Machine Learning Techniques at the LHC Experiments

Özgür Şahin

16.01.2018

DE LA RECHERCHE À L'INDUSTRIE

What to expect?

- **Introduction to Machine learning**
- **Clustering with DBSCAN**
    - Performance under pileup
- **Support Vector Machines (SVM)**
    - Hyperparameter Optimization
    - Case Studies
    - Implementation in the 8 TeV top squark search
- **Conclusion**

Machine learning

- Searching for new physics in HEP
  - Classification of Signal (New Physics) to the background (Standard Model).
  - For most of the new physics models idiom *looking for a needle in a haystack* does not actually suffice…

Machine learning

- A better version would be looking for a slightly different needle within a stack of needles.
- We need to analyze all the inputs in a very affective way.
  - Machine learning!
- Within HEP, the benefits are quite visible, the interest is high, but the support is limited.

Machine learning

- Machine learning: an autonomous process where the **P**erformance for a given **T**ask is improved by increasing **E**xperience. (T. M. Mitchell)
    - **T** — Classification of new physics events among standard model events, identification of particle of interest among vast noise…
    - **E** — Number of events: asymmetrical samples (ie most of the time input includes only 0.01% signal.)
    - **P** — "Discovery!" (or to be less dramatic sensitivity), correct identification, being resilient against pile-up…
- Rosetta for HEP - ML - Statistics (not true for all but safe to generalize)
    - *Label, Class*: *Signal* & *Background* (we would like to be more explicit… BTW *background* is *noise* for what it is worth and we use *noise* mainly for instrumental fluctuations)
    - *Performance: Sensitivity, significance*
    - *Performance criteria: Figure of Merit*
    - *Feature: Variable* (I know… It is harder to pronounce for non-native that is why it is preferred).

Machine learning

Data rate too high
@ 40 MHz

Low-level Trigger

Filtered data

Readout

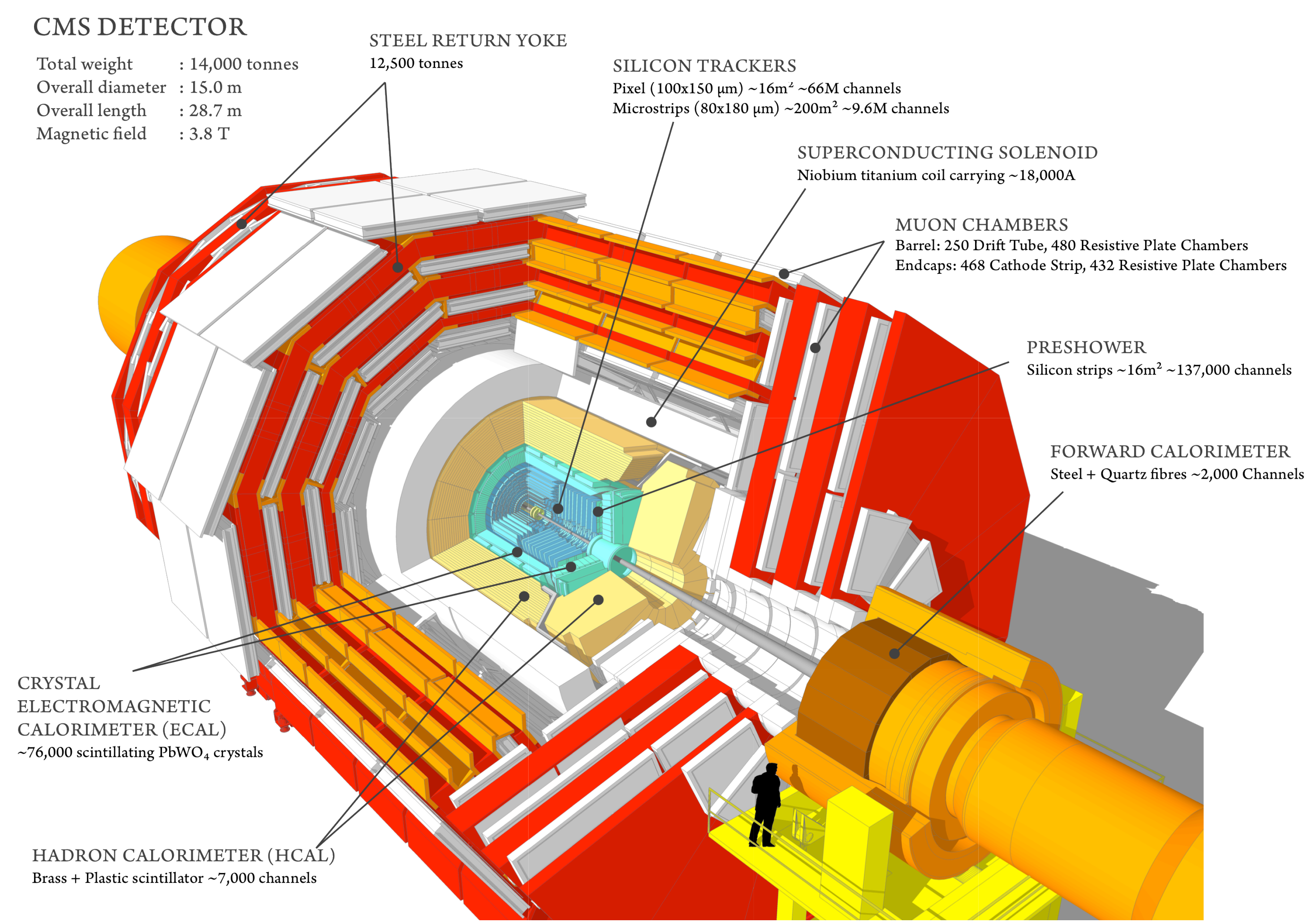
Data rate high
@ 100 KHz

High-level Trigger

Readout

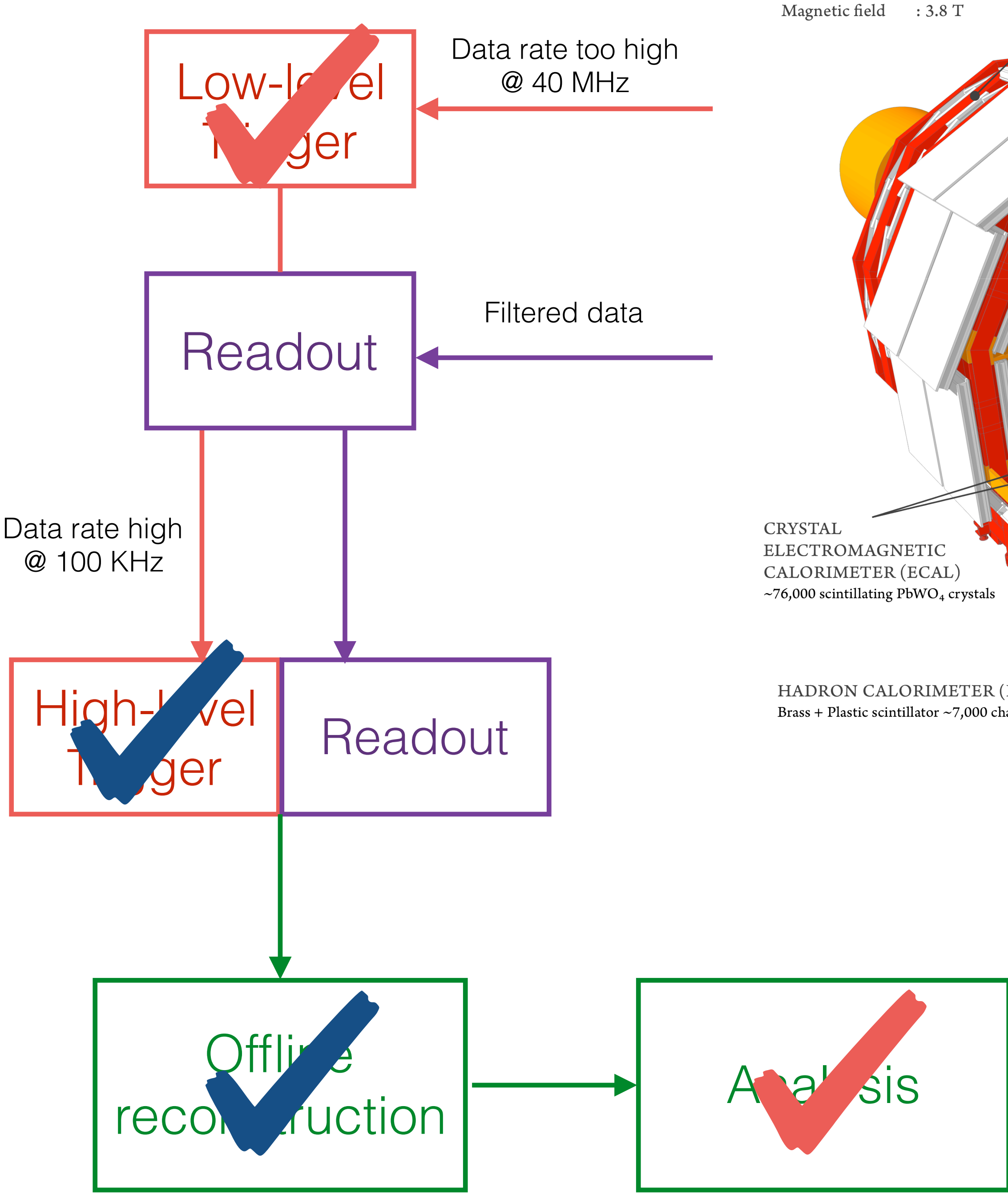
Offline reconstruction

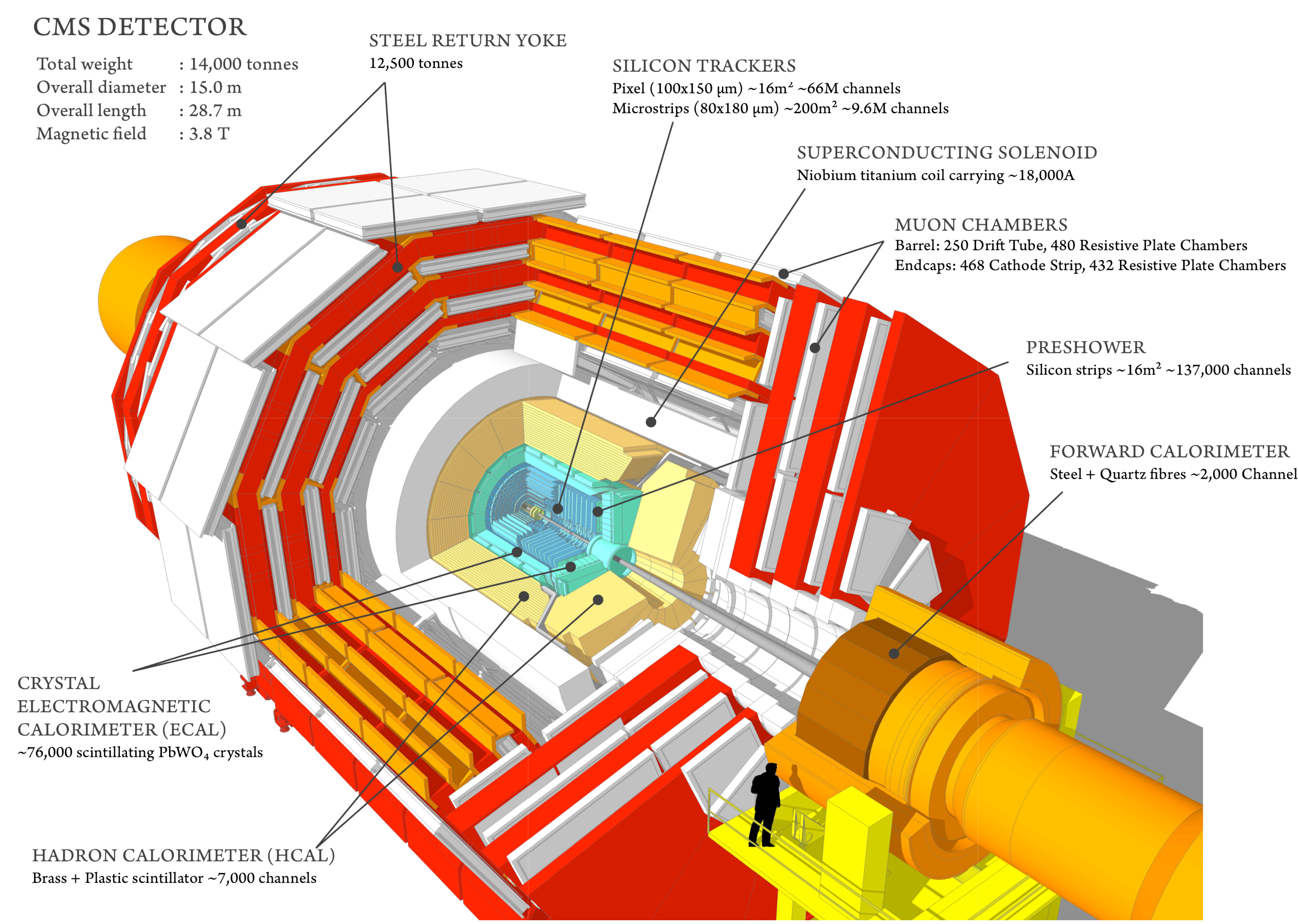
Analysis



An extremely simplified view of data path in CMS

Machine learning





An extremely simplified view of data path in CMS

Özgür Sahin

Machine learning

Supervised

- Decision trees
- Support Vector Machines
- Neural Nets
- Fischer discriminant
- ⋮

