Daytime Arctic Cloud Detection Analysis

Zhengyi Sui/3034503655 | Jiahua Zou/26209413
Contribution. Zhengyi Sui:1,2cd, 3ab, 4bde | Jiahua Zou: 2abd, 3ac, 4acd

1. Data Collection and Exploration

(a) Write a summary of the paper

The main purpose of the study is to distinguish clouds and non-cloud in the Arctic in a more stable and separable way. With MISR's nine cameras viewing from nine angles in four spectral bands (blue, green, red, and near-infrared), the data of 275-m resolution red radiation of each 7,114,248 pixels from 57 data units in 10 MISR orbits of path 26 are used mainly in this study. Three features, the linear correlation of radiation measurements from different MISR view directions (CORR), the standard deviation of MISR nadir red radiation measurements within a small region (SD) and a normalized difference angular index (NDAI), are derived from the radiation measurements. The result is ELCM and ELCM-QDA algorithms, which combines classification and clustering schemes computationally efficiently. Since the current data have no labels, the ELCM label pixels by thresholding the three features either in fixed or data-adaptive ways. Threshold are fixed for CORR and SD because they are stable and robust, while threshold for NDAI is data-adaptive determined by EM algorithm based on two Gaussian distributions. With two rationales based on the thresholds, label for each pixel is given. To overcome ELCM's absoluteness labels problem, Fisher's QDA is used to provide an estimate of probability. By comparison of ELCM, SVM, ASCM and SDCM, ELCM stands out in both 91.80% agreement with expert and 100% coverage, and further ELCM-QDA performs the probability of cloudiness at cloud boundaries, which satisfy both stability and separability. There are two significant impacts: firstly, statisticians are playing a more important role in data process in the domain of science directly; secondly, the power of statistical thinking and the ability of statistics to contribute solutions to modern scientific problems are flourishing.

(b) Well-labeled beautiful maps

Over all images, percent of each label is displayed in Table 1. Unlabeled data are the most, taking 40% of all data, and cloud takes 23.43%, clear takes 36.78% of all data.

	clear	unlabeled	cloud	obs
image1	0.4377891	0.3845560	0.1776549	115229
image2	0.3725306	0.2863522	0.3411172	115110
image3	0.2929429	0.5226746	0.1843825	115217
overall	0.3677552	0.3978950	0.2343499	345556

Table 1. Percent of each label(-1, clear;) and observations in each image and all data.

From maps we could see that cloud or not are almost consistent and clustered, except for few isolated pieces in image2 and image3. Thus the i.i.d. assumption for sample is not established.

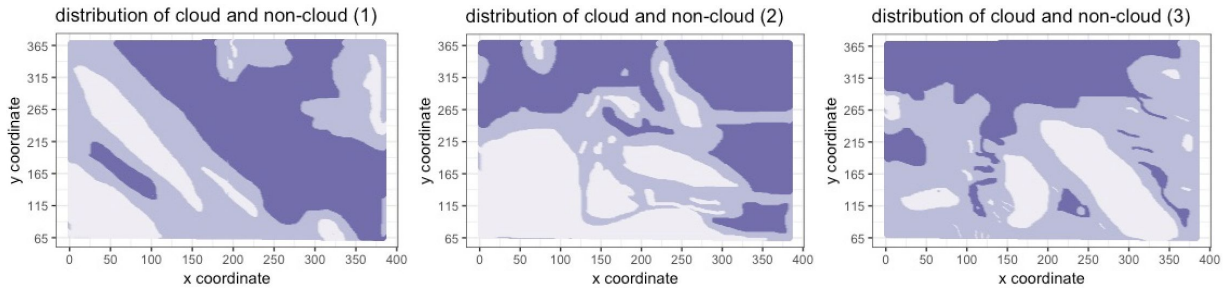

Figure 1. Distribution of cloud and non-cloud of image1, 2, 3. White represent cloudy; shallow purple, unlabeled; dark purple, clear.

(c) Visual and Quantitative EDA

Correlation plots of all features with labels indicate that NDAI has strong positive correlation with SD, and CORR has rather weak positive correlation with NDAI and SD. Inside red radiation from different angles, correlations among them are all strongly positive, while red radiations and three features are almost negative especially as measurement angles getting vertical. On the other hand, cloud has positive correlation with NDAI, SD and CORR, while basically negative correlation with five angles' measurements.

