Feature Selection Topics

Sergei V. Gleyzer

Data Science at the LHC Workshop

Nov. 9, 2015

Outline

- Motivation
- What is Feature Selection
- Feature Selection Methods
- Recent work and ideas
- Caveats

Motivation

- Common Machine Learning (ML) problems in HEP:
  - Classification or class discrimination
    - Higgs event or background?
  - Regression or function estimation
    - How to best model particle energy based on detector measurements

Motivation continued

- While performing data analysis one of the most crucial decisions is which features to use
  - Garbage In = Garbage Out
  - Ingredients:
    - Relevance to the problem
    - Level of understanding of the feature
    - Power of the feature and its relationship with others

Goal

- **How to:**

Select

Assess

Improve

Feature set

used to solve the problem

Example

- **Build a classifier to discriminate events of different classes based on event kinematics**
- **Typical initial feature set:**
  - **Functions of object four-vectors in event**
  - **Basic kinematics: transverse momenta, invariant masses, angular separations**
    - **More complex features relating objects in the event topology using physics knowledge to help discriminate among classes (thrusts, helicity e.t.c.)**

Initial Selection

- **Features initially chosen due to their individual performance**
  - **How well does X discriminate between signal and background?**
    - **Vetos: Is X well-understood?**
    - **Theoretical and other uncertainties**
    - **Monte-Carlo and data agreement**
  - **Arrive at order of 10-30 features (95% use cases)**

Feature Engineering

- By combining features with each other, boosting into other frames of reference* this set can grow quickly from tens to hundreds of features
  - That's ok if you have enough of computational power
    - Still small compared to 100k features of cancer/image recognition datasets
  - Balance between Occam's razor and need for additional performance/power

* JHEP 1104:069,2011 K. Black, et. al.

Practicum

Training Set → Feature Selection → Model Building → f() ✓

Training/Test Set → Feature Selection* → Model Building → f() ✗

Test Set → f() → Estimate Performance ✓

***Feature Selection Bias**

Methods

Filters

Feature Selection → Model Building

Wrappers

Model Building → Feature Selection

Embedded-Hybrid

Feature Selection during Model Building

Filter Methods

- **Filters: usually fast**
  - **No feedback from the Classifier**
  - **Use correlations/mutual information gain**

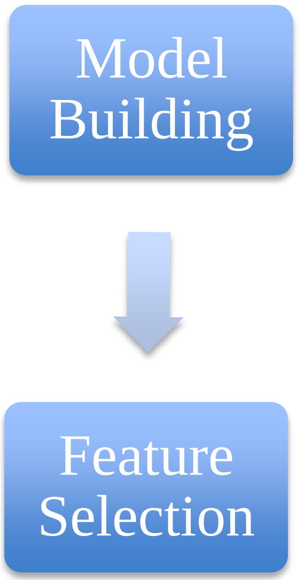
"quick and dirty" and less accurate

Useful in pre-processing

Example algorithms: information gain, Relief, rank-sum test, e.t.c.

Wrapper Methods

- **Wrappers: typically slower and relatively more accurate (due to model-building)**
  - **Tied to a chosen model:**
    - **Use it to evaluate features**
    - **Assess feature interactions**
    - **Search for optimal subset of features**
  - **Different types:**
    - **Methodical**
    - **Probabilistic (random hill-climbing)**
    - **Heuristic (forward backward elimination)**



Ex: Feature Importance

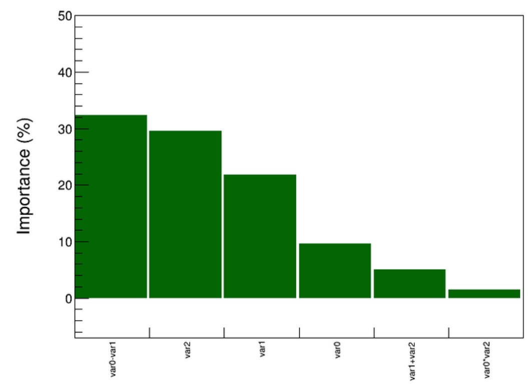
- **Feature Importance** $\longrightarrow$ **proportional to classifier performance in which feature participates**

