Automatic machine learning and experimental applications in nuclear fusion

J. Vega
Asociación EURATOM/CIEMAT para Fusión
jesus.vega@ciemat.es

CIEMAT

DIRECCIÓN GENERAL
Cayetano López Martínez
DIRECCIÓN GENERAL ADJUNTA
Ramón Gavela González
SUBDIR. GRAL. SEGURIDAD Y MEJORA DE LAS INSTALACIONES
Javier Quiñones Díez
SUBDIR. GRAL. RELACIONES INSTITUCIONALES Y TRANSFERENCIA DEL CONOCIMIENTO
Margarita Vila Pena
SECRETARÍA GENERAL
Ana Collados Martín-Posadillo
DEPARTAMENTO DE ENERGÍA
Ramón Gavela González
LABORATORIO NACIONAL DE FUSIÓN
Joaquín Pedro Sánchez Sanz
DEPARTAMENTO DE MEDIO AMBIENTE
Yolanda Benito Moreno
DEPARTAMENTO DE TECNOLOGÍA
José Manuel Pérez Morales
DEPARTAMENTO DE INVESTIGACIÓN BÁSICA
Marcos Cerrada Canales
GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y COMPETITIVIDAD
SECRETARÍA DE ESTADO DE I+D+I
CIEMAT

Data acquisition Unit

- TJ-II data acquisition
- Development of innovative data analysis techniques
  - JET and TJ-II
- EFDA Topical Group on Diagnostics
  - Area leader
- Participation in JET experiments
  - Scientific coordinator
- ITER projects
  - Design and development of prototypes for fast data acquisition
  - Design and development of prototypes for interlock fast controllers
  - Framework Service Contract for provision of System and Instrumentation Engineering Support
  - Nominal Device Support for FlexRIO 796xR and Integration with MFC Diagnostics Use Case
- **Collaborations**
  - Universities: UNED, UCM, UPM, U. of London and UPC
  - EURATOM Associations: Consorzio RFX, IST and CCFE
  - Large companies: INDRA, Iberdrola, National Instruments, GAMCO and SGENIA

Overview

- Data analysis in modern science
- Machine learning and supervised classifiers
- Reliable classifiers
- Applications in fusion

Data production rate in modern science

- JET
  - 100 Tbytes
- ITER
  - Pbytes/year
- European Space Agency
  - The satellites produce hundreds of Gbytes/day
- Large Hadron Collider (CERN)
  - ATLAS y CMS are expected to produce 10 Pbytes/year
- Hubble
  - It has sent 10 Tbytes of data

Databases in science

Nowadays, processing all the information of present databases is a much more important issue than acquiring more data

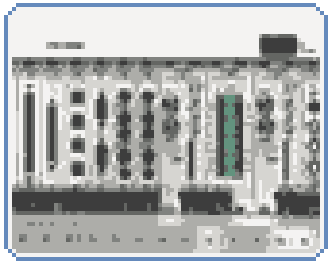
None of the above databases can be analyzed by humans or even by the best algorithms (in terms of speed and memory requirements)

This does not mean to take less measurements

Modern science and data analysis

New knowledge

Data understanding
- Computer aided understanding



Data Mining (or Knowledge Discovery in Databases):
- Pattern or correlation search
- Clustering analysis, automated classification
- Outlier / anomaly searches
- Hyperdimensional visualization

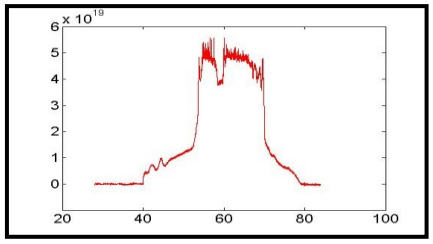
Data access
- Storage/Archiving
- Indexing, searchability

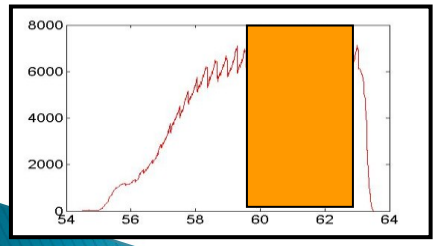
Data gathering
(from multiple inputs)

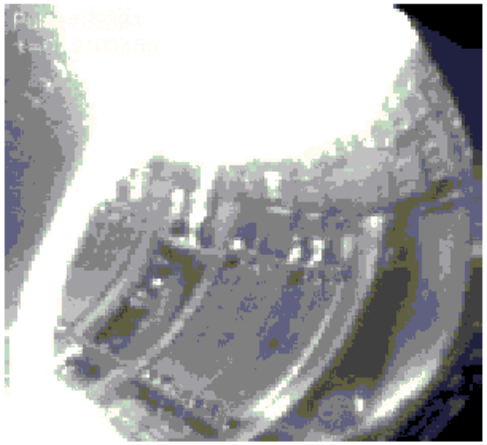
Data access

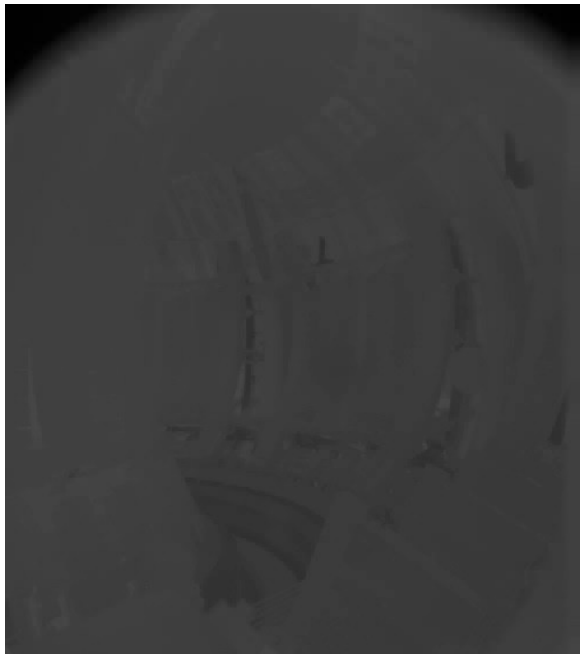
- Intelligent data retrieval of waveforms and images based on patterns from massive databases (JET and TJ-II)

	Input	Output
Classical model	Shot number	Samples
New model	Pattern	Shot numbers Locations (temporal, spatial)









HPC can be required

Data mining

- Real-time prediction of disruptions in JET
- Intelligent system for feature extraction to characterize L/H transitions in JET and DIII-D
- Automatic determination of L/H transition times in JET
- Automatic determination of L/H transition times in DIII-D
- Automatic detection of plasma events in waveforms and video-movies (JET)
- Automatic ELM location in JET
- Automatic analysis system in the TJ-II Thomson Scattering based on pattern recognition
- Noise reduction in images (TJ-II Thomson scattering)
- Outlier detection in massive databases
- Identification of "outer modes" in JET discharges
- Spatial location of local perturbations in plasma emissivity derived from projections using conformal predictors
- Change of context detection by using exchangeability
- Development of active learning methods for classification purposes

HPC is required

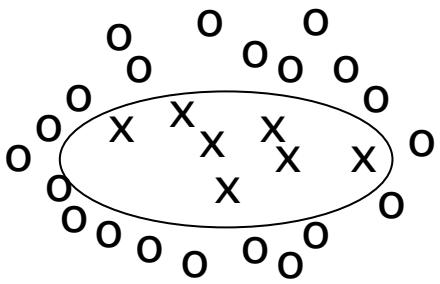
Data understanding

- Set-up of a distributed HPC environment to include *ad-hoc* software codes

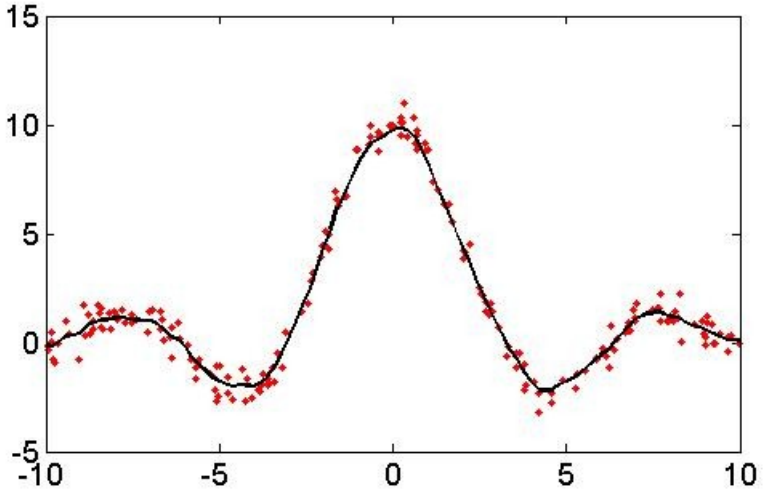
Machine learning

Machine learning: concepts

- Learning does not mean '*learning by heart*' (any computer can memorize)
- Learning means '*generalization capability*': we learn with some samples and predict for other samples
- The learning problem is the problem of finding a desired dependence (function) using a *limited* number of observations (training data)
  - Classification: the function can represent the separation frontier between two classes
  - Regression: the function can provide a fit to the data



Classification



Regression

Description of objects

Dataset: $(\boldsymbol{x}_1, y_1)$, $(\boldsymbol{x}_2, y_2)$, ..., $(\boldsymbol{x}_i, y_i)$, ..., $(\boldsymbol{x}_N, y_N)$

$\boldsymbol{x}_i \in \mathbf{R}^m$: features that are of distinctive nature (object description with attributes managed by computers)

y_i: label of the sample $\boldsymbol{x}_i$

if $y_i \in \{L_1, L_2, ..., L_K\}$: classification (pattern recognition) problem

if $y_i \in \mathbf{R}$: regression problem

Classification

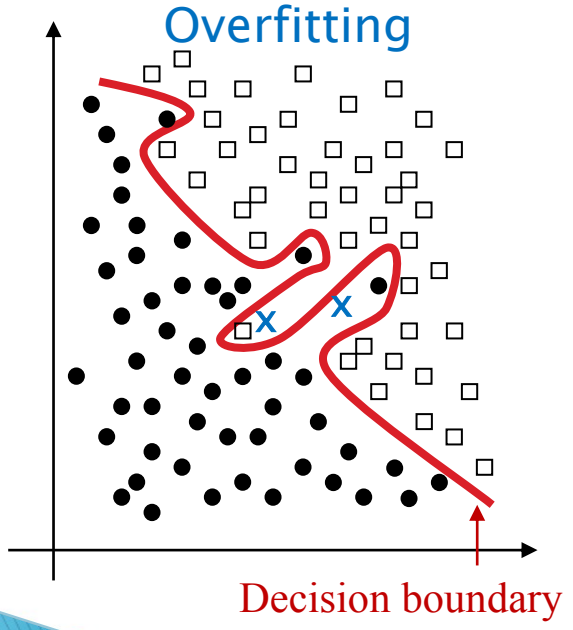
Supervised classifiers

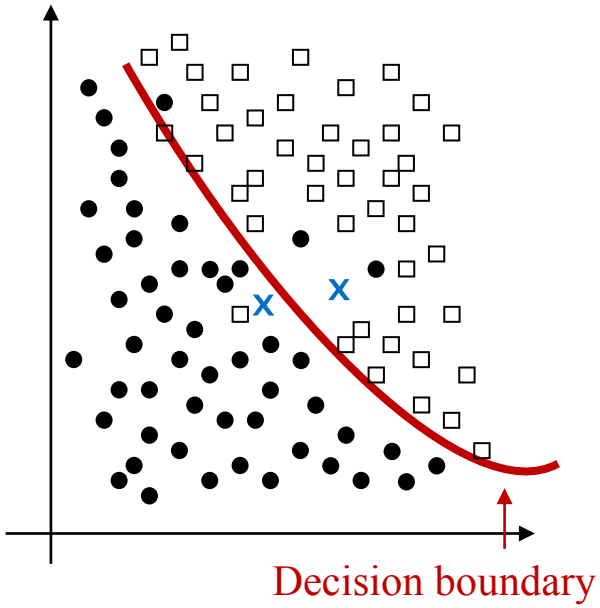
Dataset: $(\boldsymbol{x}_1, y_1)$, $(\boldsymbol{x}_2, y_2)$, …, $(\boldsymbol{x}_i, y_i)$, …, $(\boldsymbol{x}_N, y_N)$

$\boldsymbol{x}_i \in \mathbf{R}^m$: features that are of distinctive nature (object description with attributes managed by computers)

$y_i \in \{L_1, L_2, \ldots, L_K\}$: *known label* of the sample $\boldsymbol{x}_i$

Objective: to determine a separating function between classes (generalization) to predict the label of new samples with known feature vectors $((\boldsymbol{x}_{N+1}, y_{N+1}), (\boldsymbol{x}_{N+2}, y_{N+2}), \ldots)$





Supervised classifiers

- How good is a classifier?

Dataset: $(\boldsymbol{x}_1, y_1)$, $(\boldsymbol{x}_2, y_2)$, ..., $(\boldsymbol{x}_N, y_N)$ {

$(\boldsymbol{x}_i, y_i)$, i = 1, ..., J: training set

$(\boldsymbol{x}_i, y_i)$, i = J+1, ..., N: test set

- Training set: a model is created to make predictions. Given $\boldsymbol{x}_i$, the model predicts y_i

- Test set: model validation. The success rate is taken as the level of confidence and *it is assumed to be the same* for all future samples

Unsupervised classifiers

Dataset: $(\boldsymbol{x}_1, y_1)$, $(\boldsymbol{x}_2, y_2)$, …, $(\boldsymbol{x}_i, y_i)$, …, $(\boldsymbol{x}_N, y_N)$

$y_i \in \{L_1, L_2, \ldots, L_K\}$: class labels of the training samples $\boldsymbol{x}_i$ are not available

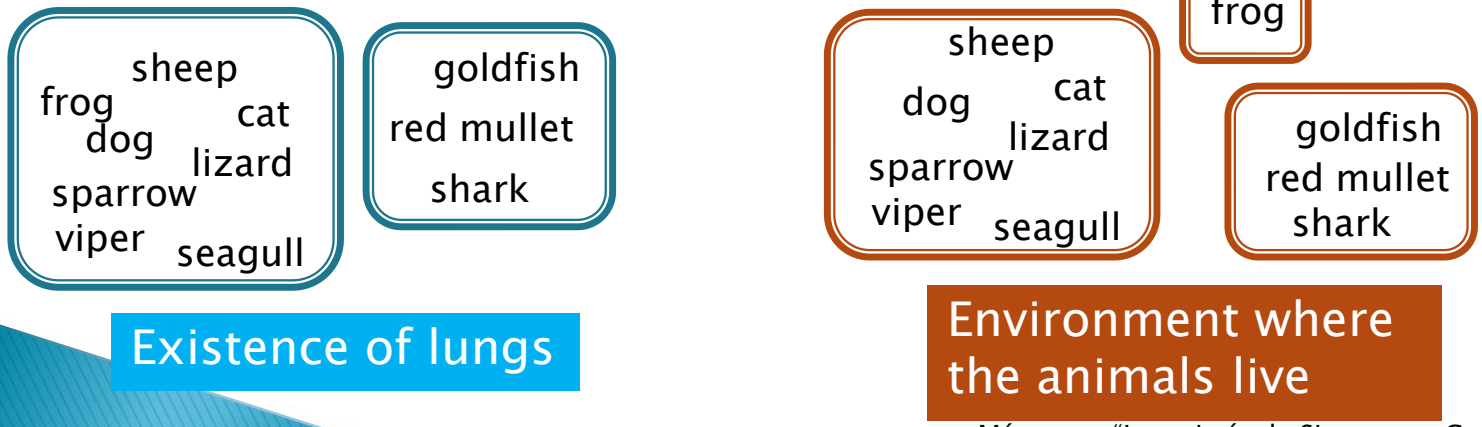
Objective: to "*reveal*" the organization of the samples into a number of "*sensible*" clusters which will allow us to discover similarities and differences among samples and to derive useful conclusions about them

