Queen Mary
University of London

DR ADRIAN BEVAN

MULTIVARIATE ANALYSIS AND ITS USE IN HIGH ENERGY PHYSICS

6) SUPPORT VECTOR MACHINES (SVMs)

Lectures given at the department of Physics at CINVESTAV, Instituto Politécnico Nacional, Mexico City
28th August - 3rd Sept 2018

LECTURE PLAN

- Introduction
- Hard margin SVM
- Soft margin SVM
- Kernel function
- Examples:
    - Checker board
    - H→$\tau^+\tau^-$ at ATLAS (Kaggle Higgs data challenge)
    - HH→bb$\tau^+\tau^-$ at ATLAS
    - stop searches at CMS
- Summary and miscellaneous notes
- Suggested tools
- Suggested reading

INTRODUCTION

- SVMs are widely used:

Astrophysics

Bioinformatics

Risk Analysis

Medicine (e.g. predicting disease/imaging etc.)

HEP

Machine Learning Community/Computer Science

Finance

Speech recognition

Handwriting recognition

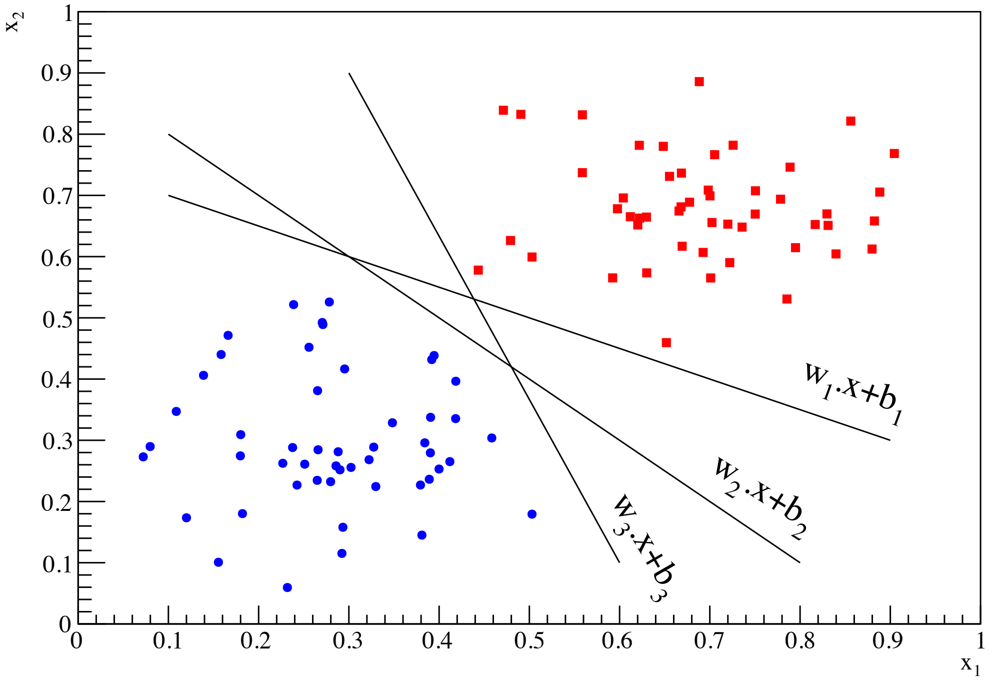
Robotics

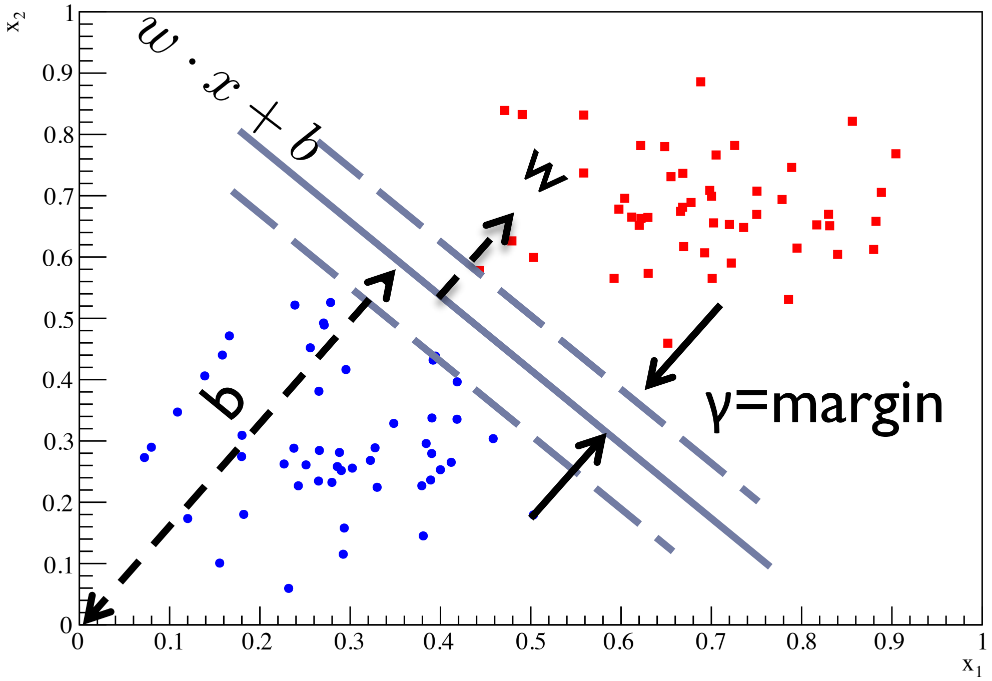
- HEP problems are low dimensional simple use cases compared with issues being addressed for some of the existing fields using these algorithms.

HARD MARGIN SVM

- Identify the support vectors (SVs): these are the points nearest the decision boundary.

- Use these to define the hyperplane that maximises the margin (distance) between the optimal plane and the SVs.





- If we can do this with a SVM – we would simply cut on the data to separate classes of event.

HARD MARGIN SVM: PRIMAL FORM

- Optimise the parameters for the maximal margin hyperplane with:

$$\arg\min_{w,b} \frac{1}{2}||w||^2$$

- such that $y_i(w \cdot x_i - b) \geq 1$ (y_i is called the functional margin)

- Equivalent to solving the following optimisation problem:

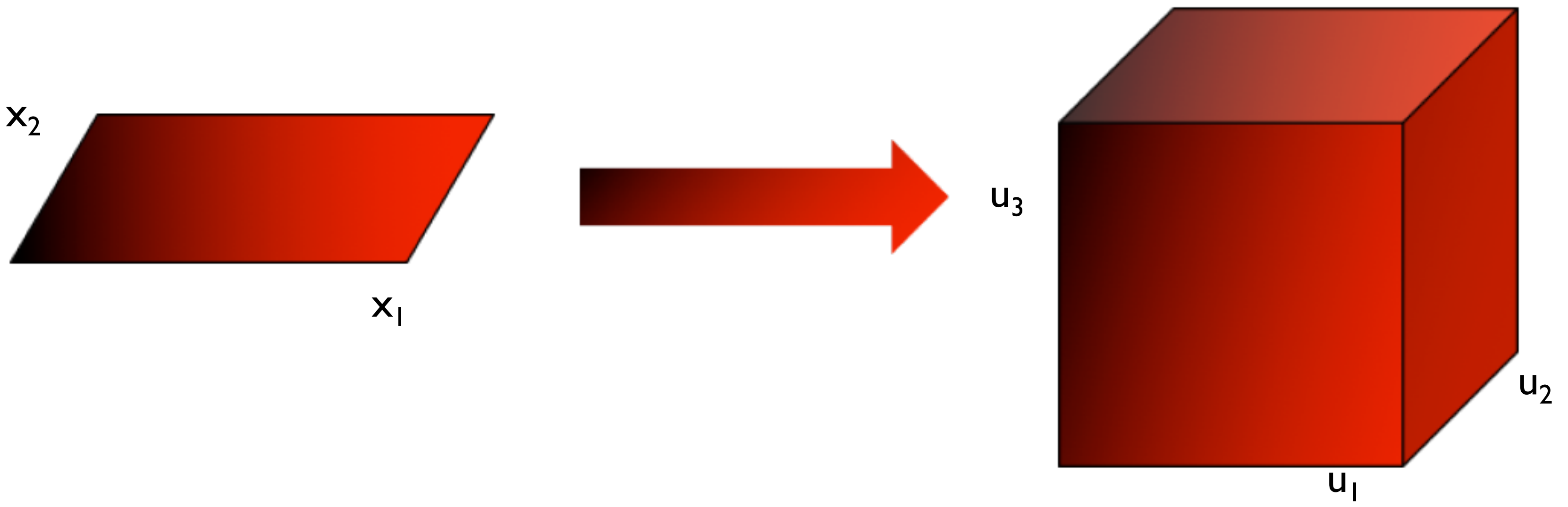
$$\arg\min_{w,b}\max_{\alpha\geq 0}\left[\frac{1}{2}||w||^2 - \sum_{i=1}^{n}\alpha_i[y_i(w \cdot x_i - b) - 1]\right]$$

- Where: $w = \sum_{i=1}^{n}\alpha_i y_i x_i$ and $b = \frac{1}{N_{SV}}\sum_{i=1}^{n}(w \cdot x_i - y_i)$

HARD MARGIN SVM: KERNEL FUNCTIONS

- We can introduce the use of a Kernel Function (KF) to implicitly map from our input feature space X to some potentially higher dimensional dual feature space F.
- Define the function: $K(x, y) = \langle \phi(x) \cdot \phi(y) \rangle$

$X \in \{x_1, x_2\}$



$F \in \{u_1, u_2, u_3\}$

- We don't need to know the details of the mapping; this is the "kernel trick".

B. Scholkopf and A. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond. MIT Press, 2002.*

HARD MARGIN SVM: KERNEL FUNCTIONS

- We can introduce the use of a Kernel Function (KF) to implicitly map from our input feature space X to some potentially higher dimensional dual feature space F.
- Define the function: $K(x, y) = \langle \phi(x) \cdot \phi(y) \rangle$

e.g.

$$x \in \mathbb{R}^n \quad \longrightarrow \quad F \in \{\phi(x) | x \in X\}$$

- We don't need to know the details of the mapping; this is the "kernel trick".

