A New Model of Artificial Intelligence: Application to Data I

Charles Davi

February 9, 2019

Abstract

In this article, I'm going to apply the new, polynomial time model of artificial intelligence that I've developed to four well-known datasets from the UCI Machine Learning Repository. For each of the four classification problems, the categorizations and predictions generated by the algorithms were generated on an unsupervised basis. Over the four classification problems, the categorization algorithm had an average success rate of 92.833%, where success is measured by the percentage of categories that are consistent with the hidden classification data. Over the four classification problems, the prediction algorithm had an average success rate of 93.497%, where success is measured by the percentage of predictions that are consistent with the hidden classification data. All of the code necessary to run these algorithms, and apply them to the training data, is available on my researchgate homepage.[1]

1 Introduction

In a previous working paper,[2] I introduced a new model of artificial intelligence rooted in information theory that can solve high-dimensional, machine learning problems in polynomial time by making use of data compression and vectorized processes. Specifically, I introduced an image feature recognition algorithm, a categorization algorithm, and a prediction algorithm, each of which has a low-degree polynomial run time, allowing a wide class of problems in artificial

[2]*A New Model of Artificial Intelligence*.

intelligence to be solved quickly and accurately on an ordinary consumer device. In this article, I'm going to apply the categorization algorithm and prediction algorithm to four well-known datasets from the UCI Machine Learning Repository. In a second article, I will apply the full set of three algorithms to image classification problems.

2 Unsupervised Data Classification

All of the classification problems that we'll analyze in this section will be solved on an unsupervised basis, with the classification labels hidden from the algorithms. This is accomplished by simply moving the classification labels to the $N+1$ entry of each vector, where N is the dimension of the dataset.[3] For each of the classification problems, we'll begin by categorizing the relevant dataset, and then measuring the performance of the categorization algorithm by analyzing the categories it generates. Specifically, we'll measure how well the categories generated correspond to the hidden classification labels. Then, we'll generate predictions using each of the datasets, and measure the performance of the prediction algorithm by counting the number of correct classification predictions.

All of the training datasets we'll analyze in this article are courtesy of the UCI Machine Learning Repository.[4]

The Iris Dataset[5]

We'll begin with the well-known "Iris" dataset, which consists of 150 data points. Each data point consists of 4 values, which are intended to provide information regarding the specific type of flower the data point represents. There is a hidden fifth value which contains the label of the actual class of flower the data point represents. There are three classes of flowers, represented by the numbers 1, 2, and 3, respectively, and as noted above, the categorization algorithm is blind to this information.

We'll begin by having the categorization algorithm take the 150 data points, and construct a set of categories. In this case, the categorization algorithm generated 40 categories. Of those 40 categories, only 3 categories contain data points from more than one class of data, which can be seen in the center of Figure 1 below. As a result, $\frac{37}{40} = 92.500\%$ of the categories generated are consistent with the hidden classification data. The three categories that contained mixed classes of data points contained data from categories 2 and 3.

[3] The algorithms ignore any data above the dimension of the dataset.

[4] Note that I've made some formatting changes to the datasets so that they can work with the algorithms. The code necessary to format the datasets appropriately is available on my researchgate homepage, at the link provided above.

[5] https://archive.ics.uci.edu/ml/datasets/Iris.

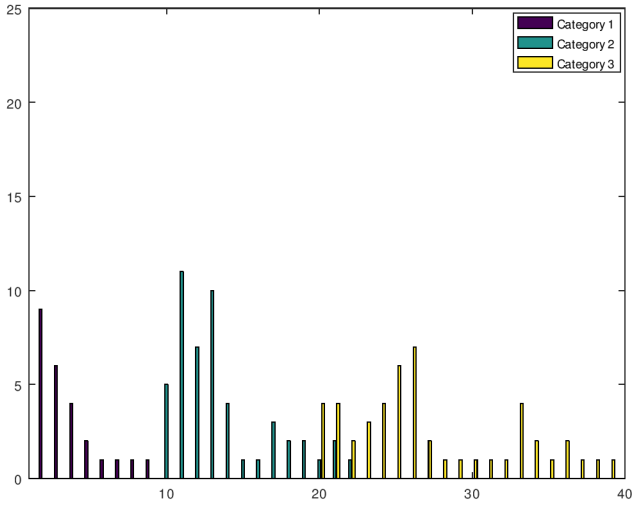

Figure 1: The number of data points in each category for the Iris dataset, with the bars colored according to the hidden classification labels.

In order to generate predictions, we'll rerun the categorization algorithm on a randomly selected subset of the original dataset, using the remaining data points as inputs to the prediction algorithm. This will allow the prediction algorithm to make non-trivial predictions using new data that was not used to generate the categories from which it will make predictions.

