JPL/Caltech Virtual Summer School in Big Data Analytics

Decision Trees and Random Forests

Exercise 2: Random Forests for Space Exploration

Overview: To date, most small bodies exploration has involved short timescale flybys that execute pre-scripted data collection sequences. Light time delay means that the spacecraft must operate completely autonomously without direct control from the ground. But in most cases the physical properties and morphologies of prospective targets are unknown before the flyby. Features of interest are highly localized, and successful observations are much dependent on geometry and illumination constraints. Under these circumstances onboard computer vision can improve science yield by responding immediately to collected imagery for example by targeting features of opportunity for additional data collection by specialized instruments with a narrow field of view. This exercise will demonstrate an approach for the detection and localization of high albedo surface features and evaluate performance using a case study with archival datasets from previous primitive bodies encounters.

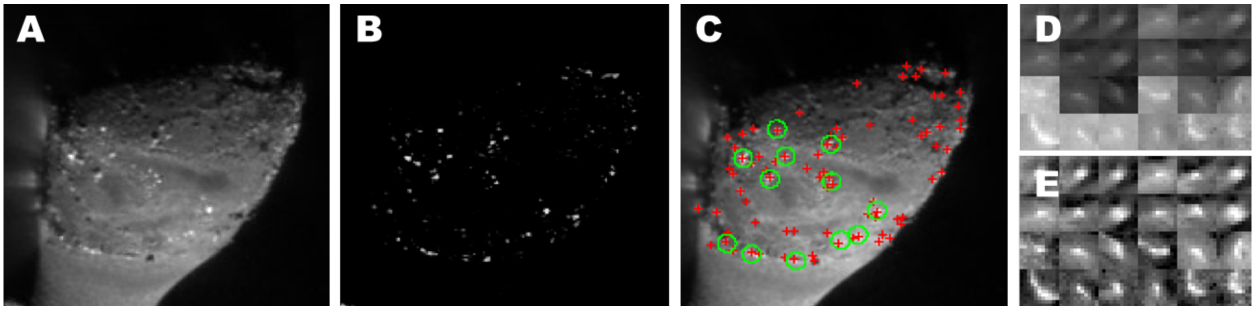

Figure 1: **A:** First we retrieve images of comet Hartley 2 taken by the framing camera of the Deep Impact probe during the EPOXI mission from PDS. **B:** The difference of the grayscale and median filtered image is renormalized. **C:** Set of possible surface features as a result of weighted mean shift clustering (red crosses) and ground truth labels from a domain expert (green circles). **D:** Patches of labeled surface features from 9P/Tempel. **E:** Locally normalized patches, constituting the positive class for training.

Exercise Goal:

Train a random forest classifier to differentiate between true surface features (labeled by an planetary scientist) and false detection. Learn how to extract features and how to test their impact on classification.

TODO:

1. Download and unzip the surface feature dataset of comet Hartley 2: http://bit.ly/1ACSIHu
2. Load the data into R
   ```
   dat <- dget("C:/summerschool/patchesHartley2.rdat")
   str(dat)
   ```
3. Extract the target variable Y from the data structure
   ```
   Y <- sapply(dat, function(x) x$Y)
   Y <- as.factor(Y)
   table(Y)
   ```
4. Create a design matrix **X** by extract features from the grayscale patches. Start by using a vector of raw grayscale values. For example:
   ```
   X <- matrix(nrow=length(dat),
   ncol=length(as.vector(dat[[1]]$gray)))
   for (i in 1:length(dat))
   {
      X[i,] <- as.vector(dat[[i]]$gray)
   }
   ```
5. Train a random forest to differentiate true from false surface features
6. Investigate the feature importance to decide which features make sense and which do not.
7. Report the OOB error of your model on Google Hangout to compare with other students.
8. Add new features to the design matrix and go back to 5

Bonus Tasks:

1. Plot an ROC curve based on the confidence of the classifier to see its performance.
2. Compare the ROC curves of different models in the same plot.

Hints:

- ROC curves can be plottet with the “epi” package or the ”ROCR” package.
- Image patches and matrices can be visualized with the “image()” function.
  ```
  image(dat[[100]]$gray)
  ```