THE EFFECT OF BIAS IN TRAINING DATA

Brian Hawkins
Sensor Signal and Information Processing (SenSIP) Center RET

Abstract – **Machine learning is only as good as the data that creates it and given the mistrust of AI and big data by the public recently, it is useful to explore scenarios where machine learning might give misleading results. In this RET study, I will study how data can be biased in multiple ways to demonstrate to students how this can be demonstrated in the confusion matrix and corresponding scatter plots.**

Index Terms: Machine Learning, Supervised Learning, Classification

I. INTRODUCTION

In supervised classification, training samples from known data distributions are tested with the goal to identify a classification boundary that separates these classes. New data samples with known labels can then be identified as belonging to one of these classes. Support vector machines (SVMs), decision tree, random forest, and other algorithms are used to solve this problem.

The training data in this scenario is provided by standard data sets, common objects meeting the color criteria, or phone applications capable of creating specific color profiles. Each of these situations will be explored for unbiased and biased data. Biased data will be created by not normalizing the data, sampling unequal data sets, and intentionally creating testing sets that are not representative of the training population to be modeled. To create biased test data, values will be added to one of the color channels in the data file and then effect will be analyzed using scatterplots and the corresponding confusion matrices. Bias introduced into data will have the intent of confusing the decision boundary between sets of data.

In this project, I first performed a literature review of the existing work in this area, examples of its use, and possible future application. I then explored the limitations of the color sensor on the Arduino BLE 33 Sense Board in an attempt to design a scenario that would produce repeatable results during training and testing.

Many problems are susceptible to various types of bias. Measurement bias is well known in the recidivism risk prediction tool COMPAS which was a factor leading to higher false positive rates for black versus white defendants. [1][6]. Population basis has been documented in ImageNet where 45% of the images are from the United States and a majority of the remaining portion are from North America or Western Europe while 3.2% are from China and India combined. [1]

Many of the journal articles highlighted the importance of careful data set selection. One study examined 10 commonly used datasets to examine labeling errors. there were estimated to be an average of 3.4% errors across the datasets.[4] Another source examined testing data sets against each other to test their performance in an attempt to categorize the types of bias that can be present along with suggestions about how to improve datasets.[6] Although my test data only dealt with one color, it did demonstrate the importance of a data set selection. Although dealing with color in images and while using a deep neural network, this source examines the importance of white balance in training image classification recognizes the role this plays in the correct identification of an object.[7]

II. RELATED WORK

In one application Support Vector Machines were trained to detect if tomatoes were mature. Even though the work used multiple features and various lighting conditions, the recall was 96.85% and the precision 98.40%. [8] The focus was to have a robot capable of picking ripe tomatoes. Another application that involved a color sensor but not machine learning was detecting a flame in a video. [9] The purpose of this project was for fire detection in real world situations and movies. They also are hoping it might be useful in forensic and fire capture for computer graphics. While although a third application did not use a color sensor, but did use the RGB colors from a color video camera to detect the minerals chalcopyrite and molybdenite in flotation froths, mineral slurries, and dry mineral mixtures. [10] The focus was on sensor to monitor a difficult situations, although they recognized further refinement was needed.

A fourth application involved the use of the Nano BLE 33 Sense board and the color sensor to determine the color of M&M's as they fall through a tube. [11] Although this is not a commercial application, any objects that would require color identification during a process could use the technology demonstrated in this project.

III. METHODS

To begin this project, my aim is to understand how data can be corrupted or biased and to examine the effect in the corresponding confusion matrices and scatter plots as they are compared to non-biased data. Objects were classified based on several different supervised machine learning algorithms, although the focus was on examining the support vector machine algorithm.

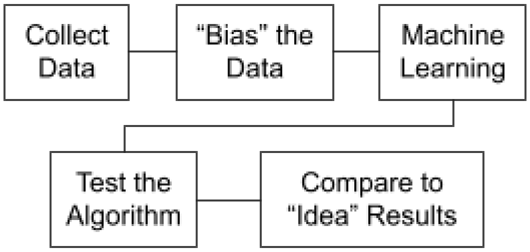

Figure 1: Project Flowchart

The project started with the collection of color data using a variety of sources to characterize the behavior of the color sensor. Color data sources were chosen starting with a uniformity of color and texture and moving towards surfaces that had less uniform color and texture. The Arduino BLE 33 Sense Board included headers so I was able to attach it to a breadboard which included an adhesive backing that I attached to a Lego panel. I built a simple platform out of Legos with wheels to maintain a uniform distance between the sensor and objects to be scanned.

