Machine Learning

Prof. Sergei Gleyzer

Lecture

PH451, PH551

February 6, 2025

Announcements

- Hands-on #3/Reading HW #2
  - due next Tue
  - Textbook: Chapter 6
- Quiz
  - Feb 13
- Mini-hackathon #1 next week
  - Feb 10-21

Outline

- Ensembles
- Bagging
- Random Forests
- Boosting

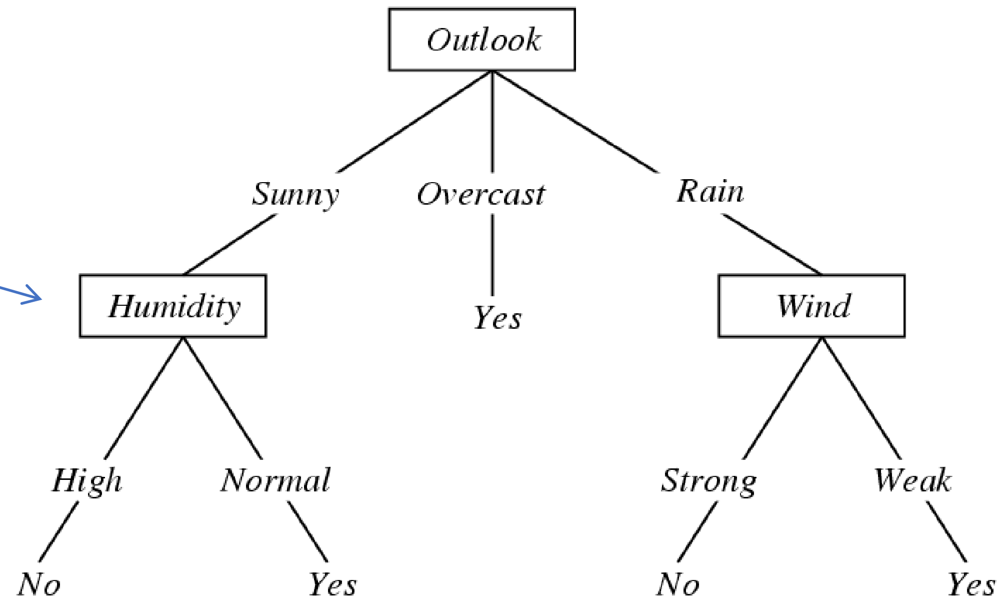
ML Methods (partial list)

- Fisher, Quadratic
- Naïve Bayes (Likelihood)
- Kernel Density Estimation
- Random Grid Search
- Rule ensembles
- Boosted decision trees
- Random forests
- Deep learning neural networks
- Support vector machines
- Genetic algorithms

Decision Trees (recap)

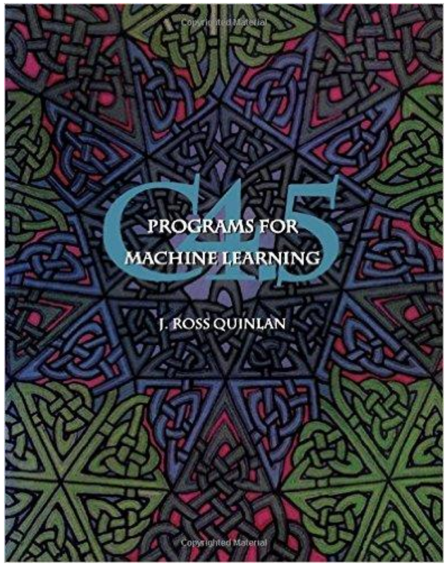
- Each **internal** node: test one attribute X_i
- Each **branch**: selects one value for X_i
- Each **leaf** node: predict Y
  - Or P(Y|X in leaf)

Decision Node



Decision Trees

- Classic **ML tool** for
  - **decision trees**
  - **rules**
  - **boosted classifiers**
- Written by **J.R. Quinlan**
  - Name: ID3 → C4.5 → C5.0
    - Use c5.0 to familiarize with decision tree classifiers



