Identifying Clouds from Satellite Images of Arctic Regions by Statistical Modelling

Team Devil's Heels
Edward Kim 3033170974
Xingzhi Wang 3033625638

Data Collection and Exploration

In the work by Shi et al from 2008, the authors reported a systematic algorithm to automatically detect regions covered by clouds in a satellite image. The data investigated in this study was collected by Multiangle Imaging SpectroRadiometer (MISR) project launched by National Aeronautics and Space Administration (NASA) in 1999. The project, as part of a global effort to monitor and fight climate change, aimed to map the distribution of cloud coverage in polar areas using MISR sensors installed on a Terra satellite. The challenge in this project lies in the fact that in snow or ice covered polar regions, clouds are hard to be distinguished from the background due to the lack of contrast, and thus traditional statistical models built on raw sensor data tend to performed poorly on these data. To address this issue, the authors constructed three new features from the raw data based on the results of exploratory data analysis and domain knowledge. Using these constructed features, the authors were able to train a enhanced linear correlation matching (ELCM) - quadratic discriminant analysis (QDA) algorithm, which out-performed the operational algorithms currently adopted by NASA. With such results, the authors claimed that they have successfully developed a n algorithm, and a feature construction scheme, that can accurately delineate cloud-covered regions from snow-covered background. The authors argued that such results demonstrated how statisticians can be directly involved in solving real-world scientific problems, as opposed to traditional belief that statistics can only play a supportive role in scientific research.

In this study, we investigated a subset of the dataset studied by Shi et al., with the goal of finding a simple statistical model that is capable of identifying clouds from snow-covered background with sufficient accuracy for real-world applications. To achieve this goal, three images with expert labels were retrieved from the MISR sensor dataset. (Figure 1) Each image contains ~115,000 data points (pixels), with 18%, 34%, 18% of pixels in each image labelled as cloud, and 47%, 37%, 29% labelled as background, respectively, and the rest unlabelled. Through visual inspection, it is clear that the dataset showed significant spatial trends with high local dependency. As such, the naive assumption that data points are independent and identically distributed cannot be applied here, and extra care needs to be taken during data processing to take into account the observed spatial trends.

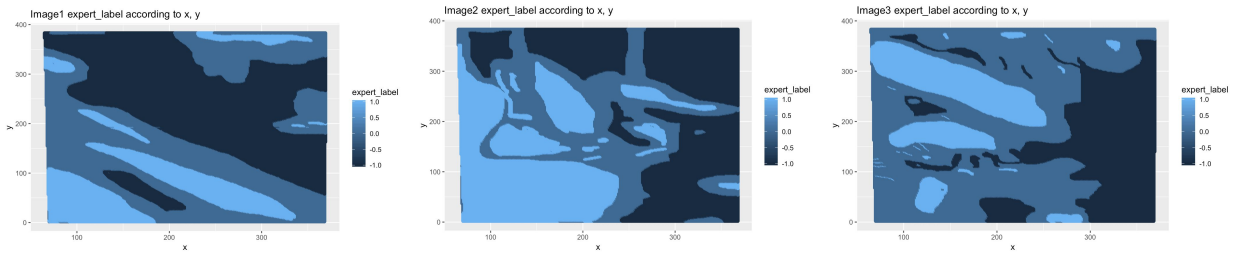

Figure 1. Three expert labelled MISR images investigated in this study

After removing unlabelled data points, exploratory data analysis was performed on the dataset to investigate the underlying correlations between the features and labels. By

constructing a correlation plot, it can be seen that labels are most strongly correlated to NDAI, a feature constructed by Shi et al, followed by CORR, radiance angle of sensor AF, radiance angle of sensor AN, and SD. (Figure 2) Such an observation was confirmed by the histograms of the three constructed features. (Figure 3) While NDAI and SD showed significantly different distribution between pixels labelled as clouds and background, the difference is not very large in the distribution of CORR. It is also noteworthy that the radiance angle of the five sensors are highly correlated to each other, and therefore including more than one of the radiance angles in a statistical model would be redundant.

