

Lab 4-Binary Classifier

Stat 215A, Fall 2014

Xiang (Lisha) Li, Jonathan Fischer, Andrew Do, Hye Soo Choi

November 10, 2014

1 Introduction

Blah blah...

2 EDA

2.1 Plots of Raw and Expertly Labelled Images

Figure 1 displays the unprocessed image files for comparison with Figure 2, the expertly-labelled files. With the human eye it is not so difficult to parse cloud from ground based on these images, but we see that cloudy and clear pixels alike run through wide spans of AN so other features must be used.

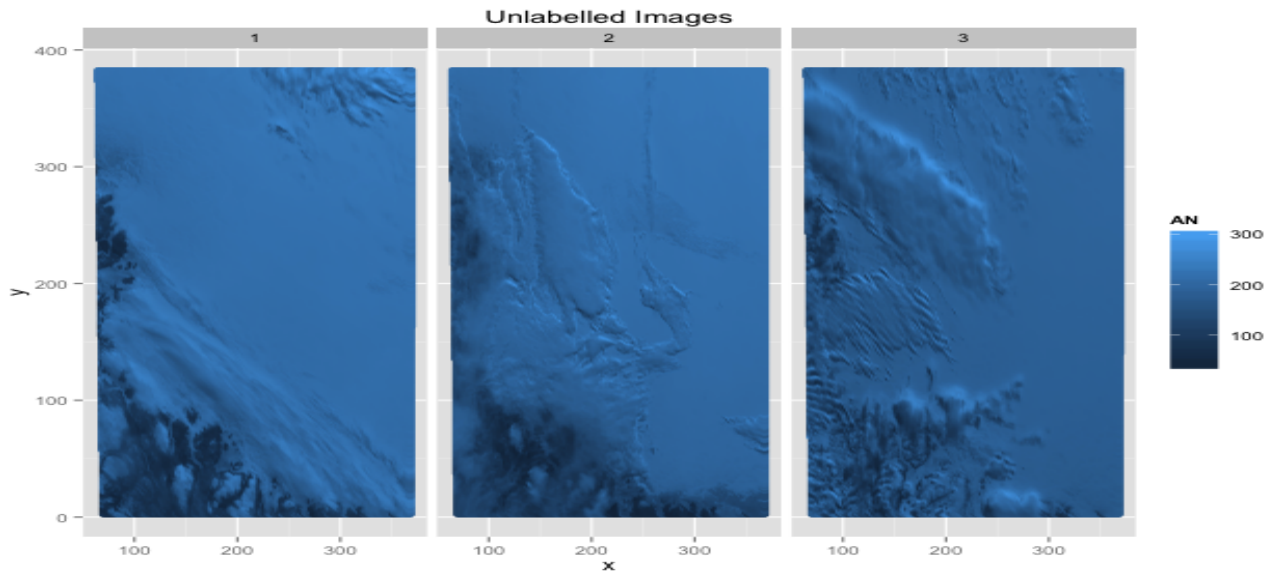


Figure 1: Raw images with AN radiances.

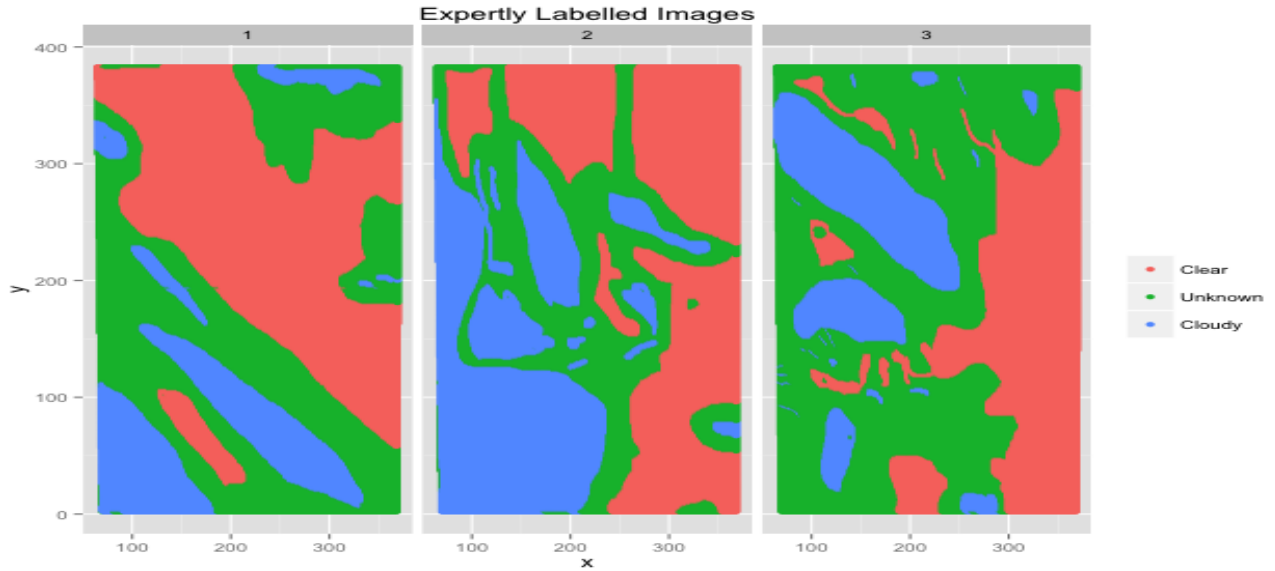


Figure 2: Images with expert classifications. Proportions: 39.8% unknown, 23.4% cloudy, 36.8% clear.