The prediction algorithm is itself capable of running in two overall modes: a "rejection on" mode, where it rejects new data that is beyond the scope of its prior data, causing it to fail to make a prediction; and "rejection off" mode, where it always makes a prediction. We'll begin by running the prediction algorithm in "rejection on" mode, which means there are three possible outcomes for each prediction: success (i.e., a correct classification); rejection (i.e., the new data is outside of the scope of the prior data); and failure (i.e., an incorrect classification).

I ran a single round of predictions, and this produced 13 successful classifications, 1 rejection, and 1 failure. If we rerun the same predictions in "rejection off" mode, this generates 14 successful classifications, and 1 failure. This run of predictions demonstrates that the rejection of a new data point does not imply that the new data point would have resulted in an incorrect prediction, but rather, that the new data point is simply not a sufficiently good fit for the prior data for the prediction algorithm to make a confident prediction.

The results of 25 rounds of predictions in "rejection on" mode, for a total

of 375 predictions, consisted of 308 successful classifications, 53 rejections, and 14 fails. This implies an accuracy of either $\frac{308}{375} = 82.133\%$ or $\frac{308}{322} = 95.652\%$, depending upon whether you do, or do not, include the rejections in the denominator, respectively.

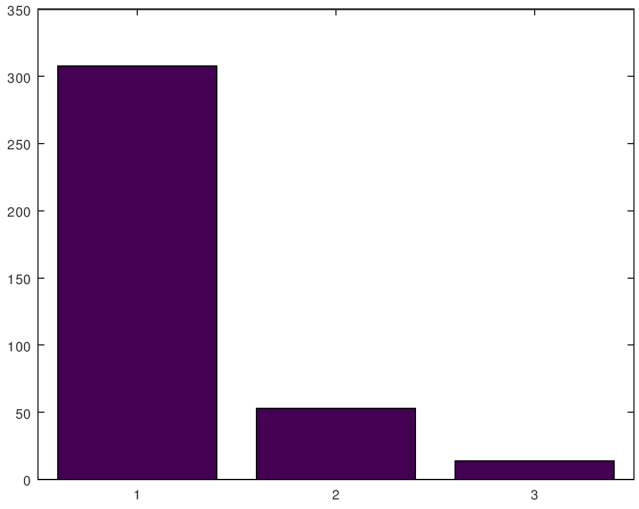

Figure 2: The number of successes, rejections, and fails for 375 classification predictions using the Iris dataset, with rejections turned on.

The results of another 25 rounds of predictions in "rejection off' mode, for another 375 predictions, consisted of 357 successful classifications and 18 fails. This implies an accuracy of $\frac{357}{375} = 95.200\%$.

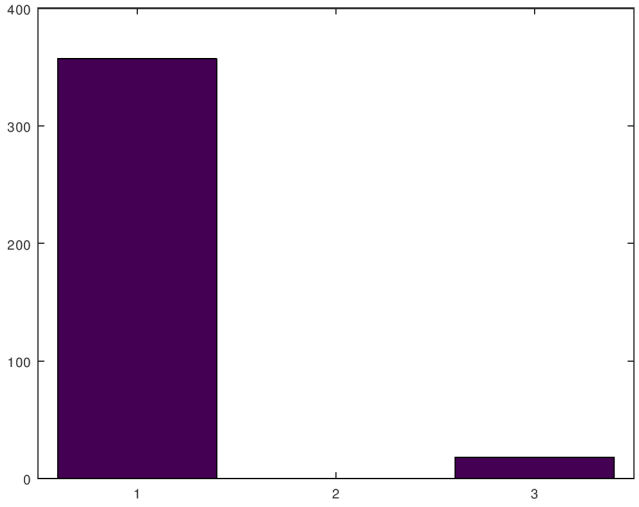

Figure 3: The number of successes, rejections, and fails for 375 classification predictions using the Iris dataset, with rejections turned off.

The Ionosphere Dataset[6]

This dataset consists of 351 data points, each with 34 dimensions of data, together with a classifier hidden in the 35-th entry of each data point that marks the data point as either "good", represented by a "g" in the original dataset, or "bad", represented by a "b" in the original dataset.[7] The classification task is to identify which data points are good and which are bad based upon the 34 dimensions of each data point. Again, as above, we begin by having the categorization algorithm construct categories using the data, blind to the classifier of each data point. In this case, the categorization algorithm generated 141 categories, of which 96.454% consisted of a single class of data.

[6] https://archive.ics.uci.edu/ml/datasets/Ionosphere.