$$FI(X_i) = \sum_{S \subseteq V : X_i \in S} F(S) \times W_{X_i}(S)$$

- **Full feature set {*V*}**
- **Feature subsets {*S*}**
- **Classifier performance *F(S)***

$$W_{X_i}(S) \equiv 1 - \frac{F(S - \{X_i\})}{F(S)}$$

- **Fast stochastic version uses random subset seeds**



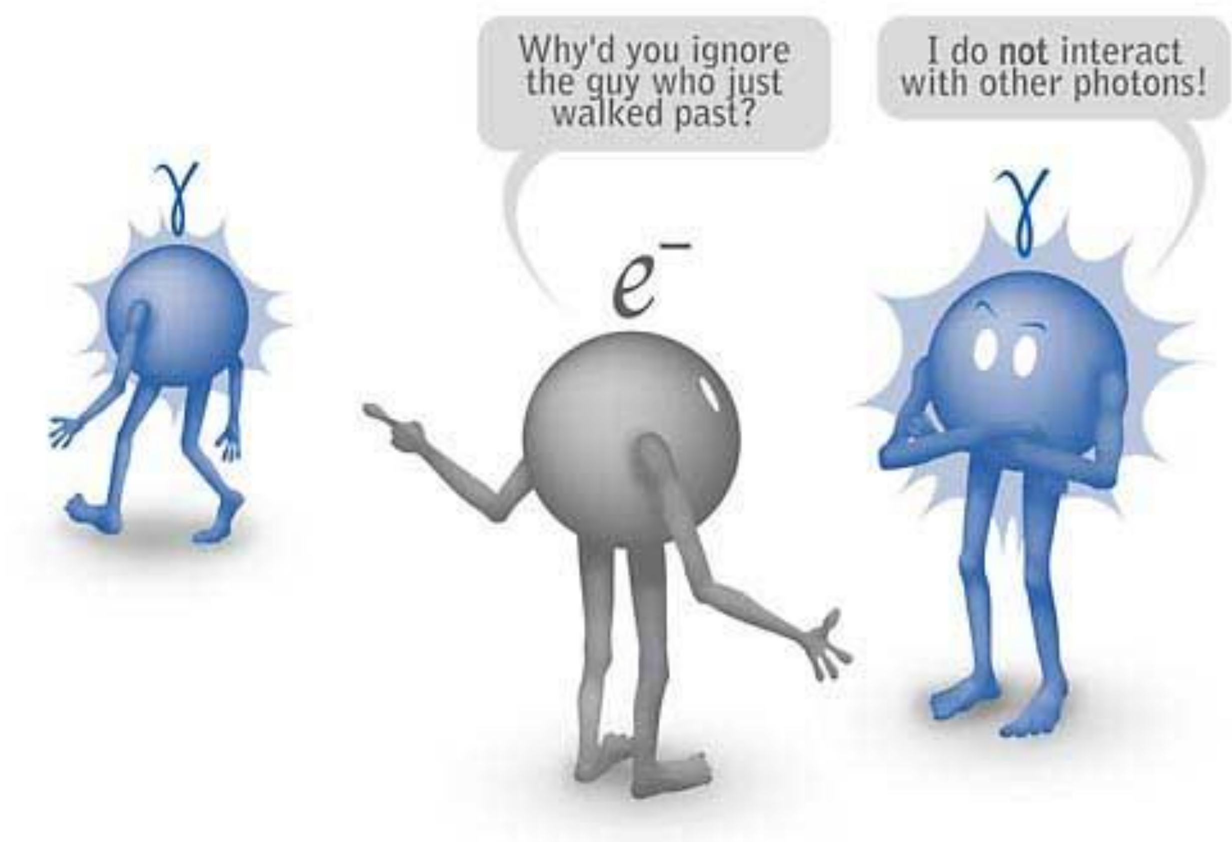
Example: RuleFit

- **Rulefit:** rule-based binary classification and regression (J. Friedman)
  - Transforms decision trees into rule ensembles
  - A powerful classifier even if some rules are poor
- **Feature Importance:**
  - Proportional to performance of classifiers in which features participate (similar)
  - Difference: no $W_i(S)$
    - Individual Classifier Performance evenly divided among participating features

