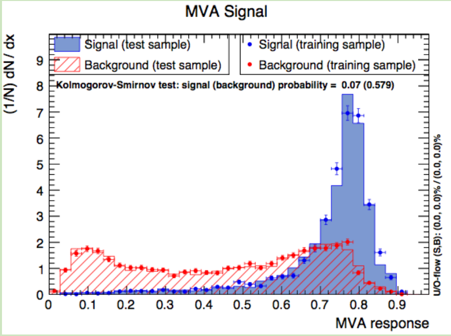
Support Vector Machines and Generalisation in HEP

A. Bevan, R. Gamboa Goñi, J. Hays, T. Stevenson

Queen Mary University of London

Multivariate analysis (MVA) commonplace in HEP



Neural networks (MLP, Deep Learning), boosted decision trees (BDT), matrix element approaches etc.

Important issues to get the best from the algorithms:

Optimization
Selecting the best hyperparameters

Generalization
Testing for and avoiding overtraining

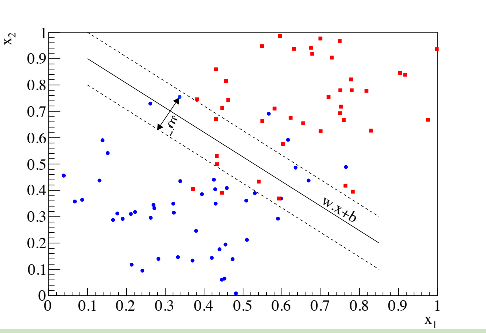
Model selection
Picking between different approaches

Widespread use aided by convenient tools

We're working in several areas:

1) Improved SVM implementation in TMVA [1].
2) Cross validation tools in TMVA.
3) Deep Learning using TensorFlow [2].
4) Generalization and model selection.

(1) Support vector machines (SVM)[3] are widely used outside particle physics but not so much within the field



SVMs classify data using a maximal margin hyperplane mapped from a linear classification problem to a possibly infinite dimensional (dual) hyperspace.

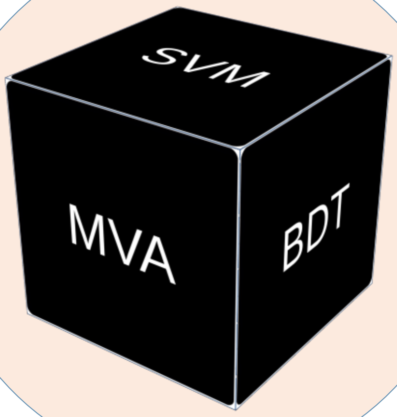
slack ξ_i: distance from hyperplane to ith support vector

cost C : tunable weight penalty for misclassification

Kernel Function K(x,y): maps input space into higher dimensional feature space where problem may be linearly separable.

Kernel optimised for each problem. Only points near the decision boundart contribute significantly so potentially less sensitive to overtraining.

New implementation in TMVA git repo with expanded set of kernel functions [4].



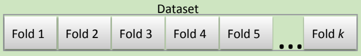
(2) Cross validation (CV) used for improving model performance [5,6]

Understanding model performance is vital.

Training data cannot be trusted – biased.

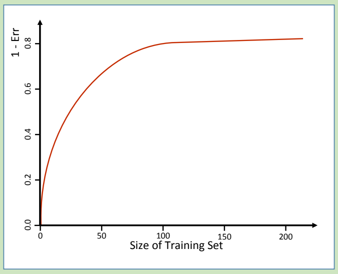
Hold-out method typical in HEP reserves substantial amount of data for testing.

K-fold CV – repeated trainings for overlapping data sets
Holding out one fold for testing each time.



Dataset

Fold 1 | Fold 2 | Fold 3 | Fold 4 | Fold 5 | … | Fold *k*



Combine repeated test results to estimate algorithm performance.

Pro:
Make use of all the data.

Con:
Involves multiple trainings so can be computational expensive.

On-going work:
CV tools for TMVA [4].

Investigating running algorithm training on GPUs.

(3) Generalization and Model Selection

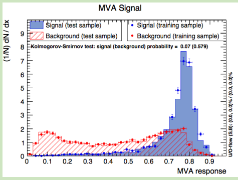
To choose between models – need estimate of generalized performance.

Different models on same data not guaranteed to rank models correctly – CV can help (variance can be large) [7,8].

Overtraining harms model performance leads to incorrect choices or biases.

Hold out method gives no indication of variance – resampling techniques like CV can help.

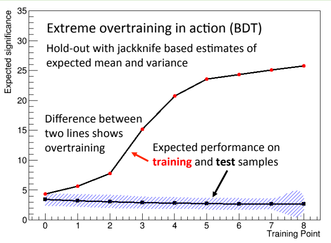
Often we're interested in distributions rather than cuts. Need to establish generalisation somehow. Binned KS test used by TMVA is problematic.



Need a measure appropriate to the problem – some differences between test and training might have no impact others might be critical.

Ongoing work investigating:

*** Measures of variance;**
*** Algorithm selection;**
*** Hypothesis test for generalisation.**



[1] A. Hoecker et al., PoS ACAT 040 (2007) 040, [2] TensorFlow; https://www.tensorflow.org, [3] C. Cortes and V. Vapnik, Machine Learning Volume 20 Issue 3 (1995), [4] T. J. Stevenson et al., Proceedings of ACAT 2016, [5] D. M. Allen, Technometrics, 16, 125-127 (1974), [6] M. Stone, Journal of the Royal Statistical Society B 36, 111-116 (1974), Journal of the Royal Statistical Society B 39, 44-47 (1977), [7] B. Efron and R. Tibshirani. Journal of the American Statistical Association, 92, 438, 548-560 (1997), [8] Y. Bengio and Y. Grandvalet, Journal of Machine Learning Research 5, 1089-1105 (2004).