Pruning

Decision trees can become large and complex and risk **over-fitting** the data

Pruning: **remove parts** of the tree that are less powerful or possibly **noisy**

- start from the leaves and work back up

Pruned trees smaller in size, easier to interpret

Ensemble Methods

Ensemble Methods

Suppose you have a **collection** of discriminants *f* (x, w_k), which, individually, perform only **marginally** better than random guessing.

$$f(x) = a_0 + \sum_{k=1}^{K} a_k f(x, w_k)$$

From such discriminants, **weak learners**, it is possible to build highly effective ones by averaging over them:

Jerome Friedman & Bogdan Popescu (2008)

Ensemble Methods

Bagging (bootstrap aggregation)

- Each tree trained on **bootstrap sample** drawn from training set

Random Forest

- Bagging with **randomized trees**
- Random subsets of features used at each split

Boosting

- Each tree trained on a **different weighting** of full training set. Usually used with decision trees but is more general

Random Forest

Random Forest

- L. Breinman, 2001
- Bagging plus:
  - Random subset of features for splitting at each node
- Benefits:
  - excellent accuracy, avoids over-fitting

Boosting

Turns weak learners to strong learners with weighted ensemble of iterative learners

- Adaptation
- Many boosting algorithms
  - differ in how to weight instances
- Benefits: excellent accuracy
- R. Shapire, 1990

Adaptive Boosting

Adaptive Boosting

Train in stages

- Adaptive weights
  - **ADABoost: Freund & Schapire 1997**
- **Misclassified** events get a **larger weight** going into the next training stage
  - Classify with a majority vote from all trees
- Works very well to improve classification power of “greedy” decision trees

Adaptive Boosting

Repeat *K* times:

1. Create a decision tree $f(x, \boldsymbol{w})$
2. Compute its error rate $\boldsymbol{\varepsilon}$ on the ***weighted*** training set
3. Compute $\boldsymbol{\alpha} = \ln(1 - \boldsymbol{\varepsilon}) / \boldsymbol{\varepsilon}$
4. Modify training set: ***increase weight*** of ***incorrectly classified examples*** relative to the weights of those that are correctly classified

