

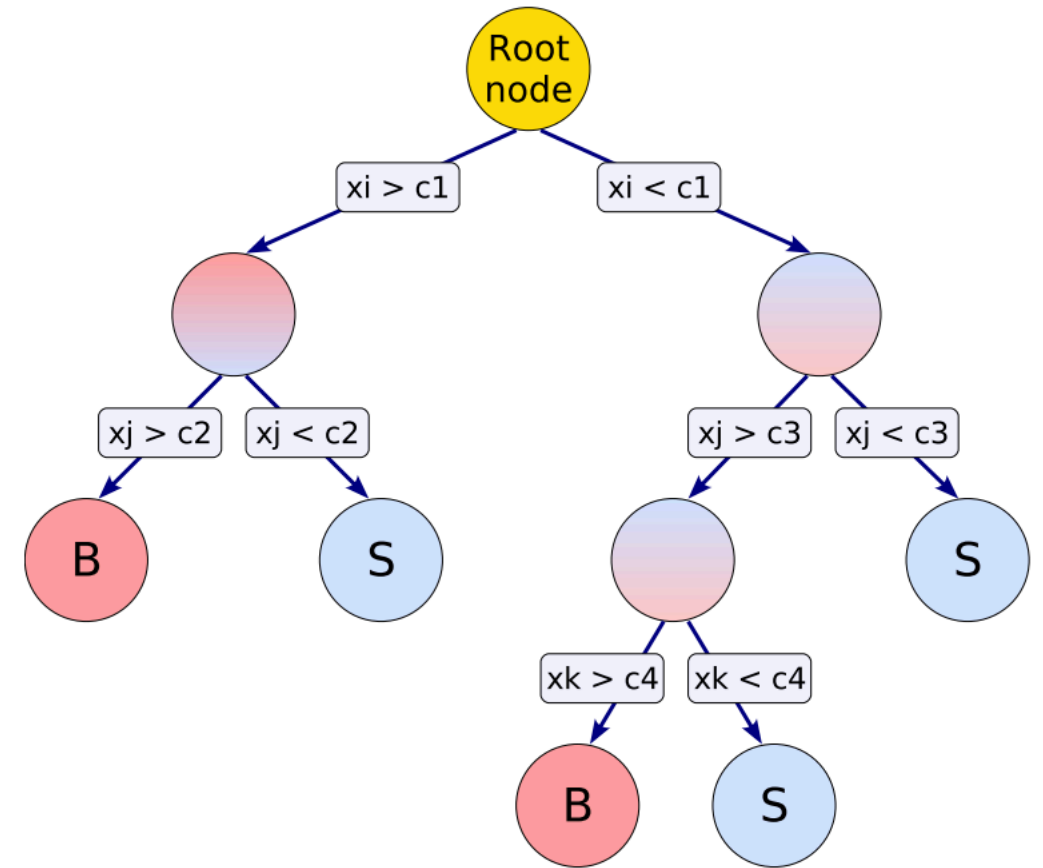
BDT Evaluation of Tagger Performance using TMVA (MV2) --Proof of Principle

Xiaoning Wang

Aug 20, 2020

BDT Basics (Or in this case, boosted Regression tree)

- Each node takes an input variable (energy fraction, pt, eta, ip3d results, etc.)
- Each node is a function maps variable value to probability.
- Each jet traverses all nodes and yield a final probability of being a signal (b-jet) or background (light or c-jet).
- Weight: weights are adjusted to exclude irrelevant factors.
- Forest: many trees of the same training samples. Each re-weighted to reduce statistical fluctuation.
 - Method of reweighting depends on “boosting” option.

