Impact of Skewed Distributions on an Automated Plankton Classifier

Will Chapman, Emal Fatima, William Jenkins, Steven Tien, Shawheen Tosifian
ECE228 Machine Learning for Physical Applications, June 2018, UC San Diego

***Abstract*— Plankton are extremely important to the marine ecosystem and form the basis for the food web and carbon cycle. Recently, large efforts have been dedicated towards classification of plankton species to better understand population dynamics. Machine learning has emerged as a way to classify large amounts of plankton images without dedicating immense amounts of man hours to this laborious task. However, plankton populations tend to be realized in large "blooms" in which one species acts as a dominant signal as compared to other species populations. This is a unique situation for Machine Learning algorithms which are typically trained and tested on evenly distributed plankton populations. This paper seeks to understand the errors of machine assisted plankton classification associated with sudden population blooms. Using an optimally tuned Random Forest algorithm, we find that the dominant species acts to converge on inherent model accuracy, while the non-dominant species spreads largely around this mean in a Gaussian distribution. Additionally, categorical species model biases are explored.**

I. INTRODUCTION

As the most abundant form of life in our oceans, plankton - microscopic organisms which include bacteria, photosynthesizing organisms (phytoplankton), and the tiny organisms that eat them (zooplankton) - are the foundation of life in the sea. Their abundance directly impacts the ecology of the oceans, as their position at the lowest trophic levels of the food chain ensure their population dynamics can be felt by even the largest of apex predators. Additionally, the combined respiratory and photosynthetic activity of plankton are major contributors to our planets atmospheric constituents, serving as a biological pump and regulating both oxygen production and carbon sequestration. Yet for all their importance, much remains to be learned about the ecology and population dynamics of these ubiquitous organisms.

In an effort to get a sense of local plankton population dynamics, in 2014-2015 the Jaffe Lab for Underwater Imaging at Scripps Institution of Oceanography (SIO) developed the Scripps Plankton Camera (SPC, pictured below in Figure 1), which has been successfully imaging tens of thousands of these small organisms each day. Near real-time images are visible at http://spc.ucsd.edu. With millions of images collected, accurate classification of each image is impossible to do by hand. To assist in the classification of these large data sets, machine learning algorithms have been implemented which rely on sets of manually labeled data to train the classifier. Of particular interest in this project was clarifying how the underlying statistics of the plankton population affect the performance of a classifier algorithm, in our case, the random forest.

In nature, a classifier will often encounter situations where the environment of interest is largely dominated by instances of a singular class, for example, during a red tide event in La Jolla. We are interested in seeing how the performance of a pre-trained classifier may be affected by an input with a skewed distribution. To do this, we trained a simple classifier that utilizes random forests and tested how its accuracy is affected by differing distributions.

II. RELATED WORKS

A variety of approaches to plankton classification have been undertaken so far. Dai et al. utilized a hybrid convolutional neural network (CNN) whose inputs included plankton images as well as local and global plankton features obtained through traditional feature extraction methods. The authors designed a pyramid structure in a fully connected layer to amalgamate the results from the three input sources. Using this hybrid structure, an accuracy of 95% was reported for 30 classes of plankton[1]. However, one drawback of this approach was that the classification was done over a homogenous distribution of plankton, when real-time classification would encounter non-uniform distributions of plankton species. Yan et al. proposed a more efficient CNN architecture which achieved a top-five accuracy of 96% with the potential to be applied

in ocean systems thanks to its small size[2], though there was no analysis of non-uniform distributions of plankton species. Tindall et al. used a transfer learning approach with a VGG16 CNN architecture and reported an accuracy of 85% for a 12 class plankton set from Woods Hole Oceanographic Institution[3]. Though the accuracy was not as high as that reported in published CNN literature, its use of transferred weights showed potential for future improvement. Lee et al. applied CNNs which incorporated transfer learning (from class-normalized data and fine-tuning with original data) to imbalanced distributions of plankton, performing better than CNNs with and without transfer learning through data augmentation techniques[4]. Over 70% of the images in the dataset belonged to one class, and 90% belonged to five classes. The classification of the top five classes achieved 95% accuracy, but the rest of the smaller classes stagnated at less than 50% accuracy, and their approach did not dramatically improve rates for non-dominant classes. Orenstein et al. explored classification with RF as well as CNNs. The RF classifier was reported to have an accuracy of 58% for a 95-class dataset and 69% for a four-class dataset (obtained from the Scripps Plankton Camera System)[5]. The RF model was trained on 72 hand-engineered features, the same that were features utilized in this analysis.