B. Scholkopf and A. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond. MIT Press, 2002.*

HARD MARGIN SVM: DUAL FORM

- The problem can be solved in the dual space by minimising the Lagrangian for the Lagrange multipliers α_i :

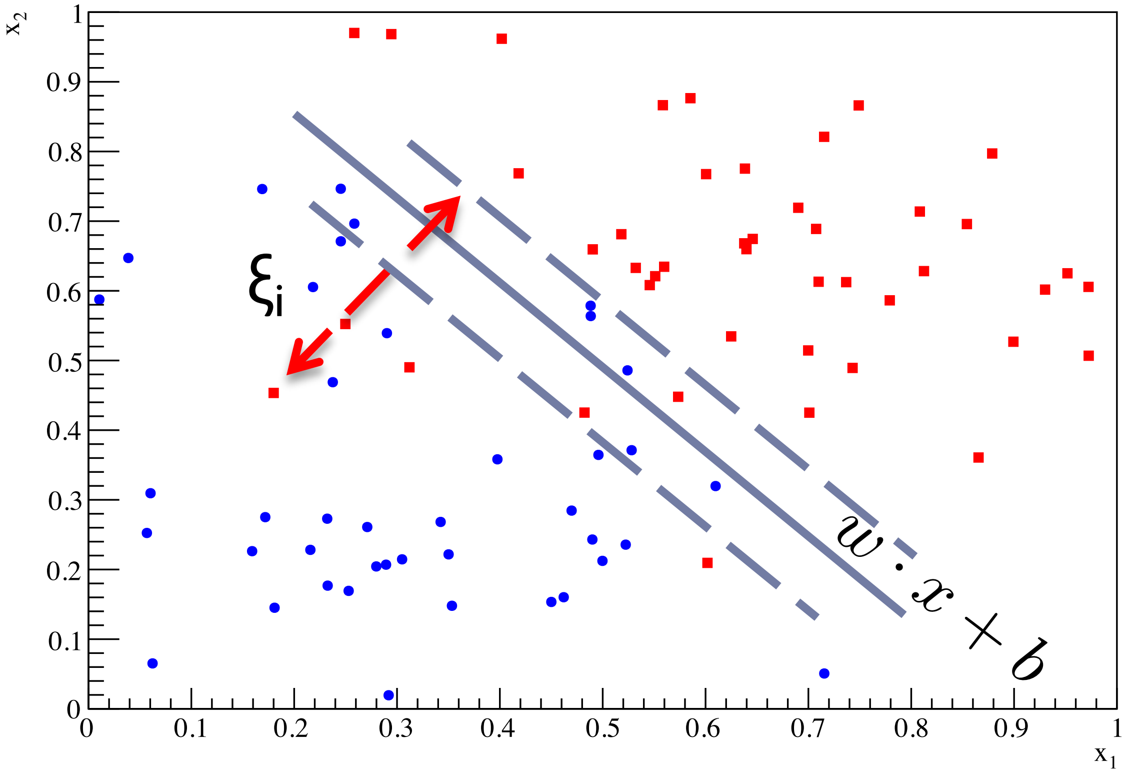
$$\begin{aligned}\widetilde{L}(\alpha) &= \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j x_i^T x_j \\ &= \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j K(x_i, x_j).\end{aligned}$$

Dot product KF

- Such that: $\alpha_i \geq 0$ and $\sum_{i=1}^{n} \alpha_i y_i = 0$.
- α_i are non-zero for SVs only.
- The sum provides a constraint equation for optimisation.

SOFT MARGIN SVM

- Relax the hard margin constraint by introducing mis-classification:
  - Describe by slack (ξi) and cost (C) parameters.
  - Alternatively describe mis-classification in terms of loss functions.
  - These are iust wavs to describe the error rate.



ξi = distance between the hyper-plane defined by the margin and the ith SV (i.e. now this is a mis-classified event).

Cost (C) multiplies the sum of slack parameters in optimisation.

MVA architecture complexity is encoded by the KF.

- These are much more useful!

SOFT MARGIN SVM

- The Lagrangian to optimise simplifies when we introduce the slack parameters:

$$\widetilde{L}(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

- Where $0 \leq \alpha_i \leq C$

- and as before we constrain:

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

The optimisation problem in dual space is essentially the same for the hard and soft margin SVMs.

- The algorithm is designed to focus on reducing the impact of misclassified events; again using those closest to the decision boundary to determine that boundary.

KERNEL FUNCTIONS

- The KF, K(x,y), extends the use of inner products on data in a vector space to a transformed space where

$$K(x,y) = \langle \phi(x) \cdot \phi(y) \rangle$$

- The book by
  - *Nello Cristianini and John Shawe-Taylor, called Support Vector Machines and other kernel-based learning methods. Cambridge University Press, 2000 (and references therein)*
- *discusses a number of KFs and the conditions required for these to be valid in the geometrical representation that SVMs are constructed from.*
- Here I'll focus on the main points and give a few examples of KFs (ones that are implemented in TMVA).

KERNEL FUNCTIONS: RADIAL BASIS FUNCTION (RBF)

- Commonly used KF that maps the data from X to F.

- Distance between two support vectors is computed and used as an input to a Gaussian KF.

- For two data x and y in X space we can compute K(x, y) as

$$\boldsymbol{K(x, y) = e^{-||x-y||^2/\sigma^2}}$$

- One tuneable parameter in mapping from X to F; given by Γ=1/σ^2.

KERNEL FUNCTIONS: MULTI GAUSSIAN KERNEL

- Extend the RBF function to recognise that the bandwidth of data in problem space can differ for each input dimension; i.e. the norm of the distance between two support vectors can result in loss of information.

- Overcome this by introducing a $\Gamma_i=1/\sigma_i$ for each dimension:

$$K(x,y) = \prod_{i=1}^{\dim(X)} e^{-||x_i-y_i||^2/\sigma_i^2}$$

- Down side … we increase the number of parameters that need to be optimally determined for the map from X to F.

KERNEL FUNCTIONS: MULTI GAUSSIAN KERNEL

- The multi-gaussian kernel does not include off-diagonal terms that would allow for accommodation of correlations between parameters.
  - De-correlate the input feature space to overcome this deficiency, or alternatively one could implement a variant of this kernel function using:

$$K(x,y) = e^{-(x-y)^T\Sigma^{-1}(x-y)}$$

  - Here Σ is an n x n matrix corresponding to the covariance matrix for the problem.
  - However this would be very computationally expensive to optimise (and is **not** implemented in TMVA).

KERNEL FUNCTIONS: POLYNOMIAL

- There are many different types of polynomial kernel functions that one can study.
- A common variant is of the form:

$$K(x,z) = (\langle x \cdot z \rangle + c)^d = \left(\sum_{i=1}^{\ell} x_i z_i + c \right)^d$$

- c and d are tuneable parameters.
- The sum is over support vectors (i.e. events in the data set for a soft margin SVM).

KERNEL FUNCTIONS: PRODUCTS AND SUMS

- Valid (Mercer) kernels satisfy Mercer's conditions(*). This allows us to construct new kernels from known Mercer kernels that are products and sums. J. Mercer. Phil.Trans.Roy.Soc.Lond., A209:415, 1909.
  - The sum of Mercer KFs is a valid KF.
  - The product of Mercer KFs is a valid KF.

* Mercer's conditions require that the Gramm matrix formed from SVs is positive semi-definite. This is a consequence of the geometric interpretation of SVMs given x is real. Modern extensions of the SVM construct allow for complex input spaces, and for example can be based on Clifford algebra to accommodate this extension.

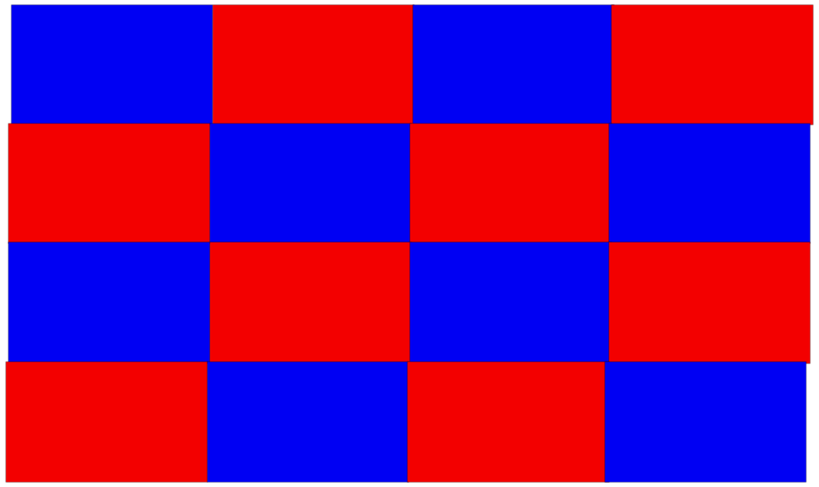
Complex input spaces are of interest for electronic engineering problems.

N.B. It is conceivable that one could be interested in using these if an amplitude analysis were to be written using SVMs to directly extract phase and magnitudes... but that could also be incorporated by mapping the complex feature space element into a doublet of reals.

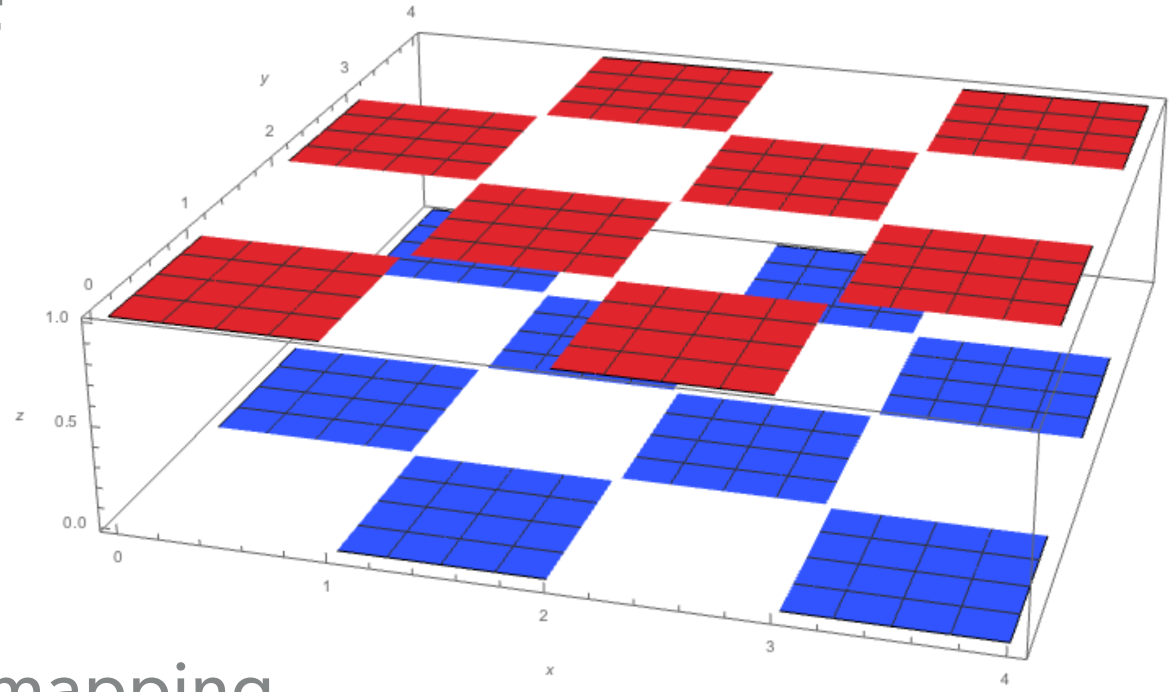
EXAMPLES: CHECKER BOARD

- Generate squares of different colour.
- Use SVM to classify the pattern into +1 and −1 targets.
- Hard margin SVM problem; but can solved for using soft margin SVM.
- Not easy to solve in 2D (x, y) with a linear discriminant, but e.g. a 3D space of (x, y, colour) allows us to separate the squares.

