Analysing Cloud Data to asses pollution impact

[redacted], [redacted] Malvika Rajeev

Bold is used when defining a term, and *emphasis* is used for drawing attention to phrase(s) or word(s). Hyperlinks are in blue. For any number n, denote the set $\{1, 2, ..., n\}$ by $[n]$.

1 Domain problem and reflections on Shi et. al.

The reference paper begins by noting the importance of global climate change in this century, and observing that atmospheric carbon dioxide concentration is increasing everywhere resulting in worldwide rise of mean annual temperature. The climate models used to predict this change have also *consistently* mentioned the Arctic region as most sensitive to its effects. While the rise of CO_2 levels is generally agreed, expert opinion seems divided on how exactly this will change local climates around the world (including in Arctic). Of the many factors affecting this change, effects of clouds appear to the foremost area where we lack clear understanding. To gain insights, a lot of high-quality cloud data needs to be gathered and analysed. This paper aspires to develop an accurate and computationally efficient algorithm that can detect clouds from MISR imaging by satellites, and analyses its results by comparing with expert-labeled data in the Arctic. Given the recent concerning developments in the global effort to fight this, we are glad that we could work on a climate change data set in this course!

The Arctic region is special; it doesn't have a lot of variety in topography: only glaciers, snow-covered coastal plains and oceans, relative to other regions of the world. Its only mountain range is concentrated in a very small area near northern Canada. Furthermore, the problem of distinguishing clouds from non-snow terrain is very easy. In fact, authors have actively removed data points where the terrain didn't have snow in their experiments. Hence, the only challenge is to distinguish (mostly flat and clear) snow terrain from cloud formations.

The fundamental proposition is that to distinguish clear snow *of Arctic* from clouds, a **Fisherian null hypothesis** of "this pixel corresponds to clear snowy terrain" can be very reliably tested. This is because while clouds can be of different types with varied radiation signature, we *know* that clear snow terrain have uniform, isotropic radiation signature. Thus, if we can confidently say that the pixel has isotropic signature, we can reliably infer that it is snow and not clouds. The paper does a great job in explaining the MISR technology and the specific manner in which the data set was collected. The arguments for developing domain-specific features from raw radiation readings are also logical. In particular, the authors have clearly explained why and how they came up with CORR, SD and NDAI, and show that their features have **stable distribution over time** for each location, and are **clearly separable** according to expert labels. However, there are a few things we would have liked to see:

- Firstly, what is the impact of Arctic haze, a collected pool of smog and pollutants persistent over the Arctic, on these radiation measurements? We would expect a pronounced effect of this on the data as its concentration is known to vary with time of year. Unfortunately, the authors have completely ignored this aspect. Similarly, a few lines on the possible effects of Ozone depletion, which also concentrates over the poles, could be useful.
- Secondly, why was red radiation in 275-m resolution and the other three bands in 1.1-km resolution? As described in Sec. 1, the satellite sensors collected 275-m resolution data for all 4 bands (red, green blue, near-infrared). Due to bandwidth constraints, the satellite could only transmit a single band in 275-m resolution, and would aggregate the the rest in 1.1-km resolution. Note that ice and snow

radiation signatures are the same for all 4 bands (as mentioned in Sec. 3.1). Then why choose red over the other three? This choice seems *arbitrary* and imposes an unfair dependence of our algorithm on the red measurements.

- And finally, the figures could be improved a lot. In particular, in Figs. 2 and 5 the satellite appears more like a flying pistol! However, the plots are great and convey the required information well.

2 Data Exploration

We begin by loading the three images from `image1.txt`, `image2.txt` and `image3.txt` provided, and find that there are no missing entries. All entries of column 3 ("label") are from $\{+1, 0, -1\}$, and all entries of correlations CORR (of column 6) have absolute values in $[0, 1]$ as expected. Hence it seems the 3 images already have clean and correct entries in the files provided.

`image1` has 115229 pixels of which 61.5% are labelled (as either "cloud" or "non-cloud"). `image2` has 115110 pixels of which 71.4% are labelled. `image3` has 115217 pixels of which 47.7% are labelled. Therefore, the data set has a spectrum of expert label availability: `image2` has most pixels labelled whereas `image3` has only a few labelled. *We will test our models on each of the 3 images to see the impact of label availability on performance.*

2.1 Plot the expert labels for the presence or absence of clouds, according to a map (i.e. use the X, Y coordinates).

2.2 Explore the relationships between the radiances of different angles, both visually and quantitatively. Do you notice differences between the two classes (cloud, no cloud) based on the radiances? Are there differences based on the features (CORR, NDAI, SD)?

