GENERALISATION IN MACHINE LEARNING FOR HEP

TOM STEVENSON

IOP CONFERENCE, UNIVERSITY OF SUSSEX
22 MARCH 2016

OUTLINE

- Generalisation
  - Motivation and the Issue
  - Hold-out validation
  - Cross-validation
  - Physics Example
  - Summary

MACHINE LEARNING

Source: deepmind.com

- Wide field:
  - Spam filtering
  - Hand writing recognition
  - Beating human at Go
- Used in HEP to separate small signals from large backgrounds.
- Many different algorithms:
  - Boosted Decision Trees
  - Neural Networks
  - Support Vector Machines

MOTIVATION AND THE ISSUE

- Need confidence that the trained MVA is robust and the performance on unseen samples can be accurately predicted, i.e. generalised.

- This motivates validation techniques which are required for:
  - Model Selection:
    - Most methods have at least one free parameter e.g.
      - BDT - #trees, min node size, etc.
      - SVM - kernel function, kernel parameters, cost, etc.
    - How are these parameters of models "optimally" selected?

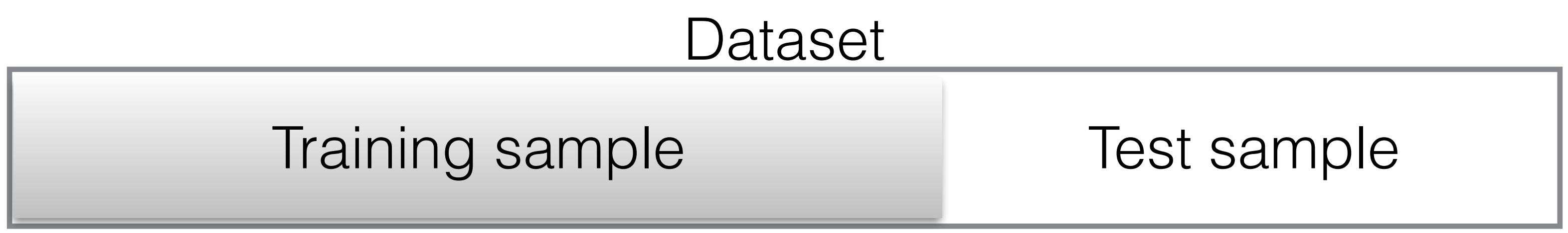
  - Performance Estimation:
    - How does the chosen model perform?
    - Usually true error rate is used (misclassification rate for the entire dataset).