- The labels are known just after the training
- A clustering criterion is needed: **proximity measure (distance or similarity)**

{sheep, dog, cat, sparrow, seagull, viper, lizard, goldfish, red mullet, blue shark, frog}

mammals | birds | reptiles | fish | amphibians

Class of animals



Existence of lungs

Environment where the animals live

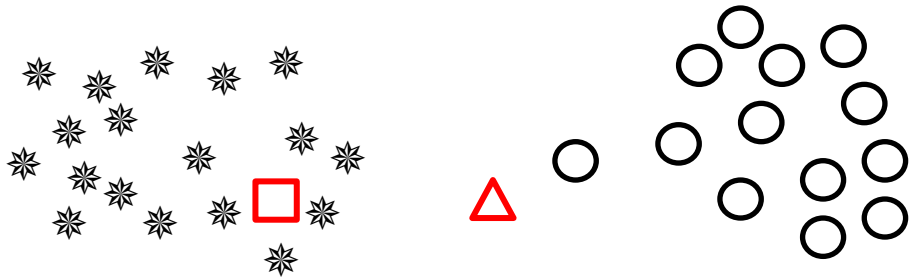
Supervised classifiers

Classification example

Binary problem: L={star, circle}
Sample: $(\mathbf{x}_i, y_i) = ((x_{i1}, x_{i2}), y_i)$
Black samples are training samples
Red ones are samples to be classified as 'star' or 'circle'

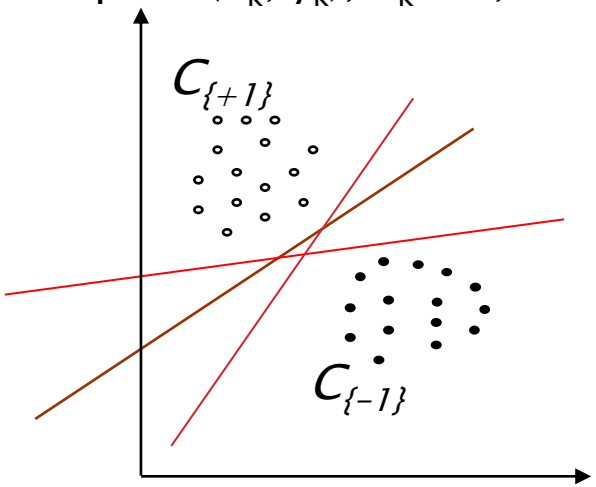
Intuitively, one establishes:

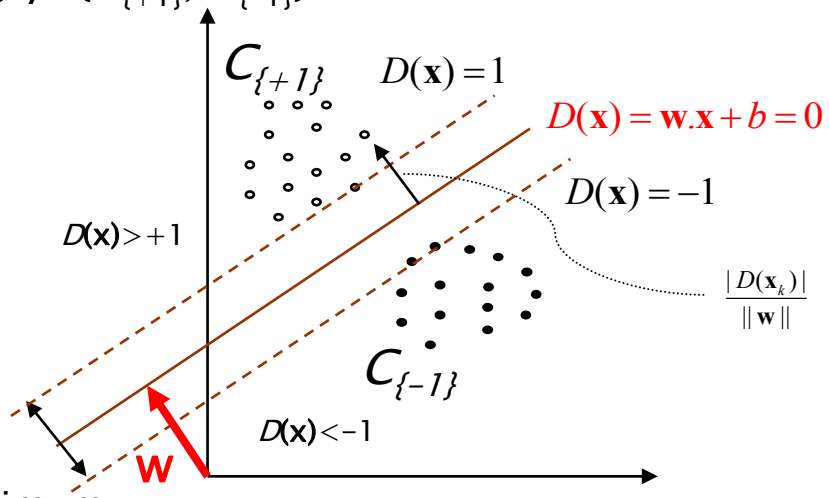
y_{square} = star
$y_{triangle}$ = circle



Support Vector Machines (SVM)

- Binary classifier
- It finds the optimal separating hyper-plane between classes
- Samples: $(\mathbf{x}_k, y_k)$, $\mathbf{x}_k \in R^n$, $k = 1, \ldots, N$, $y \in \{C_{\{+1\}}, C_{\{-1\}}\}$





Maximum margin: 2τ

$$\frac{y_k D(\mathbf{x}_k)}{\|\mathbf{w}\|} \geq \tau, \quad y_k \in \{+1, -1\}, \quad k = 1, \cdots$$

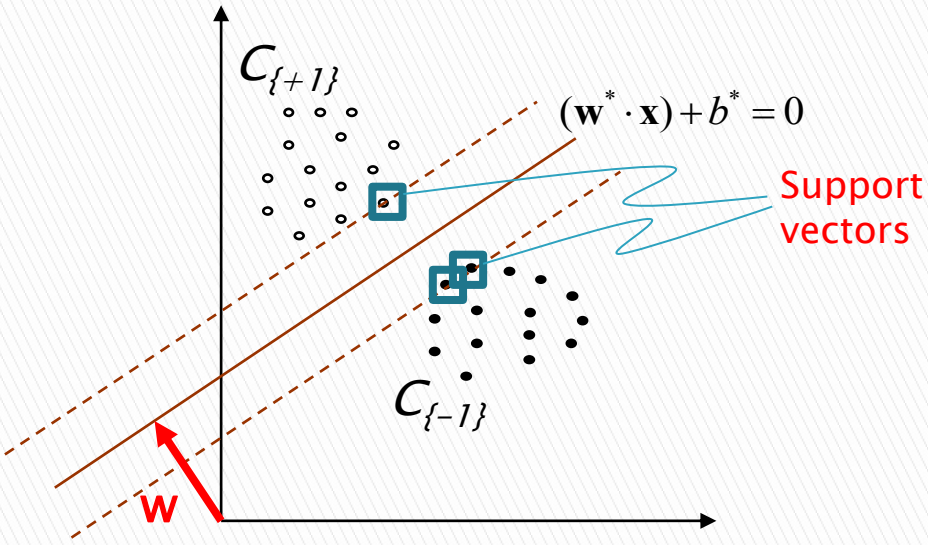
To find the optimal hyper-plane it is necessary to determine the vector **w** that maximizes the margin τ
There are infinite solutions due to the presence of a scale factor. To avoid this: $\tau \, \|\mathbf{w}\| = 1$
Therefore, to maximize the margin is equivalent to minimize $\|\mathbf{w}\|$

Optimization problem:

$$\min_{\mathbf{w}, w_0} \; J(\mathbf{w}) = \|\mathbf{w}\|^2, \quad \text{subject to} \quad y_k \left[(\mathbf{w}.\mathbf{x}_i) + w_0 \right] \geq 1$$

Support Vector Machines (SVM)

(x_k, y_k), $x_k \in R^n$, $k = 1, \ldots, N$, $y \in C_{\{+1\}}, C_{\{-1\}}\}$



- Solution: $\mathbf{w}^* = \sum_{i=1}^{N} \alpha_i^* y_i \mathbf{x}_i$

  α_i are the Lagrange multipliers
- Samples associated to $\alpha_i \neq 0$ are called "support vectors"

$$\mathbf{w} = \sum_{\text{support vectors}} \alpha_i^* y_i \mathbf{x}_i$$

The rest of training samples are irrelevant to classify new samples

- The constant b is obtained from any condition (Karush-Kuhn-Tucker)

$$\alpha_i \left[y_i \left((\mathbf{w} \cdot \mathbf{x}_i) + b \right) - 1 \right] = 0, \quad i = 1, \ldots$$

$D(\mathbf{x}) = \mathbf{w}^* \cdot \mathbf{x} + b^*$ is the distance (with sign) from **X** to the separating hyper-plane

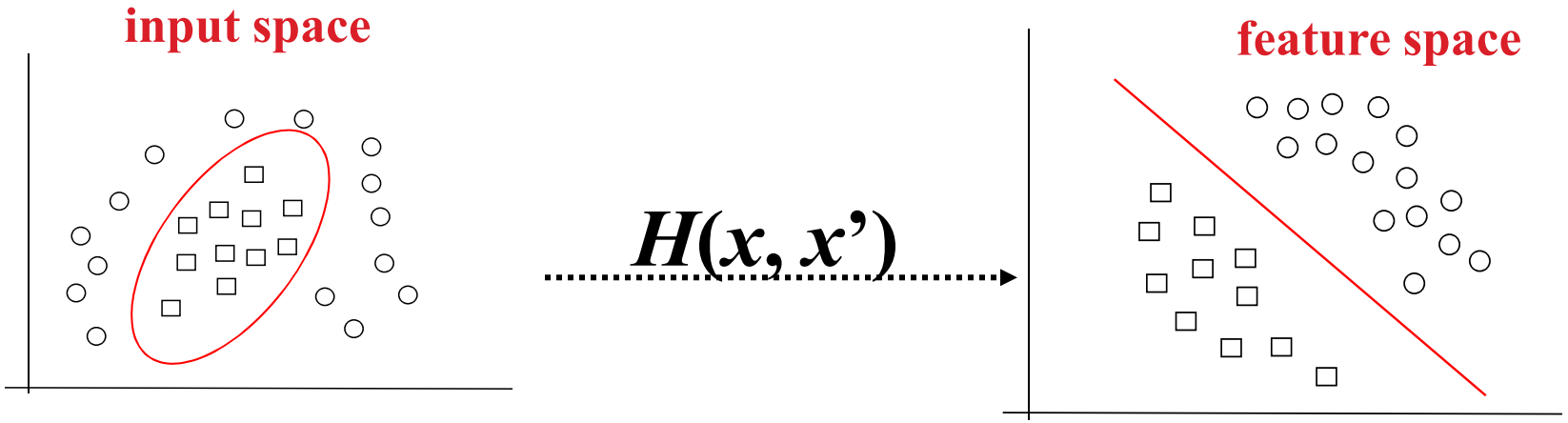
Given **x** to classify

$$\text{if } sign\left(\sum_{\text{vectores soporte}} \alpha_i^* y_i (\mathbf{x} \cdot \mathbf{x}_i) + b^* \right) \geq 0, \quad \mathbf{x} \in C_{\{+1\}}. \quad \text{Otherwise } \mathbf{x} \in C_{\{-1\}}$$

V. Cherkassky, F. Mulier. *Learning from data*. 2nd edition. Wiley-Interscience.

Support Vector Machines (SVM)

- Non-linearly separable case



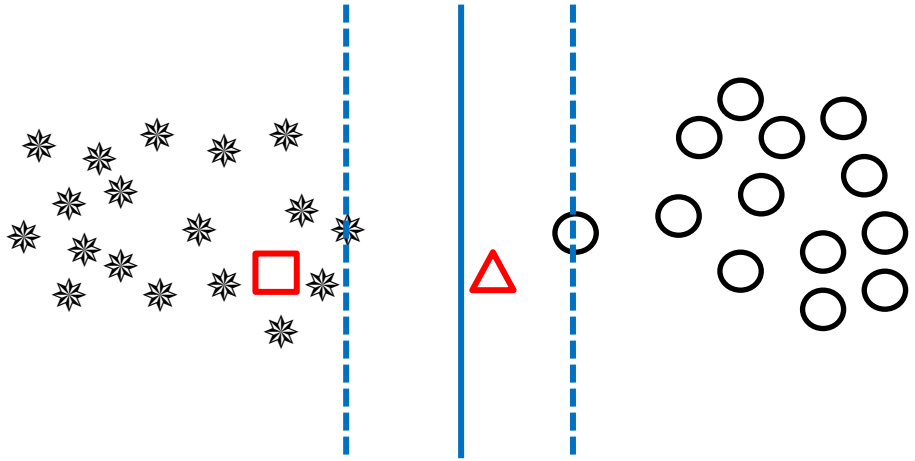
- Kernels
  - Linear: $H(\mathbf{x},\mathbf{x}')=(\mathbf{x}.\mathbf{x}')$
  - Polynomials of degree q: $H(\mathbf{x},\mathbf{x}')=[(\mathbf{x}.\mathbf{x}')+1]^q$
  - Radial basis functions: $\longrightarrow H(\mathbf{x},\mathbf{x}')=\exp\left\{-\frac{|\mathbf{x}-\mathbf{x}'|^2}{\sigma^2}\right\}$
  - Neural network: $H(\mathbf{x},\mathbf{x}')=\tanh(2(\mathbf{x}.\mathbf{x}')+1)$

Given $\mathbf{x}$ to classify

$$if\ sign\left(\sum_{vectores\ soporte}\alpha_i^* y_i H(\mathbf{x},\mathbf{x}_i)+b^*\right)\geq 0,\quad \mathbf{x}\in C_{\{+1\}}.\quad \text{Otherwise } \mathbf{x}\in C_{\{-1\}}$$