III. Dataset and Features

A. Dataset

Images were obtained from the Scripps Plankton Camera System. The raw images were then filtered, cleaned, and segmented such that the image field of each plankton was obtained and all extraneous image information discarded. The images were provided by Eric Orenstein at the Scripps Institute of Oceanography at UCSD[5]. For this project, we utilized approximately 12,000 labeled images from 12 different classes composed of various plankton types. To simulate different orientations for the plankton and increase our image count, we created transformed (rotated/sheared) copies of the images for a total of 24,000 images for 12 classes. The images varied in size from about 100-200 pixels square, and were represented in color.

B. Features

From these processed images, we extracted 72 hand-engineered features to be used as the input vectors for the classifier. Each feature corresponds to a geometric property of the plankton image, such as major axis length, minor axis length, aspect, elliptical properties, etc. These properties were determined algorithmically. This was accomplished by searching for iso-contours in the image, representing them in Fourier space, making them rotation invariant, and then taking the first 10% of the Fourier coefficients and then back projecting to a simplified boundary. Additionally, the other geometric features were calculated using the skimage library′s measure function. A gray level co-occurrence matrix was also utilized in the feature extraction to incorporate properties of the image. By using geometric properties of the plankton, we are able to take into account distinguishable aspects of the data while reducing the dimensionality of the data (versus feeding the entire image into the model).

(a) Acanthera Class (b) Copepod Class (c) Chain Class

Fig. 1: Some sample images from the dataset

IV. Methods

In order to test the impact of skewed distributions on artificially intelligent machine classifiers, we needed a machine trained to classify plankton images into 12 different categories and a method to input a skewed distribution. A skewed distribution being a set of images that intentionally have an abundance of one class over the other.

Our choice of algorithms to train our classifier was through implementation of a Random Forest. The Random Forest method is not the method with the greatest accuracy, but that fits our purpose well as its simplicity is expected to highlight the error, which was impact of interest, caused by skewed distributions.

To generate our skewed distribution, we wanted a large training set and a large testing set. With 12,000 images for 12 classes, there were only 1000 images a class, so we performed some preprocessing to create transformed images from our prelabeled ones such that the images created would be skewed or flipped, which would represent a different orientation these plankton would be in. As a result, we were able to grow our total dataset to 24,000 pictures.

On top of performing image transform on our dataset, we needed to sample it for our testing set in a

way that produced a dominance from one class. This was done by setting aside half of the 24,000 images, selected at random, for testing. With each testing run one category was artificially increased to have 10-fold the number of images of every other category.

A. *Random Forest*

For our classifier, we utilized the Random Forest algorithm. From a high level perspective, the Random Decision Forest can be thought of as a collection of Random Trees. A Random Tree takes the input (in this case an image vector/matrix) and classifies it via a random weighting of its features. The classification decision from each Random Tree is considered a "vote". All the votes are then tallied up and whichever class has the highest votes wins. Figure 2 shows this pictorially.

An optimal Random Forest algorithm is achieved by tuning the hyper parameters (external configurations inherent to the model, such as model depth, number of trees, etc.). This was achieved through a hyper parameter grid search, the model was initialized, trained and tested on a randomized set of hyper parameters. In the process, the data is 10-fold cross validated to prevent over fitting. This operation is performed roughly 500 times and the optimal model is then saved and used for the final model results. The optimal model for our random forest had the following configuration: maximum depth: 70, minimum number of samples in a leaf: 1, minimum number of samples in a split: 2, and the number of estimators: 400.

Figure 3 shows a confusion matrix of the performance of this random forest on a fairly evenly distributed testing samples, with roughly 1000 images per testing category. Accurately predicted species are contained on the diagonal, thus high numbers on the diagonal are desired. The overall model accuracy for every image classified is 86.4 percent. This is on par with the state of the art methods outlined in the previous section. Upon further inspection, it is clear that some species are more readily identified than others. The overall accuracy of the model is thus highly dependent on which categorical species is presented most often. Understanding the distributions of these accuracies motivates our study.

