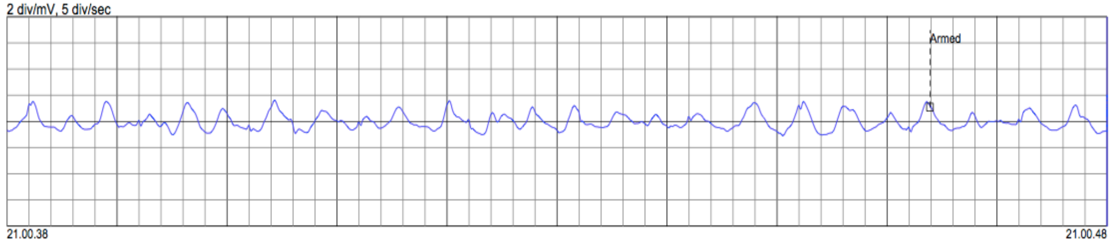
A classifier with guaranteed specificity and sensitivity

Algo Carè,

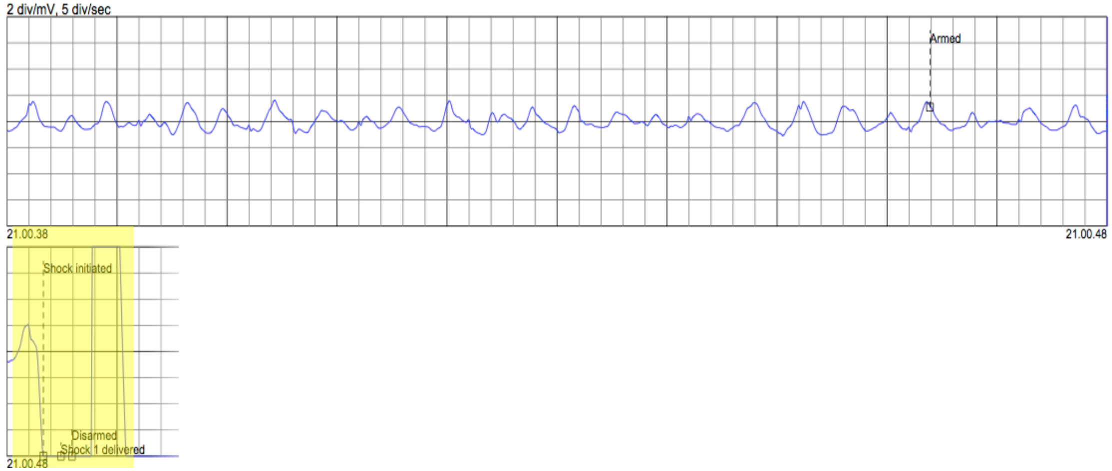
in collaboration with
Marco C. Campi & Federico A. Ramponi
University of Brescia (Italy)

and many others

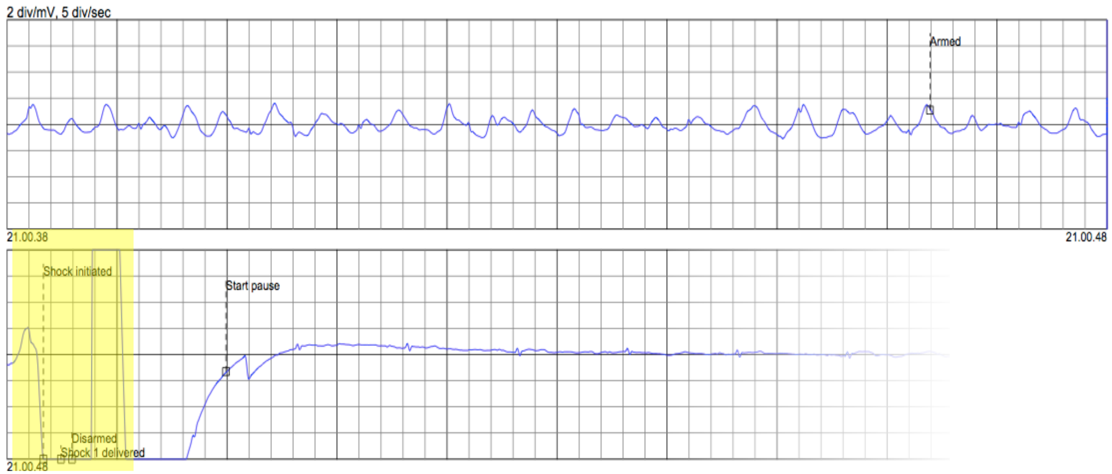
A difficult decision



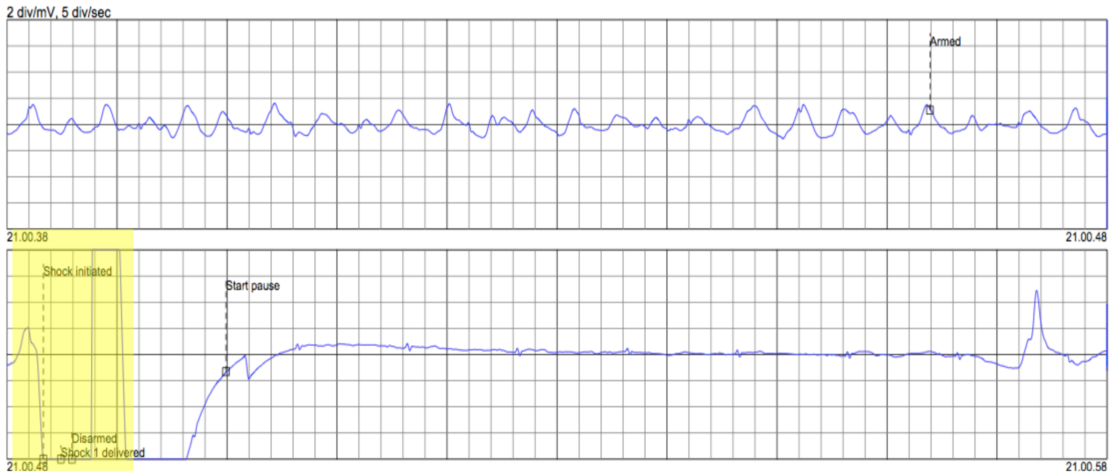
A difficult decision



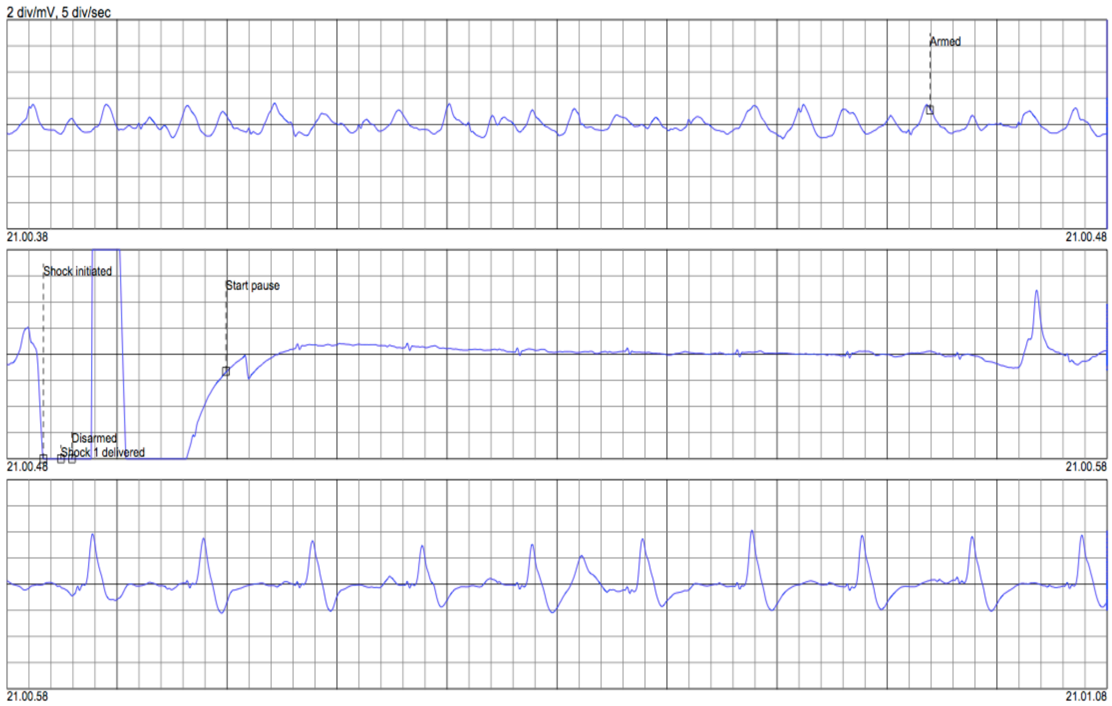
A difficult decision



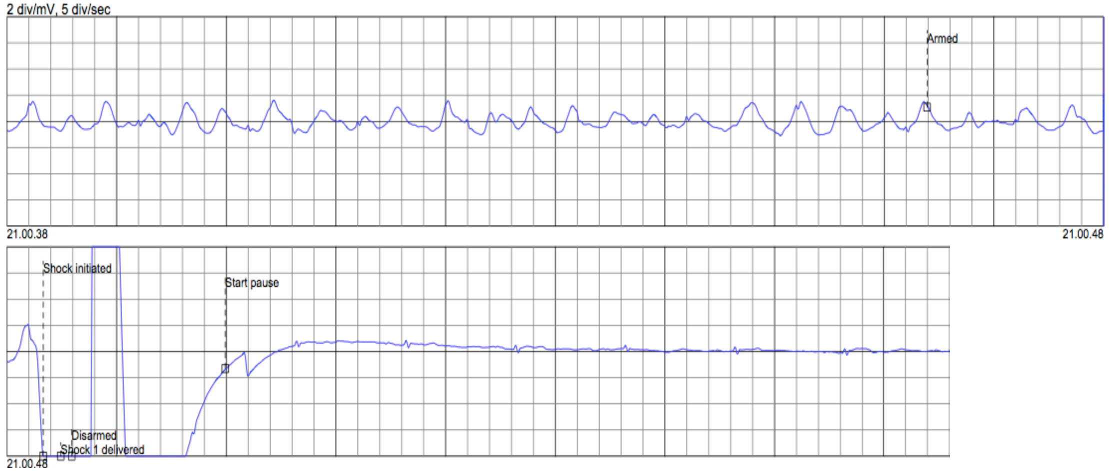
A difficult decision



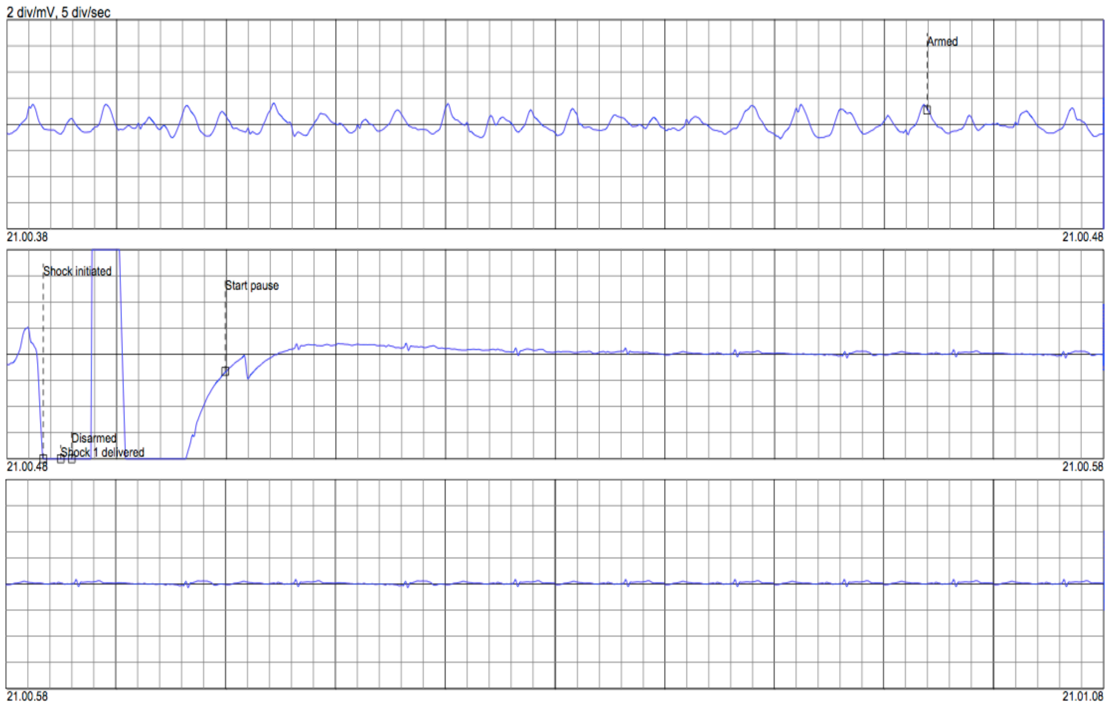
A difficult decision



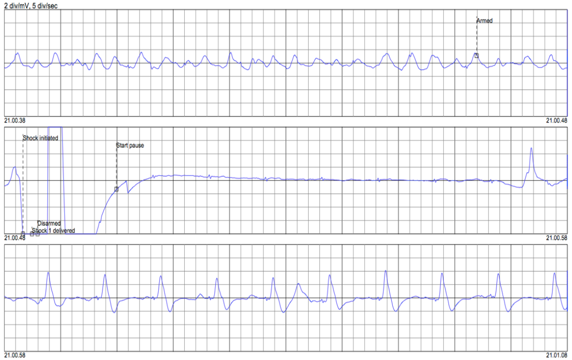
A difficult decision



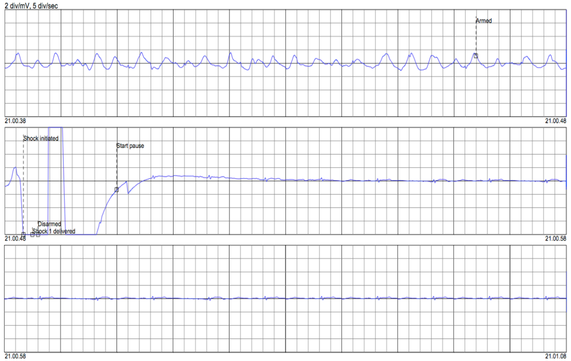
A difficult decision



A difficult decision



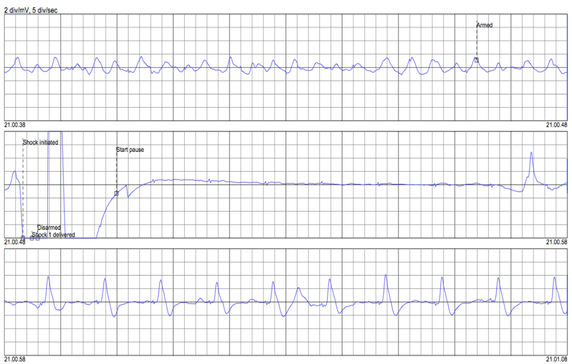
?



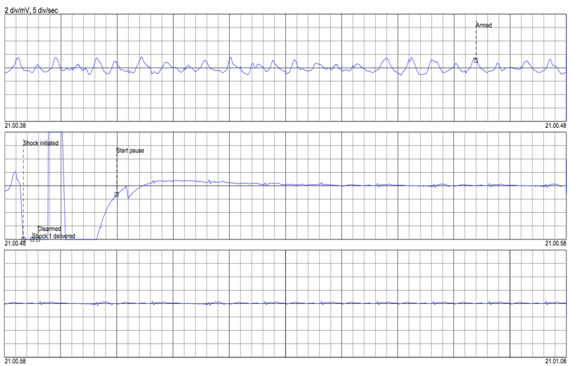
A difficult decision

"each minute of delay before defibrillation
reduces the probability of survival
to discharge by 10-12%"

European resuscitation council guidelines for resuscitation (Resuscitation, 95:100–147, 2015).



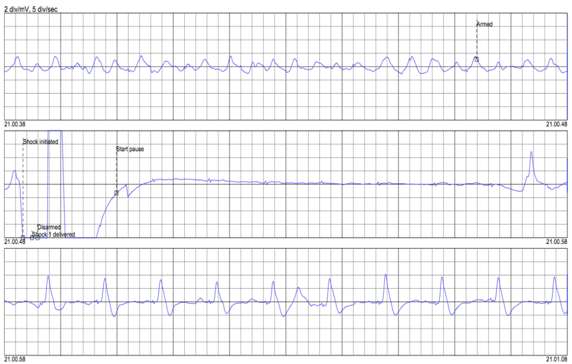
?



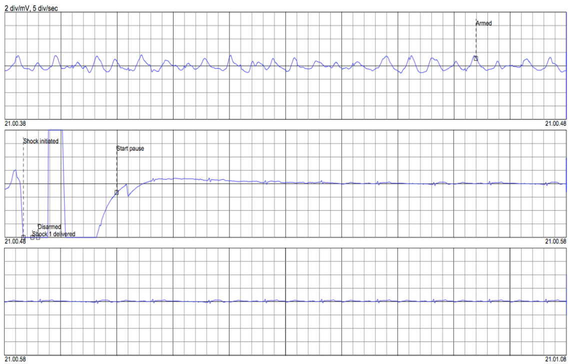
A difficult decision

"each minute of delay before defibrillation
reduces the probability of survival
to discharge by 10-12%"

European resuscitation council guidelines for resuscitation (Resuscitation, 95:100–147, 2015).

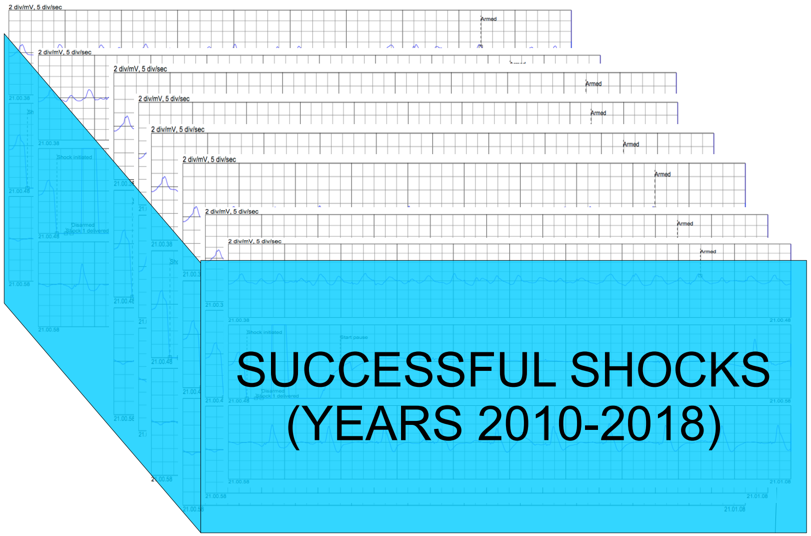


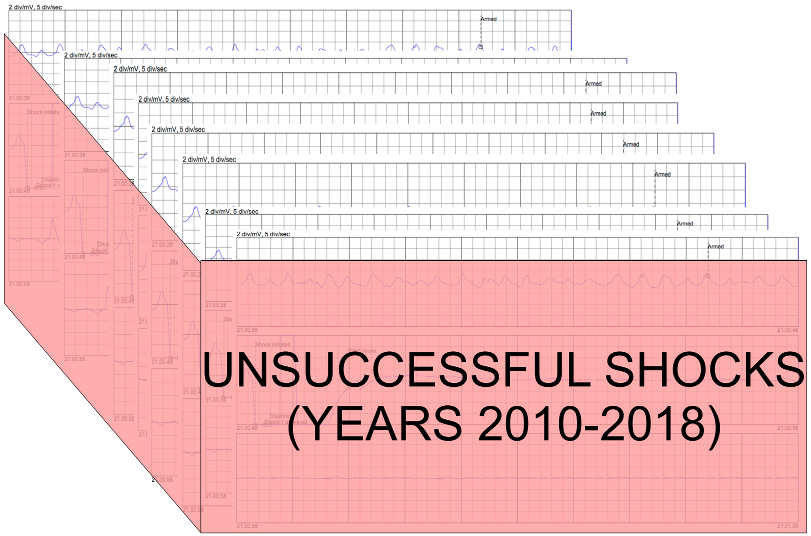
?



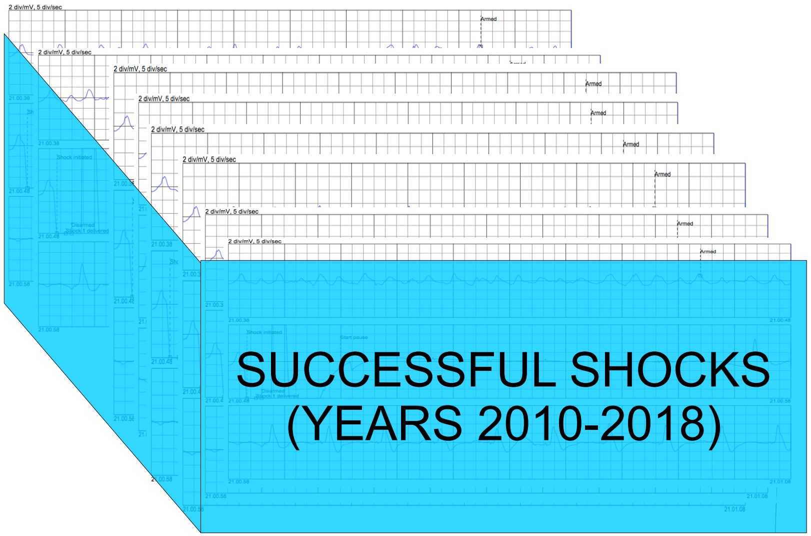
high energy shocks can cause damage
(myocardial injury)

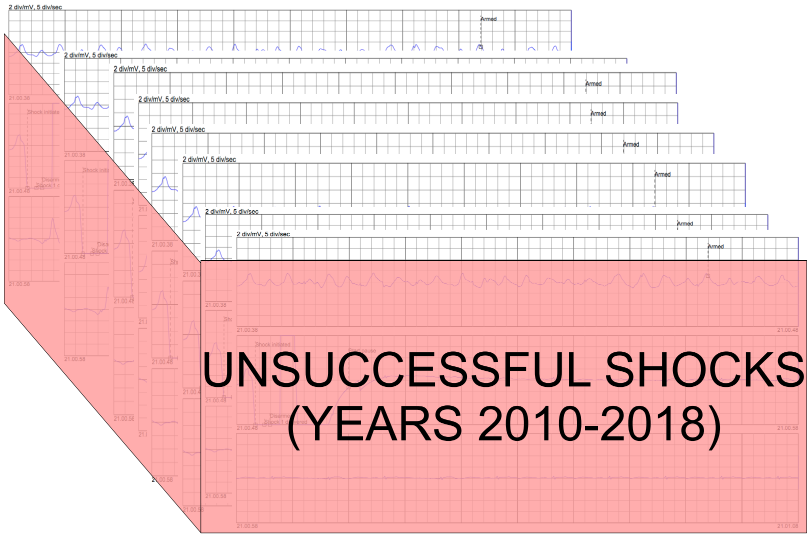
Learning from experience





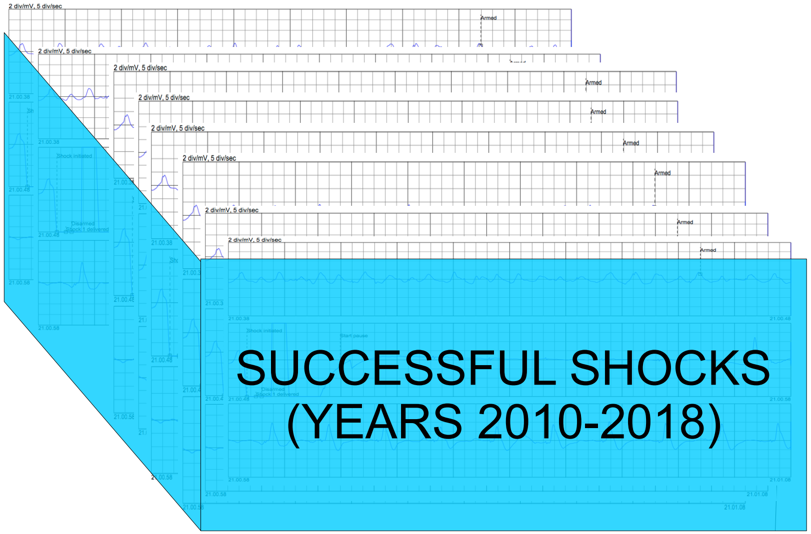
Learning from experience

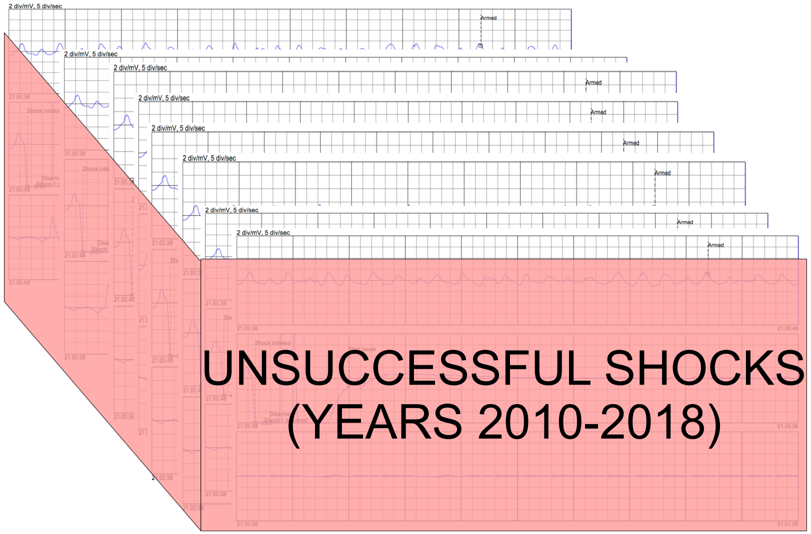




PAST

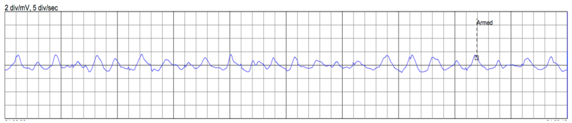
Learning from experience



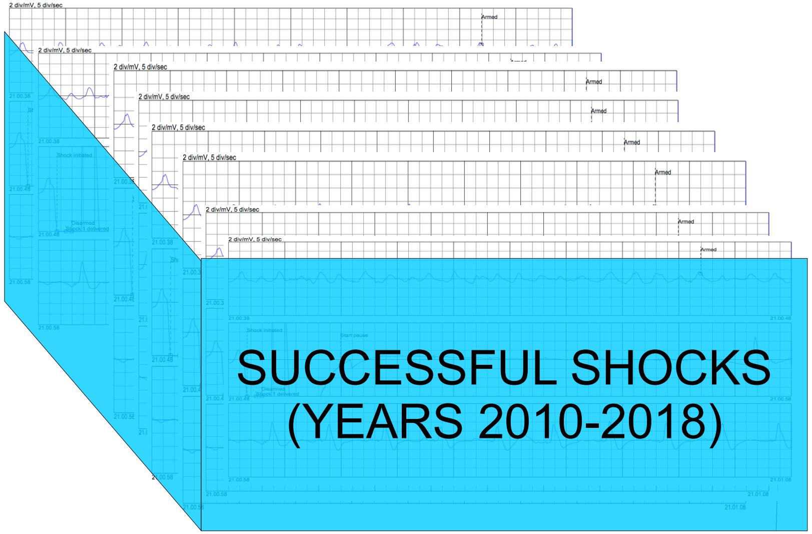


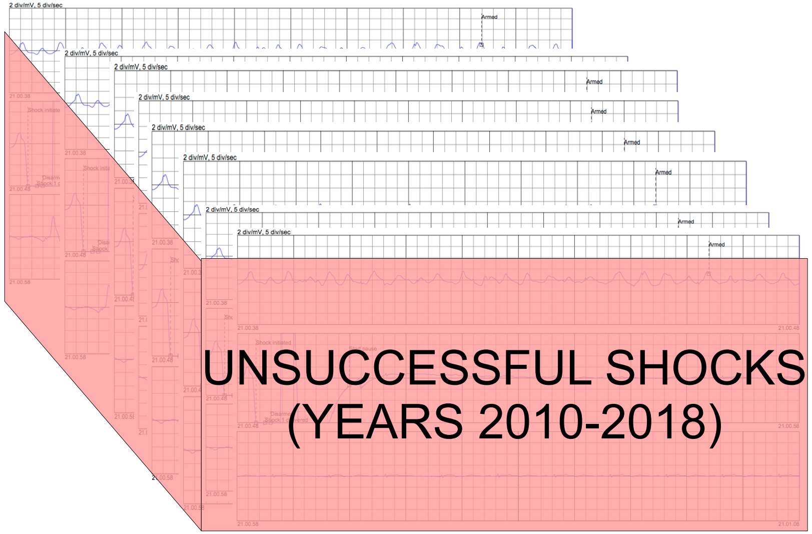
PAST

PRESENT



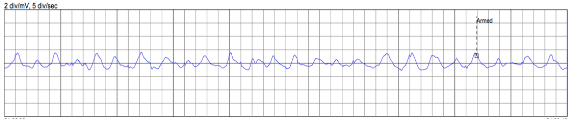
Learning from experience





PAST

PRESENT

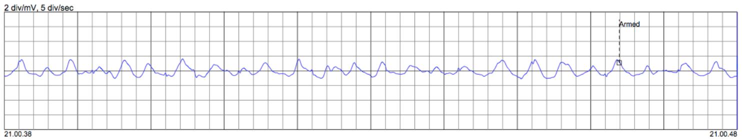


?