2.2 Densities of NDAI, SD and CORR

The following three plots gives us a sense of what can be learned from NDAI, SD and CORR. The densities are grouped by their expert labels, red is for ‘no cloud’, green are for ‘unknown’ and blue for ‘cloud’. We can see that NDAI has reasonably good separation between cloud and no cloud, in all three pictures, which is confirmed in our later modeling sections by Gini importance measures found in the random forest model and it’s importance as a variable in LDA/QDA and logit models. In comparison, SD does not have as good of a separation within the smaller SD values, however it is still clear that pixels labelled as clouds are the only pixels with higher SD values. We thus expect the two features in combination can help determine whether a pixel with high NDAI should be labelled as a cloud by using the SD feature. Finally CORR values appear to be a good separator for image 2, but much less so for image 1. This uneven distribution of CORR values between images prompted us to cross validate our models across images, in addition to the folds created by dividing each image into 4 quadrants.

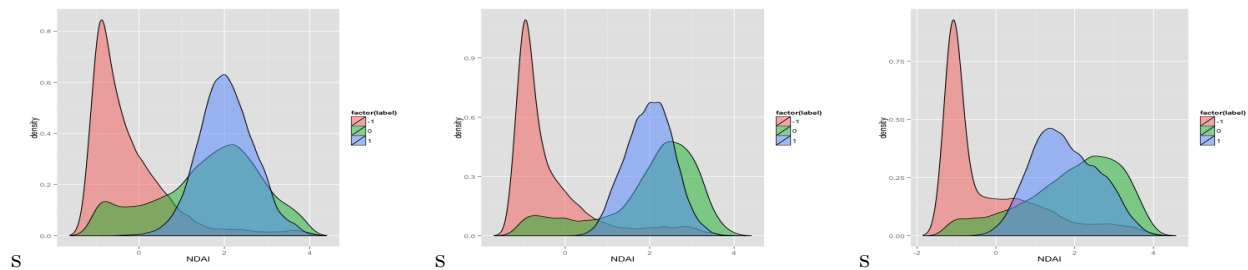


Figure 3: NDAI density plot for Image 1, 2, 3 (respectively).

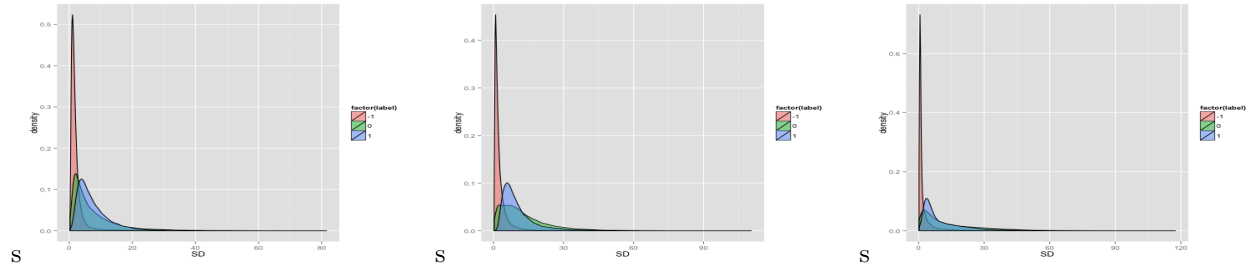


Figure 4: SD density plot for Image 1, 2, 3 (respectively).

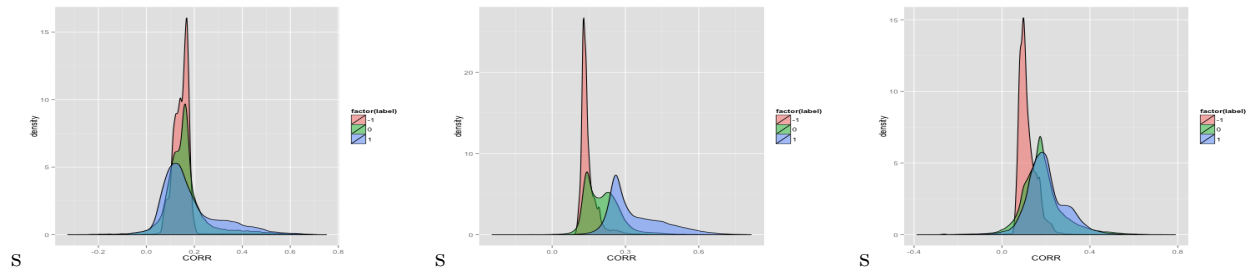


Figure 5: CORR density plot for Image 1, 2, 3 (respectively).

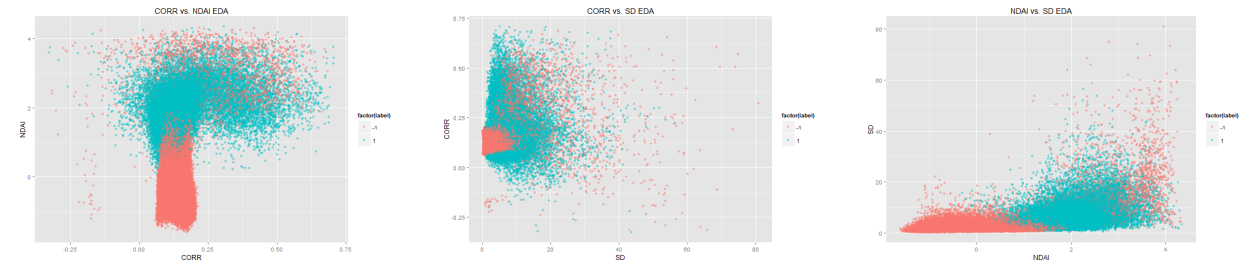


Figure 6: CORR vs. NDAI Plot of Image 1

Figure 7: CORR vs. SD Plot of Image 1

Figure 8: NDAI vs. SD Plot of Image 1

2.3 Mapped Features

The ensuing figures plot the engineered feature values (NDAI, SD, CORR) spatially. In agreement with the NDAI density plots, higher NDAI values indicate increased likelihood of the presence of clouds, and the NDAI plots look quite similar to the binary classification plots. The SD maps show areas of high variance in the radiances thereby providing a decent outline of cloud boundaries. Unfortunately, this can also lead to the highlighting of uneven ground regions. Finally, the CORR images resemble weakened versions of their NDAI counterparts though with some additional strange behavior in the bottom left corners of Images 1 and 2.