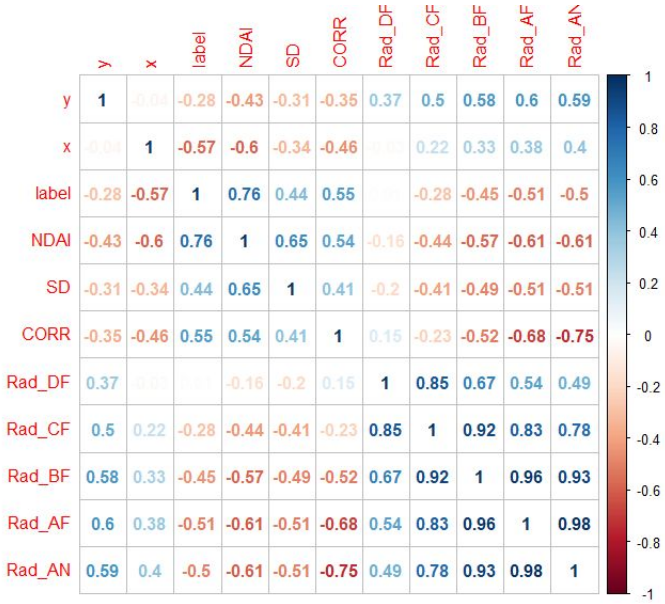

Figure 2. Correlation plot between the label and features. (NDAI: normalized difference angular index, SD: standard deviation of pixel values in the neighborhood, CORR: correlation between images of the same region viewed from different angles, Rad_DF: radiance angle of sensor DF)

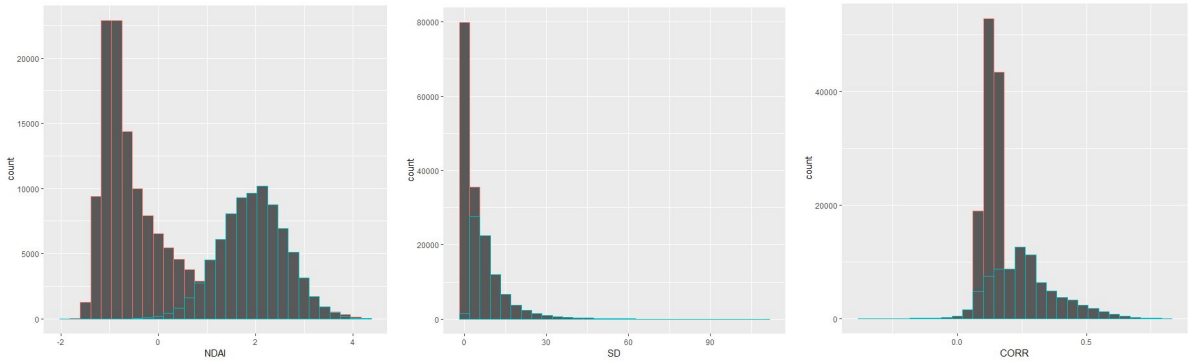

Figure 3. Distributions of NDAI, SD, and CORR values of pixels labelled as clouds (green) and background (red)

Preparation

We observed that the cloud data is not independent and identically distributed (i.i.d) from Figure 3. The labelled expert cloud data exhibited clearly visually identifiable structures of clusters which strongly indicated that our dataset is not i.i.d. We used two ways of handling this data characteristic -- either shuffle the data while maintaining the local spatial data dependency or, in contrast, partition dataset in a way that captures both types of expert labels so that the training set has diverse data to learn from.

Hence, we splitted the data in the following two ways:

Firstly, we splitted the data by taking into account the periodicity in the radiance angles of the MISR sensors. It is assumed that each period of sensor movement captures the complete spatial correlation of a region of the image. Briefly, each image in the data set was partitioned into four blocks, each block containing ~300 periods of radiance angle, obtaining a total of 12 blocks of data. The 12 blocks were then partitioned into three different sets (7 blocks in training set, 3 blocks in validation set, 2 blocks in test set). In this way, the dataset was randomized while retaining the observed local dependency.

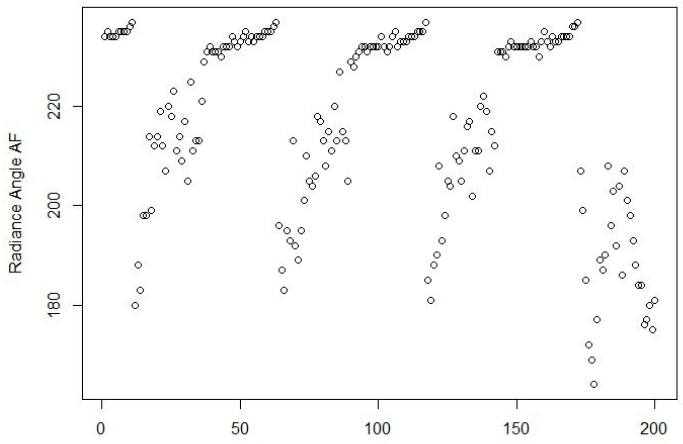

Figure 4. Periodicity of radiance angle of sensor AF. The radiance angle repeats itself after every ~50 data points.

Second, we partitioned each given image into three chunks of dataset based on x intervals. Specifically, we sliced the image every 101 increment of x. For example, the minimum x position for image1 is 65 and maximum is 369. So, one sliced chunk of dataset is all data points within 65 <= x <= 166. This was one simple approach to capture data that are marked either cloud(1) or non-cloud(-1) by experts. Visually, we observed that such interval partitioning enables us to sufficiently capture both labels of experts.

