Biobots Motion Classification

Anamika Thakur
North Carolina State University
athakur3@ncsu.edu

Shankara Narayanan Sethuraman
North Carolina State University
ssethur2@ncsu.edu

Jordan Campbell
North Carolina State University
jhcampbe@ncsu.edu

Abstract

In this study, aggregated sensor data on cybernetic Madagascar hissing cockroaches, called 'biobots' is used to understand their motion. Biobots have four states: stationary, random motion, clockwise motion and counter-clockwise motion. The performance of two standard learning algorithms – Fine K-nearest neighbor (Fine KNN) and Cubic Support Vector Machine (Cubic SVM) in this classification task have been compared and analyzed.

1. Introduction

The need to develop miniature, robotic entities that can carry a wide range of sensors and function effectively under complex, nonhomogeneous environments emerging after for example a natural disaster is very critical.

However, current technology falls short in offering mobile robotic agents that can do the job. Insects, on the other hand, exhibit an unmatched ability to navigate through a variety of environments via efficient locomotion. [2]

The Cyber Internet Networks project at the ARoS Laboratory, NCSU, uses cybernetic Madagascar hissing cockroaches, called 'biobots' and aggregates data from several biological and synthetic sensors periodically to analyze their natural movement.

The aim of the current project is to attempt to understand the aggregated biobot data and develop an approach to classify their state.

Fig. 1 Cyborg Insect Networks[2]

2. Methodology

The goal of the project is to classify the different motions of the biobots. Aggregated data consists 10000 samples of 42 features each computed over a 1 second window with 75% overlap :

The Accelerometer and Gyroscope values of the biobots are tabulated and the following parameters extracted :

1:6 - Mean

7:12 - Variance

13:18 - Skewness

19:24 - Kurtosis

25:39 - Cross Correlation

The last 3 features consist of Gyroscope Energy.

In addition, a ground truth table is also provided which consists of discrete values from 0-3 to indicate the state of the biobot at that particular timestamp.

The four states or the four motions that have been considered for classification are :

0 – Stationary / static

1 – Random motion in the arena

2 – moving clockwise on the boundary of the arena

3 – moving counter clockwise on the boundary of the arena

The Classification Learner Application in MATLAB has been used in this project to understand how a wide range of classifiers perform on the data. Two classifiers that perform the best were identified and their F1 scores computed to understand better how their performance is. Further, the trained classifiers were extracted for testing with future data.

Two families of learning algorithms have been considered: K-nearest neighbors and Support vector machines.

2.1 K – Nearest Neighbors

The KNN algorithm is a type of instance based learning which classifies the test data points based on the closest training data points. It is the simplest of all machine learning algorithms. A KNN classifier uses K training samples that are closest to the test sample to classify it. Here the samples are classified based on their distance to points in a training dataset. [3]

The difference between different types of KNN algorithms is that they either vary in the number of neighbors used or in the distance metric or distance weight that is utilized.

The Fine, Medium and Coarse KNN algorithms all use Euclidean distance but have different number of neighbors.

Weighted KNN also uses Euclidean distance but uses squared inverses as distance weights. Cubic KNN uses Minkowski (cubic) distance and Cosine KNN uses cosine as the distance metric.

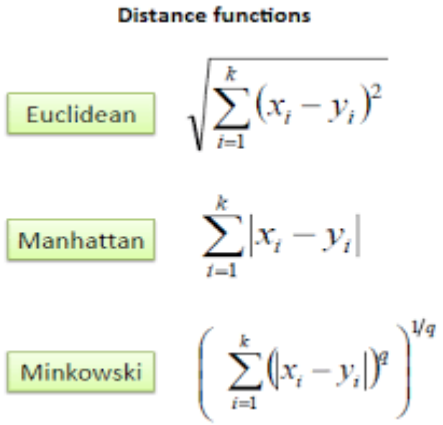

Fig 2. Distance Functions used in KNN algorithms

2.2 Support Vector Machines

A support vector machine is a classifier which constructs a hyperplane for the purpose of classification. It is a form of supervised learning which, given labeled training data, categorizes the testing data to the optimal hyperplane. It classifies the data by finding the best hyperplane that separates all data points of one class from the other classes. The goal of SVM is to produce a model which will predict the labels of test data accurately. [4]

Sometimes it is possible that the training data is not linearly separable. A kernel is used under these circumstances to classify the data.