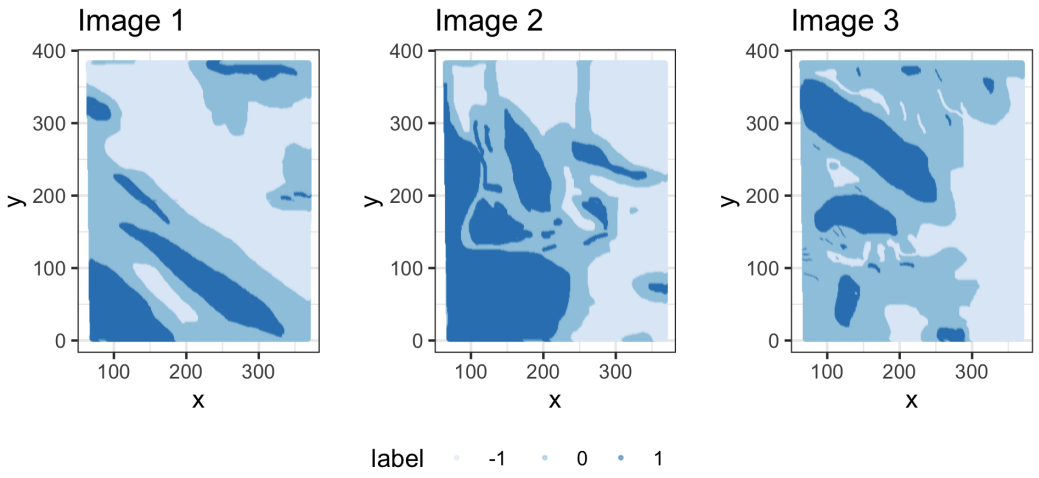

Figure 1: Plot of expert labels by x and y coordinates for all three images.

In Figure 1, we can infer that dark blue is cloud and light blue is no cloud, and middle blue is unknown. Plotted, this way, the image do appear as one would expect given the data and question.

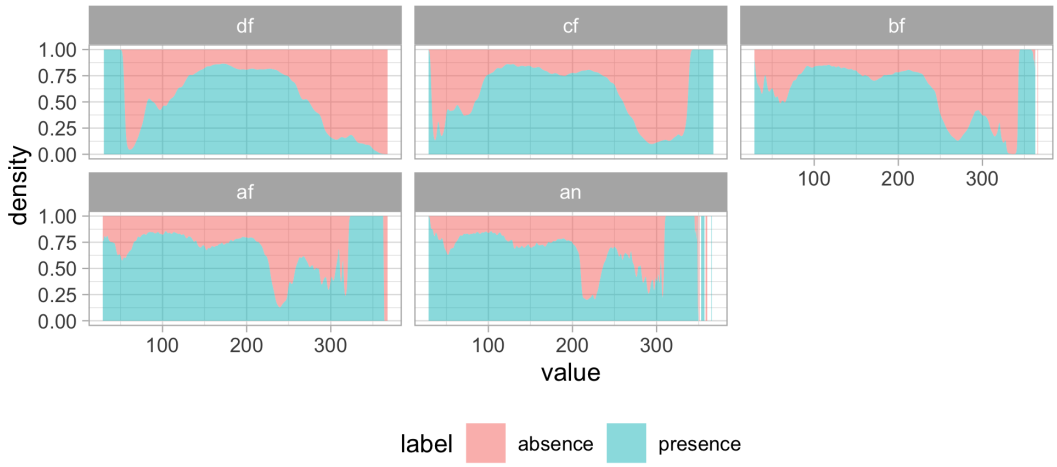

Figure 2: Conditional density of each radiance measure colored by expert labels.

In figure 2, we can see from the conditional densities of each of the radiances that presence/absence of clouds have different density. And further in figure 3 we that conditional SD densities separate presence/absence of clouds quite well in what looks like a linear-ish relationship. We do see some signal between presence/absence of clouds and NDAI and corr, but the nature of that relationship is not obvious from the plot.