[7] Prior to running the algorithms, I edited the original dataset, replacing each g with a 1, and each b with a 2.

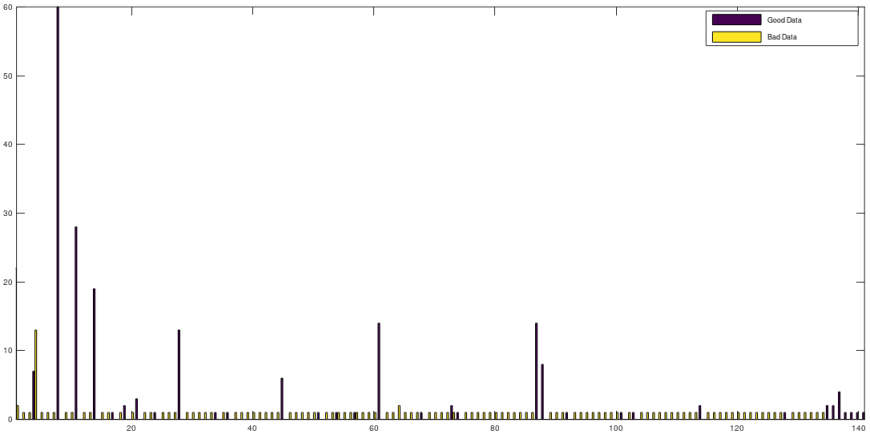

Figure 4: The number of data points in each category for the Ionosphere dataset, with the bars colored according to the hidden classification labels.

We begin by making predictions in "rejection on" mode, again by first re-running the categorization algorithm on a randomly generated subset of the dataset, and using the remaining data points as inputs to the prediction algorithm. The first run of 35 predictions consisted of 19 successful classifications, 14 rejections, and 2 failed classifications. Running the same 35 predictions in "rejection off" mode produced 28 successful classifications, and 7 failed classifications. In this case, a significant number of the rejected data points turned out to generate failed classifications.

The results of 25 rounds of predictions in "rejection on" mode, for a total of 875 predictions, consisted of 462 correct classifications, 383 rejections, and 30 failed classifications. This implies an accuracy of either $\frac{462}{875} = 52.800\%$ or $\frac{462}{492} = 93.902\%$, depending upon whether you do, or do not, include the rejections in the denominator, respectively.

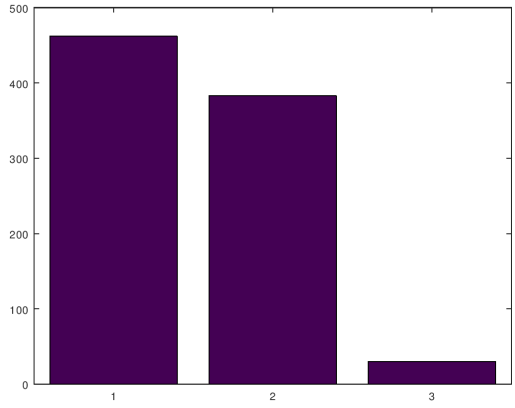

Figure 5: The number of successes, rejections, and fails for 875 classification predictions using the Ionosphere dataset, with rejections turned on.

The results of 25 rounds of predictions in "rejection off" mode, for a total of 875 predictions, consisted of 771 correct classifications, and 104 failed classifications. This implies an accuracy of $\frac{771}{875} = 88.114\%$. In this example, the ability to reject data significantly improved the accuracy of the predictions.

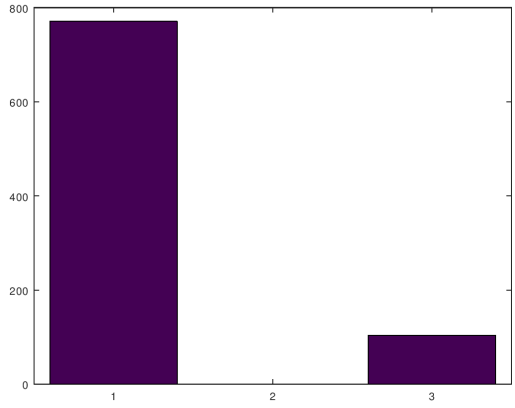

Figure 6: The number of successes, rejections, and fails for 875 classification predictions using the Ionosphere dataset, with rejections turned off.

The Parkinson's Dataset[8]

[8]https://archive.ics.uci.edu/ml/datasets/Parkinsons.