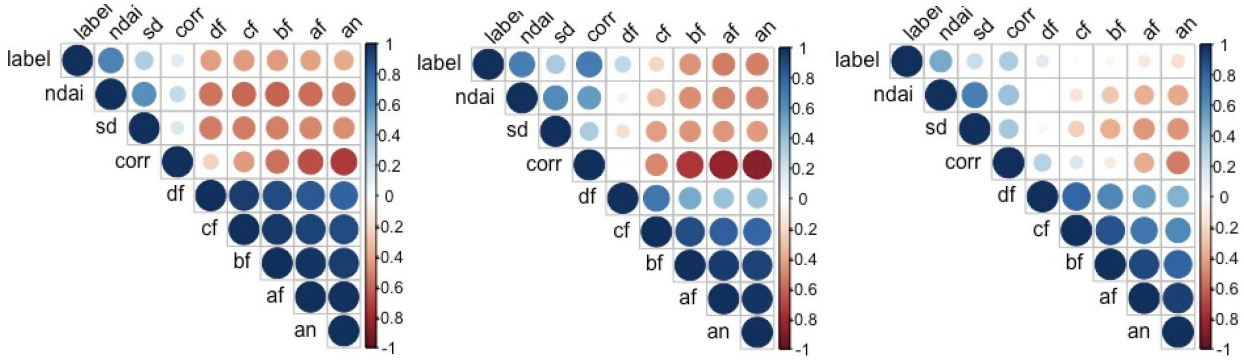

Figure 2. Correlation plot of all features of image1, 2, 3. Bigger deeper blue represents higher positive correlation, smaller shallower red represents weak negative correlation.

Boxplots of three features classified by label 1/-1 shows similar trend in all three images. Cloudy areas tend to have higher and more variant CORR, higher NDAI and relatively higher SD; while clear areas have smaller and more stable CORR, lower NDAI and relatively smaller SD.

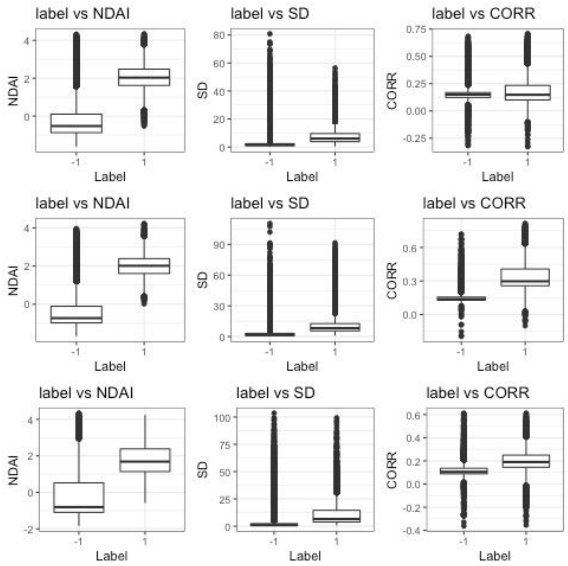

Figure 3. Boxplot of CORR, NDAI and SD of image1, 2, 3. The labels are -1(clear) and 1(cloud) from left to right; and image 1 to 3 from top to bottom.

2. Preparation

(a) Data Split

It is obvious from Figure 1 that spatial autocorrelation exist in our data. Randomly assign observations to each training and test set would destroy this autocorrelation. Therefore, we decided to split the data into "blocks", squares in equal size that divide the 2D map of x and y, and then we split the data based on "block" instead of individual observation. We first filtered out all the unlabelled data in the dataset, then divide the x values and y values into 3 equally spaced interval called xbins and ybins. Then we assign a "block" number to an individual observation based on which xbin and ybin the observation locates. There are 3 xbins and ybins each so 9 blocks in total for each image. The reason why we choose 3 is because we need the blocks to be larger enough to preserve the spatial autocorrelation and have enough observation for each block for sampling purpose. But on the other hand, we also need it to be as small as possible so that our subsequent train-val-test split is as random as possible. We try several number and subsequently arrived with 3 because 9 blocks per image return the most stable split across blocks. Divided by 4 and thus 16 blocks per image resulted in certain blocks not having enough observations and divided by 2 and have 4 blocks per image is obviously not random enough for our data split.

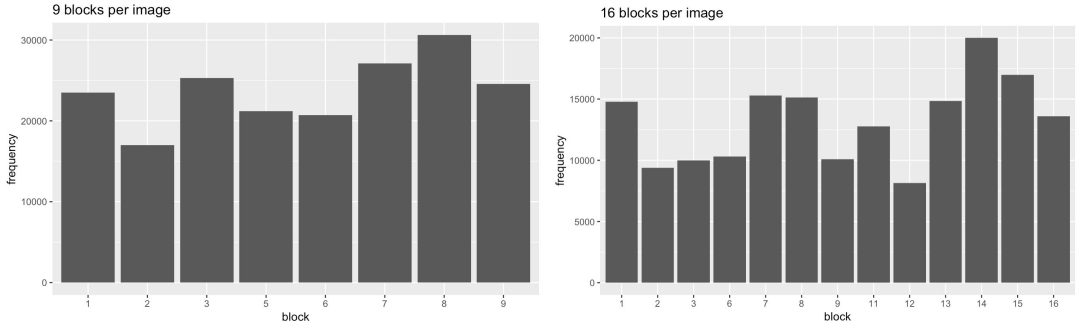

Figure 4.Barchart of the frequency of each block for 9 blocks per image (left), and 16 blocks per image(right).