To ensure that the given classification problem is not trivial, we tested our datasets with a trivial classifier which classifies all data as non-cloud (-1). The results are following:

For our first splitting method based on the periods of radiance angle, the baseline percent error on the test set was 54%. This result agreed with the expectation that in a binary classification problem, the percent accuracy of a trivial classifier should be ~50%. Therefore, it confirmed that the dataset has been sufficiently randomized.

For the second splitting method based on x intervals, the trivial classifier, depending on the test dataset, performed as low as 15.28% accuracy and as high as 95% accuracy. Such high accuracy indicated that some of our partitioned dataset contained very skewed information with most of the data being labelled as non-cloud, and vice versa. Acknowledging this issue, we selected test set that contains non-skewed data (i.e. approximately half of the data are cloud, and the other half is non-cloud) by selecting two out of nine partitioned dataset from visual inspection of partitioned datasets. This resulted in the trivial classifier performing 46.03% accuracy on the test set. And, in the remainder 7 out of 9 partitions, which will be used as train and validation set, the trivial classifier performed 64.36% accuracy. These results show that our training and validation set, as a whole, as well as test set “roughly” contains equal amount of both cloud labels.

Without using any fancy classification methods, we determined that NDAI, SD, and AF are the best features to use to predict expert label. We justify our claim based on correlation plot and box plots as shown in Figure 2 and Figure 5, respectively. The correlation plot shows the correlation among all given data excluding unlabelled ones. Figure 2 shows that NDAI, SD, CORR, and AF are four of the most highly correlated features. Since DF, CF, BF, AF, and AN are all highly correlated to each other, we choose only one of these five features to avoid unnecessary redundancy of information.

We used boxplots to validate the utility of these features. Our expectation is that if a given feature is a good feature, then we should be able to classify the label based on the values of that feature. This means that the ranges of boxplot defining a cloud should not overlap with that of non-cloud. As shown in Figure 5, all box plots are overlapping so we lowered our standard to only the boxes to not overlap. With this criterion, NDAI, SD, and AF showed non- or barely overlapping boxplots. However, CORR had a clear overlap. So, we eliminated CORR from our good feature list. In conclusion, we determined that NDAI, SD, and AF are the three best features to predict expert label.