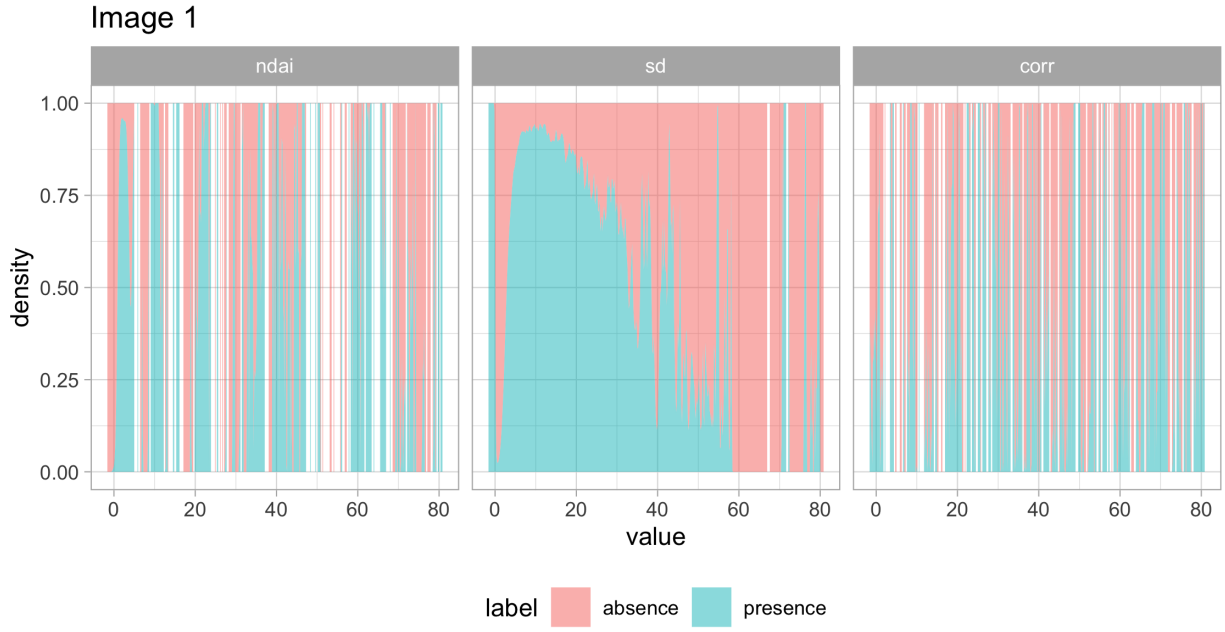

Figure 3: Conditional density of each radiance measure colored by expert labels.

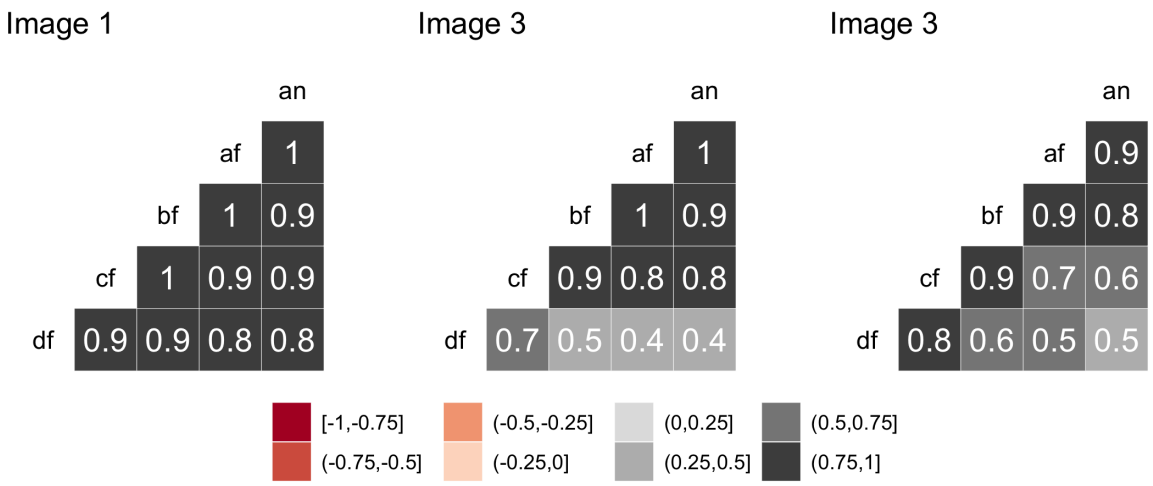

Figure 4: Corrlation plot between radiances for all three images.

The radiances themselves are highly correlated (Figure 4). There is some variance in the nature of that correlation between the images, but in general that are highly correlated.