The advantage of this method is that it is fast and easy to implement. However, we make an implicit assumption that the autocorrelation can be best preserved in a little square, which is clearly not the case judging by Figure 1. Our second method tried to find a more natural way to split the observations geometrically. We decide that instead of creating squares of equal area, we will use K Means Clustering method to cluster the observations into 9 clusters(k = 9), based on their x and y only. So this is an unsupervised learning process.

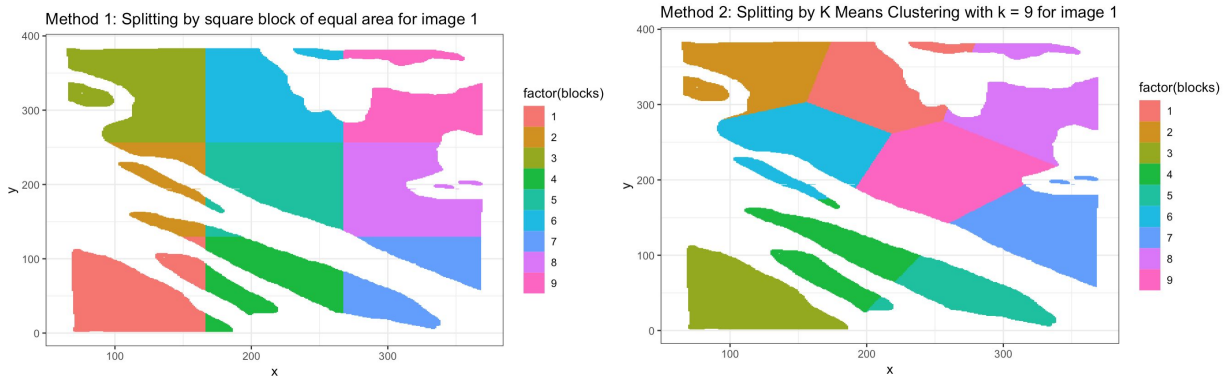

Figure 5. 2D plot of observations with x and y values, colored by each block .

Notice how the boundaries of the second split correspond more naturally to Figure 1 for image 1. We choose k = 9 to better compare the same method and for the reason that we want to have similar number of observations in each clusters.

(b) Baseline

A trivial classifier that reports everything to be no cloudy(-1) will have an accuracy of 0.6107824. This is equivalent of the fraction of -1 in the original combined data set. This means our classifier should at least be better than the trivial classifier. This trivial classifier will have high accuracy when the labels are imbalance, which mean most labels either belong to one class or another.

(c) First Order Importance

According to correlation of two split data in Figure 4, label is closely related to NDAI, SD, CORR and AN with respective correlation 0.7586, 0.4874, 0.6334 and -0.5168. However, correlation between CORR and AN is extremely high, reaching -0.7469. To avoid collinearity and provide more information, we substitute AN with SD. From this perspective, the “best” three features are NDAI, CORR and SD.

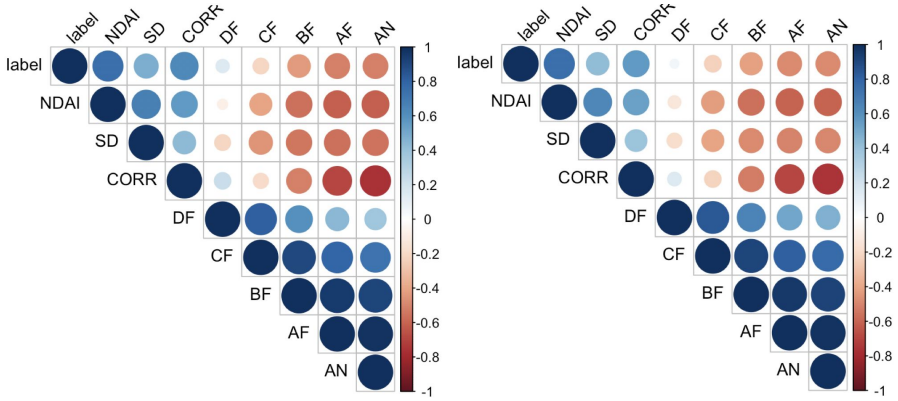

Figure 6. Correlation plot of label with all features of split data 1(left) and 2(right).

