[11] Cortes, C., Vapnik, V. (1995). Support-Vector Networks, *Machine Leaming*, 20, p. 273-297.

[12] Zhang, H. (2004). The Optimality of Naive Bayes, FLAIRS2004 conference.

[13] John, G. H., Langley, P. (1995). Estimating Continuous Distributions in Bayesian Classifiers, *In*: Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, San Mateo, p. 338-345.

[14] Wang, J., Downs, T. (2005). Improving Support Vector Machine Performance on a Problem in Object Recognition, *Complexity International*, v. 12.

[15] Han, J., Kamber, M. (2006). Data Mining: Concepts and Techniques, Second edition, Morgan Kaufmann, USA.

[16] Holmes, G., Donkin, A., Witten, I. H. (1994). Weka: A machine learning workbench, *In*: Proc Second Australia and New Zealand Conference on Intelligent Information Systems, Brisbane, Australia, Retrieved 2007-06-25.

[17] Diwakar, S., Achuthan, K., Nedungadi, P. (2010). Biotechnology Virtual Labs – Integrating Wet-lab Techniques and Theoretical Learning for Enhanced Learning at Universities, *In*: Proceedings of the International Conference on Data Storage and Data Engineering (DSDE 2010), Bangalore, Feb 10.

[18] Aviles-Arriaga, H. H., Sucar, L. E., Mendoza, C. E., Vargas, B. (2003). Visual Recognition of Gestures using Dynamic Naïve Bayesian Classifiers, *In*: Proceedings of the 2003 IEEE international workshop on Robot and Human Interactive Communication, Millbrae, California, USA.

[19] Michie, D., Speigel Halter, D. J., Taylor, C. C. (1994). Machine Learning, Neural and Statistical Classification, *Ellis Horwood Series in Artificial Intelligence*, England.

[20] Mitchell, T. M. (1997). Machine Learning, *McGraw Hill Series in Computer Science*, USA.

[21] Chen, Y., Oliver, D. S., Zhang, D. (2009). Data assimilation for nonlinear problems by ensemble Kalman filter with reparameterization, *Journal of Petroleum Science and Engineering*, 66, p. 1-14.

[22] Witten, I. H., Frank, E. (2005). Practical Machine Learning Tools and Techniques, Morgan Kaufmann, San Francisco.

[23] Roh, H. S., Kim, J. O. Manipulator Modeling from D-H Parameters, 30th Electronics Society, Busan, Korea

[24] Matuszek, C., Mayton, B., Aimi, R., Deisenroth, M., Bo, L., Chu, R., Kung, M., LeGrand, L., Smith, J., Fox, D. (2011). Gambit: An autonomous chess-playing robotic system, in Robotics and Automation (ICRA), *IEEE International Conference*, p. 4291-4297.

[25] Uchida, N., Hiraiwa, A., Sonehara, N., Shimohara, K. (1992). EMG pattern recognition by neural networks for multi finger control, *In*: Proc. Ann. Eng. Med. Biol. Sco., p. 1016-1018.

[26] Pelag, D., Braiman, E., Yom-Tov, E., Inbar, G. F. (2002). Classification of Finger Activation for Use in a Robotic Prosthesis Arm, *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 10 (4).

[27] Fod, A., Mataric, M. J., Jenkins, O. C. (2002). Automated Derivation of Primitives for Movement Classification. *Autonomous Robots*, 12 (1) 29-54.

[28] Crawford, B., Miller, K., Shenoy, P., Rao, R. (2005). Real-Time Classification of Electromyographic Signals for Robotic Control. Technical Report No. 2005-03-05, *Department of Computer Science*, *University of Washington*, March.