Then compute weighted average

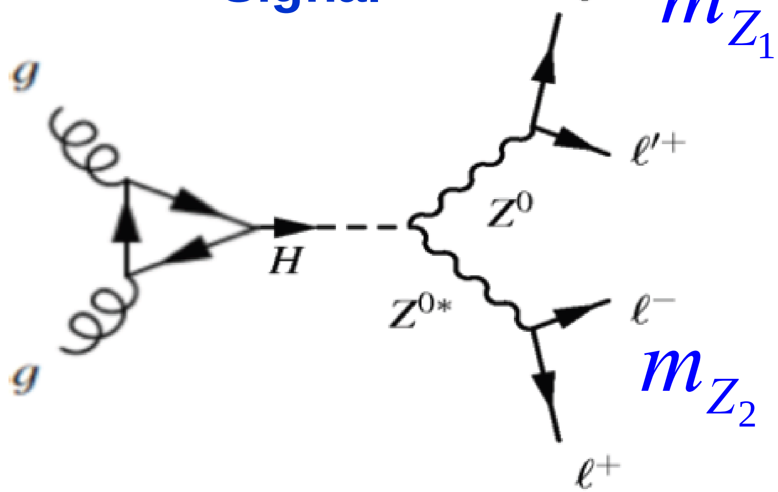
$$f(x) = \sum \alpha_k \, f(x, w_k)$$

Y. Freund and R.E. Schapire (1997)

Illustrative Example

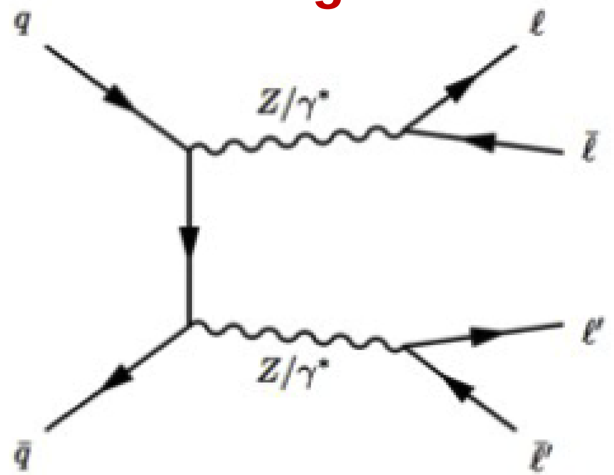
H → ZZ* → 4 leptons