B. *Skewed Distribution - Monte Carlo Analysis*

Utilizing the skewing method mentioned above, a Monte Carlo technique is leveraged to explore the error space. A Monte Carlo method randomly subsamples a space multiple times and determines the accuracy

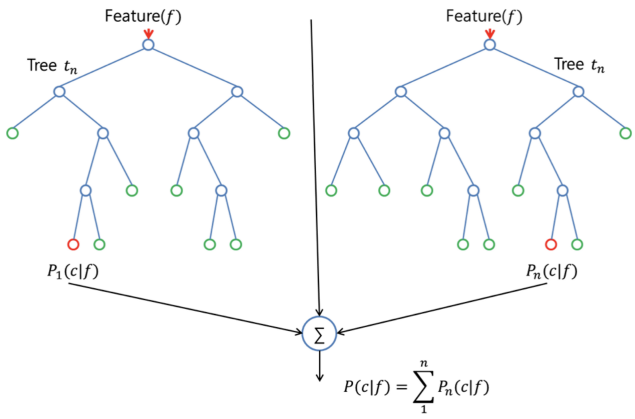


Fig. 2: Random Decision Forest

('Percent of Corect Categorization:', 86.4, '%')

Predicted Species / Actual Species	acanthera	agg	app	ceratium	chaeto	chain	copepod	flk	helix	nauplii	phaeo	small
acanthera	1054	28	16	6	14	0	26	14	2	38	6	18
agg	22	664	32	2	68	0	24	96	36	2	126	4
app	6	34	924	2	10	14	50	18	18	4	28	2
ceratium	4	0	0	940	0	0	34	0	0	2	0	14
chaeto	26	18	0	0	988	0	6	18	24	2	16	0
chain	0	2	0	16	0	1272	2	2	14	0	4	6
copepod	6	0	56	38	6	14	1260	16	4	10	50	2
flk	18	46	34	0	14	18	36	1270	30	0	42	2
helix	2	2	22	8	42	22	6	18	920	0	8	2
nauplii	4	0	0	0	2	0	18	0	0	992	0	60
phaeo	8	74	16	0	18	4	48	90	32	0	1212	0
small	0	0	0	2	0	0	0	2	0	30	0	942

Fig. 3: Confusion Matrix of roughly evenly distribution testing dataset

for each subsampling. This is motivated by the fact that in large population, it is possible that the true error of interest is obscured by other unknown factors about the sample. In more specific terms to our project, from the 12,000 images available to us for testing, we intentionally resampled it such that there is a known abundance of one class, a dominant class, so that we could observe classifier performance under this new sample of images. This was repeated 2000 times for each dominant and non-dominant class. This resulted in a probability distribution of accuracies.

The Monte Carlo was performed for every class and the performance of the classifier was determined. This is so that we could see how effective the classifier is for each class, as an 89 percent accuracy overall may just be due to the fact that the classifier attempts to minimize error by classifying everything into a class

such that it nets the greatest performance overall, even if it forsakes an underrepresented class. In essence, we implemented the Monte Carlo to test if the classifier performs based off any prior knowledge on the probable distribution from the training set. The accuracy for each instance where one class is dominant can be seen in a experiment matrix (Figure 5). Figure 4 shows the results of the twelve Monte Carlo experiments (one for each species tested), with each column representing a separate experimental run. The dominant class for each experiment is highlighted in yellow on the diagonal. Each column representing the percent of accurately categorized species, which is also summarized in the table above it, called the "Large Category Score". Each row demonstrates the performance of an individual species across every experimental run. Figure 4 shows

Raw Score

	acanthera	agg	app	ceratium	chaeto	chain	copepod	flk	helix	nauplii	phaeo	small
0	0.861951	0.761951	0.850244	0.896585	0.876098	0.909756	0.864878	0.85561	0.878049	0.885366	0.846829	0.917561

Large Category Score

	acanthera	agg	app	ceratium	chaeto	chain	copepod	flk	helix	nauplii	phaeo	small
0	0.865263	0.615789	0.831579	0.946316	0.897895	0.962105	0.861053	0.845263	0.877895	0.921053	0.804211	0.965263