(d) CV Generic Function

The function is established with inputs and outputs as request in the cv.generic.R file.

3. Modeling

(a) Comparison of Logistic, QDA, KNN and Random Forest

We adopt three features as in 2(c) to fit the models. On the one hand, paper tells that all three features are derived from different angle measurements, so they contain the information of the rest five features and could prevent overfitting. On the other hand, Figure 6 suggests NDAI and CORR have the highest correlation with label, and though AN and AF also correlate highly, they are also strongly correlated with CORR. So choosing SD instead of AN and AF could avoid multicollinearity and provide more diverse information to the whole model.

Based on CV generic function defined in 2(d), Logistic regression, LDA, QDA, KNN and Classification Tree and Random Forest methods are established on train data. The results, given in Table 2, show that from two splits of data, KNN with K = 50 is the best classifier with an accuracy of 88.66% and 89.28% that consistently outperform other classifiers. We decided that K = 50 is best by running another 4-folds cross validation to tune that hyperparameter.

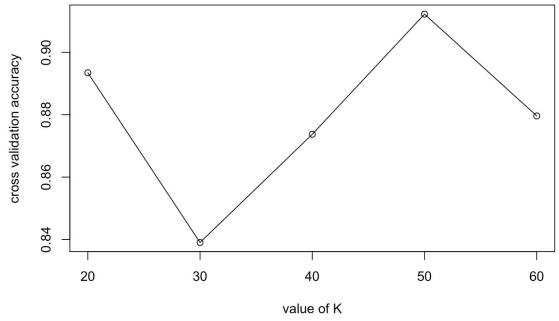

Figure 7. The cross validated accuracy of KNN classifier with different K values.

	logistic	LDA	QDA	KNN	Tree	Random Forest
1	0.8707591	0.8831658	0.9152114	0.8978656	0.9190333	0.8986888
2	0.9767658	0.9807591	0.9939898	0.9955629	0.9864870	0.9883425
3	0.7632411	0.7526385	0.7958757	0.7624629	0.6914061	0.7752541
4	0.9322675	0.9369585	0.9268585	0.9546695	0.9571586	0.9497870
5	0.8115811	0.8206227	0.8083183	0.8679928	0.8416149	0.8480226
6	0.8369923	0.8397127	0.8423787	0.8598983	0.8579940	0.8671890
7	0.8422354	0.8500044	0.8717224	0.8471793	0.8634237	0.8121303
8	0.8743590	0.8925926	0.8679012	0.9085470	0.9205128	0.9181861
average	0.8635251	0.8695568	0.8777820	0.8867723	0.8797038	0.8822001

	logistic	LDA	QDA	KNN	Tree	Random Forest
1	0.7569581	0.7739061	0.6884360	0.8088977	0.7965520	0.7782161
2	0.9865860	0.9928532	0.9998900	0.9782298	0.9998900	0.9761407
3	0.7239333	0.7521694	0.7305877	0.8491974	0.8378670	0.8014028
4	0.7979268	0.8020491	0.8198991	0.8197791	0.8014488	0.8315857
5	0.7860880	0.7839789	0.8212533	0.7891706	0.7560333	0.7859663
6	0.9214097	0.9179709	0.9312673	0.9247233	0.9214930	0.9229727
7	0.9910592	0.9931289	0.9967714	0.9941223	0.9874995	0.9889896
8	0.9553527	0.9632462	0.9644302	0.9783917	0.9742970	0.9645782
average	0.8649142	0.8724128	0.8690669	0.8928140	0.8843851	0.8812315

	logistic	LDA	QDA	KNN	Tree	Random Forest
test set 1	0.9747907	0.9802494	0.9850465	0.9853111	0.9965925	0.9782314
test set 2	0.9884669	0.9878950	0.9876090	0.9802221	0.9762665	0.9762188

Table 2. Accuracy relative to expert labels of the Logistics, LDA, QDA, KNN, Classification Tree and Random Forest. Split data 1 top, data 2 in the middle. The models are then trained on the whole training set and test it on the test set. The test accuracy for the two splits of data are shown at the bottom

KNN outstands other classifiers in this case might because of the origin of expert labels. Since expert label assemble like clusters, for most data points but the ones near edge, their surrounding observations are probably from the same labels, which improves KNN results. But in the long run, it might be lackluster.