Signal



$pp \to H \to ZZ \to \ell^+\ell^-\ell'^+\ell'^-$

Background

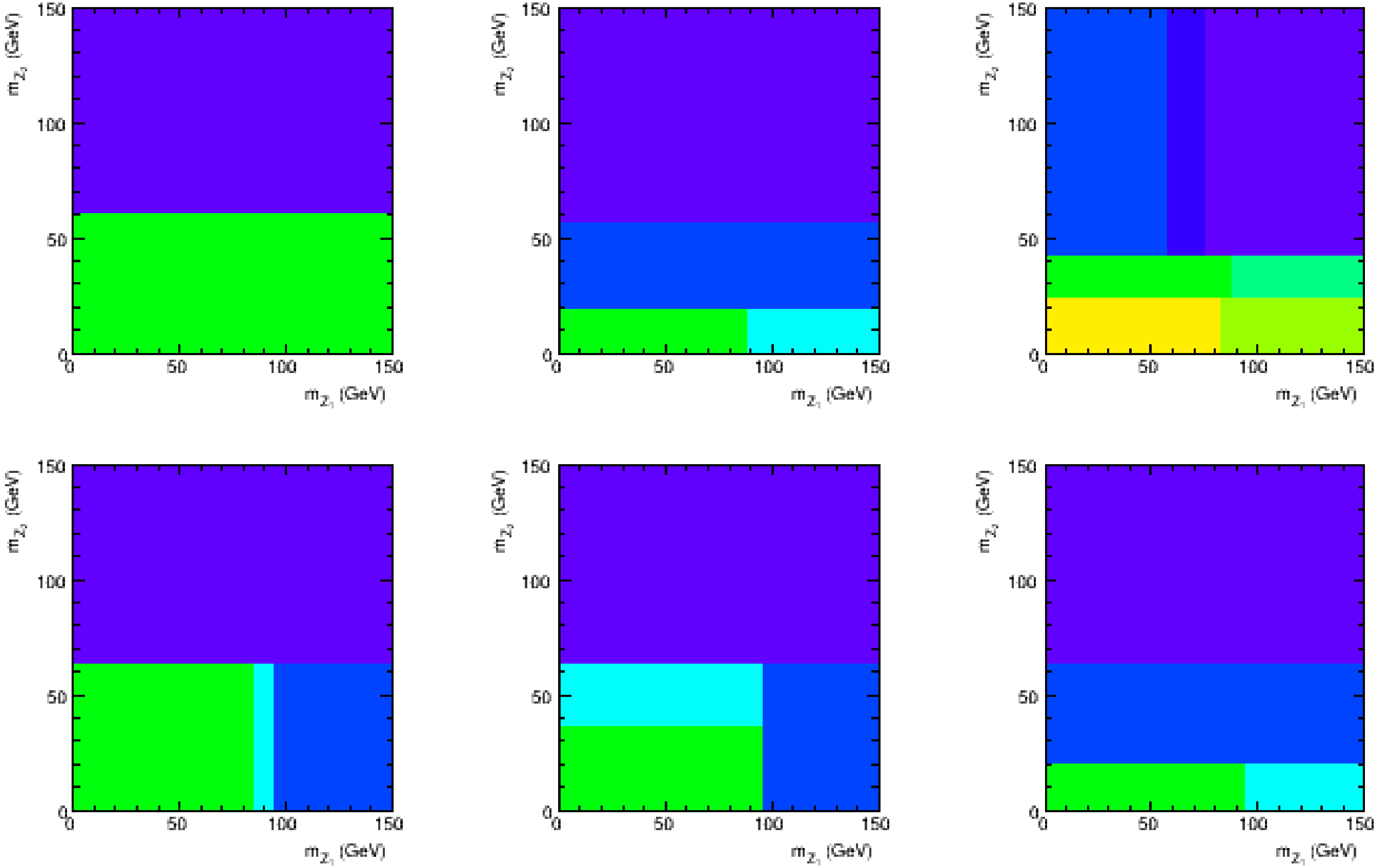


$pp \to ZZ \to \ell^+\ell^-\ell'^+\ell'^-$

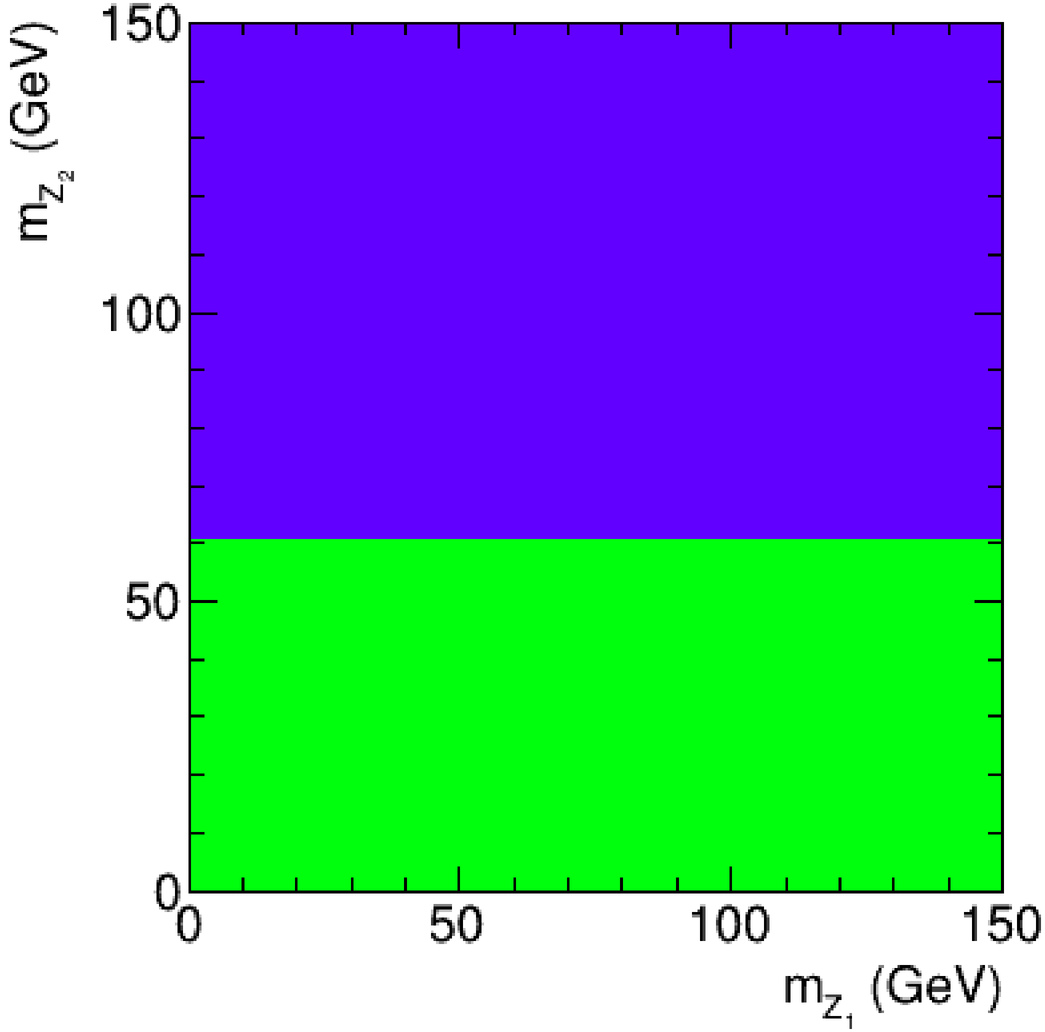
x = (m_{Z1}, m_{Z2})