SUCCESS

FAILURE

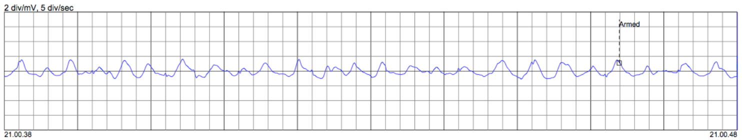
Feature extraction



- PTT (peak to peak amplitude)
- Amax (maximum amplitude)
- Amin (minimum amplitude)
- RMS (root mean square)

- DF (dominant frequency)
- CF (spectral centroid frequency)
- EF (edge frequency)
- AMSA (amplitude spectral area)

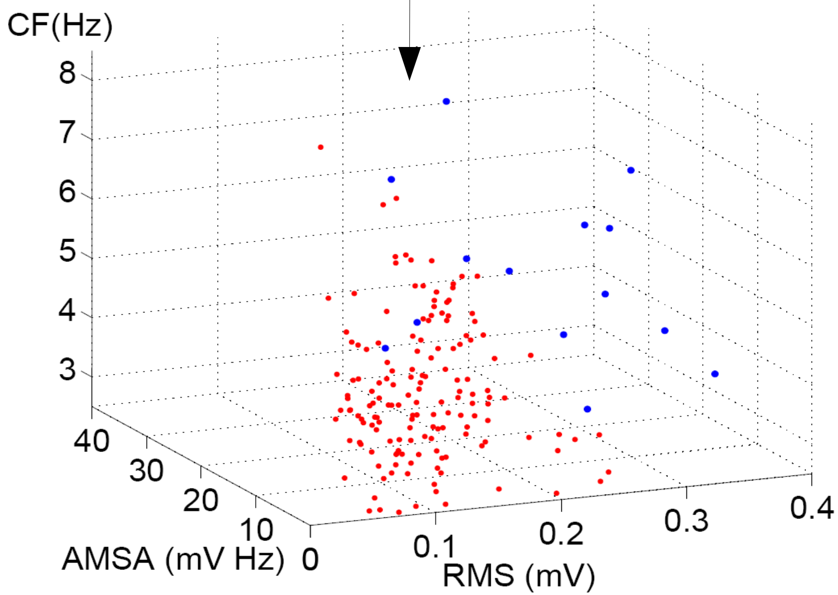
Feature extraction



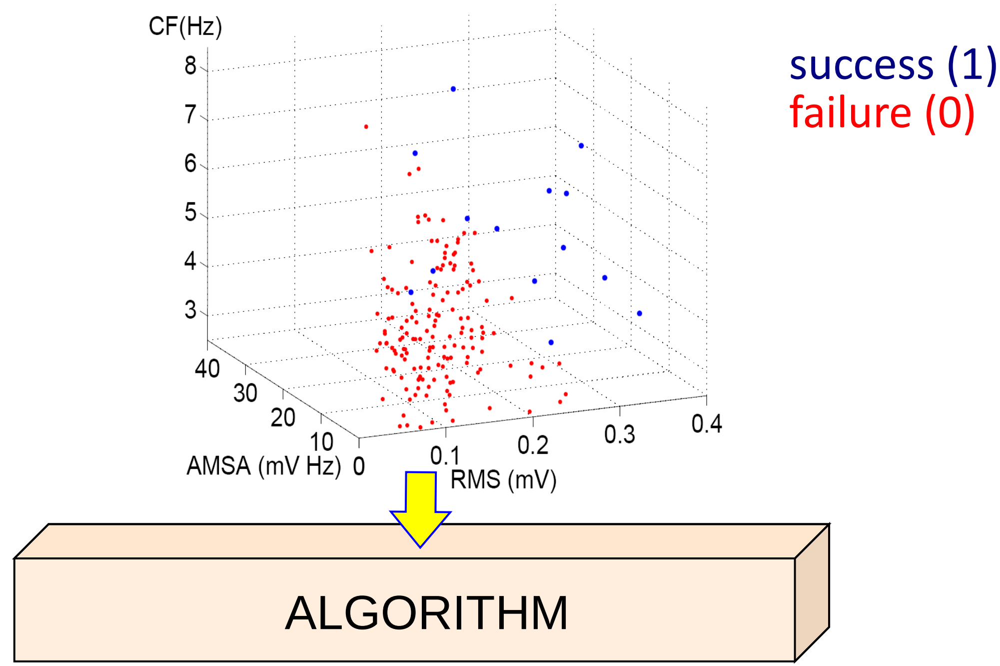
- PTT (peak to peak amplitude)
- Amax (maximum amplitude)
- Amin (minimum amplitude)
- **RMS** (root mean square)

- DF (dominant frequency)
- **CF** (spectral centroid frequency)
- EF (edge frequency)
- **AMSA** (amplitude spectral area)

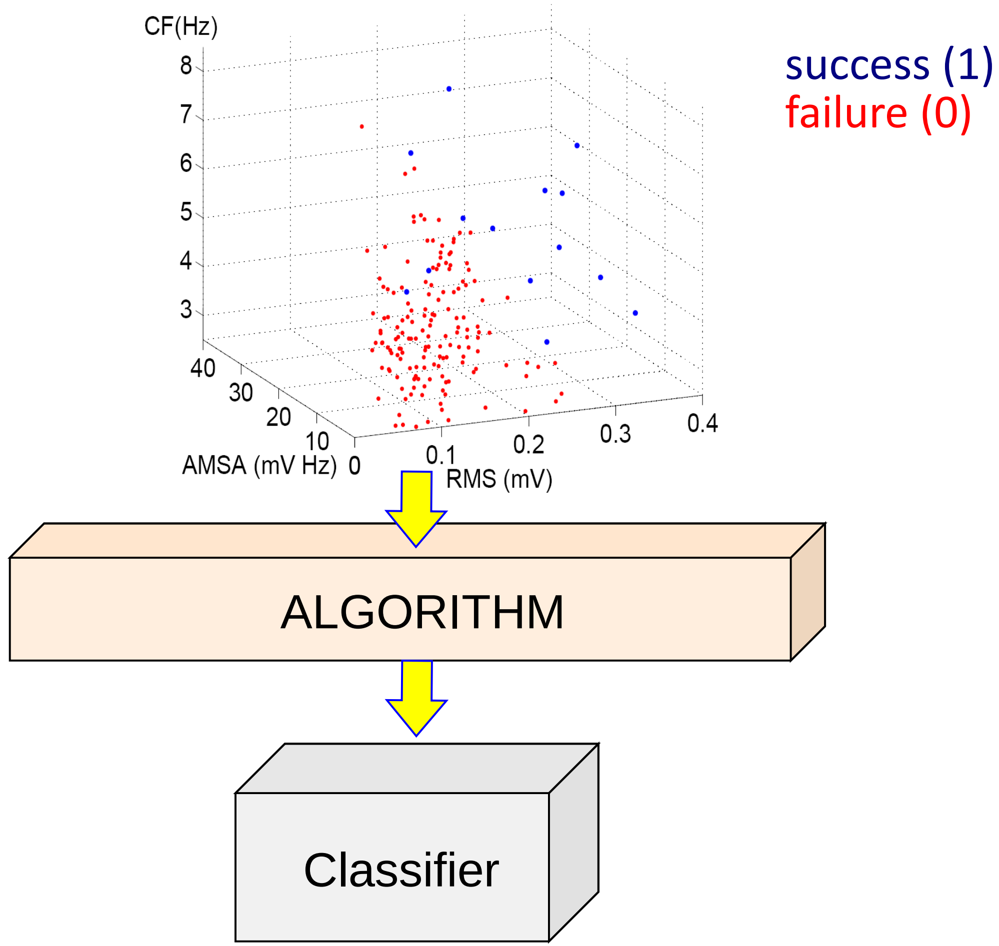
success (1)
failure (0)



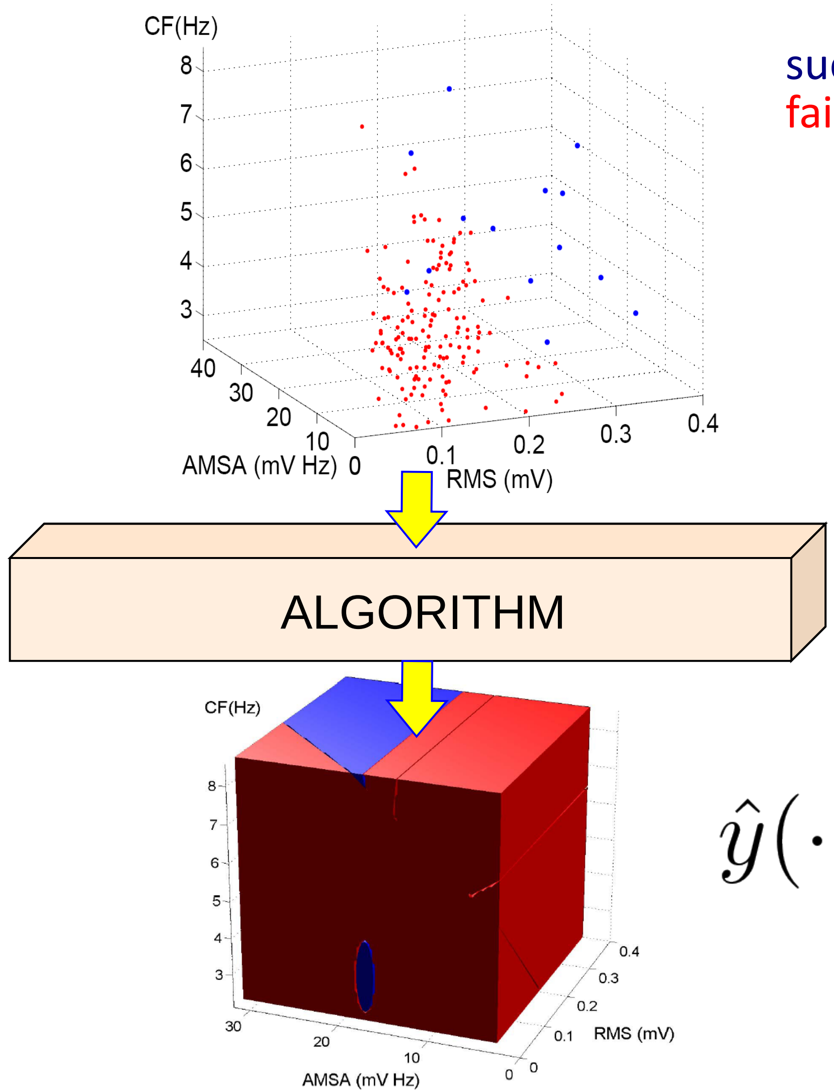
Automatic learning



Automatic learning



Automatic learning



success (1)
failure (0)

$$\hat{y}(\cdot) : \mathbb{R}^3 \to \{1, 0\}$$

Mathematical set-up

$\mathrm{x} \in \mathbb{R}^n$: patient ($n$ features)

$y \in \{1, 0\}$: true outcome

Mathematical set-up

$\mathrm{x} \in \mathbb{R}^n$: patient ($n$ features)

$y \in \{1, 0\}$: true outcome

$\hat{y}(\cdot) : \mathbb{R}^n \to \{1, 0\}$ classifier

Mathematical set-up

$\mathrm{x} \in \mathbb{R}^n$: patient ($n$ features)

$y \in \{1, 0\}$: true outcome

$\hat{y}(\cdot) : \mathbb{R}^n \to \{1, 0\}$ classifier

$$\hat{y}(x) \neq y$$

Mathematical set-up

$\mathrm{x} \in \mathbb{R}^n$: patient ($n$ features)

$y \ \in \{1, 0\}$: true outcome

$\hat{y}(\cdot) : \mathbb{R}^n \to \{1, 0\}$ classifier

$(x, y) \sim \mathbb{P}$

$$\hat{y}(x) \neq y$$

Mathematical set-up

$\mathrm{x} \in \mathbb{R}^n$: patient ($n$ features)

$y \in \{1, 0\}$: true outcome

$\hat{y}(\cdot) : \mathbb{R}^n \to \{1, 0\}$ classifier

$(x, y) \sim \mathbb{P}$

$$\mathbb{P}\{\hat{y}(x) \neq y\}$$

Mathematical set-up

$\mathrm{x} \in \mathbb{R}^n$: patient ($n$ features)

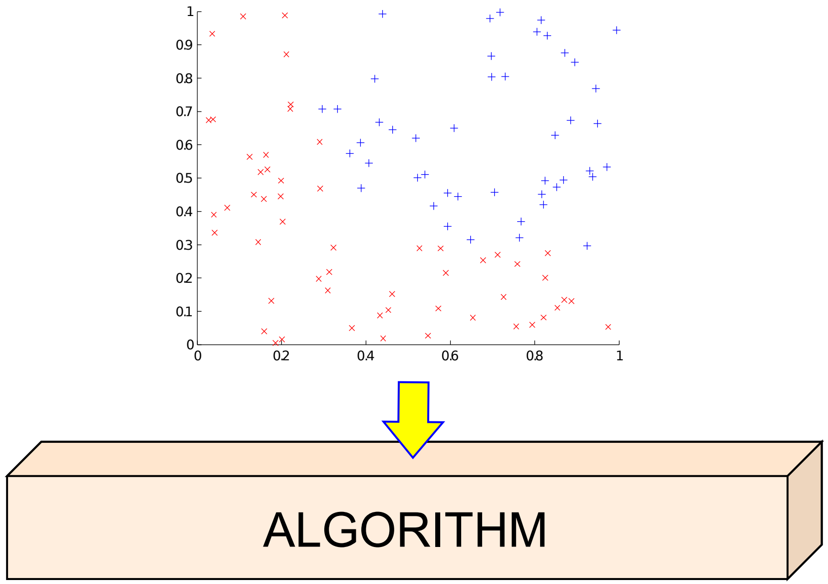
$y \in \{1, 0\}$: true outcome

$\hat{y}(\cdot) : \mathbb{R}^n \to \{1, 0\}$ classifier

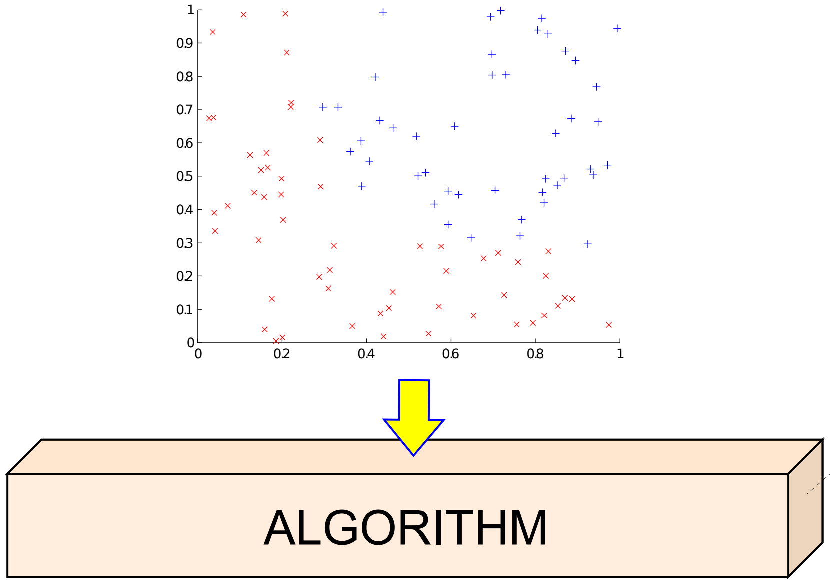
$$(x, y) \sim \mathbb{P}$$

$$PE(\hat{y}) := \mathbb{P}\{\hat{y}(x) \neq y\}$$

Classification through cvx optimisation

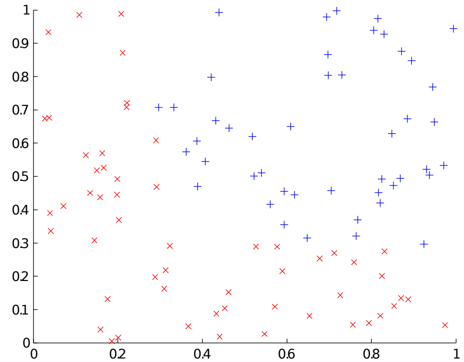


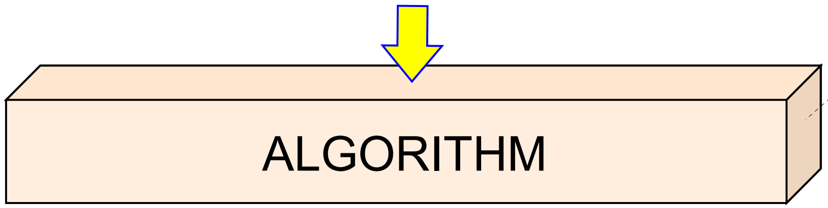
Classification through cvx optimisation

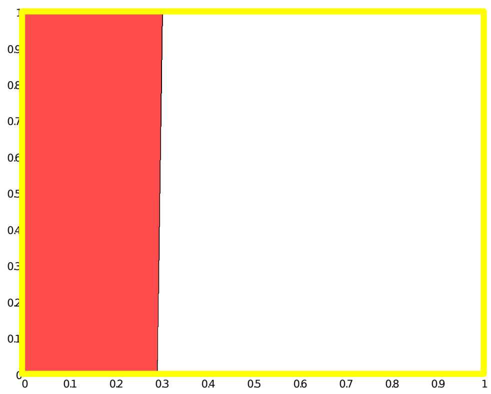


Sequence of
convex optimisation
programs

Classification through cvx optimisation



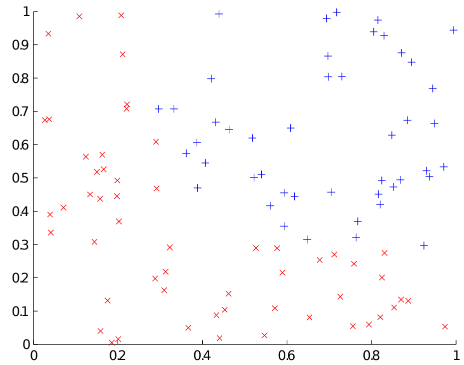


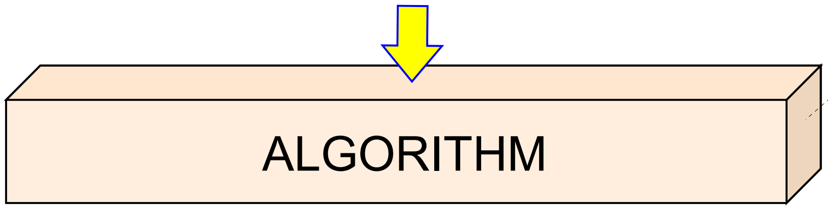


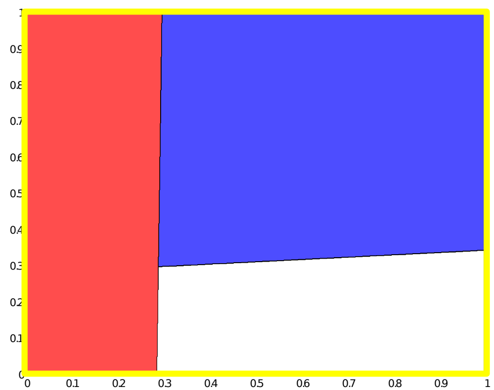
Sequence of convex optimisation programs

$$
\begin{aligned}
&\min_{A=A^T\in\mathbb{R}^{d\times d},\, b\in\mathbb{R}^d} && \mathit{Trace}(A)\\
&\text{subject to:} && (x_i - x_B)^T A (x_i - x_B) + b^T (x_i - x_B) \geq 1,\\
& && \quad \text{for all } i \in P \text{ such that } y_i \neq y(x_B)\\
& && \text{and } A \succeq 0 \quad (A \text{ positive semi-definite}).
\end{aligned}
$$

Classification through cvx optimisation





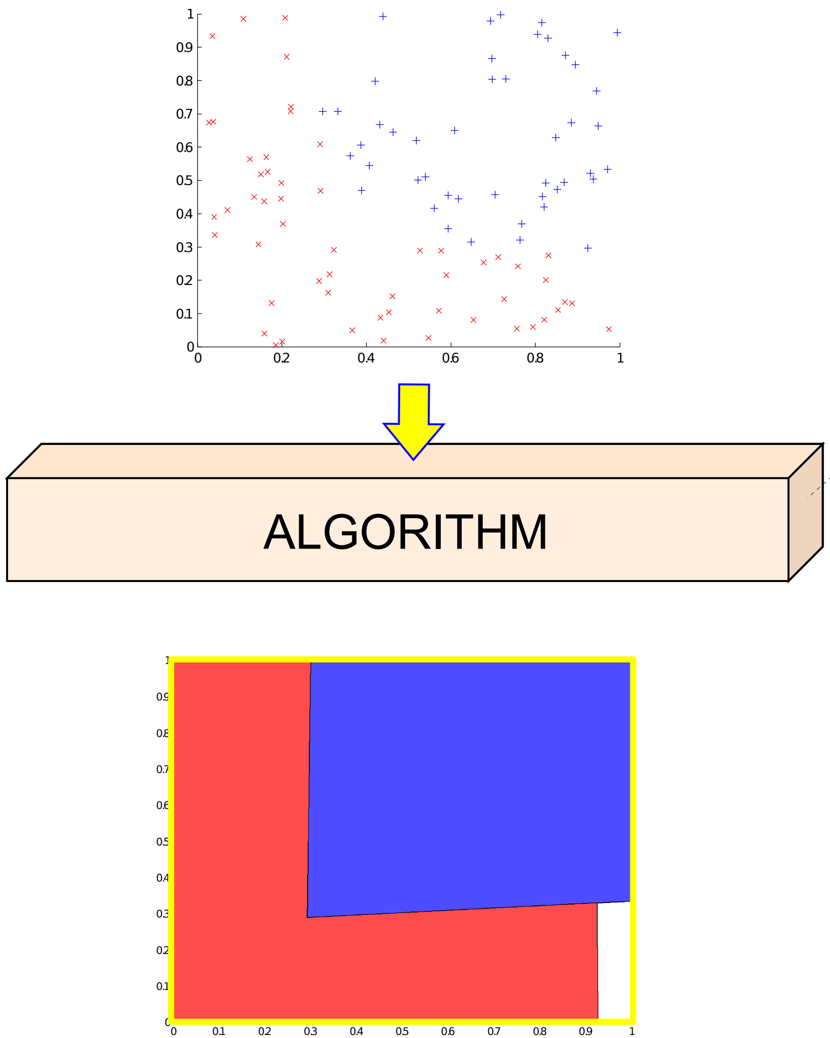


Sequence of convex optimisation programs

$$\begin{aligned}
&\min_{A=A^T\in\mathbb{R}^{d\times d},\, b\in\mathbb{R}^d} \; \mathit{Trace}(A)\\
&\text{subject to:} \quad (x_i - x_B)^T A (x_i - x_B) + b^T (x_i - x_B) \geq 1,\\
&\qquad \text{for all } i \in P \text{ such that } y_i \neq y(x_B)\\
&\qquad \text{and } A \succeq 0 \quad (A \text{ positive semi-definite}).
\end{aligned}$$

$$\vdots$$

Classification through cvx optimisation

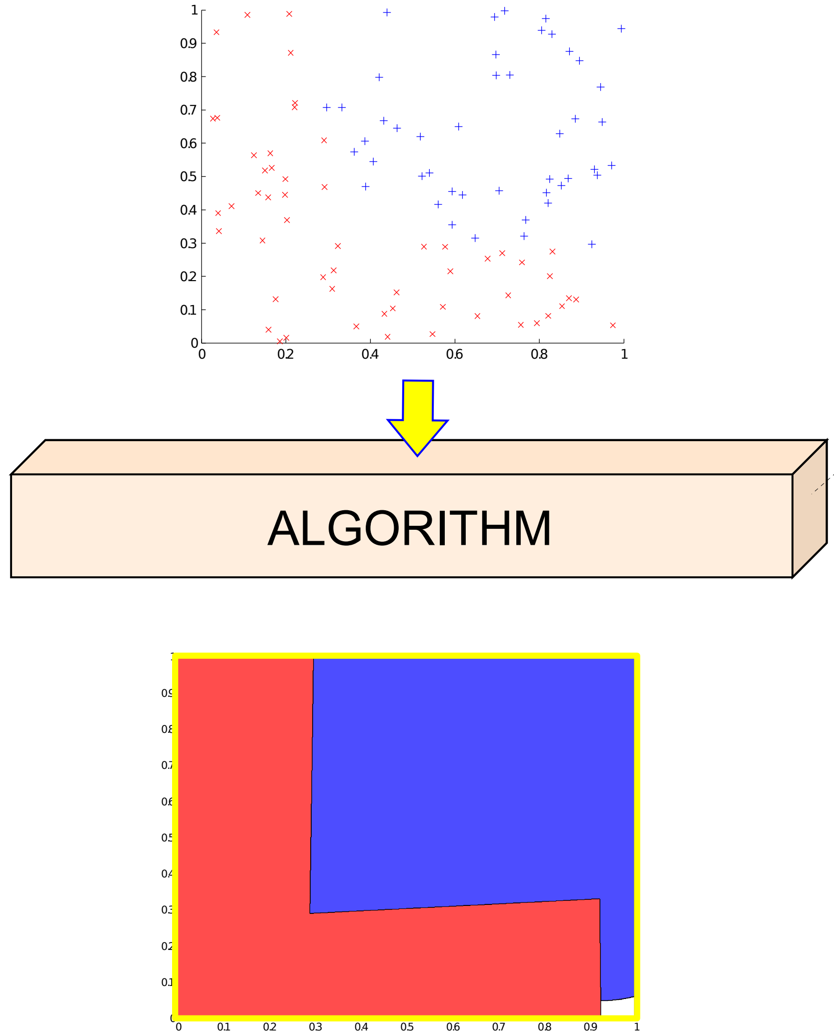


Sequence of convex optimisation programs

$$
\begin{array}{ll}
\min_{A=A^T\in\mathbb{R}^{d\times d},\, b\in\mathbb{R}^d} & Trace(A) \\
\text{subject to:} & (x_i - x_B)^T A (x_i - x_B) + b^T (x_i - x_B) \geq 1, \\
 & \quad \text{for all } i \in P \text{ such that } y_i \neq y(x_B) \\
 & \text{and } A \succeq 0 \quad (A \text{ positive semi-definite}).
\end{array}
$$

$$\vdots$$

Classification through cvx optimisation

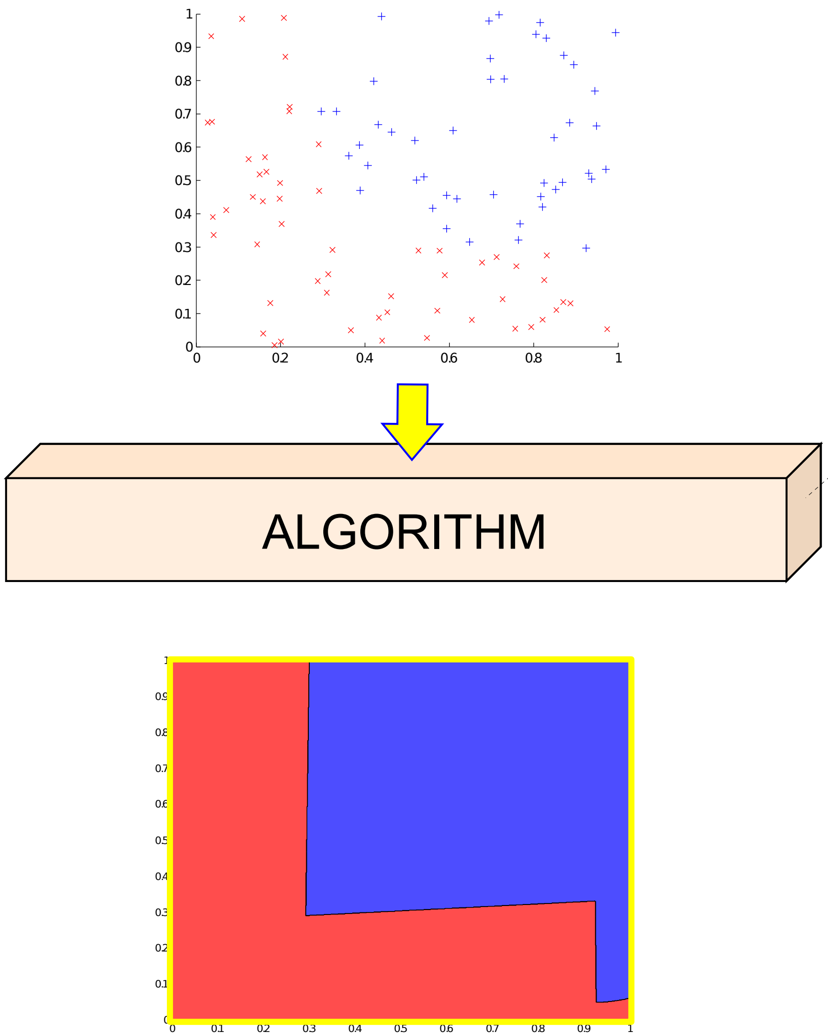


Sequence of
convex optimisation
programs

$$
\begin{array}{ll}
\min_{A=A^T\in\mathbb{R}^{d\times d},\, b\in\mathbb{R}^d} & \mathit{Trace}(A) \\
\text{subject to:} & (x_i - x_B)^T A (x_i - x_B) + b^T (x_i - x_B) \geq 1, \\
 & \quad \text{for all } i \in P \text{ such that } y_i \neq y(x_B) \\
 & \text{and } A \succeq 0 \quad (A \text{ positive semi-definite}).
\end{array}
$$

$$\vdots$$

Classification through cvx optimisation



Sequence of convex optimisation programs

$$\begin{aligned}
&\min_{A=A^T\in\mathbb{R}^{d\times d},\, b\in\mathbb{R}^d} \; \mathit{Trace}(A)\\
&\text{subject to:} \quad (x_i - x_B)^T A (x_i - x_B) + b^T(x_i - x_B) \geq 1,\\
&\qquad\qquad \text{for all } i \in P \text{ such that } y_i \neq y(x_B)\\
&\qquad\qquad \text{and } A \succeq 0 \quad (A \text{ positive semi-definite}).
\end{aligned}$$