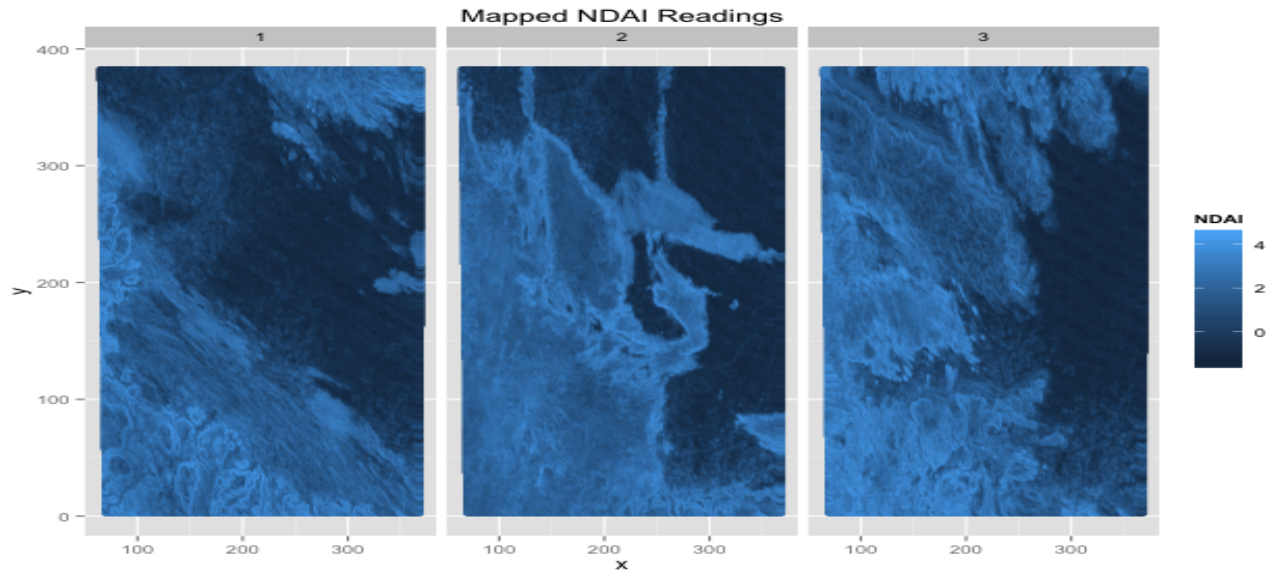


Figure 9: Mapped NDAI readings. We see good correspondence between larger values and presence of clouds.

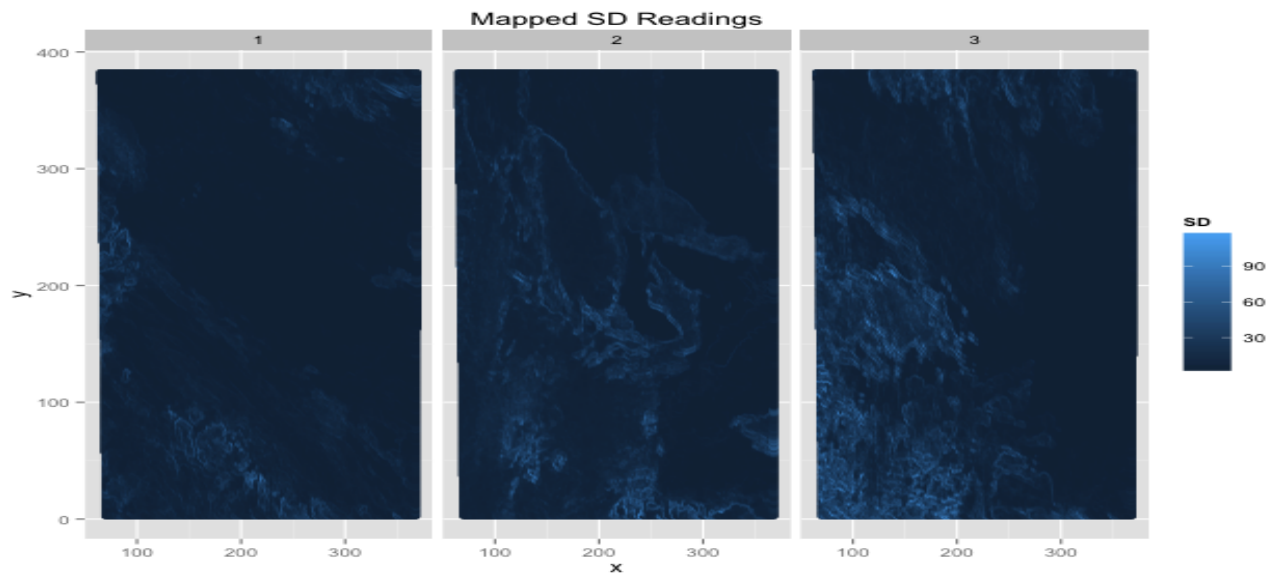


Figure 10: Mapped SD readings. Higher values show cloud boundaries, though also show uneven terrain.