Credit: H. Prosper

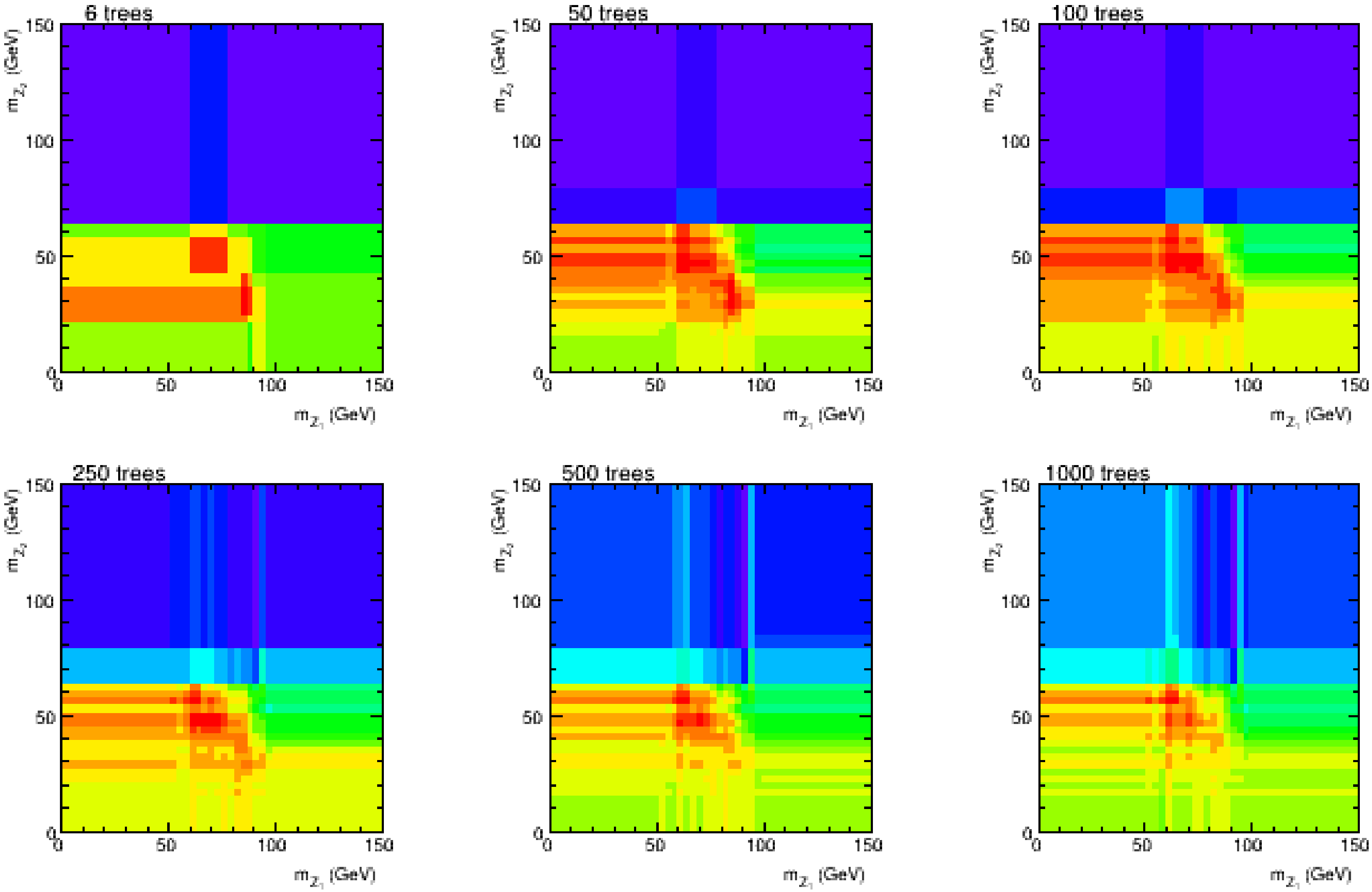
First 6 Decision Trees



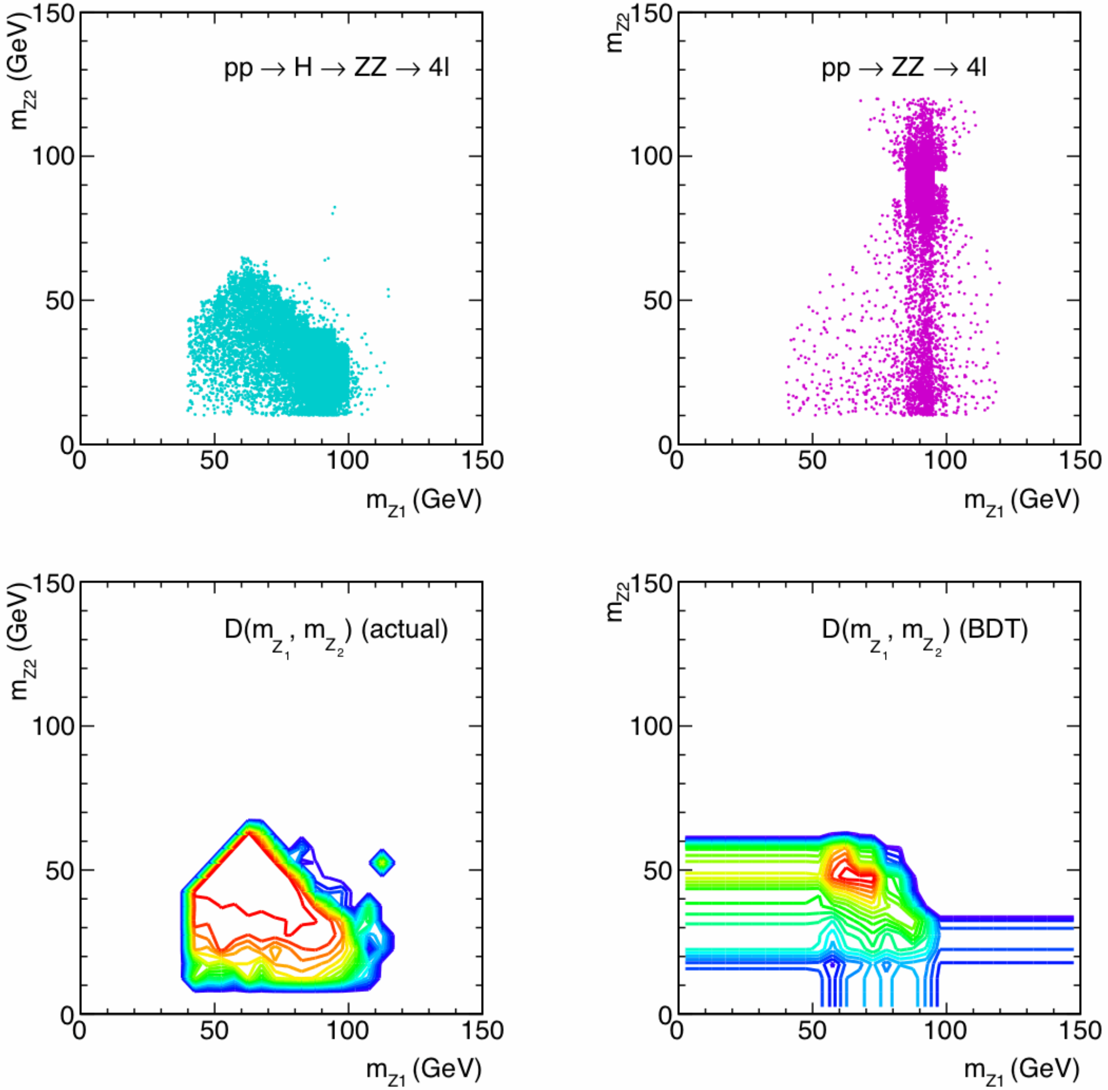
First 100 Decision Trees



Averaging over a Forest



H to ZZ to 4Leptons



Build an Ensemble