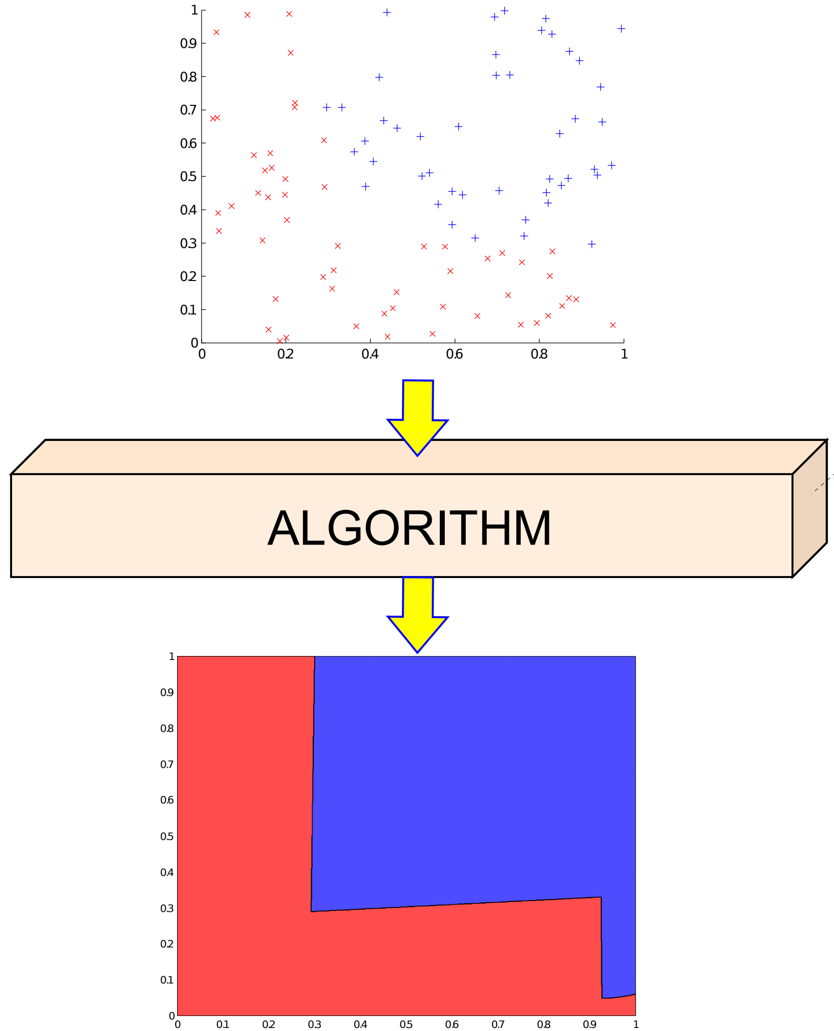
•

•

•

$$\begin{aligned}
&\min_{a\geq 0,\, b\in\mathbb{R}^d} \; a\\
&\text{subject to:} \quad a\cdot\|x_i - x_B\|^2 + b^T(x_i - x_B) \geq 1,\\
&\qquad\qquad \text{for all } i \in P \text{ such that } y_i \neq y(x_B).
\end{aligned}$$

Classification through cvx optimisation



Sequence of
convex optimisation
programs

The theory behind

Theorems for
data-driven optimisation problems

The theory behind

Theorems for
data-driven optimisation problems

Classification with

guaranteed probability of error

The theory behind

Theorems for

data-driven optimisation problems

↓

Classification with

user-chosen

guaranteed probability of error

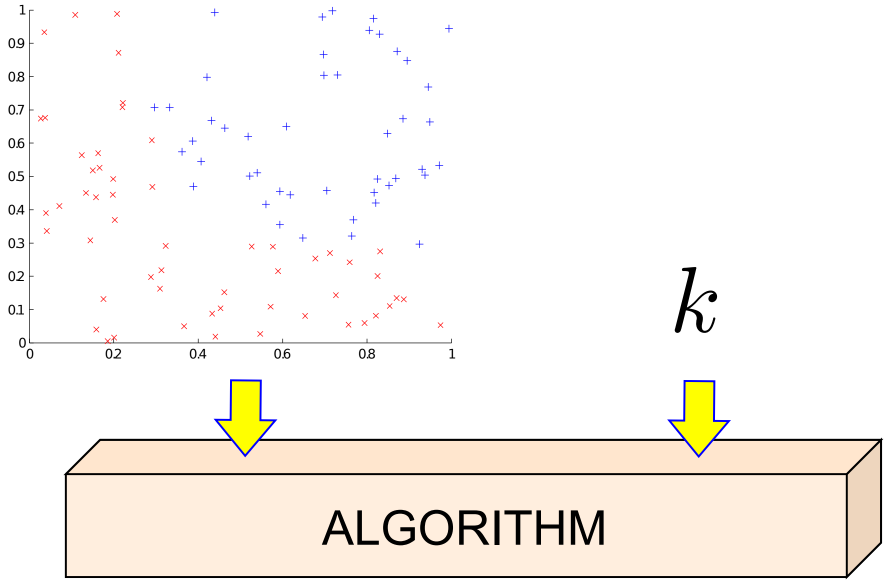
The theory behind

Theorems for
data-driven optimisation problems

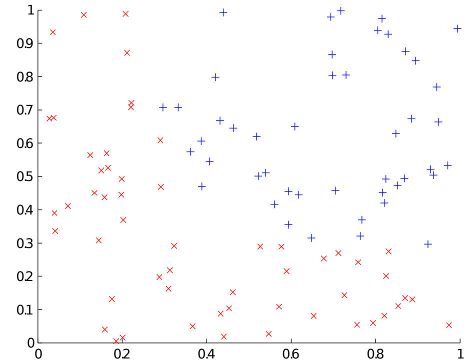
Classification with
user-chosen
guaranteed probability of error

$$\Pr\{\hat{y}(x) \neq y\} \leq 10\%$$

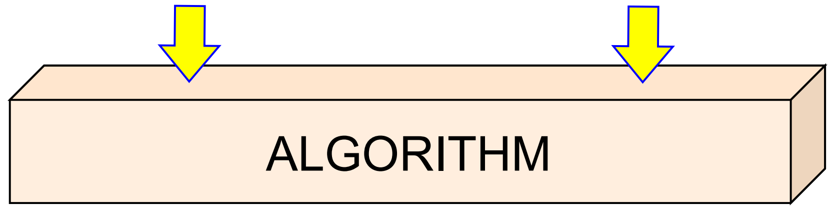
A complexity parameter



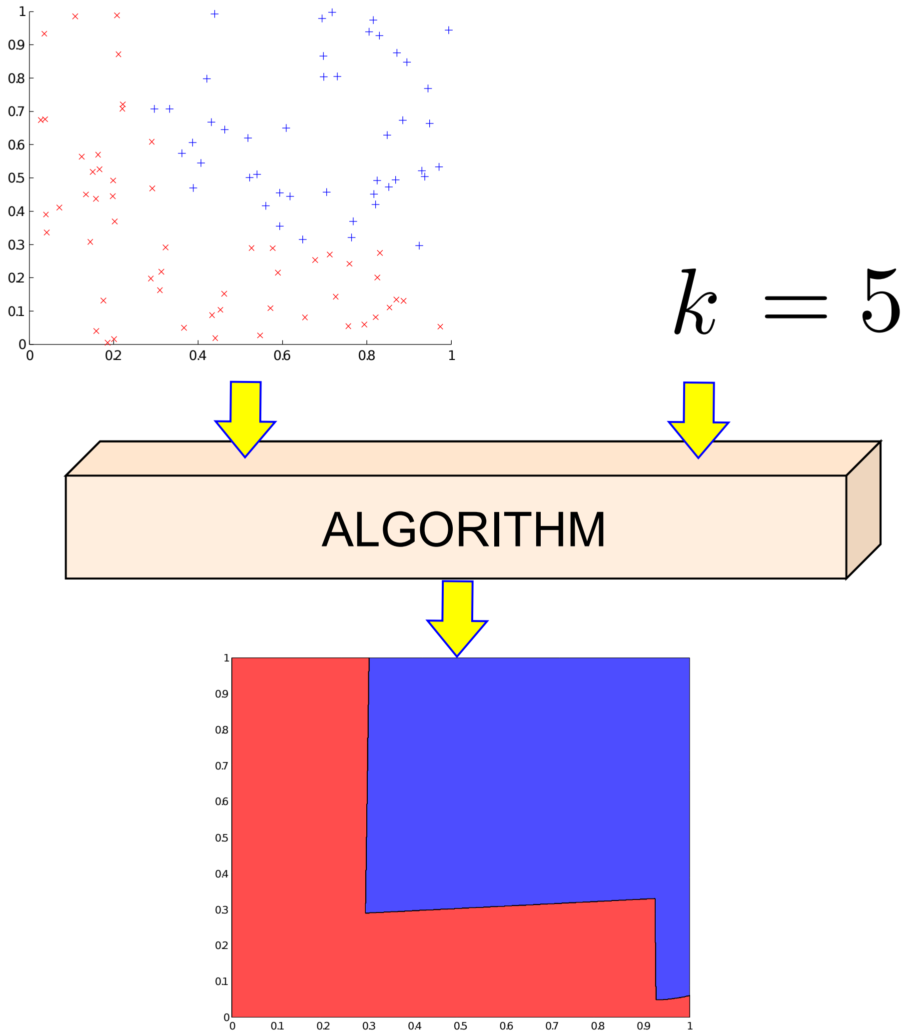
A complexity parameter



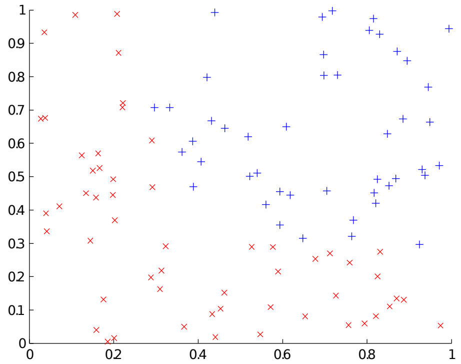
$$k = 5$$



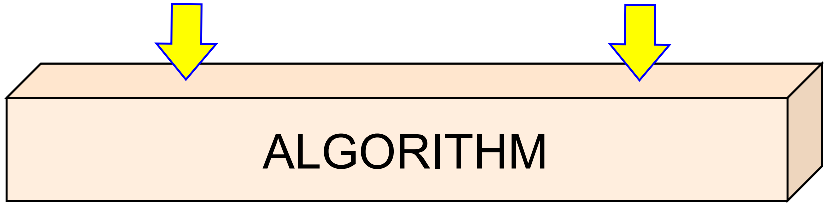
A complexity parameter



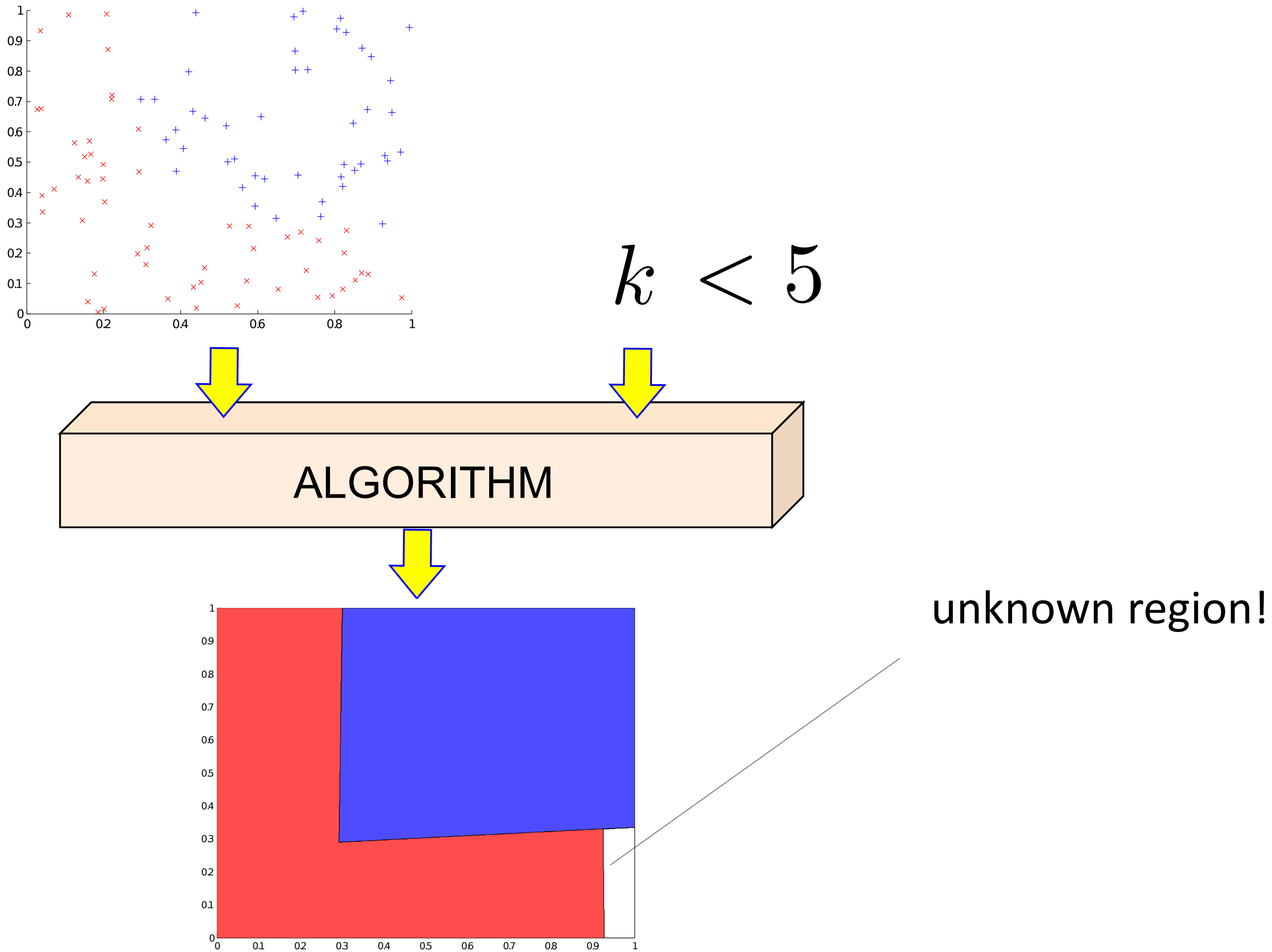
A complexity parameter



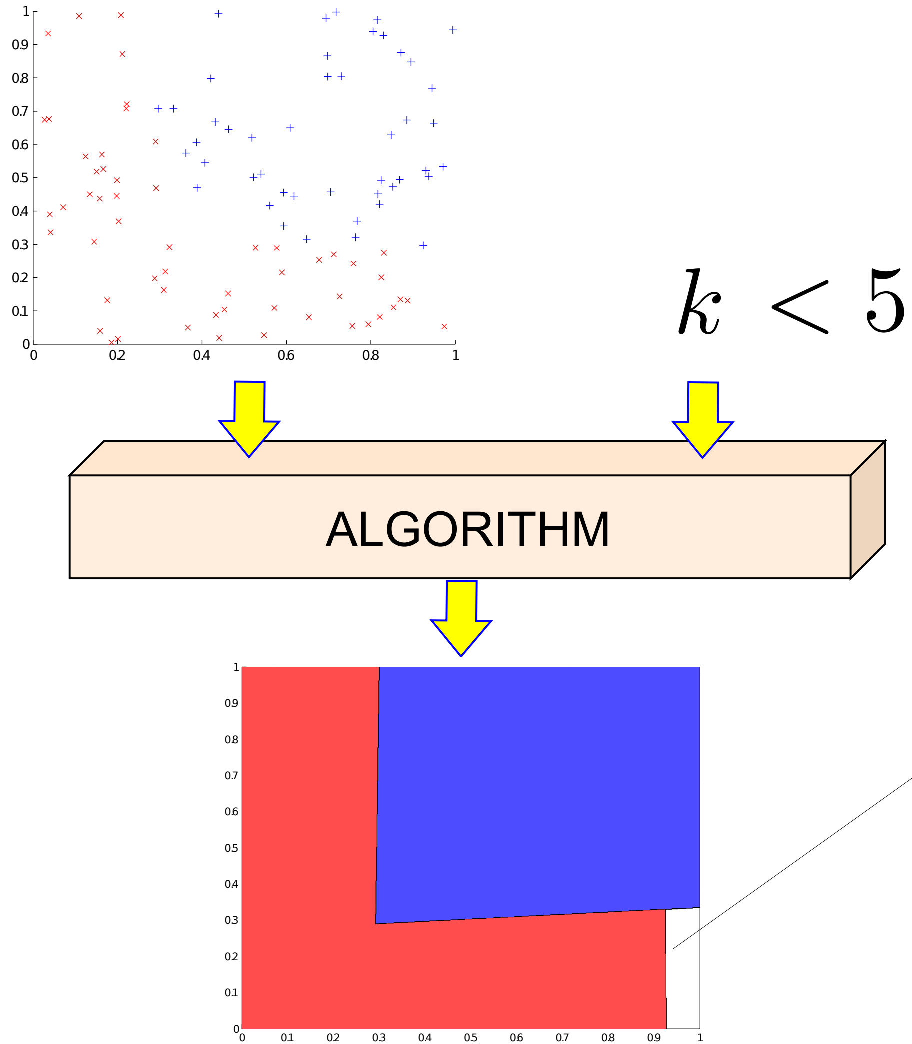
$$k < 5$$



A complexity parameter



A complexity parameter



unknown region!

see also Theorem 7.1 in
G. Lugosi, L. Devroye, L. Gyorfi:
”A Probabilistic
Theory of Pattern
Recognition”, 1996.

Classification with Guarantees

N: size of the training set

k: complexity parameter

$\downarrow$

determine the

probability of error

Classification with Guarantees

N: size of the training set

k: complexity parameter

$\downarrow$

determine the

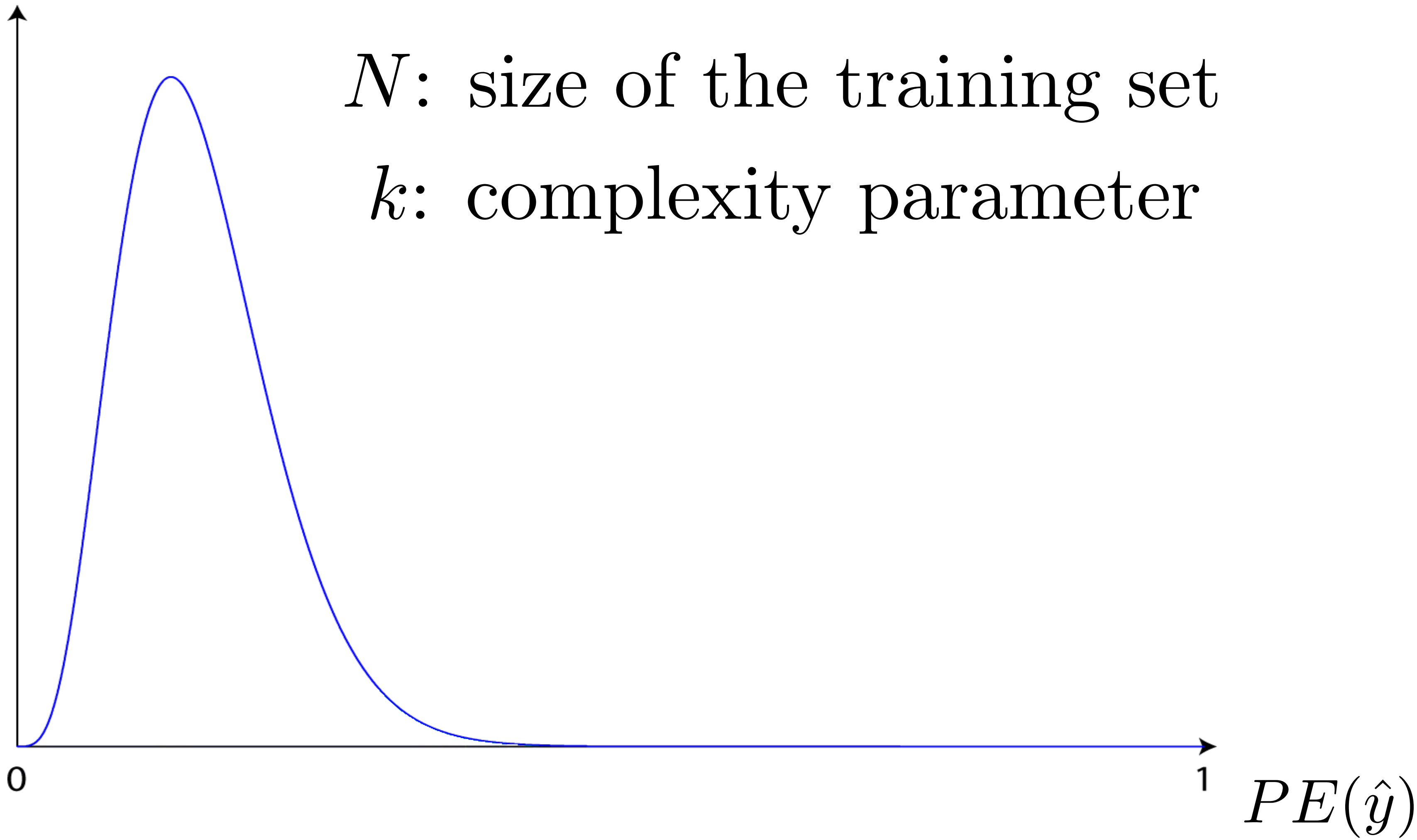
(distribution of the)