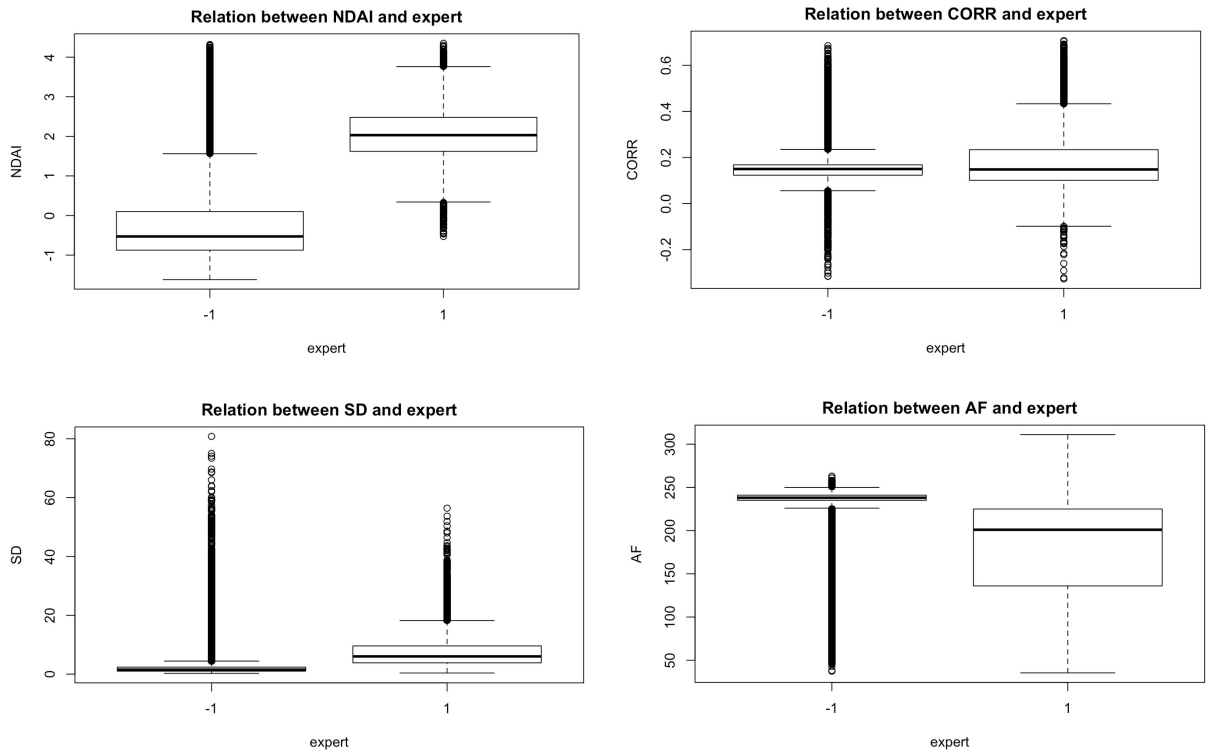

Figure 5. Box plots between features and labels

Modeling

To predict the expert label correctly using the aforementioned three best features (i.e. NDAI, SD, AF) that we selected, we experimented with four different supervised learning algorithms to model the given data. We combined our training and validation sets for cross validation where we partitioned the combined set into five disjoint folds whose union is the given combined set. The results of the cross validation errors are shown in Table 1. Also, the classification error of learned models from four algorithms on test set is shown in Table 2.

Algorithm	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average CV Error
LDA	0.1806	0.0864	0.0211	0.0943	0.0780	0.0921
QDA	0.1884	0.0835	0.0293	0.0765	0.0952	0.0946
Logistic Regression	0.1871	0.0866	0.0664	0.0788	0.0959	0.1029
Random Forest	0.1545	0.0726	0.0359	0.1122	0.0687	0.0888

Table 1. 5-Fold validation errors for four models with the first splitting method

	LDA	QDA	Logistic Regression	Random Forest
Test Error	0.1967	0.2091	0.2139	0.1831

Table 2. Test errors for four models with the first splitting method

Algorithm	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average CV Error
LDA	0.0672	0.0767	0.1956	0.0857	0.1024	0.1055
QDA	0.0757	0.0703	0.2147	0.0708	0.1192	0.1101
Logistic Regression	0.0737	0.1040	0.2134	0.0735	0.1462	0.1222
Random Forest	0.0759	0.0963	0.1831	0.0958	0.1217	0.1144

Table 3. 5-Fold validation errors for four models with the second splitting method

	LDA	QDA	Logistic Regression	Random Forest
Test Error	0.1038	0.1254	0.1236	0.9146

Table 4. Test errors for four models with the second splitting method

We also checked for the assumptions for the algorithms we experimented. LDA and QDA both assume that the likelihood (i.e. probability distribution of our feature data for each label) is jointly gaussian. In Figure 8 shows the probability density function of each of the three features on cross validation set. We observe that, in general, the distributions are not gaussian. Most of the feature distributions are skewed and, on one extreme, AF feature with -1 expert label has two peaks in its distribution which is clearly not gaussian. These individual densities indicate that the joint distribution of these three features are not gaussian.

	NDAI	SD	AF
NDAI	-0.6828816	-1.352402	12.00687
SD	-1.3524020	44.484962	-38.24532
AF	12.0068660	-38.245319	989.85141

Figure 7. Difference Matrix (i.e. Covariance Matrix for Cloud Dataset - Covariance Matrix for Non-Cloud Dataset)

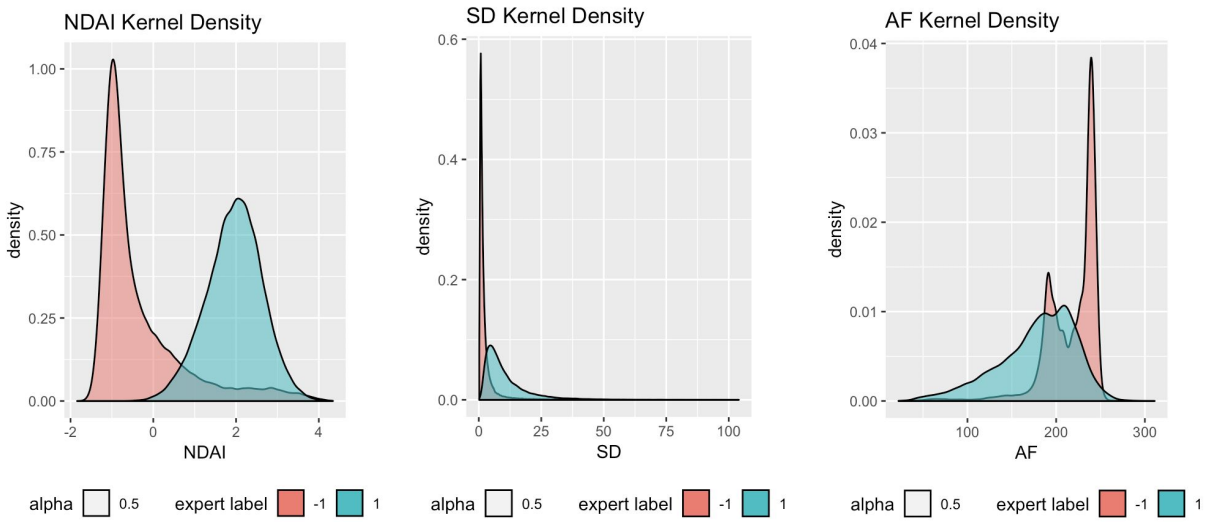

Figure 8. Kernel density of three features for the overall cross validation set

Furthermore, contrary to QDA, LDA further assumes that the covariance of the given cross validation set is equal across different expert labels. We find that this assumption is not true by computing the difference between the covariance matrix of cross validation set with only cloud label and that of only non-cloud label. Our computed result is shown in Figure 7. This means that QDA's assumption that the covariance matrices across different labels are not the same is true.