The different types of SVM classifiers available in classification learner in MATLAB are - Linear SVM, Quadratic SVM, Cubic SVM, Fine Gaussian SVM, Medium Gaussian SVM and Coarse Gaussian SVM. They all have different kernel functions or kernel scale.

Linear SVM has a linear kernel; quadratic SVM has quadratic kernel, cubic SVM has cubic kernel; fine, medium and coarse Gaussian SVM have Gaussian kernel but a different kernel scale. Kernel scale is related to how spread out the data points are.

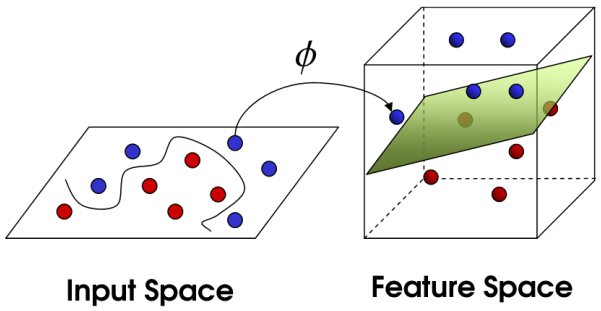

Fig 3. Result of application of a Kernel to data in SVMs

3. Results

We first run all the different SVM and KNN family classifiers over our data with 5-fold validation.

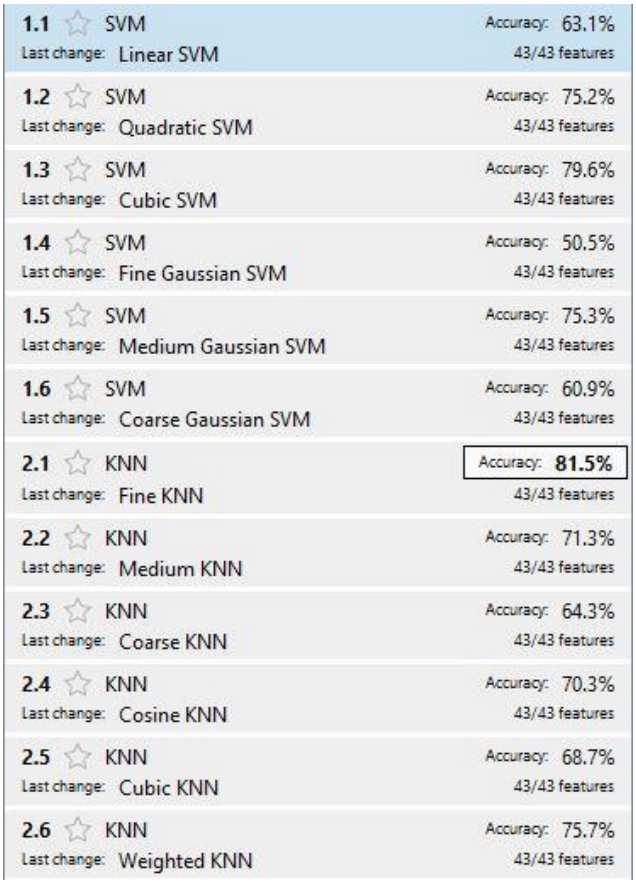

Fig.4 Accuracy results after training with different classifiers

It is observed that the Fine-KNN algorithm is the most accurate in classifying the data followed by the Cubic-SVM algorithm.

The Confusion matrices are extracted from the classification learner to compute the F1 scores for each algorithm:

a)

b)

Fig 5 a) Confusion matrix of Cubic SVM b) Confusion Matrix of Fine KNN

```
>> precision = .82 + .81 + .72 + .78;
>> recall = .84 + .84 + .66 + .73;
>> F1 = ( precision * recall ) / ( 2 * ( precision + recall ) )

F1 =

    0.7749

>> precision = .81 + .85 + .74 + .79;
>> recall = .8 + .86 + .75 + .77;
>> F1 = ( precision * recall ) / ( 2 * ( precision + recall ) )

F1 =

    0.7962
```

Fig.6 F1 scores of Cubic SVM and Fine KNN

It is observed that the Fine-KNN algorithm has a higher F1 score (~.796) that the Cubic-SVM algorithm (~.775).

4. Analysis

We used all available features, including mean, variance, skewness, kurtosis, cross correlation, and gyro energy. Through trial and error, we found Fine KNN and Cubic SVM to yield the best results.