MOTIVATION AND THE ISSUE

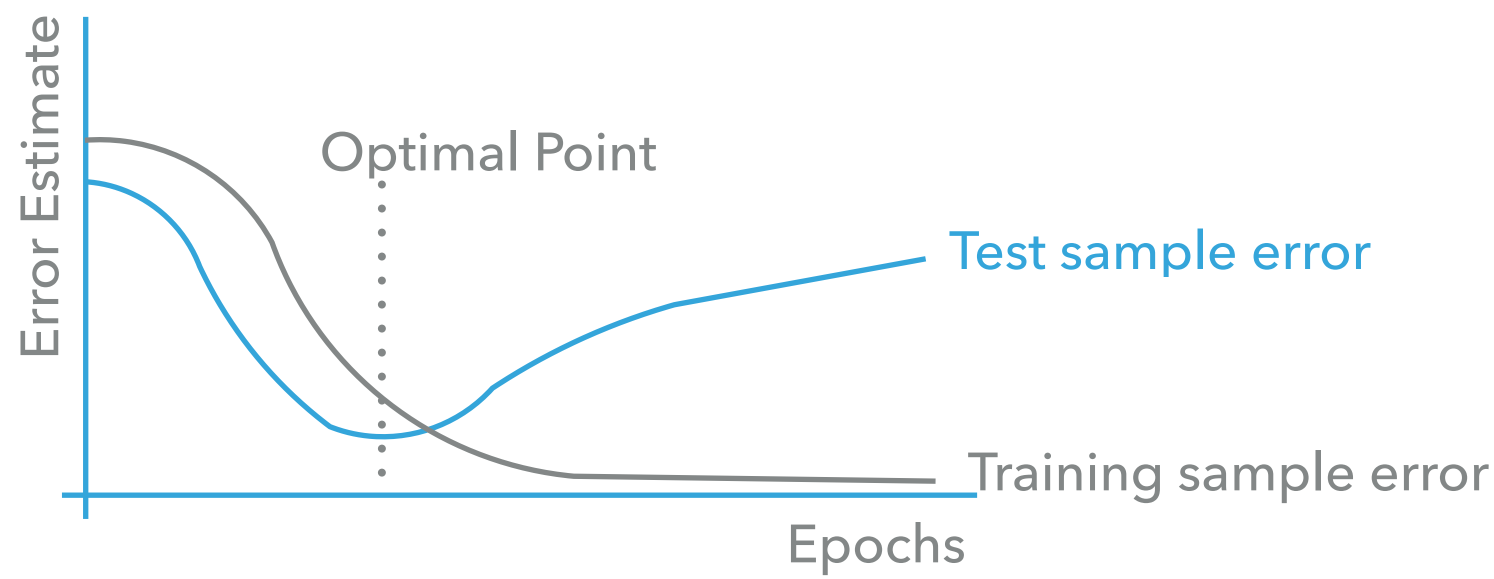
- For an unlimited dataset these issues are trivial, simply iterate through parameters and find model with lowest error rate.
- In reality datasets are smaller than we would like.
- Naïvely use whole dataset to select and train classifier and to estimate error.
  - Leads to overfitting/overtraining as classifier learns fluctuations in the dataset and performs worse on unseen data.
  - Overfitting more distinct for classifiers with large number of tuneable parameters.
  - Also gives overly optimistic estimation of error rate.

HOLD-OUT VALIDATION

- Potential way to overcome these issues is use hold-out technique, splitting the dataset into training and test subsamples.



- Can use these datasets to select "optimal" parameters, for example back-propagation for MLP.



- Can give misleading error estimate depending on how the data is split.

K-FOLD CROSS-VALIDATION

- May not be able to reserve a large portion of data for testing, so hold-out method may not be viable.
- Instead can use k-fold cross-validation:

Dataset

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	...	Fold k

  - Split the dataset into k randomly sampled independent subsets (folds).
  - Train classifier with k-1 folds and test with remaining fold.
  - Repeat k times.
- Advantage of using the whole dataset for testing and training.
- True error rate is then estimated using average error rate:

$$E = \frac{1}{k} \sum_{i=1}^{k} E_i .$$

K-FOLD CROSS-VALIDATION

- How many folds???