The logistic regression assumes that the log ratio of Bernoulli distribution defining the two classes are of a specific exponential form. However, this assumption is difficult to check so we did not validate this assumption. Finally, random forest algorithm does not hold any assumption about the dataset.

The mismatches in the assumptions for LDA and QDA explain the source of cross validation error. Also, although we did not explore in detail, we reasonably conjecture that the partitioning of the cross validation set could have also distorted the distribution over each feature, thereby further deprecating the jointly gaussian assumption of LDA and QDA. This conjecture is partially affirmed by high variance in the cross validation error across different folds for all four experimented algorithms as shown in Table 1.

In reviewing our cross validation results in Table 1, we observe that LDA, on average, performs slightly better than QDA despite the fact that more of LDA's assumptions are violated. However, because we do not know how exactly the violation of the assumption on jointly gaussian among features for LDA and QDA affect their modelling accuracy, it is difficult to directly assert that the number of violated assumption is directly proportional to classification error. This conclusion could be an explanation as to why LDA is consistently performing better than QDA. Logistic regression consistently performed the worst. The other algorithms varied in performance depending on the splitting method.

When we trained each algorithm with the entire cross validation dataset and classified the test data, we observed that the test accuracy, in general, was significantly better for the second splitting method contrary to similar performances for cross-validation. Hence, from now on we only refer to the second splitting method to measure the performance of algorithms. For test error using second splitting method, random forest performed the highest but only by small margin and it had the slowest runtime by far.

Finally, to find the optimal cut-off value for each learned models, we used two different metrics to compute the cut-offs. Metric 1 minimizes $fpr^2+(1-tpr)^2$ as cutoff is varied. Here, fpr is false positive rate and tpr, true positive rate. To minimize this function, fpr needs to be closer to zero while tpr to 1, which is what we want for high classification accuracy for both labels. Metric 2 maximizes both the true positive rate (i.e. sensitivity) and true negative rate (i.e. true negative rate). Table 5 shows the effects of these computed cutoffs on validation and test accuracy for each algorithm for only the second splitting method due to spatial constraint on this report. We notice that, except for QDA, optimal cutoffs computed using Metric 1 improved the test accuracy of all three other algorithms by small margin. On the other hand, Metric 2 heavily deprecated the test accuracy for all algorithms. Although, at an intuitive level, Metric 1 and 2 are enforcing similar constraints, the effects of the metrics were completely different. The ROC curve for each algorithm with optimal cutoff computed with Metric 1 is shown in Figure 9. All algorithms were trained using train dataset. Then, trained models are used to draw ROC curves using validation set.

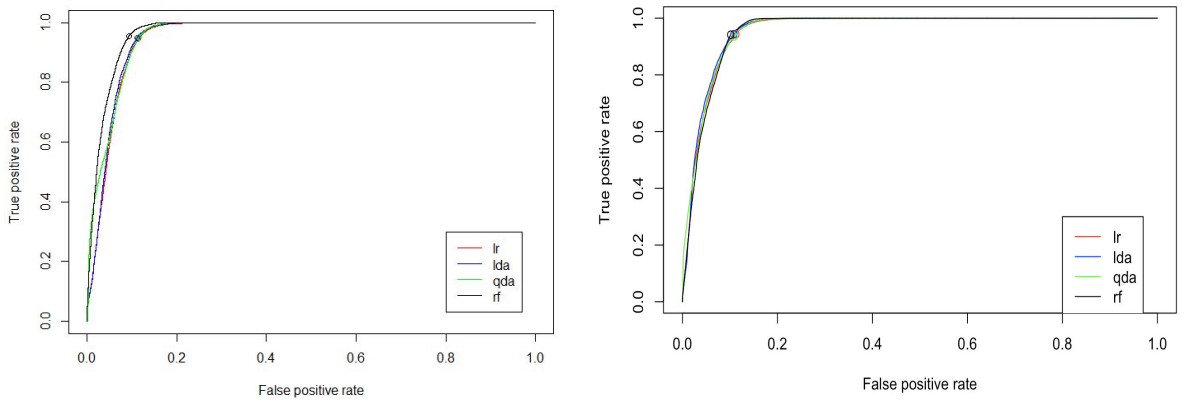

Figure 9. ROC curves of all four algorithms with optimal cutoff values computed using Metric 1 marked as circles for the first data splitting method (left) and second data splitting method (right)