Logistic regression does not show its power in this case, because normally it assumes labels independent given features, but expert labels here assemble so not i.i.d, which indicates its' invalidity. LDA and QDA assume on features' Gaussian distribution given labels. On the one hand, Figure 8 shows covariance of cloud and clear is different and the Box's M-test for homogeneity of covariance matrix is also, breaking LDA's equal covariance assumption. And more importantly, Q-Q plots in Figure 8 show in each label(cloud and clear), NDAI, CORR and SD violate the Gaussian assumptions in both split data set, which is the flaw of GDA methods. On the other hand, KNN, classification tree, and random forest are methods that do not require distributional assumption on the data, though classification tree is notorious for being unstable under the change of the data.

We also tried to run SVM but unfortunately for data set this large Rstudio crashes. We also train the models on the whole train data set (combined training set and validation set), and test the accuracy on the test set. The results shows classification tree with a near perfect 99% accuracy. However, this results should be taken with a huge grain of salt because the result vary greatly with the seed we set, and normally we only test our model when we have finally picked on that is most suitable for our data. Also, classification tree is known to vary a lot depending on the data and prone to overfit. Thus, the results here are just for grading purpose, and we stick to our original conclusion calculated by cross validation.

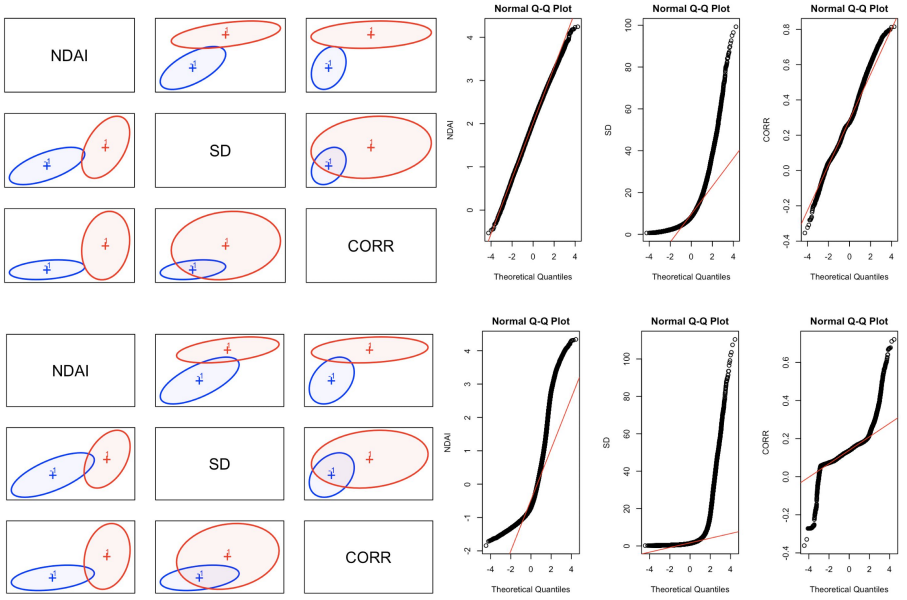

Figure 8. Covariance plot(left) and Q-Q plot(right) of features of data 1(top) and data 2(bottom).

(b) Comparison based on ROC curve

ROC curves and the AUC of six classifiers are shown in Figure 9 and Table 3. Clearly all classifiers but KNN have AUC approximately 1, indicating the models have ideal measure of separability. Almost all curves are close to the left-top point, reflecting the effectiveness of each classifier given proper threshold. In Logistics, LDA, QDA and Random Forest, cutoff value are the point closer to (0,1) using L1 distance; while since KNN and Tree has rather discreet thresholds and a clear elbow point, we pick that point as the threshold.

	Logistics	LDA	QDA	KNN	Tree	Random Forest
roc1	0.9996806	0.9997677	0.9985414	0.9378272	0.9966009	0.9991671
roc2	0.9963076	0.9963918	0.9953201	0.9154384	0.9565203	0.9927973

Table 3. AUC of each classifier of split data 1 and 2.

