Exploration of Statistical Learning Methods to Classify and Predict Water Maser Phenomena

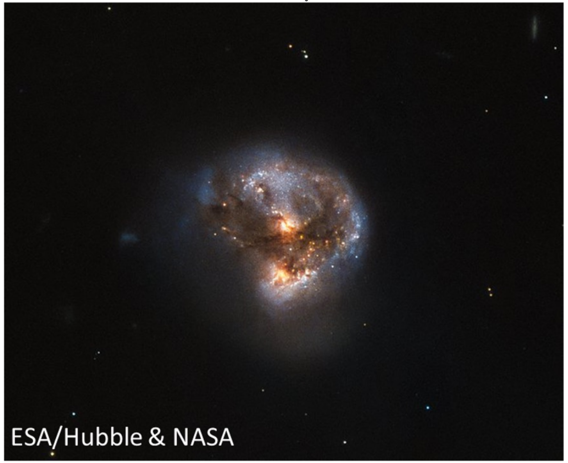
Ty Nunley

With support from Dr. Anca Constantin and Dr. Nusrat Jahan

What is a Water Maser?

- **M**icrowave **A**mplification by **S**timulated **E**mission of **R**adiation
  - Comes from water molecule clouds near star forming regions or centers of galaxies with active supermassive black holes
- Mega-Masers
  - 10^6 more luminous than regular masers.
  - Important to measure distances to galaxies, to ultimately constrain Hubble's Constant

- Data comes from MegaMaser Cosmology Project, crossmatched with data from Sloan Digital Sky Survey spectroscopic surveys.

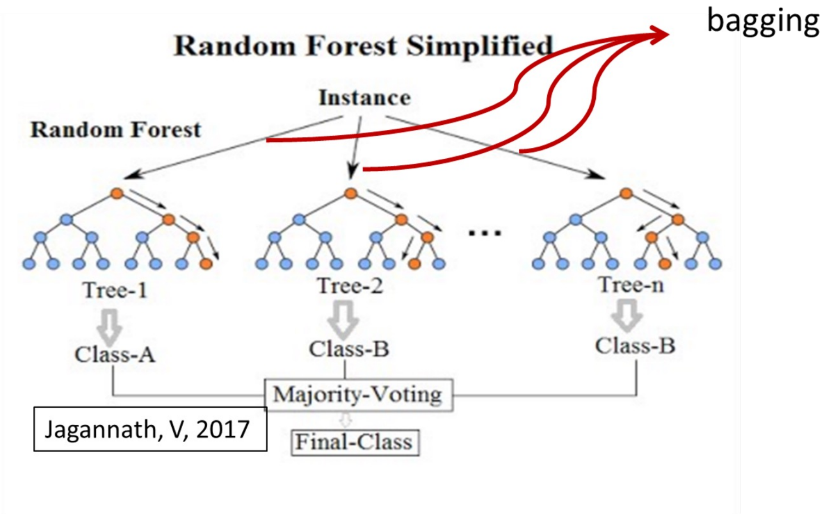


The Problem

- Imbalanced dataset
  - ~3-5% of data is mega-masers
  - In classification problems, this is hard to deal with

- The goal: create a model to classify correctly mega-masers from non-masers and (later) predict mega-maser emissions from observed galaxies

Methods Used



- **Random Forest**
  - Used to classify response using a “forest” of decision trees created through bagging (bootstrapping and aggregating).
  - Trains weak learners **simultaneously**
- **Boosting**
  - Trains **sequentially** to combine weak learners into stronger ones.
  - Boosting minimizes loss functions to better predict data!
  - LogitBoost minimizes **logistic** loss of an additive regression model.
  - AdaBoost minimizes **exponential** loss of an additive regression model.

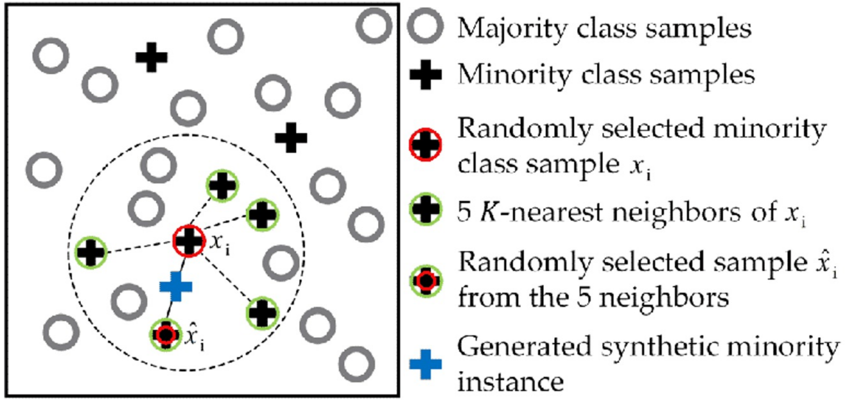
Each method is good in its own way!

