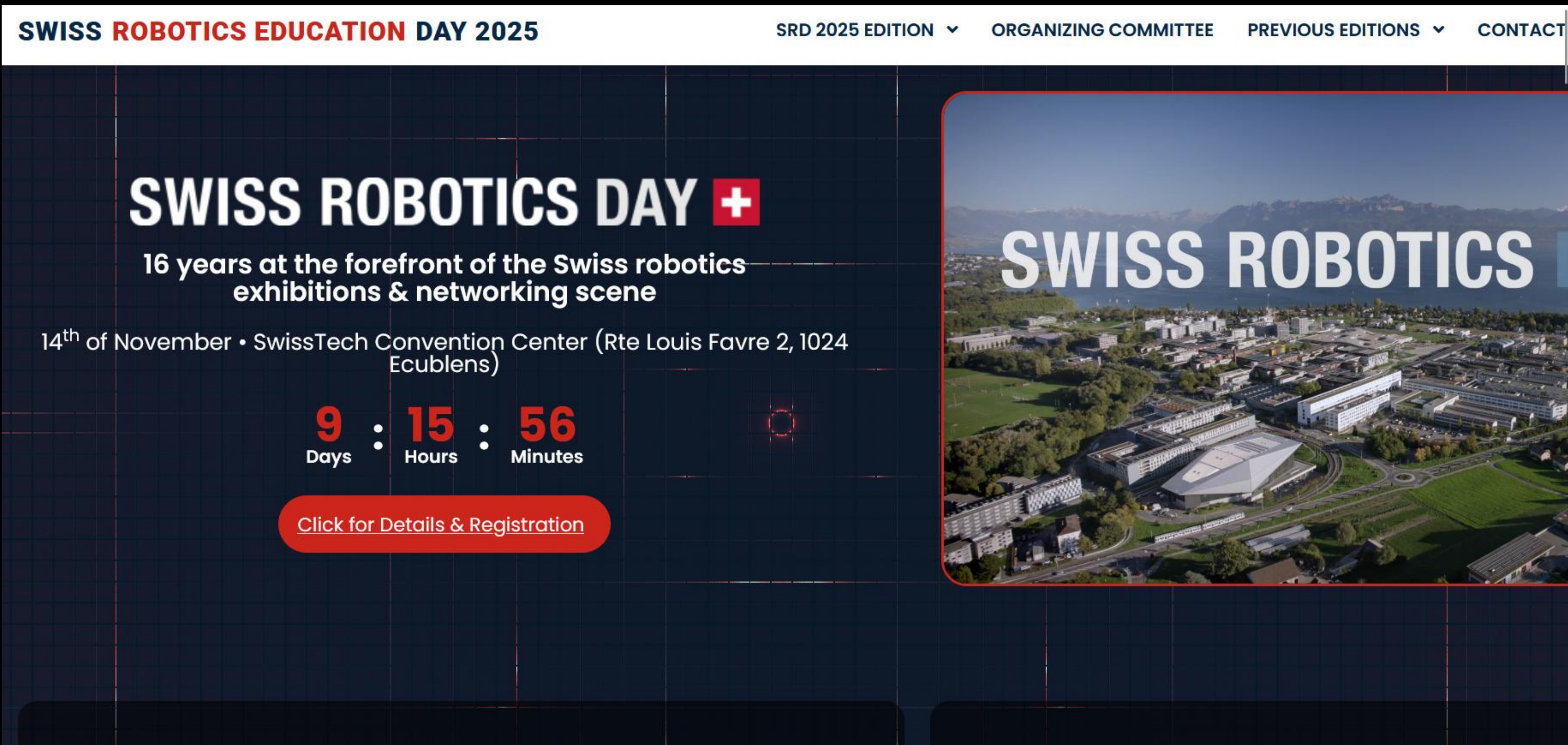
https://swissroboticsday.ch



https://swiss-robotics.org/

SWISS ROBOTICS ASSOCIATION

Home | About Us | Events | Job fair | Robotics Ecosystem Map | News | Contact | Account