$X \mapsto F$



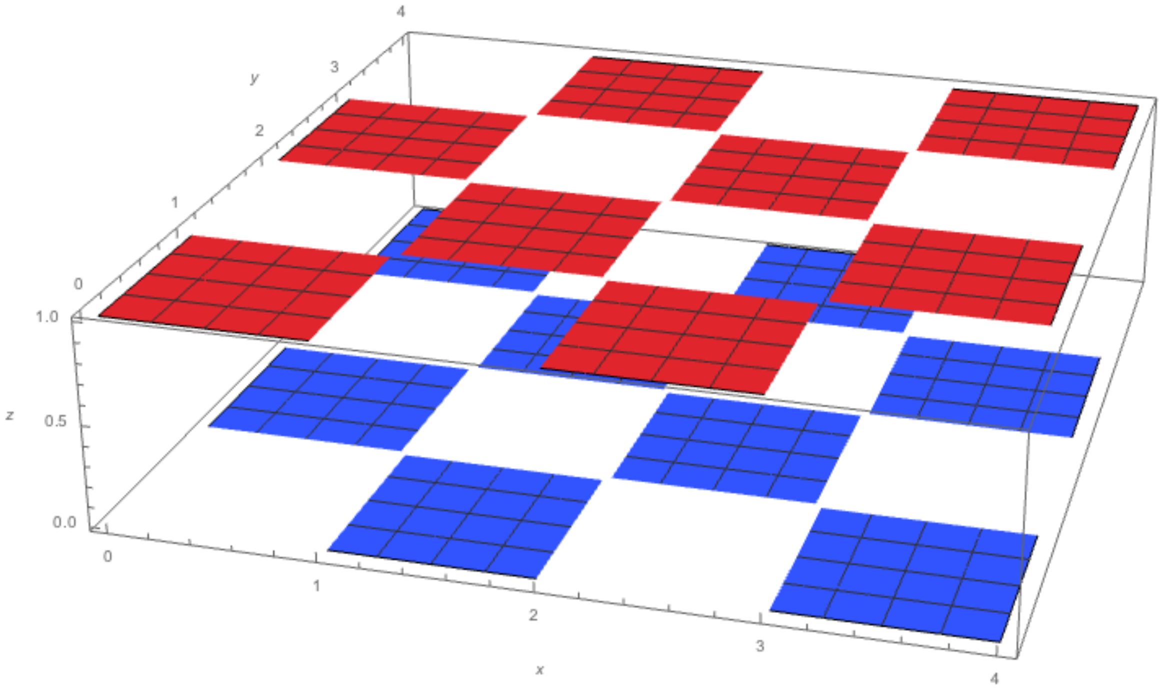
$\mapsto$



- Want to find a KF that approximates this mapping.

EXAMPLES: CHECKER BOARD

- Generate 1000 events in the blue and red squares and give each event x and y values.



This is the ideal feature space that we would like to implicitly map into.

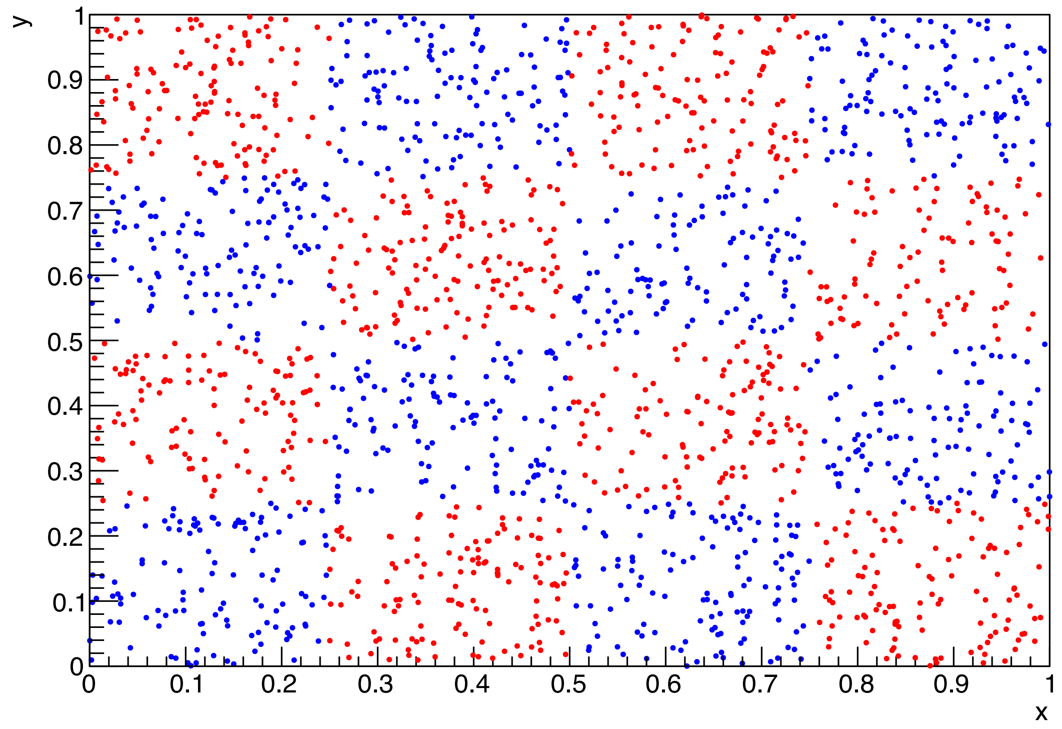
Because we implicitly do the mapping via choice of KF, in practice we don't explicitly map into this space; but we implicitly map into another space that we hope will be approximately topologically equivalent.

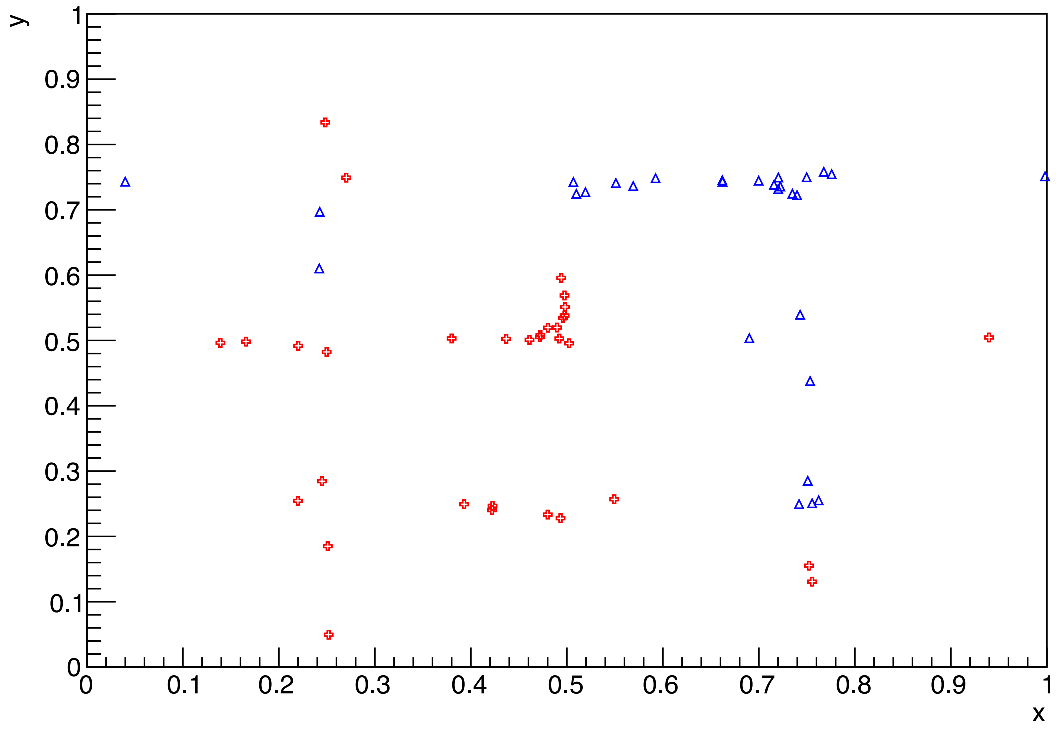
- e.g. Use a multi-Gaussian kernel function with $\Gamma_1=1$, $\Gamma_2=2$ and cost of 10^4 (not optimised) to see what separation we can obtain.