V. Cherkassky, F. Mulier. *Learning from data*. 2nd edition. Wiley-Interscience.

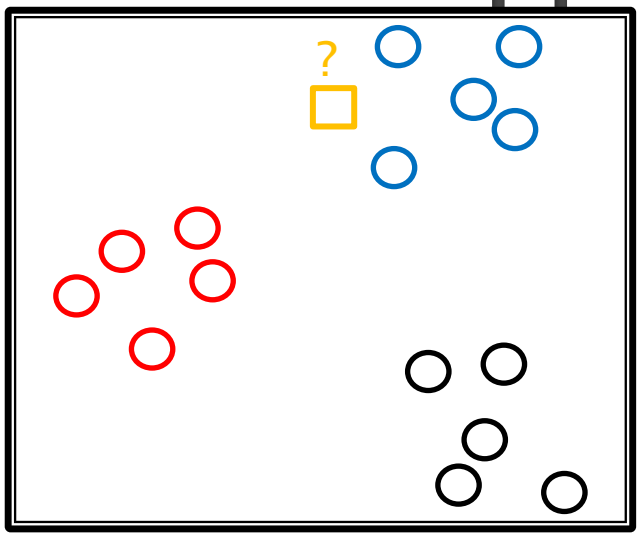
SVM example

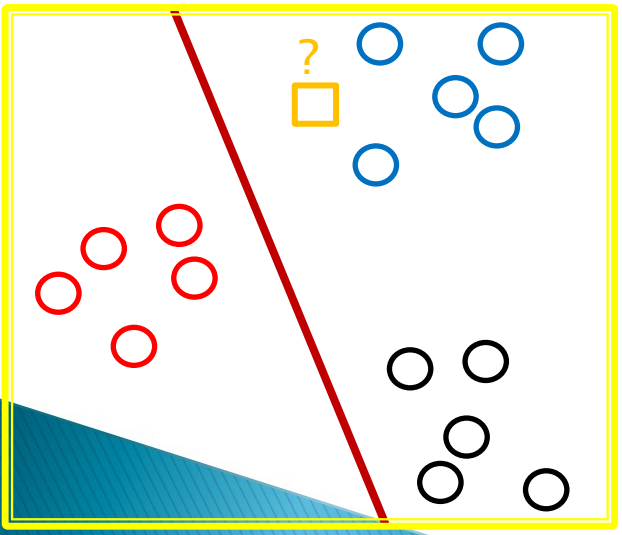


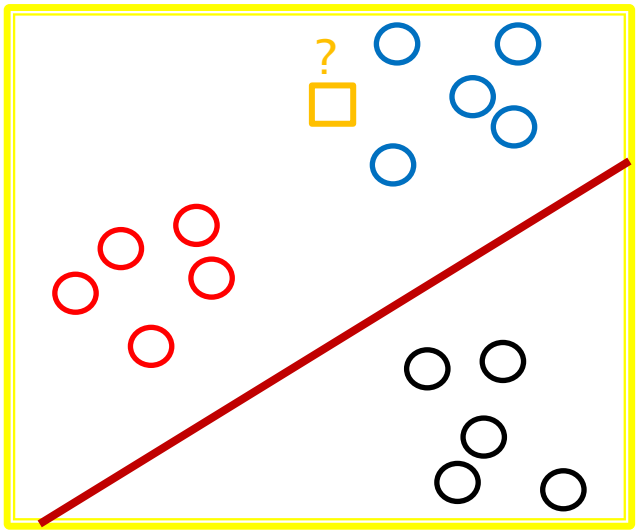
$y_{square} = star$

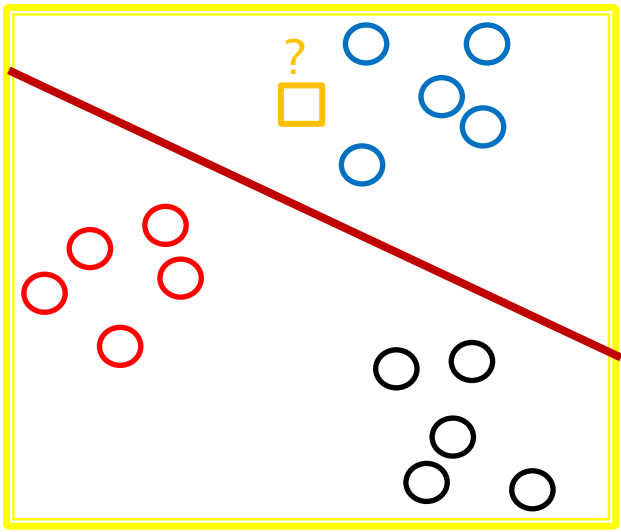
$y_{triangle} = circle$

Multiclass problem with SVM: one versus the rest approach

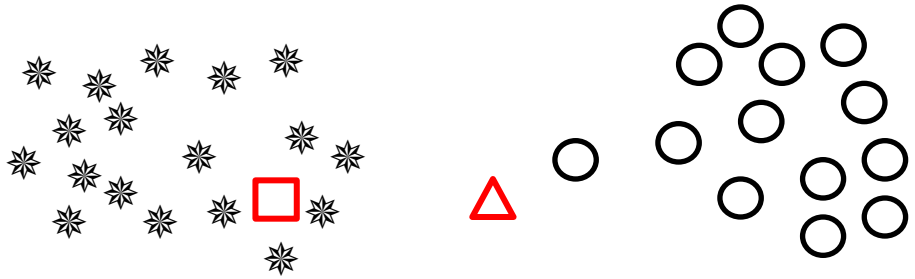


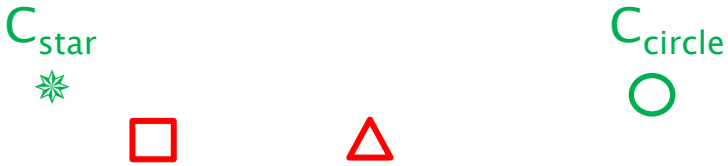






Nearest neighbour and nearest centroid





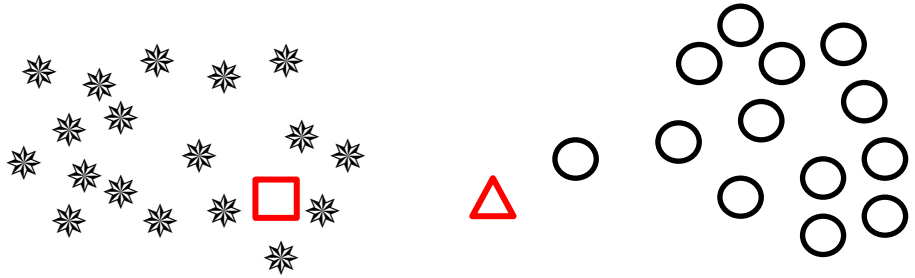
$y_{square} = star$

$y_{triangle} = circle$

How reliable are the classifiers?

- Dataset: $(\boldsymbol{x}_1, y_1)$, $(\boldsymbol{x}_2, y_2)$, ..., $(\boldsymbol{x}_N, y_N)$
  - $(\boldsymbol{x}_i, y_i)$, i = 1, ..., J: training set (model creation)
  - $(\boldsymbol{x}_i, y_i)$, i = J+1, ..., N: test set (model validation)
    - The predicted labels are compared with the real ones and the success rate that is obtained is taken as the level of confidence of the classifier.
    - This level of confidence is assumed to be the same for all future samples
  - $(\boldsymbol{x}_i, y_i)$, i = N+1, ...: predictions

- Predictions corresponding to different samples can have different levels of confidence



- *Objective*: to qualify each particular prediction with a measure of its reliability

Supervised classifiers: Bayes decision theory

Samples: $(\mathbf{x}_k, y_k)$, $\mathbf{x}_k \in R^n$, $k = 1, \ldots, N$, $y \in \{C_1, C_2\}$

Likelihood

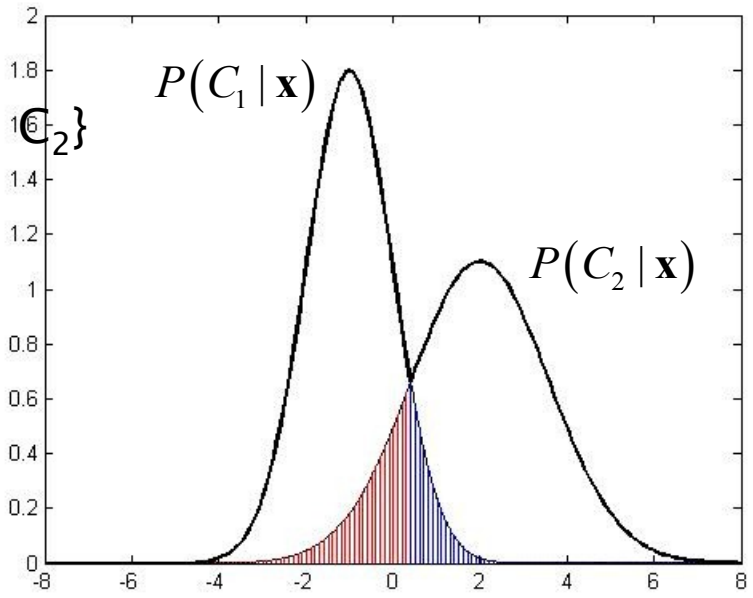
Prior

$$P(C_j \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid C_j) P(C_j)}{p(\mathbf{x})}, j \in \{L_1, L_2\}$$

Posterior

Probability distribution function of $\mathbf{x}$

$$p(\mathbf{x}) = \sum_j p(\mathbf{x} \mid C_j) P(C_j)$$



Classification rule

Given $\mathbf{x}$ to classify

$$if\ P(C_1 \mid \mathbf{x}) \geq P(C_2 \mid \mathbf{x}), \mathbf{x} \in C_1. \text{ Otherwise } \mathbf{x} \in C_2$$

$$if\ p(\mathbf{x} \mid C_1) P(C_1) \geq p(\mathbf{x} \mid C_2) P(C_2), \mathbf{x} \in C_1$$

Otherwise $\mathbf{x} \in C_2$

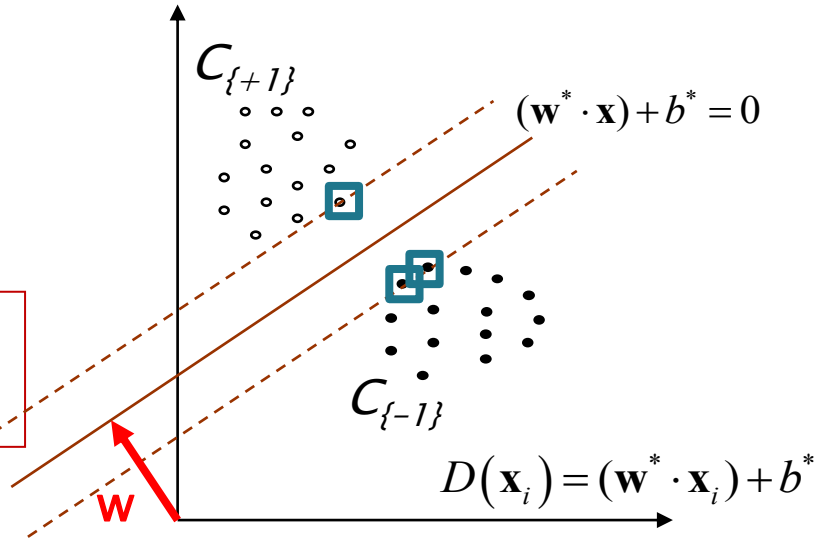
Likelihood estimation:

- Parametric: parameter estimation
- Non-parametric: Parzen window

R. O. Duda, P. E. Hart, D. G. Stork. *Pattern classification*. 2nd edition. Wiley-Interscience. (2001)

Platt's method

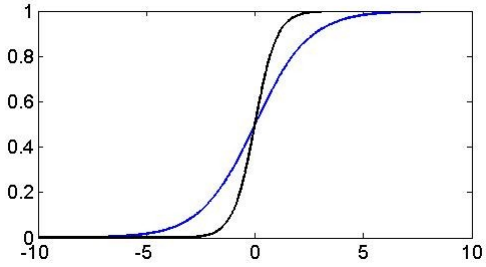
The greater the distance *D(x)* the deeper is the point in its corresponding class. This has a translation in terms of probability



- It trains SVM on the training set and then estimates probabilities based on the decision function *D(x)* by fitting a sigmoid:

$$P\left(y=1 \mid D\left(\mathbf{x}_i\right)\right) = \frac{1}{1+\exp\left(A \cdot D\left(\mathbf{x}_i\right) + B\right)}$$

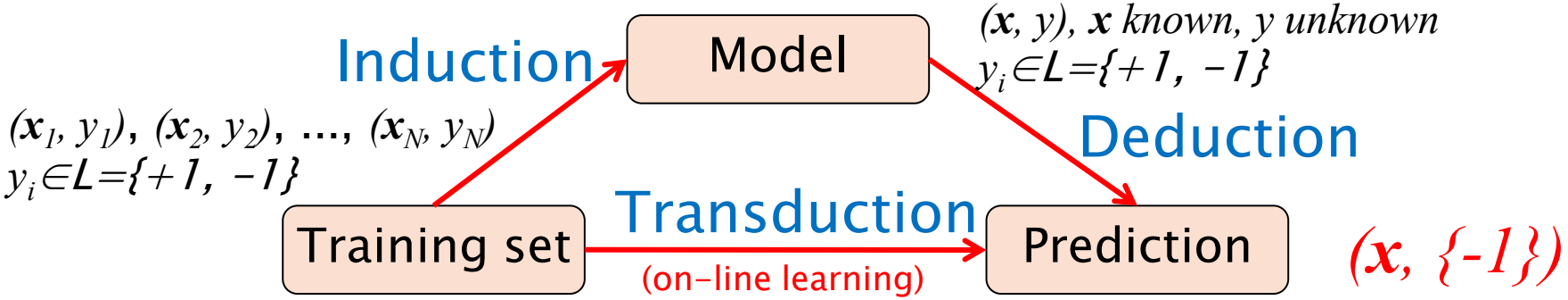
where A and B parameters are determined by using maximum likelihood estimate from the training set



$$P_{class\{+1\}} = \frac{1}{1+\exp\left[-kD\left(\mathbf{x}\right)\right]}$$

$$P_{class\{-1\}} = 1 - P_{class\{+1\}}$$

Conformal predictors



- There is not a rigid separation between learning and prediction
- One makes predictions sequentially, basing each new prediction on all the previous examples instead of repeatedly using a rule constructed from a fixed batch of examples
- Each individual prediction is qualified

$$(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N), (x, \{+1\}) \Rightarrow p_{\{+1\}}$$

$$(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N), (x, \{-1\}) \Rightarrow p_{\{-1\}}$$

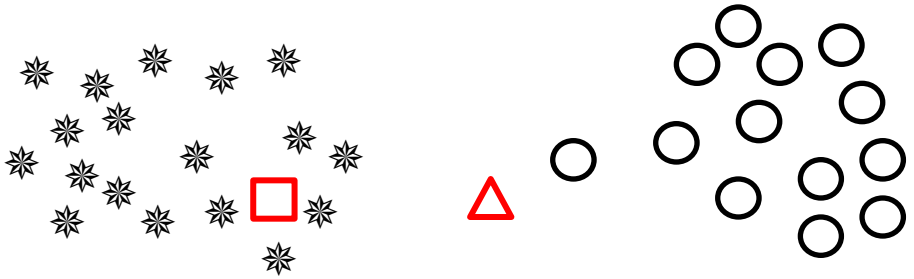
$$\Rightarrow \begin{cases} \text{Prediction: label of the largest p-value} \\ \textit{Credibility}: \max\left(p_{\{+1\}}, p_{\{-1\}}\right) \\ \textit{Confidence}: 1-2^{nd} \text{ pvalue} \end{cases}$$

- Non-conformity measure: $\alpha_i = A_n\left(\prec \quad \succ \quad , y\right)$

$$p-value = \frac{\left|\left\{j=1,\ldots,n : \alpha_j \geq \alpha_i\right\}\right|}{n}$$

Conformal prediction

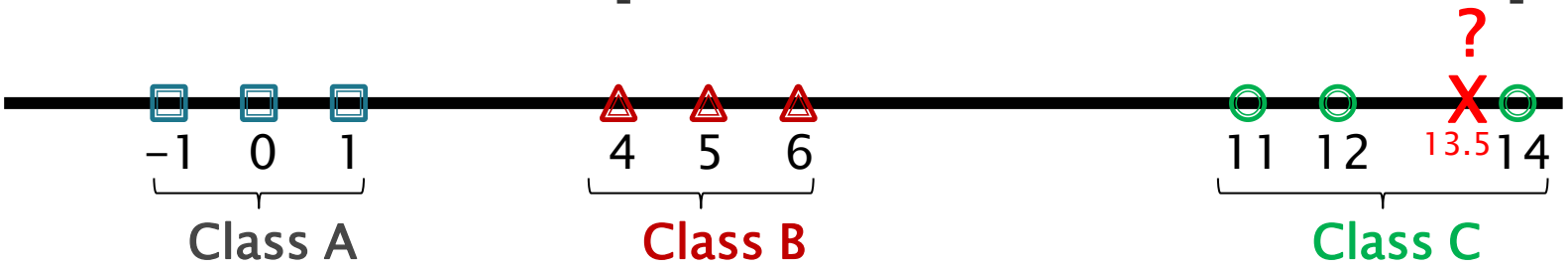
y_{square} = star
$y_{triangle}$ = circle