	0	1	2	3	4	5	6	7	8	9	10	11
acanthera	0.865263	0.88	0.86	0.9	0.9	0.85	0.88	0.81	0.82	0.87	0.87	0.82
agg	0.68	0.615789	0.61	0.65	0.57	0.64	0.66	0.63	0.65	0.58	0.66	0.73
app	0.79	0.83	0.831579	0.87	0.84	0.8	0.8	0.77	0.85	0.8	0.87	0.88
ceratium	0.96	0.94	0.93	0.946316	0.92	0.95	0.95	0.96	0.95	0.93	0.9	0.97
chaeto	0.88	0.88	0.93	0.85	0.897895	0.96	0.88	0.92	0.94	0.91	0.91	0.88
chain	0.97	0.96	0.95	0.94	0.99	0.962105	0.96	0.97	0.96	0.94	0.97	0.98
copepod	0.84	0.84	0.88	0.83	0.85	0.87	0.861053	0.88	0.88	0.86	0.88	0.85
flk	0.84	0.84	0.87	0.83	0.85	0.83	0.84	0.845263	0.85	0.87	0.87	0.9
helix	0.82	0.92	0.84	0.82	0.88	0.83	0.84	0.89	0.877895	0.88	0.92	0.9
nauplii	0.91	0.96	0.9	0.94	0.86	0.97	0.93	0.91	0.91	0.921053	0.92	0.91
phaeo	0.77	0.74	0.81	0.79	0.78	0.86	0.85	0.81	0.87	0.8	0.804211	0.82
small	0.99	0.98	0.95	0.97	0.99	0.95	0.96	0.96	0.98	0.96	0.95	0.965263

Fig. 4: Dominant class experimental runs, the dominant species in each of the 12 experiments is shown in yellow along the diagonal

the Experimental Matrix generated when each class is sampled to be the dominant one in the testing sample. If we compare this to the raw score normally obtained without a dominant class, it can be seen that the accuracy for most classes does not decrease or deviate much even when dominating the distribution. This suggests that the classifier extracts features that do not seem to depend on the training set's actual distribution. However, there is is still a 5-15 (depending on the dominant class) percent deviation of accuracy that varies depending on the supplied distribution. For the purpose of planktonic research, that may be within the tolerance of researchers who need the data, however the deviation of accuracy is worth understanding for the sake of improving classifier performance.

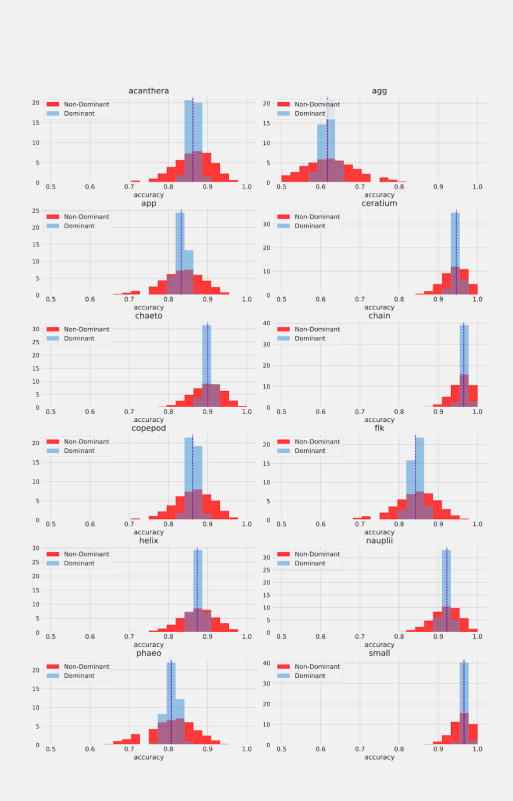
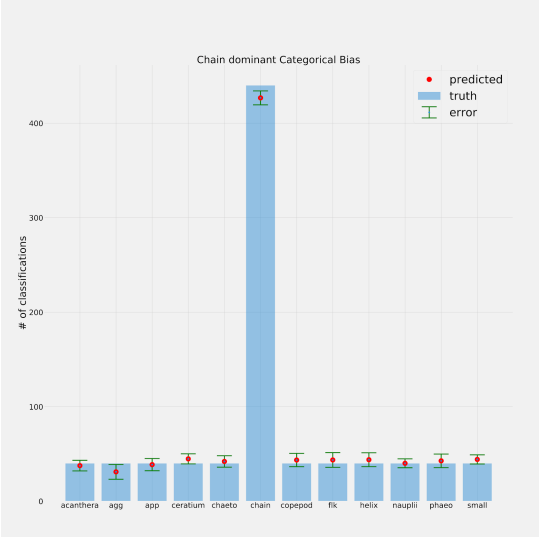


Fig. 5: The PDFs representing the Monte Carlo generated samples, in which the spread of the accuracy for the dominant and non-dominant classes can be observed.