probability of error

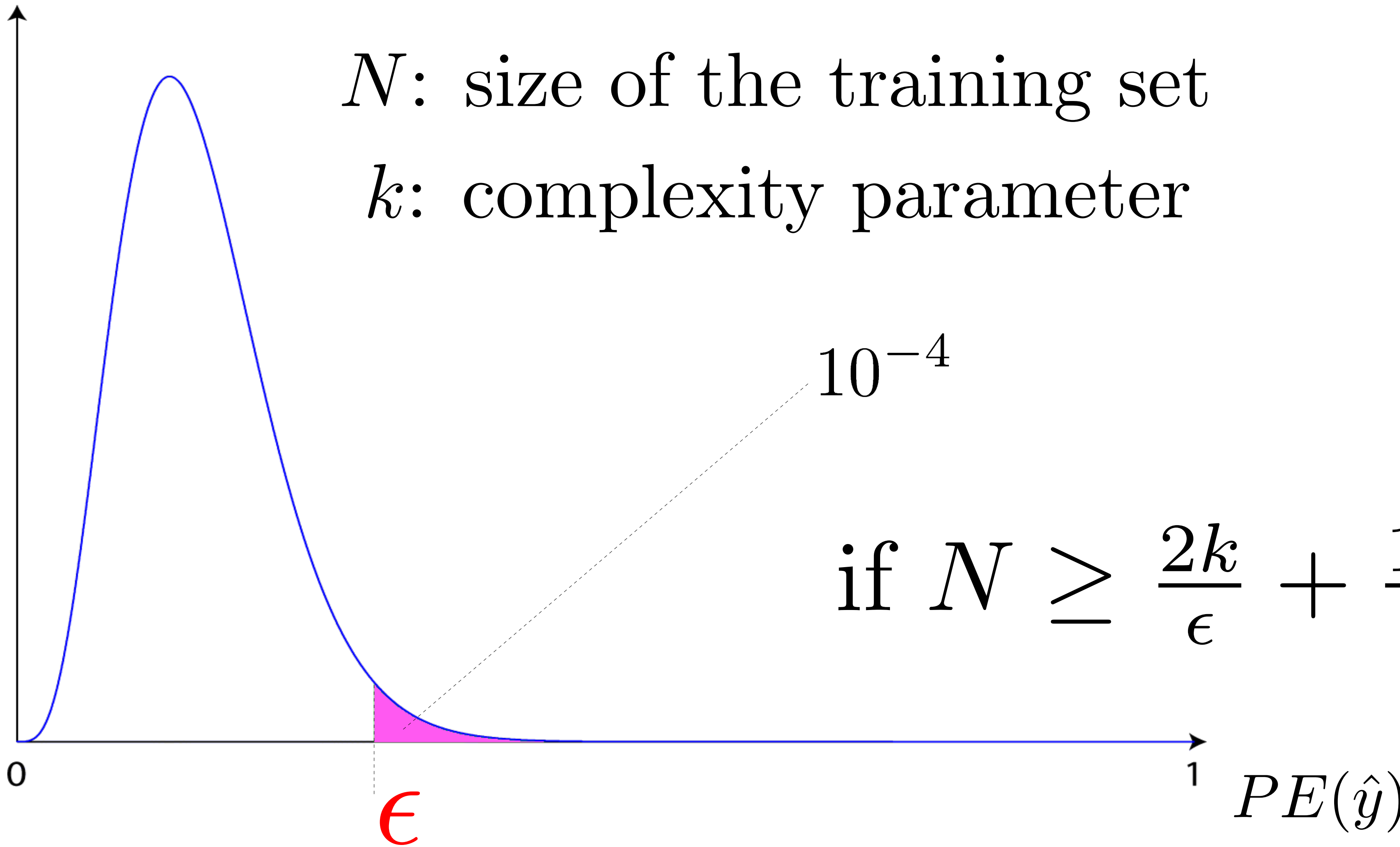
Classification with Guarantees

N: size of the training set

k: complexity parameter



Classification with Guarantees



N: size of the training set

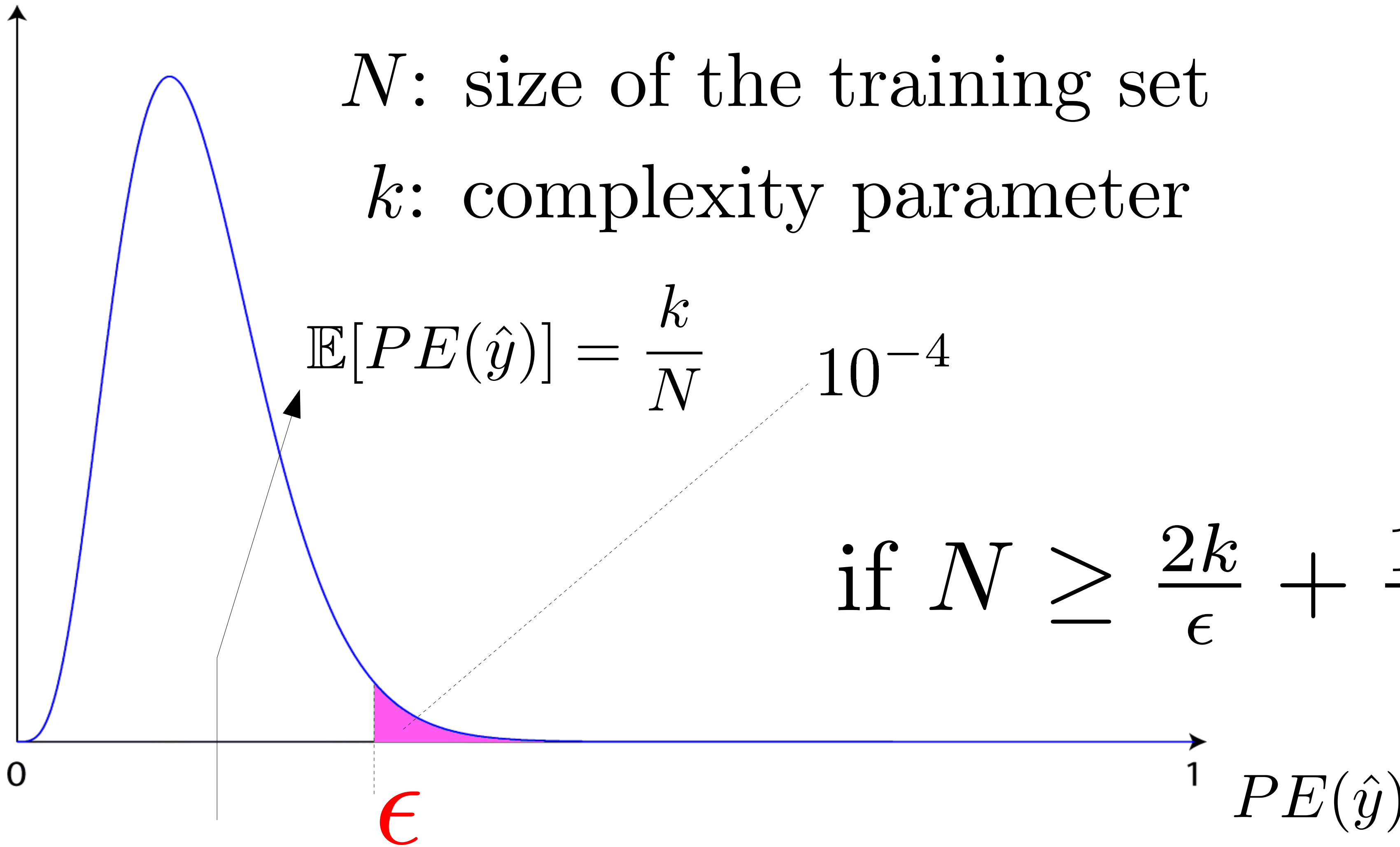
k: complexity parameter

$$\text{if } N \geq \frac{2k}{\epsilon} + \frac{19}{\epsilon}$$

Classification with Guarantees

N: size of the training set

k: complexity parameter



$$\text{if } N \geq \frac{2k}{\epsilon} + \frac{19}{\epsilon}$$

Alternative Approach: Complete Classification

IDEA:
Choose “k” large enough so that there are no “unknowns”.

Alternative Approach: Complete Classification

IDEA:
Choose “k” large enough so that there are no “unknowns”.

Such “k”

- carries strong information on the probability of error

Alternative Approach: Complete Classification

IDEA:
Choose “k” large enough so that there are no “unknowns”.

Such “k”

- carries strong information on the probability of error
- can be small even when features are many

Alternative Approach: Complete Classification

IDEA:
Choose “k” large enough so that there are no “unknowns”.

Such “k”

- carries strong information on the probability of error
- can be small even when features are many

Very technical caveat:
There is a subtle difference between trying with k=1,2,3,... and stopping when there are no “unknown”s and starting with k=∞ and evaluating the complexity a posteriori. The latter case is more difficult, see “**Wait-and-Judge scenario optimization**”

”Wait-and-Judge scenario optimization”
M.C. Campi, S.Garatti
Mathematical Programming, 2019

A fundamental distinction

Probability of **error**

A fundamental distinction

Probability of **error**

Pr("success" | failure)

Pr("failure" | success)

A fundamental distinction

Probability of **correct classification**

1 – Pr(”success” | failure)

1 – Pr(”failure” | success)

A fundamental distinction

Probability of **correct classification**

1 – Pr(”success” | failure)
=Pr(”failure” | failure)

1 – Pr(”failure” | success)
=Pr(”success” | success)

A fundamental distinction

Probability of **correct classification**

1 – Pr("success" | failure)
=Pr("failure" | failure)

specificity

1 – Pr("failure" | success)
=Pr("success" | success)

sensitivity

A fundamental distinction

Probability of **correct classification**

$1 - \Pr(\text{"success"} \mid \text{failure})$
$= \Pr(\text{"failure"} \mid \text{failure})$

specificity

Target

50%

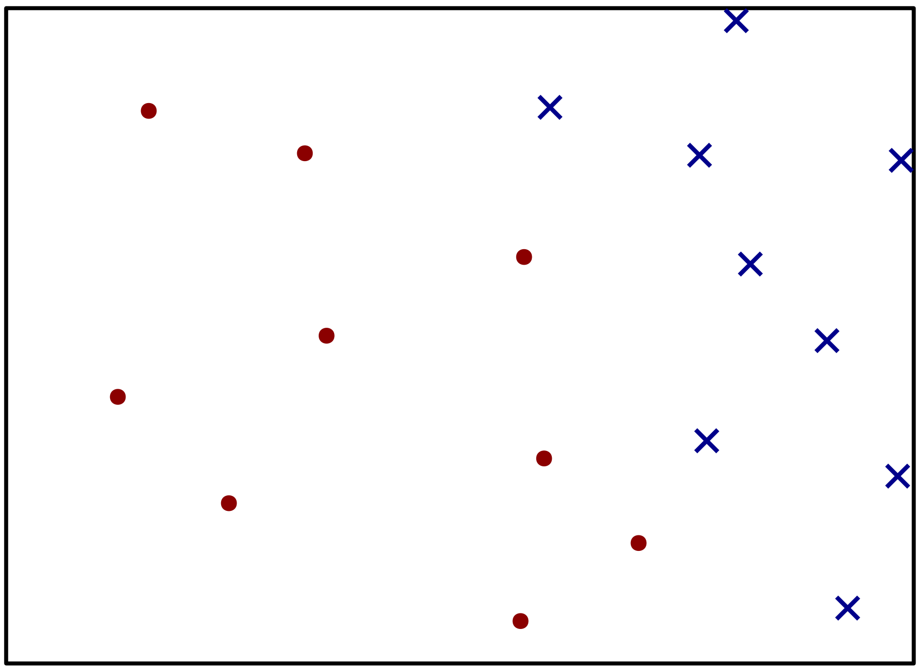
$1 - \Pr(\text{"failure"} \mid \text{success})$
$= \Pr(\text{"success"} \mid \text{success})$

sensitivity

Target

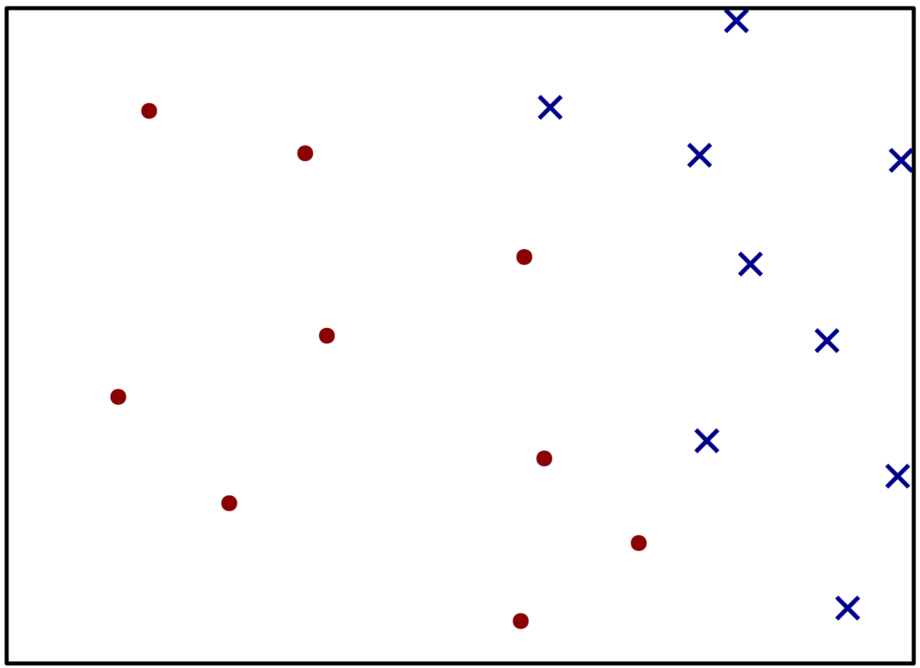
95%

GEM-BALLS



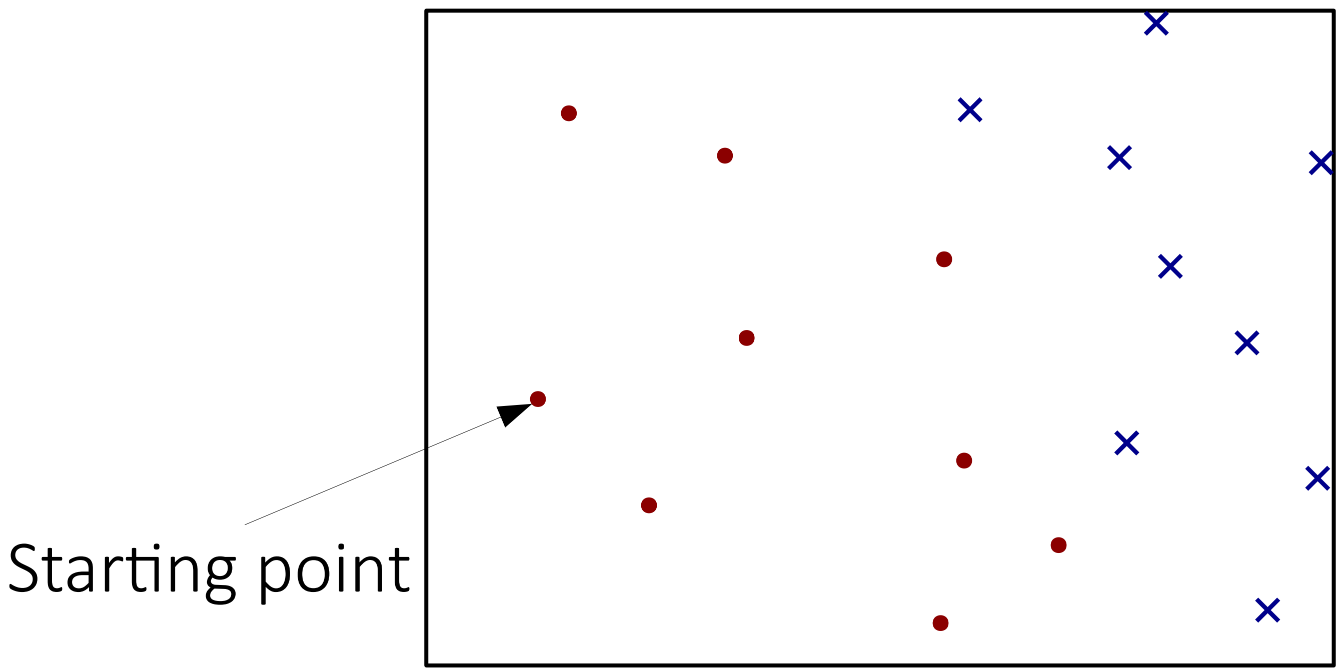
GEM-BALLS

- “0”, “negative”, “failure”
- × “1”, “positive”, “success”



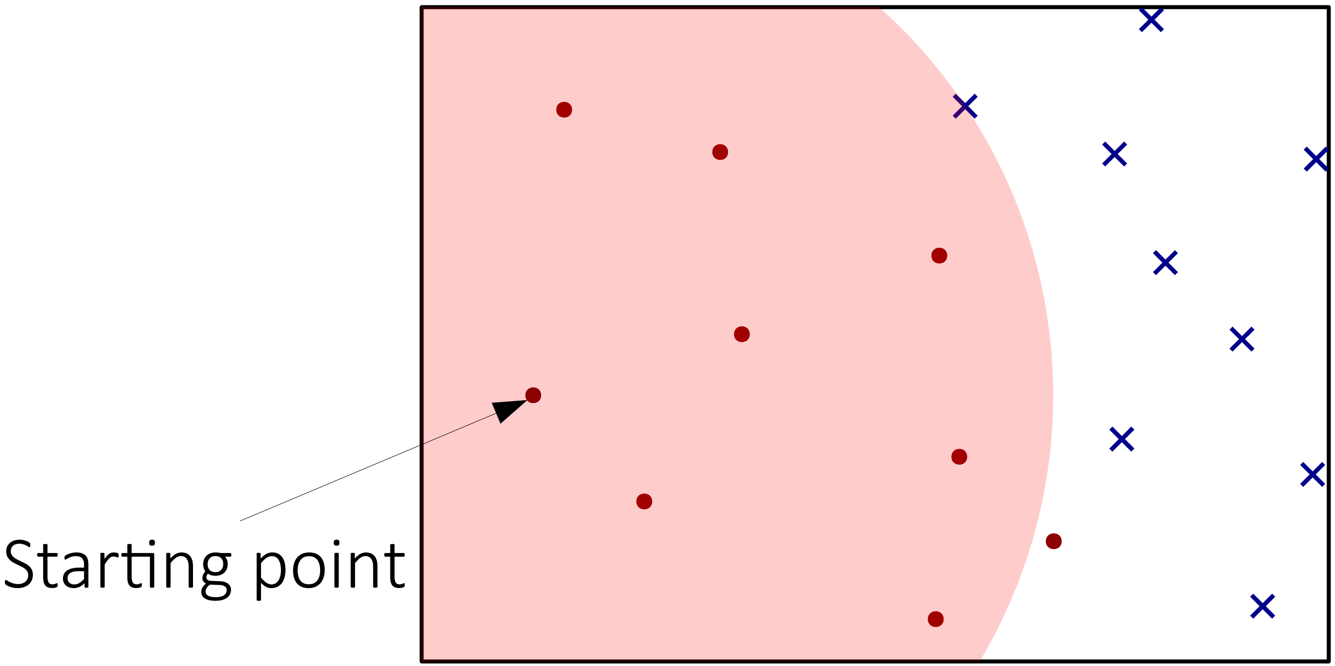
GEM-BALLS

- “0”, “negative”, “failure”
- × “1”, “positive”, “success”