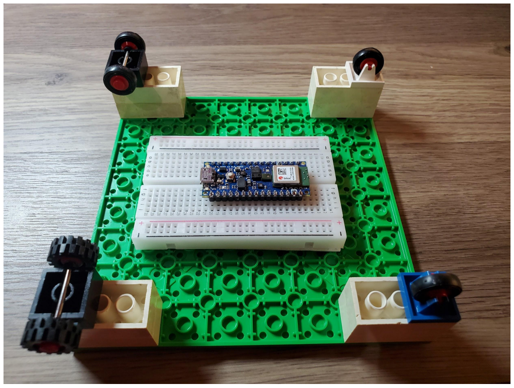

Figure 2: Arduino board attached to data collection device.

I used Legos because they are available at school and many students have them at home in case there is virtual learning in the future. I started by selecting a variety of legos that were 4 different colors. I placed a row of one color of Lego in different orientations in case surface variations might affect the color measurement. I collected approximately 20 seconds of data for each color which was between 40 and 50 data points. After creating scatter plots and confusion matrices for the Lego data I realized it did not give the variability necessary to create any potential bias. Then I switched to a variety of color palettes on my smartphone and placed the phone under the color sensor while still collecting 40 to 50 data points. Again after analyzing the data, the colors were too uniform although there was a greater variety of colors. The last item I selected was yarn because it matched the other colors I tested and it provided more texture compared to other items that I tested. There was greater variation in texture which resulted in greater variability in data, although it did not yield the noticeable bias that I was looking for.

As an aside, it was also noticed that when there was not an item in front of the color sensor, there was a default state that would in effect bias the sensor by giving additional readings of one color even when there was nothing in front of the color sensor. I attempted to use the proximity sensor on the Nano BLE 33 Sense board to determine if an item was in front of the color sensor but the sensor is extremely non-linear at distances where the color sensor is able to accurately measure colors.

Then I switched my attention to examining how unequal or non-normalized data affect machine learning. I first decreased the number of samples for one of the 4 data sets to 10 and then 5 for one of the colors by deleting part of the csv data file for that color. After testing different algorithms I realized that any bias effect of unequal sets for this data set could be compensated by changing the machine learning algorithm or varying parameters in individual algorithms. I was also curious how non-normalized data would affect machine learning so I skipped that step only to realize that it did not affect the decision boundary as much as expected. My hypothesis is that the raw data values for each color reading were in approximately the same range which acted as a de facto normalization of the data.

Then I looked at other ways that were related to different parameters in the machine learning algorithms that might affect how colors are classified. In the support vector machine algorithm, the C and gamma parameters were varied to study their effect on training and test scores to see what values were more likely to misclassify the four colors. It was also determined that it was essential to set the kernel as the radial basis function because the initial setting for the kernel as linear yielded poor results for classification. The training and test scores were also graphed versus the training split to see the optimum values that would create bias and which values would not.

It was ultimately determined that the most efficient means to decrease the margin in the support vector machine algorithm would be to artificially add an offset to different normalized color values before machine learning to in effect create "bias" in the test data. This was also pursuant to the secondary goal where students could see the margin decrease in a scatter plot and correlate this information to the confusion matrix.

IV. IMPLEMENTATION

Hardware and Software Setup: I use the APDS996 Color Sensor on the Arduino BLE 33 Sense Board to collect and record data via a serial port data logger. The serial port data logger was written by Michael Stanley and runs completely in the Edge or Opera browser. The data was recorded at 2 Hertz with a sample number appended to the beginning of each row and a total number of samples appended to the end of each row. The color channels recorded were total counts for the red, blue, green, and clear channels out of a total of 255 counts. The data was saved as a comma separated value file labeled with the name of the color that was sampled.

Sample Number	Red Counts	Blue Counts	Green Counts	Clear Counts	Total Number of Samples
1	15	16	10	41	40
2	14	15	9	38	41

Figure 3 : Sample Color Data. The actual data was saved as a csv.