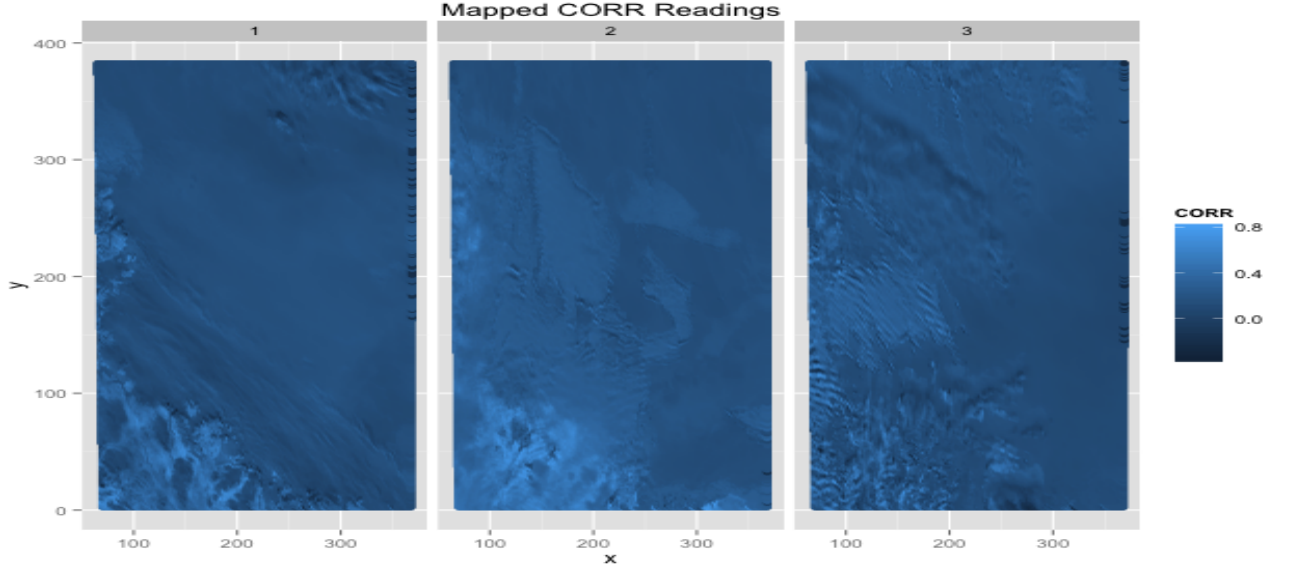


Figure 11: Mapped CORR readings. Cloudy regions tend to be lighter, but not as strongly as in NDAI.

3 Modeling

3.1 LDA

3.2 QDA

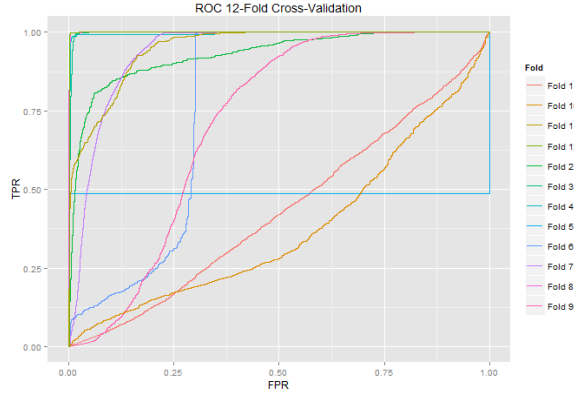


Figure 12: Response Operator Curve for 12-fold Cross-validation of QDA

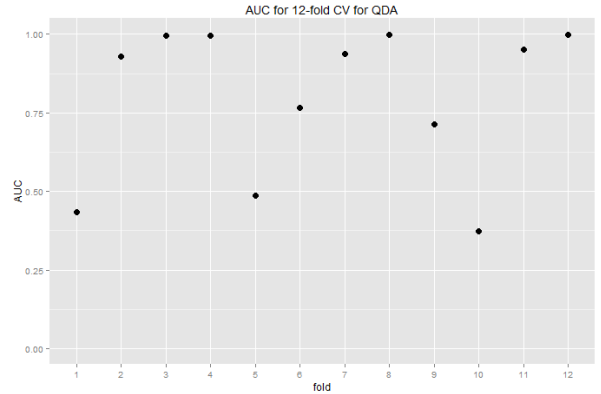


Figure 13: AUC for QDA Classifiers

Cross-validation for QDA revealed that while the method works extremely well in some cases, producing AUC scores of almost 1, it sometimes fails to perform better than even the theoretical random classifier. In particular, the classifier does not seem to be good at discerning snow from clouds in regions where there are many dark pixels. For example, during cross-validation, the watery bottom-left quadrant of image 2 and ridge-ridden left edge of image 3 poses significant problems to our classifiers.

3.3 Logit/Probit

The logit model obtained via averaging after 12-fold cross-validation has $y_i = -3.356 + 1.900 * NDAI_i - 0.074 * SD_i + 9.002 * CORR_i$ where $P_i(\text{Cloud}) = \frac{1}{1+e^{-y_i}}$.

To choose our cutoff value for declaring a pixel cloudy, we varied the cutoffs from .01 to 1 and calculated the misclassification error of our model at that threshold on expertly-labelled pixels. The figure below shows that the optimal threshold for this model is at .38, with an error of just under 10.1%. As a consequence of this low threshold, we observe more false positives than false negatives. This will be further explored below.

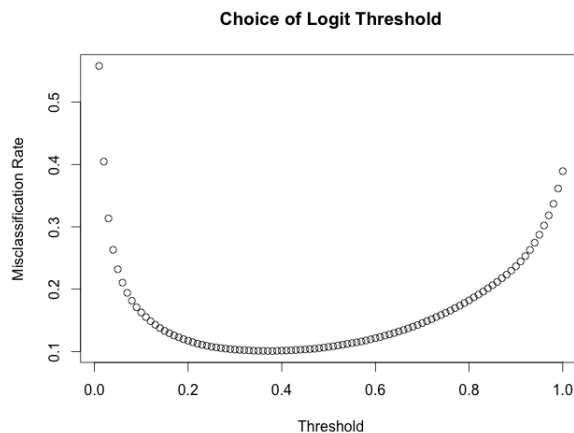


Figure 14: We see a minimum in misclassification rate at .38.

Figure 15 gives the binary classifications overlaid on the images. Visual comparison with the expertly-labelled images suggests that we are slightly biased towards the presence of clouds as 74.2% of unlabelled points have been classified as cloudy, though this may be due to some artifact of the expert labelling scheme.