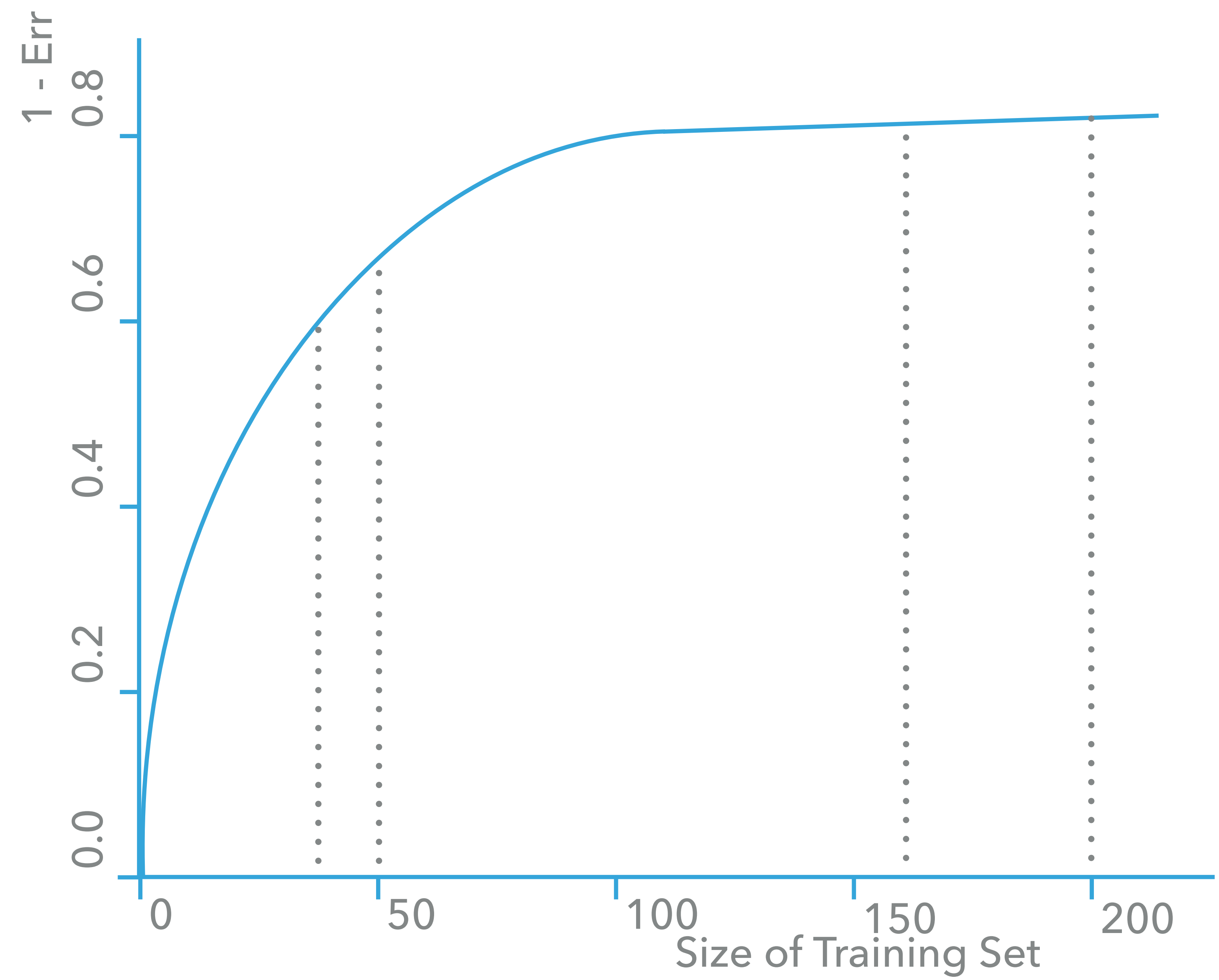
- Large number of folds:
  - Good estimate of average error rate (bias of the estimator is small).
  - Variance of the estimator is large.
  - Computational time is long.

- Small number of folds:
  - Poor estimate of average error rate (bias of the estimator is large).
  - Variance of the estimator is small.
  - Computational time is relatively short.

- In reality choice is motivated by the size of the dataset, i.e. sparse dataset need extreme of leave-one-out method to train on as much data as possible.

