Draft version June 29, 2020
Typeset using LaTeX default style in AASTeX62

Refining Machine Learning for Finding Optical Counterparts to Gravitational Wave Events

Steven Zhou-Wright[1]

[1] *The University of Arizona, Tucson, Arizona, USA*

(Received March 19, 2020; Revised April 16, 2020)

ABSTRACT

The **S**earches **A**fter **G**ravitational-waves using **AR**izona **O**bservatories (SAGUARO) uses the Catalina Sky Survey and other telescopes to search for optical counterparts to gravitational wave events. The survey produces thousands of images each day, and each transient event detected on camera requires a human to verify it. To make the workload tenable and improve the speed at which transients and GW-related transients are found, the survey employs machine learning to pre-sort the incoming images to only show the most promising candidates. By creating and curating a data set and retraining the machine learning classifier on this data set, we reduced the rate of false positive classifications and thereby reduced the workload needed to manually sort each nights observations. The new classifier reduces the number of images a human has to look at per year by tens of thousands.

1. STATEMENT OF PURPOSE

The SAGUARO project receives hundreds of images each night which are used in the search for transient astronomical events that could correspond to gravitational events detected by the LIGO/VIRGO surveys. To ease the human workload in identifying real transients from noise, camera artifacts, or other false positives, a machine learning algorithm automatically assigns a score to each image taken in a night. The higher this score, the higher the chance of the image containing a real transient.

However, the machine learning algorithm was originally trained on a small data set with minimal oversight over what constituted a real event. Essentially, only events that were verified by other surveys were counted as real. This can lead to issues if the survey's image of the event is otherwise poor- even though the event is real, the machine learning algorithm will learn the wrong lesson and begin classifying bad images as real. This combined with the small size of the training set led to an inflation of scores, and far too many obviously false positive classifications.

My task was to develop an improved machine learning algorithm that was more selective in assigning higher machine learning scores while still positively identifying all real transients. It is essential to identify these events as quickly as possible, as they are not very long lasting and we want to map as much of the light curve as possible and perform spectroscopy before the event becomes too dim. By reducing the amount of bad images humans have to sift through, the hope is to ultimately decrease the time spent looking confirming transients.

2. BACKGROUND

First, we should speak briefly on the importance of gravitational waves and their optical counterparts. Beyond being an important verification of Einstein's theory of general relativity, gravitational waves can also lend insights into several important and rare physical processes. As gravitational waves propagate, they can move through mass without being noticeably affected (Abbott et al. 2009), an advantage they have over light waves, which can be absorbed partially or blocked outright. Furthermore, the events that give off gravitational waves are incredibly rare. These events are mergers of compact objects such as black holes or neutron stars. For one of these events to occur, two of these rare objects need to form, then they need to either approach either each other or form in the same area, and then they need to actually merge while we happen to be watching. The probability of this happening anywhere near us is incredibly low, which necessitates looking further away. This is why the first confirmed observation of a black hole merger was a GW observation (Abbott et al. 2016). The study of gravitational waves can tell astronomers about these compact objects by giving constraints on formation rate, age, mass, and so on.

However, finding optical counterparts to compact object mergers is still important. Spectroscopic observations of these kilonovae (the EM counterpart to gravitational wave events) can provide crucial insight into heavy element

formation and emission mechanisms (Lundquist et al. 2019), and can also allow mergers to serve as a standard candle or standard siren in the field of high-precision cosmology (Doctor 2020).

3. METHODS

To understand the methods used to improve the machine learning, some discussion of the machine learning algorithms is necessary. The algorithm employed was a self-teaching random forest classifier. A random forest classifier works by creating a series of "decision trees" (see the example below) using randomly picked features and sub samples of the data and classifying the data based on the individual determinations of each of those trees. Then each of those individual classifications are in some way averaged to arrive at a final classification. Because each of the trees use only a subset of the data (in a process called "bagging") and are highly independent from each other, the machine learning algorithm will both have high accuracy and avoid "overfitting", or when the classifier becomes overly specific to the data set it was given to train on (Breiman 1996).