- Square
  - High credibility
  - Very high confidence

- Triangle
  - Low credibility
  - High confidence

Conformal prediction: example



Underlying classifier: 1-nearest neighbour

Non-conformity scores

$$\alpha_i = \frac{\min\left\{\left|\mathbf{x}_j - \mathbf{x}_i\right| : 1 \le j \le n \;\&\; j \ne i \;\&\; y_j = y_i\right\}}{\min\left\{\left|\mathbf{x}_j - \mathbf{x}_i\right| : 1 \le j \le n \;\&\; j \ne i \;\&\; y_j \ne y_i\right\}}$$

$$= \frac{\text{distance to z's nearest neighbor with the same label}}{\text{distance to z's nearest neighbor with a different label}}$$

$$p - value = \frac{\left|\left\{j = 1,\ldots,n : \alpha_j \ge \alpha_i\right\}\right|}{n}$$

Coordinate of object i	α_i if y_n=A	α_i if y_n=B	α_i if y_n=C
-1	1/5=0.2	1/5=0.2	1/5=0.2
0	1/4=0.25	1/4=0.25	1/4=0.25
1	1/3=0.33	1/3=0.333	1/3=0.333
4	1/3=0.33	1/3=0.333	1/3=0.333
5	1/4=0.25	1/4=0.25	1/4=0.25
6	1/5=0.2	1/5=0.2	1/5=0.2
11	1/2.5=0.4	1/2.5=0.4	1/5=0.2
12	1/1.5=0.667	1/1.5=0.667	1/6=0.167
14	2/0.5=4	2/0.5=4	0.5/8=0.063
13.5	12.5/0.5=25	-	-
13.5	-	7.5/0.5=15	-
13.5	-	-	0.5/7.5=0.067
p-value:	1/10=0.10	1/10=0.10	9/10=0.9

Label: C
Confidence: 1 – 0.1 = 0.9
Credibility: 0.9

Confidence machines

- Conformal predictors are "confidence predictors"
  - This notion of confidence should not be confused with confidence in statistical conclusions with confidence intervals
- Confidence machines are region predictors
  - In the event that a unique prediction cannot be achieved with required confidence, the method outputs a set (region) of possible labels
- The main advantage of confidence machines is their property of validity
  - The rate of erroneous region predictions does not asymptotically exceed the preset value ε, called significance level
- Confidence machines can also be forced to output singleton predictions
  - Forced accuracy

Training set: $(\boldsymbol{x}_1, y_1)$, $(\boldsymbol{x}_2, y_2)$, …, $(\boldsymbol{x}_{n-1}, y_{n-1})$, $\boldsymbol{x}_i \in \boldsymbol{X}$ (object set) *and* $y_i \in Y = \{L_1, L_2, \ldots, L_K\}$ (label set)

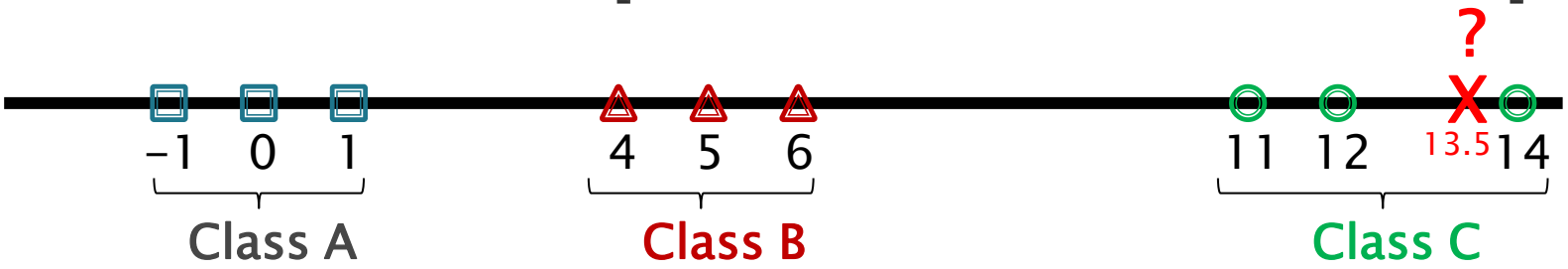
Sample to classify: $(\boldsymbol{x}_n, y_n)$, $y_i \in Y$ (unknown)

$\Gamma^{\varepsilon}(z_1, \ldots, z_{n-1}, x_n) \subset Y$ is the prediction region

Include label L_j in $\Gamma^{\varepsilon}(z_1, \ldots, z_{n-1}, x_n) \Leftrightarrow p_y > \varepsilon$, $p_y = \dfrac{\#\{i = 1, \ldots, n \mid \alpha_i \geq \alpha_n\}}{n}$

The inclusion is valid at the 1 – ε level

Conformal prediction: example



Coordinate of object i	α_i if y_n=A	α_i if y_n=B	α_i if y_n=C
-1	1/5=0.2	1/5=0.2	1/5=0.2
0	1/4=0.25	1/4=0.25	1/4=0.25
1	1/3=0.33	1/3=0.333	1/3=0.333
4	1/3=0.33	1/3=0.333	1/3=0.333
5	1/4=0.25	1/4=0.25	1/4=0.25
6	1/5=0.2	1/5=0.2	1/5=0.2
11	1/2.5=0.4	1/2.5=0.4	1/5=0.2
12	1/1.5=0.667	1/1.5=0.667	1/6=0.167
14	2/0.5=4	2/0.5=4	0.5/8=0.063
13.5	12.5/0.5=25	-	-
13.5	-	7.5/0.5=15	-
13.5	-	-	0.5/7.5=0.067
p-value:	1/10=0.10	1/10=0.10	9/10=0.9

Confidence level 90% $\Rightarrow 1-\varepsilon=0.9 \Rightarrow \varepsilon=0.1$

$$\Gamma^{0.1}\left(z_1,\ldots,z_{n-1},x_n\right) \Leftrightarrow p_y > 0.1$$

$$\Gamma^{0.1}\left(z_1,\ldots,z_{n-1},x_n\right) = \{C\} \quad \text{(single prediction)}$$

Confidence level 95% $\Rightarrow 1-\varepsilon=0.95 \Rightarrow \varepsilon=0.05$

$$\Gamma^{0.05}\left(z_1,\ldots,z_{n-1},x_n\right) \Leftrightarrow p_y > 0.05$$

$$\Gamma^{0.05}\left(z_1,\ldots,z_{n-1},x_n\right) = \{A,B,C\} \quad \text{(multi-prediction)}$$

Confidence level 10% $\Rightarrow 1-\varepsilon=0.1 \Rightarrow \varepsilon=0.9$

$$\Gamma^{0.9}\left(z_1,\ldots,z_{n-1},x_n\right) \Leftrightarrow p_y > 0.9$$

$$\Gamma^{0.9}\left(z_1,\ldots,z_{n-1},x_n\right) = \varnothing \quad \text{(empty set)}$$

Venn predictors

- Confidence machines output a set of possible labels for a new object
  - In different applications, it can be more useful to predict the probability of a label
- Can we avoid the use of statistical assumptions about the sample distribution?
  - If the assumed statistical model is not correct, predicted probabilities may be incorrect too
- A Venn machine outputs several probability distributions, one for each candidate label
  - Multiprobability prediction
  - This output can be also interpreted in a different way: a prediction with the assigned interval of probability that this prediction is correct
- Similarly to confidence machines, Venn machines are valid regardless of the sample distribution
  - The only assumption made is iid
  - Different underlying algorithms can be applied

Venn predictors: main idea

- First, a taxonomy (NN, NC, SVM, ...) is defined to divide the samples into a number of categories (can be different to the number of labels)
- Then, we try all possible labels of the new object
  - In each attempt, we can calculate the frequencies of the labels in the category which the new object falls into
  - The data processing is summarized is a square matrix (with dimension the number of labels). We obtain as many probability distributions as the number of possible labels

Quality: min

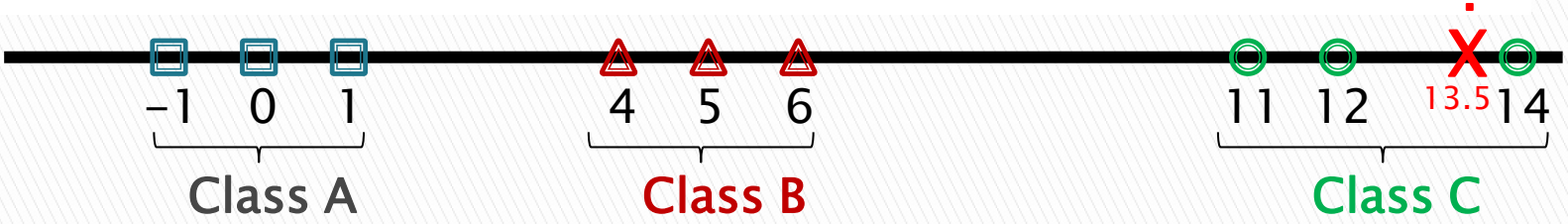
$$P_C = \begin{pmatrix} p^{Y_1}(Y_1) & p^{Y_1}(Y_2) & p^{Y_1}(Y_C) \\ p^{Y_2}(Y_1) & p^{Y_2}(Y_2) & p^{Y_2}(Y_C) \\ p^{Y_C}(Y_1) & p^{Y_C}(Y_2) & p^{Y_C}(Y_C) \end{pmatrix}$$

Venn prediction

• Label: the one corresponding to the max. quality

• Probability interval: [min freq in column, max freq in column]

Venn predictors: examples



$$P_C = \begin{array}{c} \begin{matrix} A & B & C \end{matrix} \\ \begin{pmatrix} 1/3 & 0 & 2/3 \\ 0 & 1/3 & 2/3 \\ 0 & 0 & 1 \end{pmatrix} \end{array}$$

Quality: 0 0 2/3

Prediction: label C

Probability interval: [2/3, 1]

Nearest neighbour taxonomy

$$P_C = \begin{array}{c} \begin{matrix} A & B & C \end{matrix} \\ \begin{pmatrix} 1/4 & 0 & 3/4 \\ 0 & 1/4 & 3/4 \\ 0 & 0 & 1 \end{pmatrix} \end{array}$$

Quality: 0 0 3/4

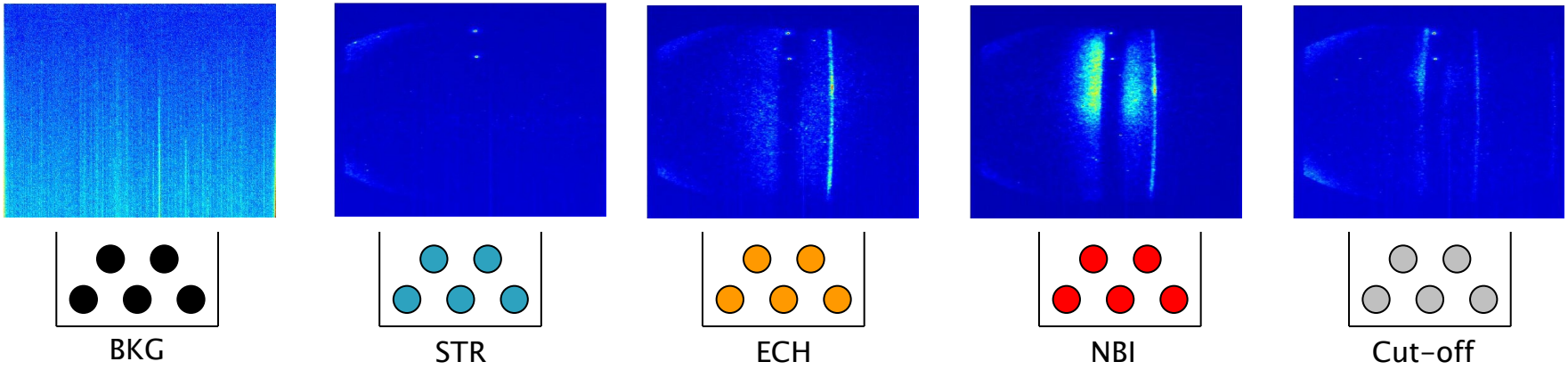
Prediction: label C

Probability interval: [3/4, 1]

Nearest centroid taxonomy

Applications using confident machines

- The TJ-II Thomson scattering case (I)



For each image $z_i=(\mathbf{x}_i, y_i)$ {

$\mathbf{x}_i$: Haar wavelet transform

y_i: Label for the class

For each new image $z=(\mathbf{x}, y)$, the label y has to be determined

- 1-nearest-neighbour method. The nonconformity scores are

$$\alpha_i = \frac{\min\left\{\left|x_j - x_i\right| : 1 \le j \le n \,\&\, j \ne i \,\&\, y_j = y_i\right\}}{\min\left\{\left|x_j - x_i\right| : 1 \le j \le n \,\&\, j \ne i \,\&\, y_j \ne y_i\right\}} = \frac{\text{distance to z's nearest neighbour with the same label}}{\text{distance to z's nearest neighbour with a different label}}$$

Applications using confident machines

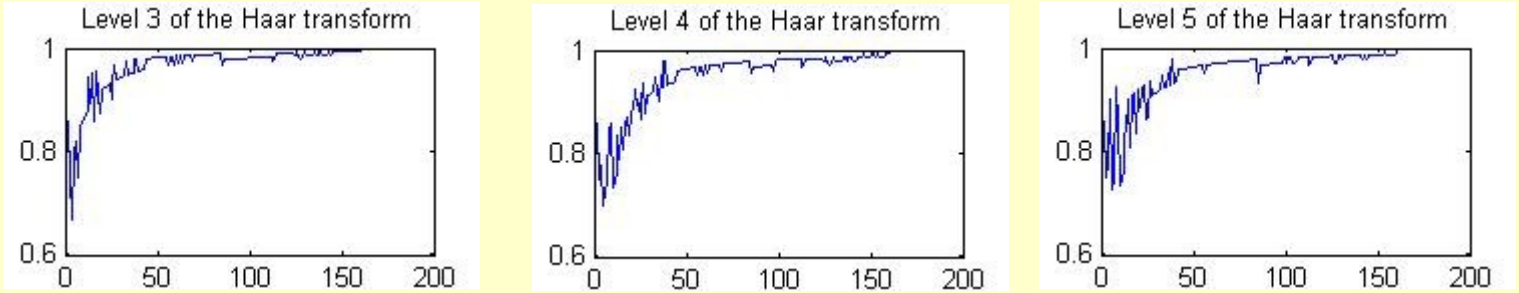
- The TJ-II Thomson scattering case

Haar decomposition level	Success rate	Confidence	Credibility	2nd p-value
3 (72x48, 1.59%)	Success: 157 (**97.52%**) Fail: 2 (1.24%) Multipredictions: 2 (1.24%)	0.963555	0.605319	0.036445
4 (36x24, 0,39%)	Success: 157 (**97.52%**) Fail: 1 (0.62%) Multipredictions: 3 (1.86%)	0.946918	0.610034	0.053082
5 (18x12, 0.10%)	Success: 153 (**95.03%**) Fail: 5 (3.11%) Multipredictions: 3 (1.86%)	0.950441	0.599851	0.049559