K-FOLD CROSS-VALIDATION

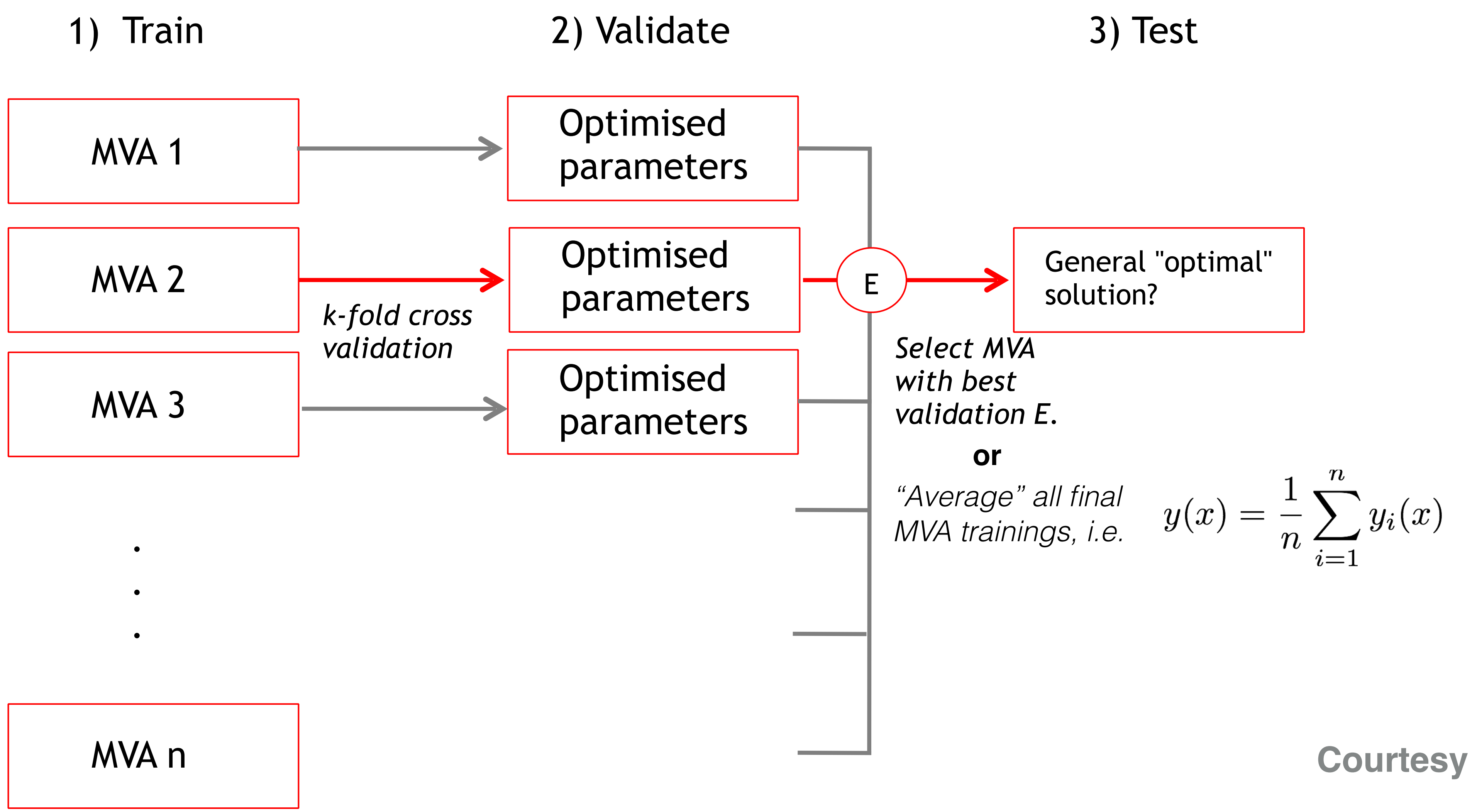
- Hypothetical example:
  - For sample size of 200, 5 fold CV will estimate the error with similar performance on training set of 160 to that of the full sample.
  - However for sample of 50, 5 fold CV will give a larger error than not using CV.



- Common choices are between 5 & 10 folds, however k **should** be determined for the given problem.

K-FOLD CROSS-VALIDATION

- Ideally 3 statistically independent datasets.