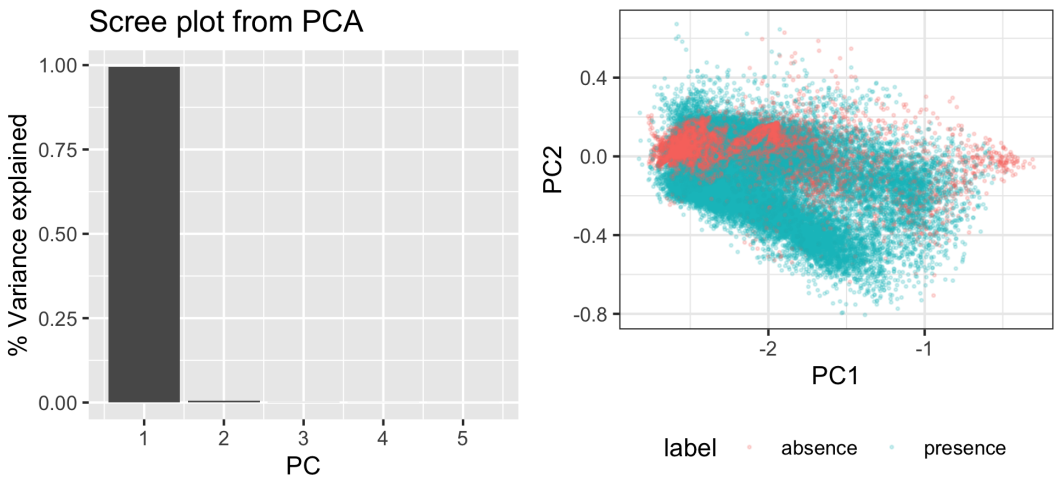

Figure 5: Corrlation plot between radiances for all three images.

We performed PCA on radiances for all three images combined, and the plotted the resulting PCs colored by the expert labels; radiances were scaled, but not centered (Figure 5). The scree plot shows that the first component explains almost all the variance, which makes sense given that the radiances are highly correlated. From the plot of the PCs we that presence or absence of clouds does appear to be correlated with the first PC, but we do not see strong clusters. Here, it seems, we see that PCA captures some signal, but that signal may not be best represented by a linear combination of the features, which PCA does best.

After the conditional density plots of the radiances: The values for the angles **af and an** behave very similar, as do the values of **df and cf** as evident in the conditional density plots. This explains that why when we conducted a PCA, we were able to explain approximately 99% of the variation in the radiances values. Individually, the values of radiances seem highly separable for the different expert labels. According to the paper, the different features of NDAI, SD and CORR have been derived using the values of these radiances. When we plot the conditional densities of the features against the classes, we notice that the

values of SD show a very evident linear trend. The values of CORR aren't very easy to separate but there does seem to be some concentration of the classes at the top and the bottom. The values for NDAI, however, are unable to show a discernible pattern.

3 Modeling

While the reference paper clearly shows that ELCM (along with a QDA add-on) performs better than SVM and conventional MISR algorithms, we believe there is still scope for improvement. In particular, as said in last para of Sec. 4.1, ELCM does poorly where there is a coastline or where the terrain information is outdated. While expert labels correctly reflect the true terrain, outdated maps result in an (artificial) disagreement for the ELCM prediction. We believe modern methods, such as a multi-layer neural network, can be used to harness this extra information and construct a better model. This should still be efficient *on today's computational resources*, and the data set (of $\approx$ 0.3 million rows) is not very large to be an obstacle for the same.

3.1 Model 1

For the first model, we used the extreme gradient boosting (xbg) algorithm to predict the expert labels. The `caret` package was be used implement `xgbLinear` model. The XGB algorithm builds on the boosting concept and adds regularization to prevent over-fitting of the trees. xgbLinear builds generalized linear models and optimizes them using L1 and L2 regularization and gradient descent. Subsequent models are built using residuals from previous iterations.

The `xgbLinear` method has four hyperparamters:

- nrounds: The number of boosting rounds
- λ: L2 regularization parameter, range $[0, 1]$
- α: L1 regularization parameter, range $[0, 1]$
- η: Step size shrinkage used in update, range $[0, 1]$

These hyperparamters will be selected by 10-fold cross-validation across a grid of the following values for each Hyperparameter:

- nrounds: 50, 100
- $\lambda \in (0, 0.25, 0.5, 1)$
- $\alpha \in 0$
- $\eta \in c(0.1, 0.2, 0.3)$

The hyper-parameters with best cross-validated accuracy are selected as the "best" and used to predict the absence or presence of clouds in each of the three images. L1 parameter was set to be zero because we already have a small number of parameter and there is no need to identify a sparse model. A larger hyper-parameter search was done previously, but for the sake of run time for the .Rmd file, a small hyper parameter tuning grid is used here.