Selection Caveats

- **Feature Selection Bias**
  - **Common mistake leading to over-optimistic evaluation of classifiers from "usage" of the testing dataset**
  - **Solution:**
    - **M-fold cross-validation/Bootstrap**
    - **Second testing sample for evaluation**

Feature Interactions



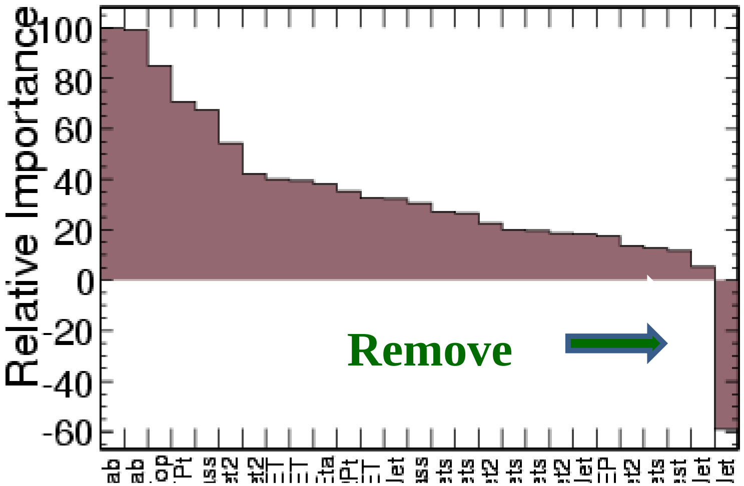
Features Like to Interact Too

Feature Interactions

- Features often interact strongly in the classification process.
  - Their removal affects the performance of remaining interacting partners
    - Strength of interaction quantified by some wrapper methods
  - In some classifiers features can be overlooked (or shadowed) by their interacting partners

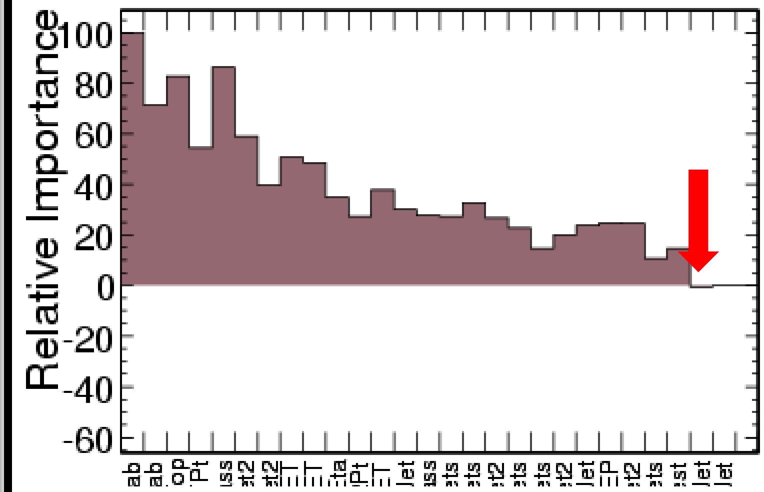
Beware of hidden reefs

Selection Caveats



Before

Importance Landscape



After

Has Changed

Holds for any criterion that doesn't incorporate interactions

Global Loss Function

- **GLoss Function ⟹ Global measure of loss**
  - **Selects feature subsets for global removal**