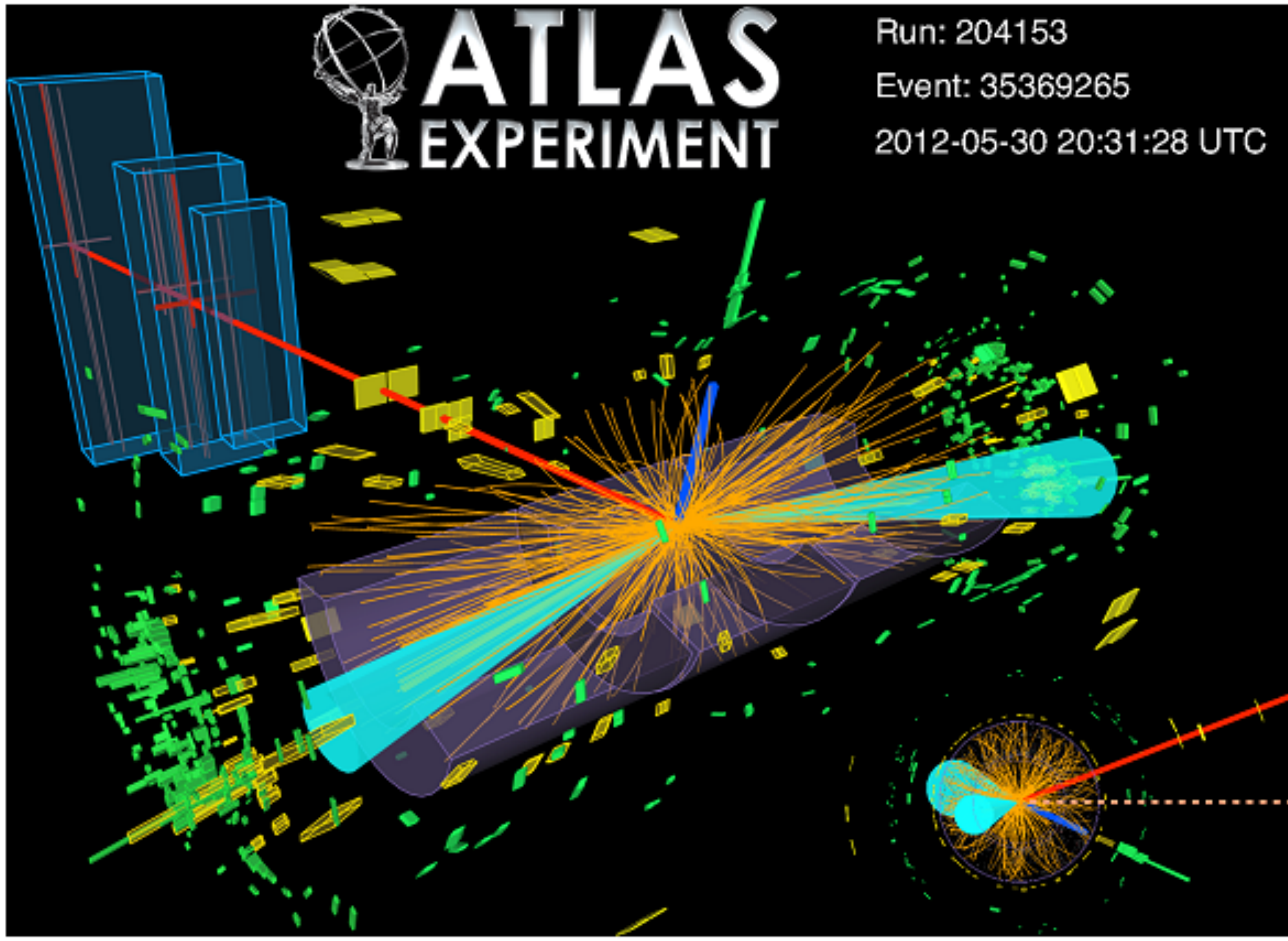
Courtesy of Adrian Bevan

- "Best" performing MVA doesn't necessarily give the desired output.
- Take aggregated output of final trained MVAs on test sample in some form of average.

