Daytime Arctic Cloud Data Report

Statistic Rocks Team: Zheyuan Liu (zl321@duke.edu), Jiongran Wang (jw770@duke.edu)

November 2021

1 Acknowledgement

We contribute equally to this project. Zheyuan is responsible for problem 1 & 2, and Jiongran works on problem 3 & 4. We worked on the summary, report and README.md file together. Based on the paper[1], we generated the summary report and get familiar with data. We also want to appreciate the help and suggestions Dr. Chen provided. Based on EDA results, we split the whole data into blocks using two methods. Then we fit several classification models and determine the best model based on evaluation metrics. For the best model, we also diagnosed and improved it further.

2 Data Collection and Exploration

(a) Based on the paper[1], cloud coverage has been a very important indicator of dependencies of surface air temperatures on increasing atmospheric carbon dioxide levels in the Arctic. However, this is a challenging problem since liquid- and ice-water cloud particles often have scattering properties similar to those of the particles that compose ice- and snow-covered surfaces. In order to overcome this, the paper proposed two novel operational Arctic cloud detection algorithms (ELCM and ELCM-QDA) using MISR imagery. This study collected the data from 10 MISR orbits of path 26 over the Arctic, northern Greenland, and Baffin Bay. These 10 orbits span approximately 144 days from April 28 through September 19, 2002 (a daylight season in the Arctic). For each orbit, six data units were included in this study. But three of the sixty data units were excluded. Therefore, the data contained 57 data units with 7,114,248 1.1-km resolution pixels with 36 radiation measurements for each pixel. For all paths, MISR collects data every sixteen days repeatedly. Each MISR pixel covers a 275 m $\times$ 275 m region on the ground, which generates a massive dataset. However, these massive data sets cannot be downlinked because of the transmission channel limitations. Hence, only the red radiances and all channels from the nadir camera are fully transmitted, whereas the blue, green, and near-infrared radiances are aggregated before transmission.

This paper concludes that three physical features, NDAI, SD and CORR, are totally helpful to separate clouds from ice- and snow- covered surfaces. The algorithms they proposed are more accurate and provide better spatial coverage than the existing algorithms. This paper has three main impacts: the first one is statistical technical development; the second is that it demonstrates the importance of statistical thinking and the ability of statistics to contribute

effective and innovative solutions to modern complex scientific problems; the third one is practical implication. This study is helpful in inferring changes in the Arctic that are brought by increasing carbon dioxide.

(b) Based on Figure 1, we realize expert label -1 appears the most in image 1 & 2, and expert label 0 appears the most in image 3. Figure 2 indicates that for all three images, there are obvious boundaries between different classes, which indicate that most close pixels (pair of coordinates) belong to the same class. For example, in image3, pixels whose x-coordinate ≥ 300 are all in not-cloud class. Hence, an i.i.d. assumption is not justified for this data set.

types <chr>	expert_label <dbl>	percentage <dbl>
image1	-1	0.3725306
image1	0	0.2863522
image1	1	0.3411172
image2	-1	0.4377891
image2	0	0.3845560
image2	1	0.1776549
image3	-1	0.2929429
image3	0	0.5226746
image3	1	0.1843825