Original size: 576x385

Only 1 image per class in the initial training set

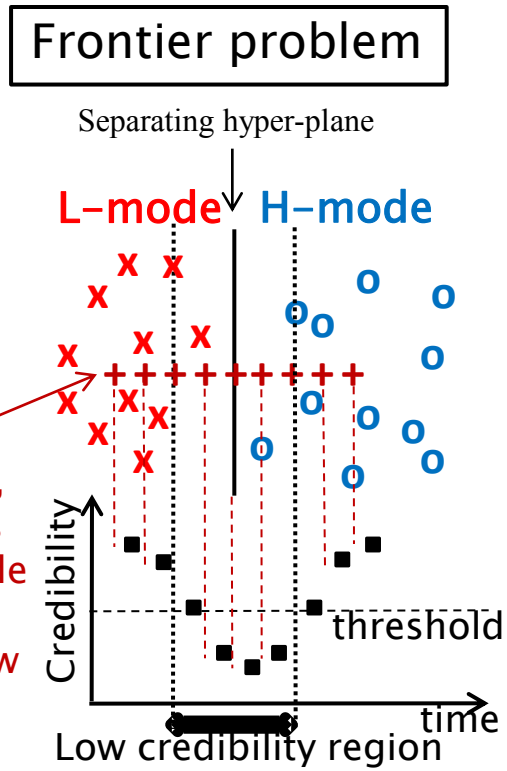
Confidence vs number of image



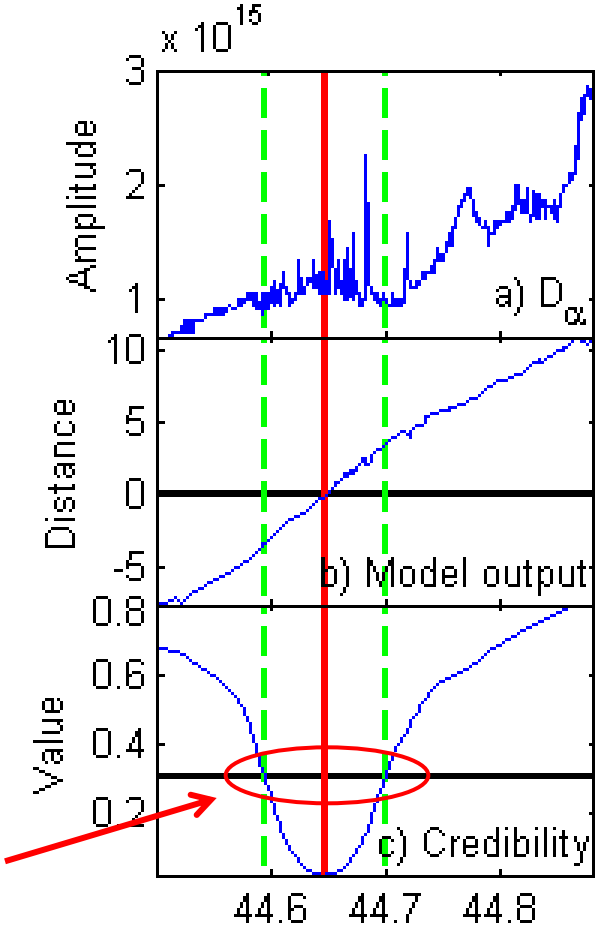
The system learns with each new image

Applications using confident machines

- Automatic prediction of JET L/H & H/L transition times



As time increases, the plasma moves from L- to H-mode and the feature vectors cross a low credibility region

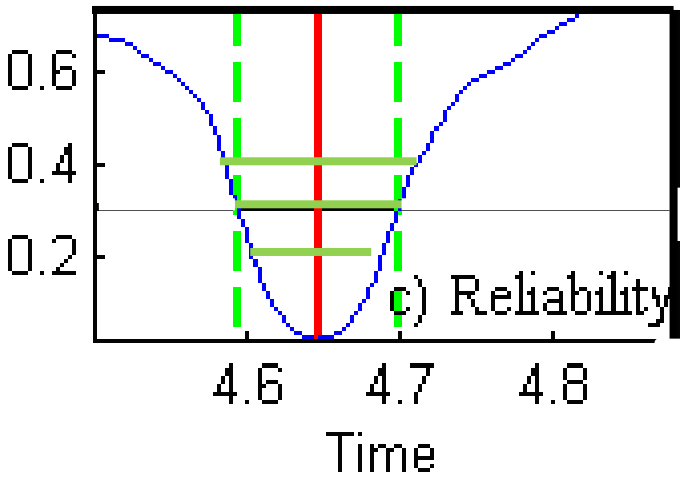


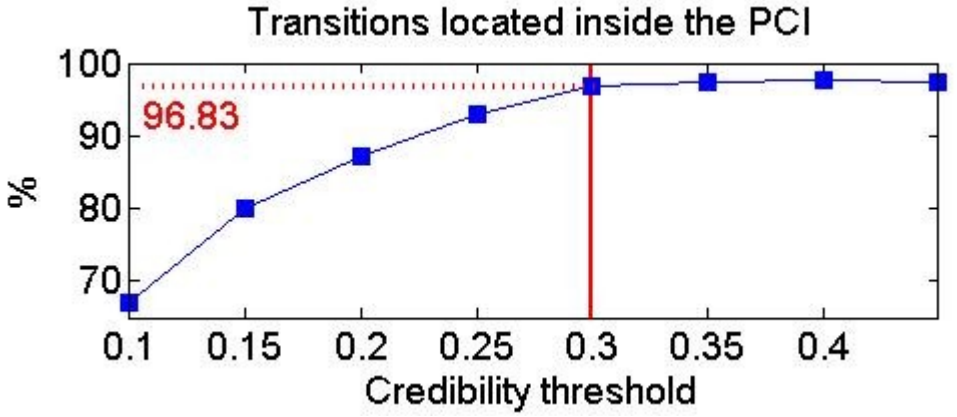
Probability confidence interval (PCI):
temporal width that delimits the pass through the low credibility region

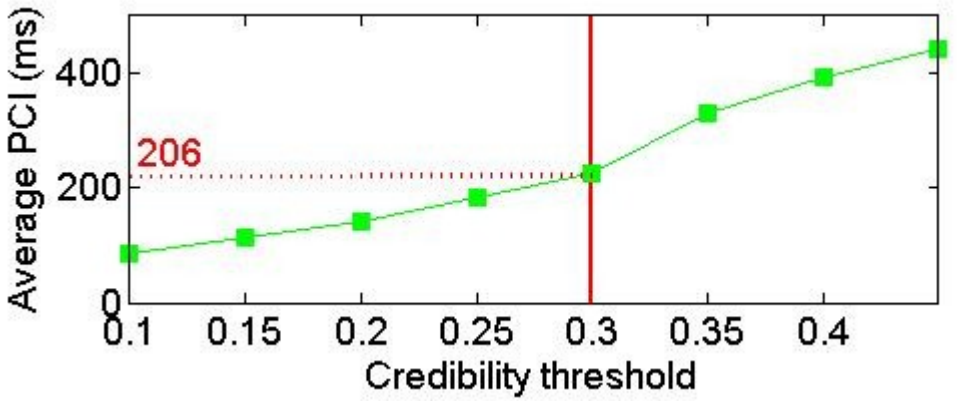
Applications using confident machines

- Automatic prediction of JET L/H & H/L transition times

551 discharges: 141 training + 410 test





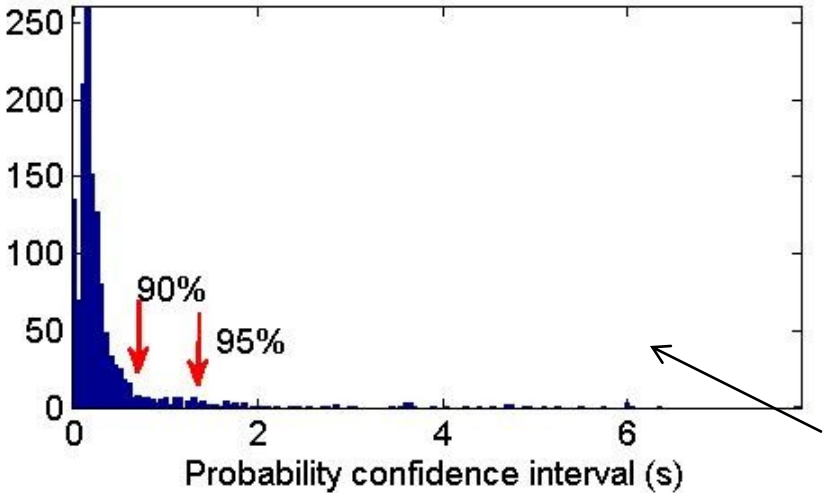


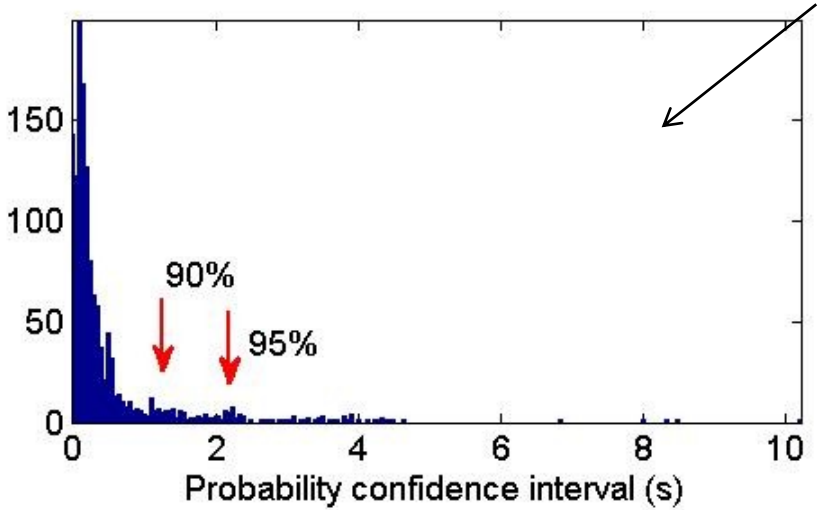
Different thresholds generate different average widths of the PCI

Applications using confident machines

- Automatic prediction of JET L/H & H/L transition times

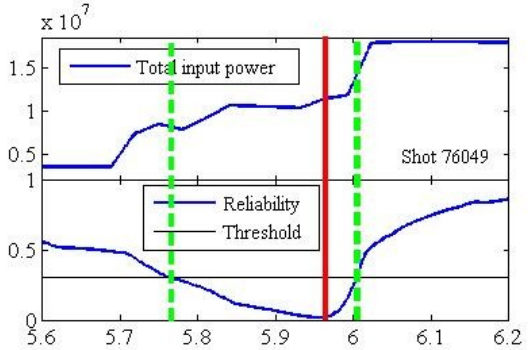
Conformal classifier applied to 1451 discharges to determine transition times and the PCI

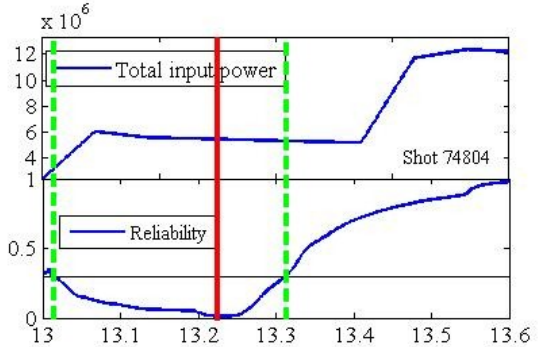


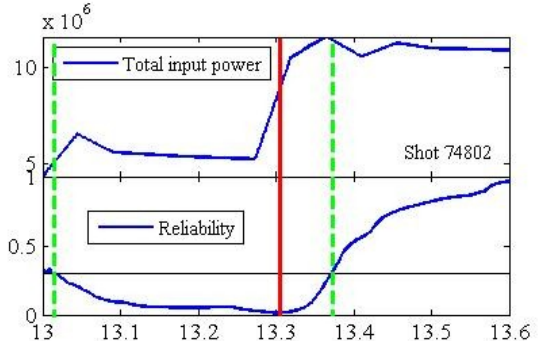


	90%	95%
L/H	191 ms	233 ms
H/L	245 ms	320 ms

Interpretation?





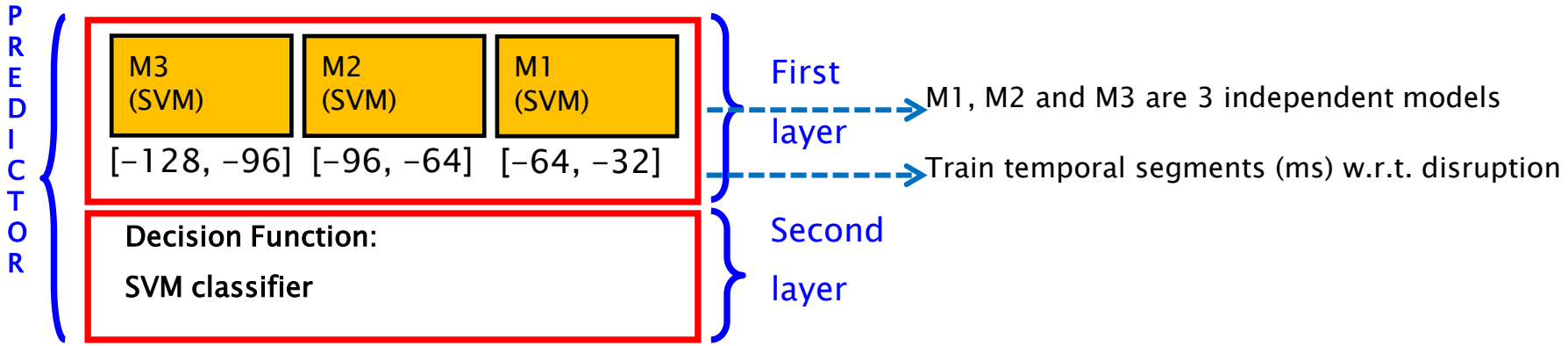


Long intervals of low credibility regions are associated to discharges in which the input power is not introduced in an abrupt way