H→ττ EXAMPLE

- H→ττ Higgs machine learning challenge dataset example.

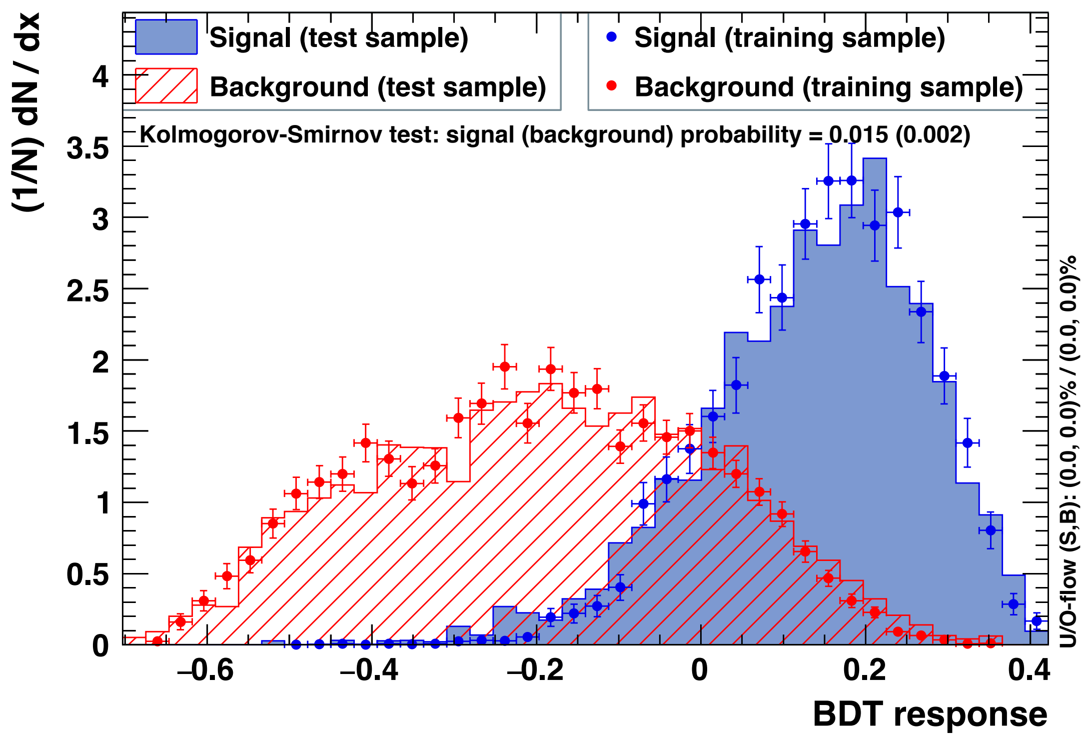
- First 16 variables chosen (not an optimised analysis).
- Following procedure outlined, using macro for TMVA.
- 5000 signal and 5000 background events.