Table 1: CV xgbLinear model prediction results

Image	Nrounds	lambda	alpha	eta	Accuracy	AUC
Image 1	50	0.50	0	0.1	0.95	0.94
Image 2	50	1.00	0	0.1	0.97	0.97
Image 3	50	0.75	0	0.1	0.90	0.89

We can see in table 1, that the xgb model performs fairly well, predicting with >90% accuracy in all three images. The algorithm performs best on model 2 and worst on model 3.

3.2 Model 2 - Random Forest

Random Forest has no model underneath, and the only assumption that it relies on is that sampling is representative. By principle since it randomizes the variable selection during each tree split it's not prone to over fit unlike other models.

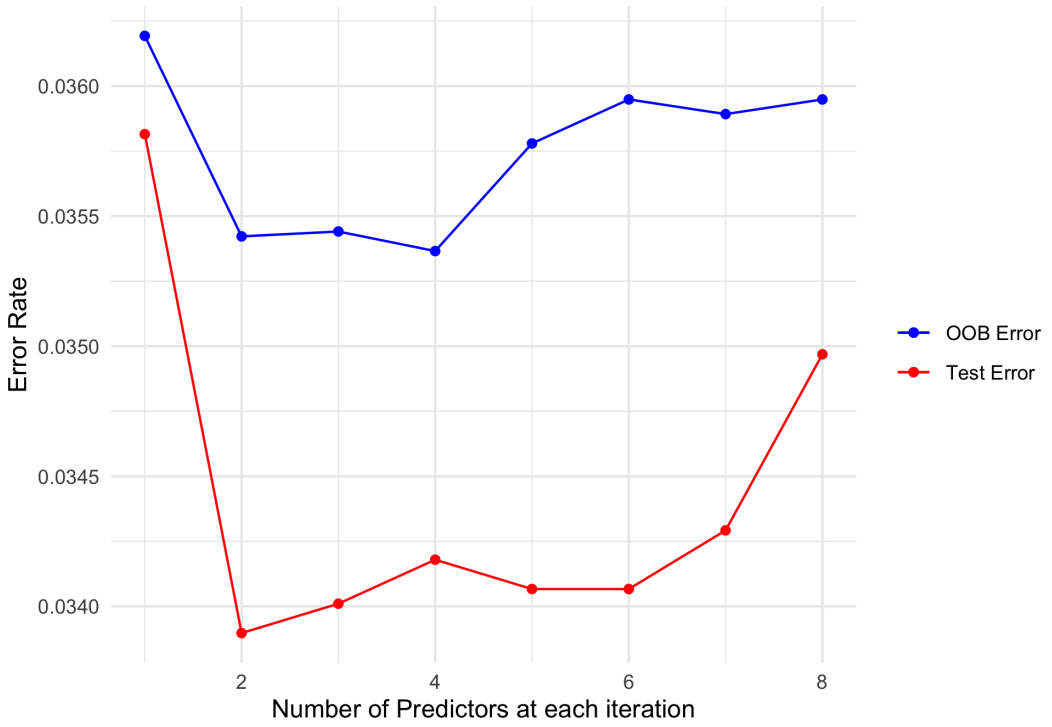

Figure 6: Errors for Different number of predictors

The main hyper parameter I considered is the number of trees to consider at each step. After plotting the error rate and out-of-bag (OOB) rate, clearly the optimal number is 3.

Image	error	oob
Image1	3.401	3.544
Image2	0.891	0.958
Image3	2.858	4.228

The results of random forests are promising. We get an average of almost 3.5% error rate both in terms of prediction and OOB across images. We did a ten-fold cross validation to achieve an accuracy rate of 93%.

3.3 Model 3 - k-Nearest Neighbors (k-NN)

While the reference paper clearly shows that ELCM (along with its QDA add-on) performs better than SVM and conventional MISR algorithms, we believe there is still scope for improvement. In particular, as mentioned in last paragraph of Sec. 4.1 in the reference paper, ELCM does poorly where there is a coastline or where the terrain information is outdated.