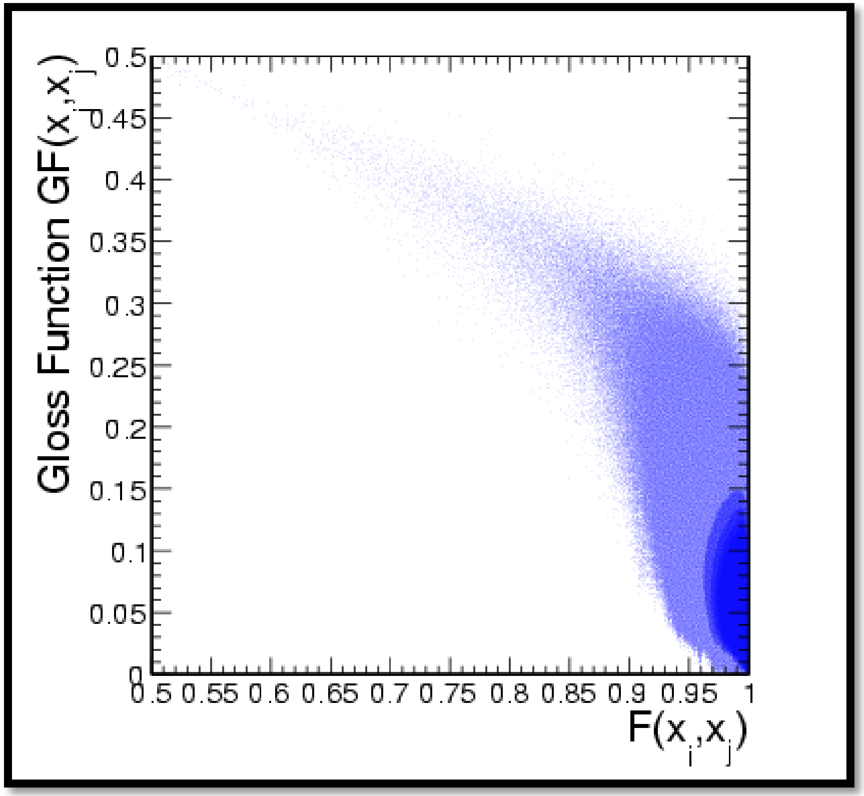
$$GF(S') \equiv 1 - \frac{\sum_{S \subset (V-S')} F(S)}{2^{|V-S'|}}$$

S' is the subset to be removed

- **Shows the amount of predictive power loss relative to the upper bound of performance of remaining classifiers**

$$\sum_{S \subset (V-S')} F(S)_{\max} = 2^{|V-S'|}$$

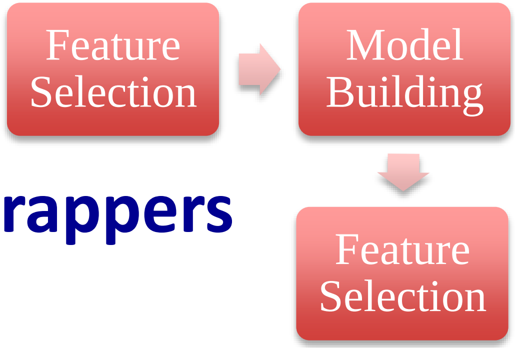
Global Loss and Classifier Performance



GLoss Function minimization NOT EQUIVALENT to Maximization of F(S) – i.e. finding the highest performing classifier and its constituent features

Recent Work

- **Probabilistic Wrapper Methods:**
  - Stochastic approach
- **Hybrid Algorithms:**
  - combine Filters and Wrappers