EXAMPLES: CHECKER BOARD

- Correctly classified events

Incorrectly classified events

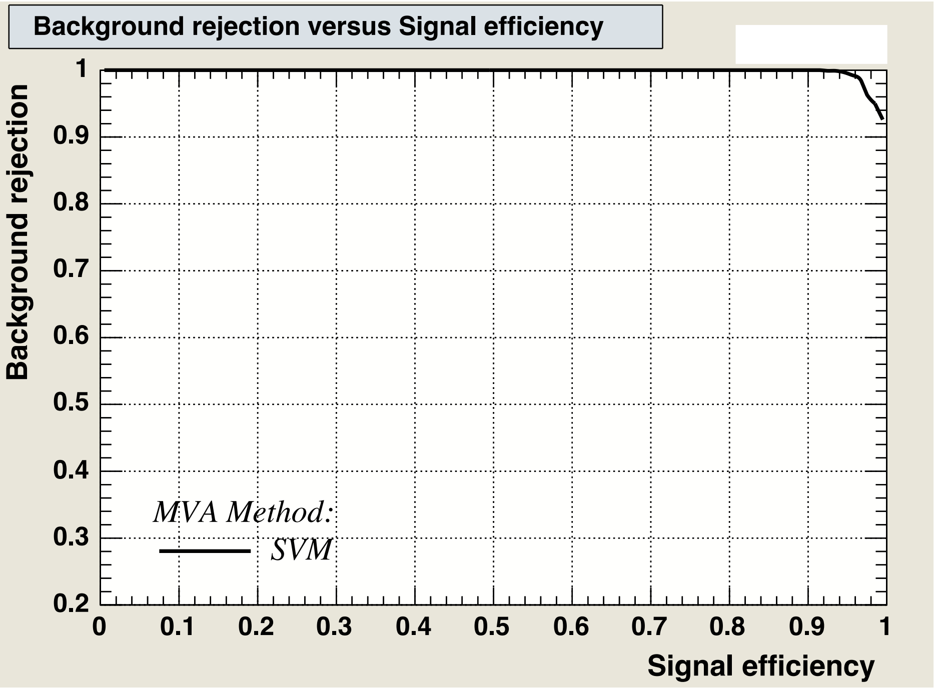


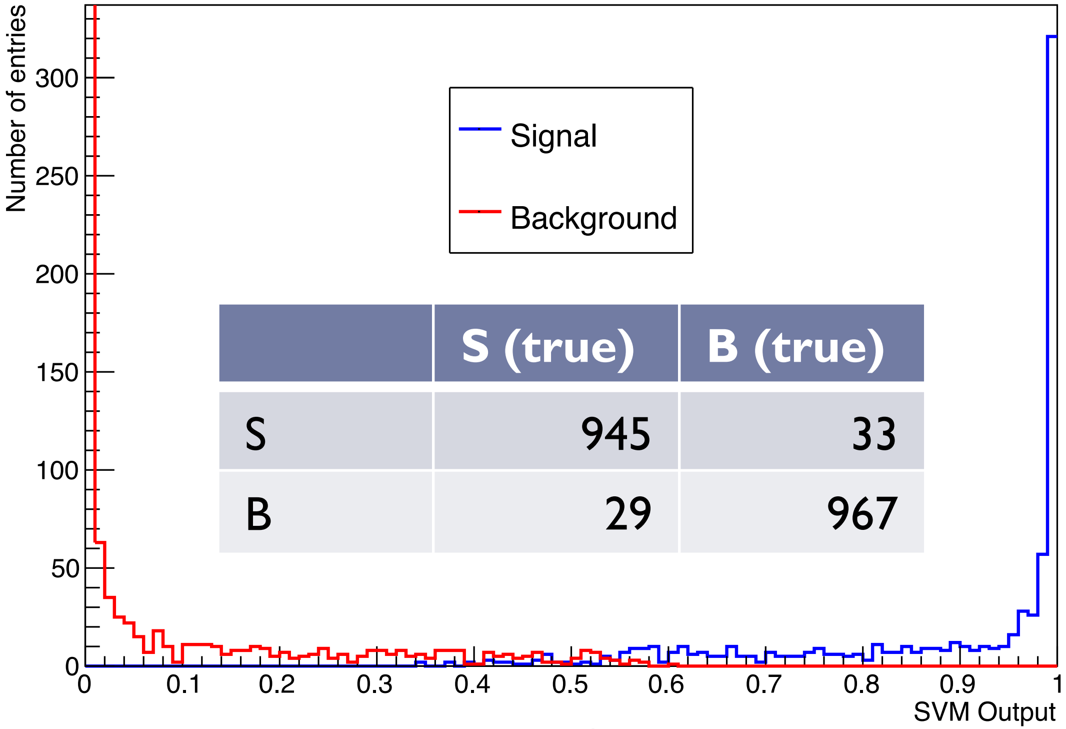


- Signal mis-classification rate ~3.3%.
- Background mis-classification rate ~3.7%.

EXAMPLES: CHECKER BOARD

- The confusion matrix ([in-]correctly classified events) for this example shows a high level of correct classification:





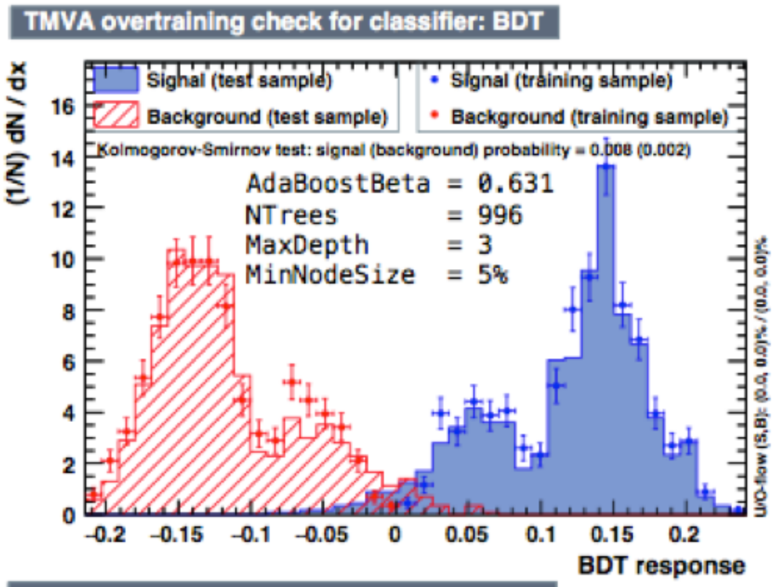
	S (true)	B (true)
S	945	33
B	29	967

- This SVM does a good job of separating signal from background.
- An optimised output would provide a better solution.
- BDTs and NNs work well with this kind of problem as well.

EXAMPLES: CHECKER BOARD

- Optimised results for comparison: Very similar responses.

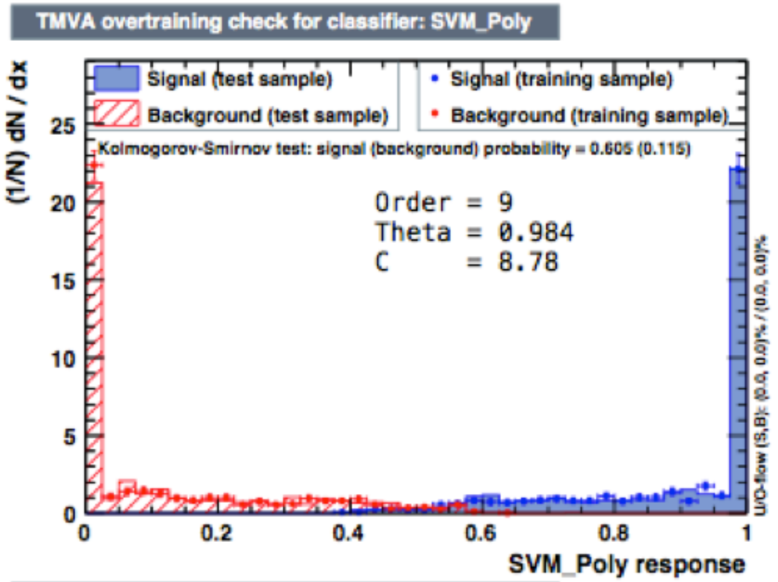
BDT


TMVA overtraining check for classifier: BDT

AdaBoostBeta = 0.631
NTrees = 996
MaxDepth = 3
MinNodeSize = 5%

Kolmogorov-Smirnov test: signal (background) probability = 0.008 (0.002)

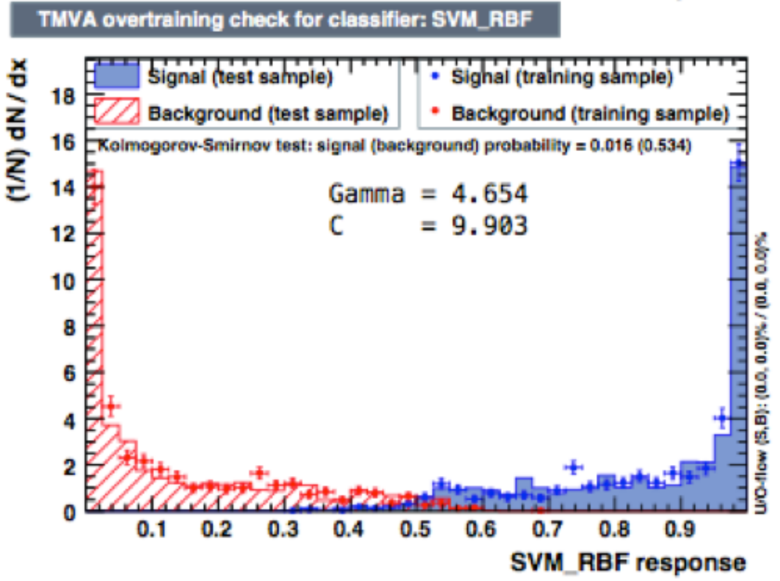
SVM Polynomial (1)


TMVA overtraining check for classifier: SVM_Poly

Order = 9
Theta = 0.984
C = 8.78

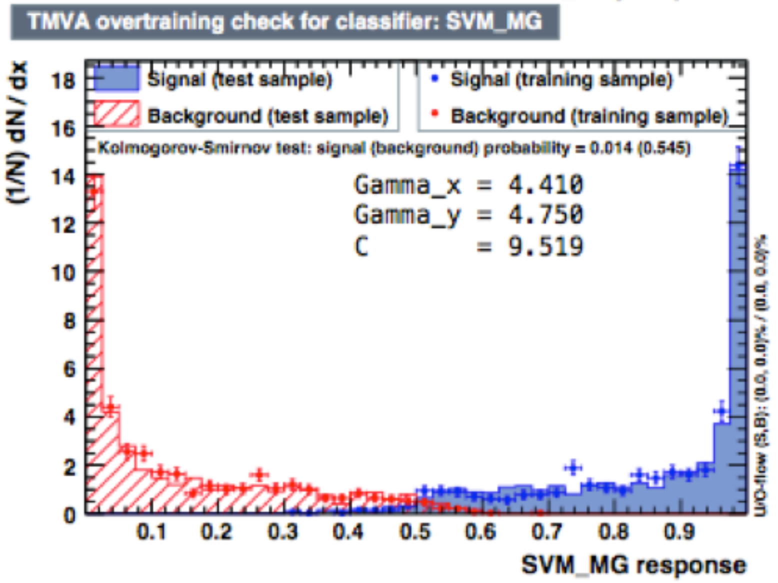
Kolmogorov-Smirnov test: signal (background) probability = 0.605 (0.115)

SVM RBF (2)


TMVA overtraining check for classifier: SVM_RBF

Gamma = 4.654
C = 9.903

Kolmogorov-Smirnov test: signal (background) probability = 0.016 (0.534)


TMVA overtraining check for classifier: SVM_MG

Gamma_x = 4.410
Gamma_y = 4.750
C = 9.519

Kolmogorov-Smirnov test: signal (background) probability = 0.014 (0.545)

SVM Multi-Gaussian (3)

Trained using the hold out method of cross validation (what is normally done in TMVA), with optimised hyper-parameters.