Unsupervised

- Hebbian learning
- Principal Component Analysis
- k-means
- DBSCAN
- ⋮

Machine learning

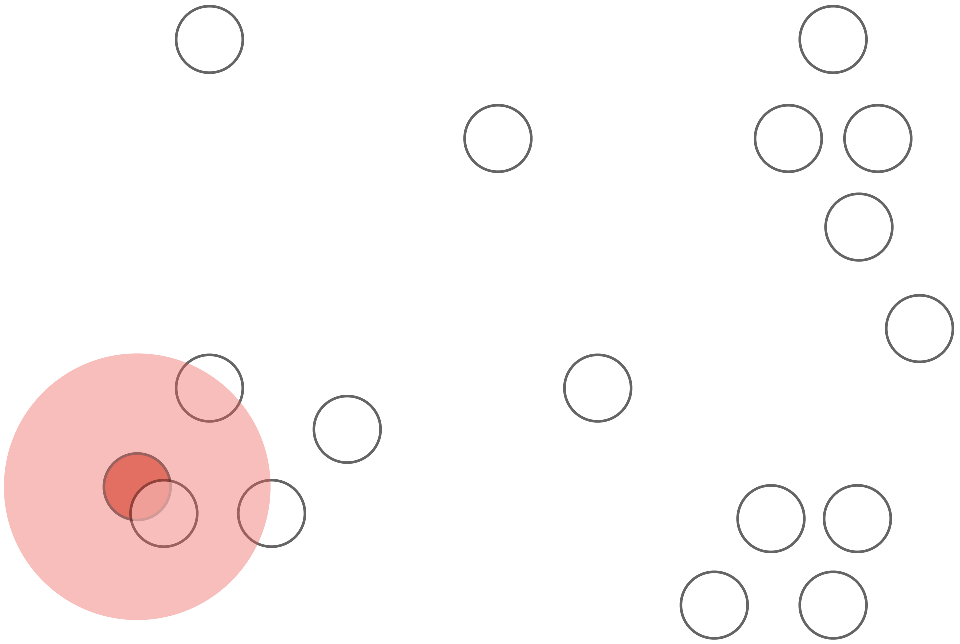
Supervised

- Decision trees
- Support Vector Machines
- Neural Nets (since 2013)
- Fischer discriminant
- ⋮

Unsupervised

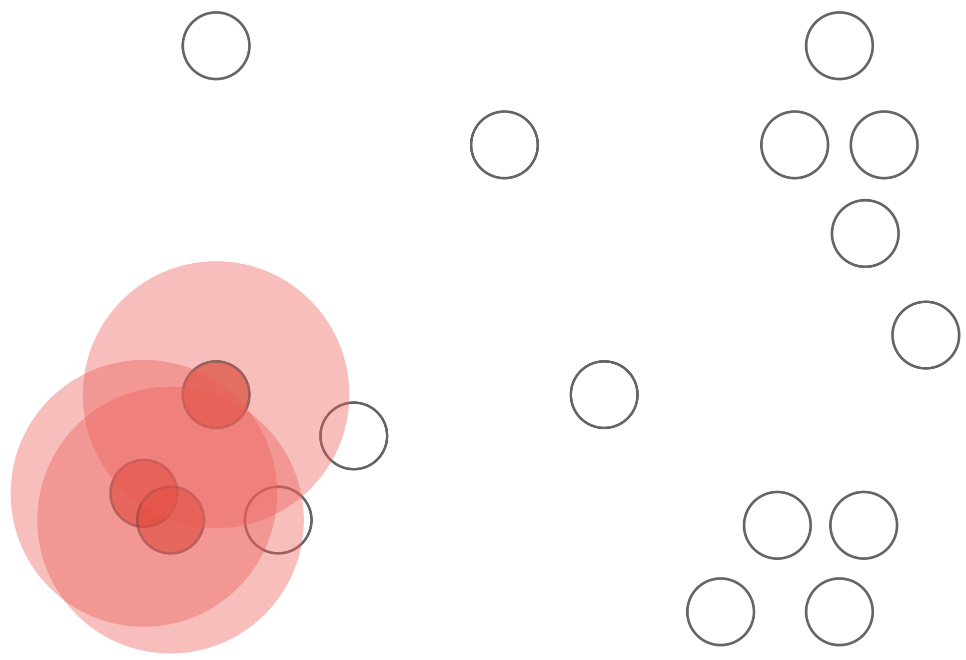
- Hebbian learning
- Principal Component Analysis
- k-means
- DBSCAN
- ⋮

DBSCAN



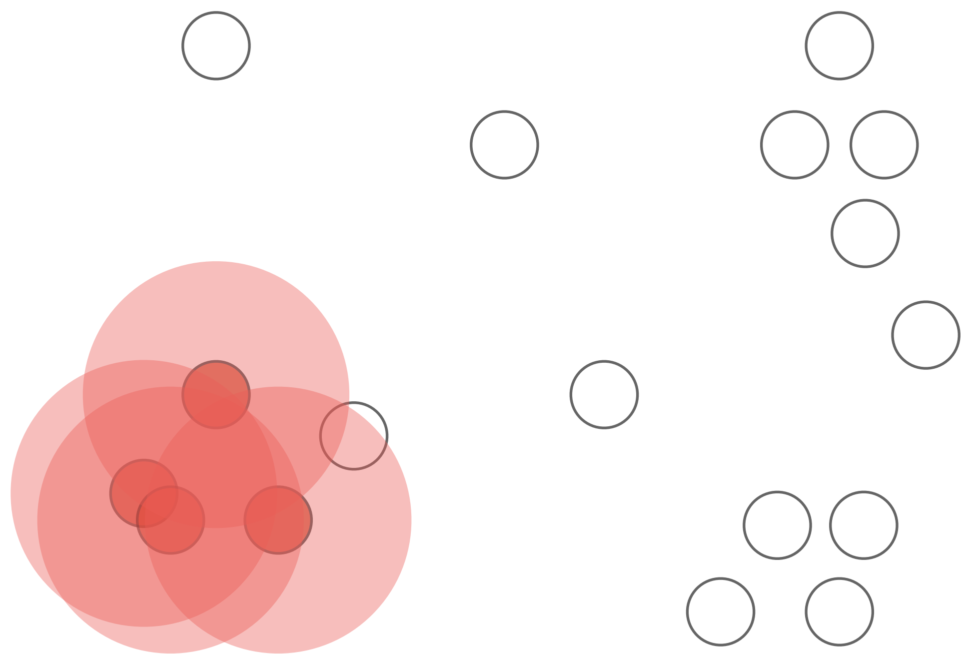
- DBSCAN is a an award winning (timing) density-based clustering algorithm. In this talk, this algorithm is compared to a simple cone algorithm.
- It has two hyperparameters: Minimum distance (radius) and minimum number of points.

DBSCAN



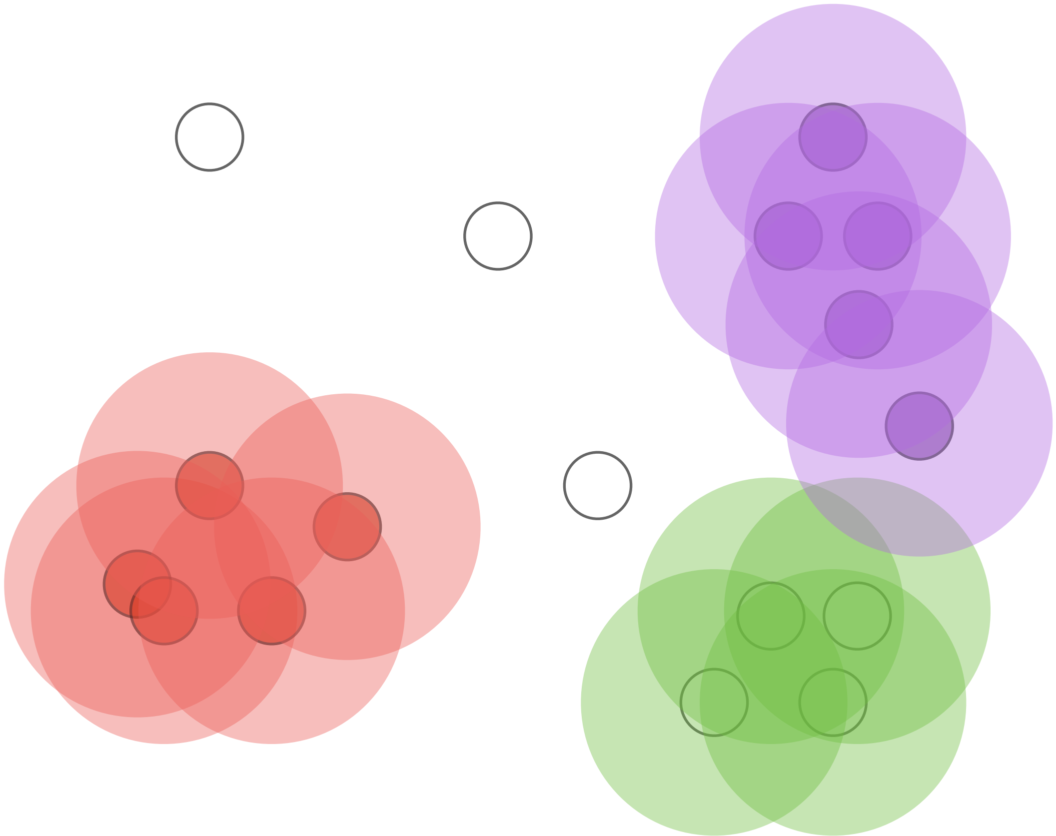
- DBSCAN is a an award winning (timing) density-based clustering algorithm.
- It has two hyperparameters: Minimum distance (radius) and minimum number of points.

DBSCAN



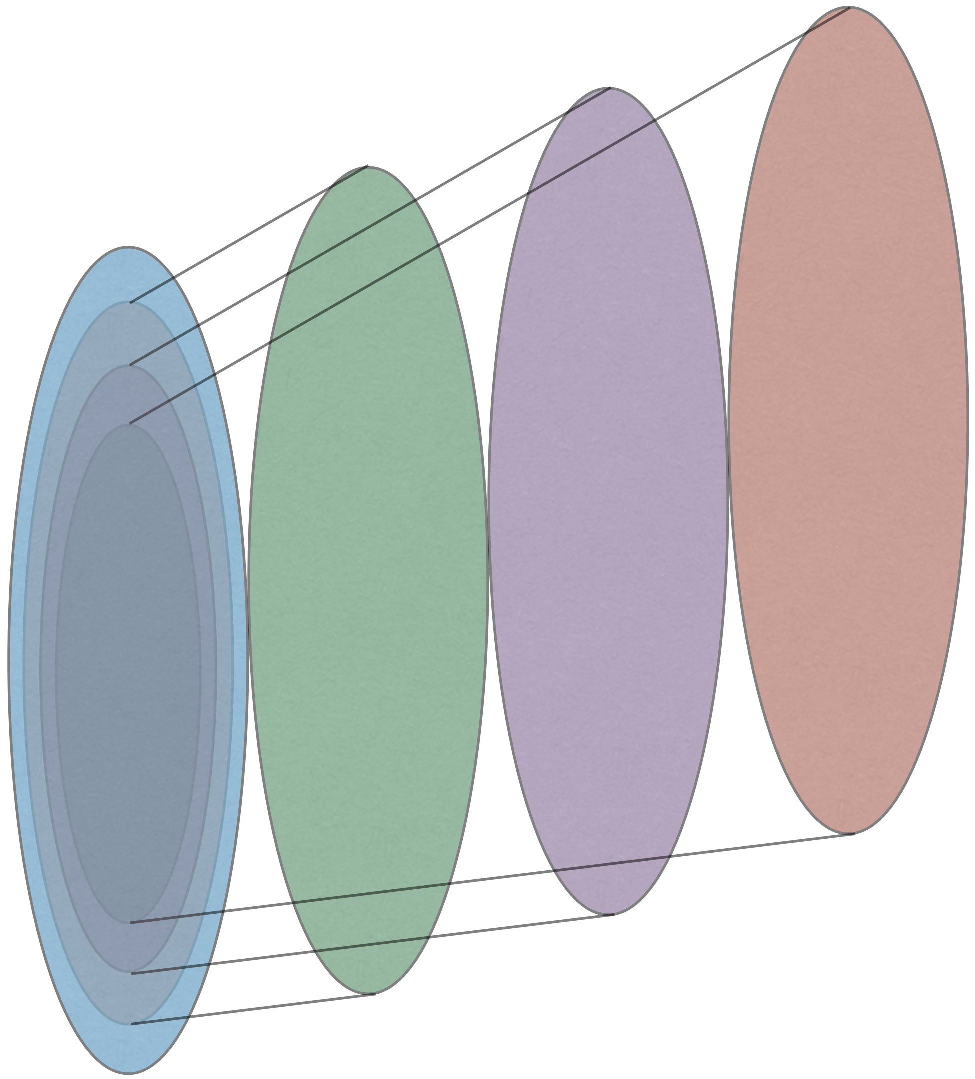
- DBSCAN is a an award winning (timing) density-based clustering algorithm.
- It has two hyperparameters: Minimum distance (radius) and minimum number of points.

DBSCAN



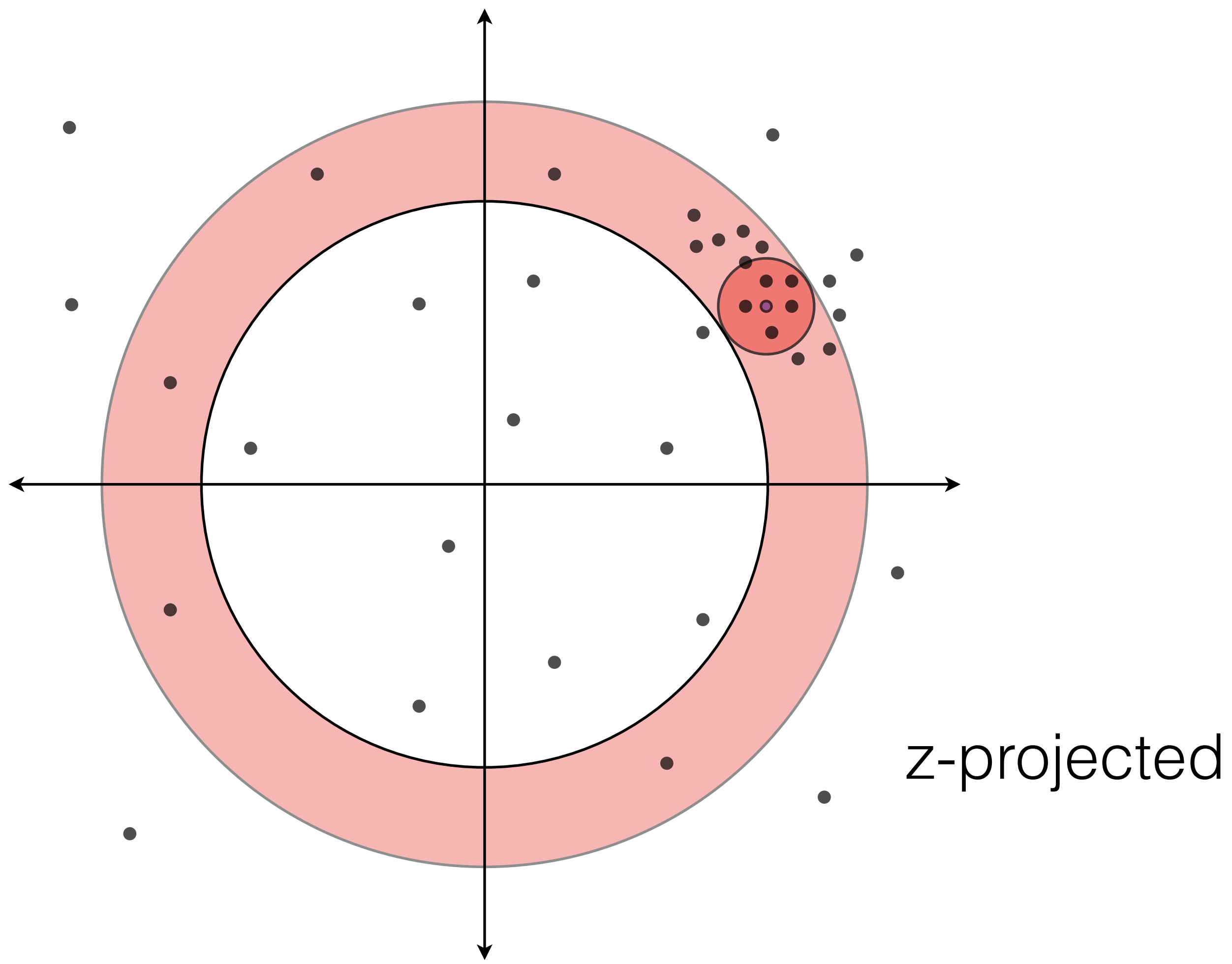
- DBSCAN is extremely efficient for
    - eliminating noise,
    - detecting non-uniform clusters.
- https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/