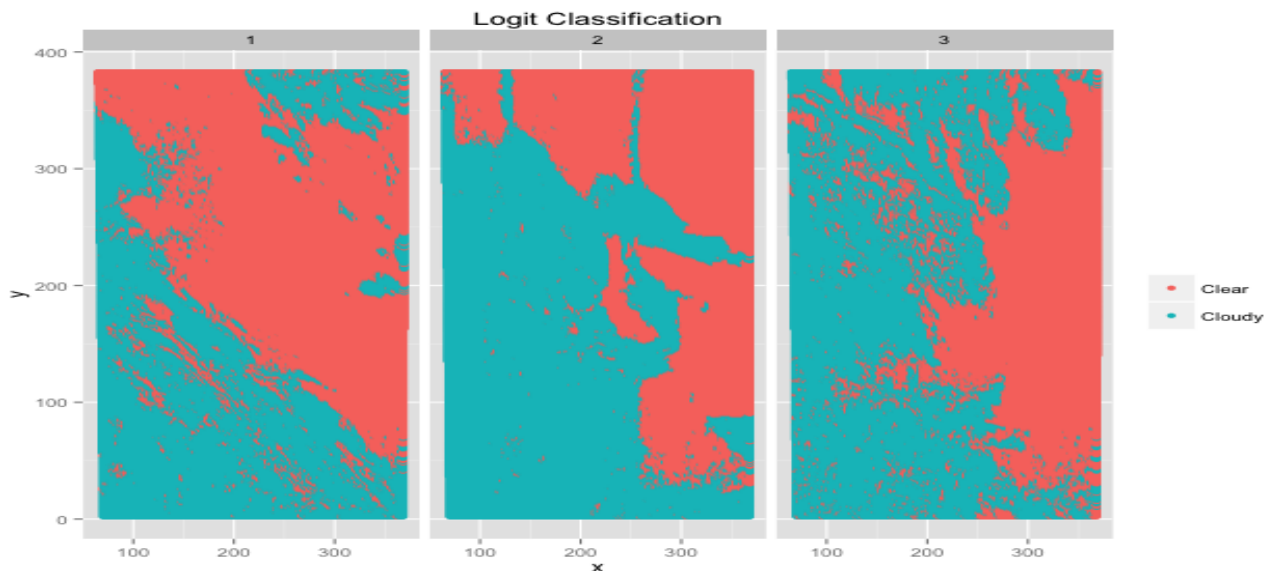


Figure 15: Binary classifications with threshold = .38 for logit trained via 12-fold CV.

Here we see the logit probabilities plotted spatially. This should approximate how the image would appear if the ground were not covered in snow and ice, so comparison with Figure 1 is especially appropriate.

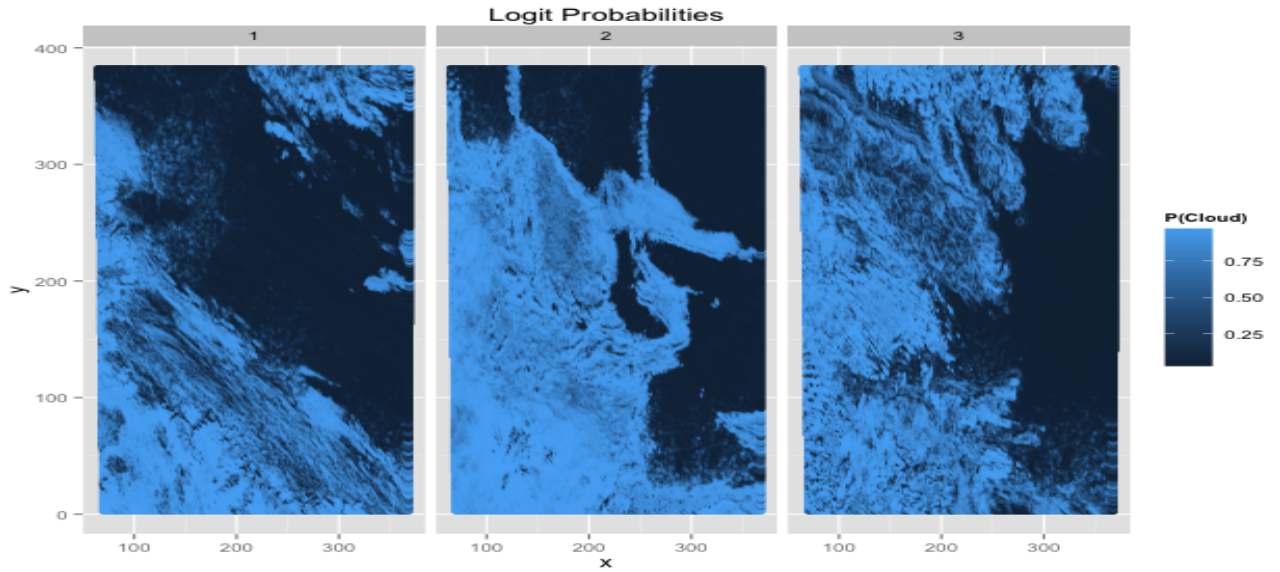


Figure 16: Probabilities for logit trained via 12-fold CV.