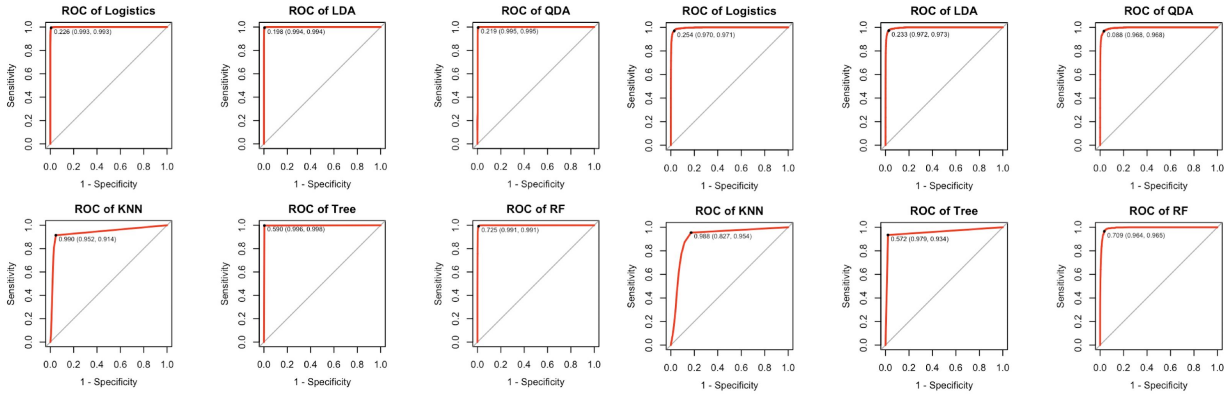

Figure 9. ROC curve for different classifiers of split data 1(left) and 2(right).

(c) Other Metrics

Thus, we also use precision and recall to evaluate our models by modifying our loss function and pass it to the CVgeneric function. The averaged cross validated precision and recall for 2 different data splits are displayed below.

	logistic	LDA	QDA	KNN	Tree	Random Forest
precision 1	0.8040476	0.8209623	0.8468692	0.8514138	0.8959971	0.8331857
precision 2	0.8194329	0.8322455	0.8246058	0.8564389	0.8479336	0.8344228
recall 1	0.8173616	0.8061395	0.8204857	0.7839618	0.7589445	0.8046530
recall 2	0.7490823	0.7380317	0.7674640	0.7533073	0.6697433	0.7309390

Table 4. averaged cross validated precision and recall for 2 different data splits

As it shows, Random Forest seems to be achieve a balanced outcome no matter what data split it is run on, or what performance measure it is being evaluated on.

Precision is the fraction of true positive over the sum of true positive and false positive, while recall is the fraction of true positive over the sum of true positive and false negative. One usually emphasizes recall when predicting 1 is more important than predicting -1, and do not care much for false positive, for example, in diamond-mining. One usually use precision as performance measure when one cares a lot about minimizing false positive, such as when deciding whether a patient should undergone a dangerous surgery. In the case of predicting whether it is cloudy or not, we do not really have a preference and would like overall low misclassification rate. Therefore, accuracy and ROC seem to be the preferred performance measure for our models than precision and recall, in this case.

4. Diagnostics

(a) Diagnostics analysis

In 3(a), we have illustrated the inefficiency of Logistic regression and LDA, QDA. Logistics doesn't hold for the labels are not i.i.d. In addition, Figure 8 shows that features in each label is not Gaussian, rejecting the Gaussian assumption in LDA and QDA.

Among KNN, Tree and Random Forest, we think that our random forest model achieves a stable performance, for different performance metrics and different data splits. Random forest accuracy reaches 97.8% and 97.6% in cross validation, AUC approaching 1, and precision and recall also keeps stable and competitive to other methods.

Since there is no parameter estimation in random forest model, we diagnose our model by plotting the error rate against the number of trees.

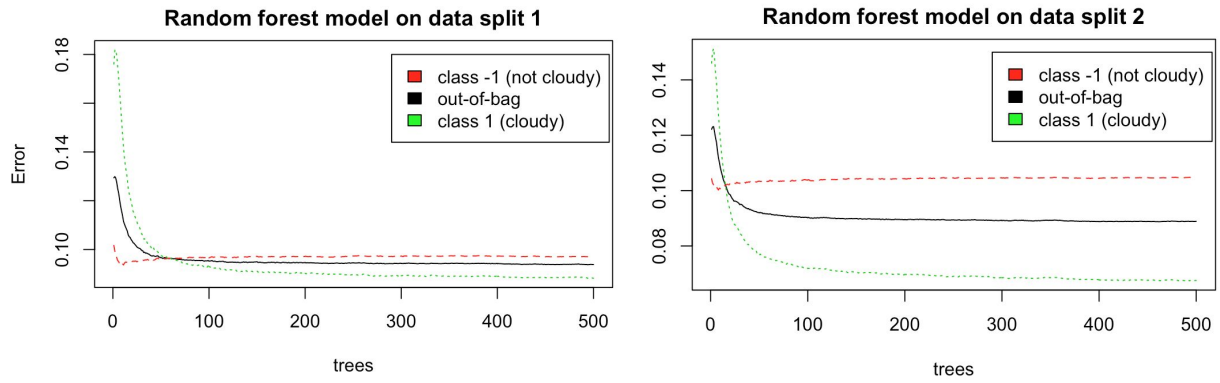

Figure 10. Misclassification rate of random forest model against the number of trees, data used to generate the plot is the combination of training and validation set.