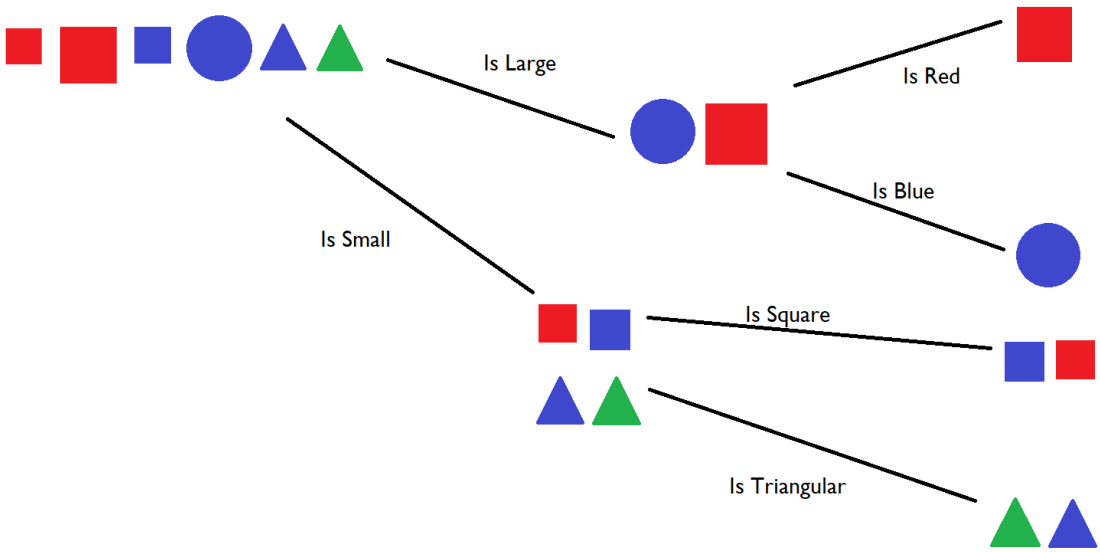

Figure 1. An example of an incredibly simple decision tree, which forms the basis of the random forest's classification scheme.

Critically, we also note that in this case the random forest acts as a "black box" in that we don't know in detail what decisions the algorithm makes on the path to its final determination. The machine learning's exact process for classification is too complex for us to understand the specifics, such as specific features used in the trees. To verify that the machine learning is working, we simply feed it a data set with known classifications, and then train several random forest algorithms with varying parameters in how the algorithm is made, and then we choose whichever combination of parameters returns the algorithm that best matches the classifications we fed in at the start.

With this in mind, one of the few ways we can actually improve the machine learning is by feeding it the best possible data set. Thus the first step of the project was producing and curating this data set. To do this, we ran the older machine learning algorithm on over 30000 images displaying objects found by the survey, and assorted them into "good" and "bad" bins. It should be noted that "good" here simply means that the image is good quality, but **not** necessarily displaying a real transient. By the same token, "bad" means the image could potentially have had a transient in it, but the other qualities of the image should not be associated with a positive identification by our machine learning.

I was a reviewed and sorted around 20000 of the images in the training data set by the following criteria:

- contrast (good images/real transients will be high contrast)

- roundness (real transients will be approximately round)
- centered-ness (a good image has the object in the center)
- good image subtraction (bad image subtraction is signified by a ring of bright pixels around the dark center)
- size (a good image contains an object about 3 pixels wide)
- singular-ness (a good image contains only one object)
- normalcy (if it is weird, it is not a good image)

With this pre-classification done, the next step was to go through each bin and pick out the images that don't belong, i.e. go through the good bin and remove any bad images. There is no way to fully automate this process, and humans were the primary determiners of what was good and bad. Once this data set is complete, in theory the newly curated data set is ready to train the new machine learning classifier.

However, the data set can be expanded and improved. By applying a series of a transformations to the data, I can effectively increase the size of the set and teach it to account for certain features in its classification. We elected for two methods of data transformation that we believed would also be relevant to the images we would get in the future: rotation and Gaussian noise addition. Rotating the images hopefully prevents the algorithm from accidentally preferring a certain orientation in the images. The inclusion of Gaussian noise is accomplished by creating a random bright/dark overlay on the image pixels- basically adding TV static to the image- and can simulate a number of non-ideal observing conditions. This will hopefully make the algorithm better equipped to deal with the noise that will appear in the actual observations.