This dataset consists of 195 data points, each with 22 dimensions of data, together with a classifier hidden in the 23-rd entry of each data point that marks the data point as corresponding to a healthy individual, represented by a 0, or an individual with Parkinson's disease, represented by a 1.[9] The classification task is to identify which data points correspond to individuals with Parkinson's disease. We begin by having the categorization algorithm construct categories using the data, blind to the classifier of each data point. In this case, the categorization algorithm generated 98 categories, of which 91.837% consisted of a single class of data.

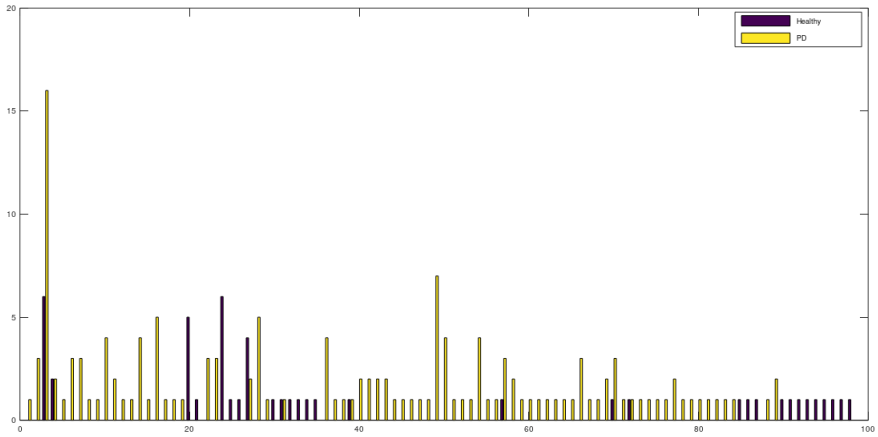

Figure 7: The number of data points in each category for the Parkinson's dataset, with the bars colored according to the hidden classification labels.

We begin by making predictions in "rejection on" mode, again by first rerunning the categorization algorithm on a randomly generated subset of the dataset, and using the remaining data points as inputs to the prediction algorithm. The first run of 19 predictions consisted of 15 successful classifications, 4 rejections, and 0 failed classifications. Running the same 19 predictions in "rejection off" mode produced 18 successful classifications, and 1 failed classification.

The results of 25 rounds of predictions in "rejection on" mode, for a total of 475 predictions, consisted of 285 correct classifications, 158 rejections, and 32 failed classifications. This implies an accuracy of either $\frac{285}{475} = 60.000\%$ or $\frac{285}{317} = 89.905\%$, depending upon whether you do, or do not, include the rejections in the denominator, respectively.

[9] Although the original dataset contains 23 dimensions, the first column of the dataset is the patient's name, which I've removed. Also, after removing the column headers in the top row of the dataset, there are 195 rows of data remaining. As a result, I believe that the UCI website erroneously reports that the dataset contains 197 rows of data.

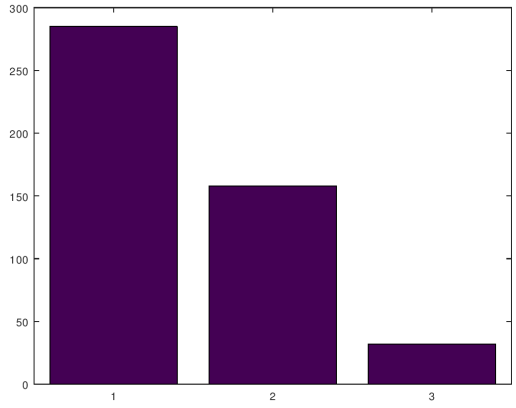

Figure 8: The number of successes, rejections, and fails for 475 classification predictions using the Parkinson's dataset, with rejections turned on.

The results of 25 rounds of predictions in "rejection off" mode, for a total of 475 predictions, consisted of 390 correct classifications, and 85 failed classifications. This implies an accuracy of $\frac{390}{475} = 82.105\%$.

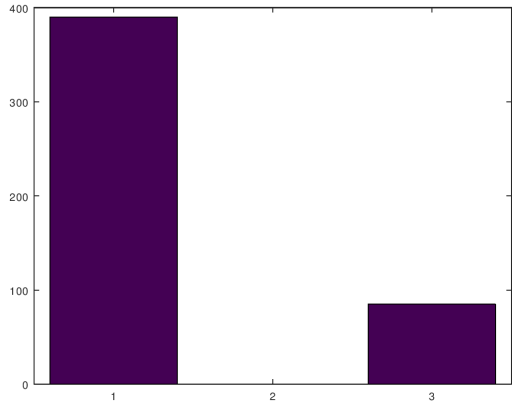

Figure 9: The number of successes, rejections, and fails for 475 classification predictions using the Parkinson's dataset, with rejections turned off.