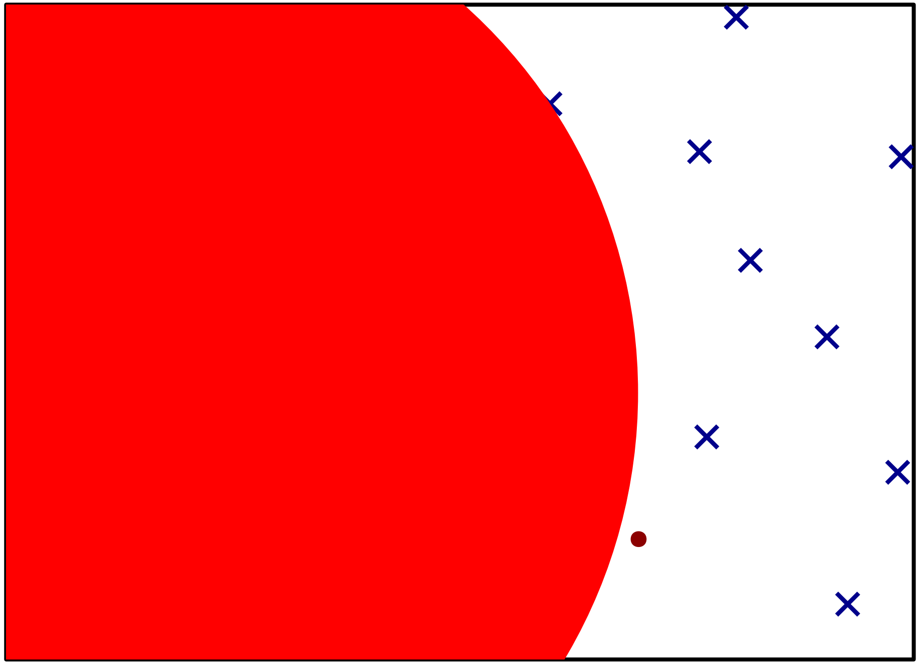


GEM-BALLS

Largest ball that does not include blue points

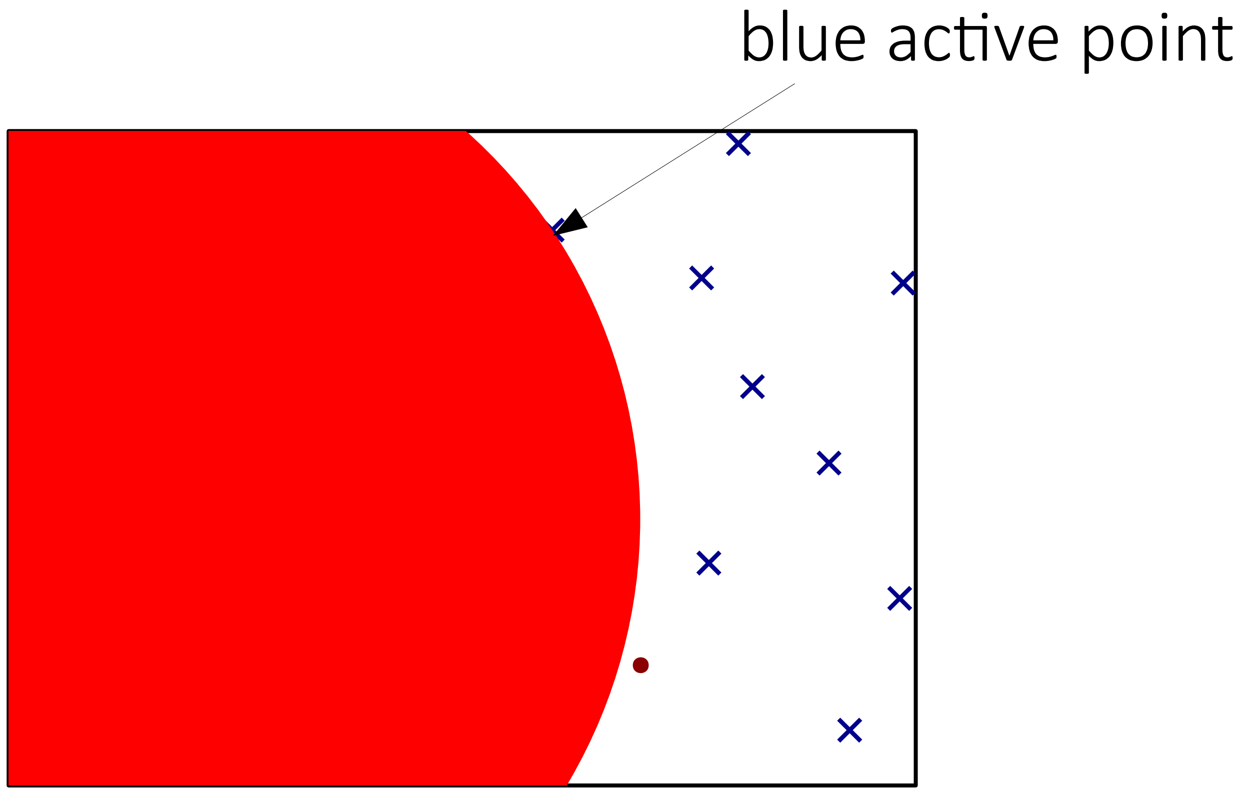


GEM-BALLS



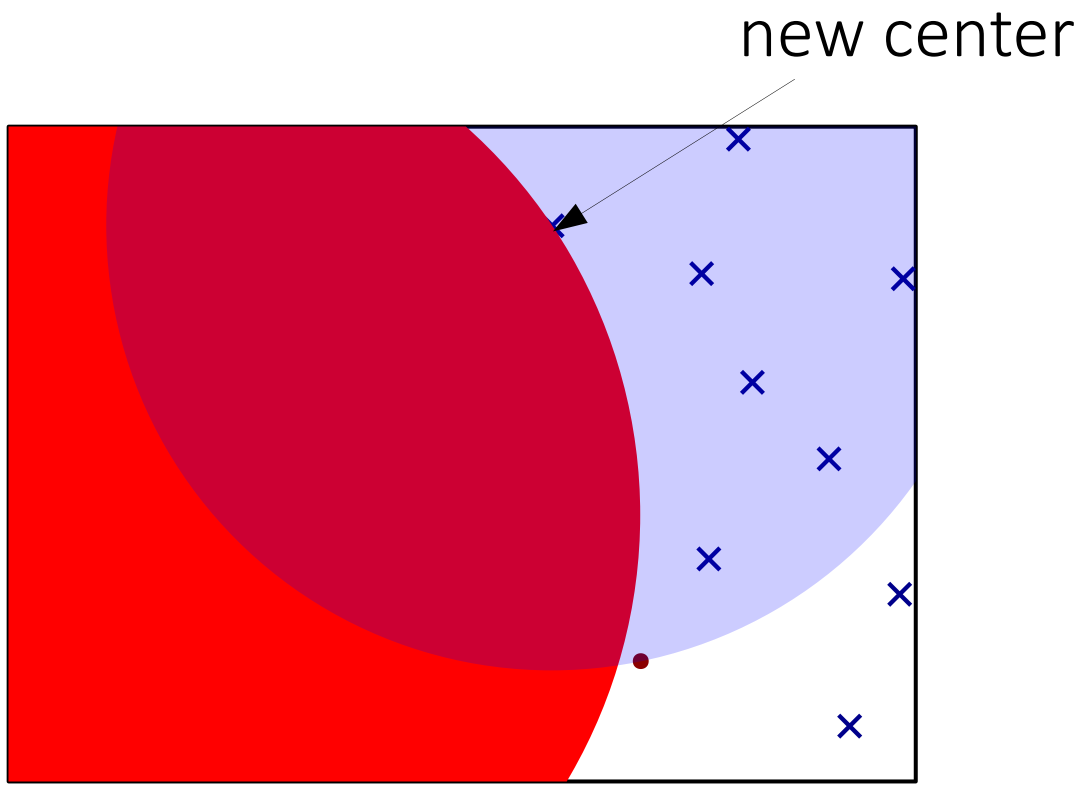
GEM-BALLS

#blue active points so far =1



GEM-BALLS

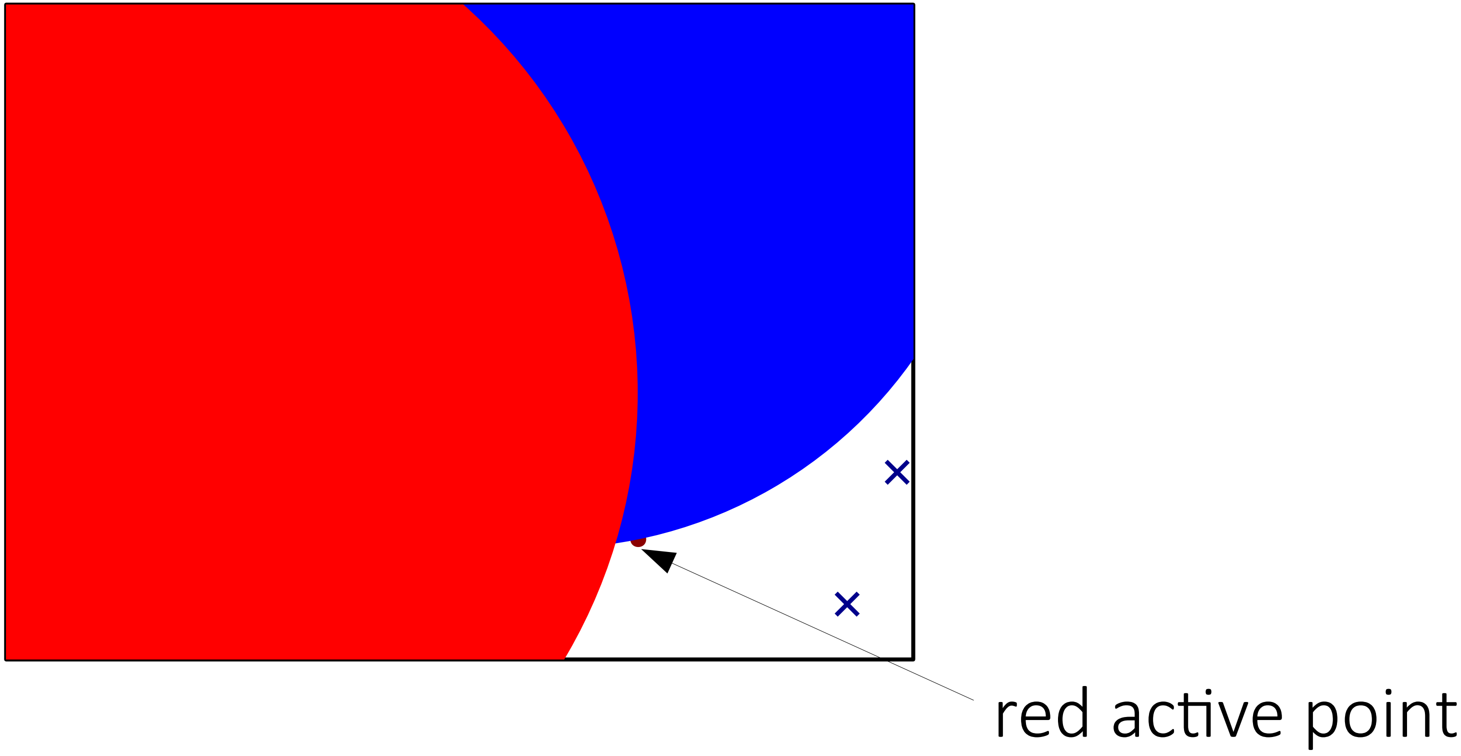
#blue active points so far =1



GEM-BALLS

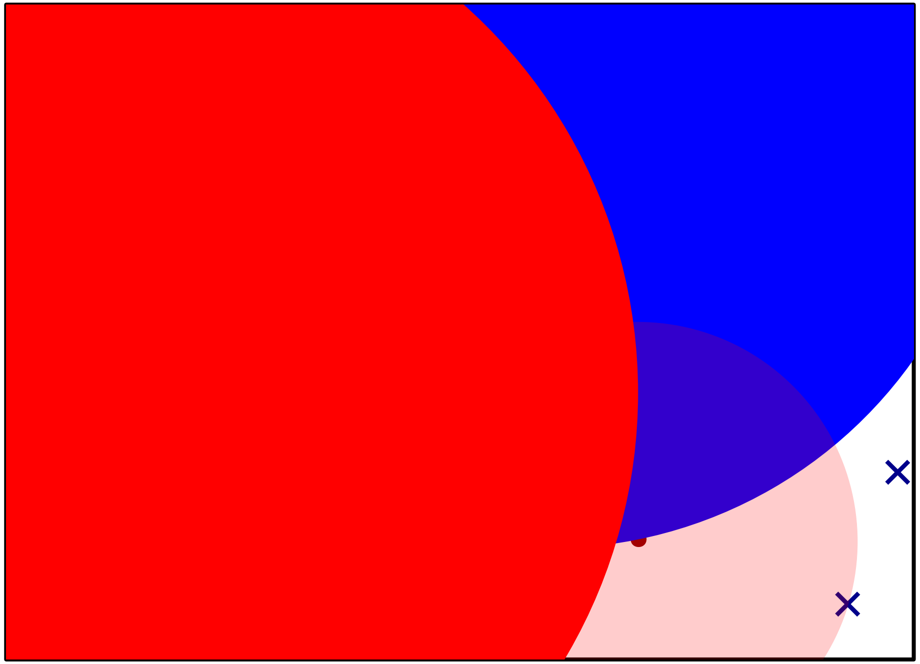
#blue active points so far =1

#red active points so far =1



GEM-BALLS

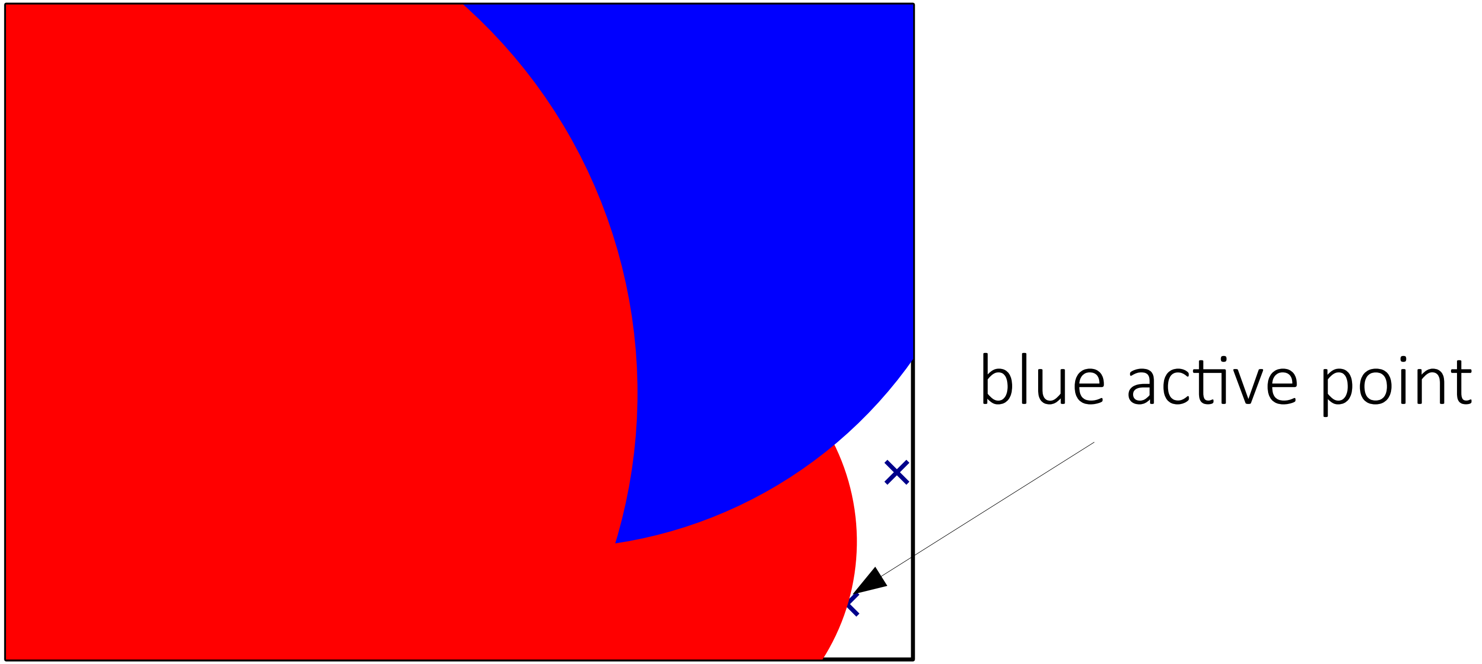
#blue active points so far =1
#red active points so far =1



GEM-BALLS

#blue active points so far =1+1

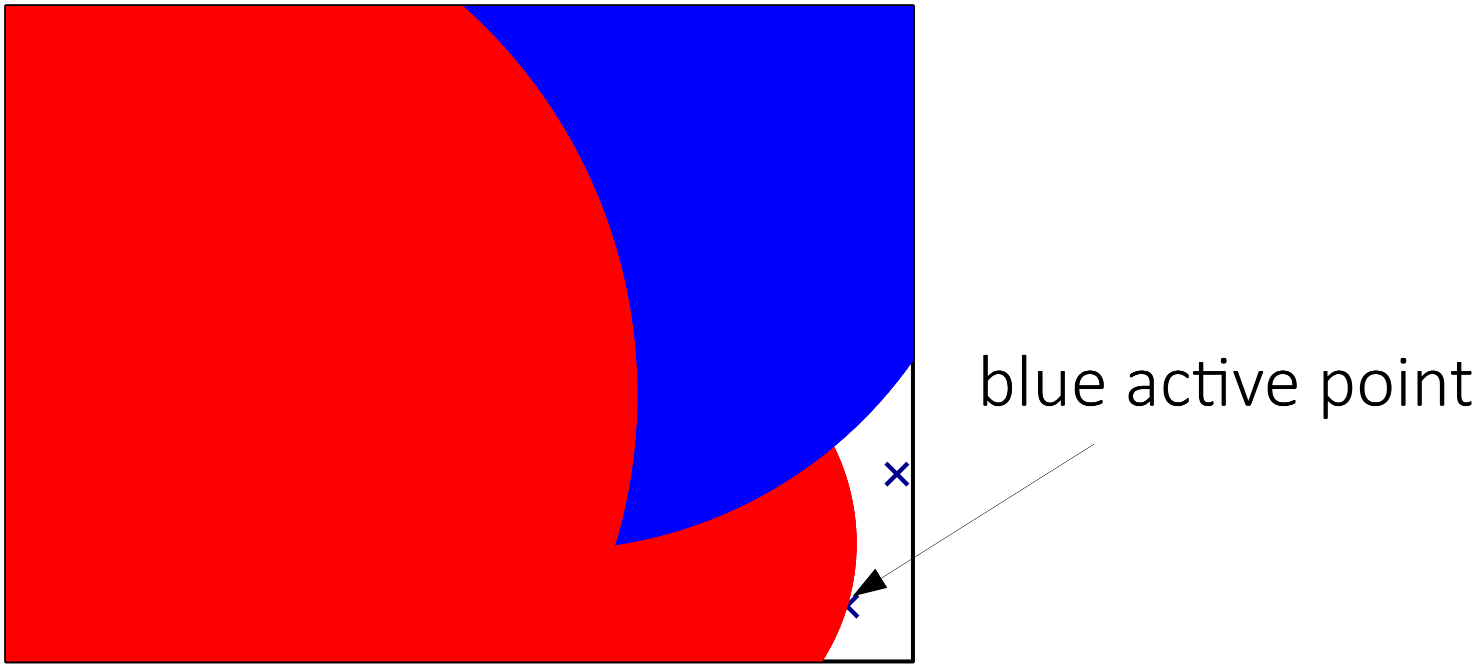
#red active points so far =1



GEM-BALLS

#blue active points so far =2

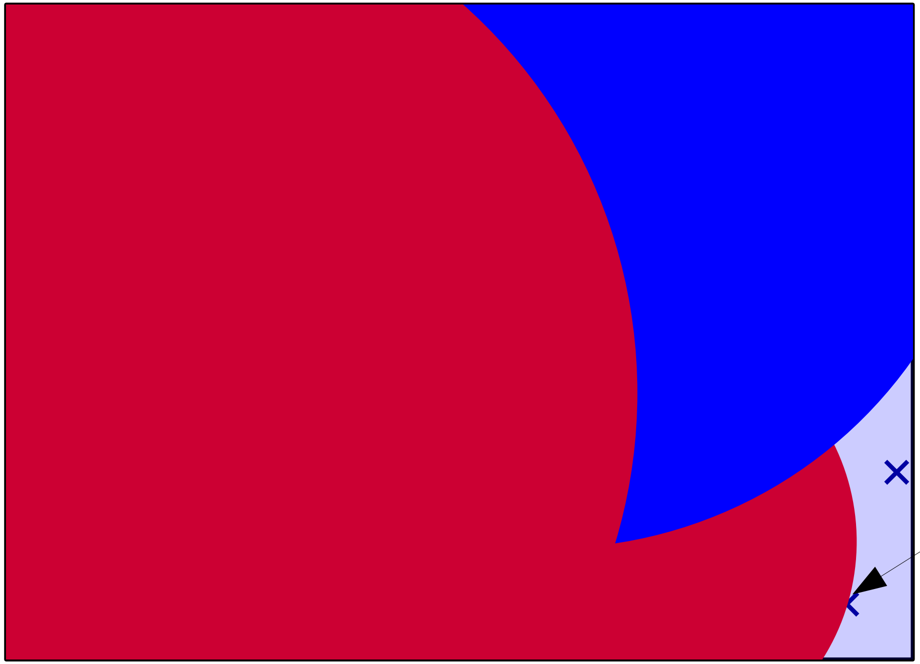
#red active points so far =1



GEM-BALLS

#blue active points so far =2

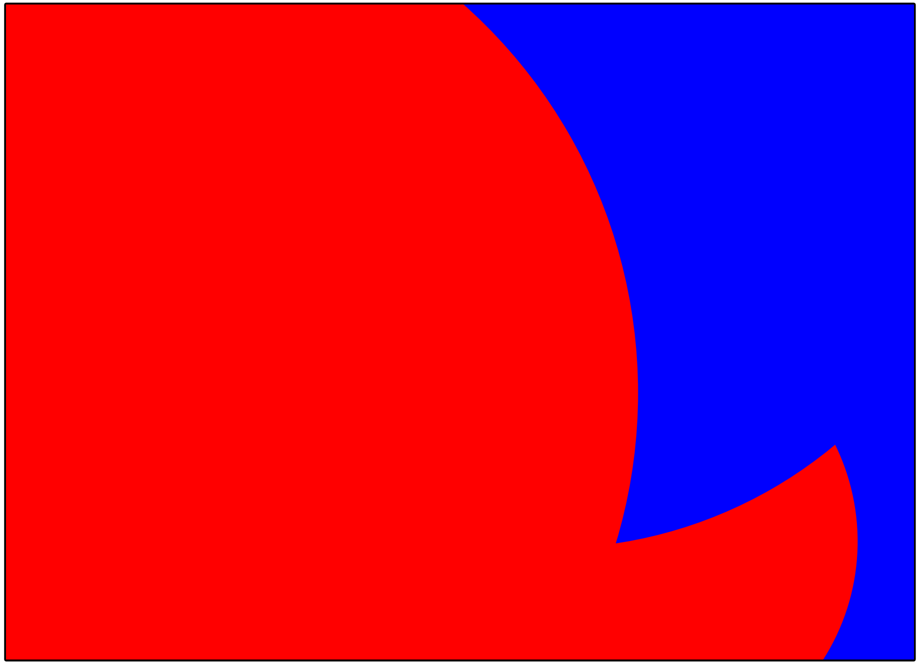
#red active points so far =1



new center

GEM-BALLS

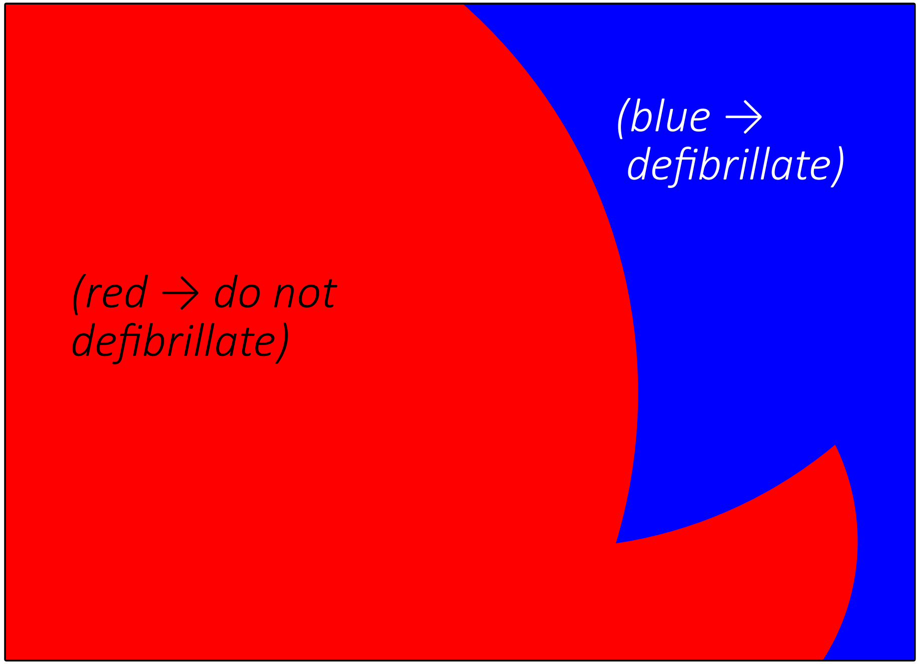
#blue active points so far =2
#red active points so far =1



GEM-BALLS

#blue active points so far =2 → k_1

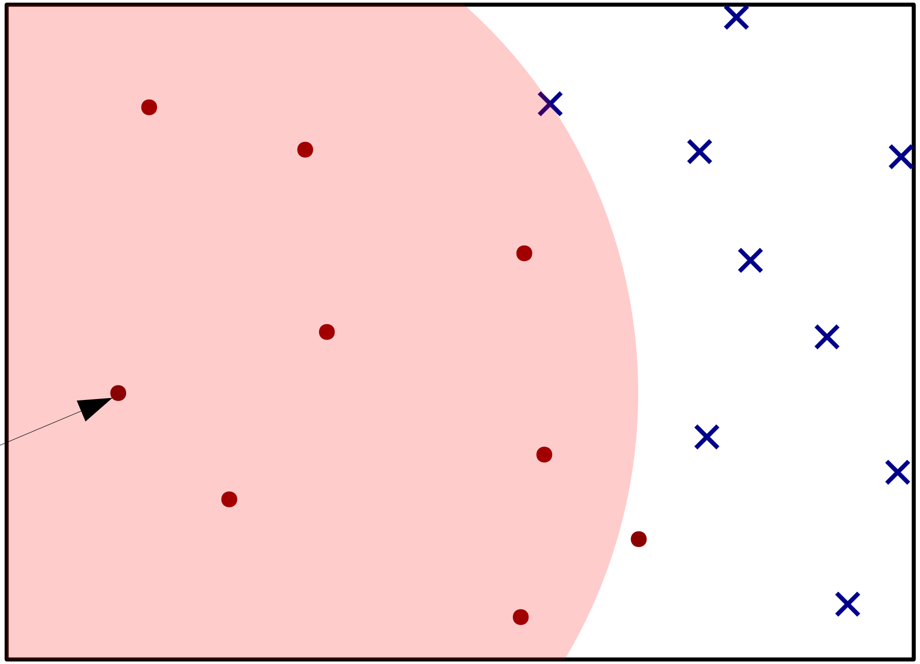
#red active points so far =1 → k_0



The tuning knob

When $C_1=1$

Largest ball that does not include blue points

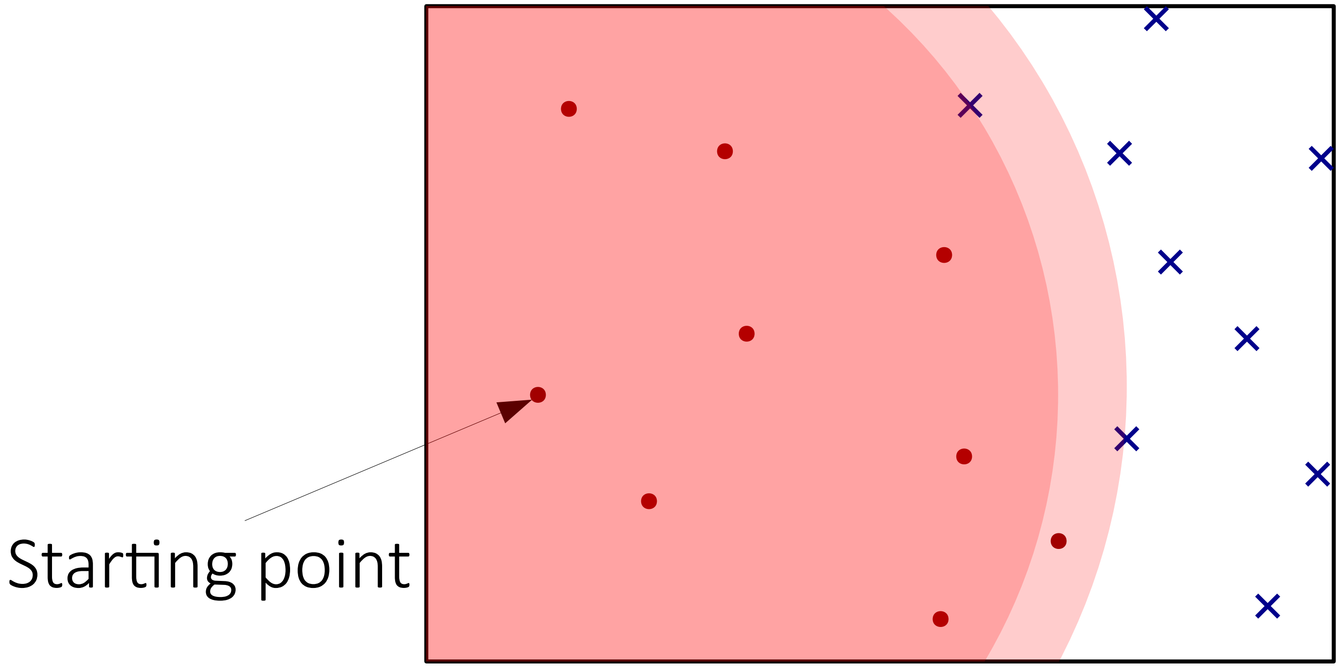


Starting point

The tuning knob

When $C_1=2$

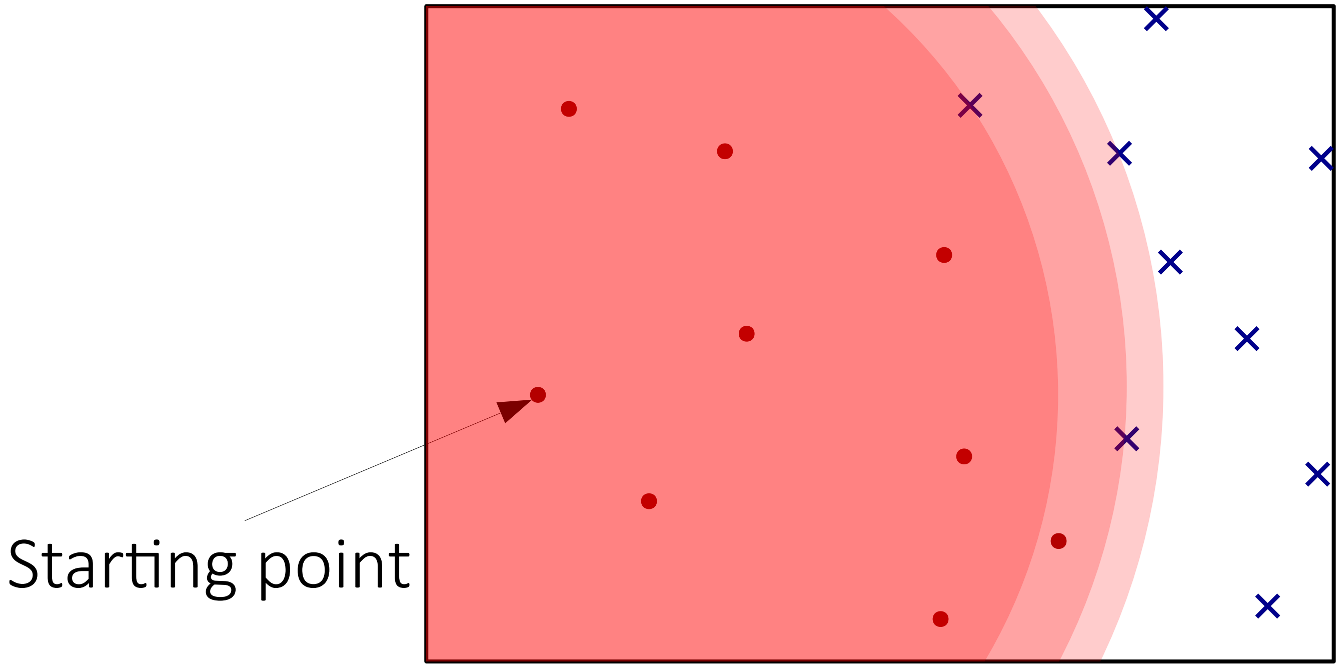
Largest ball that includes less than **2** blue points



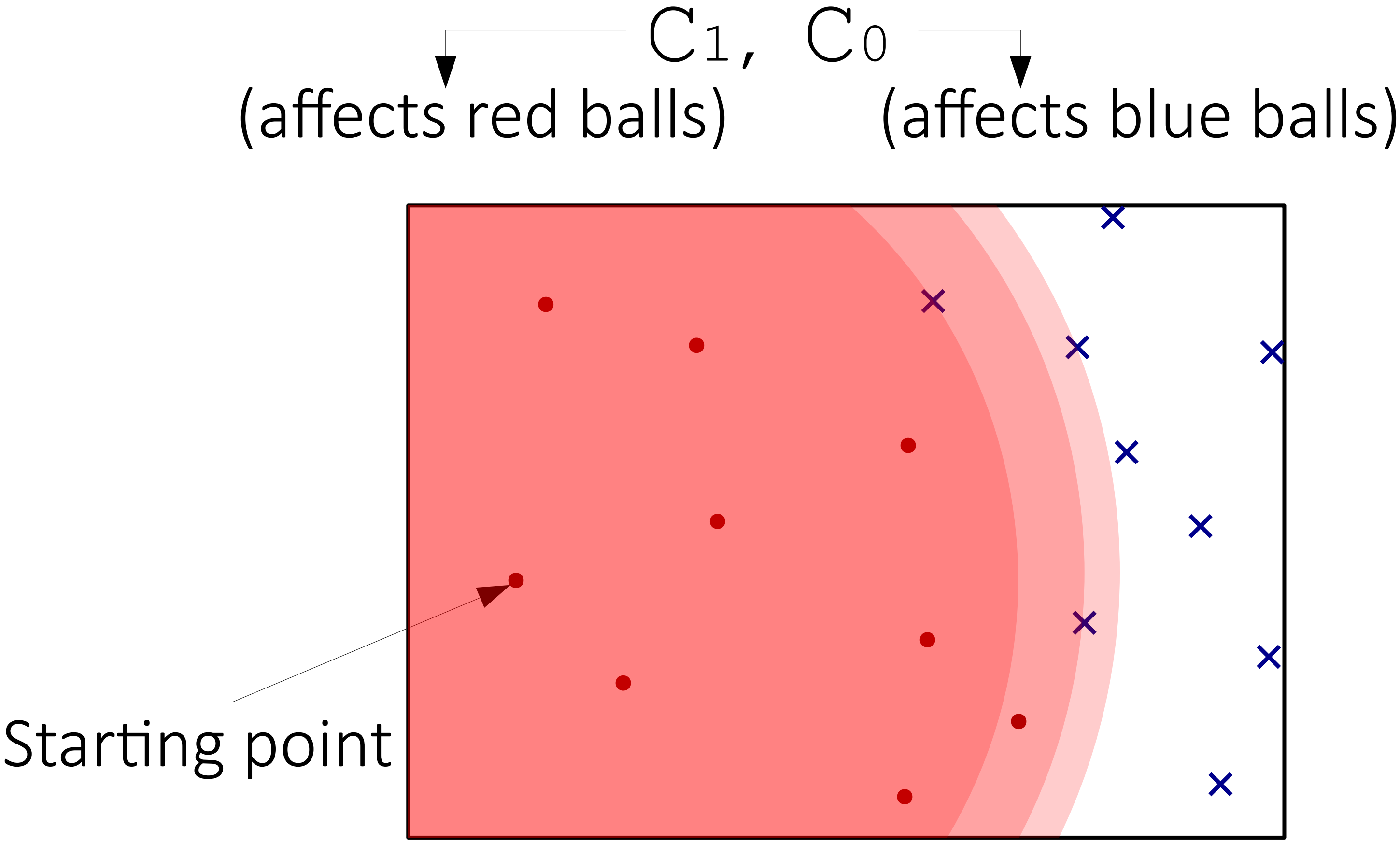
The tuning knob

When $C_1=3$

Largest ball that includes less than **3** blue points



The tuning knob



Theorem - preliminaries

β: small (e.g. 10^{-6}) confidence parameter

Theorem - preliminaries

β: small (e.g. 10^{-6}) confidence parameter

N_1: # of positives

Theorem - preliminaries

β: small (e.g. 10^{-6}) confidence parameter

N_1: # of positives

k_1 : # active positives

Theorem - preliminaries

β: small (e.g. 10^{-6}) confidence parameter

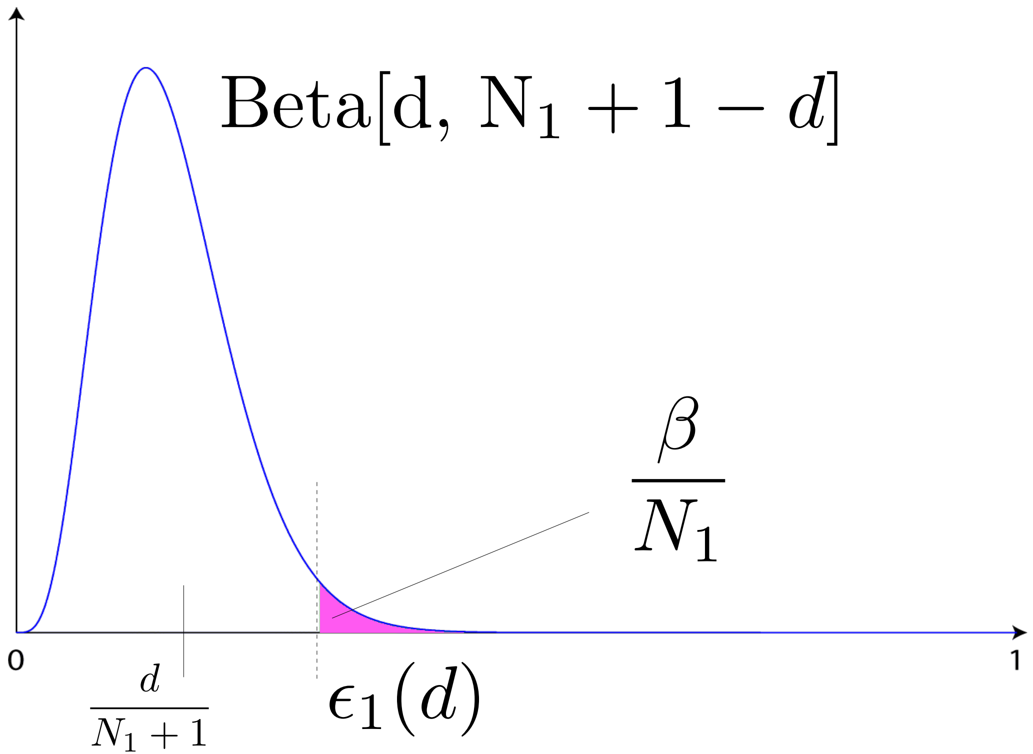
N_1: # of positives

k_1 : # active positives

$\{\epsilon_1(d) : d = 1, \ldots, N_1\}$: are pre-computed thresholds.