V. RESULTS

Although multiple tests were conducted with a variety of data, only selected samples are present here for brevity. The initial tests where the sample size was varied or the data was not normalized did not yield noticeable bias when looking at the confusion matrix or scatter plot. To me, this indicated that machine algorithms can be very robust despite best efforts to corrupt them, if the correct algorithm and parameters are chosen.

Then there were attempts to affect the training or test scores by changing the training split which is indicated in the next graph. The support vector machine algorithm was used with the radial basis function as the kernel and C set at 100 and gamma at .25. These values showed the highest values of training and test scores and indicated the importance of selecting C and gamma wisely.

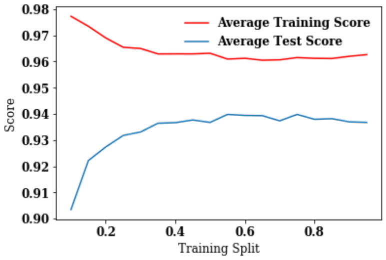

Figure 4: Training and test score versus training split (kernel = rbf, C = 100, gamma = .25)

An example that could lead to low training and test scores, depending on the training split are when kernel is the radial basis function, C is 1, and gamma is set to auto.

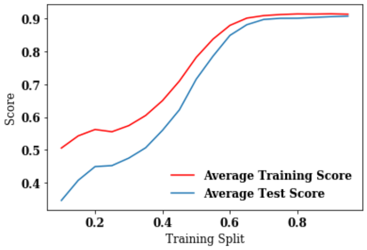

Figure 5: Training and test score versus training split (kernel = rbf, C = 1, gamma = auto)

Or to take a different perspective of low training and test scores, if the C value is constant at .1 but then gamma is varied.

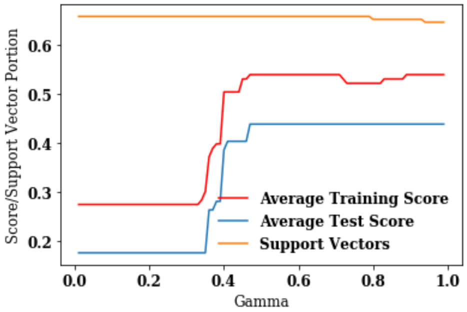

Figure 6: Training and test score versus training split (kernel = rbf, C = .1)

But the ultimate goal was to modify specific channels of the color sensor data in an attempt to bias the data by creating an offset from the original data. I did this manually for a couple of channels by adding .25 to the normalized values of the red and green channels, but neglected to limit the maximum value to 1.

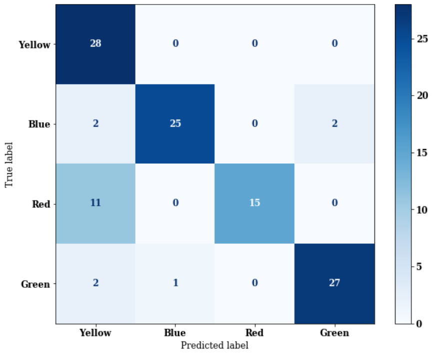

Figure 7: Confusion matrix for the training set with .25 added to the red and green channels)

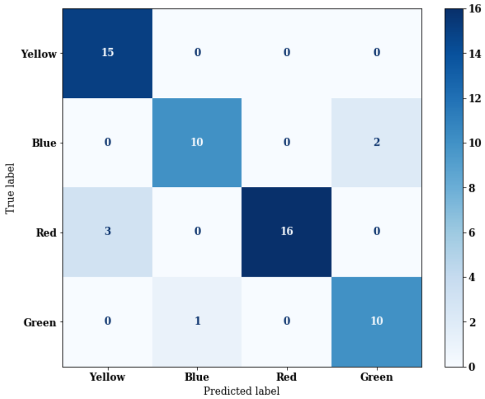

Figure 8: Confusion matrix for the training set with .25 added to the red and green channels)

After doing this manually, it was realized that it would be more efficient to do this via software, so with the assistance of Michael Stanley's code I added offsets to each channel, although at the time these graphs were created, the maximum value was not limited to 1 and the minimum was not limited to 0.