The Wine Dataset[10]

This dataset consists of 178 data points, each with 13 dimensions of data,

[10]https://archive.ics.uci.edu/ml/datasets/Wine.

together with a classifier hidden in the 14-th entry of each data point that indicates the class of wine that the data point represents. There are three classes of wines, represented by the numbers 1, 2, and 3, respectively. The classification task is to identify the class of the wine given the first 13 dimensions of the data. We begin by having the categorization algorithm construct categories using the data, blind to the classifier of each data point. In this case, the categorization algorithm generated 74 categories, of which 90.541% consisted of a single class of data.

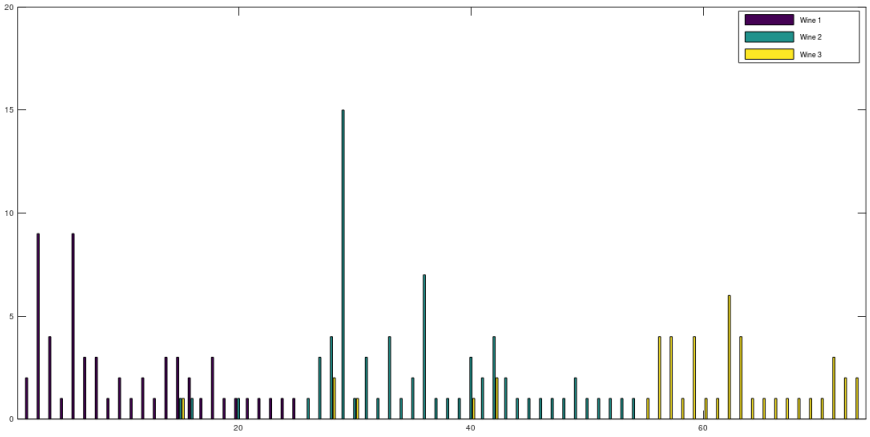

Figure 10: The number of data points in each category for the Wine dataset, with the bars colored according to the hidden classification labels.

We begin by making predictions in "rejection on" mode, again by first rerunning the categorization algorithm on a randomly generated subset of the dataset, and using the remaining data points as inputs to the prediction algorithm. The first run of 17 predictions consisted of 12 successful classifications, 4 rejections, and 1 failed classifications. Running the same 17 predictions in "rejection off" mode produced 16 successful classifications, and 1 failed classifications.

The results of 25 rounds of predictions in "rejection on" mode, for a total of 425 predictions, consisted of 311 correct classifications, 96 rejections, and 18 failed classifications. This implies an accuracy of either $\frac{311}{425} = 73.176\%$ or $\frac{311}{329} = 94.529\%$, depending upon whether you do, or do not, include the rejections in the denominator, respectively.

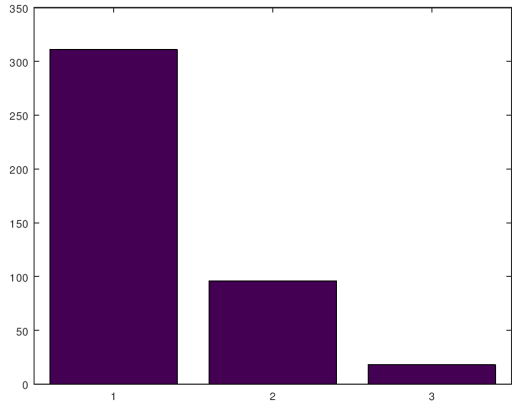

Figure 11: The number of successes, rejections, and fails for 425 classification predictions using the Wine dataset, with rejections turned on.

The results of 25 rounds of predictions in "rejection off" mode, for a total of 425 predictions, consisted of 387 correct classifications, and 38 failed classifications. This implies an accuracy of $\frac{387}{425} = 91.059\%$.

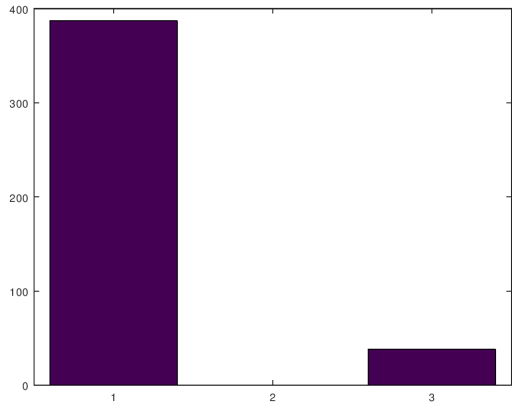

Figure 12: The number of successes, rejections, and fails for 425 classification predictions using the Wine dataset, with rejections turned off.