9 rows

Figure 1: % of pixels for the different classes

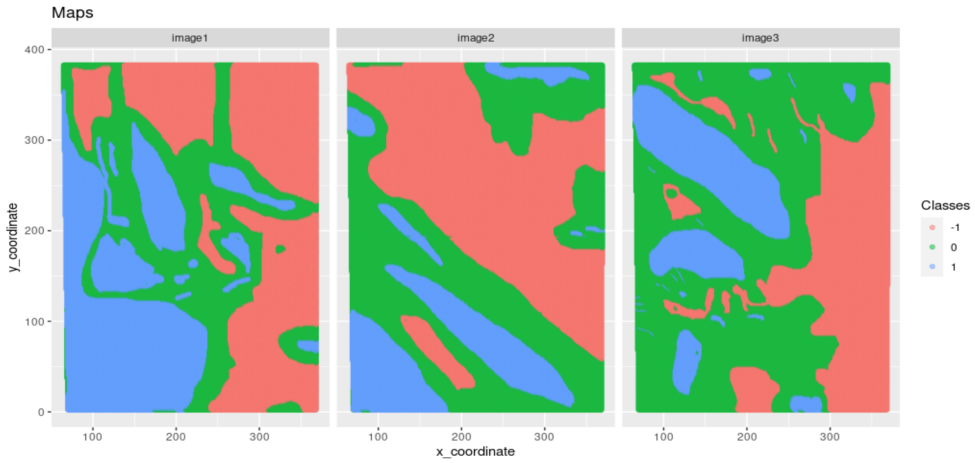


Figure 2: Expert Label Visualization for Each Image

(c) Based on the Figure 3, the highest correlation is 0.971 between Radiance angle AN and Radiance angle AF. Furthermore, correlation within all radiance variables are all above 0.5. For other features, we found there's a high correlation (0.631) between NDAI and SD, but correlation within other pairs of NDAI, SD, and CORR are not significant. We first exclude unlabeled observations, then we compare each feature's distribution for cloud and no-cloud class. Based on the Figure 4, we noticed distribution differences for NDAI and SD between those two classes, specifically, cloud class has in average higher NDAI and SD compare to no-cloud class. However, for other radiance variables, there are slight distribution differences for BF, AF, and AN where no-cloud group has in average higher values.

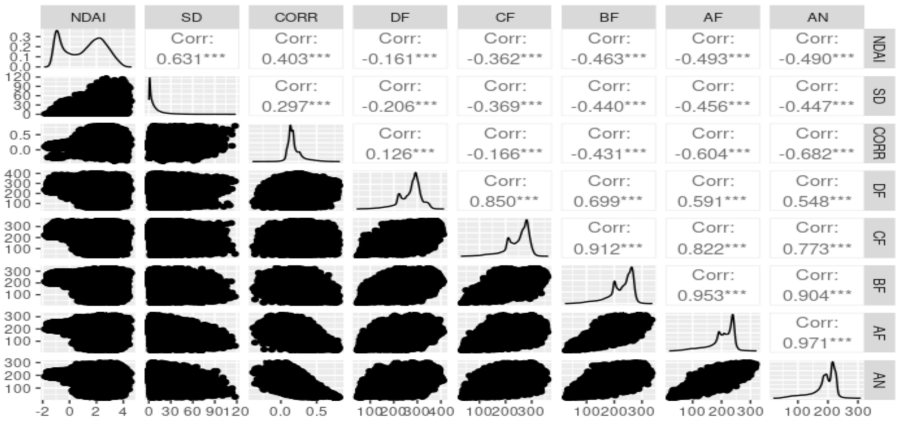


Figure 3: Pairwise Correlation Between Features

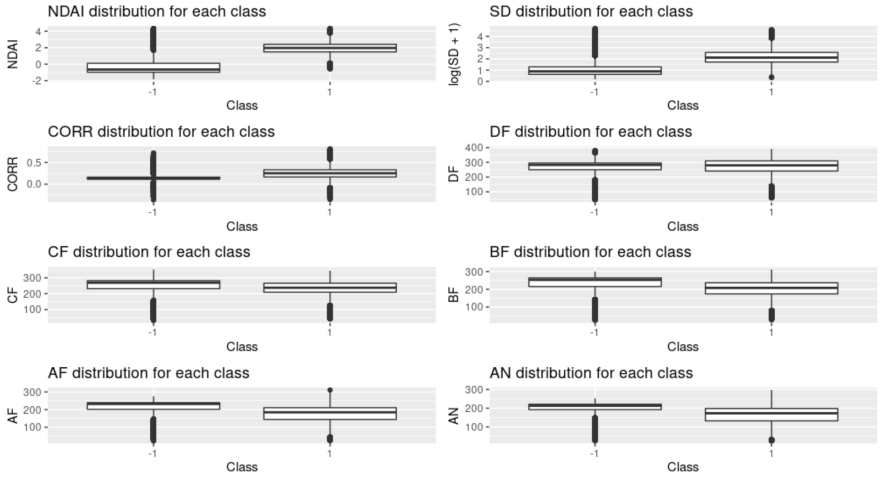


Figure 4: Relationship Between Expert Label with Individual Features

3 Preparation

(a) Method 1: We divided the entire data into 10 blocks based on the range of x-coordinate and y-coordinate. Specifically, we divided x-axis into five blocks and y-axis into two blocks. Then for each observation, we identify the block number it belongs to. We realize the number of observations is uniform across all blocks. Then we decided to use 2 blocks as test set, 2 blocks as validation set and 8 blocks as training set. Specifically, we randomly sample four numbers from 1 to 10, then the first two numbers will be the test block number and the remaining two will be the validation block number. The rest blocks will be the training set.