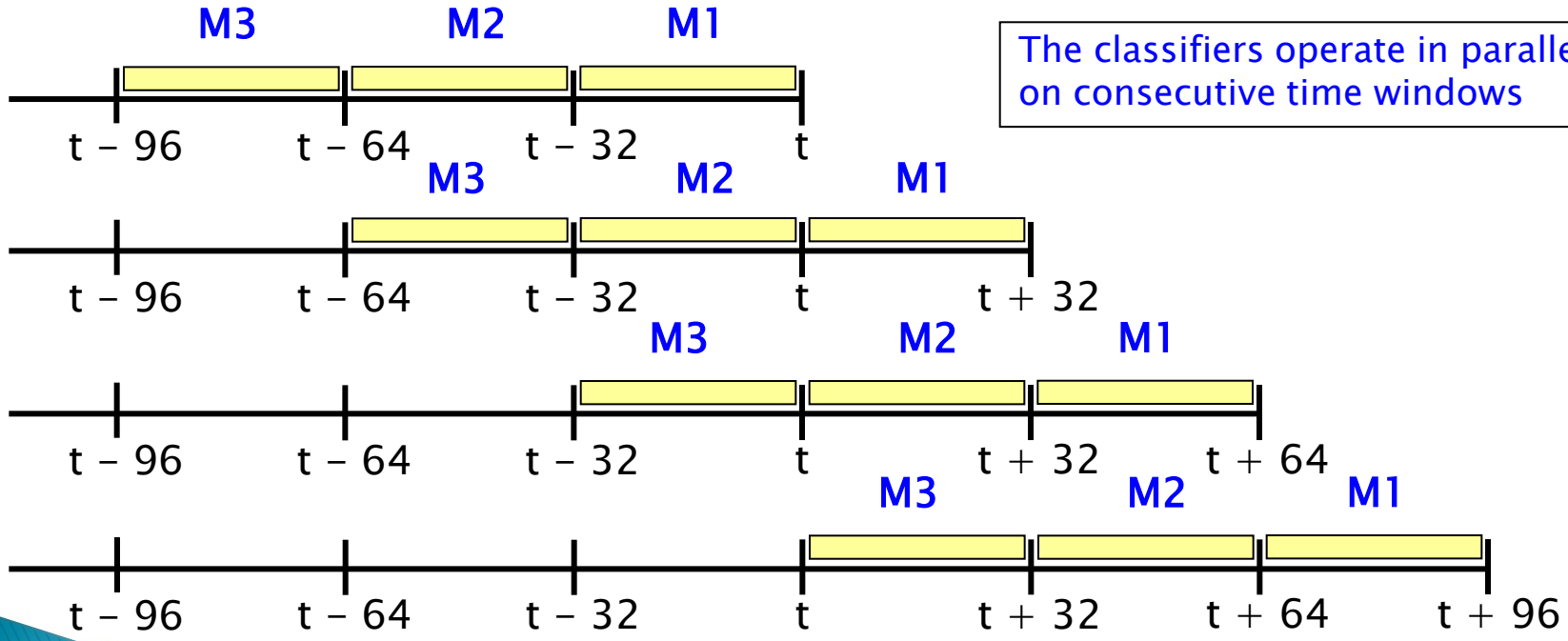
Applications using confident machines

- Disruption predictor in JET: APODIS predictor
- What is APODIS and how is related to other disruption predictors?
  - APODIS is a real-time disruption predictor that follows the temporal evolution from plasma breakdown to extinction
  - APODIS *is not* based on an *individual signal*
    - On JET, protection against locked mode disruptions exists and uses the mode amplitude as a threshold
  - APODIS *is* based on a *multi-signal scheme*
    - Previous predictors based on Artificial Neural Networks have not been reliable enough (high degradation in successive campaigns)
  - APODIS also uses signal information in the *frequency domain*
    - Better knowledge of the parameter space

Disruption predictor in JET: APODIS predictor



- As a discharge is in execution, *the most recent 32 ms temporal segments* are classified as disruptive or non-disruptive



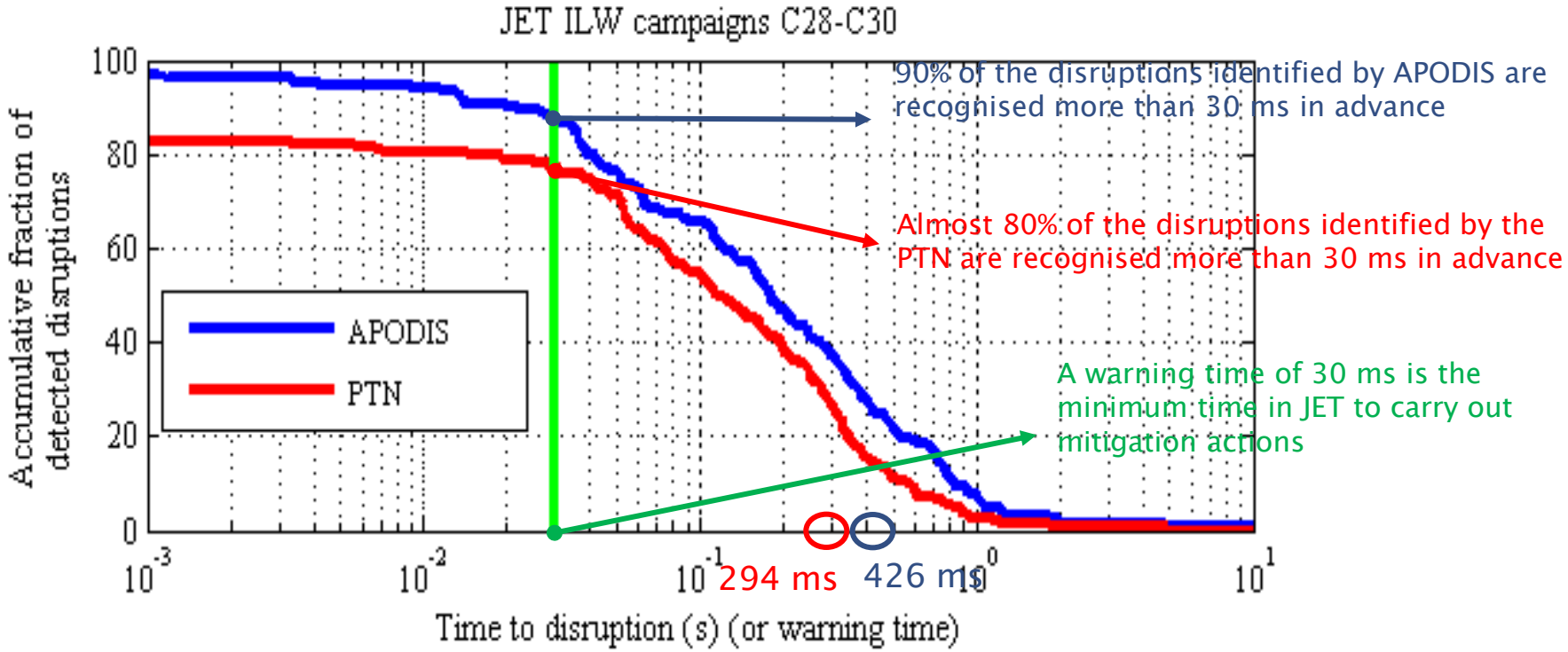
The classifiers operate in parallel on consecutive time windows

- The three models may disagree about the discharge behaviour ⇨ 2nd layer

Disruption predictor in JET: APODIS predictor

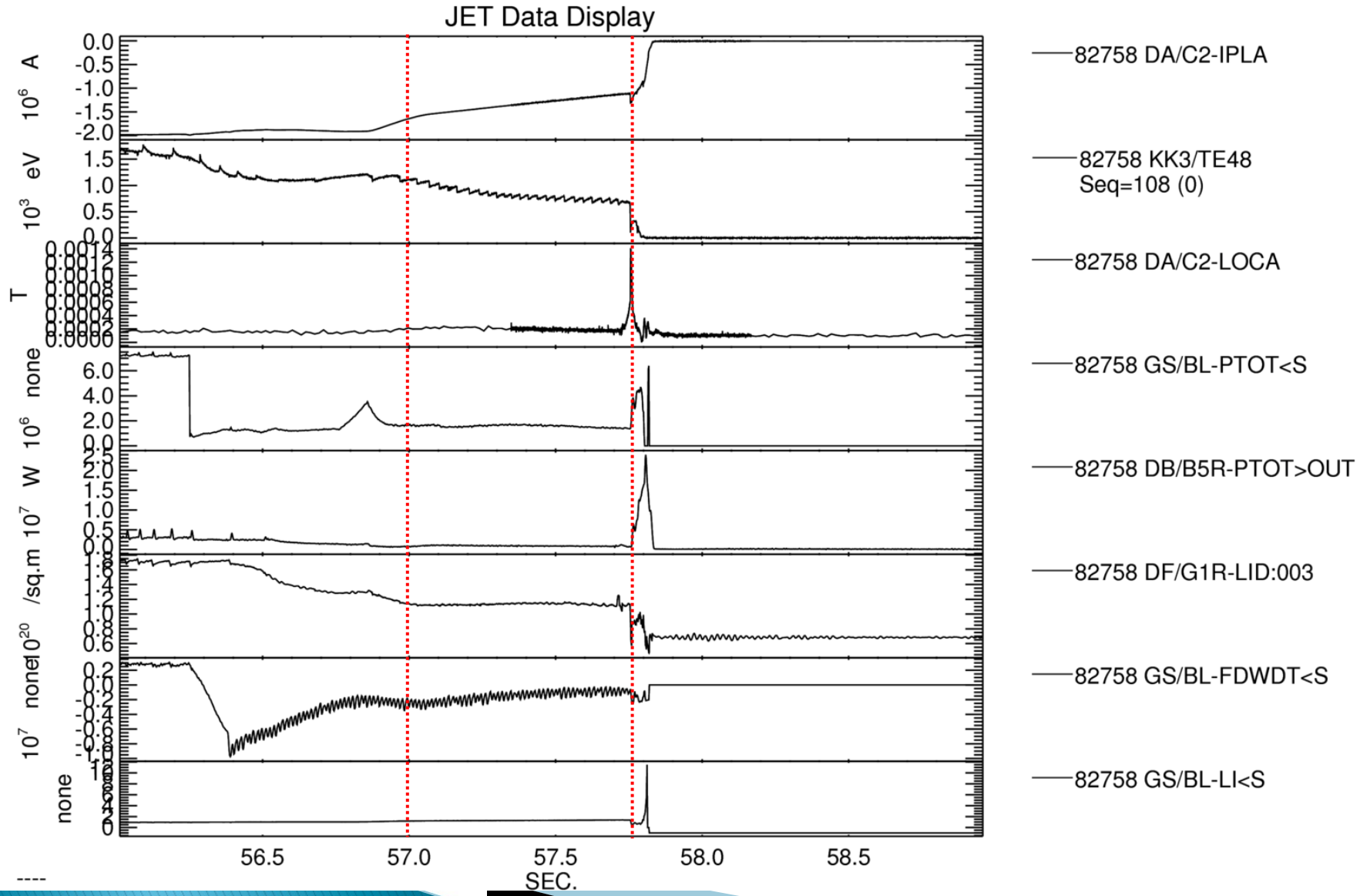
- The APODIS version installed to operate in the ILW campaigns (C28-C30) has been trained with CFC wall data (C19-C22) and ***no retraining*** has been performed

	Safe	False alarms	Unintentional disruptions	Missed alarms	TOTAL	Intentional disruptions
JET off-line classification	651	n/a	305	n/a	956	35
APODIS prediction	645 (99.08%)	6 (0.92%)	300 (98.36%)	5 (1.64%)	956	n/a



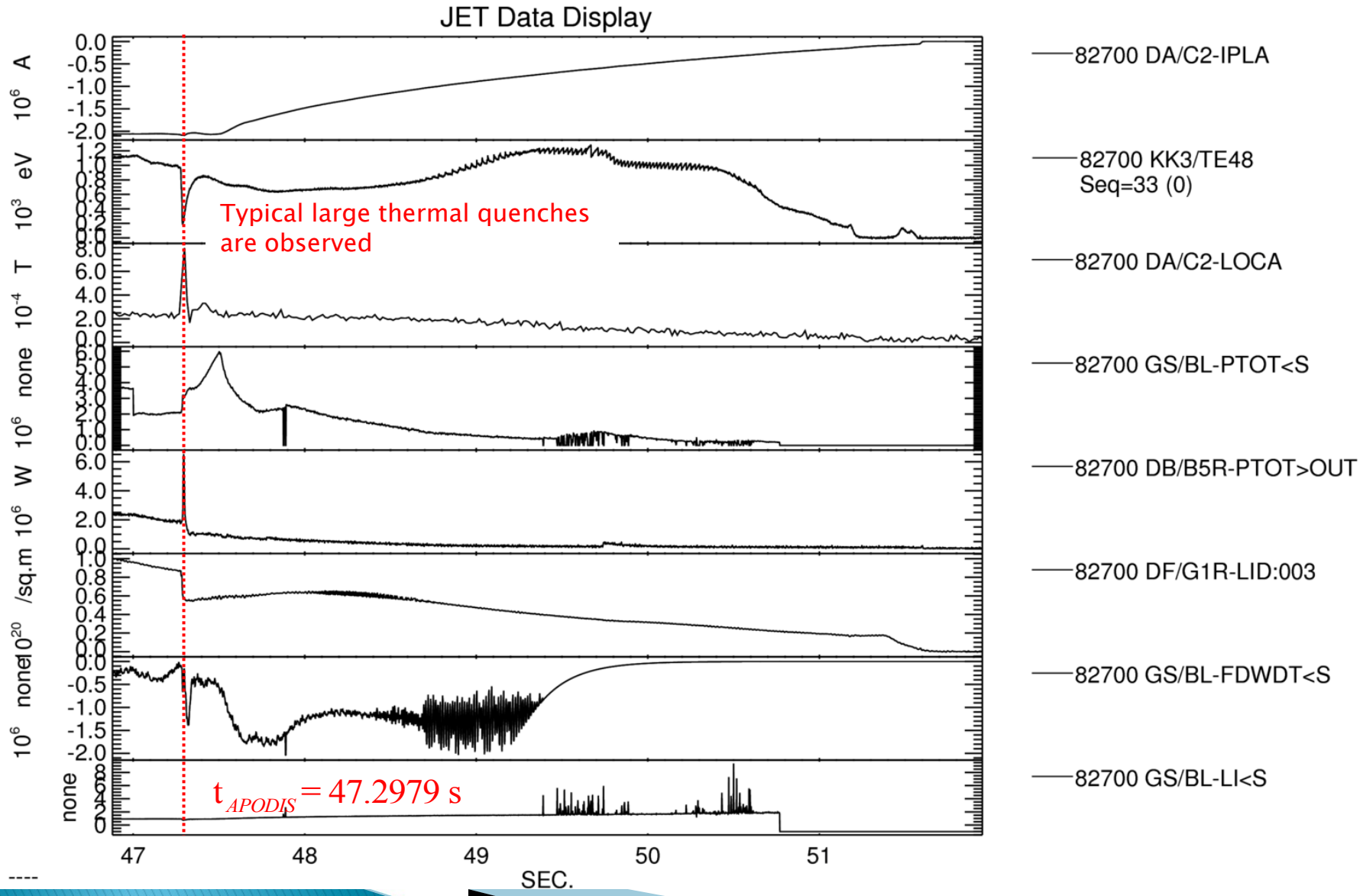
- APODIS outperforms the results of the Pulse Termination Network (PTN)
  - APODIS is faster and recognises more disruptions

Disruption predictor in JET: APODIS predictor



Disruption predictor in JET: APODIS predictor

Shot 82700. An off-line analysis has recognised all false alarms as minor disruptions

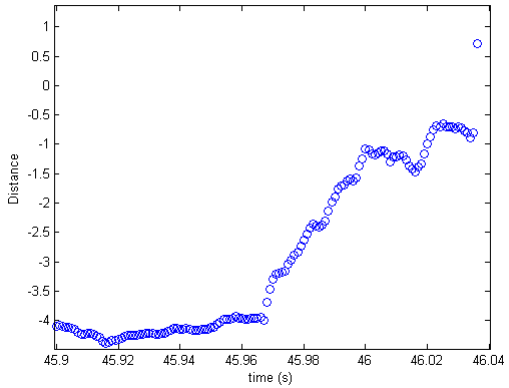


Disruption predictor in JET: APODIS predictor

- Probabilistic predictors
  - Instead of bare predictions, a probability can qualify each prediction (Bayesian classifiers, Platt's method, binning techniques, Venn predictors)
  - Could we filter false alarms?
  - Could we learn something about these plasmas?