Theorem - preliminaries

$\epsilon_1(d)$: for $d = 1, \ldots, N_1$ is the quantile at level $1 - \frac{\beta}{N_1}$ of $\mathrm{Beta}[\mathrm{d},\ \mathrm{N}_1 + 1 - d]$



Theorem - preliminaries

$\hat{y}$ (k_1)

Theorem - preliminaries

$$\mathbb{P}\{\underbrace{\hat{y}(x) = 0}_{\text{“failure”}} \mid \underbrace{y = 1}_{\text{success}}\} \qquad (k_1)$$

Theorem - preliminaries

$$\mathbb{P}\{\overbrace{\hat{y}(x) = 0}^{\text{“failure”}} \mid \overbrace{y = 1}^{\text{success}}\} \qquad \underbrace{\epsilon_1(k_1)}_{\text{precomputed threshold}}$$

Theorem - preliminaries

$$\mathbb{P}\{\overbrace{\hat{y}(x) = 0}^{\text{"failure"}} \mid \overbrace{y = 1}^{\text{success}}\} \leq \underbrace{\epsilon_1(k_1)}_{\text{precomputed threshold}}$$

Theorem - Statement

$$\mathbb{P}\{\overbrace{\hat{y}(x) = 0}^{\text{“failure”}} \mid \overbrace{y = 1}^{\text{success}}\} \leq \underbrace{\epsilon_1(k_1)}_{\text{precomputed threshold}}$$

$$\text{with confidence } 1 - \beta$$

Theorem - Discussion

$$\mathbb{P}\{\underbrace{\hat{y}(x) = 0}^{\text{"failure"}} \mid \underbrace{y = 1}^{\text{success}}\} \leq \underbrace{\epsilon_1(k_1)}_{\text{precomputed threshold}}$$

$\Updownarrow$

$$\text{sensitivity} \geq 1 - \epsilon_1(k_1)$$

$$\text{with confidence } 1 - \beta$$

Theorem - Discussion

$$\mathbb{P}\{\overbrace{\hat{y}(x) = 0}^{\text{“failure”}} \mid \overbrace{y = 1}^{\text{success}}\} \leq \underbrace{\epsilon_1(k_1)}_{\text{precomputed threshold}}$$

$$\text{sensitivity} \geq 1 - \epsilon_1(k_1)$$

$$\text{with confidence } 1 - \beta$$

Theorem - Statement

$$\text{sensitivity} \geq 1 - \epsilon_1(k_1)$$

$$\text{with confidence } 1 - \beta$$

Theorem - Discussion

$$\text{sensitivity} \geq 1 - \epsilon_1(k_1)$$

$$\text{with confidence } 1 - \beta$$

Theorem - Discussion

$$(x_i, y_i) \sim \mathbb{P}$$

$$\text{sensitivity} \geq 1 - \epsilon_1(k_1)$$

Theorem - Statement

$$(x_i, y_i) \sim \mathbb{P}$$

$$\mathbb{P}^N\{\text{sensitivity} \geq 1 - \epsilon_1(k_1)\} \geq 1 - \beta$$

Theorem - Statement

$$(x_i, y_i) \sim \mathbb{P}$$

$$\mathbb{P}^N\{\text{sensitivity} \geq 1 - \epsilon_1(k_1)\} \geq 1 - \beta$$

$$\mathbb{P}^N\{\text{specificity} \geq 1 - \epsilon_0(k_0)\} \geq 1 - \beta$$

Theorem - Statement

$$\text{sensitivity} \geq 1 - \epsilon_1(k_1) \;\&\; \text{specificity} \geq 1 - \epsilon_0(k_0)$$

$$\text{with overall confidence } 1 - 2\beta$$

Theorem - Statement

$$\text{sensitivity} \geq 1 - \epsilon_1(k_1) \ \& \ \text{specificity} \geq 1 - \epsilon_0(k_0)$$

$$\text{with overall confidence } 1 - 2\beta$$

Take-home message:

“GEM-BALLS is a self-testing algorithm!”

Benchmark datasets

BreastW (239 positive instances, 444 negative instances), $\beta = 5 \cdot 10^{-3}$							
$c_1 : c_0$	$1:1$	$1:2$	$1:3$	$1:5$	$1:10$	$10:100$	$1:50$
$\mathbf{k}_1 : \mathbf{k}_0$	$17:17$	$11:21$	$10:28$	$6:28$	$4:33$	$10:17$	$2:66$
$Sens$:$Spec$	83% : 91%	87% : 89%	88% : 87%	90% : 87%	92% : 86%	88% : 91%	94% : 76%

Pima (268 positive instances, 500 negative instances), $\beta = 10^{-3}$							
$c_1 : c_0$	$1:1$	$1:2$	$2:4$	$1:4$	$1:8$	$1:10$	$1:50$
$\mathbf{k}_1 : \mathbf{k}_0$	$125:125$	$88:175$	$96:189$	$61:245$	$38:300$	$33:324$	$9:424$
$Sens$:$Spec$	40% : 65%	54% : 55%	51% : 52%	65% : 41%	75% : 30%	70% : 20%	90% : 9%

Haberman (75 positive instances, 219 negative instances), $\beta = 10^{-3}$					
$c_1 : c_0$	$1:1$	$1:3$	$1:5$	$1:10$	$1:20$
$\mathbf{k}_1 : \mathbf{k}_0$	$46:46$	$28:82$	$23:114$	$14:139$	$9:176$
$Sens$:$Spec$	20% : 67%	41% : 49%	48% : 35%	62% : 24%	71% : 10%

Benchmark datasets

Breast W (239 positive instances, 444 negative instances), $\beta = 5 \cdot 10^{-3}$							
$c_1 : c_0$	$1:1$	$1:2$	$1:3$	$1:5$	$1:10$	$10:100$	$1:50$
$\mathbf{k}_1 : \mathbf{k}_0$	$17:17$	$11:21$	$10:28$	$6:28$	$4:33$	$10:17$	$2:66$
$Sens{:}Spec$	$83\% : 91\%$	$87\% : 89\%$	$88\% : 87\%$	$90\% : 87\%$	$92\% : 86\%$	$88\% : 91\%$	$94\% : 76\%$

Pima (268 positive instances, 500 negative instances), $\beta = 10^{-3}$							
$c_1 : c_0$	$1:1$	$1:2$	$2:4$	$1:4$	$1:8$	$1:10$	$1:50$
$\mathbf{k}_1 : \mathbf{k}_0$	$125:125$	$88:175$	$96:189$	$61:245$	$38:300$	$33:324$	$9:424$
$Sens{:}Spec$	$40\% : 65\%$	$54\% : 55\%$	$51\% : 52\%$	$65\% : 41\%$	$75\% : 30\%$	$70\% : 20\%$	$90\% : 9\%$

Haberman (75 positive instances, 219 negative instances), $\beta = 10^{-3}$					
$c_1 : c_0$	$1:1$	$1:3$	$1:5$	$1:10$	$1:20$
$\mathbf{k}_1 : \mathbf{k}_0$	$46:46$	$28:82$	$23:114$	$14:139$	$9:176$
$Sens{:}Spec$	$20\% : 67\%$	$41\% : 49\%$	$48\% : 35\%$	$62\% : 24\%$	$71\% : 10\%$

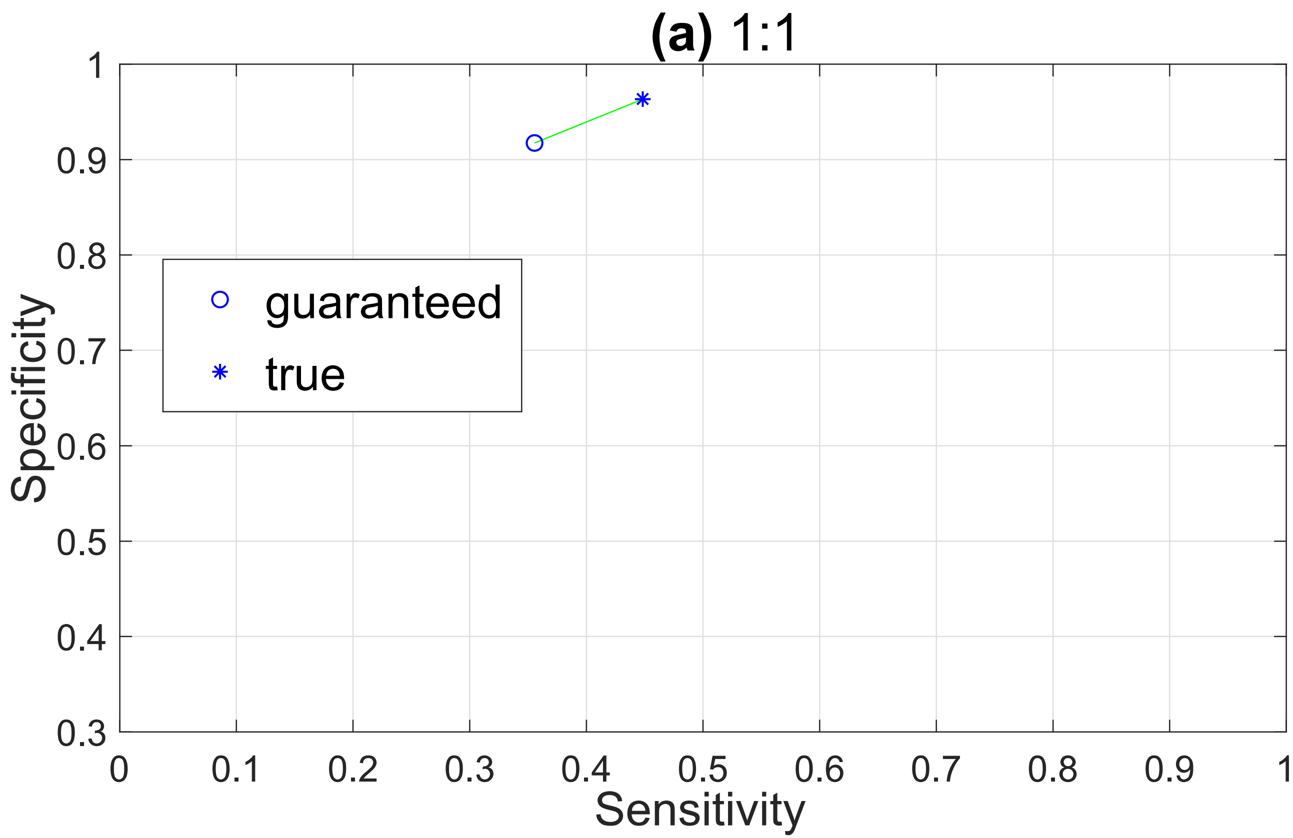
Benchmark datasets

BreastW (239 positive instances, 444 negative instances), $\beta = 5 \cdot 10^{-3}$							
$c_1 : c_0$	$1:1$	$1:2$	$1:3$	$1:5$	$1:10$	$10:100$	$1:50$
$\mathbf{k}_1 : \mathbf{k}_0$	$17:17$	$11:21$	$10:28$	$6:28$	$4:33$	$10:17$	$2:66$
$Sens$:$Spec$	83% : 91%	87% : 89%	88% : 87%	90% : 87%	92% : 86%	88% : 91%	94% : 76%

Pima (268 positive instances, 500 negative instances), $\beta = 10^{-3}$							
$c_1 : c_0$	$1:1$	$1:2$	$2:4$	$1:4$	$1:8$	$1:10$	$1:50$
$\mathbf{k}_1 : \mathbf{k}_0$	$125:125$	$88:175$	$96:189$	$61:245$	$38:300$	$33:324$	$9:424$
$Sens$:$Spec$	40% : 65%	54% : 55%	51% : 52%	65% : 41%	75% : 30%	70% : 20%	90% : 9%

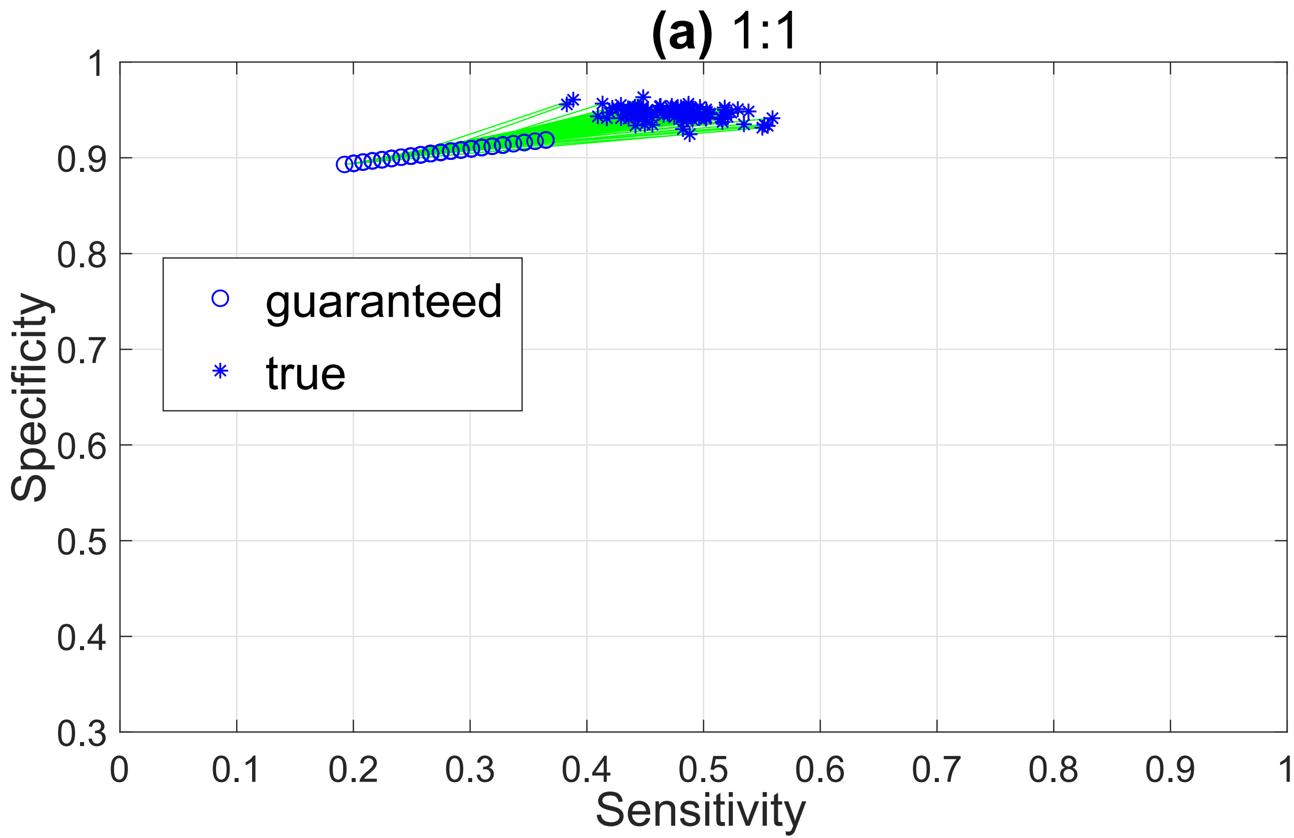
Haberman (75 positive instances, 219 negative instances), $\beta = 10^{-3}$					
$c_1 : c_0$	$1:1$	$1:3$	$1:5$	$1:10$	$1:20$
$\mathbf{k}_1 : \mathbf{k}_0$	$46:46$	$28:82$	$23:114$	$14:139$	$9:176$
$Sens$:$Spec$	20% : 67%	41% : 49%	48% : 35%	62% : 24%	71% : 10%

Tests on reproducible simulated data



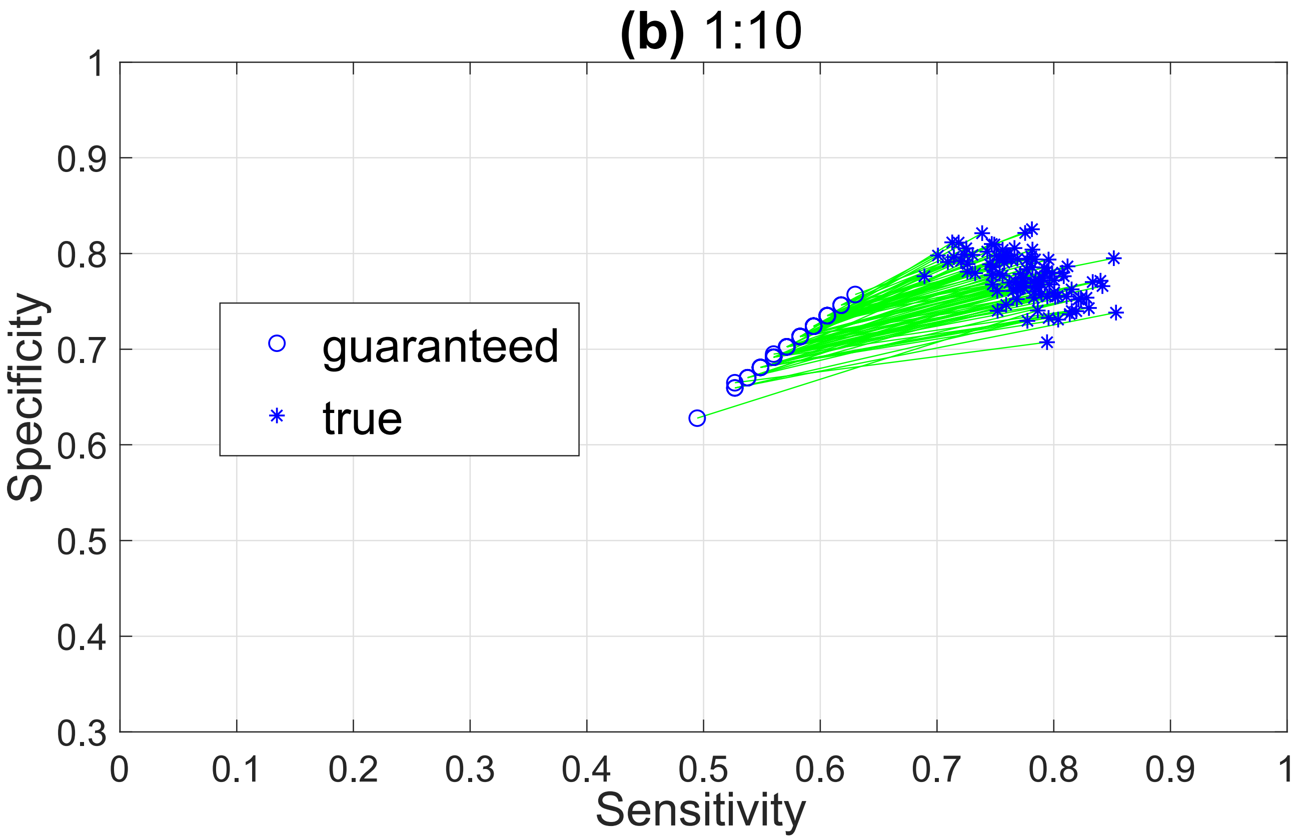
Labels are generated by using the MATLAB kstest function.
$N_0 = 1000,\ N_1 = 100.\ \beta = 10^{-3}$

Tests on reproducible simulated data



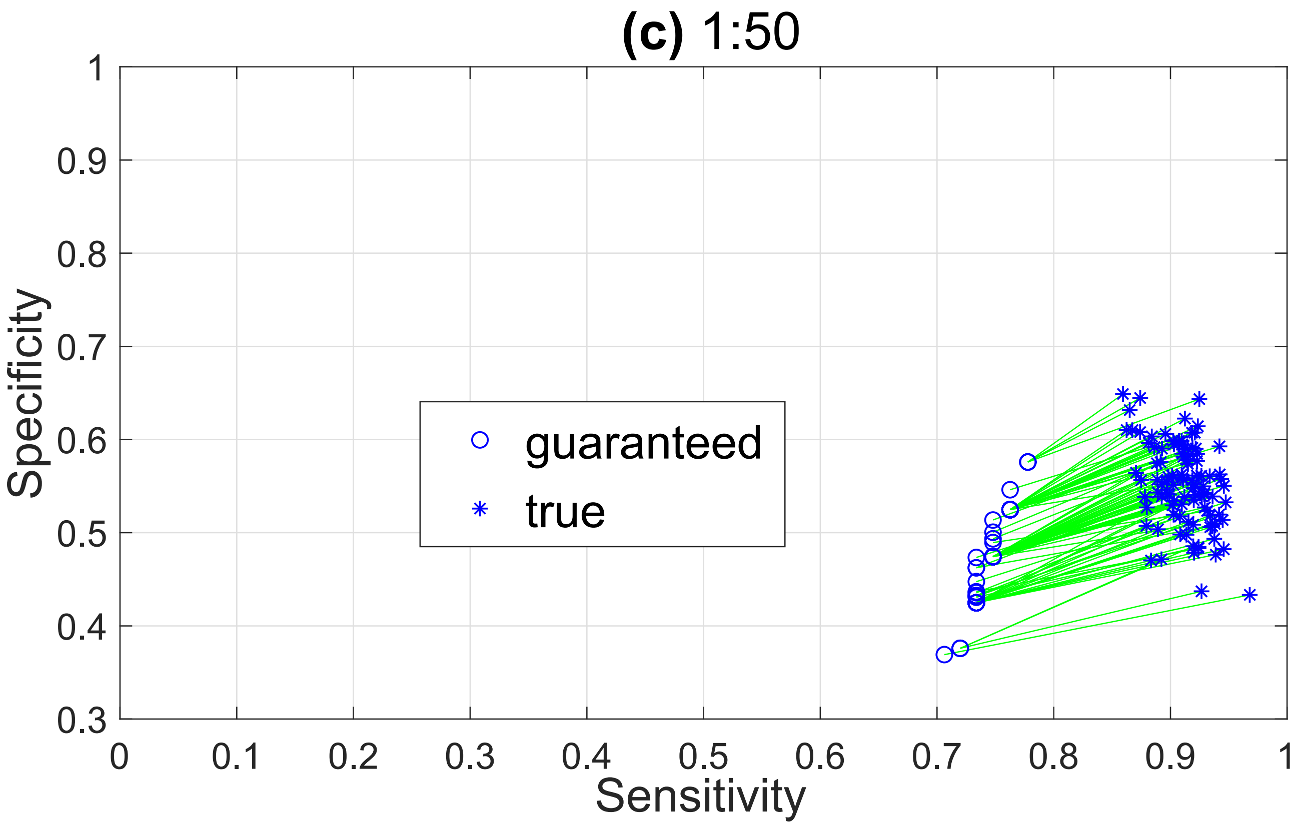
Labels are generated by using the MATLAB kstest function.
$N_0 = 1000$, $N_1 = 100$. $\beta = 10^{-3}$

Tests on reproducible simulated data



Labels are generated by using the MATLAB kstest function.
$N_0 = 1000$, $N_1 = 100$. $\beta = 10^{-3}$

Tests on reproducible simulated data



Labels are generated by using the MATLAB kstest function.
$N_0 = 1000,\ N_1 = 100.\ \beta = 10^{-3}$

Ventricular Fibrillation (VF) dataset

VF dataset (15 pos, 155 neg), $\beta = 10^{-2}$			
$c_1 : c_0$	$1 : 1$	$1 : 10$	$1 : 80$
$\mathbf{k}_1 : \mathbf{k}_0$	$9 : 9$	$5 : 41$	$2 : 90$
$Sens : Spec$	$11\% : 85\%$	$30\% : 59\%$	$51\% : 28\%$

Ventricular Fibrillation (VF) dataset

VF dataset (15 pos, 155 neg), $\beta = 10^{-2}$			
$c_1 : c_0$	$1:1$	$1:10$	$1:80$
$\mathbf{k}_1 : \mathbf{k}_0$	$9:9$	$5:41$	$2:90$
$Sens : Spec$	$11\% : 85\%$	$30\% : 59\%$	$51\% : 28\%$

Expanded VF dataset (240 pos, 2477 neg), $\beta = 10^{-3}$				
$c_1 : c_0$	$1:1$	$1:10$	$1:80$	$1:240$
$\mathbf{k}_1 : \mathbf{k}_0$	$16:16$	$1:121$	$8:568$	$4:1055$
$Sens : Spec$	$84\% : 98\%$	$86\% : 93\%$	$89\% : 73\%$	$92\% : 53\%$

SMOTE (Synthetic Minority Over-sampling Technique)

New lines of research

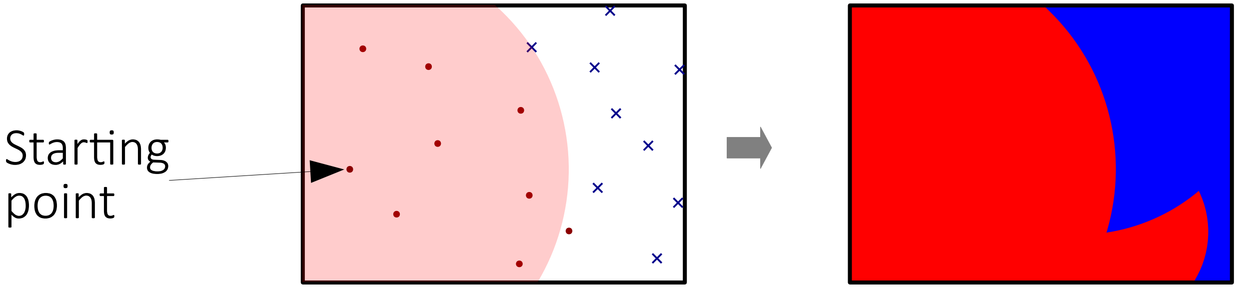
Pool of guaranteed classifiers:

Train many guaranteed classifiers simultaneusly.

New lines of research

Pool of guaranteed classifiers:

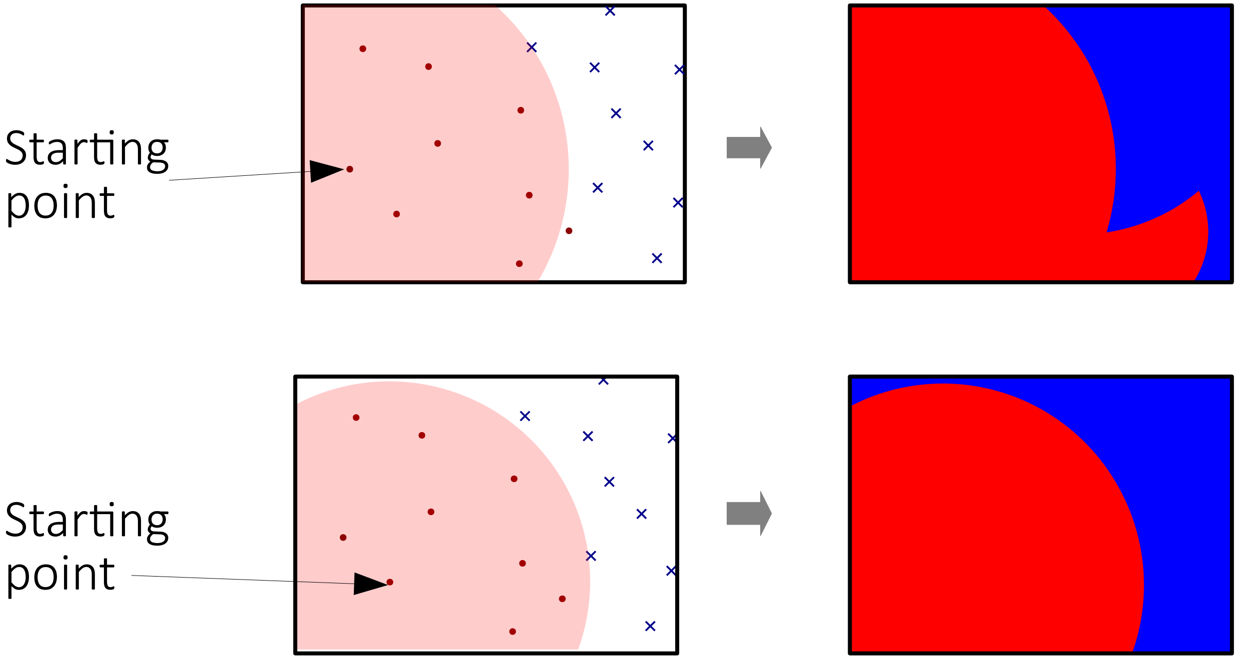
Train many guaranteed classifiers simultaneusly.