Clustering algorithms



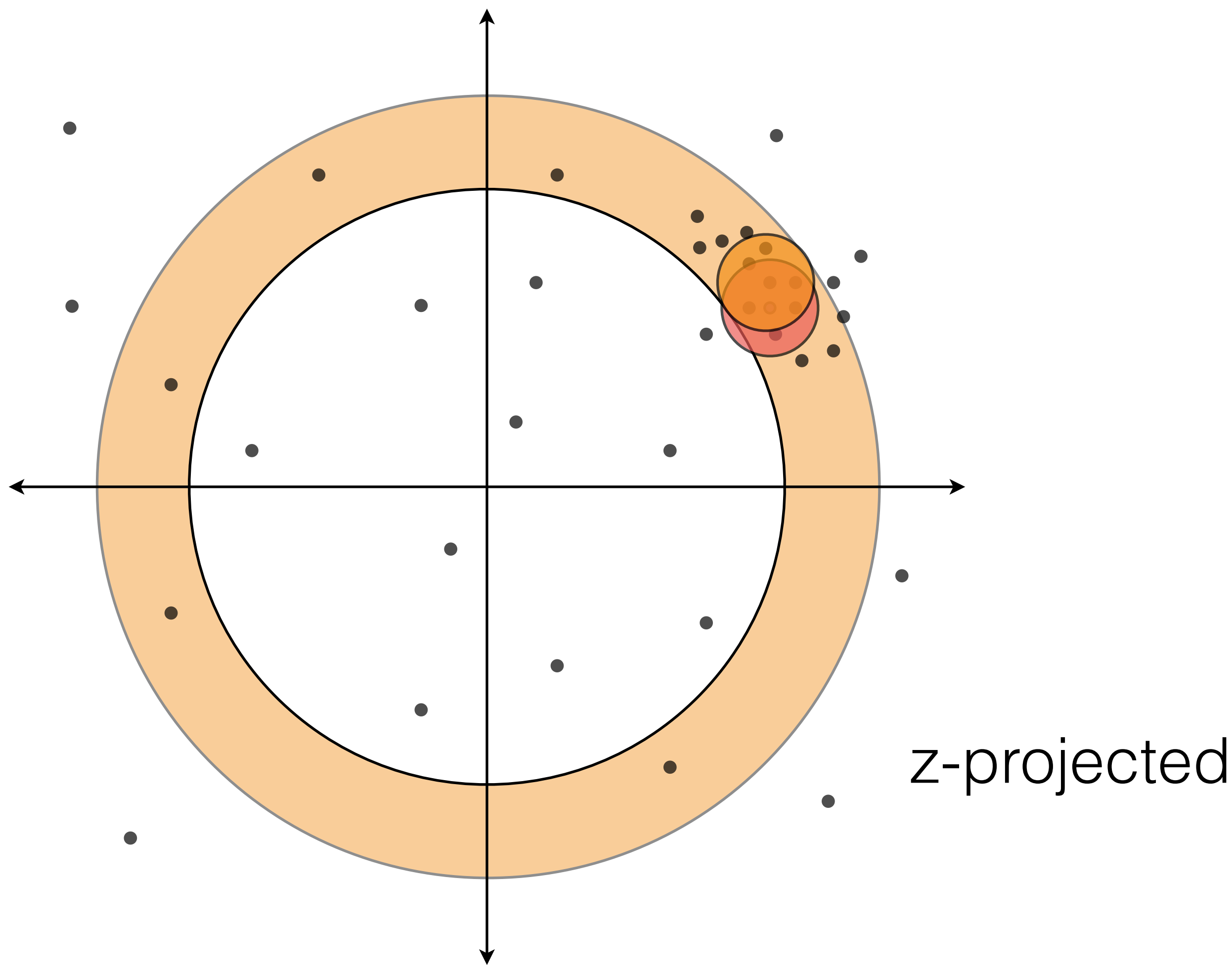
- To reconstruct the objects, we cluster the hits in our detectors using various algorithms. The readout is synchronous with LHC (ie ~40 MHz).
- These algorithms are meant to be implemented in low level electronics.
- In order to simplify the problem, a dimension reduction is performed by projecting the "layers" into a single plane in Z.

Software implementation



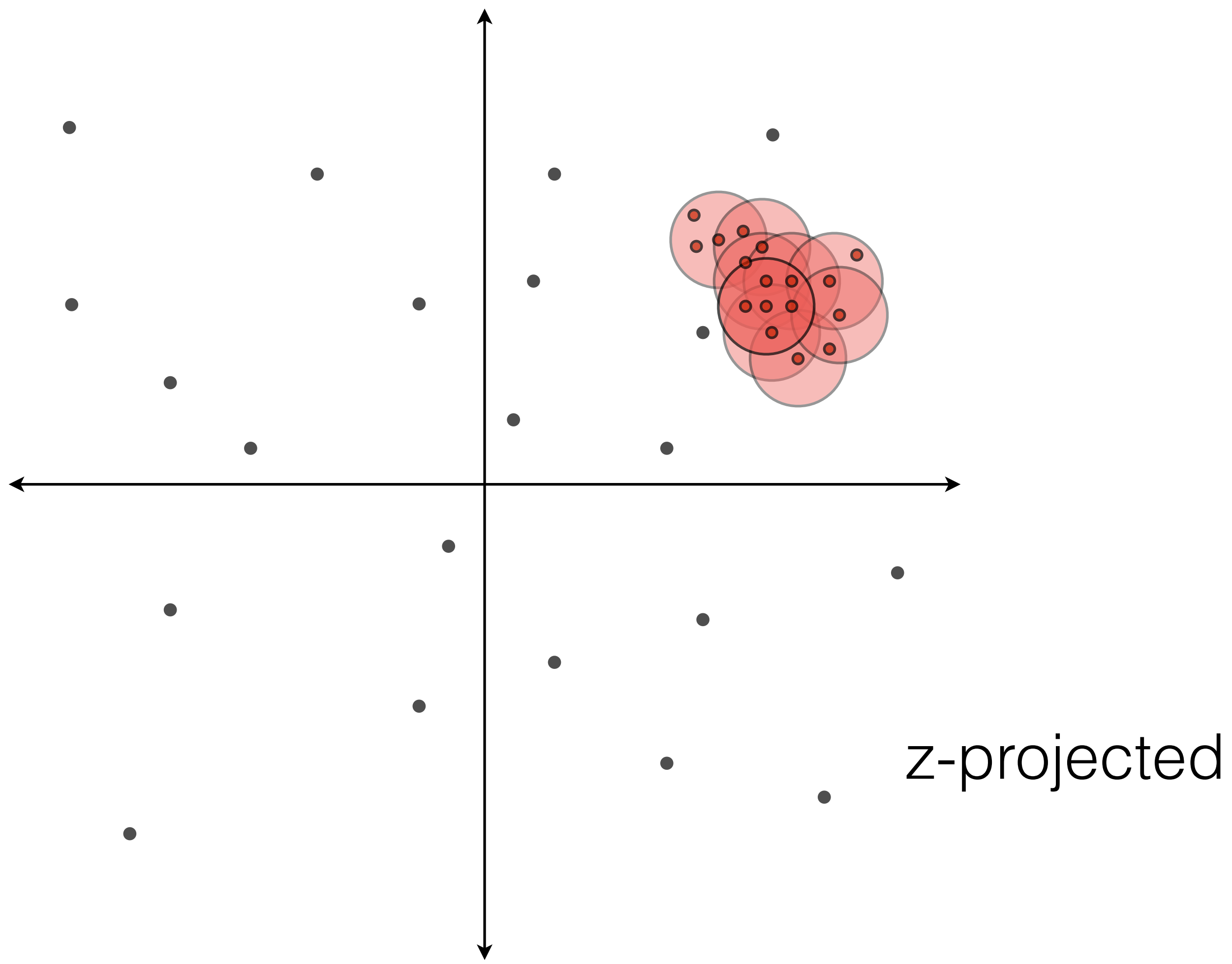
- The clusters are ranked wrt to distance to center.
- A secondary search is performed to find the neighbors.

Software implementation



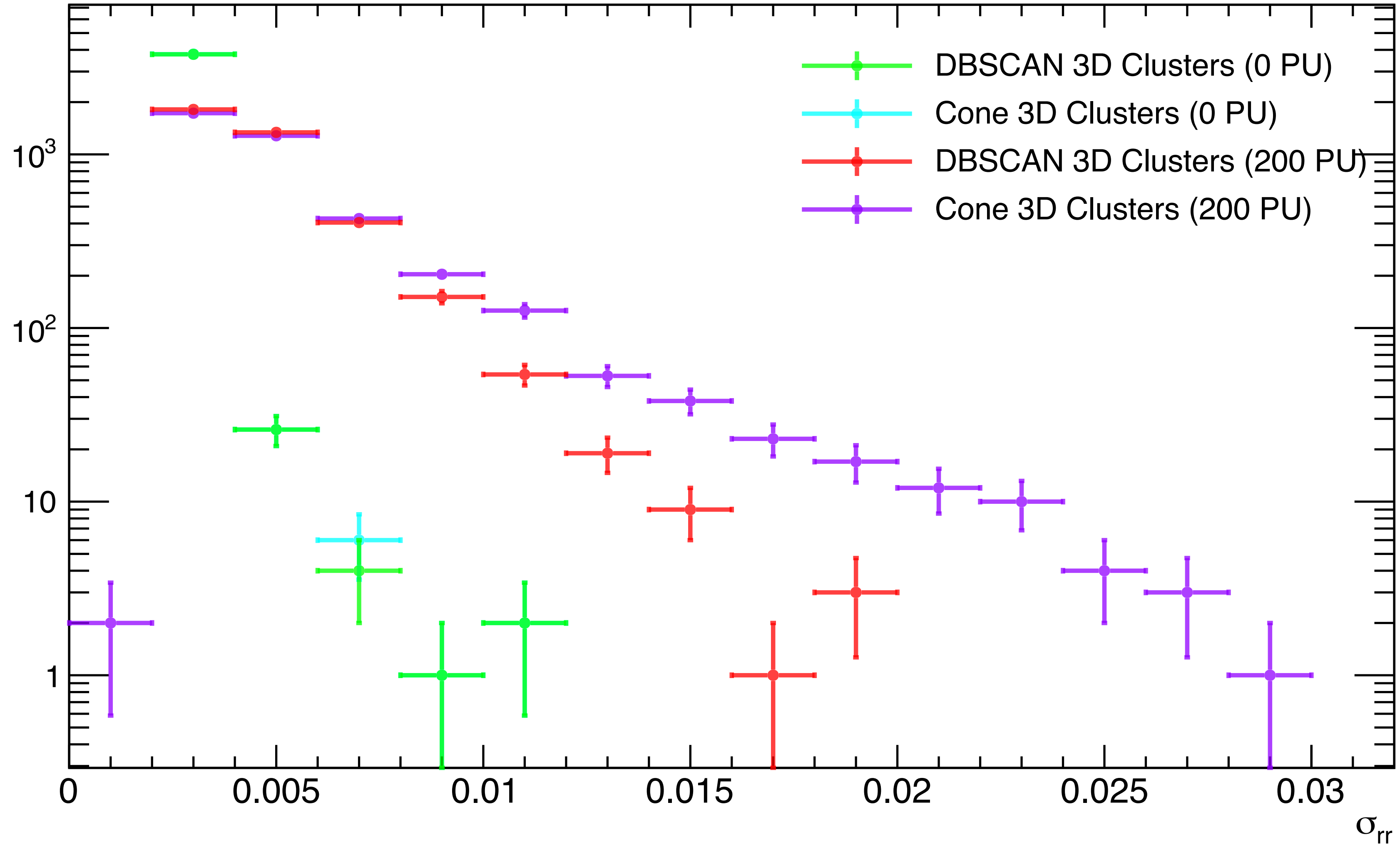
- The clusters are ranked wrt to distance to center.
- A secondary search is performed to find the neighbors.