Because expert labels correctly reflect the true terrain, outdated maps result in disagreements for the ELCM prediction. We believe modern methods, such as a multi-layer neural network, can be used to harness this extra information and construct a better model. However, the time and computing resources required did not

allow us to do a meaningful neural network exploration for this lab. Thus, perhaps its prudent to consider an *alternative* but much simpler model. ***k*-Nearest Neighbors (kNN)** algorithm is one such candidate. It returns the (*majority* of k) closest training vectors' labels as the prediction. This assumption is reasonable: if the radiation measurements are similar for two pixels, they are more likely to cloud *together* (or "non-cloud snow" together) than not.

In this section, we show the results of k-NN applied to *only* features CORR, NDAI and SD, This is because, these features have contributions of all 4 raw measurement bands (not just red) as well as the geography at the coordinates. And as we saw in EDA, these composite features also lead to much better separability of expert labeling.

The features are centered to their *minima* and scaled so that they are transformed to a positive fraction $\in (0, 1)$. This scaling is required to remove the effect of different units and map the unbounded feature space into the compact unit cube $[0, 1]^3$.

A major shortcoming of k-NN could be the fact that one may need to store a lot of training data as "model" to have any meaningful agreement with expert labeling. As shown in the following section, this is surprisingly not the case!

3.3.1 Effect of model/training size

The k-Nearest Neighbor algorithm is one of the simplest that one can think of. Remarkably, it is also *very* small and fast in this case, . Fig. 7 plots the average validation accuracy over 20 independent runs with a specified size of random training data. All the 3 images are have about 0.11 million pixels but *we only need* 100 *pixels to get an agreement better than* 99%*!*

Consequently, the trained mode needs minimal memory for 100 or so **exemplars**. Because the number of points in the training set are so small, each prediction query is quite fast requiring a small memory overhead (for instance, it took us about $5 - 6$ mins. to generate Fig. 7). Interestingly, for very few training examples such as 10, the agreement percentage is in order of the expert label coverage in the images (`image2` > `image1` > `image3`).

Note that the agreement values of Fig. 7 take into account "unmarked" labels (i.e. 0). Thus, a refined method would be to look at average coverage and accuracy rates of only "cloud" or "non-cloud snow" test data. Fig. 8 shows that the average coverage of predictions is very close to the training image's label coverage. This is reasonable to expect, as for a random pixel the nearest neighbor will likely have an expert label in ratio similar to the total coverage. Thus `image3` has $\approx 45\%$ coverage whereas `image1` has about 70% coverage.

```
## Warning: Continuous limits supplied to discrete scale.
## Did you mean `limits = factor(...)` or `scale_*_continuous()`?
```

```
## Warning: Continuous limits supplied to discrete scale.
## Did you mean `limits = factor(...)` or `scale_*_continuous()`?
```

Similarly, for those test pixels where expert labels are available, how does our 100-example, 2-NN model agree? Fig. 9 shows the same; the agreement ration rapidly reaches to about 98% if we use more than 50 examples to train. **Hence, the** 100**-example,** 2**-NN model is versatile in learning expert labels!**

```
## Warning: Continuous limits supplied to discrete scale.
## Did you mean `limits = factor(...)` or `scale_*_continuous()`?
```

3.3.2 Effect of number of neighbors (k)

How does the choice of k affect the result? This mostly depends on the data distribution: a more **isotropic distribution** (i.e. it is *approximately* symmetric around *most* of its members) could do with a smaller value of k without degrading its agreement with expert labels. Fig. 10 shows the agreement results of $k \in [6]$ for 100-example k-NN model.

```
## Warning: Continuous limits supplied to discrete scale.
## Did you mean `limits = factor(...)` or `scale_*_continuous()`?
```