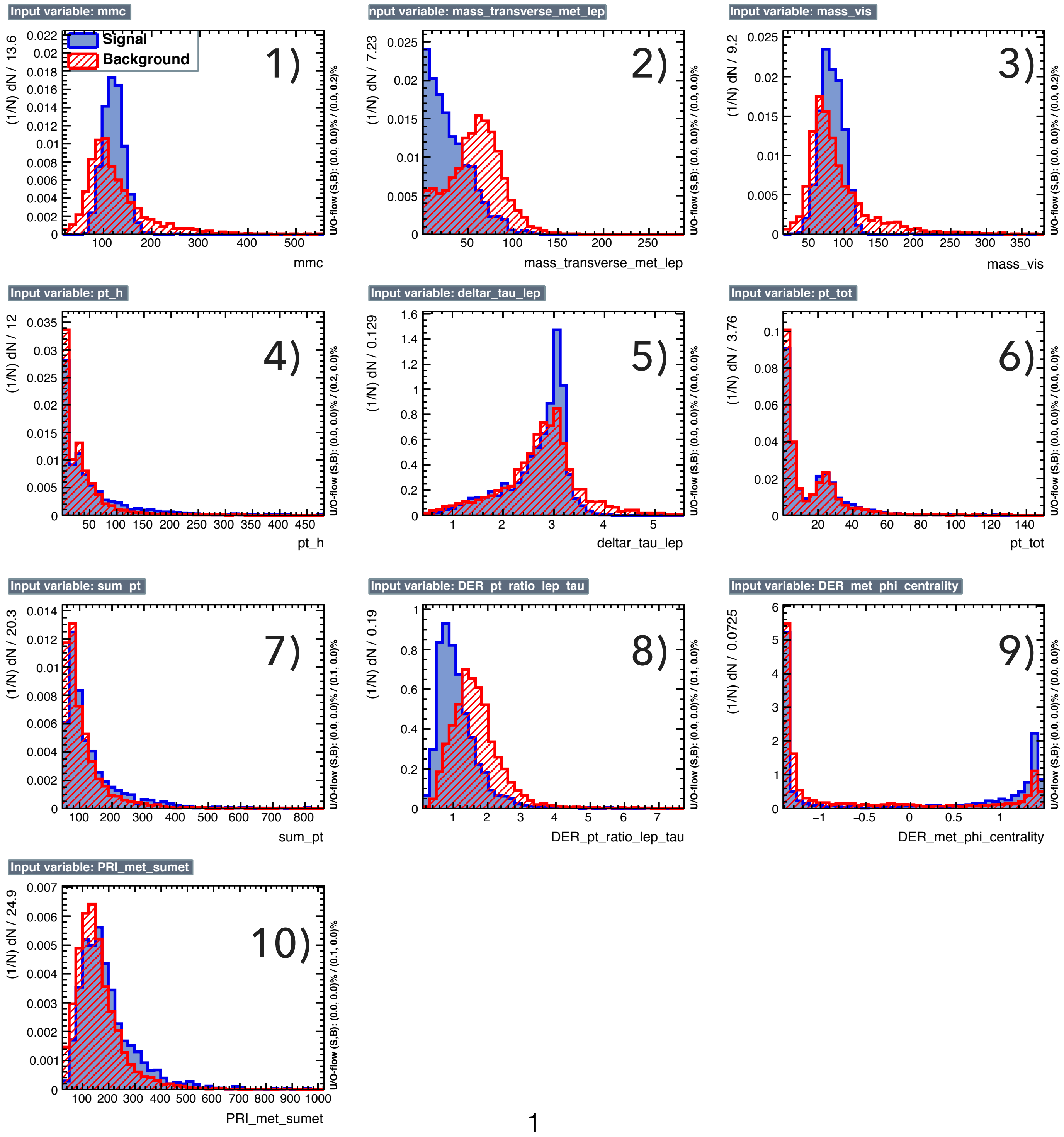
EXAMPLES: H→$\tau^+\tau^-$ (HIGGS KAGGLE DATA CHALLENGE)

- Use the Kaggle data challenge sample of signal and background events. LHC data (from ATLAS).
- Packaged up in a convenient format (CSV file).
- Sufficient description of variables provided for non-HEP users to apply machine learning (ML) techniques to HEP data.
- Real application to compare performance for different KFs and different MVAs.

https://www.kaggle.com/c/higgs-boson

EXAMPLES: H→$\tau^+\tau^-$ (HIGGS KAGGLE DATA CHALLENGE)

- Use 10 variables as inputs; 20K events.



1) MMC
2) transverse mass between MET and lep
3) Visible invariant mass of H
4)$p_T(H)$
5) R between τ_{had} and lepton
6) $p_T(tot)$
7) Σp_T
8) $p_T(lepton)/p_T(had\ \tau)$
9) MET ϕ centrality
10) ET_{total}

This selection of variables is not optimised, and is selected in order to show a physics example for illustrative purposes.

EXAMPLES: H→$\tau^+\tau^-$ (HIGGS KAGGLE DATA CHALLENGE)

- NOTE: this is an illustrative example – not a fully optimised analysis of the sample; hyper-parameters are optimised.

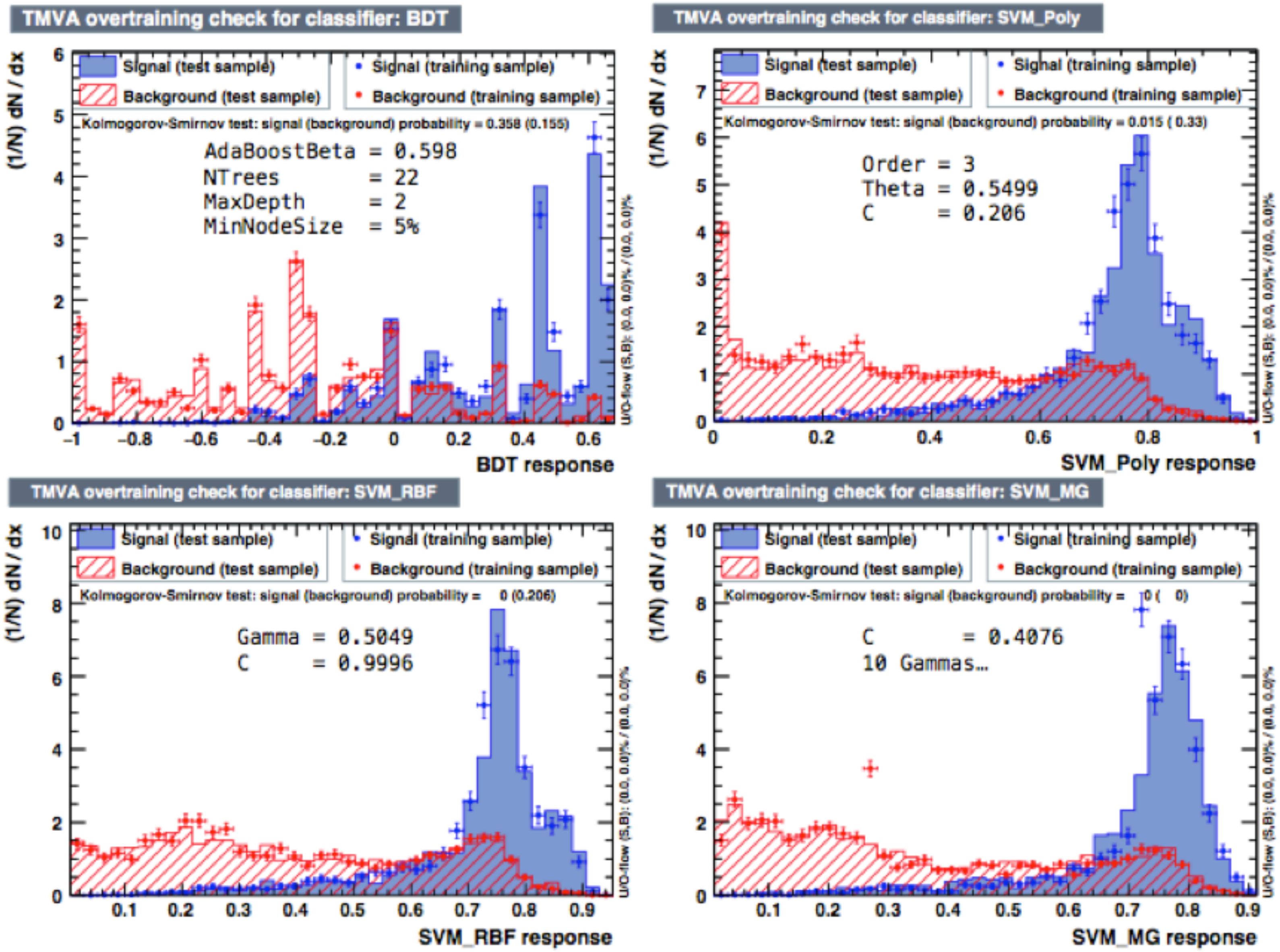
BDT

Spiky as optimisation chooses a low number of trees.

SVM Polynomial (1)

SVM RBF (2)

SVM Multi-Gaussian (3)



```
AdaBoostBeta = 0.598
NTrees       = 22
MaxDepth     = 2
MinNodeSize  = 5%
```

```
Order = 3
Theta = 0.5499
C     = 0.206
```

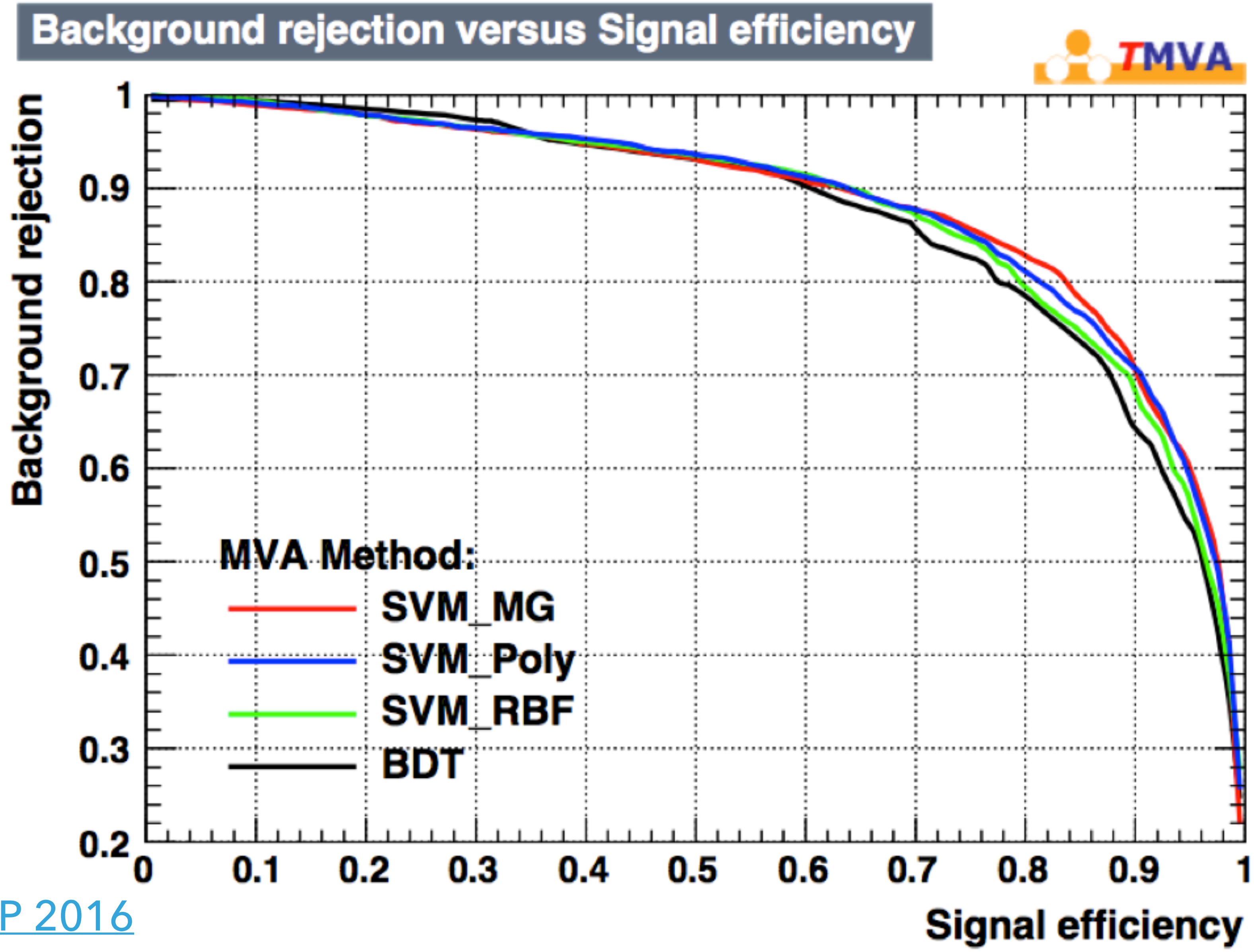
```
Gamma = 0.5049
C     = 0.9996
```

```
C        = 0.4076
10 Gammas…
```

Trained using the hold out method of cross validation (what is normally done in TMVA), with optimised hyper-parameters.