Method 2: We divided three images respectively. For each image, we divide it into 9 blocks as the first method. Then we will randomly sample two numbers from 1 to 9 for each image: one for test set and the other for validation set. The rest 7 blocks are training set. Then we will combine training set, validation set and test set from three images to form our final sets.

(b) We removed expert label class 0 and only focus on cloud-free and cloud label in method 1. When we set all expert label to be -1, which is the trivial classifier, the accuracy of the trivial classifier on validation set and test set is 0.959 and 0.570 respectively. When the test set is imbalanced, i.e. cloud-free observations takes a large proportion, the accuracy of this trivial classifier on test set will be high.

(c) We utilize single feature classifier to determine the best three features. For each feature, we iterate the range of its values. Then for each specific value, we regard it as the threshold, which means for the observations whose feature value is equal to or greater than it, the predicted label will be 1, otherwise it will be -1. Then for each feature, we will choose the lowest misclassification error and compare with other features'. Our best three features are NDAI, SD, and CORR whose misclassification error is the lowest three.

feature <chr>	error <dbl>
NDAI	0.10
SD	0.16
CORR	0.16
DF	0.32
CF	0.39
BF	0.39
AF	0.39
AN	0.38

8 rows

Figure 5: Single Feature Classifier Results

4 Modeling

Based on previous analysis, we realize these data points are not independent and not linearly separable. Hence we decided to use soft-margin SVM with RBF kernel to classify the data set. This model does not have rigid assumption. Besides that, we also applied Boosting Tree and Random Forest, which both of them do not have strong assumptions. For both logistic regression and boosting tree, they require y is either 0 or 1. Hence we changed "-1" label to "0" label. We think these four models are all satisfied in this case. Besides classification accuracy, we also compute precision, recall and F1 score to compare these methods.

In our first splitting method, we only did model selection for logistic regression since it does not have hyperparameter that needs to be tuned. For all four models, we created 3 folds based on the block(group) numbers. Validation accuracy across folds for Logistic Regression in method 1 are 0.89, 0.86, 0.90, and thus the average validation accuracy is 0.89. For SVM, we applied CVmaster to tune hyperparameter C. Possible values of C are: 0.2, 0.4, 0.6, 0.8 and

1 and the optimal C returned is 1. For Random Forest, we applied CVmaster again to tune hyperparameter mtry. Since there are only 8 predictors, we set the range of mtry between 3 and 7. The optimal mtry is 3. The across folds and average validation accuracy for Random Forest are 1. It is a great classifier. Lastly, we applied Boosting Tree to fit our model with CVmaster to tune n.trees(number of trees) where the possible candidates are 100 to 500 incrementing by 100. The optimal n.trees is 500. Figure 6 shows across folds and average validation accuracy for SVM and boosting tree in method 1 respectively. (***Annotation***: each row represents the index of hyperparameter list, not the values)

In our method 2, except the splitting method we kept everything the same as in the first splitting method, specifically the cross validation is only used for model selection in logistic regression and CVmaster was used to tune hyperparameters in SVM, random forest and boosting tree and possible values for each hyperparameters are the same. Validation accuracy across folds for Logistic Regression in method 2 are 0.855, 0.886, 0.945 and thus the average validation accuracy is 0.895. For random forest, the across folds and average validation accuracy are still 1. Figure 9 shows the related results for SVM and Boosting Tree.

Figure 7 and 10 show the ROC curves for each of the model under different splitting data methods and we marked our cutoff points with the black dot based on validation set. Each point on the ROC curve represents a threshold value(probability) in deciding the predicted labels. We calculated the Euclidean distance between coordinate (0, 1), which represents the perfect classification with 0 false positive rate, and each point on the ROC curve. The point that has the smallest distance is our cutoff point. (***Annotation***: the FPR range for some of the ROC curves are not between 0 and 1 and this may cause visual distortion.)

Figure 8 and 11 show test accuracy, precision, recall, f1_score and ROC cutoff value for each of our models based on these two methods.

In conclusion, under method 1, SVM and Random Forest have better performance than Logistic Regression and Boosting Tree in terms of test accuracy, precision, recall, and f1 score. In contrast, SVM, Random Forest, and Boosting Tree have relative better performance under method 2. In both splitting method, the ROC curves for Random Forest indicate it is a perfect classifier, but the test accuracy is 0.91 on average. We also realized cutoff values vary in different model under different method. Specifically, in method 1, the highest cutoff value is 0.64 from SVM while in method 2, the highest cutoff value is 0.43 from Boosted Tree. We think the interpretability is important in this case. Hence we choose Random Forest under method 1 as our best classification model by comparing its metrics between two methods.