Note that these modifications don't change the classification of the image, so there is no need to manually re-classify them as was done at the outset of the project. By developing the appropriate python script to pull the data from an SQL database, transform the images, and re-upload them, we can double (or triple, or quadruple, and so on, depending on how many different transformations are saved to the database) the size of the training set in the time it takes to run the script we wrote. This also allows me to expand the size of the class of images that are classified as real. After the data set was initially reviewed by humans, the real image only numbered 3000- this small of a sample in a training set may hinder any classifier made by the machine learning. Thus, the data modification step is expected to improve the efficacy of the classifier overall in addition to immunizing it too specific biases.

We can also implement some methods to improve the testing of our machine learning algorithm. Specifically, we employed k-folding cross-validation for testing. Essentially, the data set is sorted in k random subsets, and all but one of those subsets called k-folds is used to create the classifier. Then the classifier is tested using that last subset. This process then repeats until each k-fold has been used once for testing. The benefit of this method is that we can test the classifier on data that was **not** used to teach the classifier. This in turn makes sure that the classifier isn't good only for the data we used to create it, and will actually work when we start using it on incoming data that doesn't have a predetermined classification.

We also need some method to validate the new classifier by answering the question: "Is this classifier better?" That question can be broken down further into two questions:

- Does this classifier deflate the scores assigned to the incoming images?
- Does this classifier correctly identify transients, given that the image is otherwise good?

I was tasked with developing a method of using the classifier's output (recall that the random forest is a black box solution in this application) to determine if the new classifier is better than the old. The method I suggested was a histogram of the machine learning scores. By running each classifier,counting how many images score in each range, and comparing them quickly, I can make a python script to compare them. If the new classifier has a lower count of scores in the higher range (particularly scores greater than 0.5) then the classifier is more selective.

I also used python to make a scatter plot of each image's score, with the new score on one axis and the old score on the other. Then we split the plot into 4 sections, one where both classifiers thought the image was real, one where neither thought it was real, and a section each where one thought it was real and the other thought it was fake. By examining the section that contains images the old classifier said was real but the new classifier said was fake (and vice versa) we can conclude whether the new classifier is more selective.

Figure 2. An example of images we considered "good" and included in the real data set.

By consulting the Transient Name Server, the images that correspond to real transient events can be picked out automatically and highlighted on the scatter plot above. Ideally, the classifier should consider these events real, although again a bad image containing a transient may be correctly excluded.

4. RESULTS & DISCUSSION

With these results, we can begin the process of verifying the new classifier's effectiveness. To do this, we simply run the older classifier on as much data as is possible or useful, and compare the score of each image with both classifiers.

Figure 3. An example of images we considered "bad" and included in the fake data set.

I then scatter plotted each image based on its scores as explained above. The histogram mentioned is overplotted on each axis, displaying the frequency of each score (in bins of 0.05) for each classifier. The results are shown below:

The high population of the Fake/Real section (especially in comparison to the Real/Fake section) of the graph shows that our classifier certainly achieves its goal of reducing scores overall. This is corroborated by the histogram as well. Further, only one confirmed event (indicated with red stars) was classified as fake, This is not necessarily an indictment of the classifier's efficacy, as it is still possible to have a transient show up in a "bad" image. As it stands, the classifier would accomplish its goals of reducing the human oversight needed to find real transients.