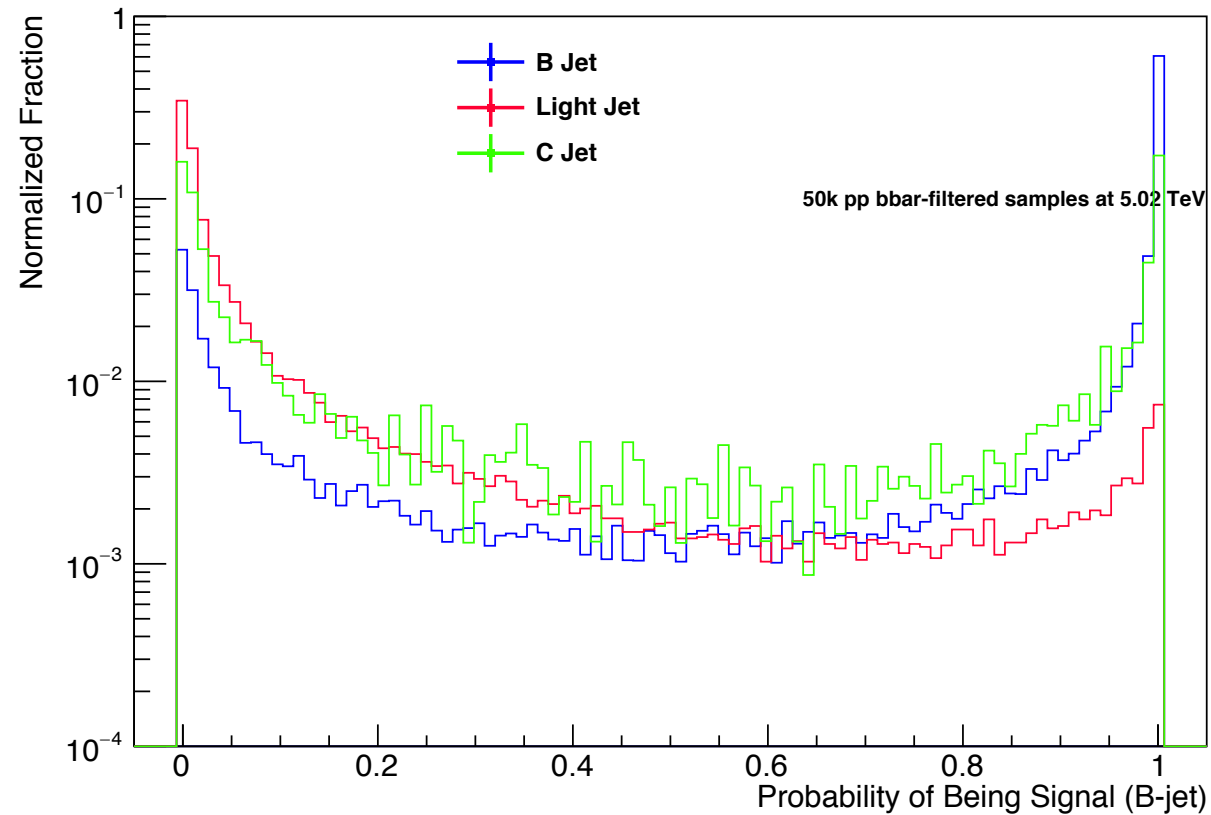


BDT Training and Testing

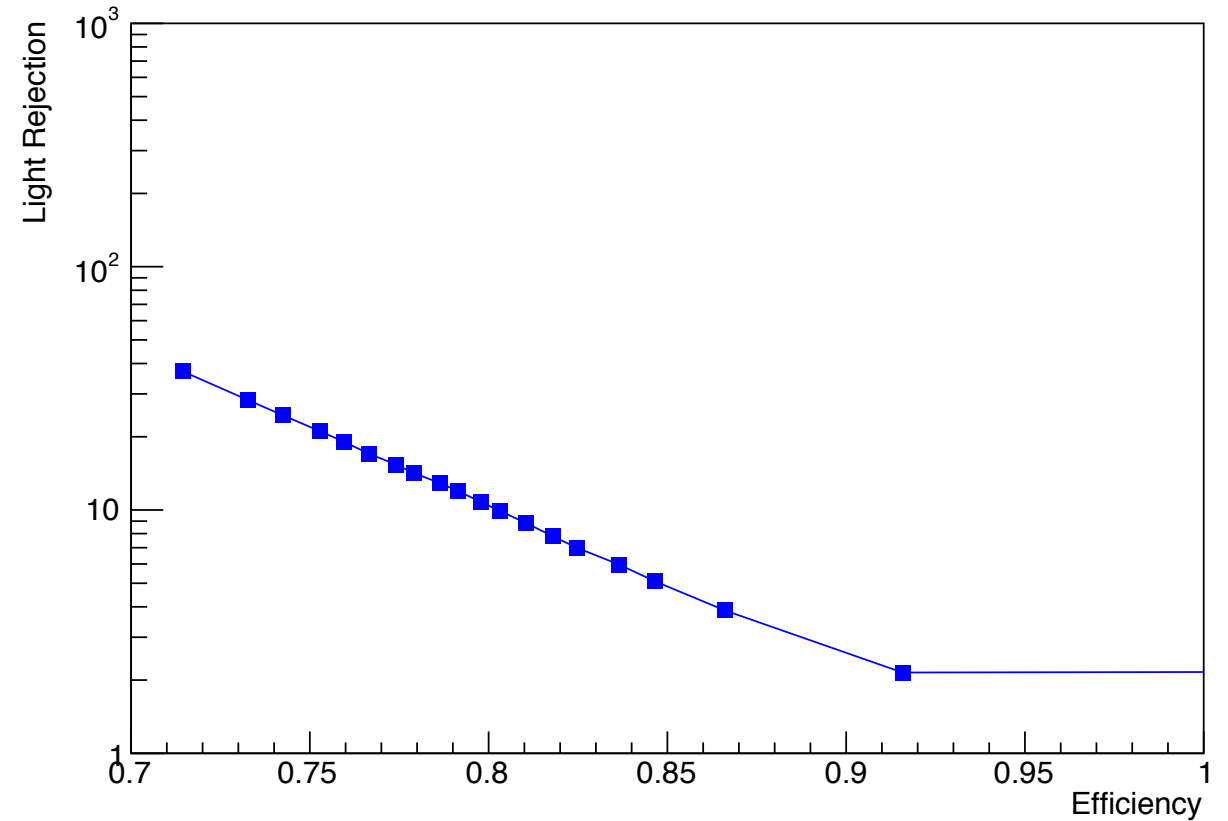
- Jet selection:
 - Truth matched.
 - Has a flavor label 0(u) 4(c) or 5(b).
 - Label determined by closest associated truth hadron within $dR = 0.3$
- Jets are into 2 halves (alternating between training then testing)
- Reweighting: b and c jets are re-weighted according to light jet pt-eta 2d distribution * original MC weight.
 - Can also reweight so each flavor takes a certain fraction.
 - Based off MV2
(<https://twiki.cern.ch/twiki/bin/viewauth/AtlasProtected/BTaggingMV2>)
- Training time:
 - ~80k jets used ~2 hours to train in BNL NoMachine. About 100M jets in inclusive dijets.
 - Is parallel training possible?

Proof of Principle with MV2 using 50k pp Samples

Probability of Being a B-jet for 3 flavors



Performance for MV2 (IP3D + JF + SV)



Why the double peak for all?

Thought: if a jet is b-like, most variables are b-like?

Summary and Plan

- Small samples (50k) can work with TMVA with a reasonable training time. Large samples can take a day.
 - If it doesn't crash for memory leaking...
 - Working on implementing randomly selecting small samples and centrality selection
- Split into small taggers for each tagger's performance using modified cuts.
 - comparing pp for JF/SV on their own.
- This framework outputs an xml file that can be made into "calibration file" which Athena can use in the retagging framework, when a reasonable training is done, proceed to implement it into a calibration file.