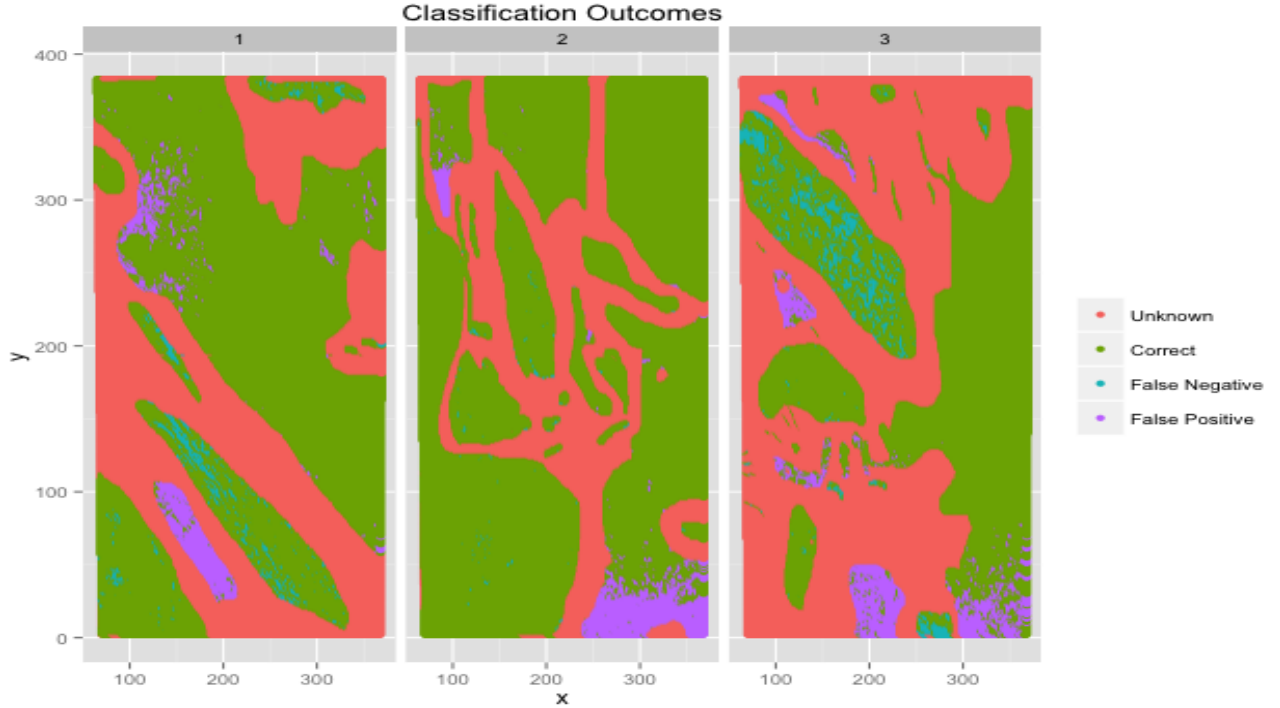


Figure 17: Classification results from logit with respect to the expert labels.

We now discuss the trends in misclassification. As depicted in the above figure, we have many more false

positives than false negatives with large regions sometimes completely misclassified. These regions are often situated on boundaries of labelled and unlabelled points. Calculation shows the false negative rate to be 7.5% and the false positive rate to be 11.7%.

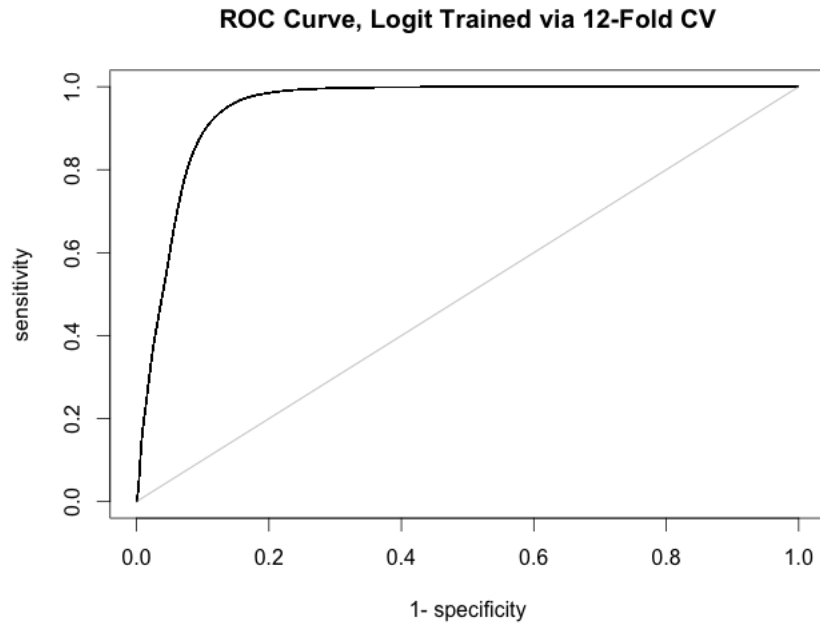


Figure 18: ROC curve for our logit model. AUC is .95.

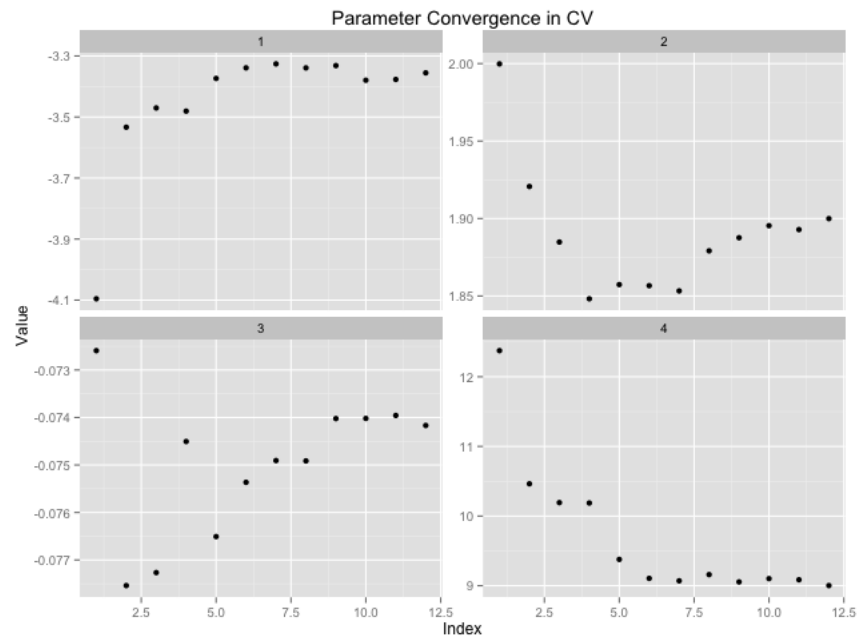


Figure 19: Coefficient values through iterations of 12-fold CV.