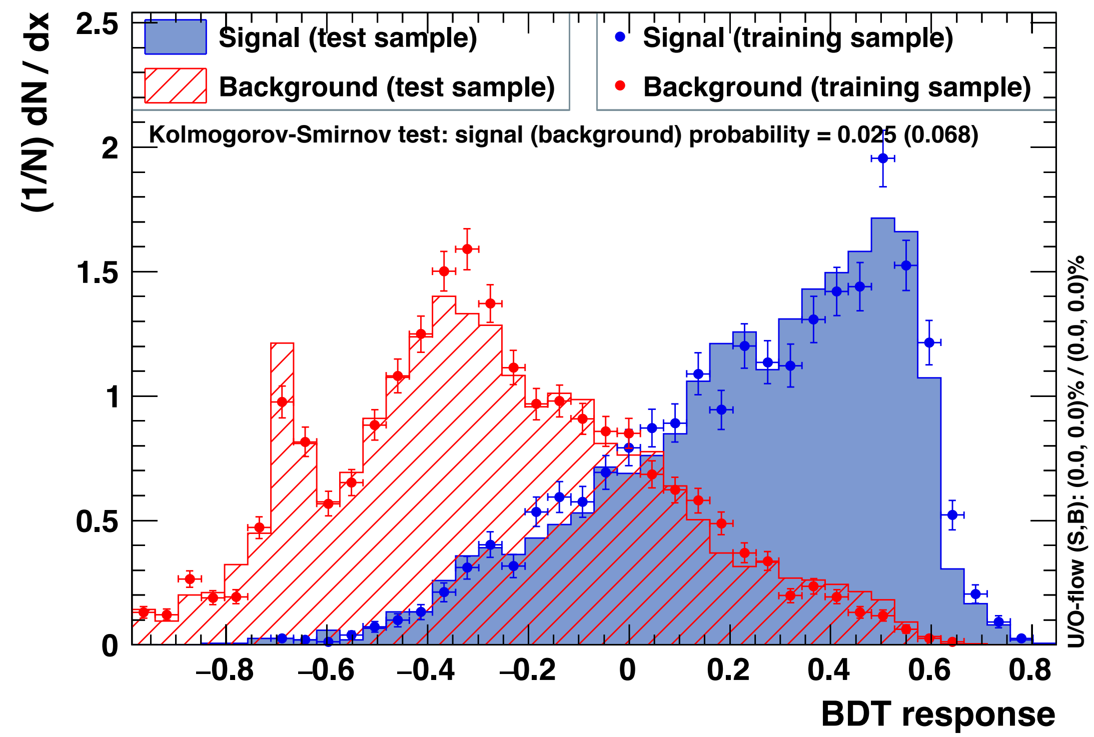
- 3-fold CV BDT presented (next slide) with hold-out validated BDT for comparison.
  - Best performing CV BDT has spiky structure due to picking low number of trees.
  - CV averaged BDT has better agreement between training and testing samples than hold-out BDT.
    - Potentially more generalised.