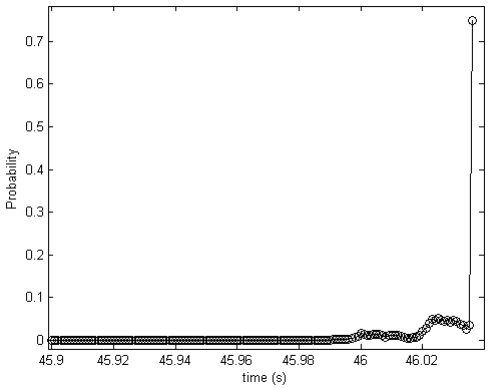
Applications using confident machines

APODIS RT

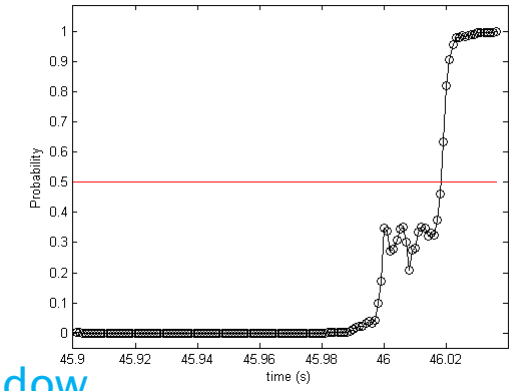


JET shot 83564

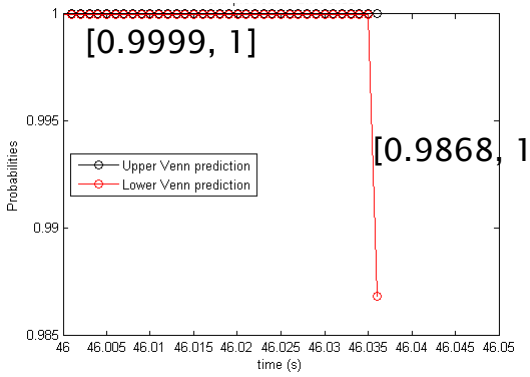
Platt's method



Bayesian classifier



Venn predictor (NC taxonomy)



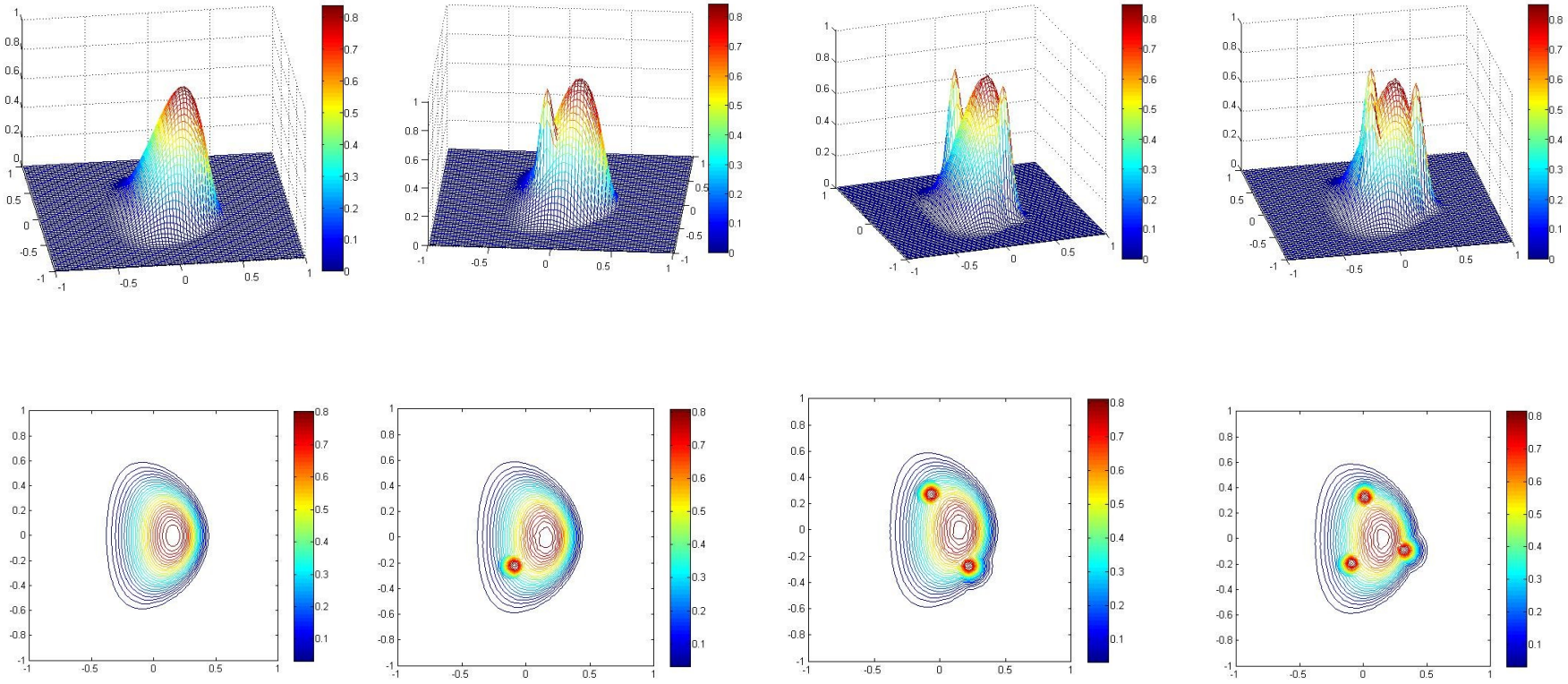
Parzen window

Prior

$$P(C_j \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid C_j) P(C_j)}{p(\mathbf{x})}, j \in \{D, ND\}$$

Applications using confident machines

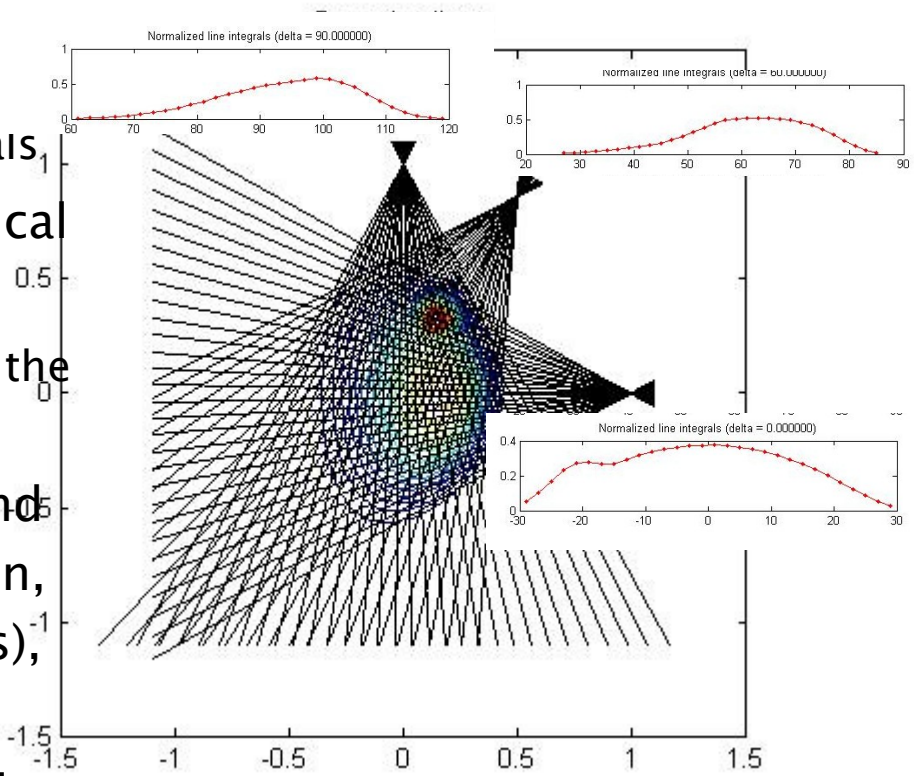
- Alternative methods to tomography in plasma physics
  - Soft X-rays, bolometry ,neutron or gamma emissions



Applications using confident machines

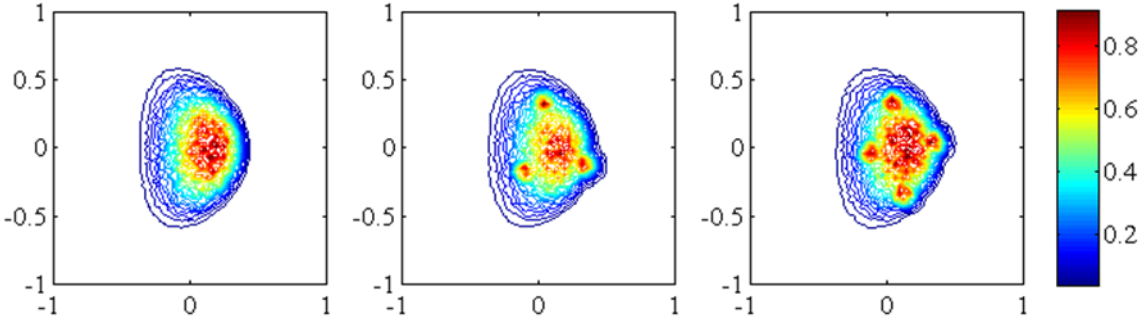
- Alternative methods to tomography in plasma physics

- Plasma can be "*seen*" through a very limited number of projections
  - Projections are made up of line integrals
- Arrays with fan-like geometry are typical in fusion
  - Detectors occupy a reduced space but the line integrals cover all the plasma
- Depending on the problem to solve and inherent constraints (initial distribution, spatial resolution, ill-posed problems), tomography can be unfeasible
- Can we determine the number of local perturbations at a given time instant?

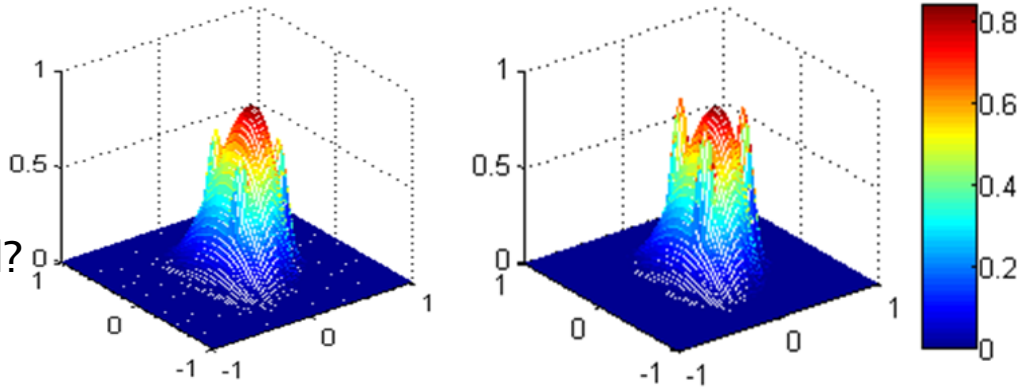


Applications using confident machines

- Alternative methods to tomography in plasma physics



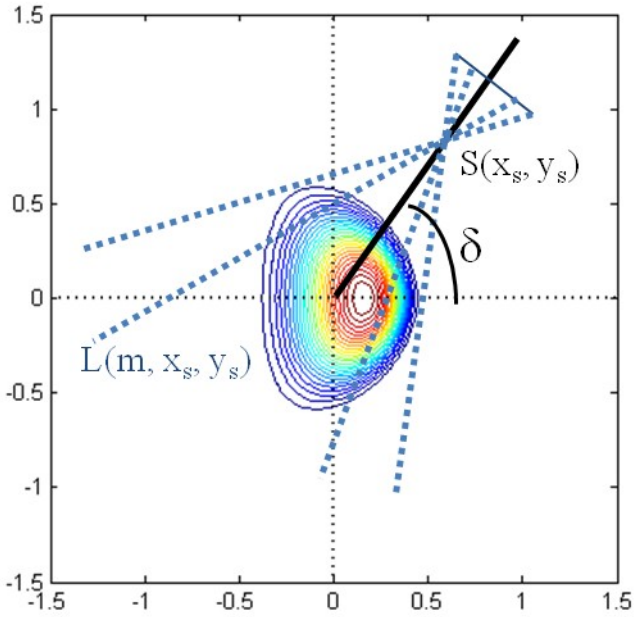
- Emissivity distributions with 0, 1, 2, 3 and 4 perturbations
  - 10% Gaussian noise
- Characteristic of perturbations
  - Narrow
  - Amplitude $A_p = K\, A_E$ ($K = 0.8, 1, 1.5$)
- How many projections are required?
- How many training samples?



Applications using confident machines

- Alternative methods to tomography in plasma physics

Projection geometry



$$I_\delta\left(m, x_s, y_s\right) = \int_{L(m,x_s,y_s)} E(x', y')dl.$$

$$P(\delta) = \left(I_\delta\left(m_1, x_s, y_s\right), \ldots, I_\delta\left(m_{l\delta}, x_s, y_s\right)\right) \in \mathbb{R}$$

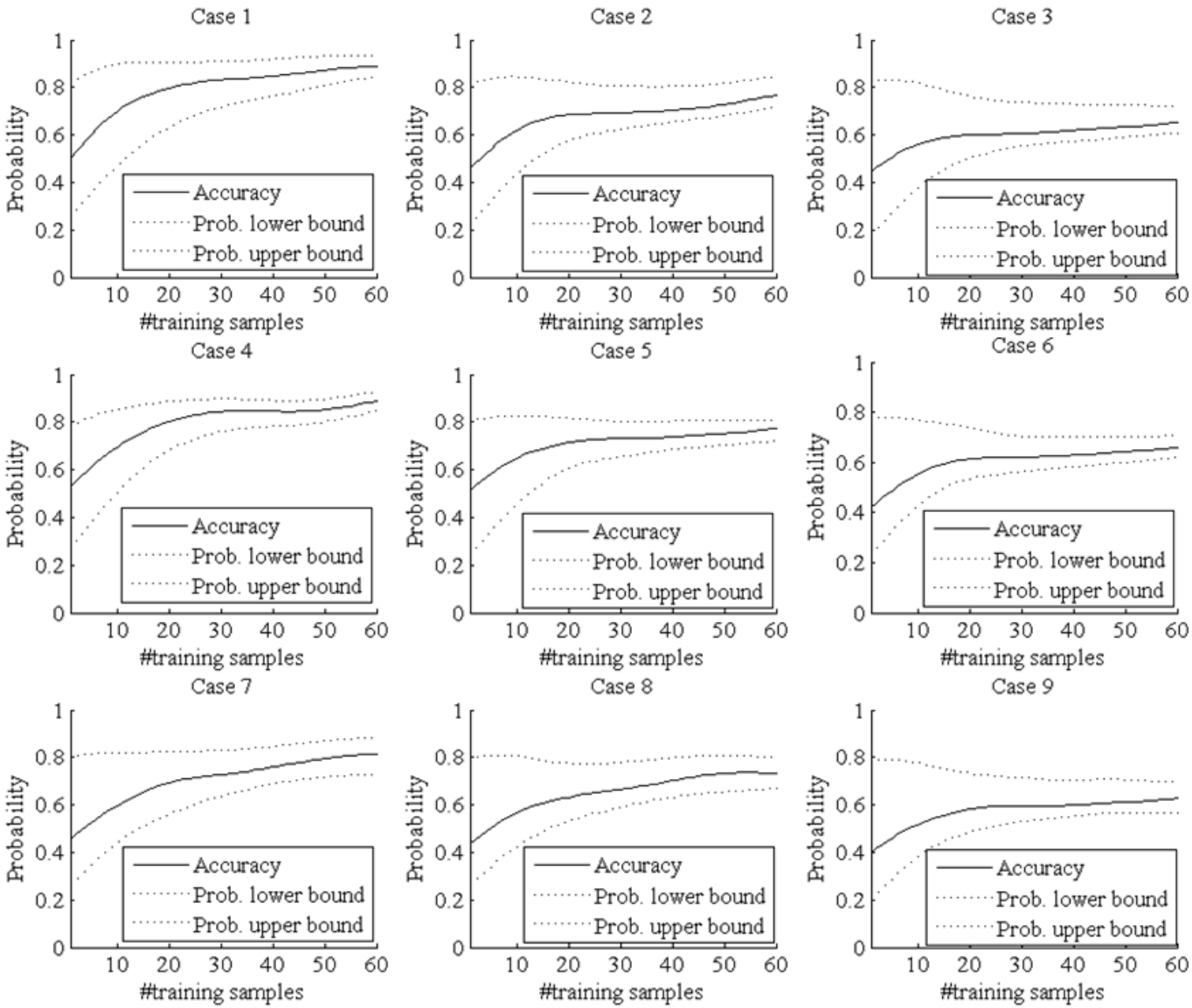
$$\mathbf{x} = \left(P\left(\delta_1\right), \ldots, P\left(\delta_p\right)\right) \in \mathbb{R}$$