3.4 Random Forest

For random forest, as in all the other classifiers, we divided the three images into equal sized quadrants (2X2) rectangles in order to do 12 fold validation on the dataset. That is, for each iteration of the validation, we dropped one of the quadrants as a test set, and trained on the remaining 11 quadrants. Keeping the test segments as disjoint chunks from the test set ensured that our models were picking up on ‘higher’ level structure of the dataset, and not the continuous variation of neighbouring pixels. As such, we also expect the performance of the model on these test sets to be a much better predictor of how our model will perform on new images than if we randomly subsampled the validation folds. Indeed, after doing random subsampling, we noticed that the AUC was at a high 0.999 for all cross validations, whereas by cross validating on the isolated quadrants, our AUC curves had much higher variance, as seen below:

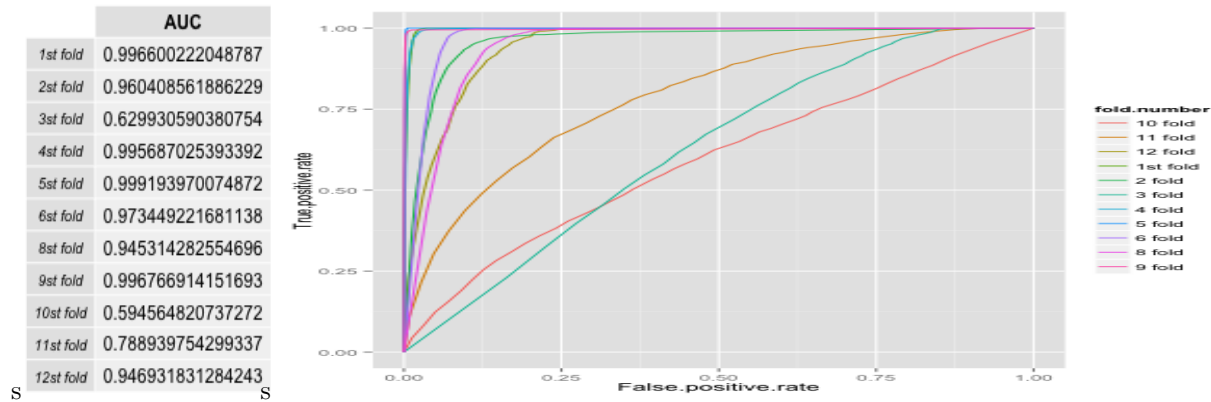


Figure 20: ROC across all 12 folds of quadrants

We also trained on each image and tested on the remaining two.

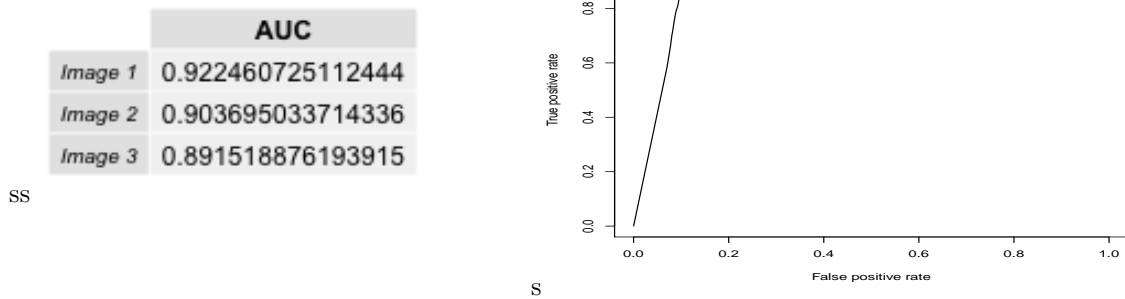


Figure 21: Convergence of ROC training on increasing number of quadrants in shuffled order