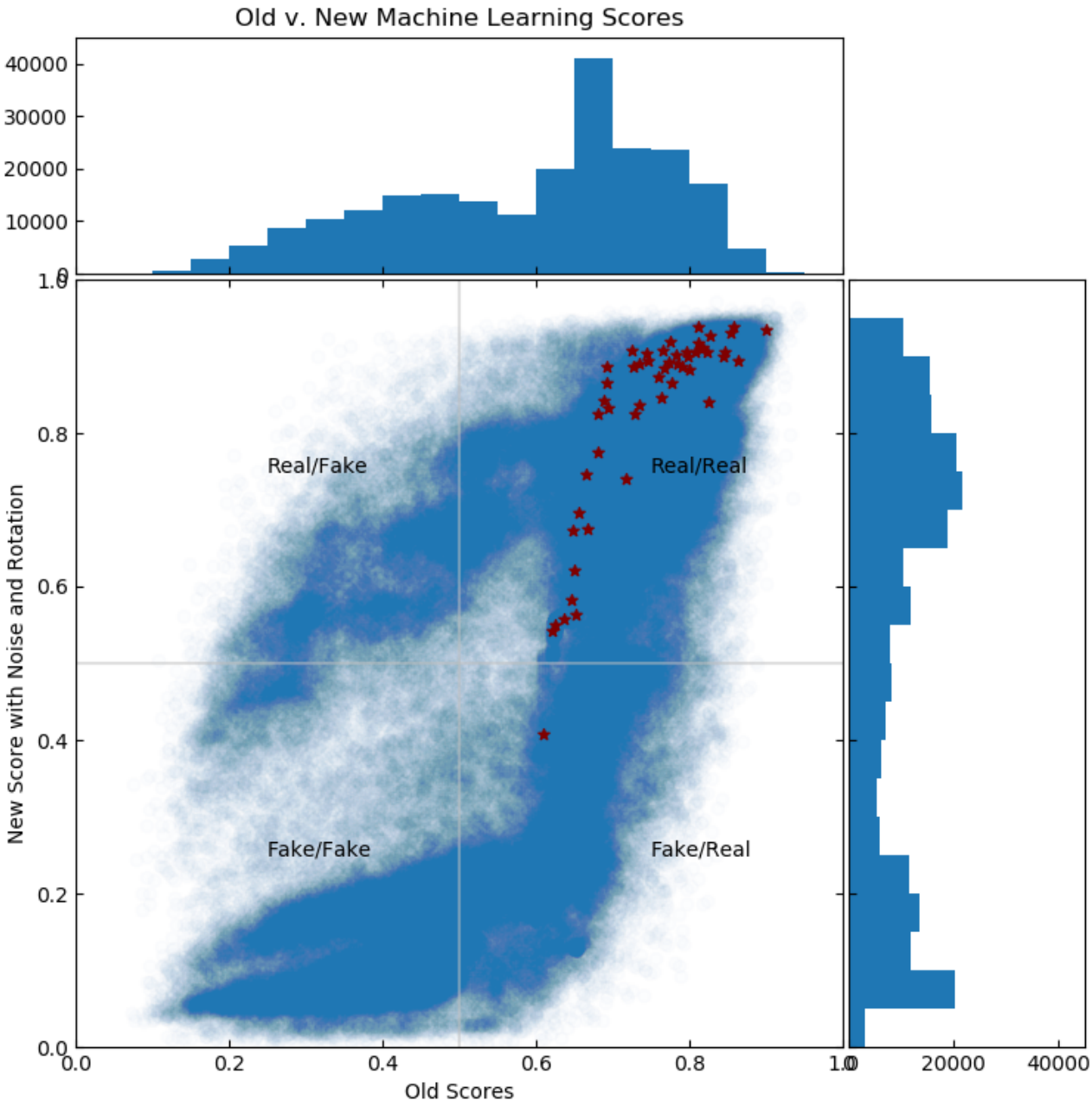

Figure 4. The histogram on top corresponds to the older machine learning score, the one on the right to the newer machine learning score. The red stars correspond to confirmed transients taken from Transient Name Server. The data used to compare scores comprises tens of thousands of images taken by CSS and stored by SAGUARO

Another encouraging sign is the score histogram for each classifier; the newer one has a much more even distribution with far fewer in the

The population of the Real/Fake section could either be bad images that are somehow harder for the newer classifier to properly identify, or they could be images that always should have been classified as real, and were instead improperly discarded by the old algorithm. A critical next step is finding some method of proving one of the above cases correct-though the only method that springs to mind is manually looking at images in that section and seeing if they still look

good to human eyes. Given that the new classifier was trained on a human-made set, it seems appropriate that this would be the case.

5. ACKNOWLEDGEMENTS

Thank you to my research advisor David Sand and Michael Lundquist for the opportunity to work on the SAGUARO project, in addition to the lessons and skills they taught me along the way.

REFERENCES

Abbott, B. P., Abbott, R., Adhikari, R., et al. 2009, Reports on Progress in Physics, 72, 076901

Abbott, B. P., Abbott, R., Abbott, T. D., et al. 2016, PhRvL, 116, 061102

Breiman, L. 1996, Machine learning, 24, 123

Doctor, Z. 2020, ApJL, 892, L16

Lundquist, M. J., Paterson, K., Fong, W., et al. 2019, ApJL, 881, L26