	1	2	3	Average
[1,]	0.90	0.94	0.99	0.94
[2,]	0.91	0.94	0.99	0.95
[3,]	0.91	0.95	0.99	0.95
[4,]	0.91	0.95	0.99	0.95
[5,]	0.91	0.95	0.99	0.95

(a) SVM

	1	2	3	Average
[1,]	0.88	0.94	0.92	0.91
[2,]	0.89	0.94	0.93	0.92
[3,]	0.90	0.94	0.94	0.93
[4,]	0.90	0.94	0.94	0.93
[5,]	0.91	0.94	0.94	0.93

(b) Boosting Tree

Figure 6: Validation Accuracy in Method 1

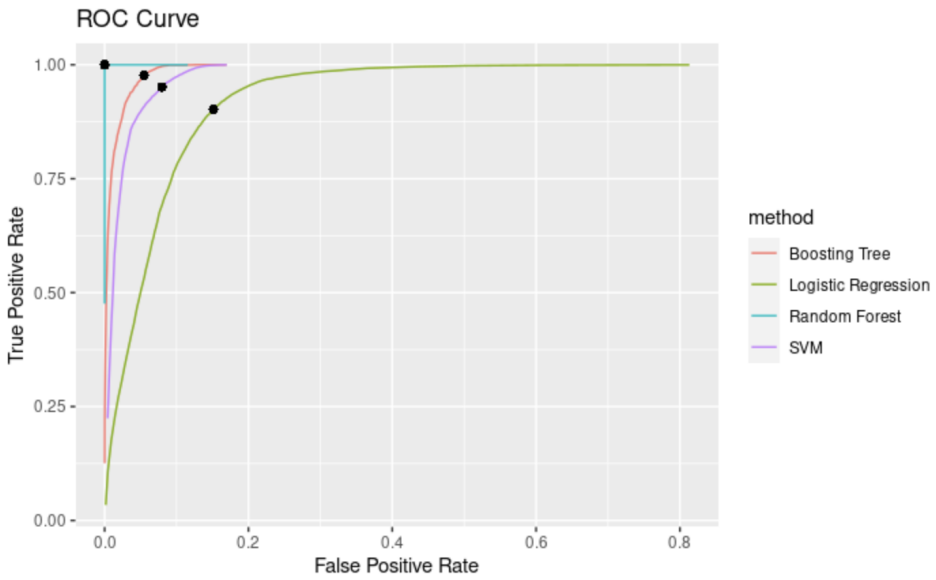


Figure 7: ROC curves in Method 1

Method <chr>	Test_accuracy <dbl>	Precision <dbl>	Recall <dbl>	F1_score <dbl>	Cutoff Value <dbl>
Logistic Regression	0.82	0.76	0.90	0.82	0.35
SVM	0.92	0.89	0.95	0.92	0.64
Random Forest	0.90	0.86	0.95	0.90	0.43
Boosted Tree	0.87	0.81	0.96	0.88	0.35

4 rows

Figure 8: Model Evaluation Metrics Report in Method 1

	1	2	3	Average
[1,]	0.94	0.97	0.92	0.95
[2,]	0.95	0.97	0.93	0.95
[3,]	0.95	0.97	0.93	0.95
[4,]	0.95	0.97	0.93	0.95
[5,]	0.95	0.97	0.93	0.95

(a) SVM

	1	2	3	Average
[1,]	0.92	0.88	0.97	0.92
[2,]	0.93	0.90	0.96	0.93
[3,]	0.94	0.91	0.96	0.94
[4,]	0.94	0.91	0.96	0.94
[5,]	0.94	0.92	0.96	0.94