4.1 KNN

KNN is a very simple algorithm. For any given new input vector to be classified, the most frequently occurring class out of its k neighbors is chosen to classify the input. This works very well for our application because one of the main objectives is to distinguish between the biobot being in the middle or on the boundary.

Thus, KNN can easily classify whether the biobots neighbors are more likely in the middle or on the boundary. Classifying clockwise vs. counterclockwise is much more difficult to classify and this is reflected in the precision and recall. For example, the first two numbers in recall (.81 and .85) are higher than the last two (.74 and .79). This indicates classification was much more accurate for determining if the biobot was stationary or in the middle than the other two scenarios, moving clockwise and counterclockwise on the boundary.

Classifying if the biobot is stationary is easily observed based on the accelerometer and gyro values. When the biobot is stationary, the accelerometer and gyro values will be nearly zero.

There are several KNN classifiers to choose from. But, Fine KNN worked best. Fine KNN models a fine distinction between classes with the number of neighbors, k, set to 1. All of the other KNN models yield medium distinctions

between classes, with k = 10, or in the case of Course KNN, k = 100 [1].

More neighbors essentially add more noise to the results because if there are a high density of neighbors in another class nearby, the new input will be wrongly classified as the higher density class even though it's nearest neighbor is in another class.

4.2 SVM

The reason Linear SVM doesn't work well is because it aims to separate classes with a simple linear line, but our classes are not linearly separable due to the fact that the biobot is moving on the boundary of a circle. The other SVM models differ based on internal parameters such as kernel function, box constraint level, kernel scale mode, and multi class method [1].

A mathematical proof of why Cubic SVM's parameters worked better for this particular application is beyond the scope of this paper. In the F1 score below, a similar result to Fine KNN is observed where the classification accuracy of being stationary or in the middle of the arena is much higher than moving on the boundary counterclockwise and clockwise.

4.3 Fine KNN vs. Cubic SVM

Observing the precision and recall values, we understand that while for class 1 the cubic SVM algorithm performs better than the fine KNN algorithm, the same is not true for the rest of the classes. The difference is stark in the recall value for class 3. This difference in performance can be attributed to the nature of the dataset and number of data points where the true label is from a particular class.

It is clear that there are more sample from class 2 (4434) than class 1 (1872), class 3 (1327) or class 4 (1675). Since Class 3 has the least number of true values (< 15% of total), the performance of both classifiers are the worst in said class. Class 4 with 18% of all samples has much better performance. There is further improvement in Class 1 (20% of total) from both classifiers. Class 2 which has ~ 50% of all samples in it has slightly better performance than class 1. Hence we understand that if each class has at least 20% of true values in the dataset, their performance would be over .8 in both precision and recall for both the algorithms.

Finally, it is clear that the Fine-KNN algorithm is ~2% more efficient (+.02 in F1 score) for the entire problem. This can be attributed to the way both the algorithms accomplish the classification task. All of the four states in the problem are continuous in nature. Hence, at any given time the state is affected by the previous state or neighbors. Therefore, the KNN algorithm performs better overall and is more suited to this type of problem.

We can also expect the accuracy to increase for both algorithms if the number of samples per state are more equal. The same can be said for increasing the total number of samples.

5. Conclusion

In this project two different classification approaches have been utilized to classify the motion of the biobots in four different states. The algorithms used for classification were Cubic SVM and Fine KNN.

Based on the F1 score, which has been used to evaluate the performance of the classifier, Fine KNN was found to be more effective than Cubic SVM.

From the confusion matrix it was evident that it was far easier to classify the movement into states 0 and 1 (stationary and moving within the arena) than to classify when the biobot moves clockwise or counterclockwise on the boundary.

The accuracy of the classifier could be improved by including the temporal structure of locomotion in the list of features, and making the number of samples per state more even.

References

[1] "Choose Classifier Options." *Choose Classifier Options*. Mathworks, n.d. Web. 13 Nov. 2016. <https://www.mathworks.com/help/stats/choose-a-classifier.html?requestedDomain=www.mathworks.com>.

[2] "Cyborg Insect Networks | ARoS Lab." ARoS Lab. N.p., n.d. Web. 13 Nov. 2016.

[3] k-nearest neighbors algorithm (https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm)

[4] Support vector machine (https://en.wikipedia.org/wiki/Support_vector_machine)