Software implementation



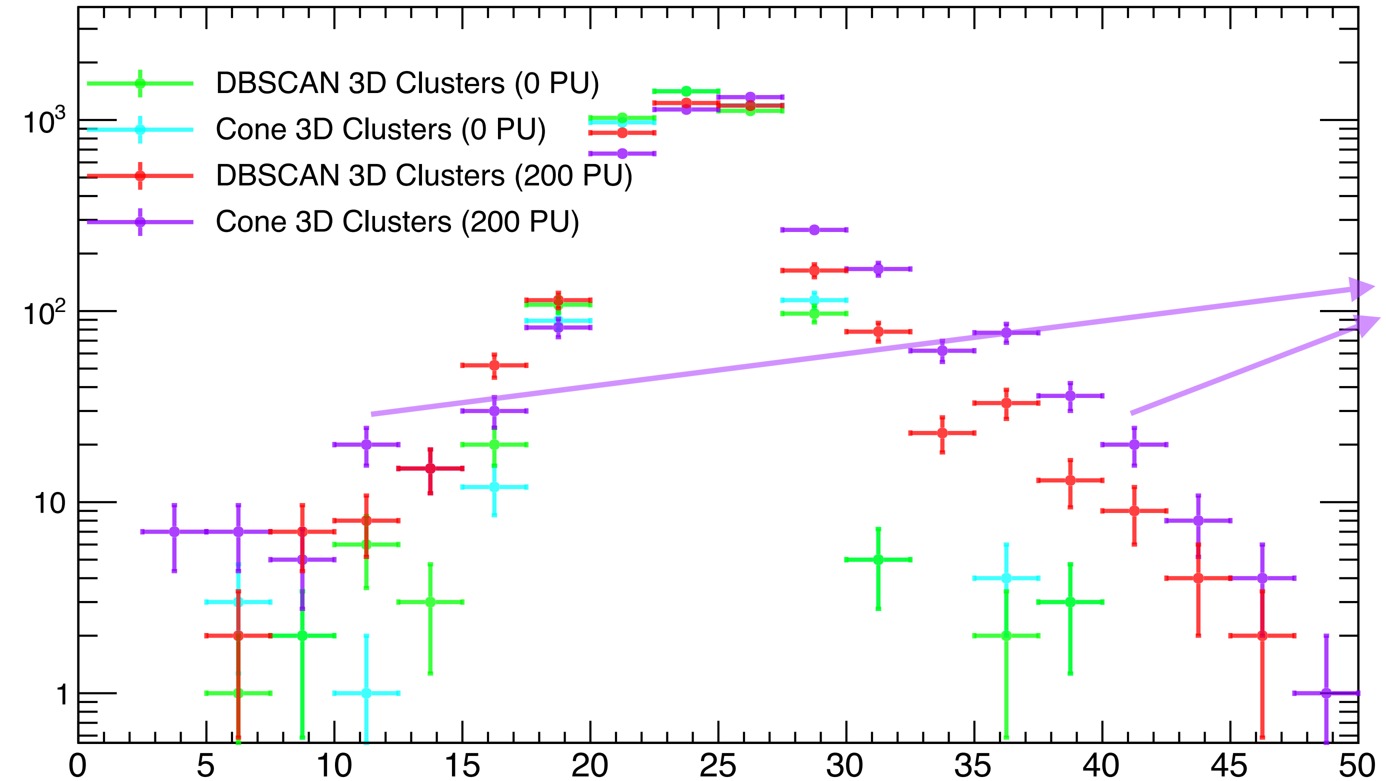
- The sorted DBSCAN algorithm is approximately 20 times faster than the unsorted search.
- The sorted DBSCAN is as fast as the Cone clustering algorithm.

Cluster shape comparison

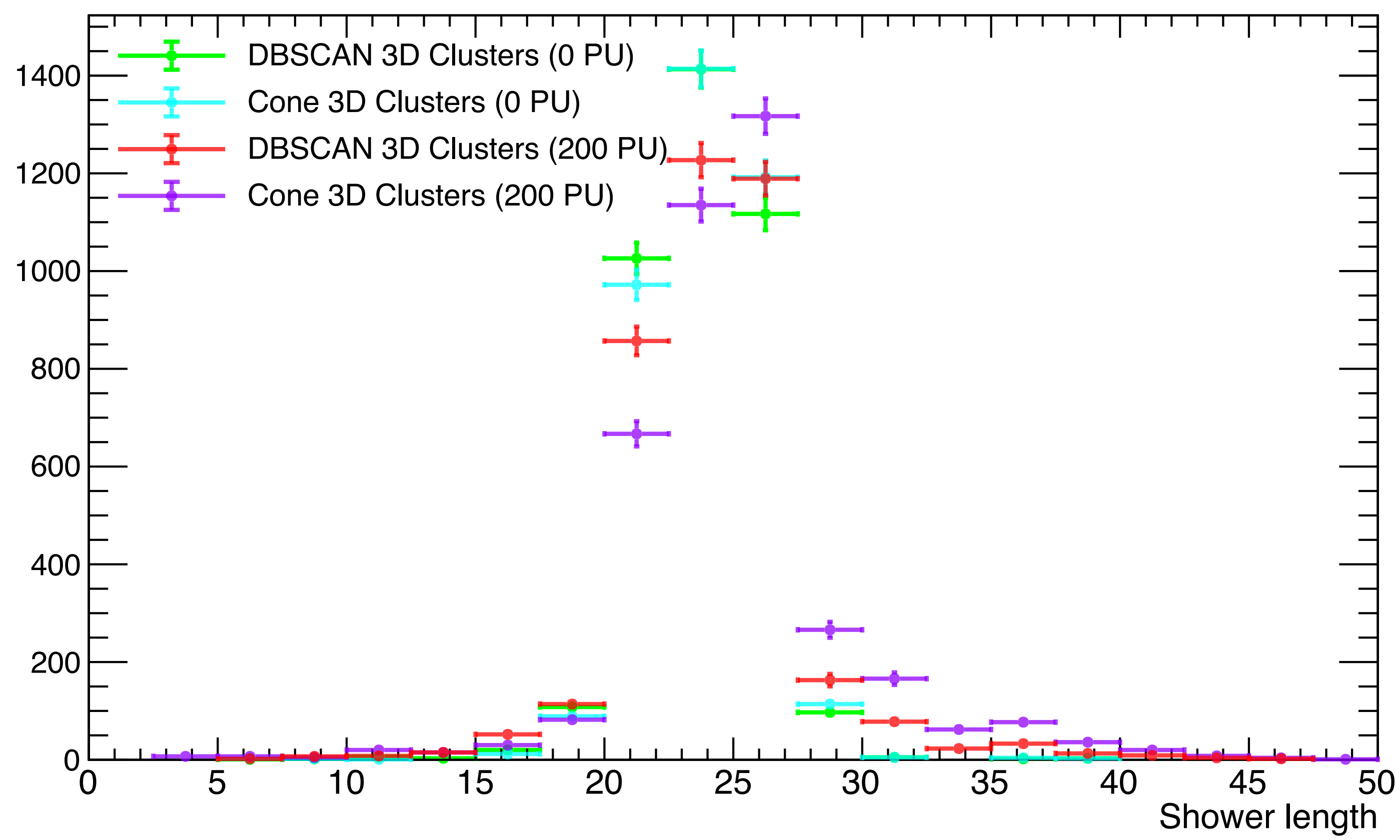


- With no PU, both algorithms produce clusters with similar widths (overlapping distributions).
- Even though it is still significantly affected by the high PU, the DBSCAN algorithm produces narrower clusters.

Cluster shape comparison



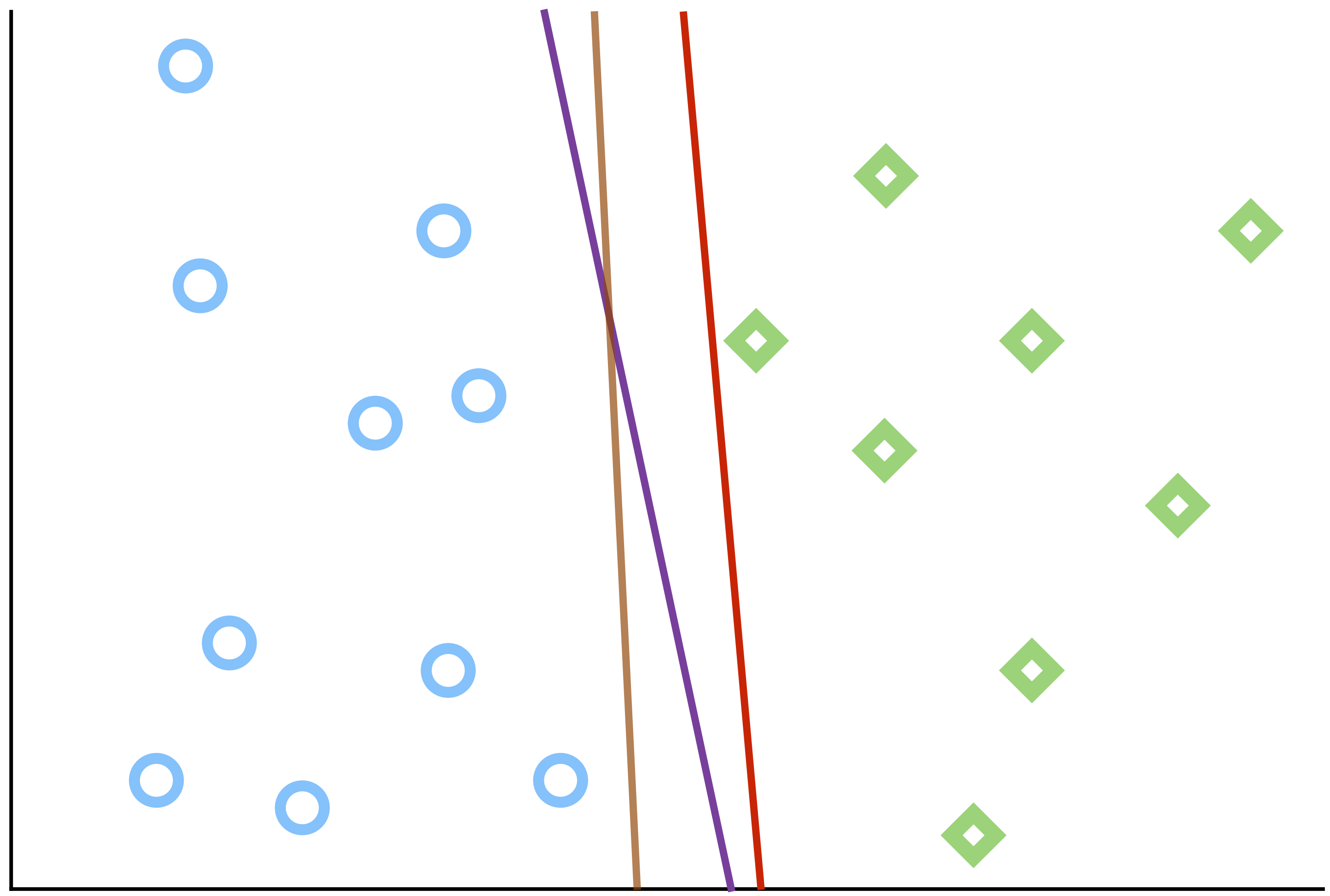
Cone algorithm yields to a wider shower length distribution in high PU



SVM

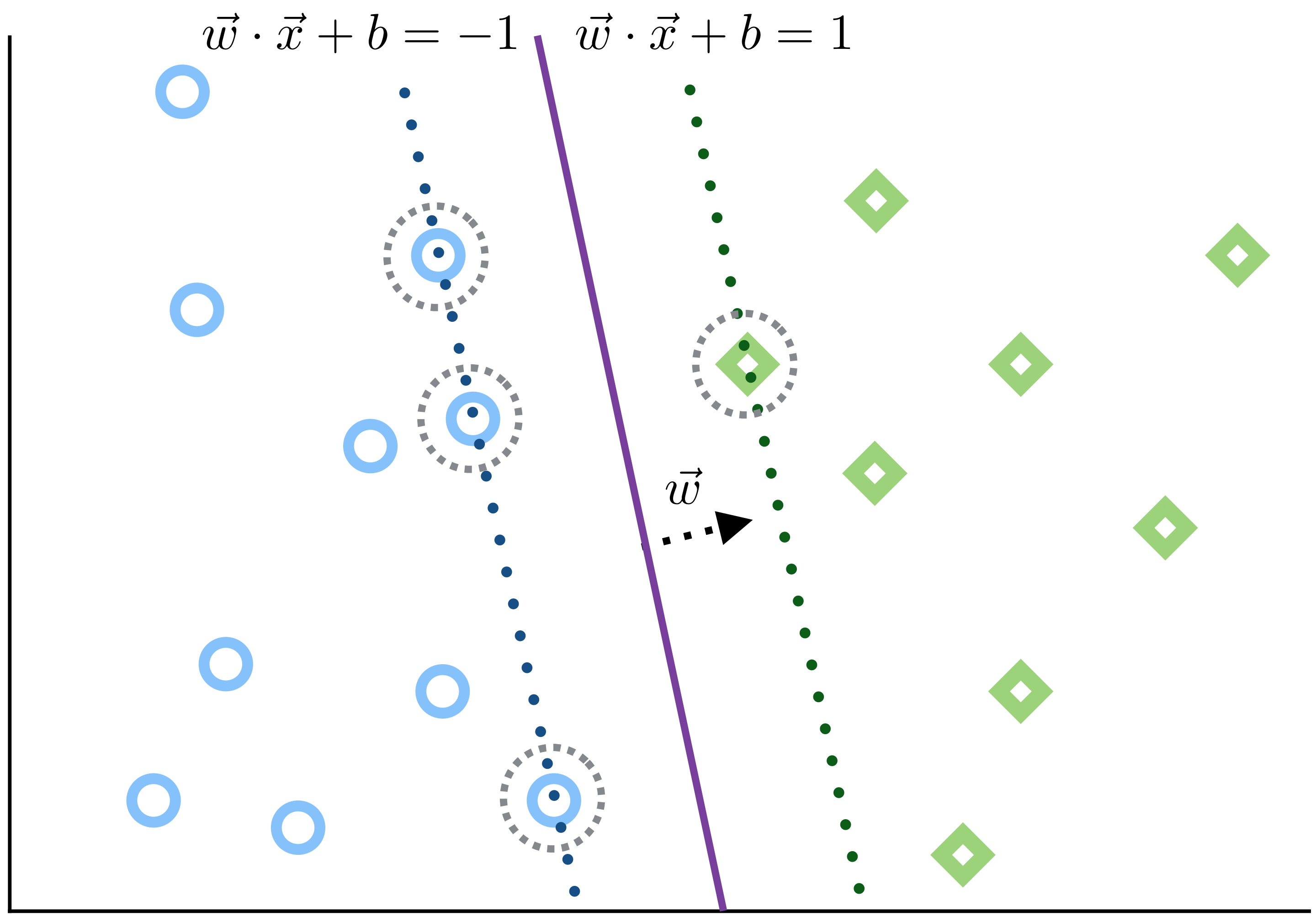
- As a rather modern machine learning algorithm, SVM is not widely used in the HEP data analysis, with few exceptions. In this talk I want to focus on this particular implementation.
- We used a discovery significance based hyperparameter tuning algorithm.
- We introduced an SVM classifier interface (SVM-HINT), which is based on a widely used SVM library (LIBSVM), tailored for HEP searches.
    - Performance and optimization of support vector machines in high-energy physics classification problems
    - https://github.com/ml-hint/svm-hint

Support Vector Machines



- What is the optimal way to separate two linearly separable distributions?