Applications using confident machines

- Alternative methods to tomography in plasma physics

Study cases

Case	#projections	Feature vector dimension	δ angles (º)	K
1	3	90	0, 60, 90	1.5
2	3	90	0, 60, 90	1.0
3	3	90	0, 60, 90	0.8
4	1	30	90	1.5
5	1	30	90	1.0
6	1	30	90	0.8
7	1	30	0	1.5
8	1	30	0	1.0
9	1	30	0	0.8



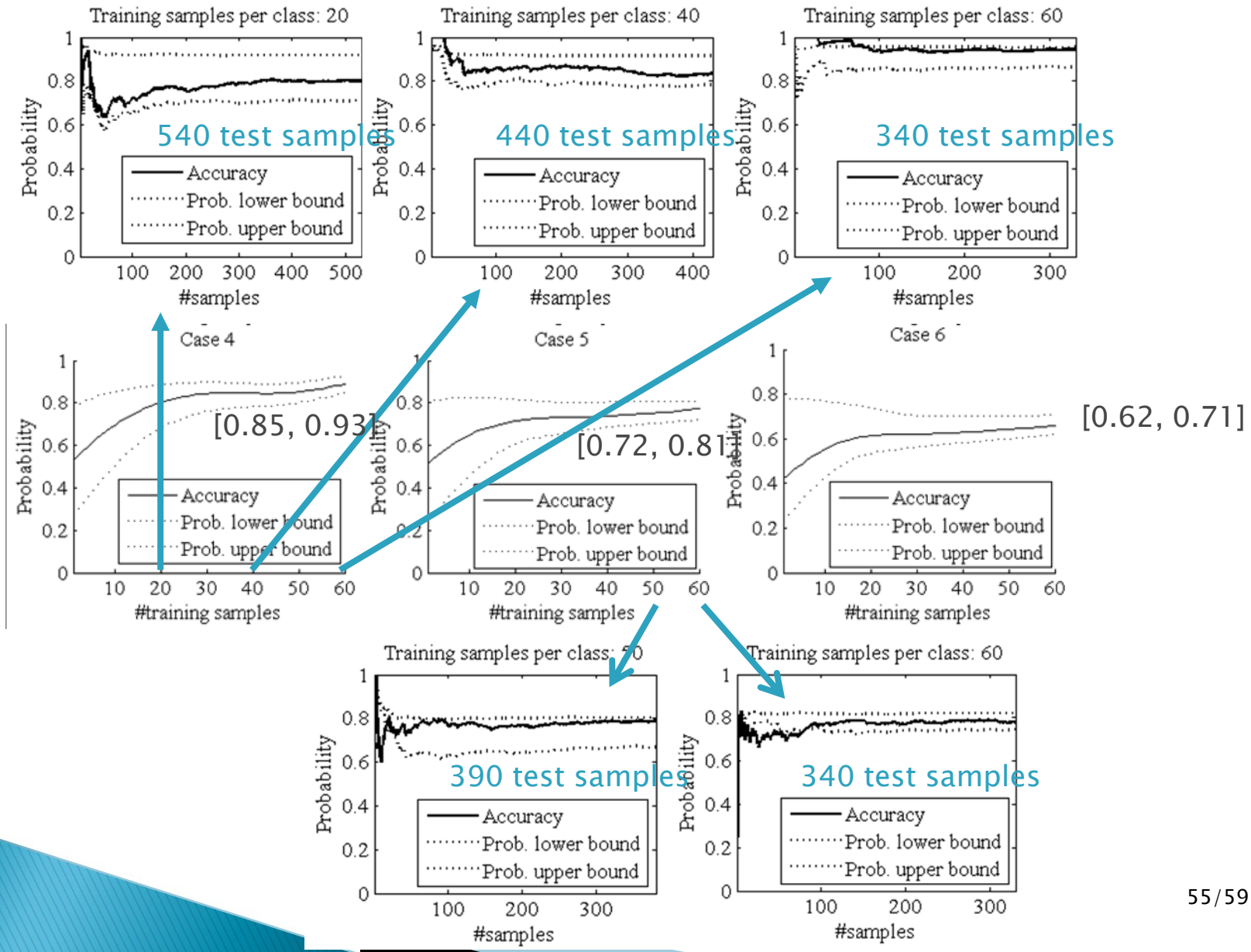
3 projections

1 projection
$\delta = 90$

1 projection
$\delta = 0$

$K = 1.5$ $K = 1$ $K = 0.8$

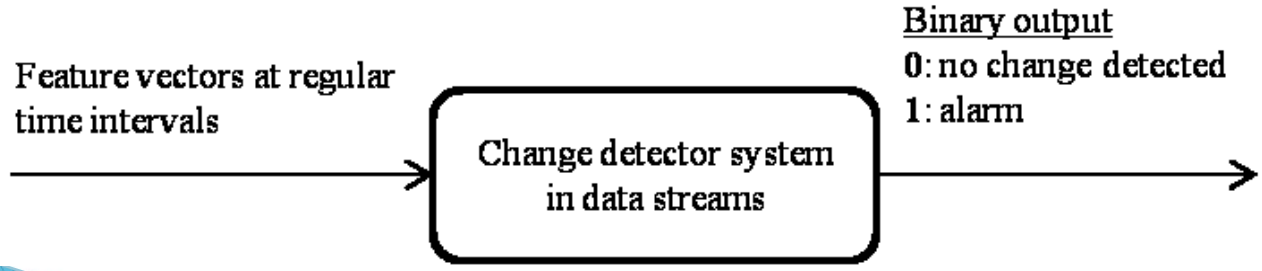
Venn predictors evaluations



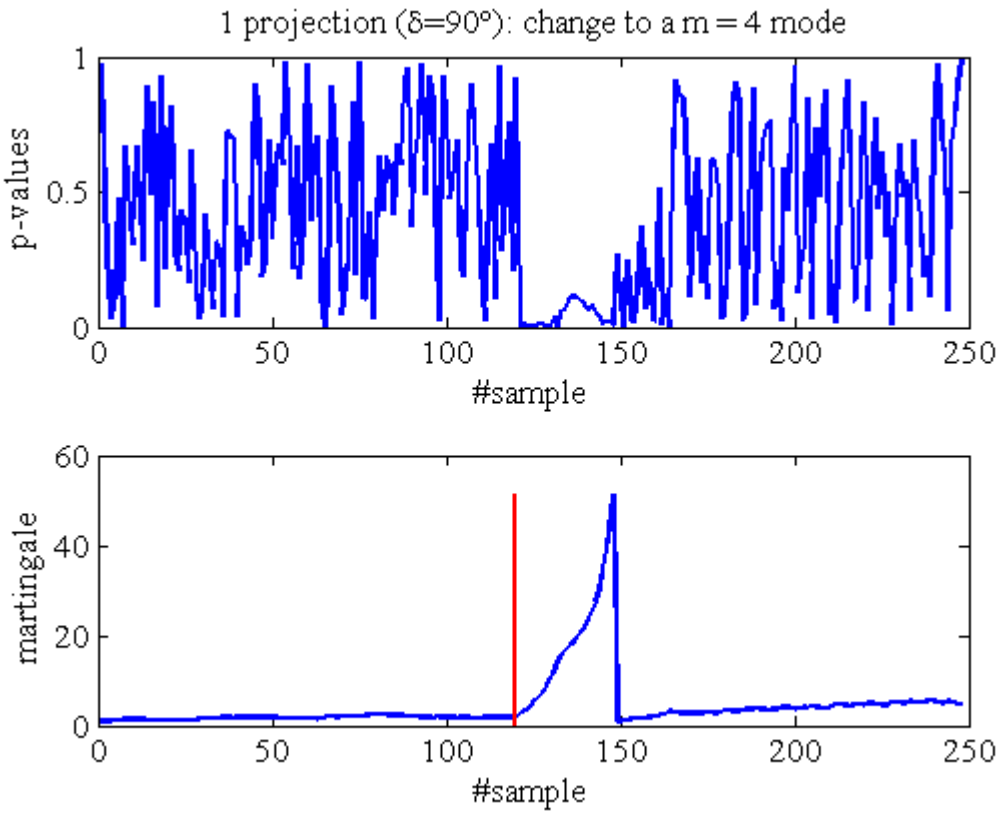
Recognition of plasma status changes through martingales

- What happens in the case of
  - Low efficiency predictors?
  - Not enough labelled samples to train a classifier?
- A martingale framework for detecting changes in data streams by testing exchangeability
  - Randomized power martingale:

$$M_n^{(\varepsilon)} = \prod_{i=1}^{n} \left(\varepsilon \hat{p}_i^{\varepsilon - 1} \right)$$



Results



	Delay (samples)			
	m = 1	**m = 2**	**m = 3**	**m = 4**
ε = 0.95	44 ± 6	27 ± 2	24 ± 3	25 ± 1
ε = 0.96	61 ± 2	40 ± 3	35 ± 1	36 ± 2
ε = 0.97	81 ± 6	69 ± 5	57 ± 3	58 ± 2

The martingale operates as long as $0 < M_n^{(\varepsilon)} < \lambda \left(\lambda = 50\right)$

False alarm rate: $1/\lambda$

Neither false alarms nor missed alarms observed for $\lambda = 50$ and $0.95 \leq \varepsilon \leq 0.97$

For $\varepsilon \in \left[0.9, 0.95\right)$, one false alarm appears sometimes in the transition to m = 1

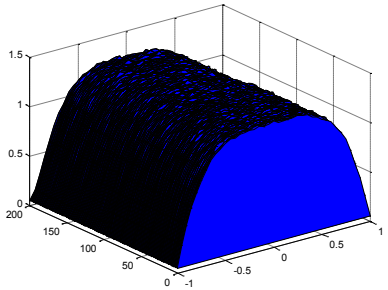
Missed alarms appear if $\varepsilon > 0.97$

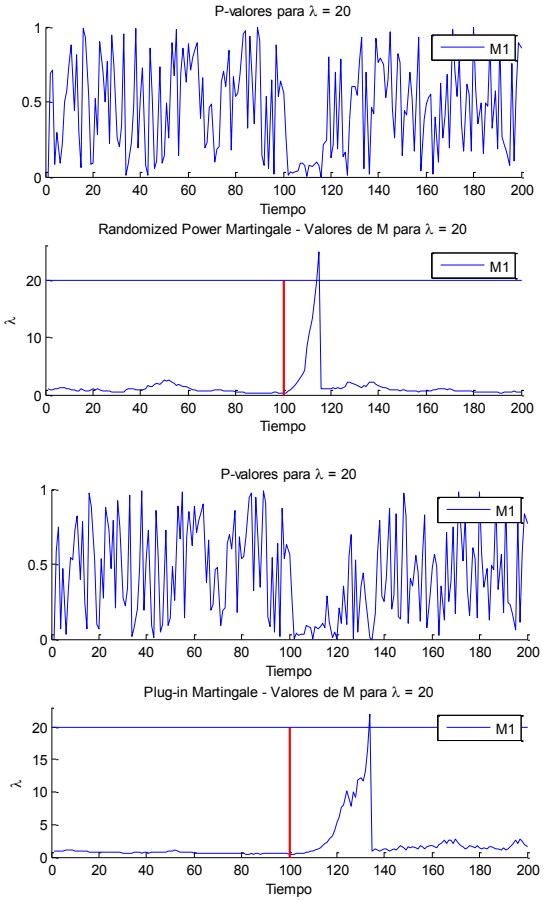
The approach is valid for RT and off-line analysis

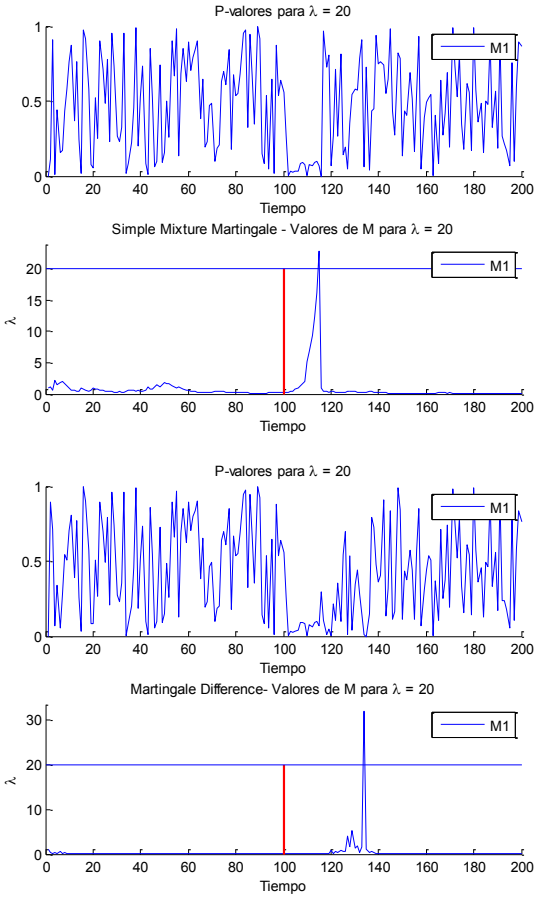
Change of context

$$A(r) = A\left(1 - r^{\alpha_1}\right)^{\alpha_2} + A_0, -1 \le r \le 1$$

- Sequence 1: α2 = 1.015
- Sequence 2: α2 = 1







Summary and future work

- Massive databases in nuclear fusion require intelligent data analysis techniques
- Machine learning techniques (classification and regression) have proved to be powerful tools in fusion
  - Data access, real-time and off-line processing
- Confidence machines are essential elements to obtain accurate and reliable predictions
- HPC shared environments can play a crucial role in the dissemination of intelligent techniques for data analysis