(b) Boosting Tree

Figure 9: Validation Accuracy in Method 2

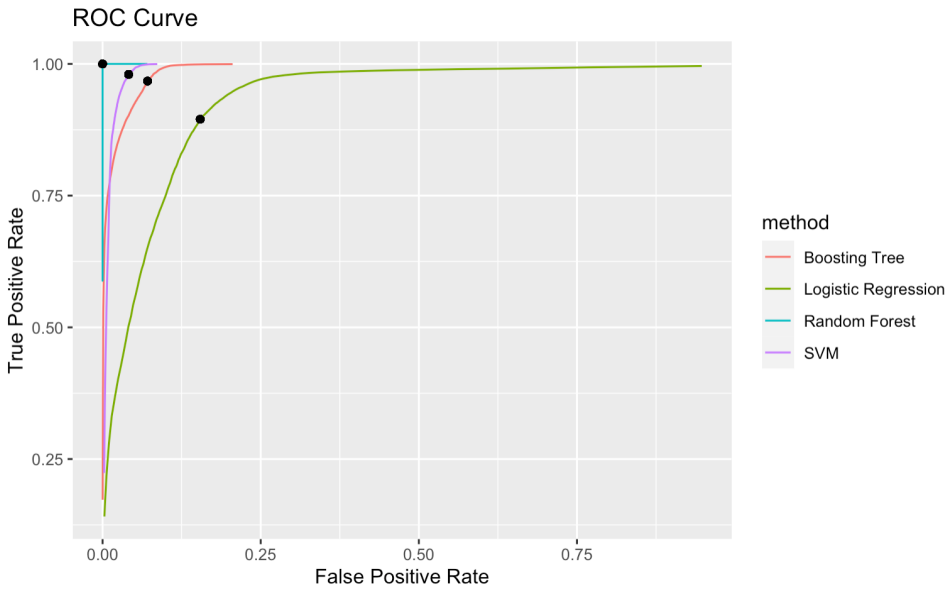


Figure 10: ROC curves in Method 2

Method <chr>	Test_accuracy <dbl>	Precision <dbl>	Recall <dbl>	F1_score <dbl>	Cutoff Value <dbl>
Logistic Regression	0.898	0.794	0.794	0.794	0.42
SVM	0.908	0.938	0.940	0.939	0.28
Random Forest	0.923	0.832	0.862	0.847	0.17
Boosted Tree	0.927	0.823	0.901	0.860	0.43

Figure 11: Model Evaluation Metrics Report in Method 2

5 Diagnostics

(a) We generate an importance plot from Random Forest model as Figure 12 and plot the trend of validation error changes as number of trees increases as Figure 13. Based on Mean Decrease Accuracy, the 3 most important variables are NDAI, AN, and CORR. But based on Mean Decrease Gini, the 3 most important variables are NDAI, SD and CORR. Even though the order of variables changes in those two plots, NDAI is the most important variable. Figure 13 shows that the lowest validation error is achieved when the number of trees is approximately 2100. After that the validation error begins to increase. We think this indicates the overfitting happens and introducing regularization technique may prevent it.

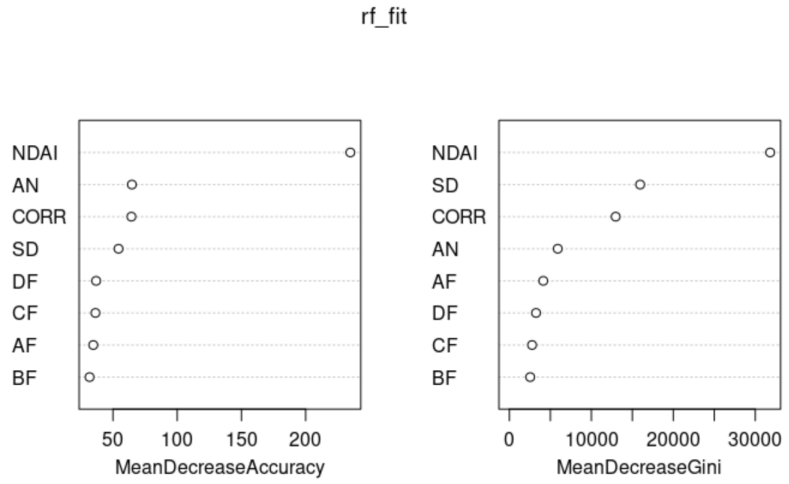


Figure 12: Importance Plot

Figure 13: RF Validation Error Changes with n.trees

(b) Figure 14 shows the differences between the predicted labels and the expert labels on test set. The difference in classification is obvious, especially on some regions where x-coordinate is greater than 187.5 and y-coordinate is between 0 and 60. In this region, random forest misclassified the cloud-free label (class -1) to cloud label(class 1). We also noticed in the region where x-coordinate is between 125 to 187.5 and y-coordinates lies between 250 to 300, our model misclassified class 1 label to class -1 label. For features, we focus on NDAI, SD, CORR and AN since these four are the most important predictors based on our previous analysis. We divided the ranges of these four features on test set into three equal-length intervals respectively. Then for each feature and each interval, we calculated the misclassification rate based on the expert label and predicted label. As shown in the below table, each entry is the misclassification rate for each feature in specific interval. For NDAI, interval 3, which has range 2.41 to 4.34, has the largest missclassification error of 0.35. For SD, the lowest misclassification error happens

on interval 1 (0.36, 37.06) and the highest misclassification error of 0.74 is on interval 3 (73.77, 110.47). The rest two predictors, CORR and AN, the misclassification error is relative uniform as misclassification errors do not vary a lot between different intervals.