- **Embedded Methods**

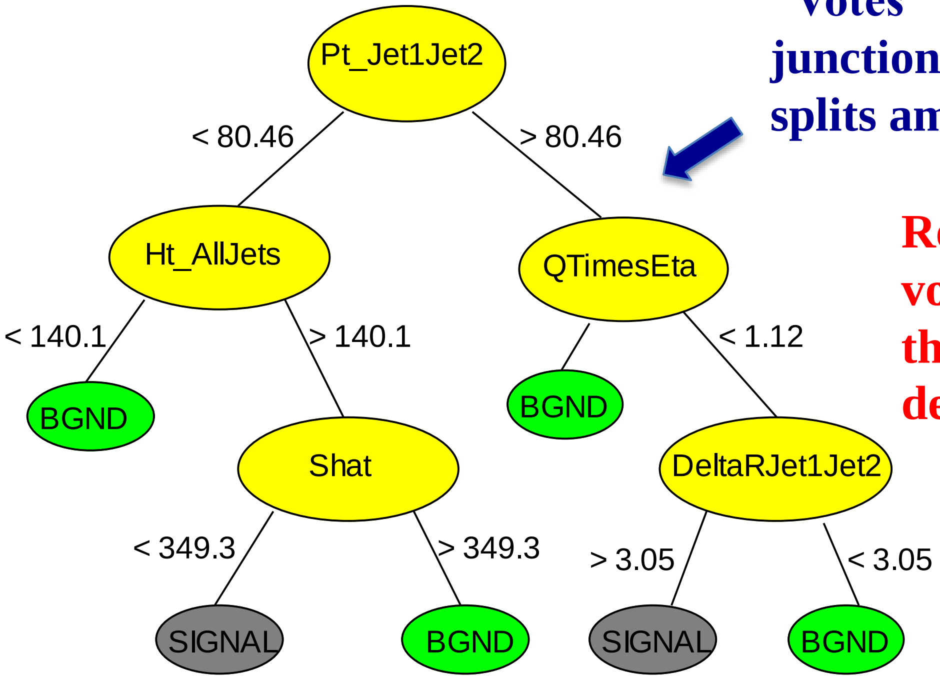
Feature Selection during Model Building

Embedded Methods

- At model-building stage assess feature importance and incorporate it in the process
  - Way to penalize/remove features in the classification or regression process
    - Regularization
    - Examples: LASSO, RegTrees

Regularized Trees

- **Inspired by Rules Regularization in Friedman and Popescu 2008**
- **Decision Tree Reminder:**



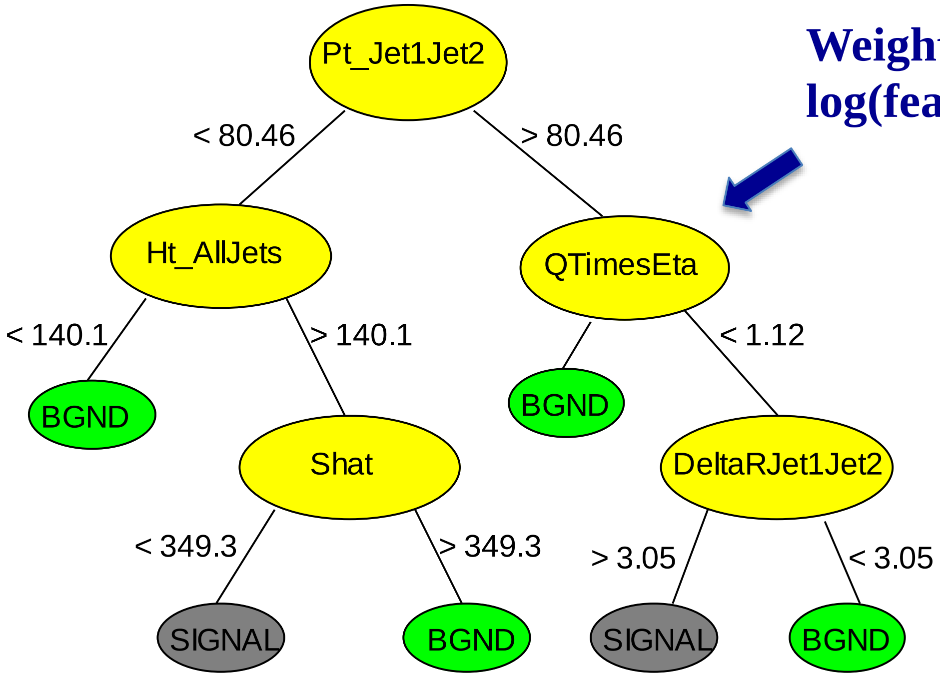
"Votes" taken at decision junctions on possible splits among the features

RegTrees penalize during voting features similar to those used in previous decisions

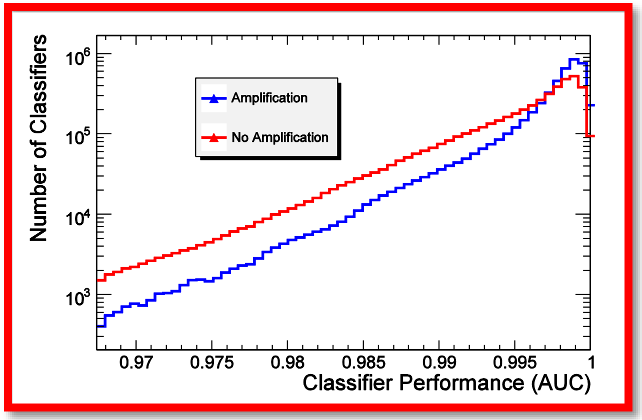
End up with a high quality feature set classifier

Feature Amplification

- **Another example: feedback feature importance into classifier building**



Weight votes by log(feature importance)