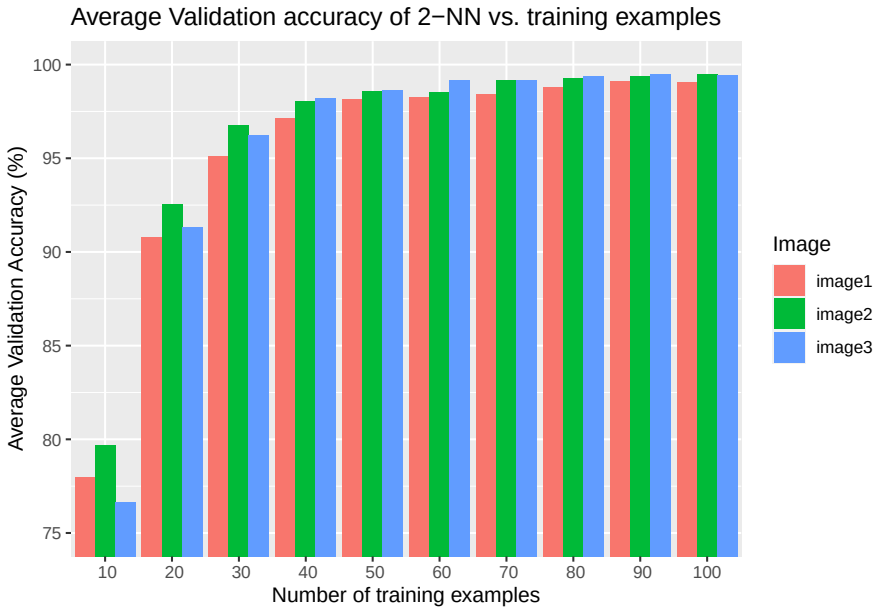

Figure 7: Average Validation accuracy of k-Nearest Neighbors (k = 2).

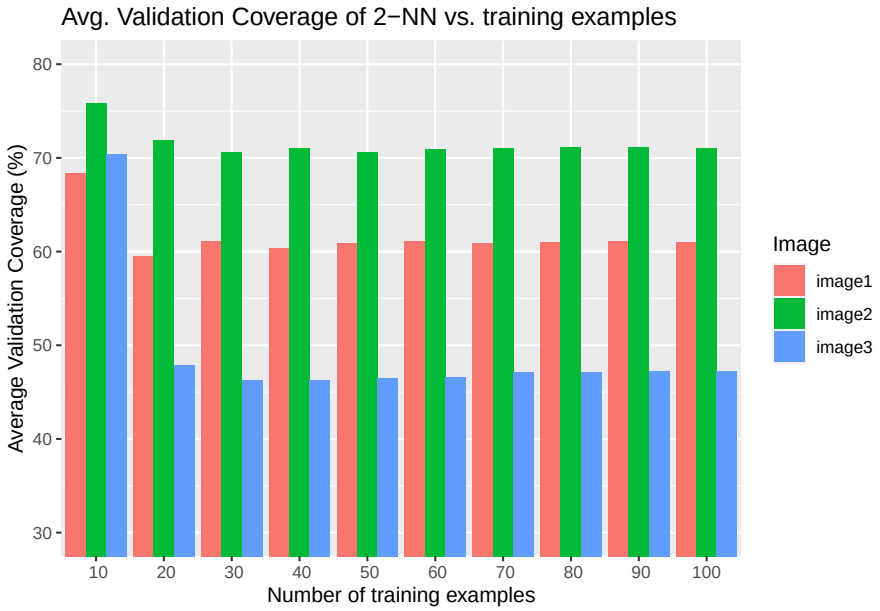

Figure 8: Average Validation Coverage of k-Nearest Neighbors (k = 2).

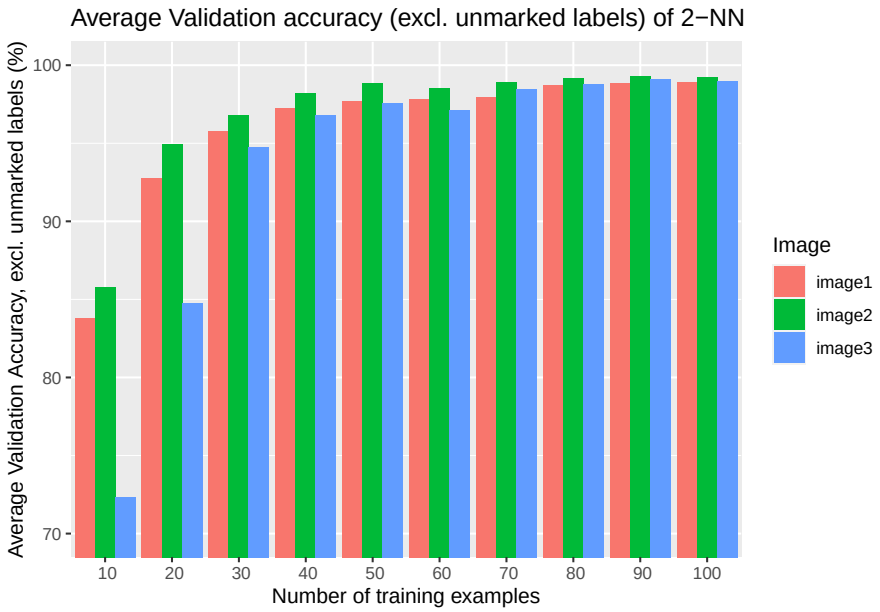

Figure 9: Average Validation accuracy of k-Nearest Neighbors (k = 2).