Support Vector Machines



- SVM provides a unique solution to separate two class problems i.e. signal from background.

$$\mathscr{L}(\vec{w}, \vec{x}, \alpha) = \frac{1}{2}|\vec{w}|^2 - \sum \alpha_i [y_i(\vec{w} \cdot \vec{x} + b) - 1]$$

Karush Kuhn Tucker Conditions

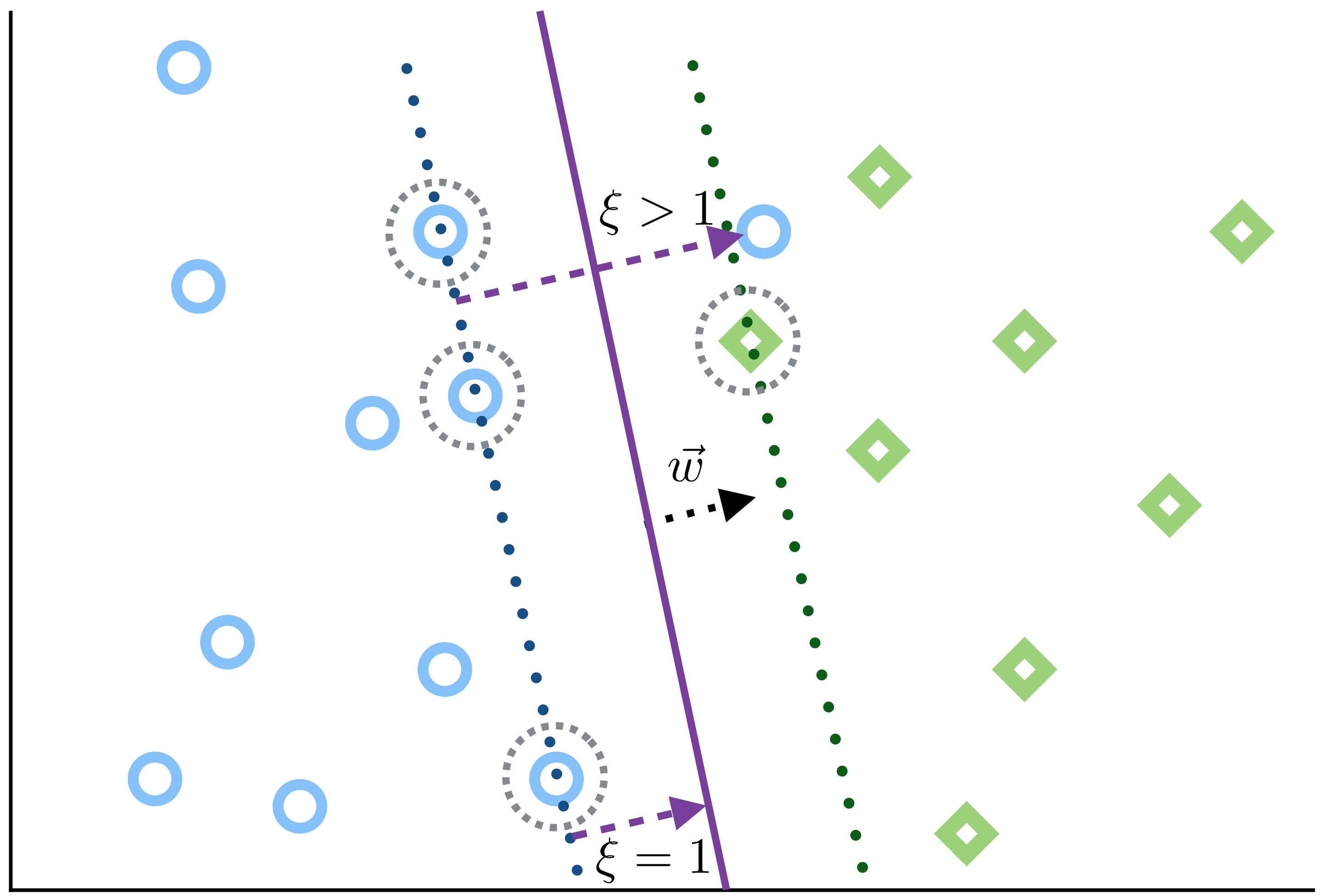
- If the problem has a solution and it satisfies the following conditions:

$$\vec{\alpha} \geqslant 0,$$
$$\alpha_i[y_i(\vec{w} \cdot \vec{x}_i + b) - 1] = 0, \qquad i = 1, ..., N.$$

- These conditions are known as the KKT conditions.

- The KKT conditions provide a generalization of the method of Lagrange multiplier for the case of inequalities .

Support Vector Machines

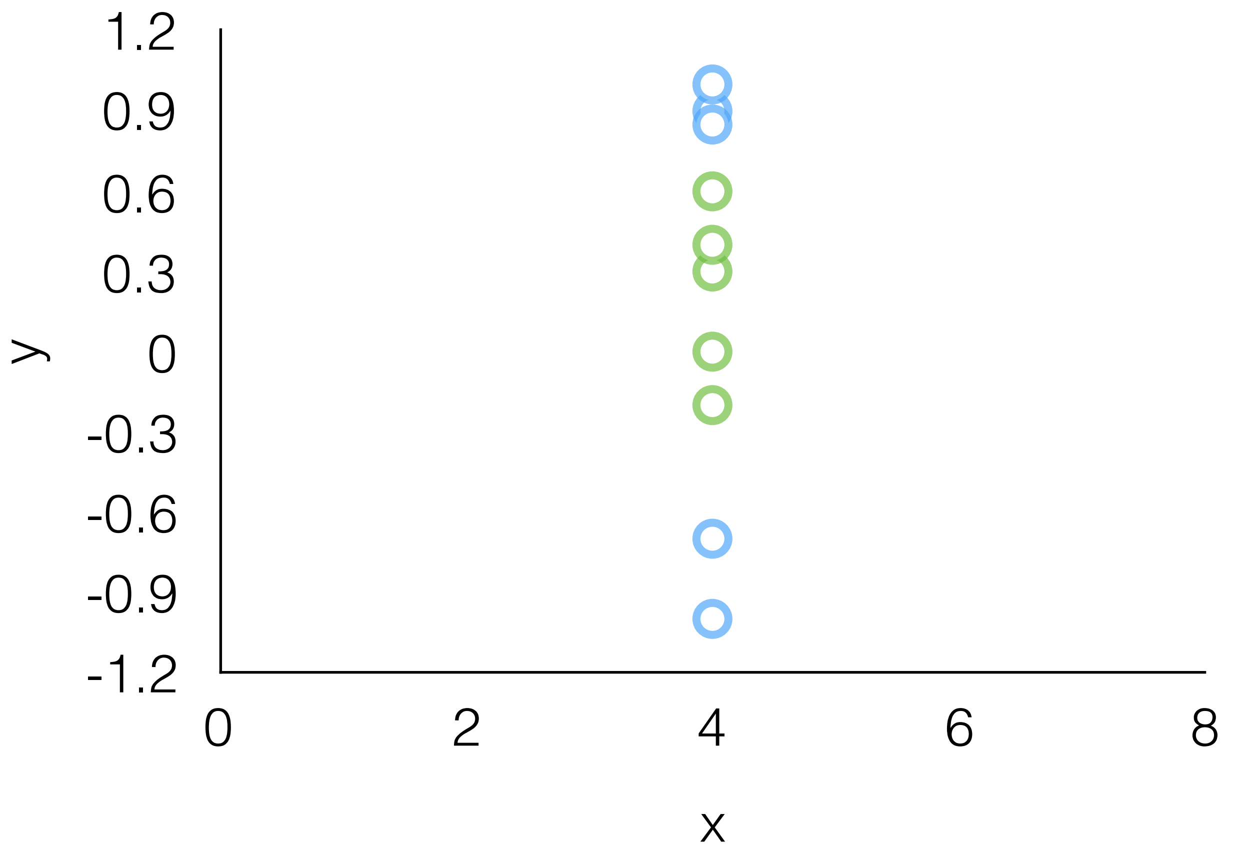


- Method of Lagrange multipliers — can be generalized respecting KKT conditions.
- Slack variables:

$$\mathscr{L} = \frac{1}{2}|\vec{w}|^2 + C\sum_i \xi_i - \sum \alpha_i[y_i(\vec{w}\cdot\vec{x}+b) - 1 + \xi_i] - \sum \beta_i \xi_i$$

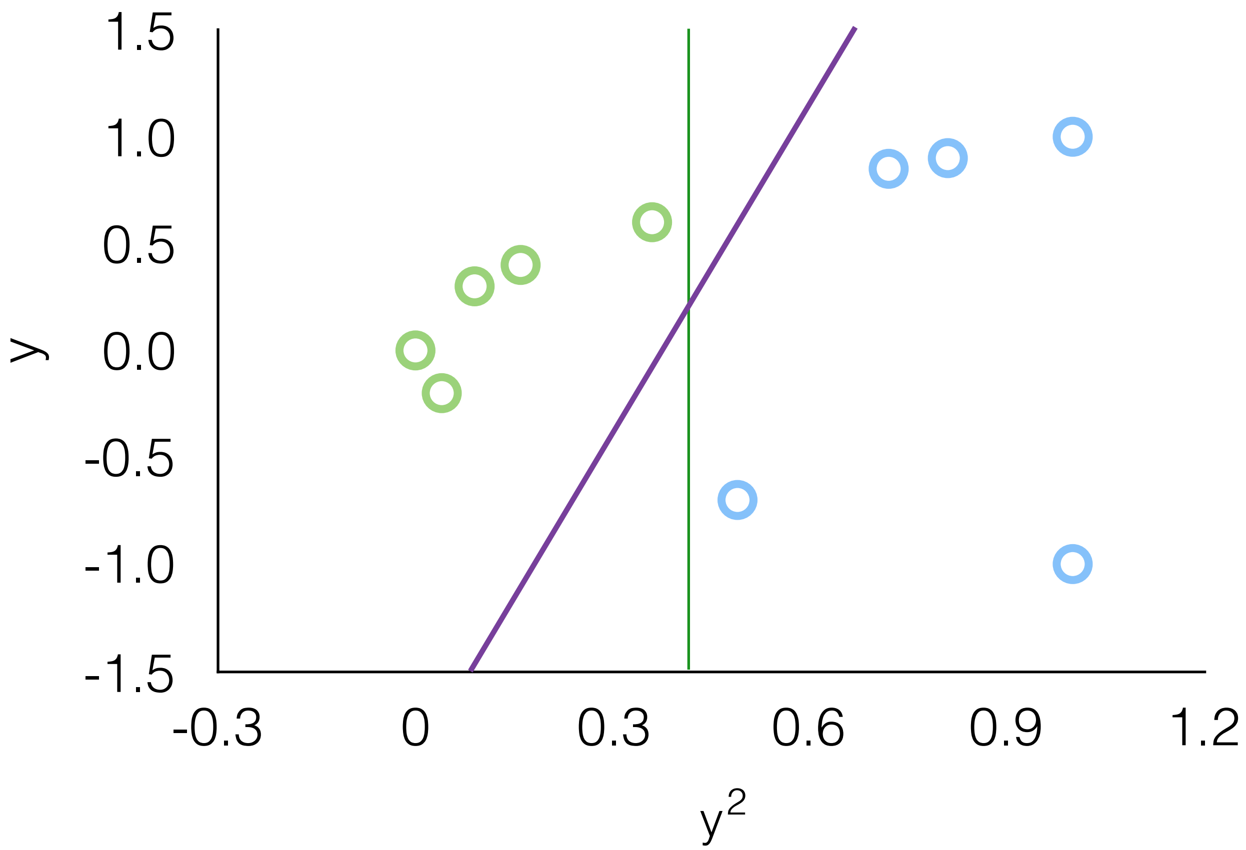
SVM: Non-linear case

- Is it possible to classify these samples with a linear decision function?



SVM: Non-linear case

- With a mapping in the form of $x \rightarrow y^2$
- Calculating a non-linear mapping may become cumbersome for large number of dimensions.



SVM: Non-linear case

- The decision function of SVM must be linear.
- Two non-linearly separable distributions can be linearly separable in a non-linearly transformed space.
- Calculating a non-linear mapping may become cumbersome for large number of dimensions (in some cases infinite).
- Way out: SVM uses a 'kernel trick' to circumvent this problem. By imposing the stability conditions, the Lagrangian can be reduced to the functional:

$$\mathscr{W}(\vec{\alpha}) = -\frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j (\vec{x}_i \cdot \vec{x}_j) + \sum \alpha_i$$

- Various kernels are available (Mercer's theorem). In this study, RBF (Radial Basis Function) kernel is used:

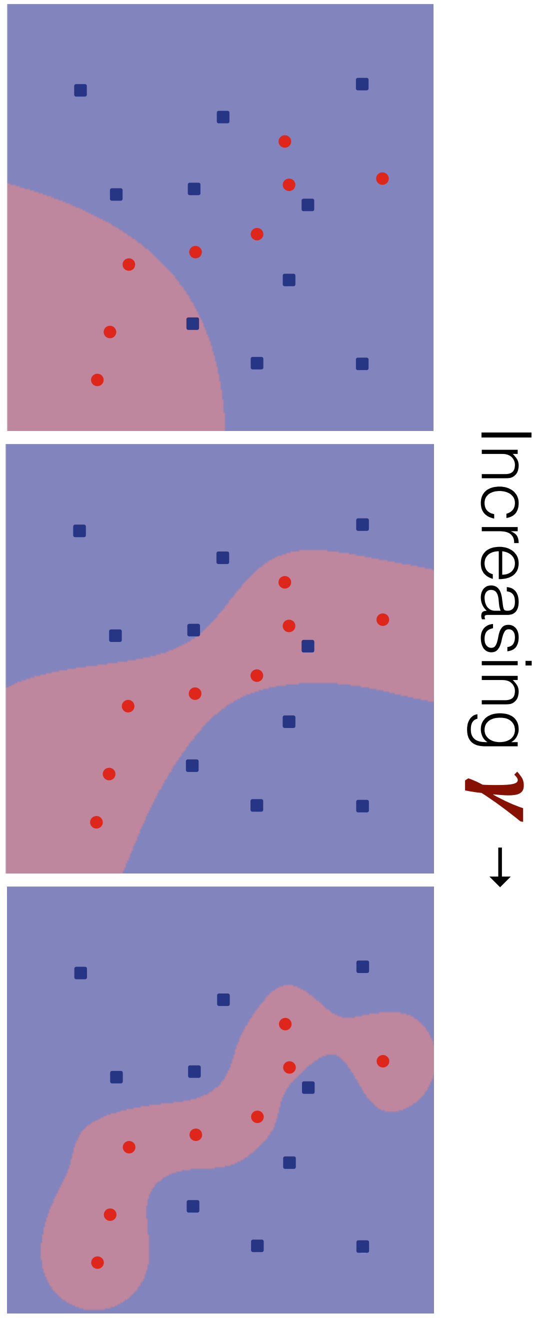
$$K(\vec{x}_i, \vec{x}_j) = e^{-\gamma|\vec{x}_i - \vec{x}_j|^2}$$

libSVM

- libSVM (Chang et. al.) is a well tested and widely used SVM library.
  - http://www.csie.ntu.edu.tw/~cjlin/libsvm/
- libSVM is using an improved version of Sequential Minimal Optimization.
- The library is optimized and tested over many years.