EXAMPLES: H→$\tau^+\tau^-$ (HIGGS KAGGLE DATA CHALLENGE)

- SVM provides comparable performance to BDT (and neural networks)*.

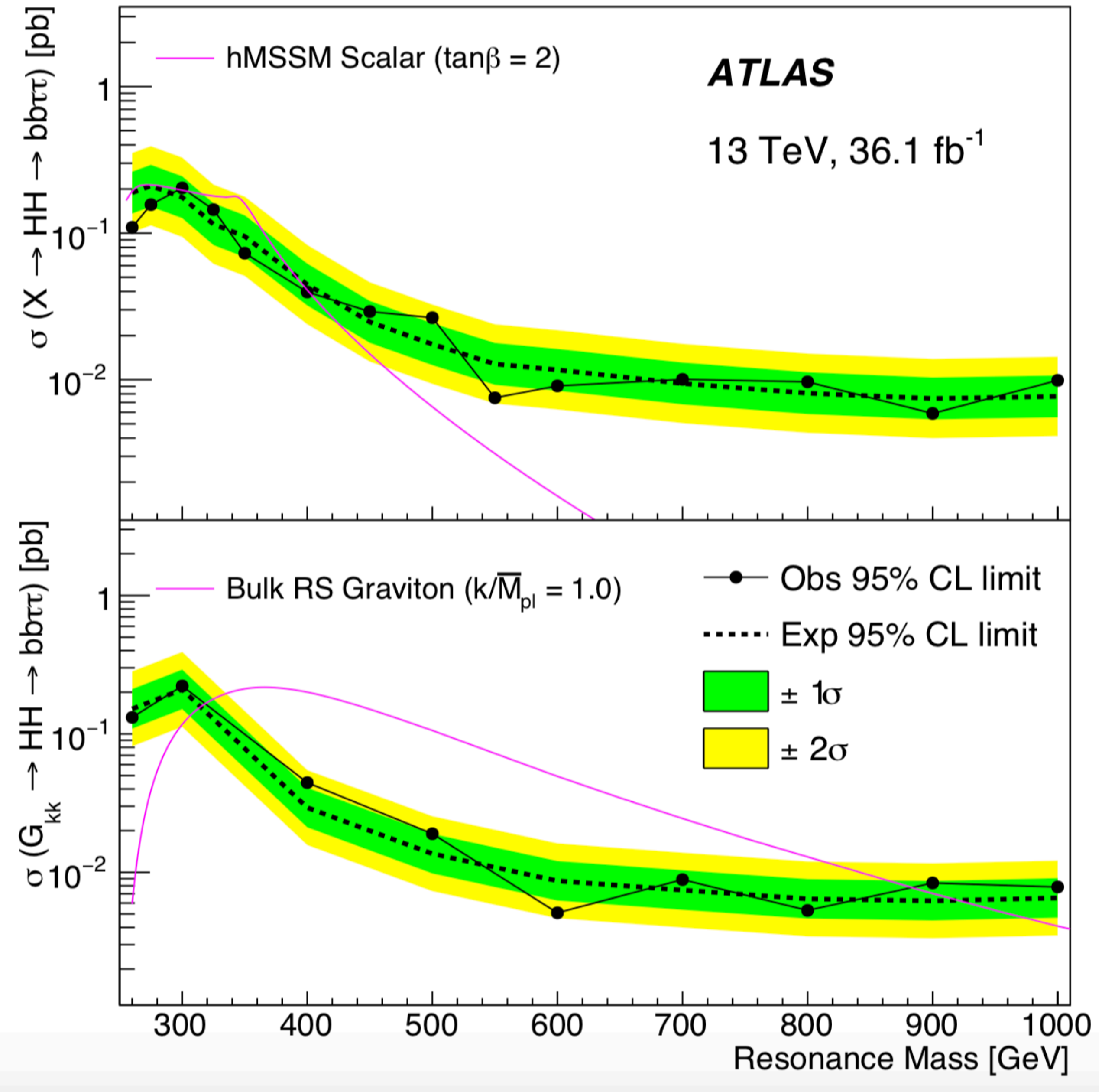


Bevan et al., proc CHEP 2016

*This general conclusion has been reached in one form or another by people studying BDTs vs SVMs and NNs vs SVMs for HEP problems. The take home message is that SVMs require less data to train in order to obtain a generalised result (follows from the fact there are fewer hyper-parameters to determine for SVMs vs other algorithms).

EXAMPLES: HH→BB$\tau^+\tau^-$ (ATLAS – OFFICIAL RESULT)

- ATLAS recently reported limits on resonant and non-resonant production of HH via bb$\tau^+\tau^-$.



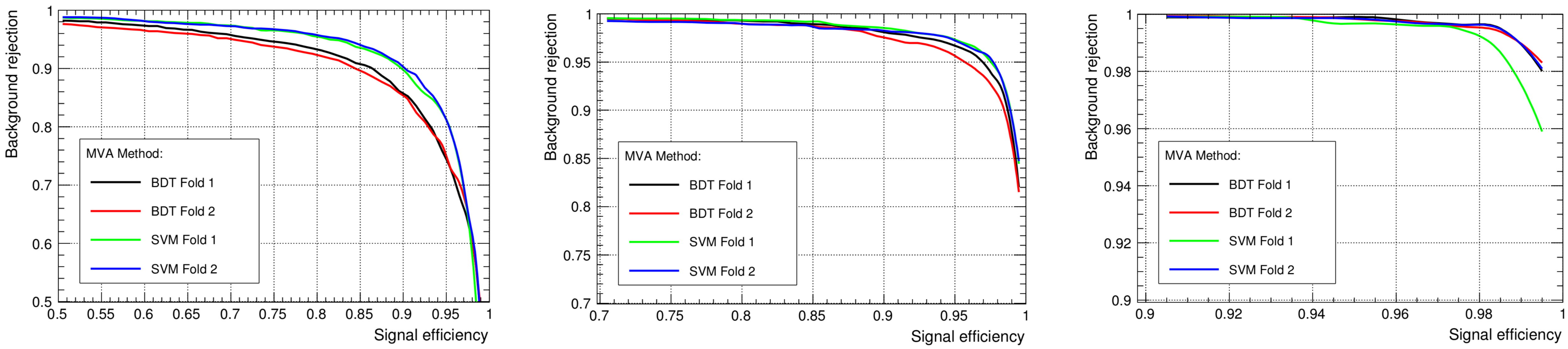
https://arxiv.org/abs/1808.00336

- The standard analysis shown here uses a BDT for both channels that contribute to the final state:
  - Two hadronically decaying τ leptons.
  - One hadronically and one leptonically decaying τ.
- Results for the SM search are 12.7 times the Standard Model expected sensitivity.

EXAMPLES: HH→BB$\tau^+\tau^-$ (ATLAS THESIS)

- A student working on this mode also looked at using SVMs (instead of BDTs) for the analysis.
- Similar performance obtained to the official result when using an SVM for both ROC curves and limit plots.

ROC curves for different mass points in the 2HDM search, using one of the trigger lines for the bb$\tau^+\tau^-$ channel.



(a) 2HDM ($m_H = 300\,\mathrm{GeV}$)

(b) 2HDM ($m_H = 500\,\mathrm{GeV}$)

(c) 2HDM ($m_H = 800\,\mathrm{GeV}$)

- SVMs less susceptible (than BDT) to overtraining for small samples.

T. Stevenson, CERN-THESIS-2018-119

EXAMPLES: HH→BB$\tau^+\tau^-$ (ATLAS THESIS)

- A student working on this mode also looked at using SVMs (instead of BDTs) for the analysis.
- Similar performance obtained to the official result when using an SVM for both ROC curves and limit plots.

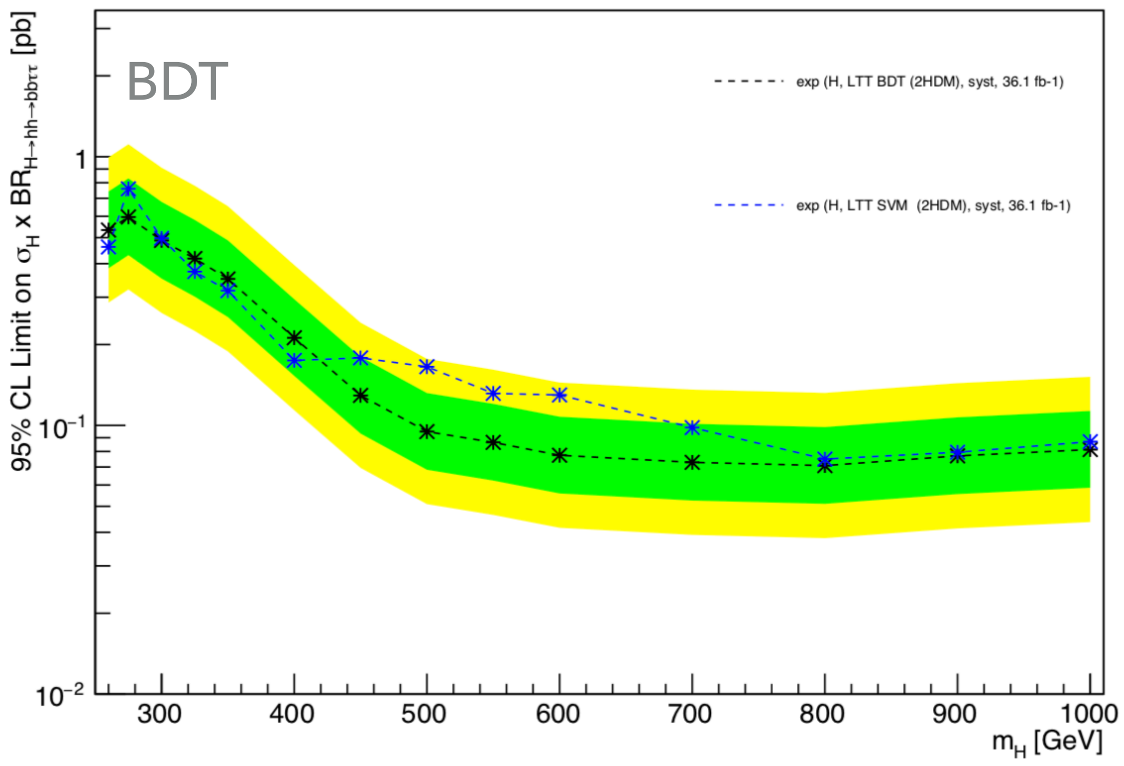


Figure 11.5: Expected limits for the BDT (black) and SVM (blue) at 95% C.L. on the cross-section times branching ratio of the 2HDM heavy scalar Higgs, $H \to hh \to bb\tau\tau$, process in the LTT channel.

