Gradient Boosted Decision Trees and Particle Physics

Aaron Webb

October 27, 2017

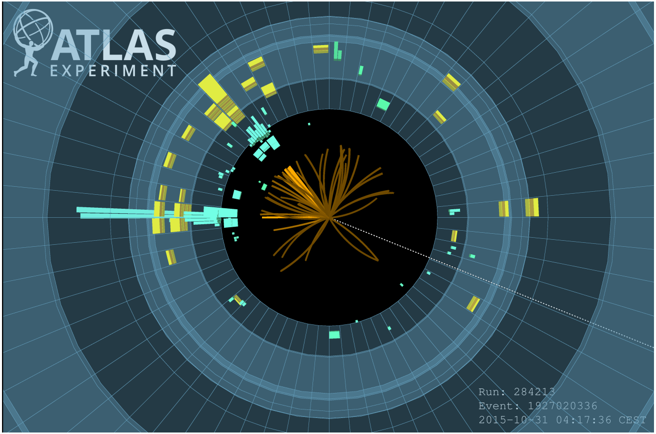
The LHC and Big Data

- Bunches of 10^{11} protons are collided every 25 ns
- Produces $\approx$ 50 PB of data per year
- Particle lifetimes $\mathcal{O}(10^{-25})$ seconds, only ever see decay products

- Many processes look the same in the detector
- Interesting interactions are rare

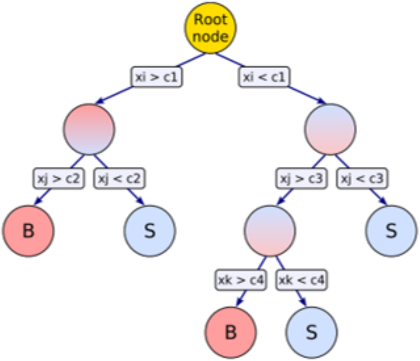
The ATLAS Dectector

- Tells us the types of particles, their momentum, energy, and location
- Use these to reconstruct interaction, e.g. $m^2 = E^2 - p^2$



Gradient Boosted Decision Trees

- Combines a set of weak "learners" into a single "strong" learner
- Start with a simple model - single binary decision tree
- Construct a new tree to correct the weaknesses of the model
- Iterate till it converges



Gradient Boosting Algorithm

- Begin with a simple model $F_0(x) = \underset{\gamma}{\arg\min} \sum_{i=1}^{n} L(y_i, \gamma)$
- Compute pseudo-residuals, $r_i = -\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}$
- Fit a new learner, $h_m(x)$, to maximize r_i
- Compute a weight, γ, for $h(x)$ using line search
- Update the model $F_{m+1}(x) = F_m(x) + \gamma_m h_m(x)$
- Iterate till the model converges
- Final model $F(x) = \sum_{i=1}^{M} \gamma_i h_i(x) + \text{const}$

Decision Trees

- Gradient Boosting is a general algorithm
- For BDTs, each $h(x)$ is a decision tree
- Scan feature set for the split that produces greatest gain
- Repeat for each node that results till max depth is reached

Improvements

- l_2 penalty term
  - Penalize complex trees, remove branches that produce little differentiation
- Shrinkage
  - $F_m(x) = F_{m-1}(x) + \nu \cdot \gamma_m h_m(x), \quad 0 < \nu \leq 1$
  - ν is the "learning rate", typically <0.1
  - Improves results, but increases computation costs
- Stochastic Boosting
  - Each successive tree is fit to a random subsample
  - Prevents overfitting, improves speed

Pros and Cons

Pros

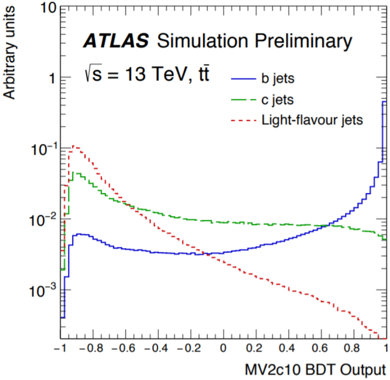
- Easy to use once model has been developed
- Few input parameters needed to tune
- General framework, relevant for a large number of applications

Cons

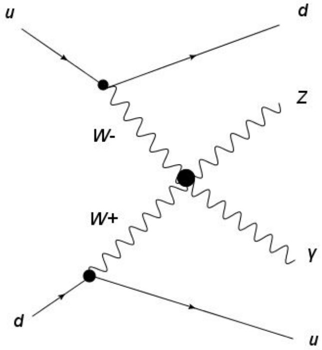
- Training the model can be slow
- Difficult to interpret the output
- Not ideal for sparse data, large numbers of features

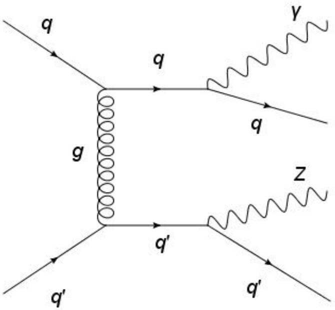
Uses in Particle Physics

- Separate signal and background events
  - Use Monte Carlo simulations to train the model, use for data
- Distinguish "real" particles from "fakes"
  - Particles misidentified by the detector, from secondary sources
- "B-tagging" - identifying different flavors of quarks
  - Quarks "hadronize", leaving complex signatures



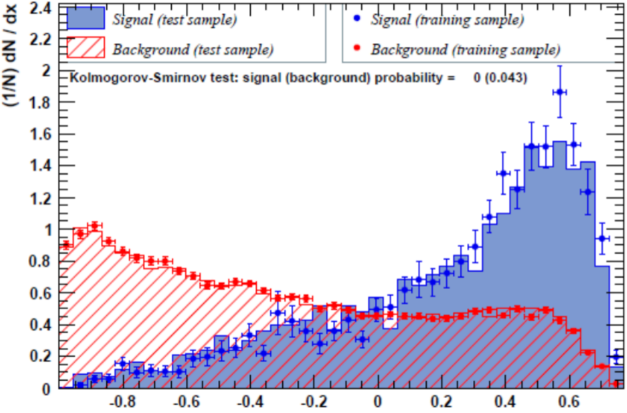
Vector Boson Scattering





Results

Input Variables	
M_{jj}	η_γ
ΔY_{jj}	$\Delta R_{\mu\gamma}$
$\Delta R_{j\gamma}$	$\Delta R_{\mu\mu}$
η_μ	$p_{T\gamma}$
p_{Tjet}	$p_{T\mu}$



Results

- Cut to maximize significance of the signal: $S/\sqrt{B}$
- BDT achieves 81% better significance than square cuts