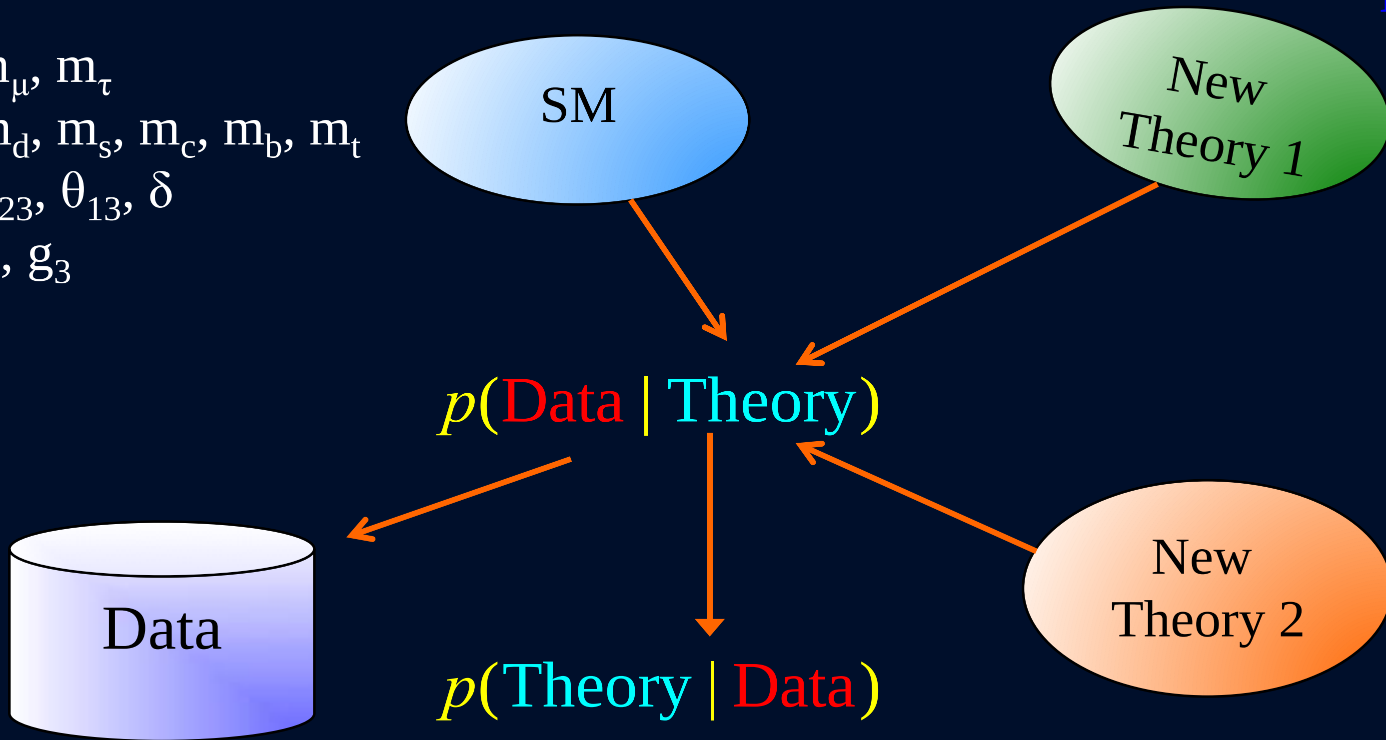


Decade Ahead

All interesting theories are ***multi-parameter*** models

H. Prosper

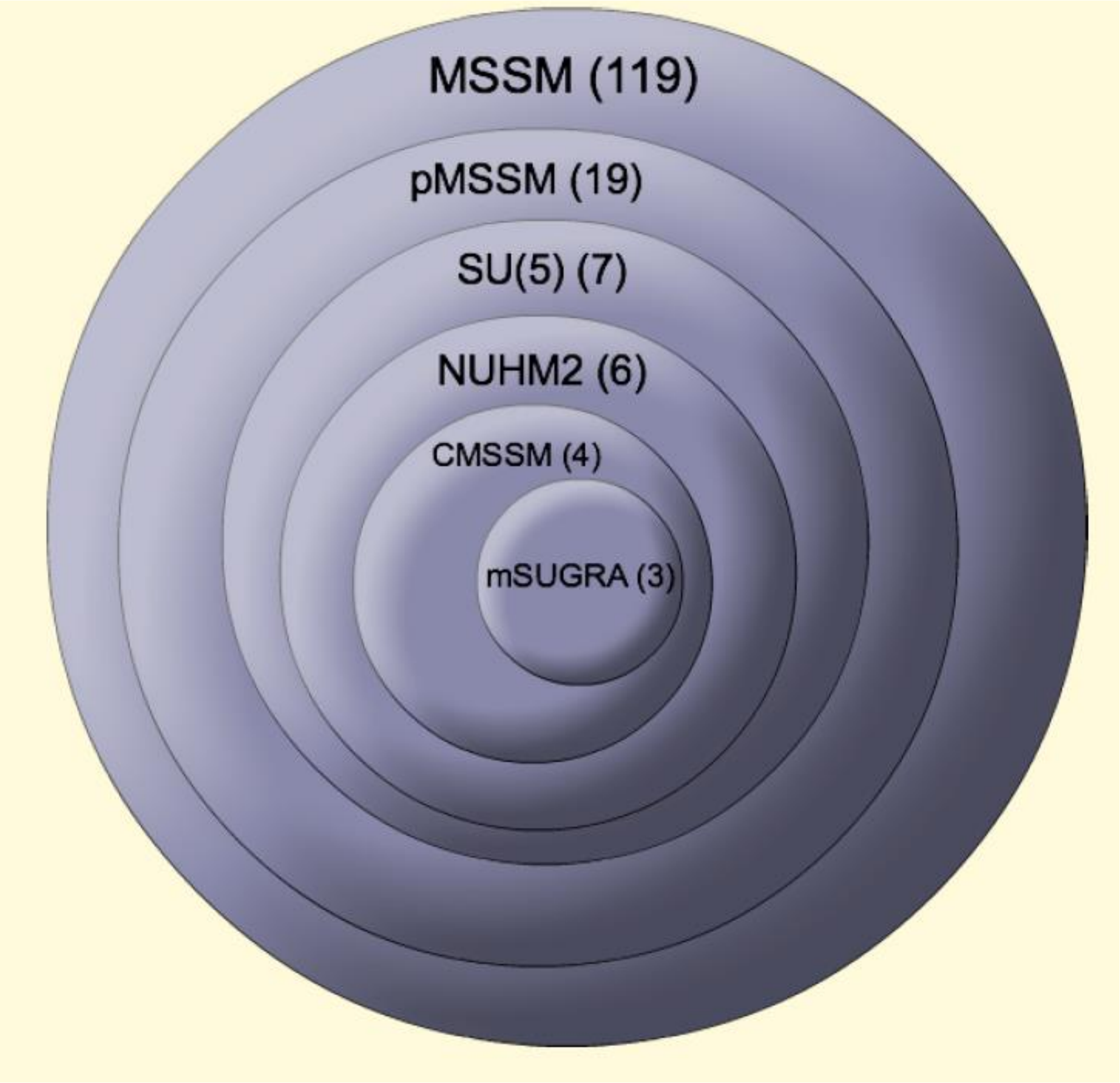
m_e, m_μ, m_τ
m_u, m_d, m_s, m_c, m_b, m_t
θ_{12}, θ_{23}, θ_{13}, δ
g_1, g_2, g_3
θ_{QCD}
μ, λ



Basic *statistical* questions:

1. Which theories are preferred, given the data?
2. And which parameter sub-spaces within these theories?

Minimal SUSY



In HEP

- **Often in HEP one searches for new phenomena and applies classifiers trained on MC for at least one of the classes (signal) or sometimes both to real data**
  - **Flexibility is KEY to any search**
  - **It is more beneficial to choose a reduced parameter space that consistently produces strong performing classifiers at actual analysis time**
    - **Useful for general SUSY and other new phenomena searches**

Feature Selection Tools

- R (CRAN): Boruta, RFE, CFS, Fselector, caret
- TMVA: FAST algo (stochastic wrapper), Global Loss function
- Scikit-Learn
- Bioconductor

Summary

- Feature selection is important part of robust HEP Machine Learning applications
- Many methods available
- Watch out for caveats
- Happy ANALYZING