New lines of research

Pool of guaranteed classifiers:

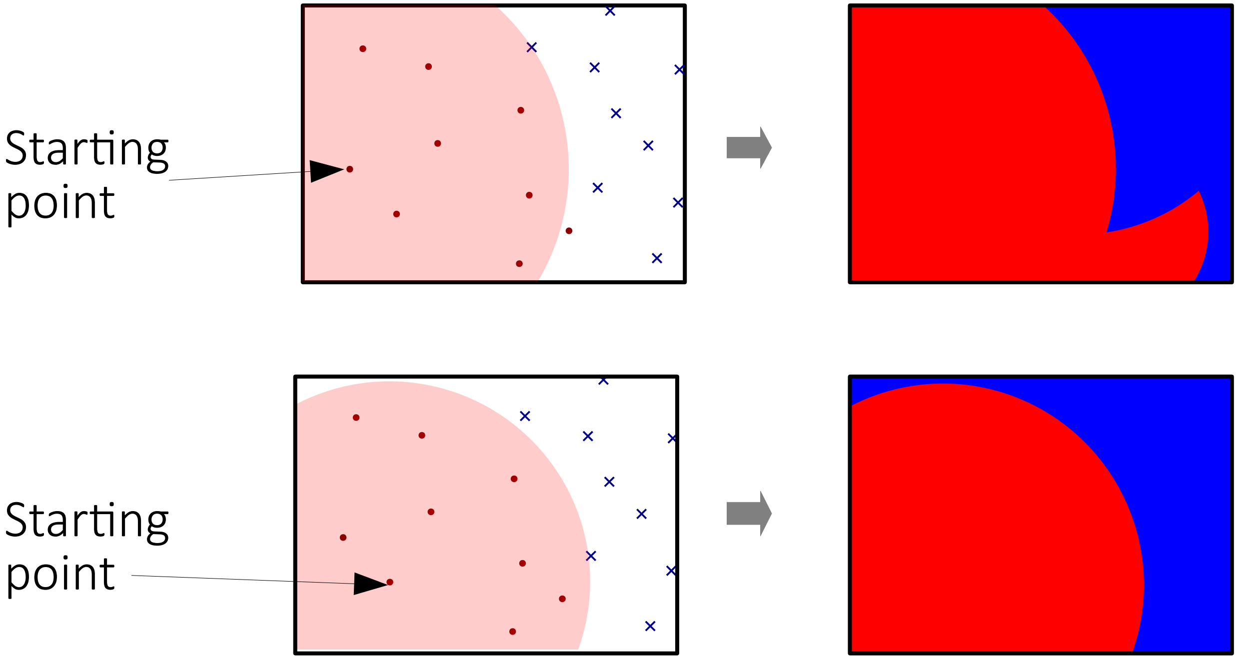
Train many guaranteed classifiers simultaneusly.



New lines of research

Pool of guaranteed classifiers:

Train many guaranteed classifiers simultaneusly.



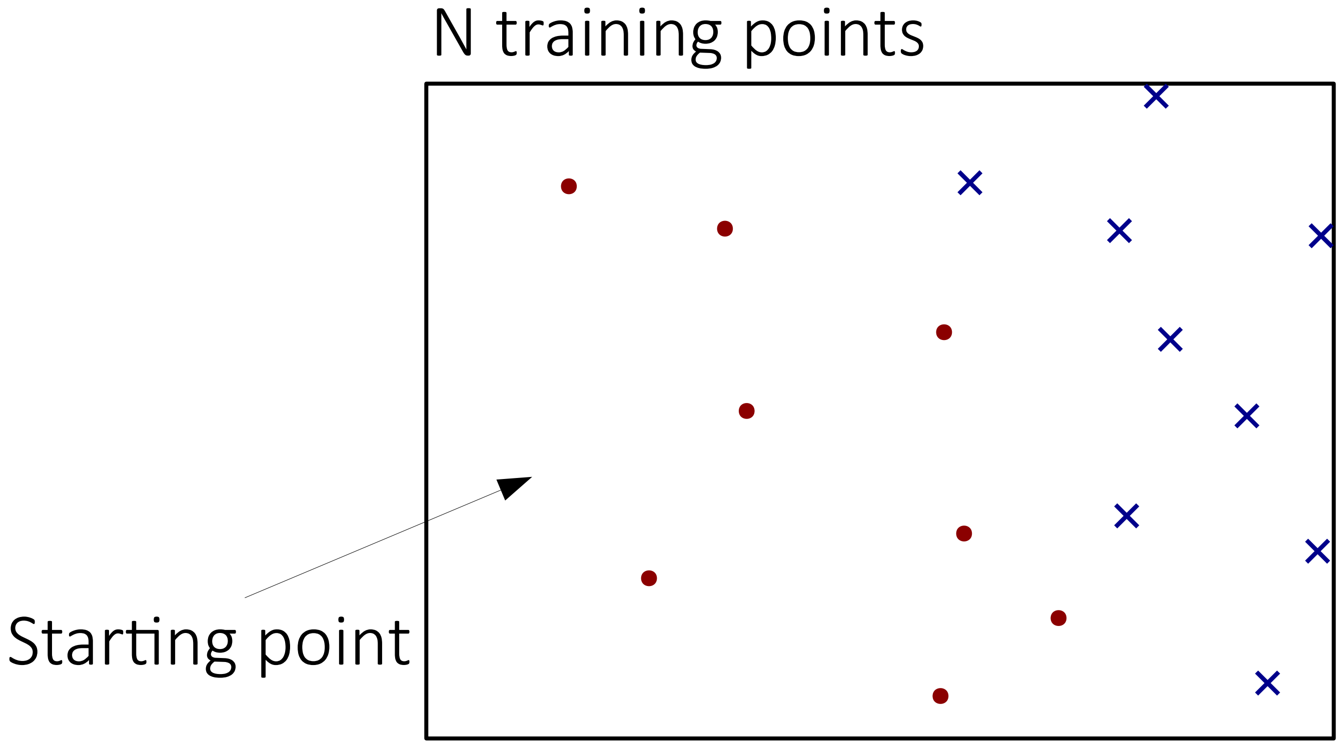
*Can we be **more** confident in the case of agreement?*