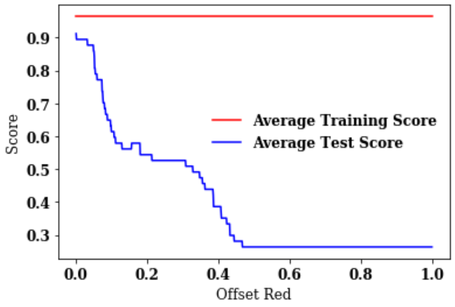

Figure 9: Offset to the red channel

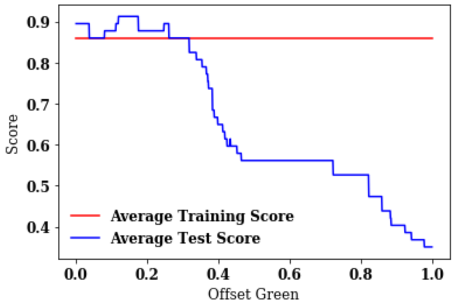

Figure 9: Offset to the green channel

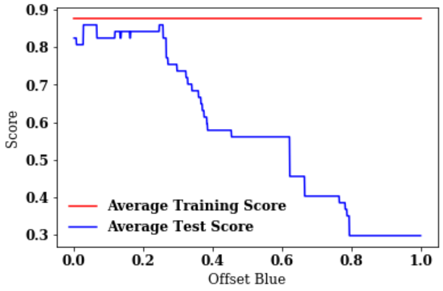

Figure 10: Offset to the blue channel

At the time of submission, work was still being completed on creating scatterplots of the normalized color data. I was able to write csv files for all of the data with an offset added to the blue channel, but even though I was able to read the data from a file, the data did not appear to be graphed correctly on a scatter plot.

VI. DISCUSSION

Limitations: The work described is limited to the Nano BLE 33 Sense board.

Future Work: For the lesson plan that I envision, I want students to be able to select any csv file and then make a scatter plot of that data with the correct color classifications. This is the remaining work to be completed for this lesson plan. Future steps for this project could be examining in more detail how an offset applied to multiple channels of the color sensor can affect data acquisition and analysis.

ACKNOWLEDGMENT

This work is funded in part by SenSIP center and the NSF NCSS I/UCRC.

All work was completed thanks to the assistance of Michael Stanley.

REFERENCES

[1] Suresh, H., & Guttag, J. (2019). A Framework for Understanding Unintended Consequences of Machine Learning. *ArXiv, abs/1901.10002*.

[2] Mehrabi, Ninareh & Morstatter, Fred & Saxena, Nripsuta & Lerman, Kristina & Galstyan, Aram. (2019). A Survey on Bias and Fairness in Machine Learning.

[3] Olteanu, A., Castillo, C., Diaz, F., & Kıcıman, E. (2019). Social Data: Biases, Methodological Pitfalls, and Ethical Boundaries. *Frontiers in big data*, *2*, 13.

[4] Northcutt, C. G., Athalye, A., & Mueller, J. (2021). Pervasive label errors in test sets destabilize machine learning benchmarks. *arXiv preprint arXiv:2103.14749*.

[5] Corbett-Davies, S., & Goel, S. (2018). The Measure and Mismeasure of Fairness: A Critical Review of Fair Machine Learning. *ArXiv, abs/1808.00023*.

[6] Torralba, Antonio & Efros, Alexei. (2011). Unbiased look at dataset bias. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 1521 - 1528. 10.1109/CVPR.2011.5995347.

[7] Afifi, M., & Brown, M. S. (2019). What else can fool deep learning? Addressing color constancy errors on deep neural network performance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 243-252).

[8] Liu, G., Mao, S., & Kim, J. H. (2019). A Mature-Tomato Detection Algorithm Using Machine Learning and Color Analysis. *Sensors (Basel, Switzerland)*, *19*(9), 2023. https://doi.org/10.3390/s19092023

[9] W. Phillips, M. Shah and N. Da Vitoria Lobo, "Flame recognition in video," *Proceedings Fifth IEEE Workshop on Applications of Computer Vision*, 2000, pp. 224-229, doi: 10.1109/WACV.2000.895426.

[10] Oestreich, J., Tolley, W., & Rice, D.A. (1995). The development of a color sensor system to measure mineral compositions. *Minerals Engineering, 8*, 31-39.

[11] *Use SEFR (ML) on Arduino Nano for Color Recognition*. Arduino Project Hub. (2020, October 5). https://create.arduino.cc/projecthub/alankrantas/use-sefr-ml-on-arduino-nano-for-color-recognition-b59e24.