H → ττ EXAMPLE

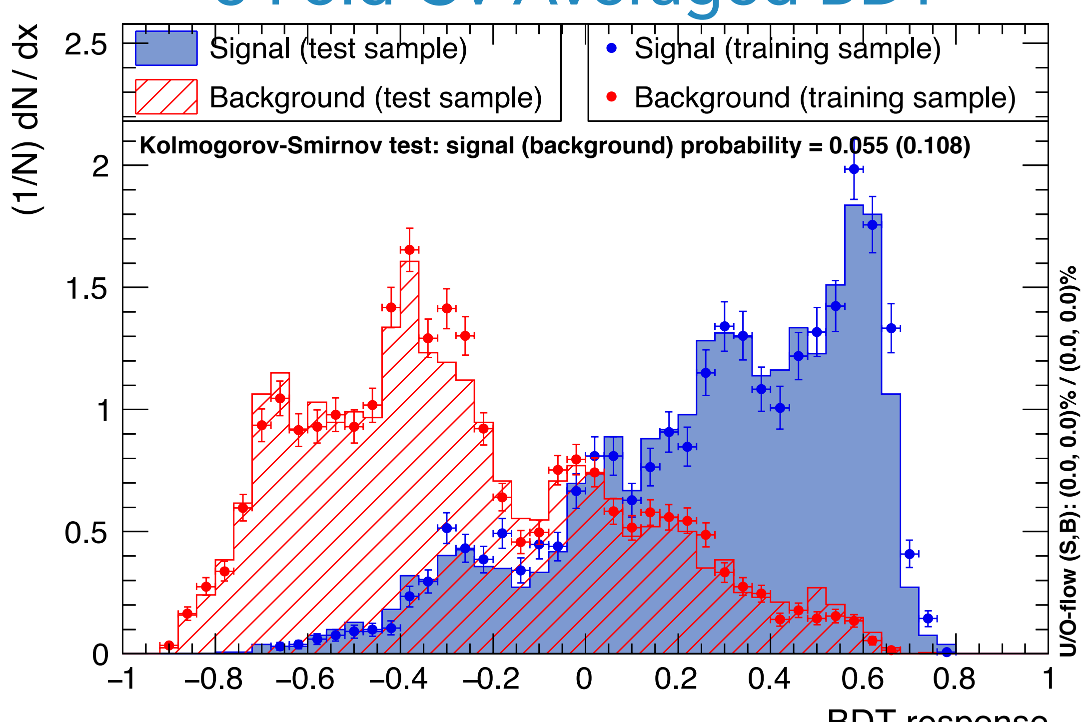
Holdout BDT



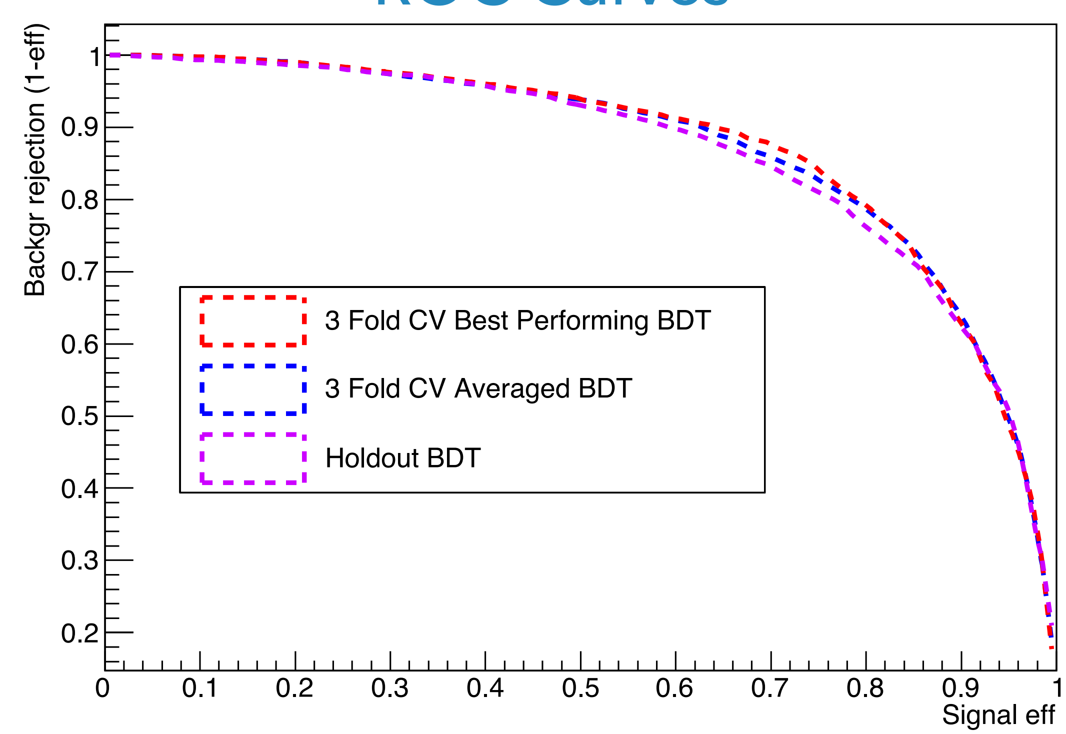
3 Fold CV Best BDT



3 Fold CV Averaged BDT



ROC Curves



SUMMARY

- HEP generally uses hold-out CV.
- k-fold CV used in the wider ML community.
- A multistage training/validation/testing process have been detailed.
- Example macro to perform k-fold CV with TMVA soon available in ROOT release.
- For $H \rightarrow \tau\tau$ example k-fold CV shows improved generalisation when compared with hold-out CV.