A GMM approach to cloud removal from the ICESat-2 Satellite Photon Return Data

Matthew Lawson and Benjamin Morris

Code link: https://github.com/benjijamorris/ICESat2

File Name: ATL03_20181028022213_04500105_001_01.h5

Photon return data from the ICESat-2 satellite (https://nsidc.org/data/atl03) is used for a variety of purposes such as tracking glacial activity and measuring rainforest canopy density. However, photon dispersion from clouds and daylight result in noisy data, making it difficult to identify relevant features. We experimented with various pre-processing steps and modifications to the expectation-maximization algorithm to create a Gaussian Mixture Model (GMM) to separate cloud and ground photon data points to aid additional analyses.

We worked primarily on using two-component, two-dimensional Gaussian Mixture Models to hard cluster points as either ground or cloud using the Expectation-Maximization (EM) algorithm. Two features of the data are considered - elevation (calculated from the photon return time) and distance along the satellite's track. Gaussian Mixture Models are a clustering method that aims to detect latent normally distributed populations in data. Essentially, a number of "mixtures" are created that are modeled as two-dimensional Gaussian distributions parametrized by a mean, covariance matrix, and probability of a point belonging to that mixture. Using the Gaussian Probability Density Function, we can then assign each point a probability of belonging to each mixture. To find the two mixtures that gave our data the highest likelihood, we used the EM algorithm. EM is a way to find the maximum likelihood estimation in an iterative way. Since we are using it with 2D Normal distributions, the expectation step first determines how likely each data point is to be in each distribution. Then, it recalculates the parameters of each distribution to maximize the likelihood. It repeats these two steps until the parameters converge.

We discovered early on that running our model on a dataset with too long of a distance did not separate cloud and ground points well due to changes in the ground elevation, which prompted our exploration into how to window the data. We started windowing by the distance along the satellite track, but noticed that some windows with fewer points did not cluster as well. We then moved to a windowing approach based on the number of points per window, but that also did not perform well. Finally, we settled on an approach that identifies windows containing clouds using the rolling variance of the window grouped by 1000 points (Fig. 1). We noted that windows with variances above the 60th percentile had clouds, so we used this cutoff to decide whether to use the GMM or the less computationally demanding median to identify ground points. With more time, this is the approach that we would implement.

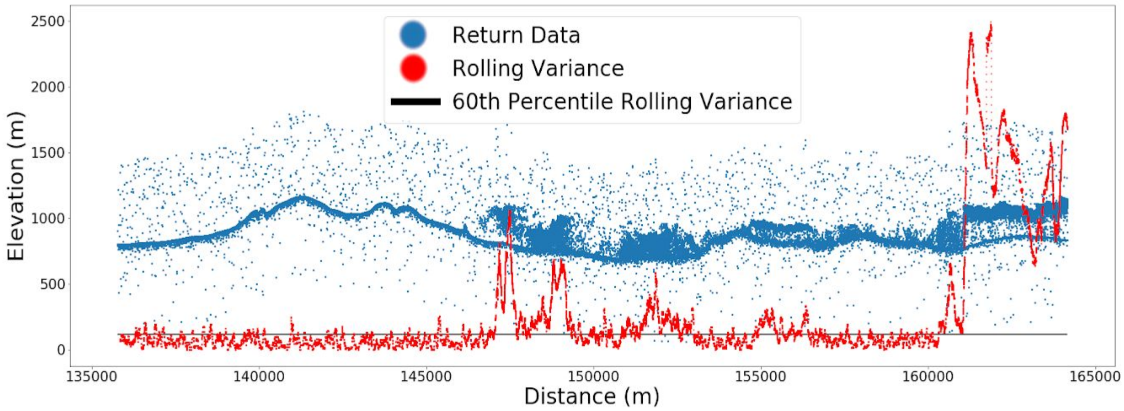

Figure 1: Elevation and rolling variance vs. distance. Cloudy regions are consistently above the 60th percentile.

We also experimented with different methods of calculating the covariance matrix for each of our mixtures. In each iteration, we used a covariance update that scaled each point's contribution to the covariance matrices of each mixture by its probability of belonging to that mixture. In an early iteration, we mistakenly calculated the variance in the XY direction as $\sum(X-\mu_X)\times(Y-\mu_X)$ and found that this worked well (Fig. 2).

However, as we increased the number of windows, our covariance matrix eventually became singular, meaning it did not interact well with our PDF function. Even after correcting the mistake in the update, this issue continued. As an additional attempt, in the maximization step of EM, we hard clustered each point to one of the mixtures and calculated the covariance of each mixture. This also did not work at small window sizes. To deal with this, we just used larger window sizes.

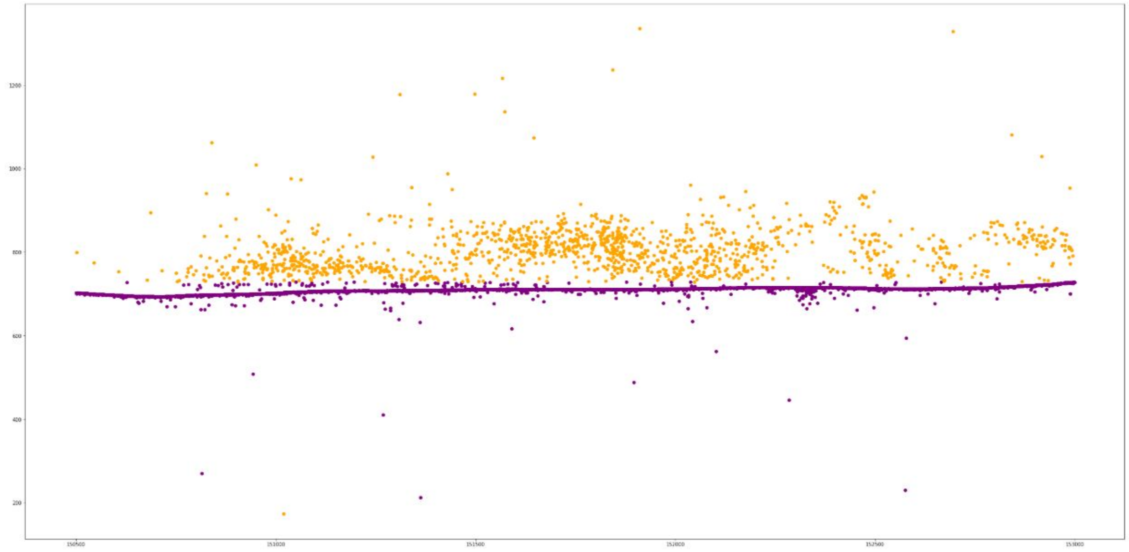

Figure 2: This is an early iteration with the incorrect covariance calculation. It worked for small subsets of data but did not generalize.

As a final pre-processing experiment, we noted that windows with noisy points scattered above the clouds and below the ground line seemed to cluster poorly, so we decided to remove the top and bottom 2% of points in each window. This did not noticeably improve accuracy on these difficult windows.

In our final model, several issues persisted. Chiefly, some windows marked nearly all points as ground, when some points are clearly cloud. In a plot of rolling median (Fig. 3), we noted that the points where the median approach to ground identification fails (i.e. when the median approaches the level of the clouds) indicating that the clouds are especially dense. However, this went against our intuition that GMMs would be best suited for two, dense clusters. This likely is an issue with our initialization of the ground and cloud cluster centers as the median and maximum points respectively for each window. To address this, we tried initializing the cluster centers as the min and max for each window (Fig.4). While this approach seemed to work better in areas with dense clouds, it failed in other areas that our naive initialization worked well in (Fig. 5). Moving forward, we would thus use a hybrid approach to determine the initialization method based on changes in the median from window to window and also use rolling variance to determine whether the GMM or median should be used in cloud/ground classification.

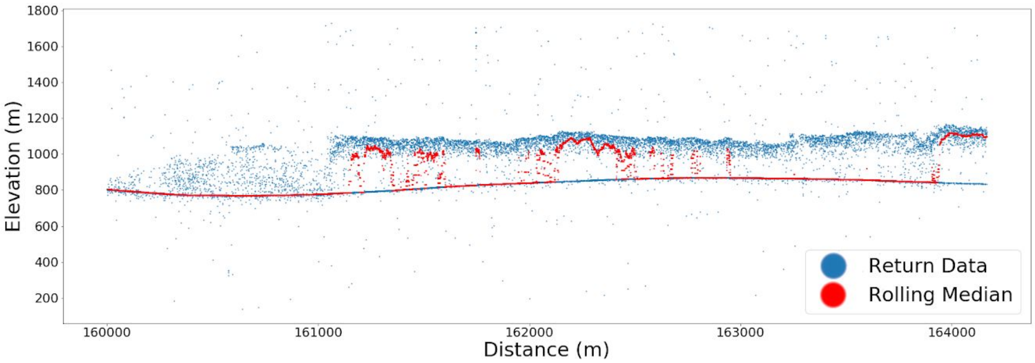

Figure 3: The rolling median is not a good indication of the ground when there are dense clouds present.

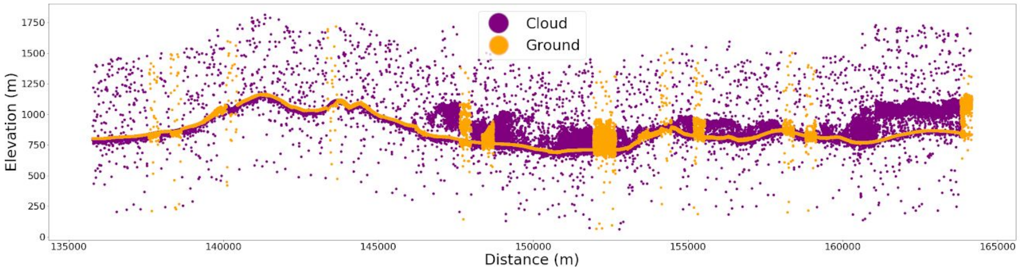

Figure 4: Using minimum and maximum as the initial cluster centroids.

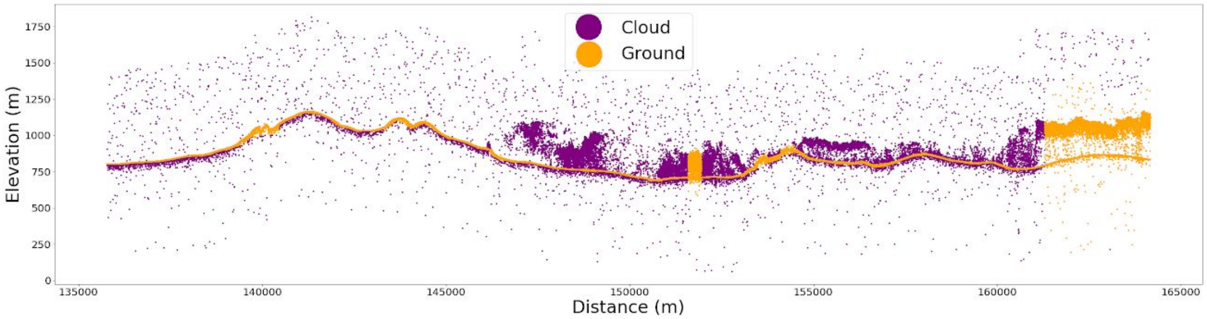

Figure 5: There are still some problems with false positives (cloud marked ground), but very few false negatives (ground marked ground).

Our primary goal for the semester was to get experience in data science outside of the toy problems we have seen in class. We learned about the importance of factors besides the actual algorithm used. Specifically, we spent the majority of our time on cleaning and preprocessing the data before running the GMM. Additionally, we learned about troubleshooting issues that came up. For example, when we found that a window was poorly clustered, we came up with an idea about why that might be and added a preprocessing step to address that issue.

Due to our difficulties with fully implementing *one* method, we did not meet our goal of implementing several methods and comparing their performance. However, now that we have learned some additional clustering methods, we have a better perspective on what sorts of clustering tools might work better. For example, we would now consider using some sort of transformation on the data to make it more separable (e.g. representing y-coordinates as an absolute value from the median of each window to make it more easily separable or some other mapping).

We also developed a perspective on algorithm choice. As the data is clearly not normally distributed, we could have experimented with a mixture model of k=2 chi-squared distributions. Finally, we improved our ability to make plots that give insight into the data being presented. Specifically, our plot of rolling variance placed over the raw data gave us the idea to base our windows on variance rather than size and our plot of rolling median gave us insight as to why our algorithm was failing on some windows with dense clouds.