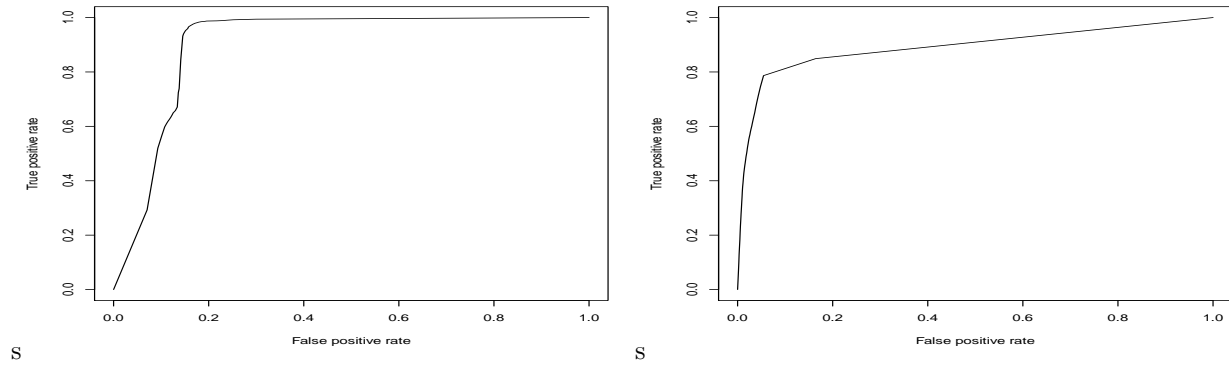
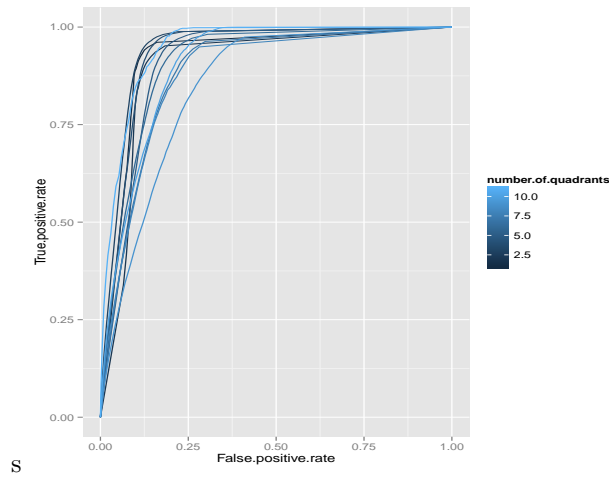
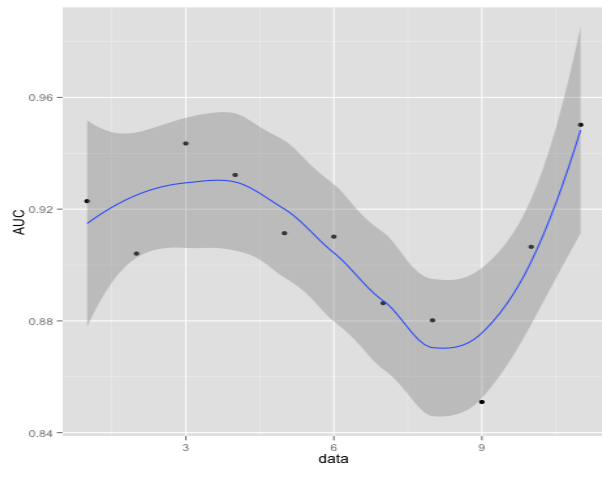


Figure 22: Convergence of ROC training on increasing number of quadrants in shuffled order

To test convergence, one of the things we did was increase the training set from including 1 quadrant, to including 2 quadrants, up to including 11 quadrants (using the complement as the test set). For each of the 3 aforementioned classes of training, we trained on a range of forest sizes, from 2 trees to 50. It was interesting to note that the continuity of increasing the training set by neighbouring quadrants displayed much less convergence than by increasing the training quadrant set in a randomly subsampled way. The following three plots make this clear. In the first case....



S



S

Figure 23: Convergence of ROC training on increasing number of quadrants

Figure 24: Smoothed convergence of AUC for growing training set 50 trees and 3 features

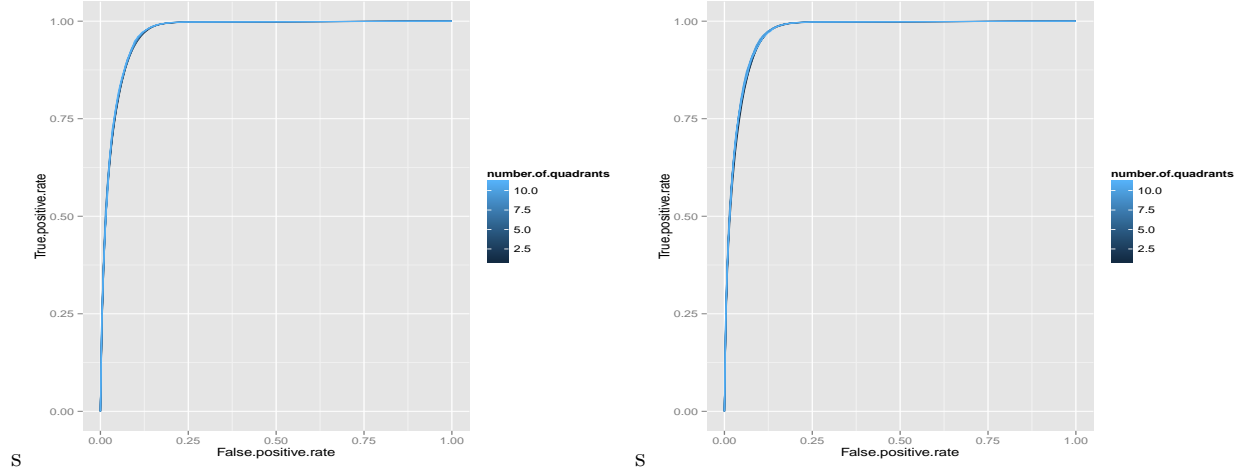


Figure 25: Convergence of ROC training on increasing number of quadrants in shuffled order

To gather a quantitative measurement of the importance of each feature, we first ran random forest using all the features and looked at the Gini importance measure. This is calculated by recording the difference between the Gini measure of a random forest's predictions on a fold and the Gini measure with a particular feature's values randomly shuffled. The intuition is that if a feature were crucial to the forest's trees, then randomly shuffling that feature will drastically decrease its Gini measure. Sure enough, NDAI, SD and CORR consistently ranked as the clear top three in all cross validations.

Where was the model performing poorly?

4 Reproducibility

How we organized out code and github repo

References

- [1] Ben-Hur, A., Elisseeff, A., Guyon, I.: A stability based method for discovering structure in clustered data. In: Pacific Symposium on Biocomputing, pp. 617.

	mean	variance	sd
NDAI	31851.137865015	5541519.63141267	2354.04325181435
SD	15181.3036164789	7250037.34195744	2692.58933778574
CORR	16696.8895555848	15065266.3880018	3881.40005513498
DF	5153.57742916125	715182.664503553	845.684731151954
CF	3725.98512846366	238611.088513114	488.478339860749
BF	4741.67090014784	908691.31228918	953.253015882551
AF	6307.45653651382	1489619.01113017	1220.49949247436
AN	7656.24468437671	5730133.08077185	2393.76963820077

Figure 26: Mean Gini Importance Measure across 12 folds