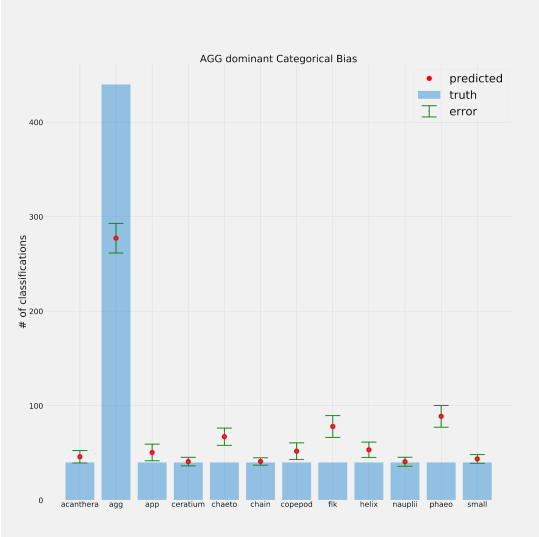
V. Experiment/Results/Discussion

Figure 5 gives a graphical representation of the Monte Carlo Simulations, and a graphical representation of Figure 4. As can be seen, there is no meaningful shift in the means of the distribution. Although, the distribution does widen, in the non-dominant species (shown as the red distribution). The narrowing of the dominant class (shown as the blue distribution) which is a result that can be attributed to the classifier seeing more and more images and converging to an accurate class skill.

The categorical biases depicted in Figure 6 show how the Agg class dominant distribution led to poorer performance of the model, as opposed to the Chain class dominant distribution; other dominant distributions performed similarly to the Chain class. The actual distribution numbers for each experiment are shown in blue bars. The predicted categories and the range of error are determined by the mean and the spread of the Monte Carlo methods described above. This can be attributed to the physical nature of the Agg class,



(a) Chain Class



(b) Agg Class

Fig. 6: Derived Biases from Monte Carlo methods for the Chain class (top performing) dominant distribution and Agg class (worst performing) dominant distribution are shown.

which is composed of dead pieces of various plankton and those particles which cannot be classified as one of the other classes. As a result, the classifier performs poorly as there is a varied distribution within the dominant class, leading to the difficulty of the classifier to correctly identify the class itself along with the other non-dominant classes. The goal of these figures is to determine a signal to noise ratio, of when a species is detected in mass, when can it be trusted that it is in fact a species bloom. In addition, it appears that a general trend in the classifiers bias toward dominant classes is that the bias is generally underspecified, as seen in Figure 3. We found that no class, as the dominant species, is biased high, this is fairly unexpected, but intuitive as 12 separate classes exist. This would not be the intuitive result for a binary classification system.

VI. Conclusion/Future Work

We suggest extending this experiment to another classification model, such as a convolutional neural network. By comparing our results with other models, strengths and weaknesses of the models can be better quantified. Afterwards, the models could be tested against larger and more varied distributions that are more representative of real-time conditions so their performances could be measured and evaluated against other metrics such as computational resources. Eventually, a classifier would be implemented which is capable of accurately identifying plankton images near real time.

References

[1] Dai, J., Yu, Z., Zheng, H., Zheng, B., and Wang, N. (2016, November). A Hybrid Convolutional Neural Network for Plankton Classification. In Asian Conference on Computer Vision (pp. 102-114). Springer, Cham.

[2] Yan, J., Li, X., and Cui, Z. (2017, October). A More Efficient CNN Architecture for Plankton Classification. In CCF Chinese Conference on Computer Vision (pp. 198-208). Springer, Singapore.

[3] Tindall, L., Luong, C., Saad, A. (2017, June). Plankton Classification Using VGG16 Network. ECE 228 Final Project. University of California, San Diego.

[4] Lee, H., Park, M., and Kim, J. (2016, September). Plankton classification on imbalanced large scale database via convolutional neural networks with transfer learning. In Image Processing (ICIP), 2016 IEEE International Conference on(pp. 3713-3717). IEEE.

[5] Orenstein, E. C., and Beijbom, O. (2017, March). Transfer Learning and Deep Feature Extraction for Planktonic Image Data Sets. In Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on (pp. 1082-1088). IEEE.

[6] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.