When the distribution of each label is far from isotropic, then adding more neighbors makes the boundary softer. To see this, consider a node with a lot of nearest neighbors sharing its label is the one with nearly-certain label. Such nodes, by their definition, inhabit the high-density region of their label. Consequently, a node with a lot of neighbors with a different label near it is more likely to be mislabeled. From Fig. 10, we see that $k = 2$ is optimal. Hence, we chose 2-NN, 100-example file for our final display.

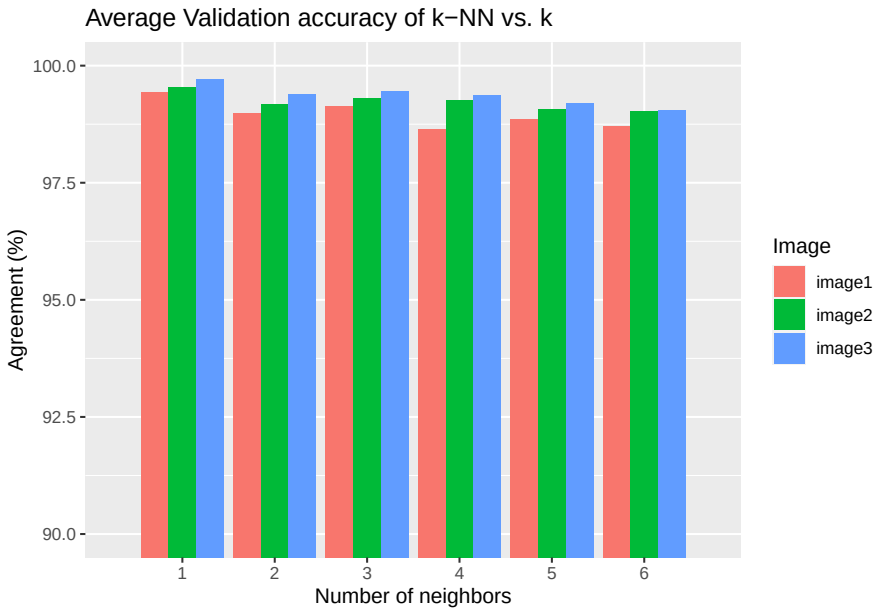

Figure 10: Average Validation accuracy of k-Nearest Neighbors with k

Table 3: Average agreement percentages over 20 independent runs of 100-example, 2-NN based on raw red features (DF, BF, AN) and derived features (CORR, NDAI, SD).

Image	Raw.Features	Derived.Features
image1	99.18895	99.09297
image2	99.38666	99.41349
image3	99.65739	99.45634

3.3.3 Raw vs. Derived features

We chose the red features AN, BF and DF for our raw-feature-based image as they are *directly related* to the construction of the derived features: CORR uses linear correlation between AN and BF, SD is related to the standard deviation of AN, and NDAI uses AN and DF.

Table 3 shows that the use of raw features and derived features are equally accurate for k-NN. Both result in *over* 99% *accuracy*! At such high agreement rates, any identity of the mislabeling are *very likely* to be random rather than being a persistent issue.

4 Conclusion

As we see from Sec. 3, the best performance was given by 2-NN trained with 100 examples (in Sec. 3.3) with more than 99% accuracy *for all* 3 *images*. The best extreme gradient-boosted linear model (with L_2 regularization and 0.1 learning rate, shown in Sec. 3.1) produces 95%, 97% and 90% agreements respectively in `image1`, `image2` and `image3`. The best random forest (in Sec. 3.2) produces about 96.5% agreement rate for the 3 images.

We feel such extremely high accuracy with respect to the expert labels for 2-NN is possible only because the authors have *already architected* carefully derived features from raw measurements. However, it is also important to note that the 2-NN *does not discover* any new knowledge about clouds versus snow any more than those provided by the labels. Sec. 3.3.1 illustrates this specifically where the prediction coverage was very close to the true label coverage. Therefore, it is crucial to remember that k-NN is *strictly supervised* model. In particular, for a new data set the 2-NN labeling will only reflect the expert knowledge used to label this data set. A refinement would be to try and discover new information for a semi-supervised learning model.