	LDA	QDA	Logistic Regression	Random Forest
Default Test/ Validation Error	0.1028/ 0.0872	0.1255/ 0.1031	0.1236/ 0.1275	0.0912/ 0.1046

Metric 1 Test/Validation Error	0.1017/ 0.0889	0.1667/ 0.1345	0.1182/ 0.0877	0.0885/ 0.0797
Metric 2 Test/Validation Error	0.5134/ 0.4393	0.5398/ 0.4395	0.4604/ 0.5567	0.4285/ 0.4636

Table 5. Comparing the effects of three types of thresholds computed by default, metric 1, and metric 2 on test and validation for each algorithm using only second splitting method

Diagnostics

To evaluate the rate of convergence of the models, we performed diagnostic analysis on LDA, the most promising model. LDA was chosen as the subject of study because although it was the second best performing model next to random forest, it showed comparable accuracy and much higher computational efficiency compared to random forest. Since we are aiming to develop an algorithm capable of classifying data collected by a satellite in ***real-time***, we judged that efficiency should be perceived as equally important as, if not more important than, accuracy.

To better justify the parameters of LDA, we looked into the gaussian means of each feature values computed from LDA trained on train set for each label, as shown in Table 6. Also, we observed that the coefficients assigned to each feature: 1.1367 for NDAI, -0.0283 for SD, and -0.00176 for AF. We see that LDA assigns very high coefficient for NDAI as a determining factor of its decision. In fact, the NDAI's gaussian means computed for both labels are approximately the medians of NDAI boxplots shown in Figure 5. Also, the standard deviation of NDAI for cloud labelled data is 0.6734, and for non-cloud labelled data is 1.066. The standard deviations are high such that there is much overlap between the gaussian distributions of two labels for NDAI features. However, from Figure 8, the kernel density function of NDAI feature for non-cloud is actually quite skewed, which makes the linear classification considerably easier. This explains LDA's high performance.

	NDAI	SD	AF
No Cloud	-0.3243	2.97451	217.4795
Cloud	1.9548	10.5397	178.6983

Table 6. Average values for gaussian fitting of each feature for cloud and non-cloud label computed using LDA on training set

Furthermore, as shown by Figure 10, validation error converged to minimum when the size of training size reached ~84,000, roughly 67% of the size of the training set used in previous

analysis. Such a result showed that LDA is a stable and robust model for solving this specific problem.

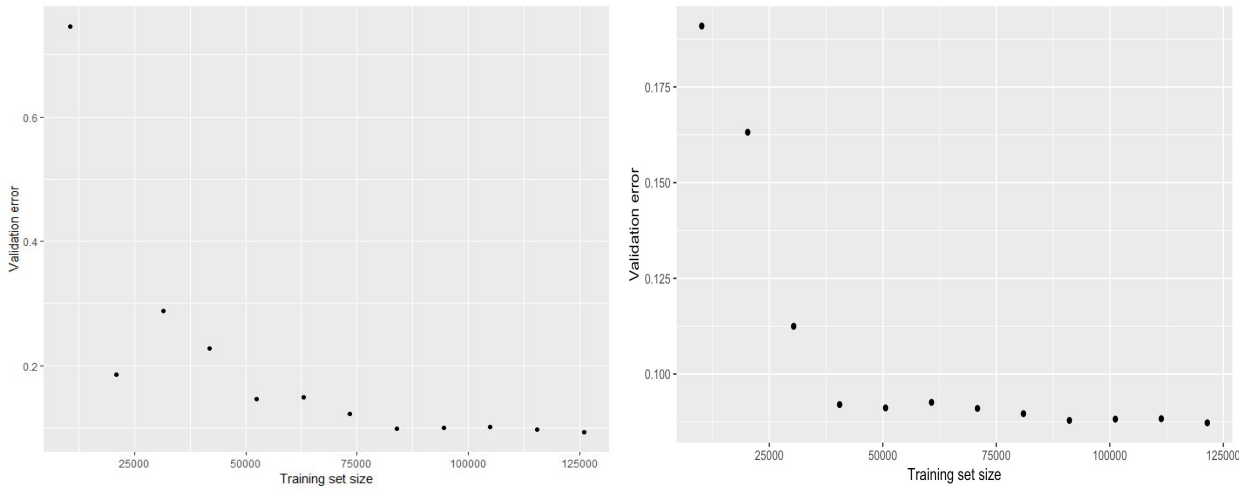

Figure 10. Validation errors of LDA trained on training sets with different sizes for the first splitting based on periodicity (left) and the second splitting method (right)