Hyperparameter optimization

- SVM with RBF kernel has two hyper-parameters which need to be set before the application:
  - the slack variable parameter **C**,
  - the inverse kernel width $\boldsymbol{\gamma}$.
- A discovery significance based algorithm outperforms other performance measures for the physics searches.
- We implemented a custom grid search algorithm for the Hyper-parameter optimization based on the Asimov significance estimator.



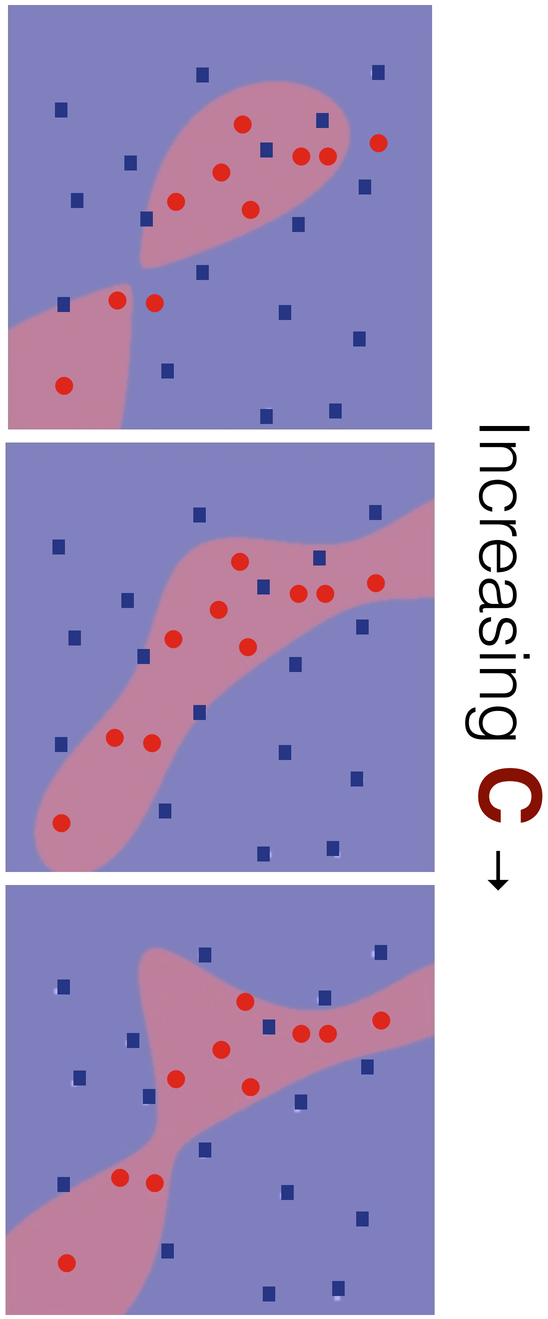
Two classes: Blue is background and red is signal

Hyperparameter optimization

- A custom iterative grid search algorithm is employed to optimize the hyper-parameters:
  - In order to prevent over-training in the least computationally cumbersome way:

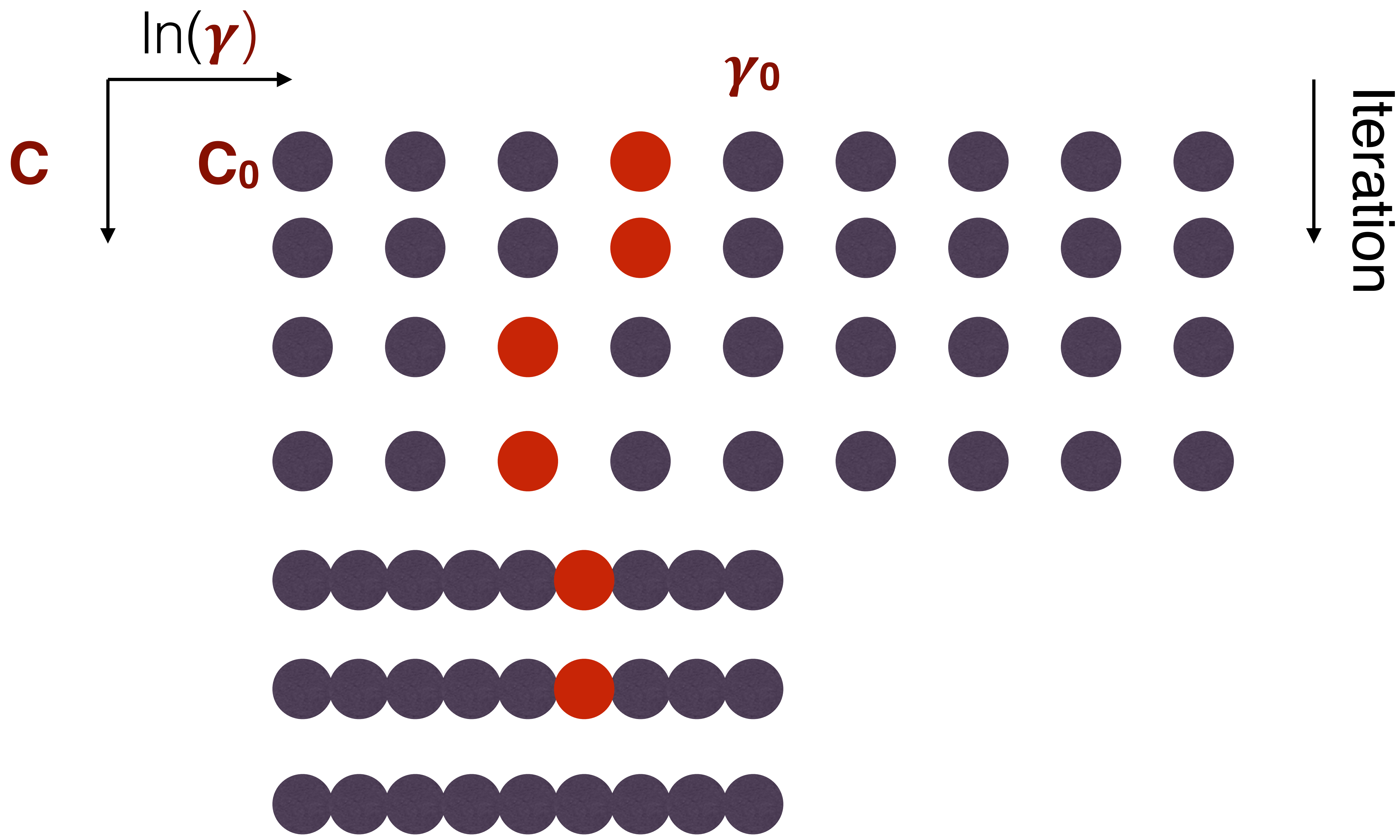
    $$\tilde{Z} = Z_A^{(\text{test})}\left[1 - \frac{|Z_A^{(\text{test})} - Z_A^{(\text{train})}|}{Z_A^{(\text{test})} + Z_A^{(\text{train})}}\right]$$

  - Starting from an initial set, **C** and $\boldsymbol{\gamma}$ parameters are scanned with a focusing parameter (see the next slide)

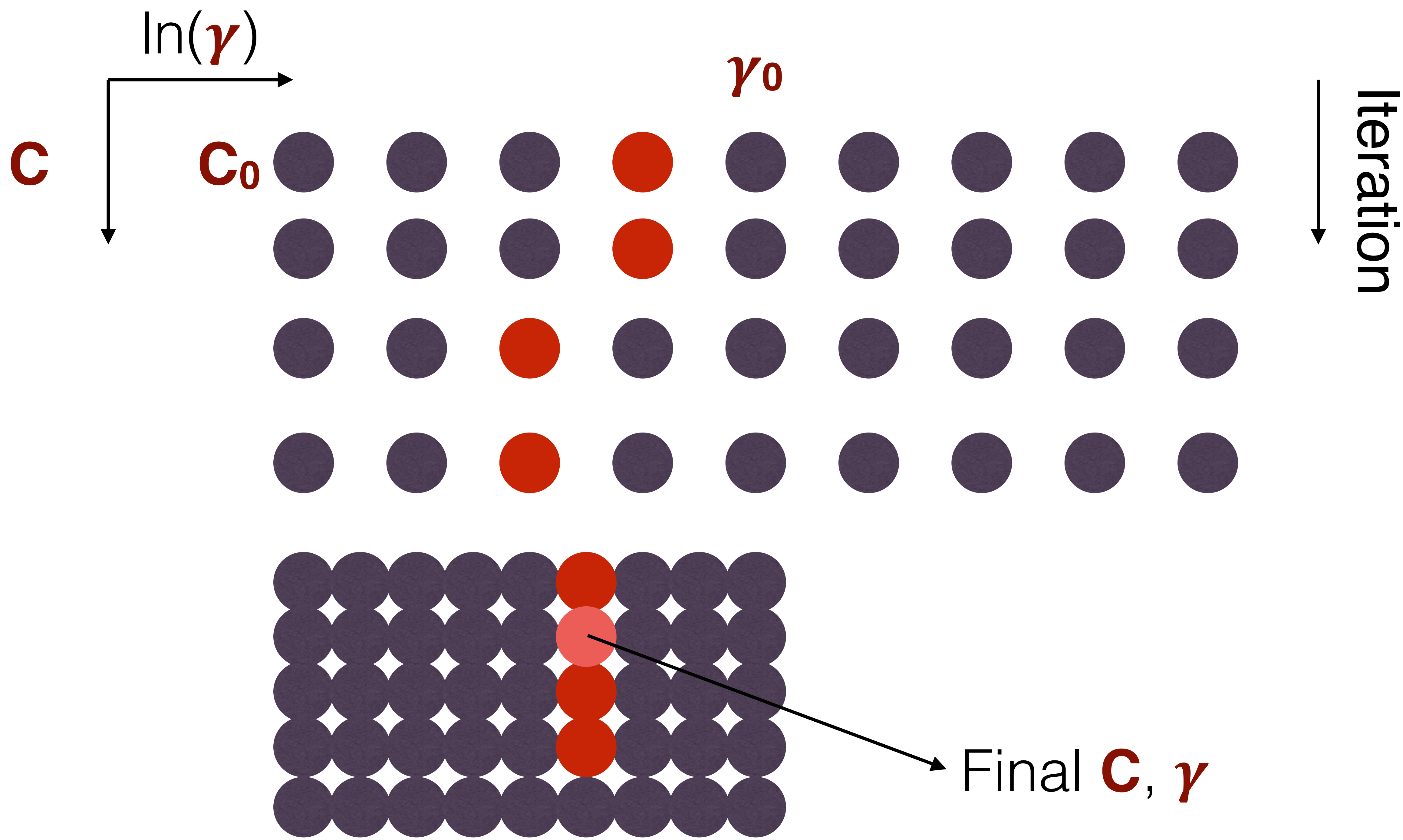


Two classes: Blue is background and red is signal

Hyperparameter optimization



Hyperparameter optimization



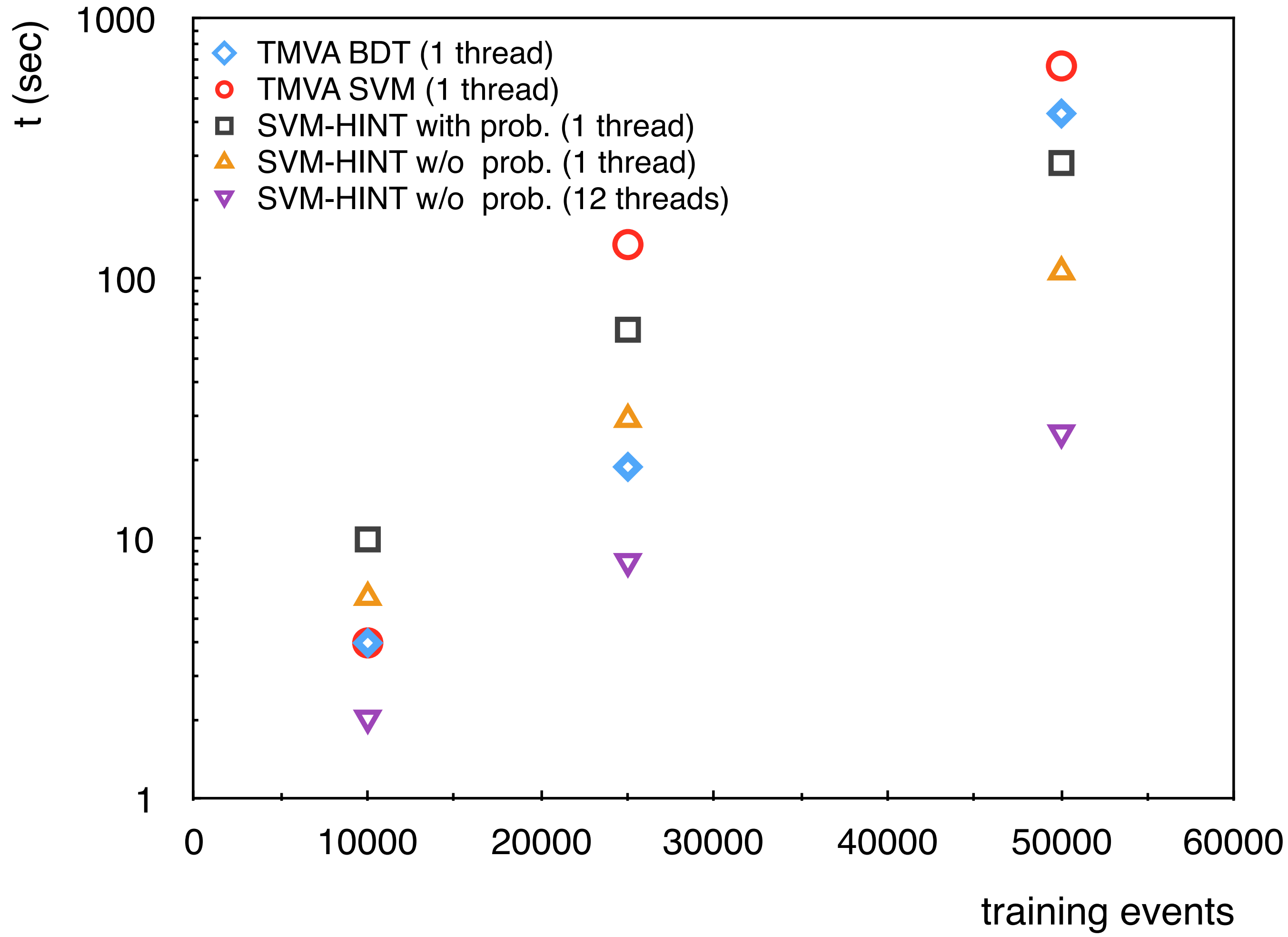
Case Study I

- A simple toy example is used to compare the time performance of the implementations.
- The variables are sampled from:

$$\begin{array}{lcll} V_1 = & \sin(x_1); & & x_1 \sim g(x_1|a,b) \\ V_2 = & x_2; & & x_2 \sim \exp(-x_2/c) \\ V_3 = & x_3; & & x_3 \sim g(x_3|d,e) \\ V_4 = & \sqrt{x_4}; & & x_4 \sim \exp(-x_4/f) \end{array}$$

- TMVA-BDT is used as a benchmark implementation.