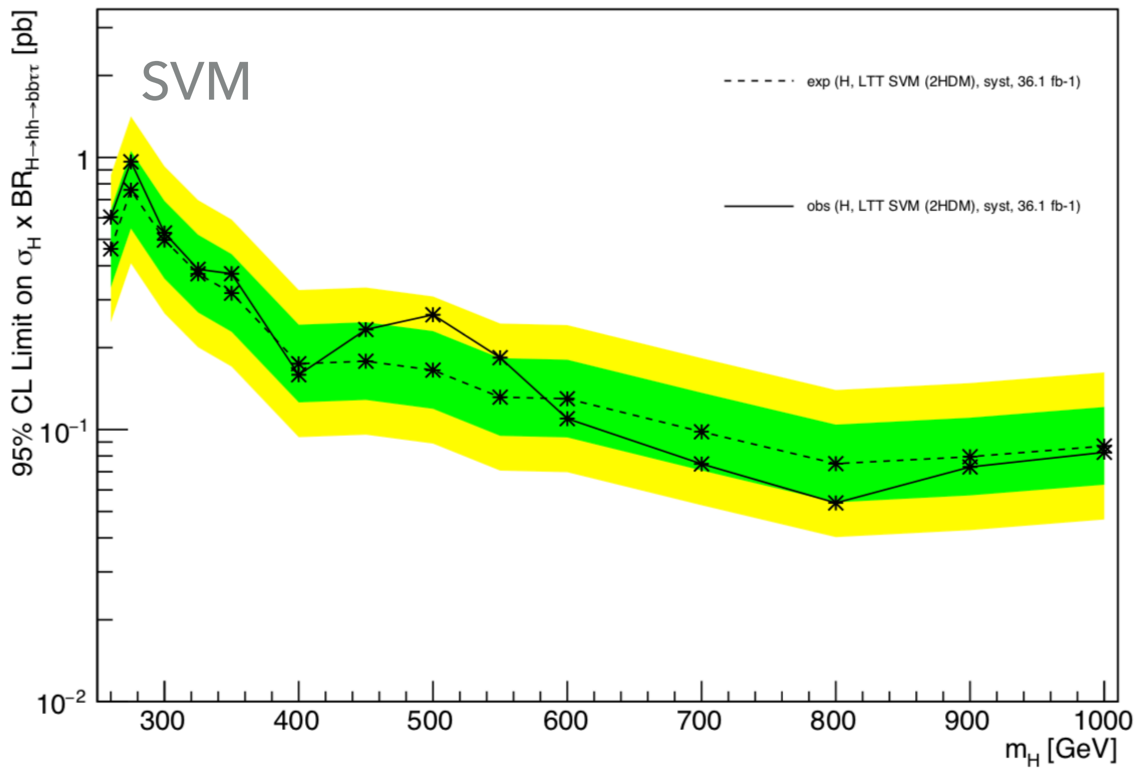


Figure 11.6: Expected (dashed black) and observed (solid black) limits using SVMs at 95% C.L. on the cross-section times branching ratio of the 2HDM heavy scalar Higgs, $H \to hh \to bb\tau\tau$, process in the LTT channel.

T. Stevenson, CERN-THESIS-2018-119

EXAMPLES: SVM HINT APPLIED TO CMS DATA

- Uses libsvm with an RBF kernel function to optimise two parameters: C and Γ.

- Benchmark example of searching for top squark pair production with stops decaying into the lightest supersymmetric particle (LSP) and a top quark.
  - Could use the ROC area under the curve (AOC) to optimise on, but this is not directly related to the result being produced.
  - Instead use the Azimov estimate of the significance of the result as the figure of merit to compare and optimise performance on:

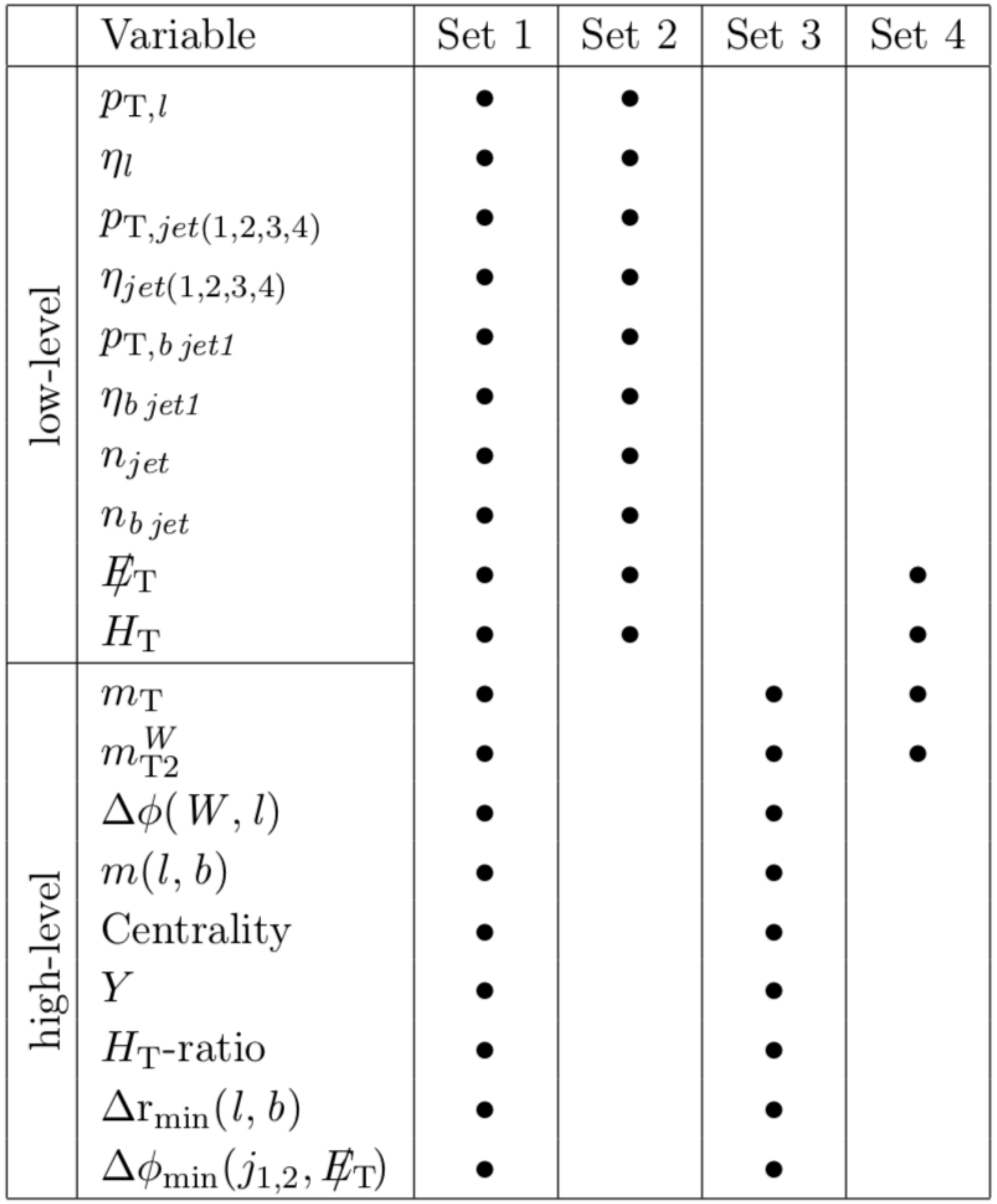
$$Z_A = \left[2\left((s+b)\ln\left[\frac{(s+b)(b+\sigma_b^2)}{b^2+(s+b)\sigma_b^2}\right] - \frac{b^2}{\sigma_b^2}\ln\left[1+\frac{\sigma_b^2 s}{b(b+\sigma_b^2)}\right]\right)\right]^{1/2}$$

This is the median discovery significance from the Poisson form of the signal (s) and background (b), with an uncertainty on the background of σ_b.

M. Sahin et al., Nucl. Instrum. Meth. A838 (2016) 137-146.

EXAMPLES: SVM HINT APPLIED TO CMS DATA

- The variable sets used for the SVM-HINT paper are



	Variable	Set 1	Set 2	Set 3	Set 4
low-level	$p_{\mathrm{T},l}$	●	●		
	η_l	●	●		
	$p_{\mathrm{T},jet(1,2,3,4)}$	●	●		
	$\eta_{jet(1,2,3,4)}$	●	●		
	$p_{\mathrm{T},b\ jet1}$	●	●		
	$\eta_{b\ jet1}$	●	●		
	n_{jet}	●	●		
	$n_{b\ jet}$	●	●		
	$\not\!\!E_{\mathrm{T}}$	●	●		●
	H_{T}	●	●		●
high-level	m_{T}	●		●	●
	m_{T2}^{W}	●		●	●
	$\Delta\phi(W,l)$	●		●	
	$m(l,b)$	●		●	
	Centrality	●		●	
	Y	●		●	
	H_{T}-ratio	●		●	
	$\Delta\mathrm{r}_{\mathrm{min}}(l,b)$	●		●	
	$\Delta\phi_{\mathrm{min}}(j_{1,2},\not\!\!E_{\mathrm{T}})$	●		●	

As with other work on using ML methods the expected result that the combination of high level and low level (derived and primitive) features provides better performance than using just one of those sets.

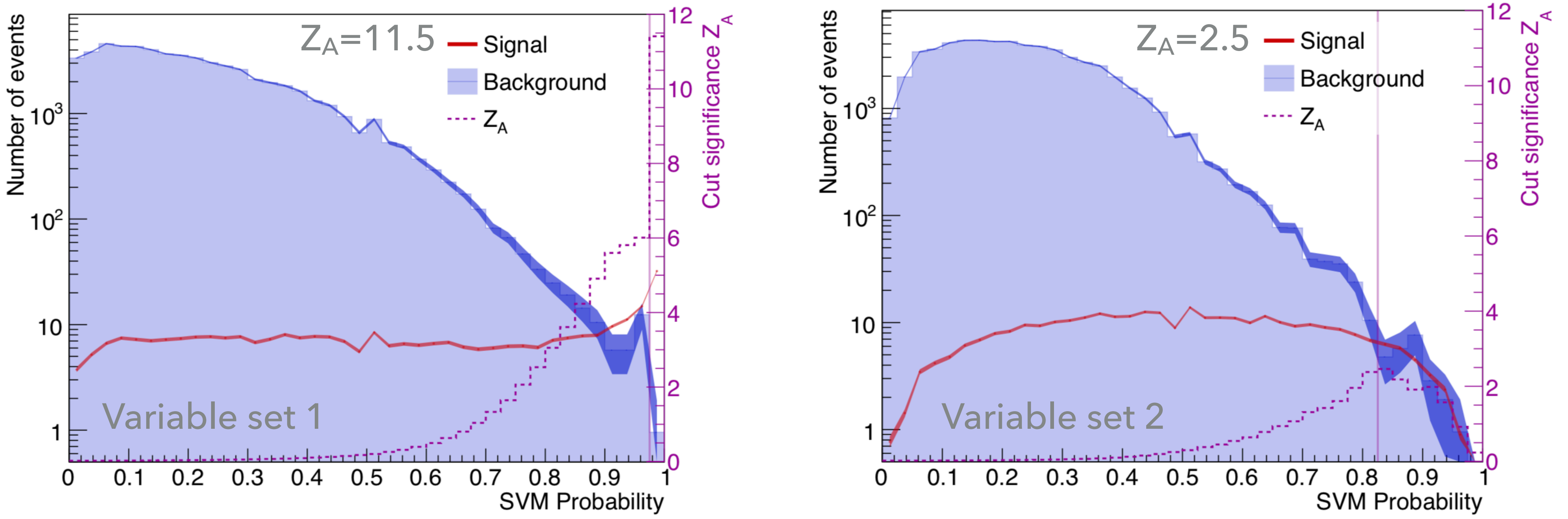
Results on the next two pages illustrate this.