To better understand when the model will fail to accurately classify data points, we studied the distributions of the three constructed features (NDAI, SD, radiance angle AF) among the misclassified data points. (Figure 11) While SD and radiance angle AF showed similar distributions in misclassified and all data points, a significantly narrower distribution of NDAI was observed in the misclassified data points compared to all data points. To further investigate the potential of cause of such an observation, a colormap of NDAI values for a segment of validation set was plotted on x-y coordinates. (Figure 12) From the color maps, it can be seen that LDA model tends to misclassify when, within a block of pixels with the same label, drastic differences in NDAI values between neighboring regions were present. Considering the high correlation observed between NDAI and the labels, such an observation likely indicates that the classification results of LDA is highly dictated by NDAI. We observe that very similar results occur with misclassification with the second splitting method as shown in Figure 13 and 14.

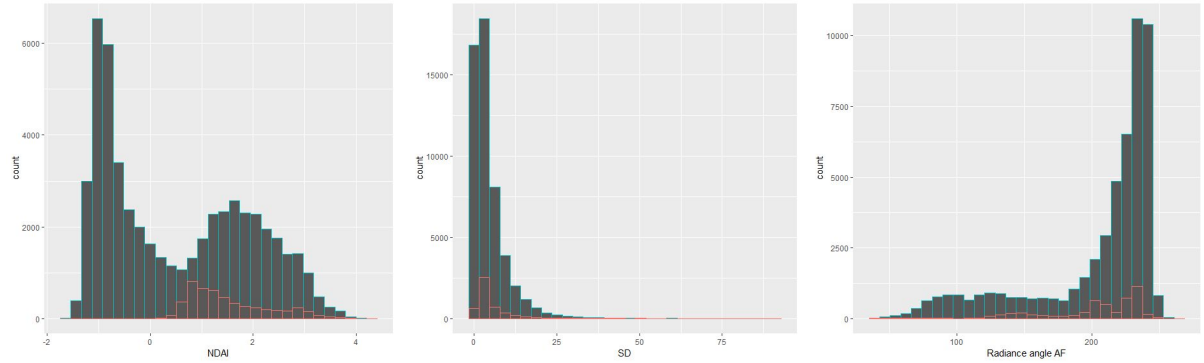

Figure 11. Distribution of constructed features in misclassified data points (green: all data points in the validation set, red: misclassified data points in the validation set) for first splitting method

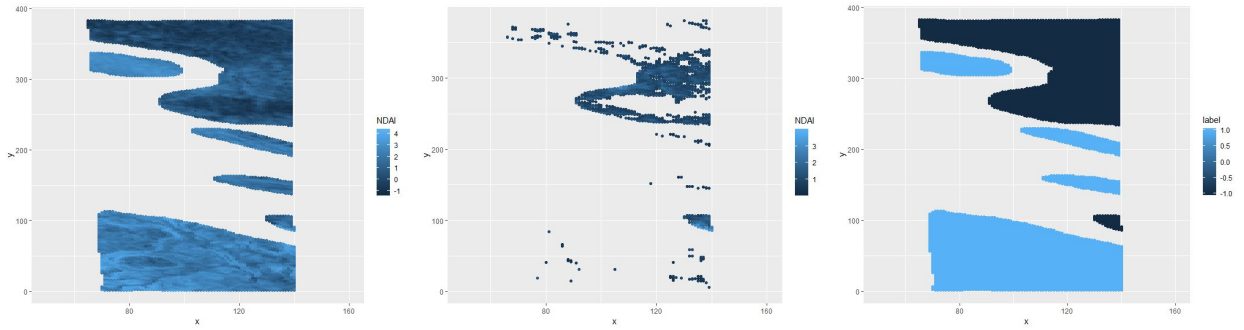

Figure 12. Colormap of NDAI values of all validation data points (left), NDAI values of misclassified validation data points (middle), expert labels (right) on x-y coordinates for first splitting method

One suggestion to improve our classifier is to enhance NDAI feature to reduce sudden drastic changes in NDAI values in near neighborhood. It is desirable to “smooth” out the changes in NDAI so that it vary gradually over variations in x,y space. We propose to do so by altering the way it is computed. In the paper [Shi, Yu, et al. 2008], NDAI is computed over average of 4x4 pixel values (which amounts to 1.1km width square area) of camera captured from two different angles. Instead, if we compute NDAI over larger area, for example, 20x20 pixels (which amounts to 5.5km width square area), then we can have enforce more gradual NDAI variation across x,y space. We hypothesize that this feature enhancement could improve our accuracy. This hyperparameter of how many pixels average over to compute NDAI values need to be experimentally tested and tuned.