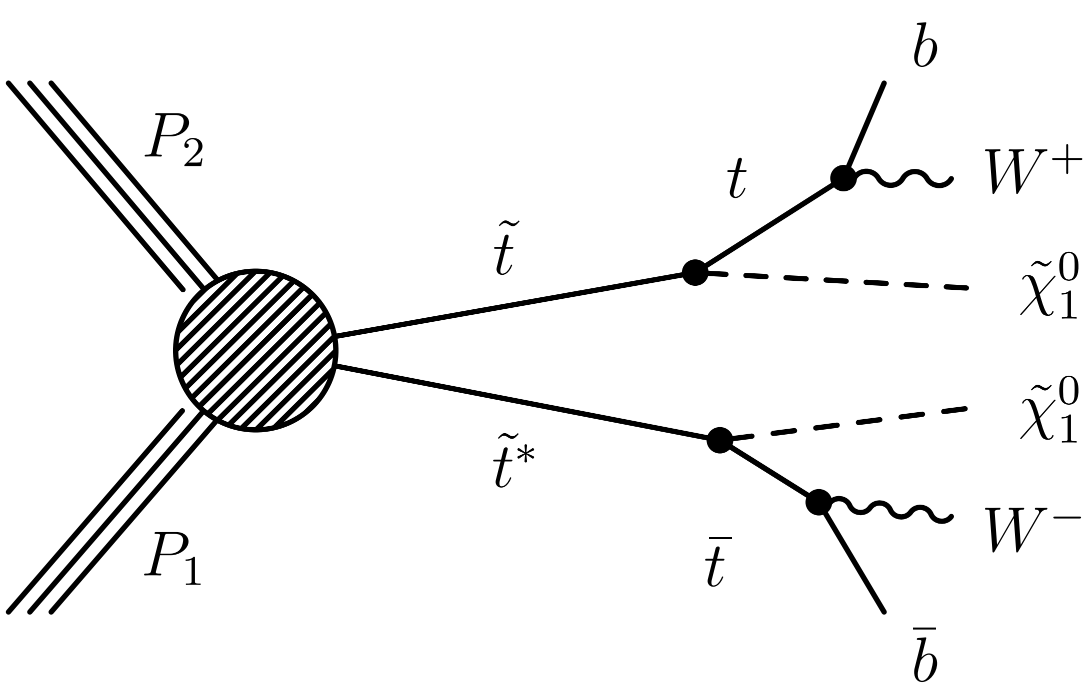
Case Study I: Runtime performance



- The algorithms are optimized to have similar accuracies.
- SVM-Hint with 12 threads has the **fastest timing performance**.
- SVM-Hint scales with number of training events.
- BDT almost competitive in speed with lower number of training events.

Case Study II

- Search for supersymmetry in single lepton final states (14 TeV).
- A benchmark SUSY model - an SMS simplified model. The particular mass parameters: Top quark super partner mass of 900 and LSP mass 100 GeV.
- Only the dominant SM background $t\bar{t} \to l(l + \mathrm{jets})$ is and only this background is considered for the sake of simplicity.
- Simulations performed with Delphes fast simulation program including pileup (same as used for Snowmass study)



T2tt

Özgür Sahin

Case Study II

- 25 variables are considered to discriminate signal and background.
- The variables are grouped into four different sets with respect to their complexity.
- Low-level variables include basic features of physics objects whereas high-level variables are constructed from these objects.
- TMVA-BDT is used as a benchmark implementation.

	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	•		•	
	$\Delta\mathrm{r}_{\min}(l,b)$	•		•	
	$\Delta\phi_{\min}(j_{1,2},\not{E}_{\mathrm{T}})$	•		•	

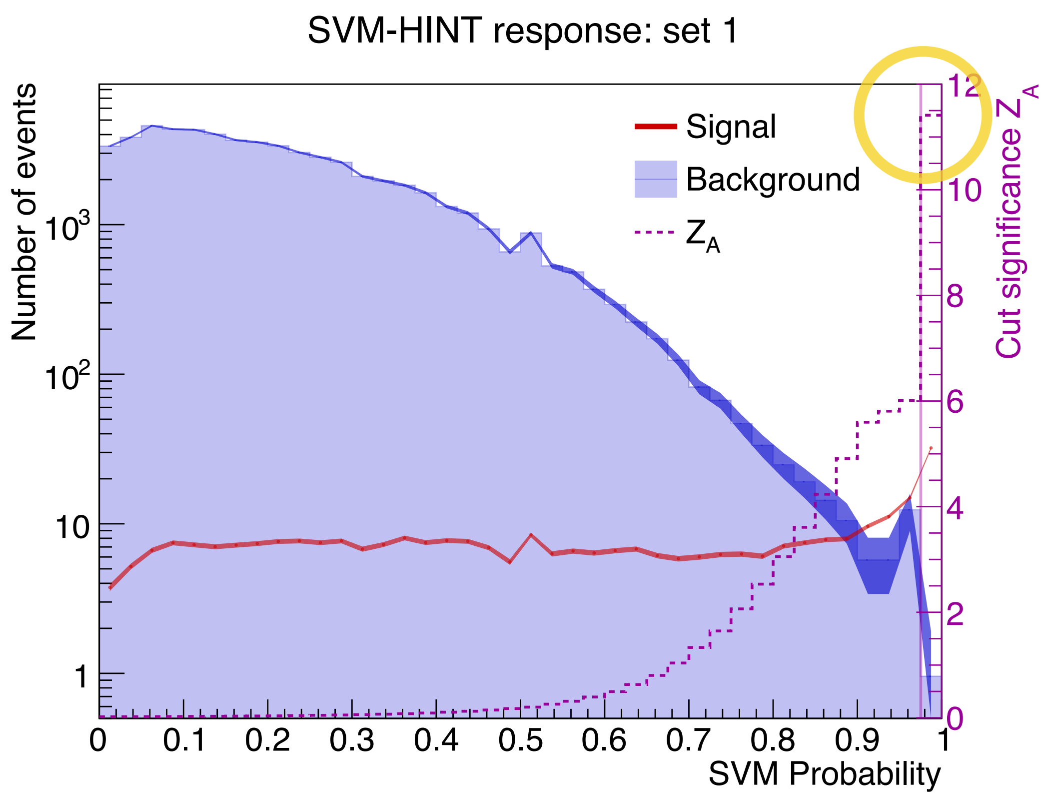
SVM-HINT

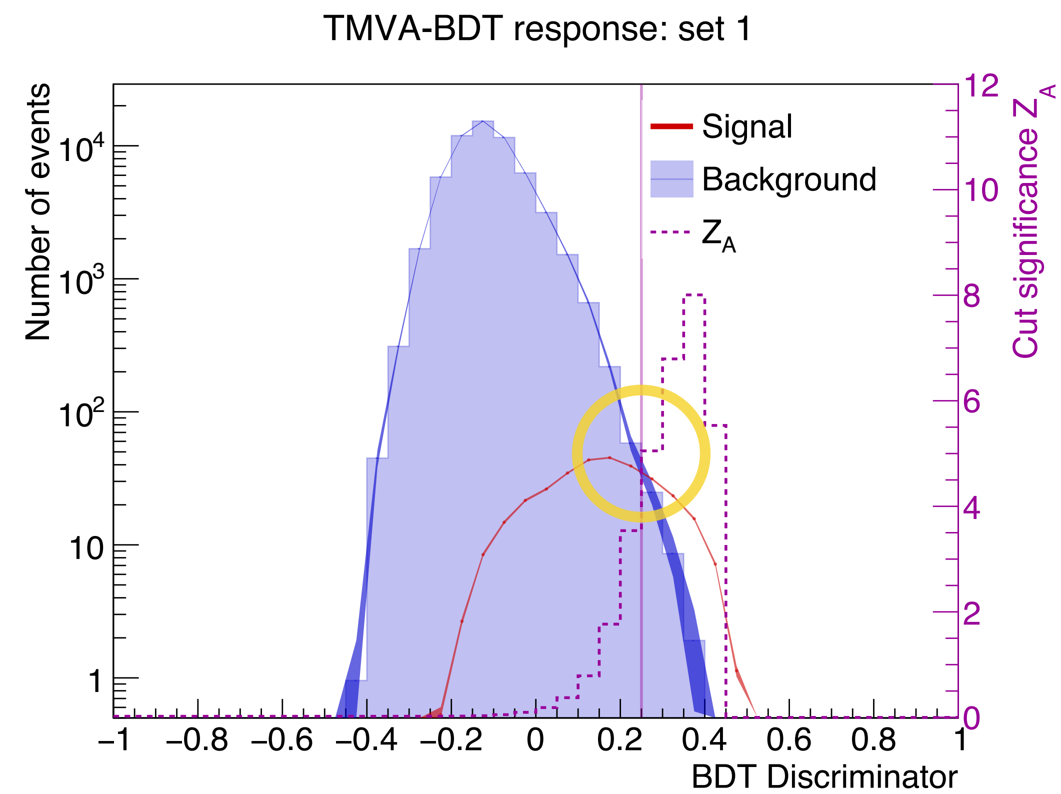
	SVM-HINT			TMVA-BDT			
Set	N_s	N_b	$Z_A^{\max}$	N_s	N_b	$Z_A^{\max}$	Z_A^{prov}
1	32.1 ± 0.6	1.0 ± 1.0	11.4	24.1 ± 0.5	1.9 ± 1.4	8.0	5.1
2	23.2 ± 0.5	23.9 ± 4.8	2.5	16.2 ± 0.4	10.5 ± 3.2	3.0	1.5
3	37.8 ± 0.6	9.6 ± 3.0	6.1	40.5 ± 0.7	9.6 ± 3.0	6.4	3.7
4	33.4 ± 0.6	20.1 ± 4.4	3.7	40.5 ± 0.7	35.4 ± 5.8	3.0	1.8

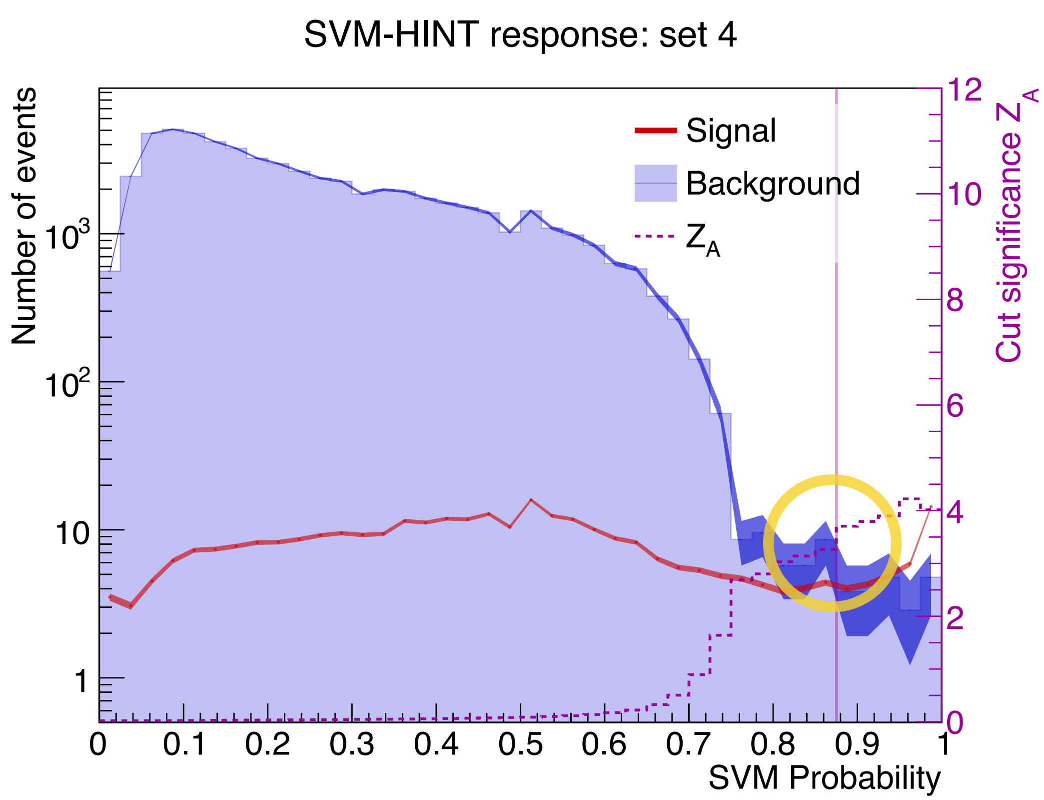
- https://github.com/ml-hint/svm-hint
- SVM-HINT provides typically **strong classification out of the box.**