M. Sahin et al., Nucl. Instrum. Meth. A838 (2016) 137-146.

EXAMPLES: SVM HINT APPLIED TO CMS DATA

- Results are turned into a probabilistic score using a sigmoid function:

$$P(y=1|\hat{f}) = \begin{cases} \frac{\exp(-t)}{1+\exp(-t)} & : t \equiv A + B\hat{f} \geqslant 0 \\ \frac{1}{1+\exp(t)} & : t < 0 \end{cases}$$

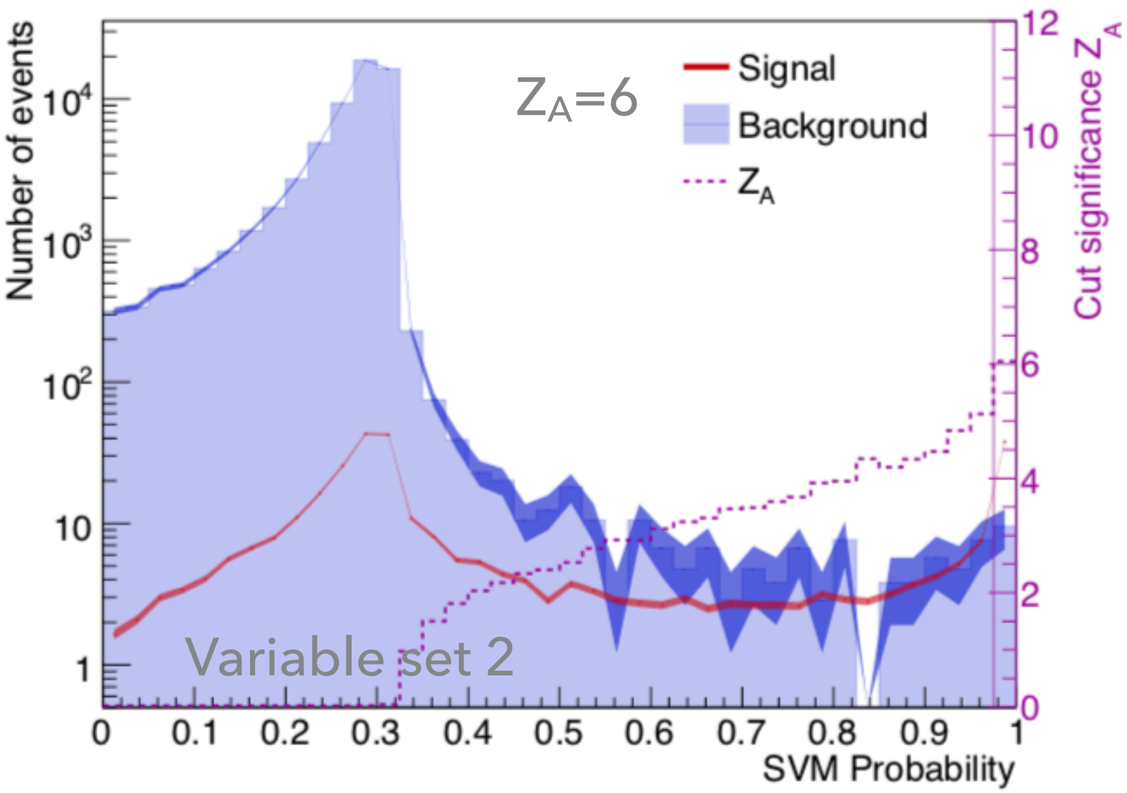


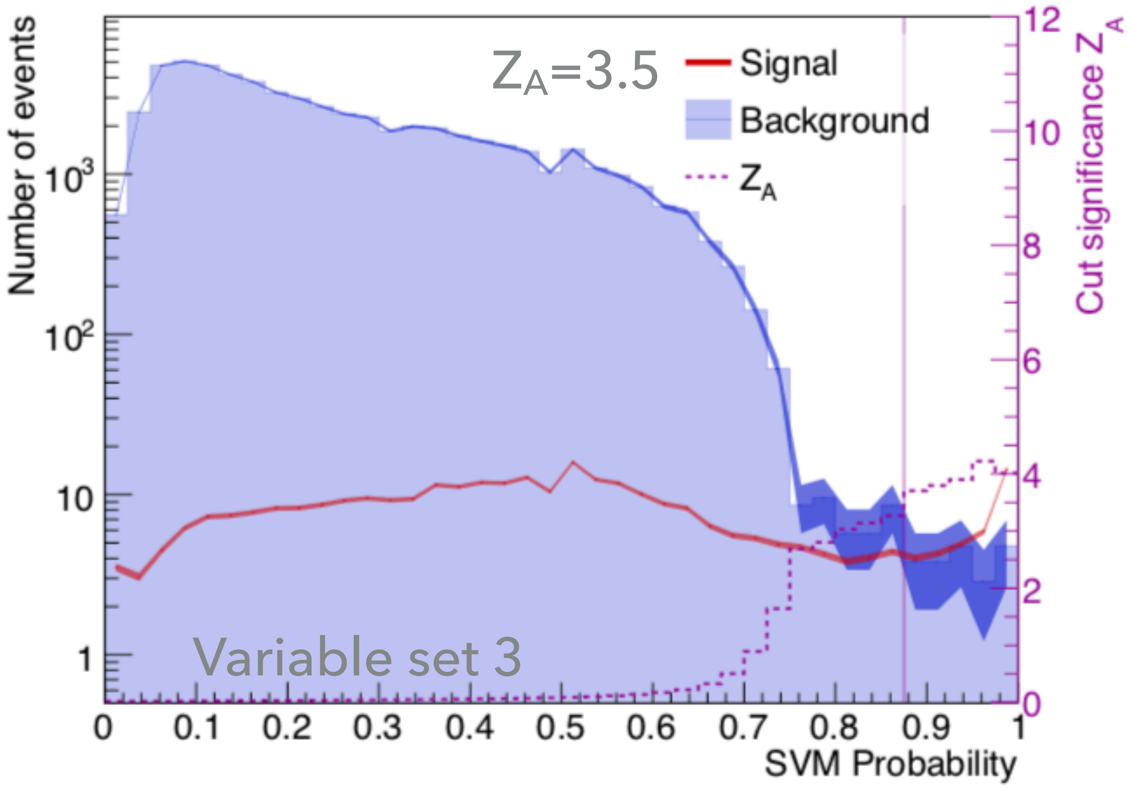
M. Sahin et al., Nucl. Instrum. Meth. A838 (2016) 137-146.

EXAMPLES: SVM HINT APPLIED TO CMS DATA

- Results are turned into a probabilistic score using a sigmoid function:

$$P(y=1|\hat{f}) = \begin{cases} \frac{\exp(-t)}{1+\exp(-t)} & : t \equiv A + B\hat{f} \geqslant 0 \\ \frac{1}{1+\exp(t)} & : t < 0 \end{cases}$$





M. Sahin et al., Nucl. Instrum. Meth. A838 (2016) 137-146.

SUMMARY AND MISCELLANEOUS NOTES

- Use SVMs when:
  - You have small or very small training examples.
  - and you care about obtaining a generalised result (reproducibility of the output matters even if the data fed to the algorithm changes).
  - Computing time/resource (incl. memory) is not a problem.

- Do not use an SVM when:
  - You have a lot of training examples and/or very little computing resource.

SUMMARY AND MISCELLANEOUS NOTES

- We've looked at the hard and soft margin SVMs.
    - The algorithm stems from the same linear separation problem that is addressed by Rosenblatt's perceptron paper.
    - However this focusses on how far an example is from the margin defining the separating hyperplane.
    - Can't understand the mapping from the input feature space to the dual space (but we don't have to).
- SVMs are widely used outside of HEP.
- They have been used for a broad range of physics studies in HEP, but the algorithm has not been widely adopted.
- There are specific reasons why you would or would not want to use the algorithm.
- Searches where you have limited training examples available (e.g. SUSY or Higgs BSM) are cases where you might want to look at the algorithm.

SUGGESTED TOOLS

- The SVM algorithm is implemented in a few different code bases.
- scikit learn/R/Matlab/SVM-HINT*/ROOT:
  - libsvm from: https://www.csie.ntu.edu.tw/~cjlin/libsvm/
- TMVA in ROOT:
  - HEP community developed SVM.

* see Sahin et al in the references.

SUGGESTED READING (HEP RELATED)

- Background suppression (jets):
    - F. Sforza, V. Lippi, Nucl. Inst. Meth. A722, (2013), p11-19 (arXiv:1407.0317).
- Flavour Tagging:
    - P. Vannerum et al., Freiburg EHEP-99-01 (hep-ex/9905027).
- Machine Physics:
    - Bijan Sayyar-Rodsari, C. Schweiger, SLAC-R-948.
- Review:
    - A. Vossen, Part of the proceedings of the Track 'Computational Intelligence for HEP Data Analysis' at iCSC 2006 arXiv:0803.2345.
- Top:
    - A. Vaiciulis, Nucl. Instrum. Meth. A502 (2003) 492-494 (hep-ex/0205069).
    - S. Ridella et al., IEEE Conf.Proc. (2004) no.3, 2059-2064.
    - B. Whitehouse, FERMILAB-THESIS-2010-61.
- SUSY:
    - M. Sahin et al., Nucl. Instrum. Meth. A838 (2016) 137-146.
- Higgs:
    - Tom Stevenson, Thesis (QMUL 2018), CERN-THESIS-2018-119.

SUGGESTED READING (NON-HEP)

- Nello Cristianini and John Shawe-Taylor, "Support Vector Machines and other kernel-based learning methods", Cambridge University Press, 2000. [and refs. therein]
- B. Scholkopf and A. Smola, "Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond", MIT Press, 2002.
- J. Mercer. Phil. Trans. Roy. Soc. Lond., A209:415, 1909.
- C.C.Chang,C.J.Lin,ACMTransactionsonIntelligentSystemsandTechnology2,27:1 (2011). Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm