Lab 4 - Cloud Data
Stat 215A, Fall 2014

Ruoxi Jia

# 1 Introduction

In this Lab, our mission is to predict cloud in the polar regions using a group of selected features. Firstly, we carry out EDA to explore feature distribution and their potential correlations. Then based on EDA observation, three features are selected as discriminators of cloud detection. Secondly, variety of statistical and non-probabilistic models are fitted to the data and compared with each other with both visual and quantitative methods. Finally, we try to diagnose fitted models and explore error patterns for possible model improvement.

# 2 EDA

# 3 Modeling

## 3.1 Feature Selection

In this section, we perform feature selection in a top-down manner, i.e. we begin with a full set of features, and sequentially remove "ineligible" features from both visual justification by plotting the conditional densities of different features and quantitative exanmination of features' seperability under different conditions using classical permutation test. We also use Akiake Information Criterion (AIC) to measure the relative quality of features, which takes into account both the goodness of fit of the model and the complexity.

In order to support classification, features' distributions should have a good separation between distinctive classes. The dissimilarity of distributions can be measured by Jensen-Shannon divergence (JSD), which can be derived from KL divergence and is symmetric distance metric. JSD is given by the following formula:

$$JSD(P||Q) = \frac{1}{2}[KL(P||Q) + KL(P||Q)] \tag{3.1}$$

where $KL(P||Q)$ is the KL divergence, defined as

$$KL(P||Q) = \sum_i ln(\frac{P}{Q})P \tag{3.2}$$

We verify the seperability of a certain feature using permutation test. The idea is to randomly permuate labels of pixels and each time obtain a JSD. The null hypothesis is that the observed JSD for a given feature is independent of the cloud/clear labelings, namely

$$H_0^{Feature} : JSD_{Observed}^{Feature} = JSD_{Permutated}^{Feature} \tag{3.3}$$

The result for permutation test is illustrated in Fig.??, where the histogram is obtained by shuffling the labels for 100 times and the red vertical lines denote the JSD with correct labels. We can see that for features except $AF$ and $AN$, the true JSD significantly deviates from the JSD distribution in permutation figures/PermutationTest.png

Figure 3.1: Results for permutation test

test. In conclusion, *AF* and *AN* cannot achieve very high seperability and should be pushed out of our feature candidate pool.

Secondly, we perform a stepwise AIC algorithm on sampled data set (all three images included) to examine the goodness of features. Stepwise AIC is a algorithm to keep tracking the information loss when removing each feature and select the model with the minimum AIC value. The AIC results are shown in Table ??. We can see that removing *AF, BF, CF, AN* results in the least increase of AIC, which indicates that these four features are not very informative and thereby we will remove them from our feature pool.

From the successive exclusion of features in proceeding steps, we now have four features at hand, namely *NDAI, SD, CORR, DF*. We plot the conditional distribution of these four features (Fig. ??) and it is shown that they all achieve good seperability.

| Features | Df | Deviance | AIC |
|---|---|---|---|
| <none> | | 110896 | 110914 |
| - AF | 1 | 111068 | 111084 |
| - BF | 1 | 111071 | 111087 |
| - CF | 1 | 111103 | 111119 |
| - AN | 1 | 111826 | 111842 |
| - CORR | 1 | 112499 | 112515 |
| - DF | 1 | 112672 | 112688 |
| - SD | 1 | 114236 | 114252 |
| - NDAI | 1 | 168818 | 168834 |

Table 1: Results for AIC

In order to further justify our choice visually, we try to consider combinations of features but reduce their dimentionality to 2D for visualization purpose. Here we repeatly apply PCA to a random sample data set with different combinations of features. In the first trial, we use all features, in the second one only *NDAI, SD, CORR, DF* are incorporated and in the last case the other features are used. From the resulted Fig. ??, where points with different labels are marked differently, we can tell that the second case is the best for classification, since points from two groups are more seperated that in the other two cases. This is a good justification for our choice of features in the feature selection step. In order to make a comparision with result in Bin's paper, we will proceed our exploration on diverse classification models with four features: *NDAI, SD, CORR.*