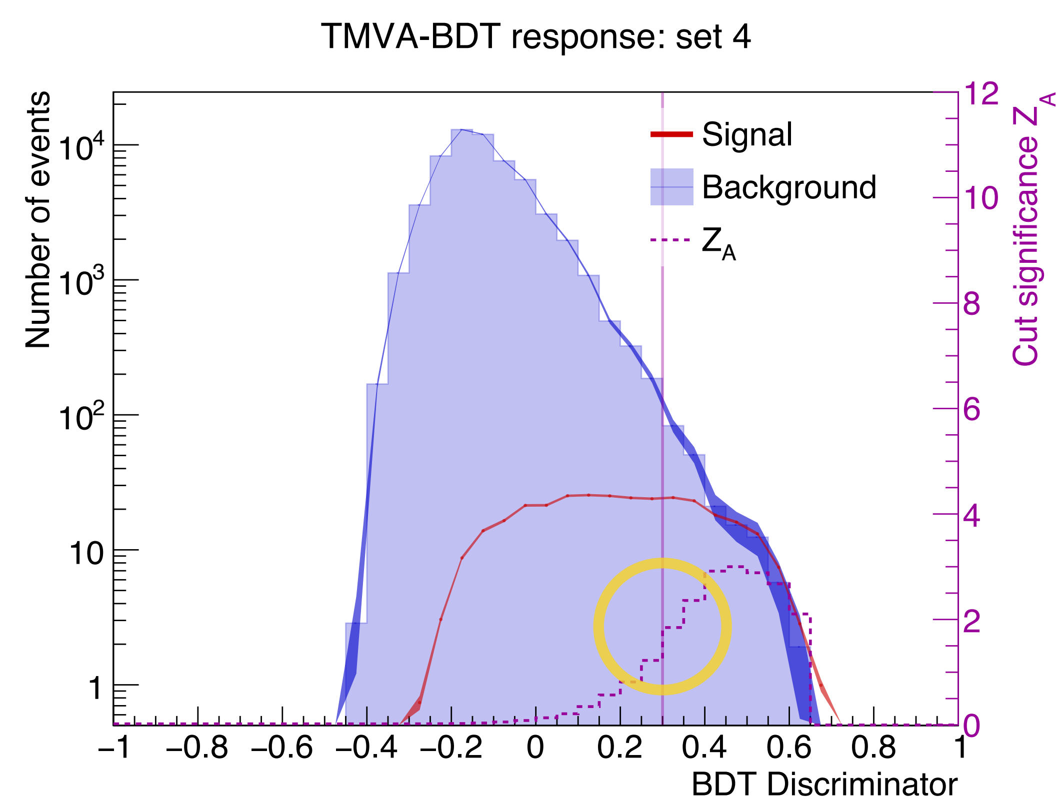
Sahin et al DOI:10.1016/j.nima.2016.09.017

Case Study II: Classifier Responses





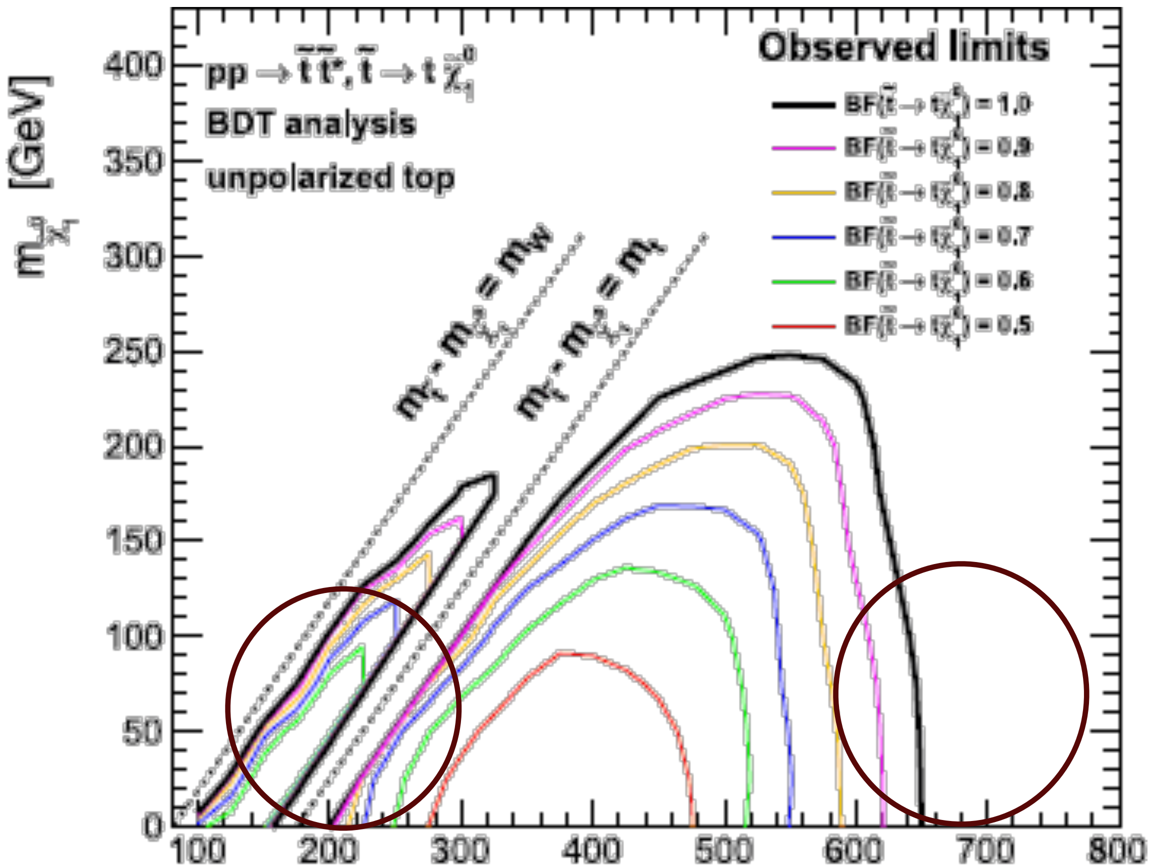




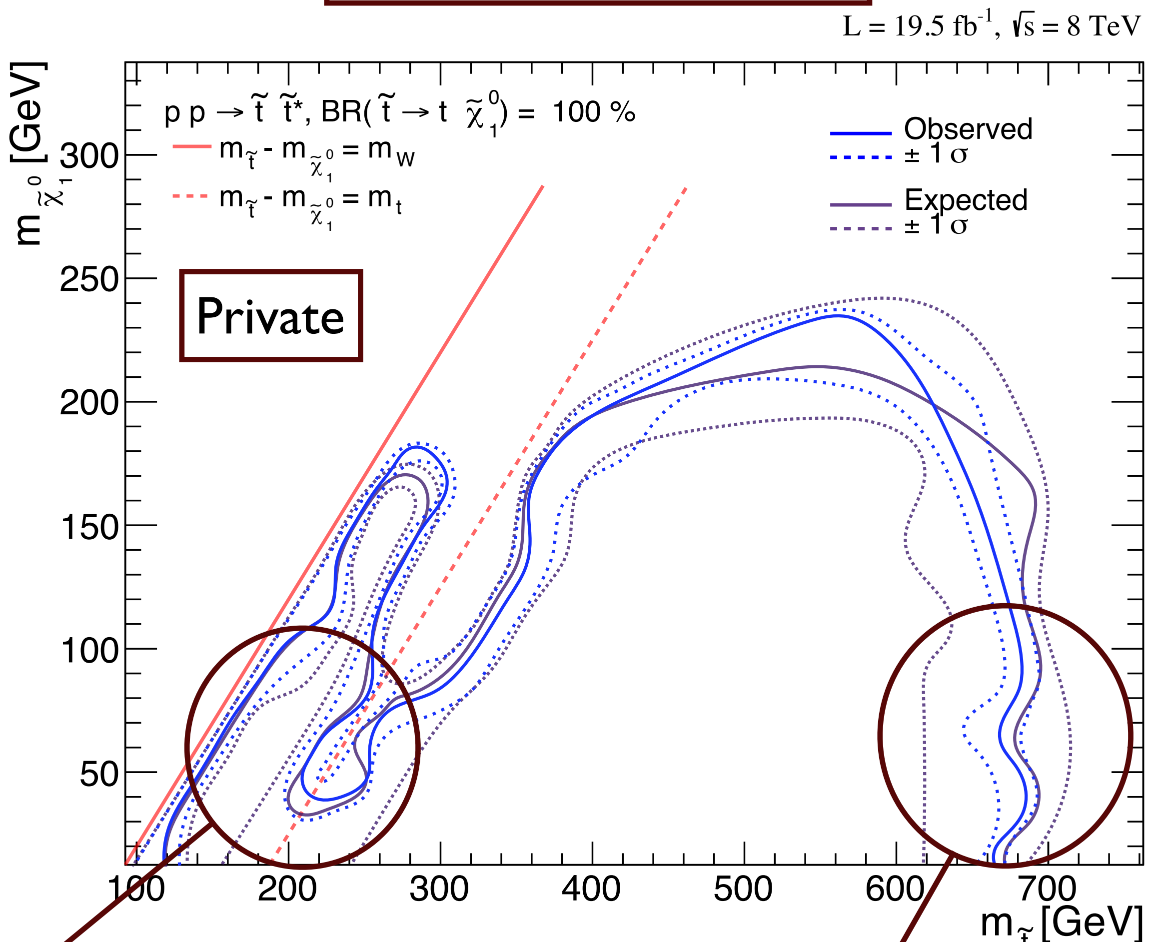
Both SVM-HINT and TMVA-BDT benefit from high number of variables.

Comparison of the results

CMS Col., EPJ C 73 (2013) 2677



DESY-THESIS-2017-014



- Improvement in the **compressed region** and in the **high top-squark mass**.

Conclusion

- A range of machine learning algorithms is widely employed in collider physics.
  - IMHO, we are still not at the front-line…
  - The possibilities are vast, but the support is limited. I hope collaborations with other disciplines also help people to realize the importance.
- As an unsupervised learning algorithm DBSCAN clustering algorithm is presented. DBSCAN is proven to be fast, flexible and resilient against possible noise.
  - DBSCAN seems to perform better under high-PU.
- An SVM interface for the HEP data classification problems using a statistical significance (Asimov Significance estimator) based hyperparameter optimization is presented. Available on Github:
  - https://github.com/ml-hint/svm-hint
  - The implementation provides a strong classification out of the box with a built-in autonomous optimization.

thank you!

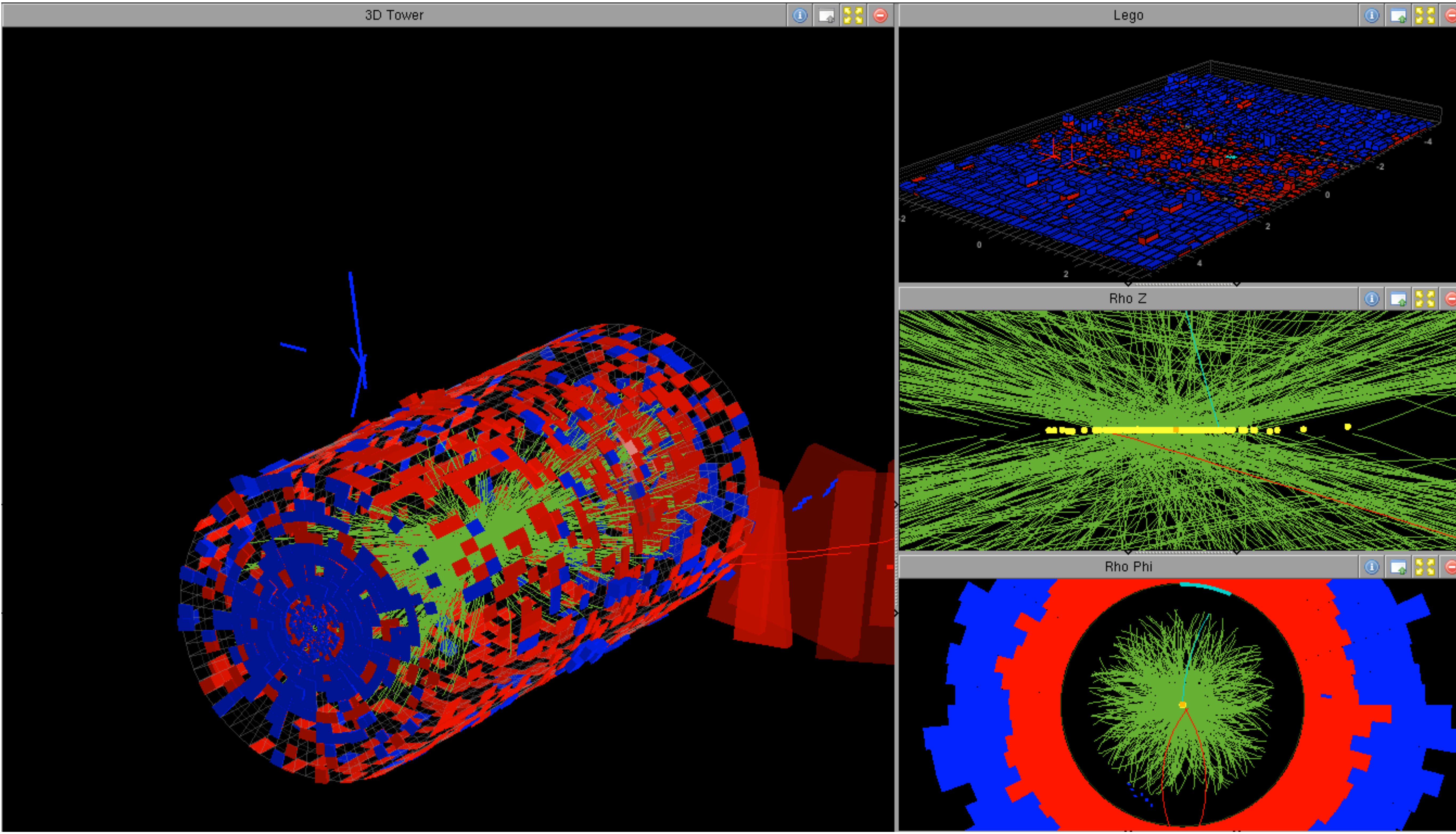
References

References

- TMVA Package (a complete and easy to use HEP - ML software) built in to the ROOT framework (a C++ based CERN - Fermilab developed analysis framework for Big Data analysis):
  - TMVA: https://github.com/root-project/root/tree/master/tmva
  - ROOT:https://github.com/root-project/root/
- Some books for interested (the first book is more of the fundamentals the second book is related to Deep Learning which is not covered in this talk):
  - https://www.amazon.com/Pattern-Recognition-Learning-Information-Statistics/dp/0387310738
  - https://www.amazon.com/Deep-Learning-Adaptive-Computation-Machine/dp/0262035618/
- Cowan et al (Asimov Paper) Eur. Phys. J .C71:1554, 2011
- Sahin et al (SVM-HINT paper) DOI:10.1016/j.nima.2016.09.017

Backup

CMS event display



Asimov Significance

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

- http://www.pp.rhul.ac.uk/~cowan/atlas/cowan_atlas_15feb11,
  - * Eur. Phys. J .C71:1554, 2011.
- Gives a very close estimation of the quoted significances.

$\hat{\mu}_b$	1.3	3.8	3.8	388.6	493434	2109732
$s = n - \hat{\mu}_b$	4.7	5.2	13.2	134	4992	9717
$f = \sigma_b/\hat{\mu}_b$	0.231	0.237	0.158	0.0207	0.00142	0.000206
Quoted Z	2.7	1.9	4.6	5.9	5.0	6.4
Z_A	2.8	2.0	4.6	5.9	5.0	6.4