A taste of the theory

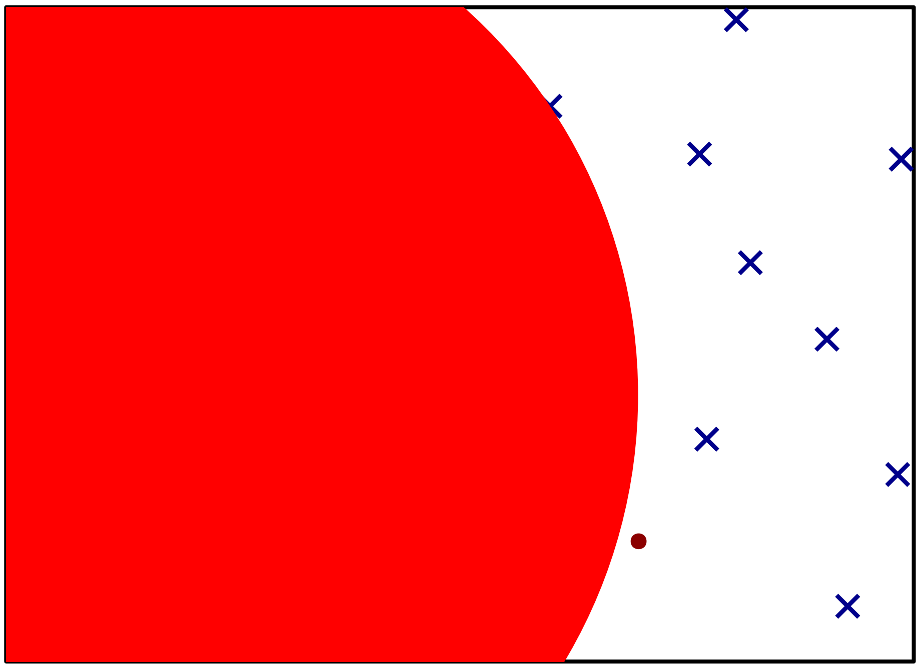
Just one ball

A taste of the theory

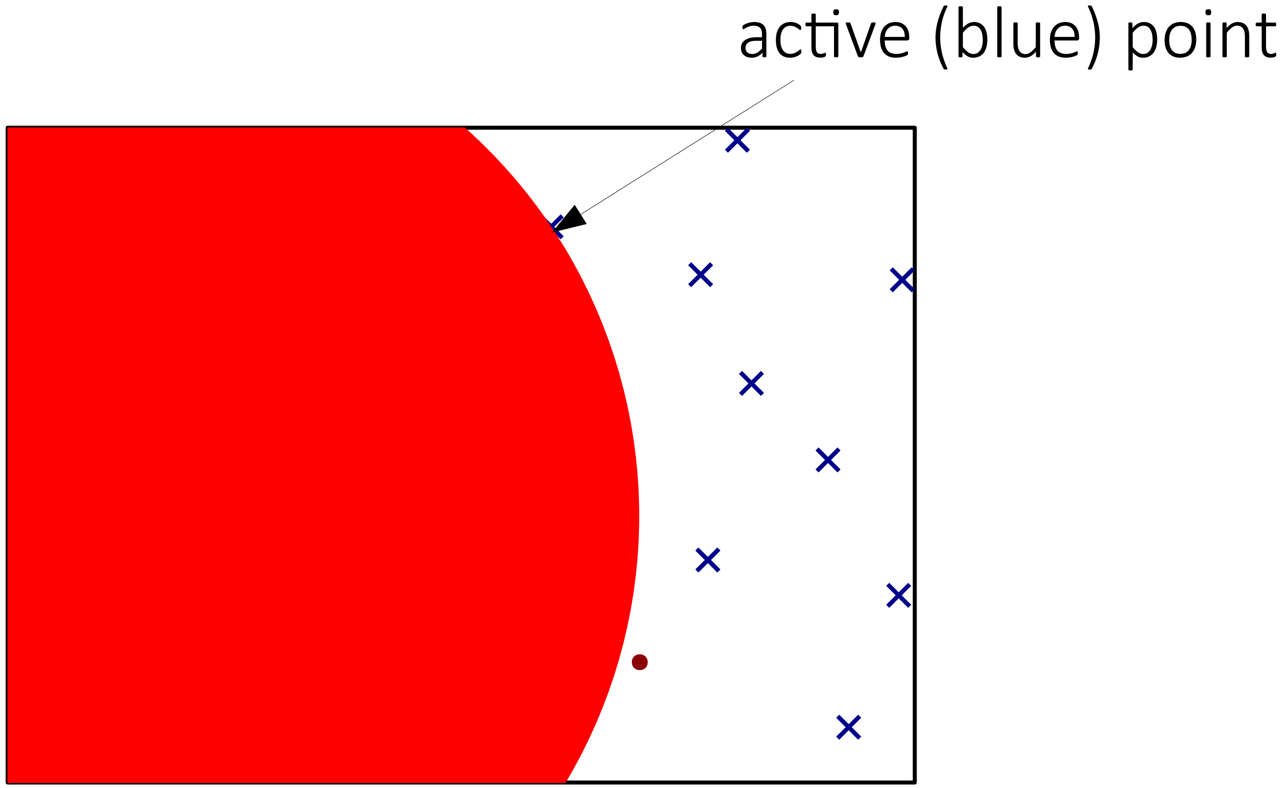
Just one ball



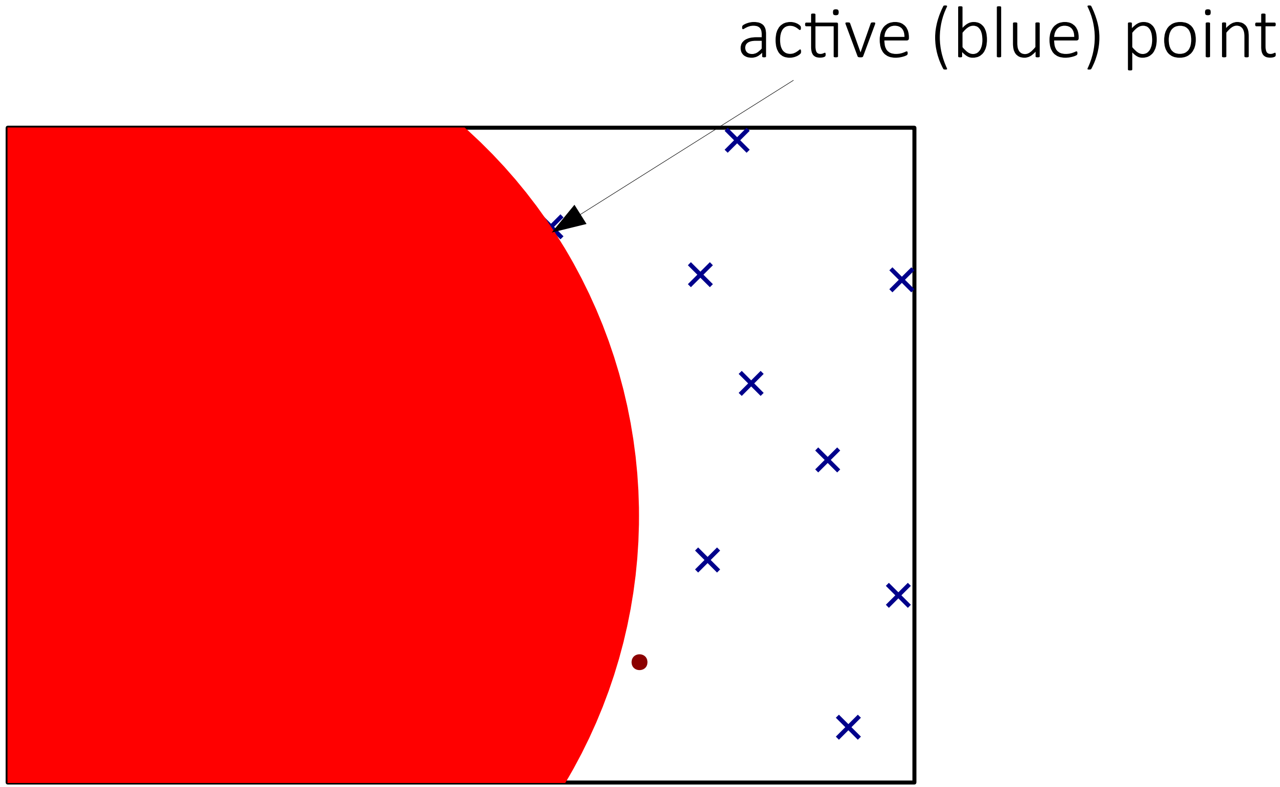
A taste of the theory



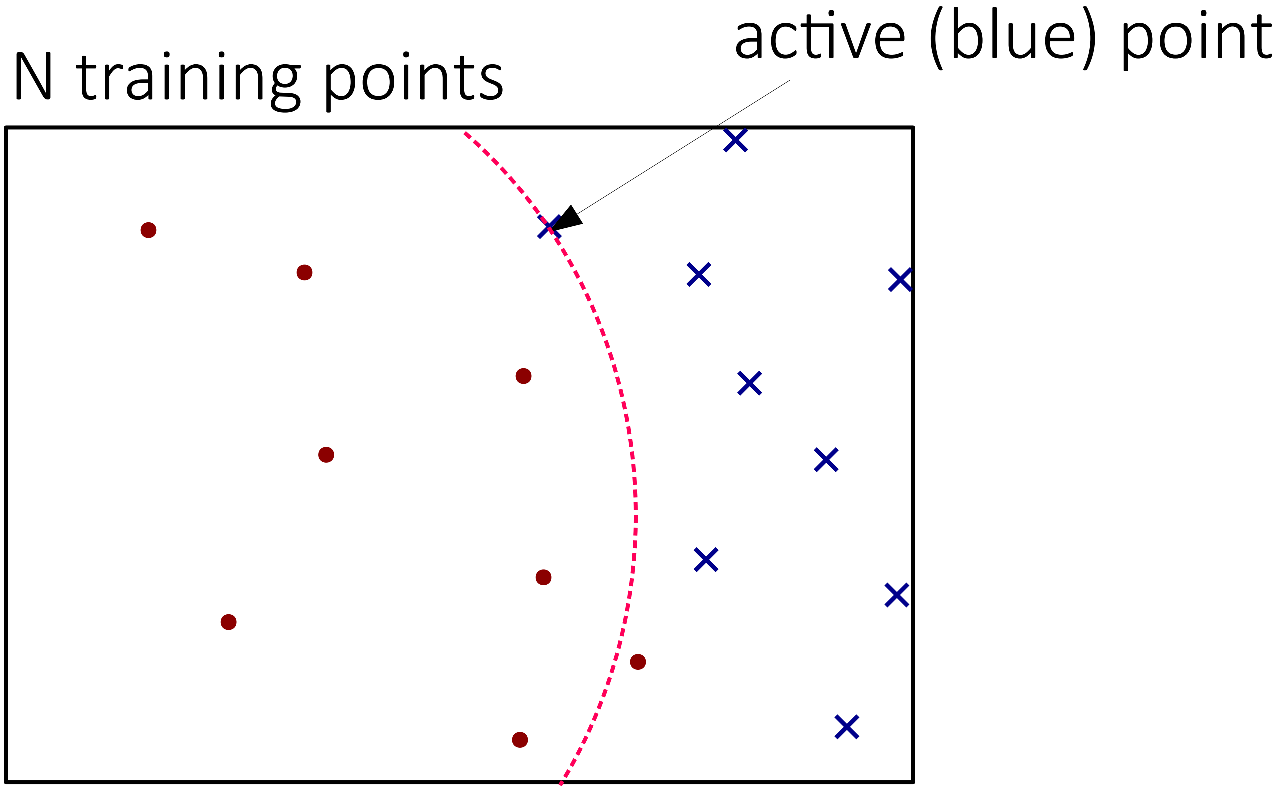
A taste of the theory



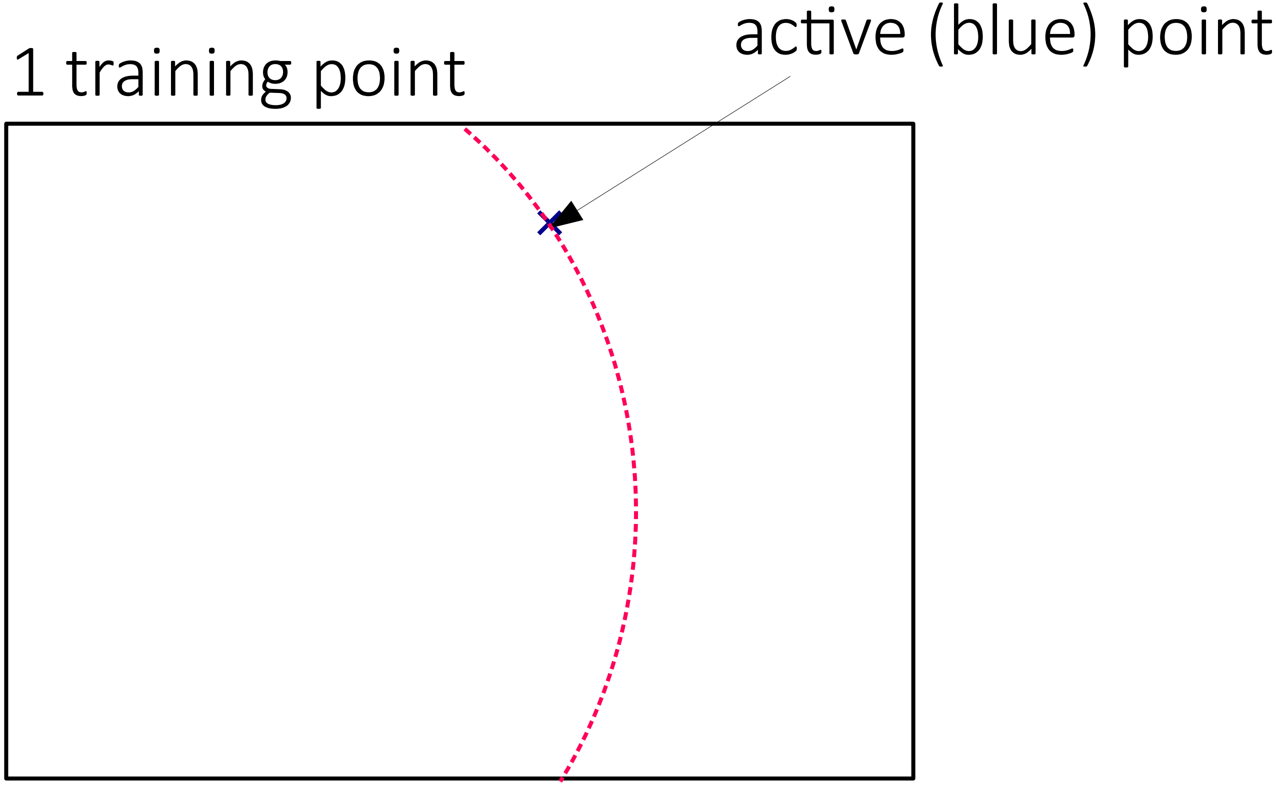
(1) There is one important point



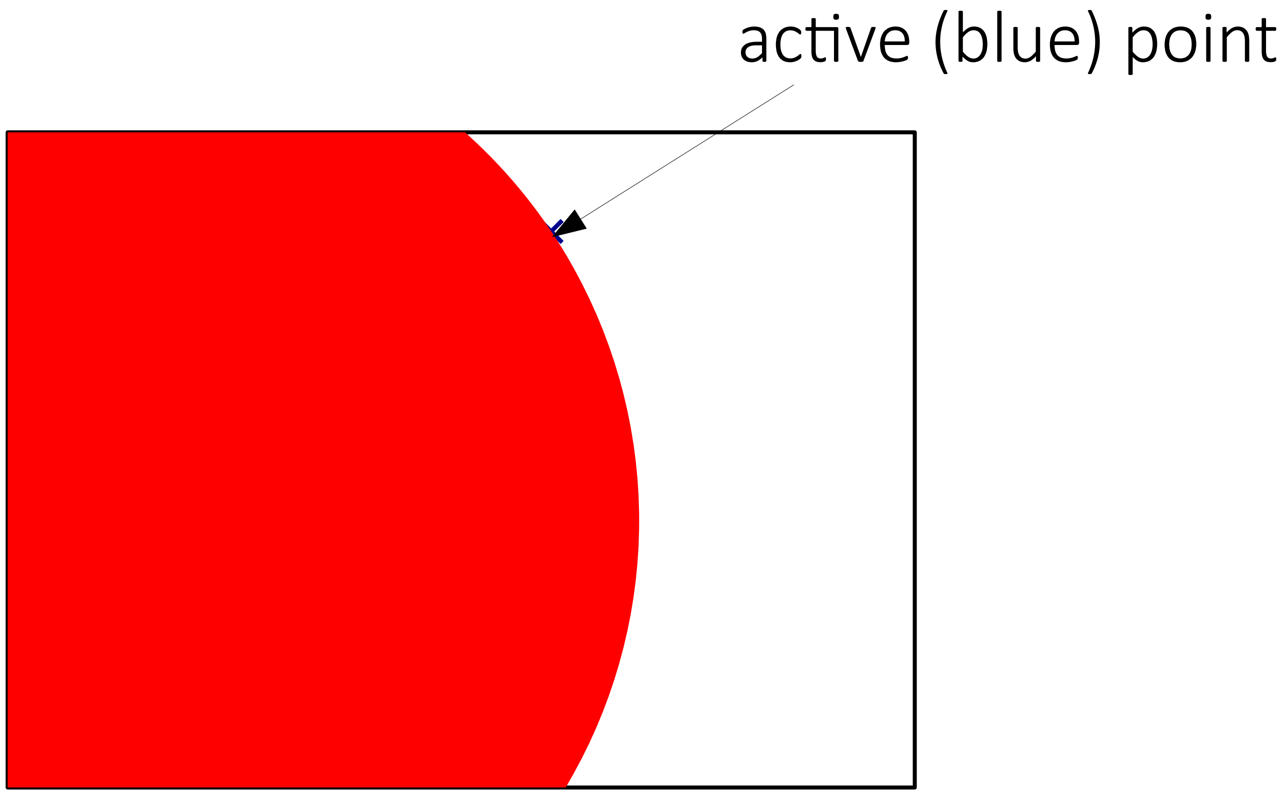
(1) There is one important point



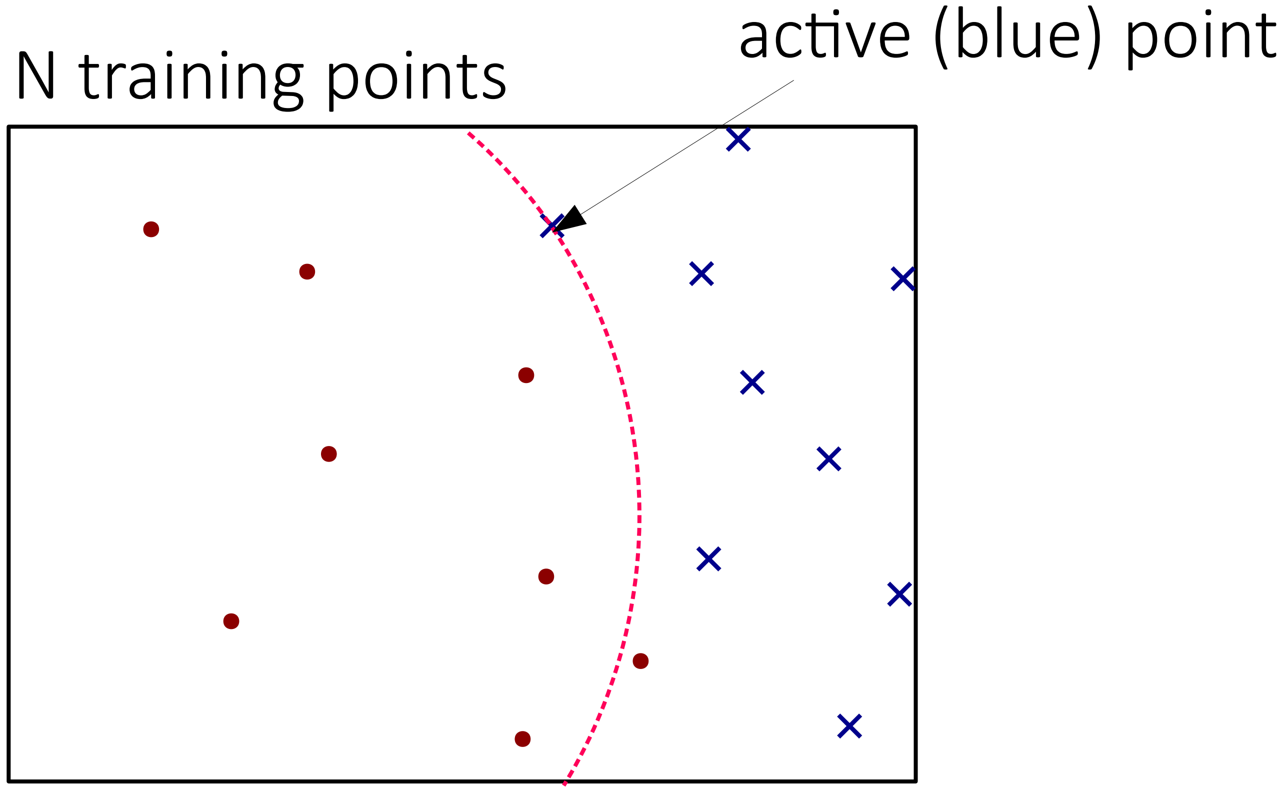
(1) There is one important point



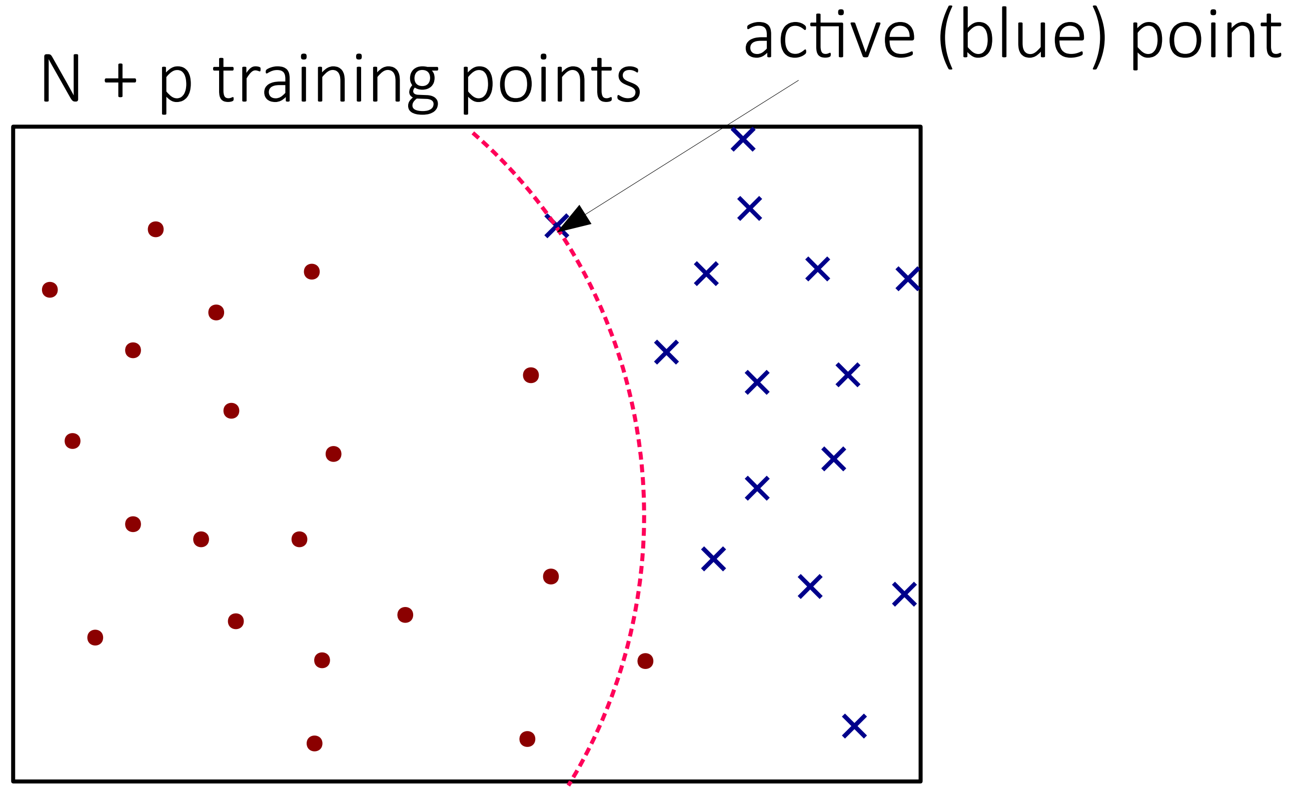
(1) There is one important point



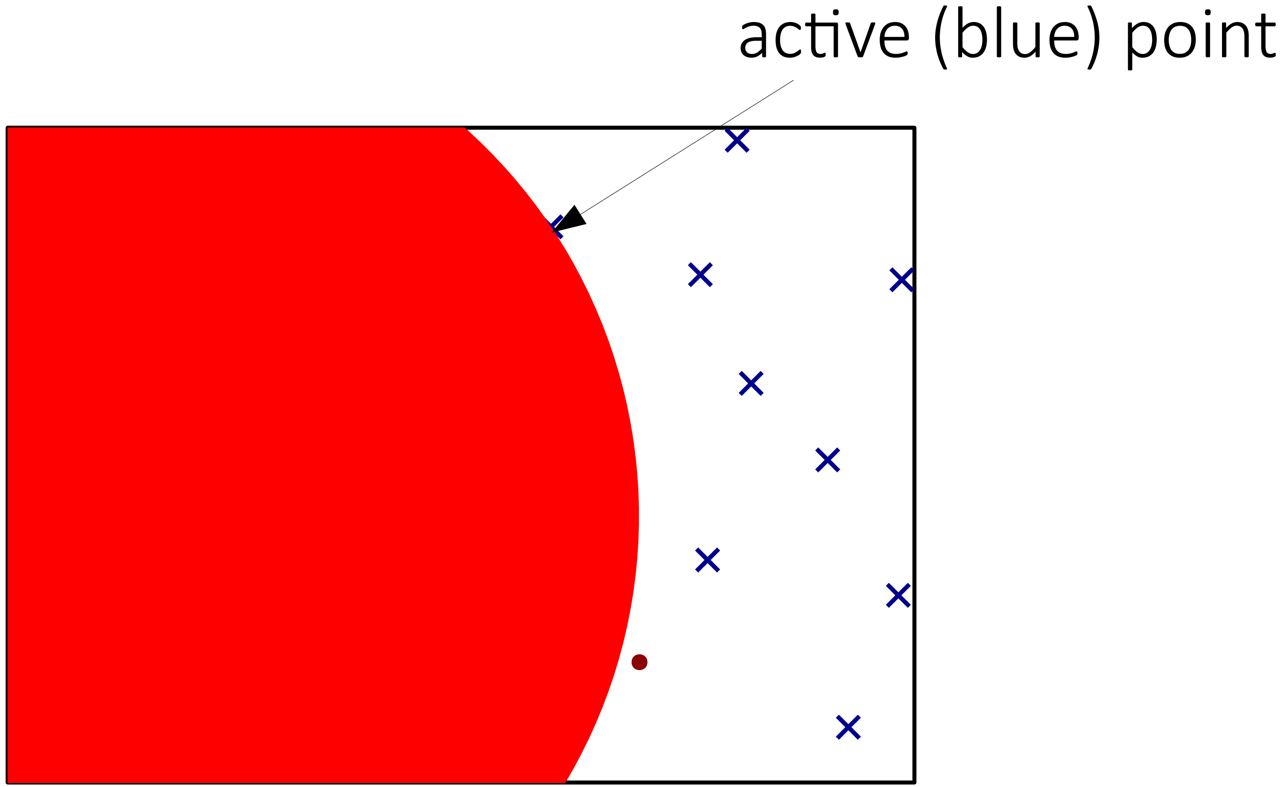
A taste of the theory



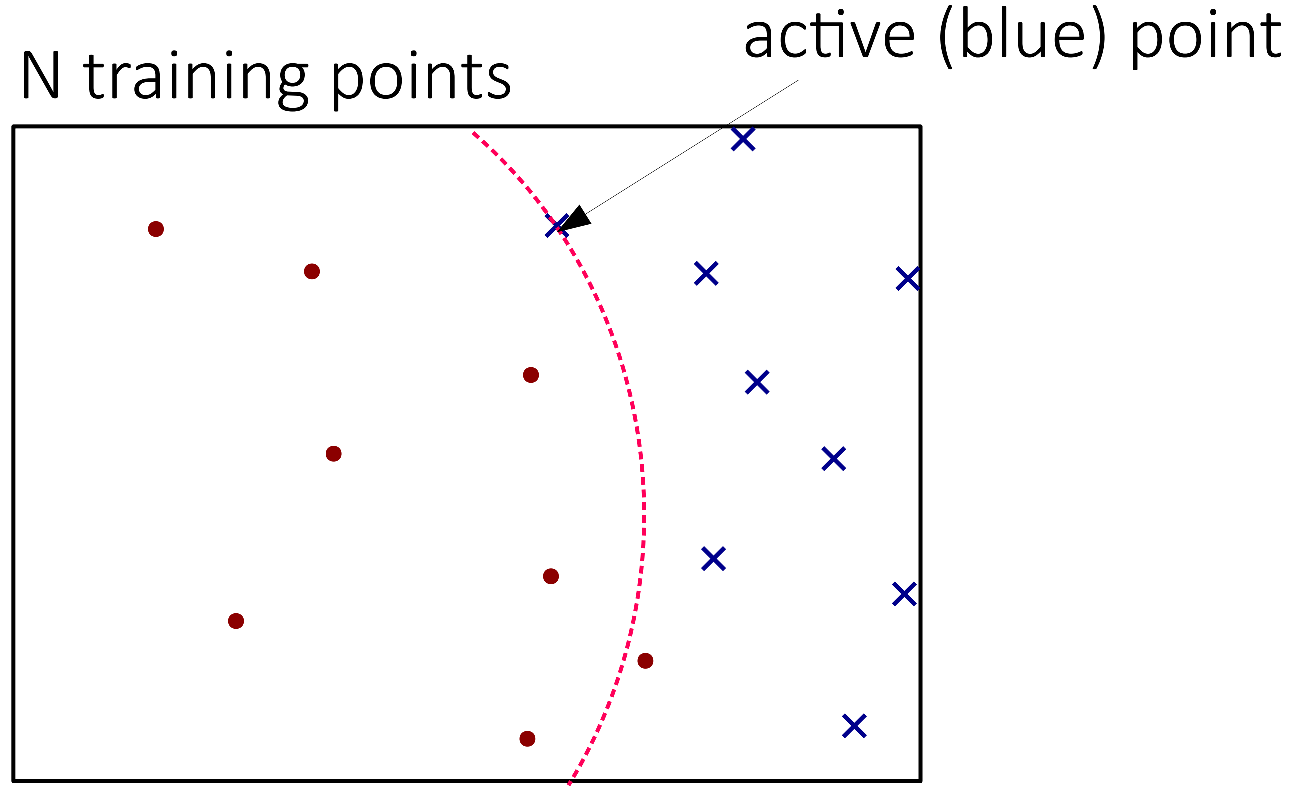
A taste of the theory



(2) Stability in the case of confirmation

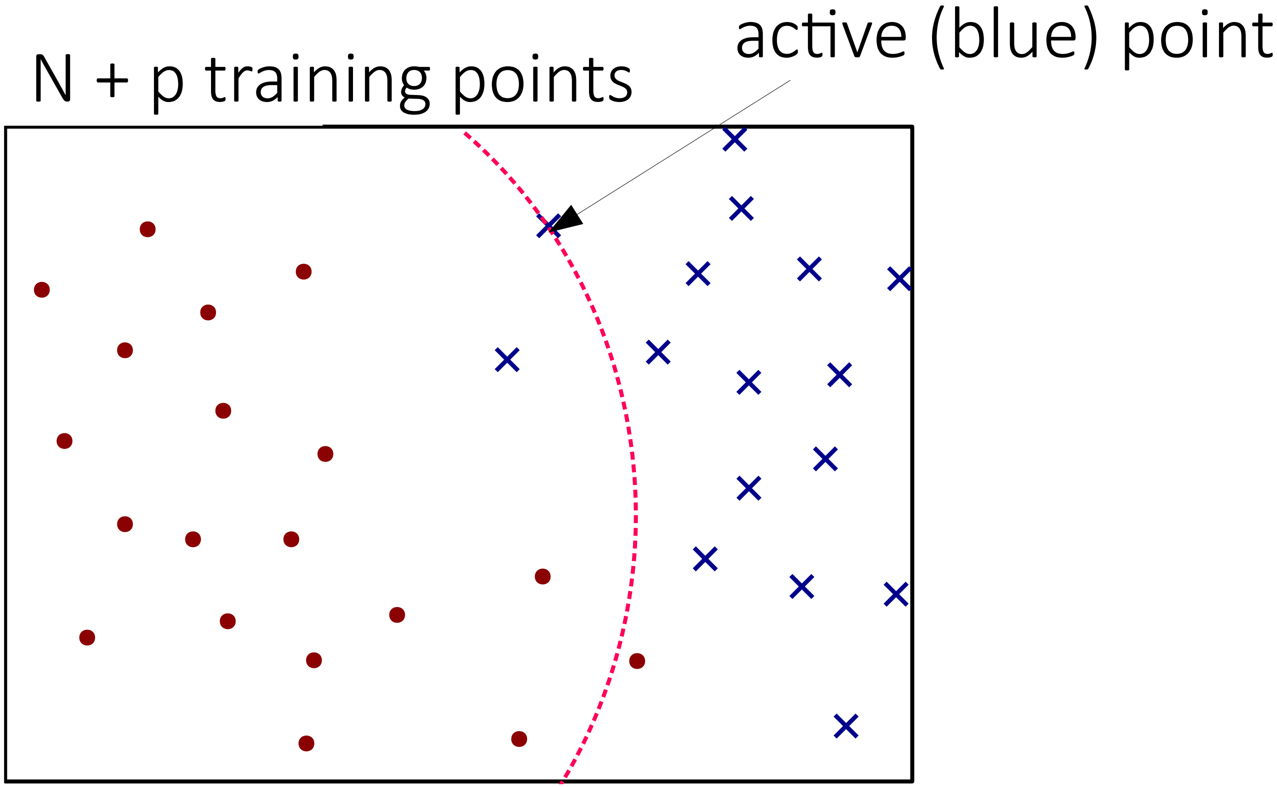


...

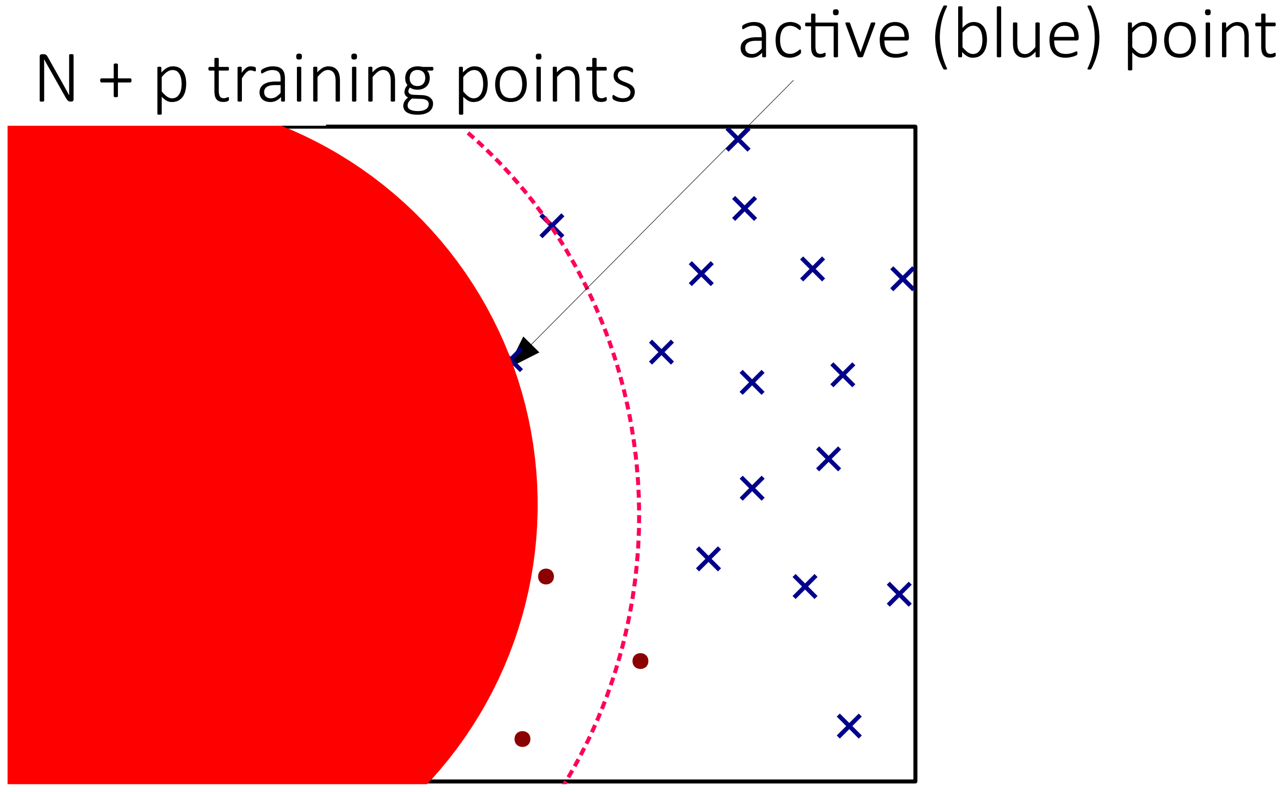


active (blue) point
N training points

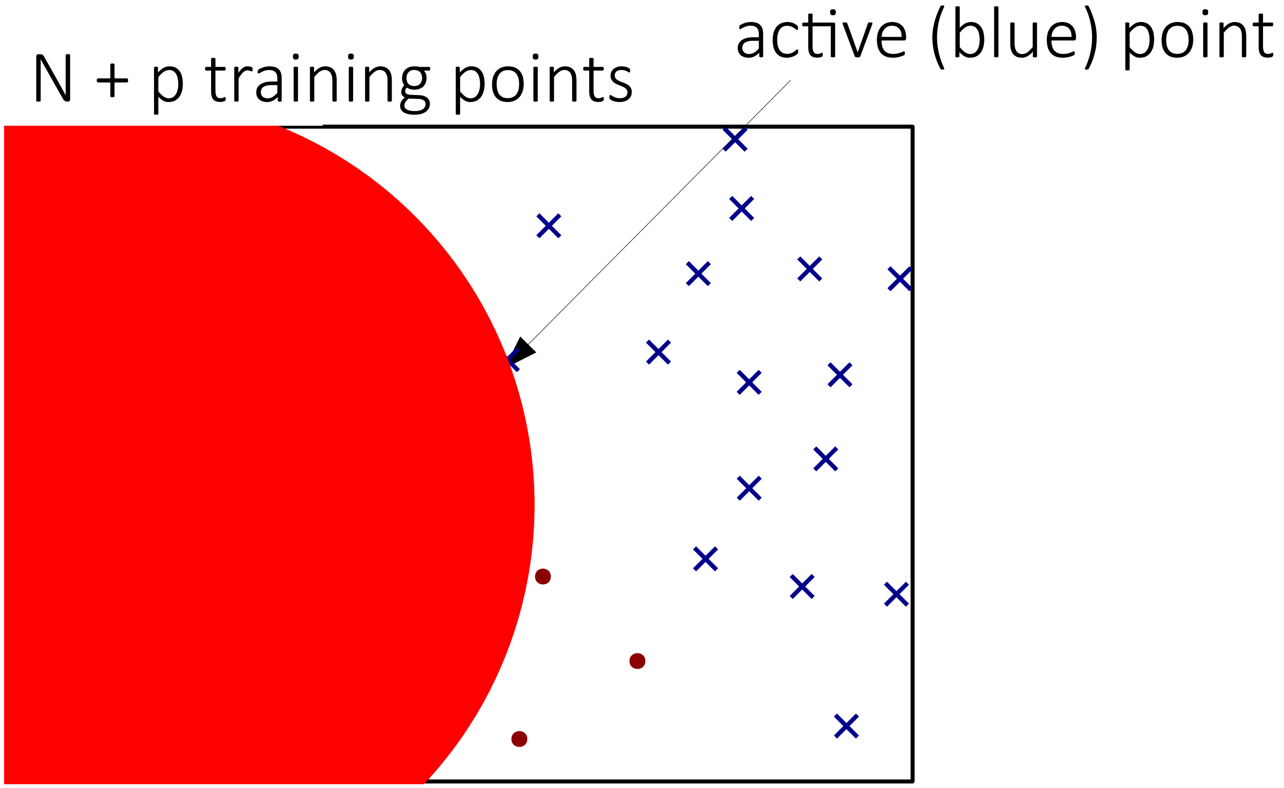
A taste of the theory



(3) Responsiveness to contradiction



(3) Responsiveness to contradiction



Thanks to (1),(2) and (3):

Take N+p points

Thanks to (1),(2) and (3):

Take N+p points

Let $\hat{y}_N(x)$ be the ball trained with the first **N points**;

Thanks to (1),(2) and (3):

Take N+p points

Let $\hat{y}_N(x)$ be the ball trained with the first **N points**;
Let $\hat{y}_{N+p}(x)$ be the ball trained with all the **N+p points.**

Thanks to (1),(2) and (3):

Take N+p points

Let $\hat{y}_N(x)$ be the ball trained with the first **N points**;
Let $\hat{y}_{N+p}(x)$ be the ball trained with all the **N+p points.**

C1 : "$\hat{y}_N(x)$ correctly classifies the last p points"

Thanks to (1),(2) and (3):

Take N+p points

Let $\hat{y}_N(x)$ be the ball trained with the first **N points**;
Let $\hat{y}_{N+p}(x)$ be the ball trained with all the **N+p points.**

C1 : "$\hat{y}_N(x)$ correctly classifies the last p points"

$$\Longleftrightarrow$$

Thanks to (1),(2) and (3):

Take N+p points

Let $\hat{y}_N(x)$ be the ball trained with the first **N points**;
Let $\hat{y}_{N+p}(x)$ be the ball trained with all the **N+p points.**

C1 : "$\hat{y}_N(x)$ correctly classifies the last p points"

$$\Longleftrightarrow$$

C2 : "the active point of $\hat{y}_{N+p}(x)$ is one of the first N points"

Thanks to (1),(2) and (3):

Take N+p points

Let $\hat{y}_N(x)$ be the ball trained with the first **N points**;
Let $\hat{y}_{N+p}(x)$ be the ball trained with all the **N+p points.**

C1 : “$\hat{y}_N(x)$ correctly classifies the last p points”

$$\Longleftrightarrow$$

C2 : “the active point of $\hat{y}_{N+p}(x)$ is one of the first N points”

$$\mathbb{P}^{N+p}\{C1\} = \mathbb{P}^{N+p}\{C2\} = \frac{N}{N+p}$$

Thanks to (1),(2) and (3):

Take N+p points

Let $\hat{y}_N(x)$ be the ball trained with the first **N points**;
Let $\hat{y}_{N+p}(x)$ be the ball trained with all the **N+p points.**

C1 : "$\hat{y}_N(x)$ correctly classifies the last p points"

$$\Longleftrightarrow$$

C2 : "the active point of $\hat{y}_{N+p}(x)$ is one of the first N points"

$$\mathbb{P}^{N+p}\{C1\} = \mathbb{P}^{N+p}\{C2\} = \frac{N}{N+p} = \mathbb{E}[(1 - PE(\hat{y}_N(x))^p]$$

Thanks to (1),(2) and (3):

Take N+p points

Let $\hat{y}_N(x)$ be the ball trained with the first **N points**;
Let $\hat{y}_{N+p}(x)$ be the ball trained with all the **N+p points.**

C1 : "$\hat{y}_N(x)$ correctly classifies the last p points"

$$\Longleftrightarrow$$

C2 : "the active point of $\hat{y}_{N+p}(x)$ is one of the first N points"

$$\mathbb{P}^{N+p}\{C1\} = \mathbb{P}^{N+p}\{C2\} = \frac{N}{N+p} = \mathbb{E}[(1 - PE(\hat{y}_N(x))^p]$$

$$\Longrightarrow \quad PE(\hat{y}_N(x)) \sim \mathrm{Beta}[1, N]$$

Thank you!

Are you interested in knowing more?
Write to us!

algo.care@unibs.it

Try GEM-BALLS now!
www.algocare.it/GEM-BALLS/

Bibliography about GEM

GEM-BALLS

- "A new classification algorithm with guaranteed sensitivity and specificity for medical applications"
  A. Carè, F.A. Ramponi, M.C. Campi
  IEEE Control Systems Letters. 2(3):393-398 [doi: 10.1109/LCSYS.2018.2840427], 2018

The original GEM

- "Classification with Guaranteed Probability of Error"
  M.C. Campi
  Machine Learning, 80:63-84, 2010.

More about the VF application

- "Ventricular defibrillation: Classification with GEM and a roadmap for future investigations"
  F. Baronio, M. Baronio, M.C. Campi, A. Carè, S. Garatti, G. Perone
  2017 IEEE 56th Annual Conference on Decision and Control (CDC), 2718-2723

Bibliography about GEM

New lines of research

- "Novel bounds on the probability of misclassification in majority voting: leveraging the majority size"
  A.T.J.R. Cobbenhagen, A. Carè, M.C. Campi, F.A. Ramponi, D.J. Antunes, W.P.M.H. Heemels
  IEEE Control Systems Letters, vol. 5, no. 5, pp. 1513-1518, 2020.

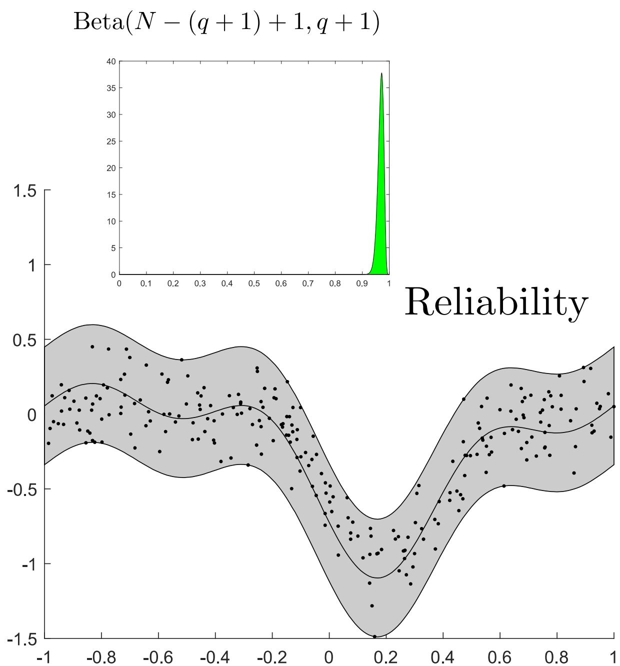
- "A study on majority-voting classifiers with guarantees on the probability of error"
  by A. Carè, M.C. Campi, F.A. Ramponi, S. Garatti, A.T.J.R. Cobbenhagen
  IFAC World Congress 2020.

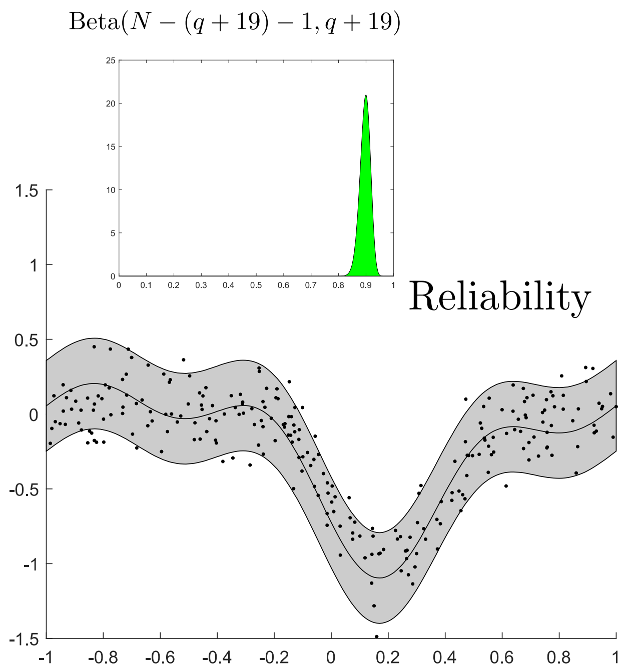
- "Consensus and Reliability: The Case of Two Binary Classifiers"
  A.T.J.R. Cobbenhagen. A. Carè, M.C. Campi, F.A. Ramponi, W.P.M.H. Heemels
  8th IFAC Workshop on Distributed Estimation and Control in Networked Systems
  Sept. 16-17, 2019, Chicago, IL, USA.

Bibliography beyond GEM

Regression (L-infinity)

- "On a class of Interval Predictor Models with universal reliability,"
  S. Garatti, M.C. Campi, A. Carè
  Automatica. 110, 108542 [doi: 10.1016/j.automatica.2019.108542], 2019.





Bibliography beyond GEM

Other decision schemes with strong generalization properties

- “Scenario optimization with relaxation: a new tool for design and application to machine learning problems”
  M.C. Campi, S. Garatti
  2020 59th IEEE Conference on Decision and Control (CDC), 2463-2468

- Book:
  “Introduction to the scenario approach”
  M.C. Campi, S. Garatti.
  SIAM, 2018

- Forthcoming review paper:
  “Data-driven decision-making and the scenario approach“

algo.care@unibs.it