## 3.2 Diverse Classification Models

### 3.2.1 Iterative Conditional Random Field Cloud Segmentation

We start from the ELCM-QDA model presented in Bin's paper. ELCM-QDA algorithm basically exploits the seperability property of the feature space by thresholding feature values. However, clear and cloudy pixels are not stable in the sense that the feature values might change across time and space. In order to deal with the instability of features, thresholds are chosen by a data-adaptive way, where a mixed-Gaussian model is fitted to the data and the threshold is derived from the dip between two Gaussian distributions. The paper also used Fisher's QDA model to estimate the probability of cloudiness. The result of cloud detection using ELCM-QDA is illstrated in Fig. ??. The agreement with expert labels for three images are 93.39%, 93.5%, 82.84%, respectively. However, if we scrutinize the prediction and the true labels, we can see that adjacent pixels tend to have the same labels, while ELCM-QDA features a simple thresholding without considering this spatial pattern.

Here we present a Iterative Conditional Random Field Cloud Segmentation (ICRFCS) model which outperforms ELCM-QDA by taking into account spatial homogeneity in cloud images. ICRFCS relies on Bayesian estimation via Markov random field, in which the spatial information in an image is encoded through contextual constraints of neighboring pixels. By imposing such constraints, we expect neighboring pixels to have the same class labels. Let's formalize our algorithm as follows: For each pixel $s$, the region-type that the pixel belongs to is specified by a class label, $w_s$, which is binary in our case, i.e $w_s$ is modeled as a discrete random variable taking values in $\Lambda = \{-1, 1\}$. The set of these labels $w = w_s, s \in S$ is a random field, called the label process. The observed image features are supposed to be a realization $\mathrm{F}\{f_s | s \in S\}$ from a another random field,w hcih is a function of the label process $w$. Basically, the image process F represents the manifestation of the underlying label process. Thus, the overall ICRFCS model is composed of the hidden label process $w$ and the observable noisy image process F. ICRFCS aims to find an optimal labeling $\hat{w}$ which maximize the posterior probability $P(w|\mathrm{F})$, that is the MAP estimate

$$\hat{w} = argmax_w P(\mathrm{F}|w)P(w) \tag{3.4}$$

According to the derivation in [??], the optimization problem above is equivalent to the following energy minimization problem:

$$\hat{w} = argmin_w U(w, \mathrm{F}) \tag{3.5}$$

figures/ConditionalDistribution.png

Figure 3.2: Distributions for NDAI, SD, CORR, DF

where the energy function $U(w, \mathrm{F})$ is given by

$$U(w, \mathrm{F}) = \sum_{s \in S} (ln(\sqrt{(2\pi)^n |\Sigma_{w_s}|}) + \frac{1}{2}(f_s - \mu_s)\Sigma_{w_s}^{-1}(f_s - \mu_s)^T) + \beta \sum_{\{s,r\} \in C} \delta(w_s, w_r) \tag{3.6}$$

Minimizing the first term will give us the MLE estimate of labels assuming the features are independent and follow a multi-variate Gaussian distribution. The second term assign greater clique potentials if neighboring pixels have similar classes. $\beta$ is a weighting parameter controlling the importance of local homogeneity. Now, the cloud segmentation probelm is reduced to the minimization of the above function. Since, it is non-convex, combinational optimization techniques are needed to find the global minimum. Due to the computation complexity, we solve this problem in an iterative way. The idea is that we adopt the conventional figures/PCA_selection.png

Figure 3.3: PCA for different combinations of features

MLE, while at each step we consider only one pixel and ignore goemetrical considerations and merely choose the optimal by minimizing the energy function assiciated with this pixel. We apply the algorithm until convergence. In our experiment, 5 cycles are enough for the result to be converged.

The cloud dection result using ICRFCS is shown in Fig. ??. The accuracy of cloud detection for three images are $96.41\%, 95.42\%, 92.24\%$, respectively, which are higher than ELCM-QDA algorithm presented in original paper.

## 3.3 Model Assessment

figures/ELCMQDA.png

Figure 3.4: Classification results of ELCM-QDA

figures/CRF.png

Figure 3.5: Classification results of ICRFCS