- **Random forest**
  - Pros:
    - Does not overfit with many predictors.
    - Efficient in classification, but not typically the best
  - Cons:
    - Struggles with computational time
    - Struggles to make a predictive model with significance of each parameter.
- **Boosting (LogitBoost and AdaBoost)**
  - Pros:
    - Good with missing data and binary classification problems
    - Combines weak learners to train itself over time.
  - Cons:
    - Boosting in general is difficult to fine-tune

SMOTE (Synthetic Minority Oversampling Technique)

Description:

- Synthetically generates new data
- Oversamples minority class / undersamples majority class
- Then run analysis / ML



Pro & Con:

- **Great** at dealing with imbalanced data
- Can overfit with lots of noise, especially with high oversampling

Assessment Tools for Classification

Confusion Matrix

TP	FP
FN	TN

TP: True Positive; positive classification
TN: True Negative; negative classification
FP: False Positive; incorrect positive classification
FN: False Negative; incorrect negative classificiaton

- Kappa

Kappa is accuracy (TP + TN) / (TP + FP + FN + TN) when random chance is introduced.

- Sensitivity

Sensitivity = TP / (TP + FN)

- Specificity

Specificity = TN / (TN + FP)

Our Goal this Summer

- Lots of fine-tuning and exploration this summer!
  - A considerable chunk was spent fine tuning code, testing arguments that we thought would change the results but didn’t.
- Machine learning methods are tested at various splits of training/testing set ratios.
  - 50/50 //// 60/40 //// 75/25
  - Different split ratios can impact the results!
- Each method was iterated 100 times under the same seed for reproducibility.

Results

Test Set Kappa Values for Various Methods						
Split (Train/Test)	Random Forest	RF with SMOTE	LogitBoost	LB with SMOTE	Adaboost	AB w/ SMOTE
50%/50%	0.3139015	0.3595253	0.5035764	0.3847365	0.4601629	0.4239876
60%/40%	0.3521292	0.3644899	0.4820578	0.3839502	0.4953192	0.3873108
75%/25%	0.3986459	0.3719552	0.5116384	0.4068302	0.5105	0.3983504

Kappa values are low! Typically we want 0.80+.

Results (cont.)

Test Set Sensitivity Values for Various Methods						
Split (Train/Test)	Random Forest	RF with SMOTE	LogitBoost	LB with SMOTE	Adaboost	AB with SMOTE
50%/50%	0.9927	0.913203	0.971296	0.9861795	0.9678561	0.9819045
60%/40%	0.9921992	0.9133835	0.9729595	0.9875934	0.9693536	0.9831904
75%/25%	0.9915	0.9106006	0.9731638	0.9875662	0.9697	0.9825604

Sensitivity is high! This is what we expect.

Test Set Specificity Values for Various Methods						
Split (Train/Test)	Random Forest	RF with SMOTE	LogitBoost	LB with SMOTE	Adaboost	AB with SMOTE
50%/50%	0.2303	0.6677143	0.9002408	0.3015472	0.6880046	0.3547212
60%/40%	0.2678571	0.6768	0.6511	0.2969201	0.7307905	0.3125355
75%/25%	0.3116667	0.6961111	0.7098248	0.3186489	0.7270	0.3256054

Specificity is also low most of the time. This is because the data is imbalanced!

Conclusions & future work

- No conclusions.... Yet!
- Explore different methods and explore other measures of classification
- Make a prediction model based on the data using an equation derived from the optimal classifier and optimal measure.
- Conduct an investigation using ROC (Receiving Operating Characteristic) curve to determine the optimal tradeoff between sensitivity and specificity.
- Investigate why SMOTE underperforms with Boosting.

References

Friedman, J. et al., 2000

Zierr, C. & Biermann, P., 2018, A&A, 69, 1

Bowyer et al., 2002

Chawla, N. et al., 2001

Bühlmann, P & Dettling, M., 2002

Images:

https://rikunert.com/smote_explained

Jagannath, V, 2017

ESA/Hubble & NASA