It is obvious that the misclassification rate for both classes and out-of-bag estimates are steadily decreasing as the number of trees used to train model increase. It is also worth noting that class -1 has a higher misclassification rate than class 1 constantly. It may be the case that our random forest model is very aggressive and predict a lot of non-cloudy to cloudy.

Theoretically, unlike linear regression or other similar methods, adding more tree will not overfit the data because trees are not features, and will not add to the complexity of the model. We choose a maximum number of trees of 500 because we want our model to be as stable as possible, and we want to be trained within a reasonable time (1 minute for full train and validation set, 6 minutes for CVgeneric). In the future, it is probably safe to cut back the maximum number of trees to, perhaps a 300, without losing a lot of accuracy. This will speed up the runtime.

(b) Misclassification Patterns

The agreement of Random Forest on testset 1 and 2 reaches 97.8% and 97.6%.However, different from the paper, we don't find certain pattern in cloud regions. Instead, from Figure 11, both misclassifications happen, in both edge and inside of each assembly. So we deduce the reason of disagreements does not lie in edge area but in other potential factors, such as sharp terrain changes as is mentioned in paper.

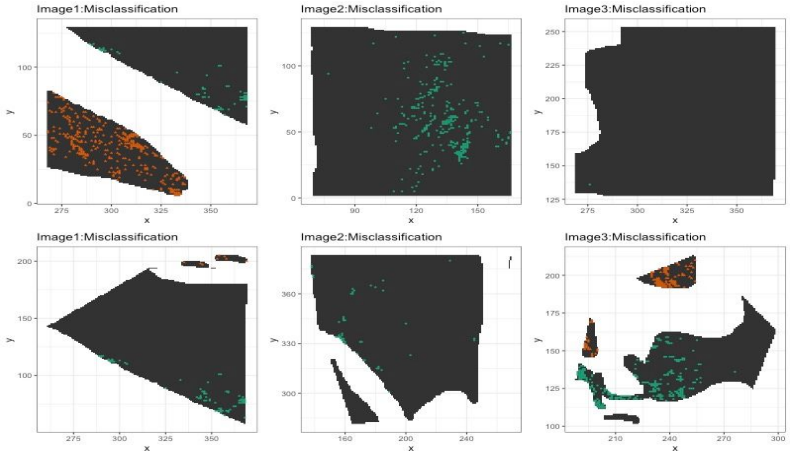

Figure 11. The misclassification in test region. Image 1 to 3 from left to right, data 1(top) and 2(bottom).

Figure 12 shows the pattern between disagreements and features. In each boxplot, it displays the distribution of one feature within each label (cloud and clear), and the red/blue plots are the points misclassified to the opposite class. In both split method 1(red) and 2(blue), cases are similar. The clear data misclassified to cloud mainly has NDAI and SD out of IQR and higher than 25% quantile, while CORR distributes rather symmetrically. This implies these points have unusual larger NDAI and SD, which is the characteristic of cloud area derived from ELCM algorithm rationale in paper, while CORR is not the deterministic factor in this case. As for the cloud points misclassified to clear, no certain pattern are found since all features seem evenly distributed, which might due to the abnormal combination of the features but unclear to us.