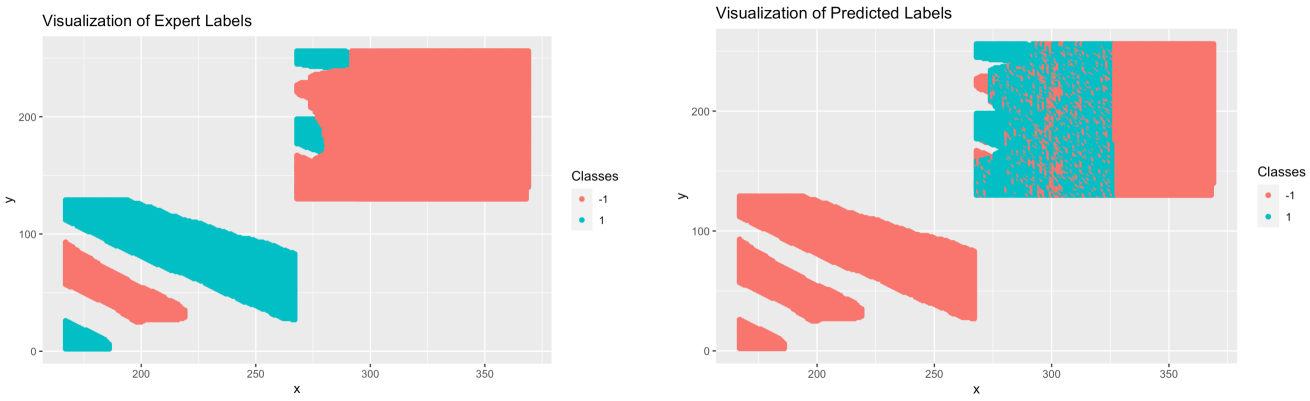
Feature	Interval_1	Interval_2	Interval_3
NDAI	0.014	0.1	0.35
SD	0.092	0.44	0.74
CORR	0.27	0.092	0.09
AN	0.18	0.18	0.05

(a) Expert Label (b) Predicted Label

Figure 14: Visualization of labels

(c) As we mentioned in problem 4.a, our model may overfit the data and have a weak generalization ability. Hence we believe the regularization technique is necessary. We want to apply early stopping during the training process. Specifically, we will monitor the whole training process and the process will stop once the validation error started to increase. In this way we are able to avoid a highly complicated model by preventing endlessly adding trees. We think this method would improve our classification a little better for future data set since the model we train will not running into a overfitting problem so that it can be more robust and have a stronger generalization ability.

(d) We plot the visualization of predicted labels , generated from random forest, and expert labels in test data created by our second method of splitting data as Figure 15. It is easy to see that the patterns in misclassification errors are totally different comparing to the first method. Therefore, if the way to split the data is more random, the test accuracy may actually goes down. As for overfitting, this is a general phenomenon for complicated model no matter how you actually split the data. Therefore, we might also expect overfitting in our second splitting method if the number of trees increases endlessly. In conclusion, the way you split the data may affect 4(b) but may not affect 4(a).



(a) Expert Label (b) Predicted Label

Figure 15: Visualization of labels

(e) Random forest is not guaranteed the best model, i.e. sometimes it's the best while it may not be the case in other times. However, in general, when the number of trees grows too large, the model will running into overfitting problems, and this is irrelevant to how you split the data. When we encounter the overfitting problem, one solution is to implement a regularization method so that it will put constraint on number of tree. One example of regularization can be applying an early stopping method in the training process so that the number of tree will not keep growing once the training error starts to increase. In contrast, different way of splitting the data will affect the misclassification error region. In our above example, those error regions varies a lot in method 1 and method 2. To conclude, overfitting is a general problem for classification model which does not depend on the way of splitting the data, but in contrast, the misclassification error region is determined by the splitting method.

References

[1] Tao Shi, Bin Yu, Eugene E Clothiaux, and Amy J Braverman. Daytime arctic cloud detection based on multi-angle satellite data with case studies. *Journal of the American Statistical Association*, 2008.