Additionally, another suggestion for improving classifier is the following. Since the classification made by the LDA model depends highly on the values of NDAI, when applied to classify unlabelled future data, this model will only perform well in datasets where NDAI is a good predictor of the presence of clouds. One possible way to address this issue would be to construct a classifier based on clusters of pixels, rather than single pixels. In this way, the label of a given pixel will be predicted based on not only the feature values associated to the pixel itself, but also that of its neighboring pixels. Taking into account the fact that clouds and background tend to appear in blocks (Figure 1), such a model would better capture the spatial trends in the dataset. As an example, if we could isolate each cluster of clouds or background from Figure 1, and use them to train a convolutional neural network with 2D convolutional layers, the resulting model will likely outperform those discussed in this report.

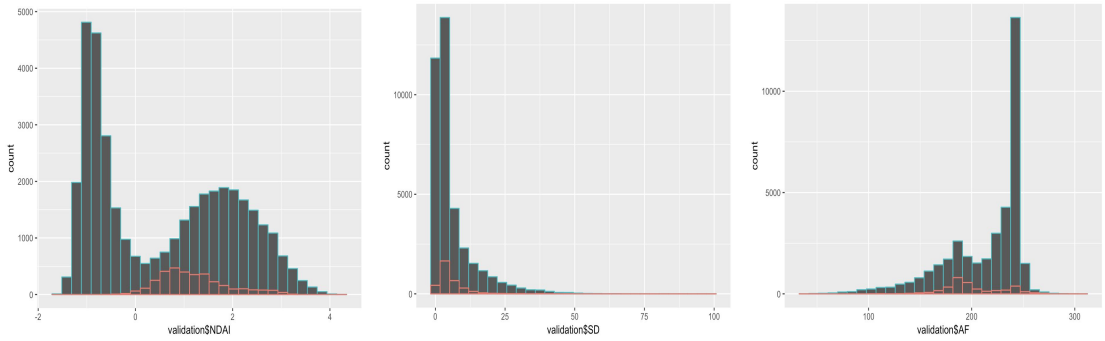

Figure 13. Distribution of constructed features in misclassified data points (green: all data points in the validation set, red: misclassified data points in the validation set) for second splitting method

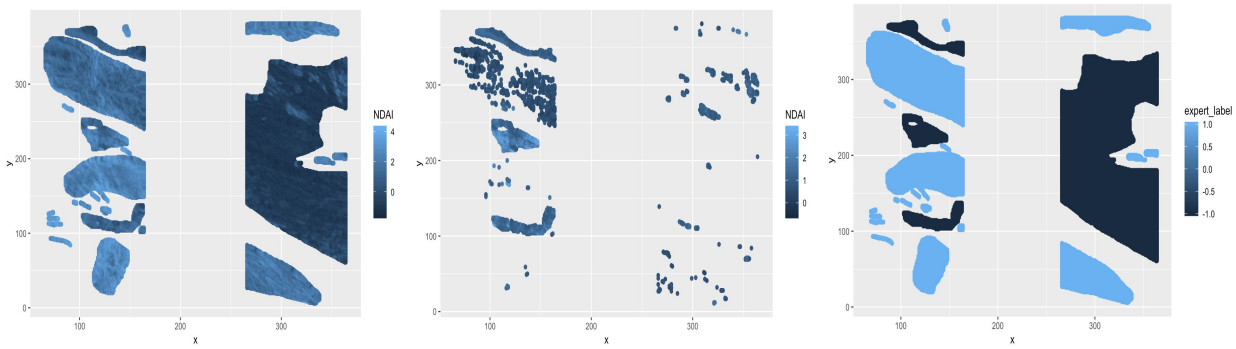

Figure 14. Colormap of NDAI values of all validation data points (left), NDAI values of misclassified validation data points (middle), expert labels (right) on x-y coordinates for second splitting method

In conclusion, we demonstrated that several statistical models, including QDA, LDA, and random forest, can be applied to separate clouds from the background with approximately 90% or higher accuracy. The performance of LDA was shown to be robust and stable as size of training set and data splitting method changed. However, the classification made by LDA was found to be highly dependent on the values of NDAI, which could limit its application to future data. Therefore, further steps need to be taken to improve the performance of the model by reducing its dependence on NDAI values of single pixels. All our figures, tables, and computed data can be reproduced by downloading our code from a public git repository at https://github.com/ek65/Artic_Cloud_Detection.git

Acknowledgement

Xingzhi Wang developed and analyzed the data splitting method based on the periodicity of radiance angles. He also wrote codes for generating correlation plots, parts of other EPA, K-fold cross-validation, and convergence analysis. Xingzhi composed the Data Collection and Exploration section, part of the Preparation section, and most of the Diagnostics section.

Edward Kim developed and analyzed the data splitting method based on x intervals. He wrote codes for visualizing the given images, boxplots, and ROC curves with optimal points. Edward wrote Data Preparation and Modelling section, and part of Diagnostics of the report.