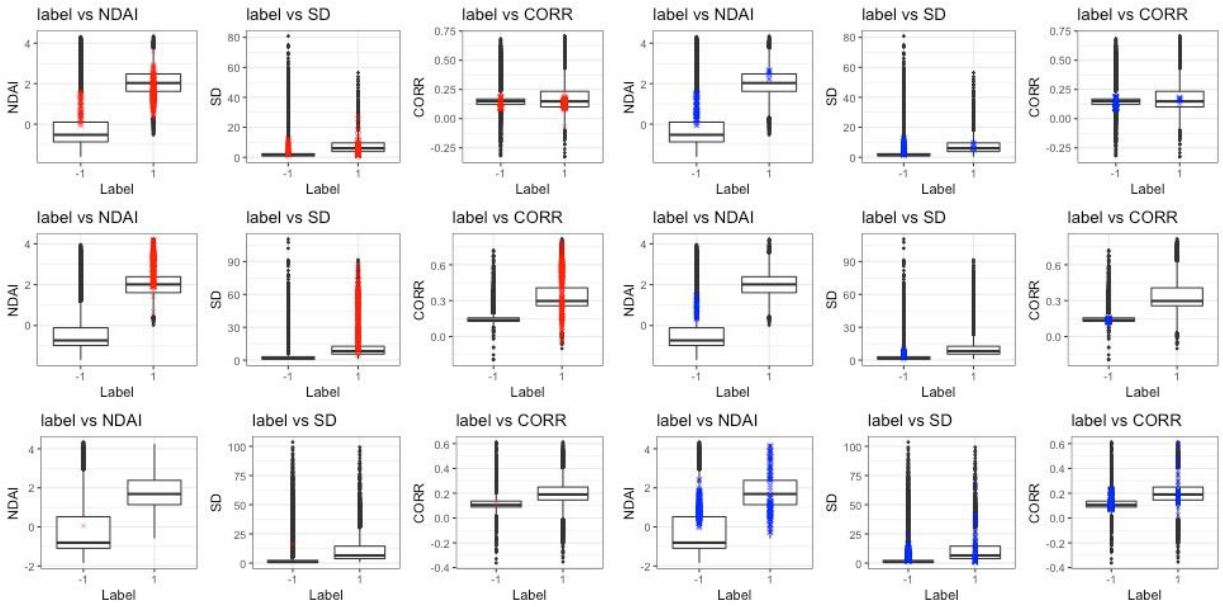

Figure 12. The misclassification in view of test features. Image 1 to 3 from top to bottom, split method 1 and 2 in red and blue.

(c) Better Classifier

Based on 4(a), the first we can do is to choose an alternate cutoff value other than the default 0.5, as we have done in ROC curve. However, it is quite impossible to do CV with Random Forest to select a cutoff value that is both stable and optimal because CV with Random Forest and with so many possibility of cutoff values take forever to run. Thus we have arbitrarily choose a cut of value of 0.48, which is not far from the default 0.5, to see what will happen if we adjust the cutoff.

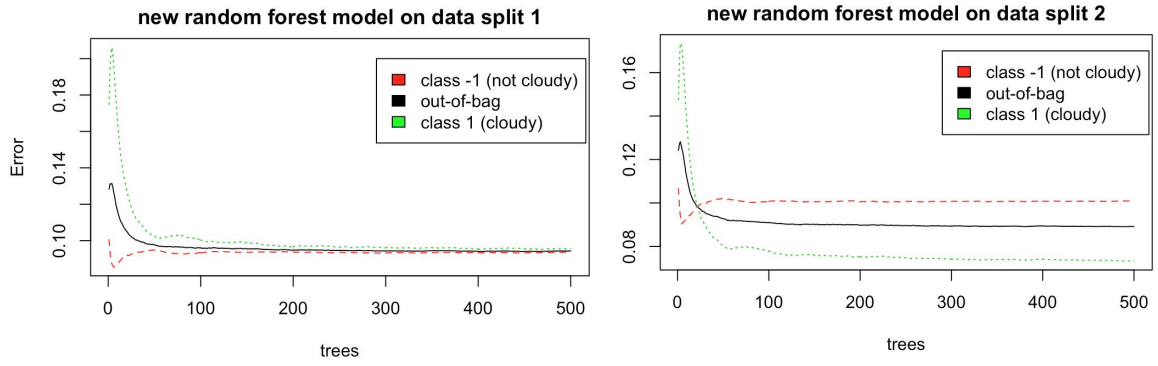

Figure 13. Misclassification rate of random forest model against the number of trees, data used to generate the plot is the combination of training and validation set, cutoff values are 0.48.

From Figure 13 it is evident that we have closed some of the gap of the misclassification rate between two classes, which is a good thing for our model. Another way to improve our model is to use more features from the original dataset such as AN, though this could overfit or lead to much slower runtime which is one of the reason why we do not attempt to try all the features in the first place. We expect our model to work pretty well because the cross validation error is quite good across all folds, and we believe our way of splitting data based on blocks is reasonable and replicable when predicting on new images.

(d) Comparison between two split data

Based on Figure 10, we can see there's distinction between two splits of data. It shows the misclassification rate of not cloudy in the second split is significantly higher than that in the first split.

However, both split data have similar misclassification patterns. On the one hand, no clear patterns are found in both ways whether disagreements pile up near edge or inside. And on the other hand, both clear area classified to cloud tend to have higher NDAI and SD, which is the characteristic of the cloud area; while the cloud area classified to clear has ambiguous pattern, which might under other potential factors impact.

(e) Conclusion

To conclude, error rate of random forest converge as tree gets larger, since different threshold result in different convergence, to minimize it, we adopt 0.48 as threshold instead of the default 0.5. Besides, only misclassification of clear to cloud has clear feature patterns that these disagreement points have higher NDAI and SD, which is the characteristic of cloud area. No apparent region pattern nor misclassification of cloud to clear pattern is found. Different data split methods have different Random Forest error rate, but they perform similarly in patterns.

5. Reproducibility

GitHub link:
https://github.com/JiahuaZou/arctic_cloud_recognition.git