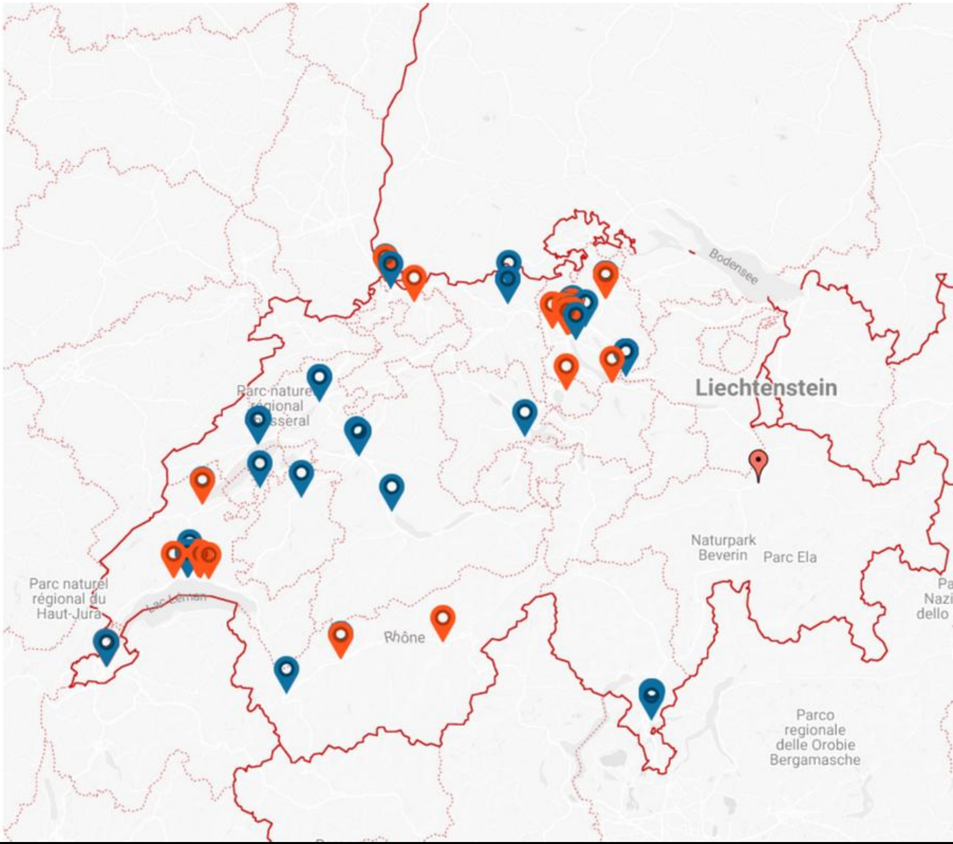
Become a member ↗

ROBOTICS ECOSYSTEM MAP

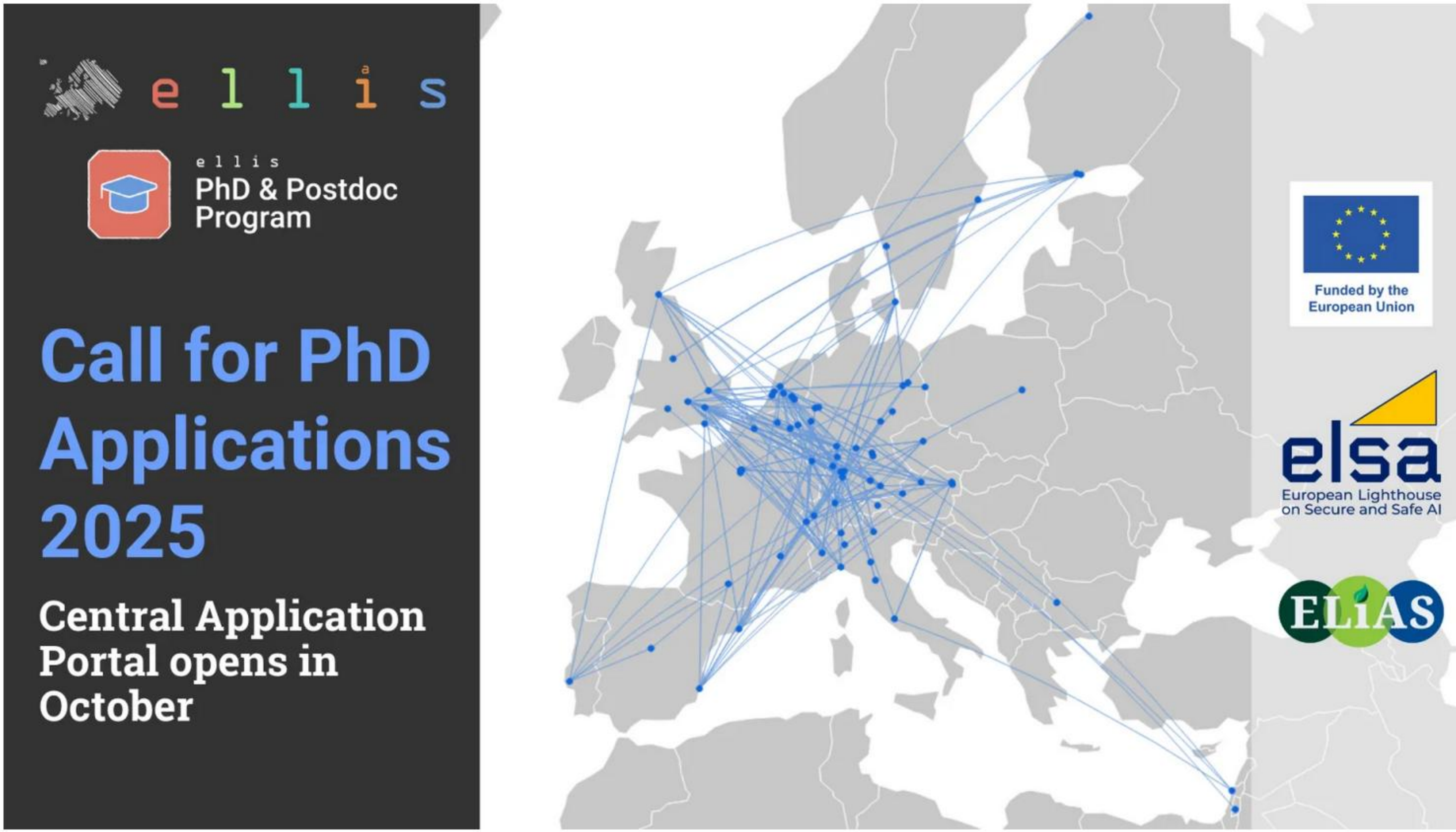
Explore Switzerland's Robotics Landscape

Discover key players in Switzerland's robotics ecosystem with our interactive map. Navigate by industry, academia, or innovation to find connections and opportunities.

VIEW THE MAP



ELLIS PHD PROGRAM



https://ellis.eu/news/ellis-phd-program-call-for-applications-2025

Interactive Lecture

Classification with
Gaussian Mixture Models (GMM) + Bayes
K-nearest neighbors

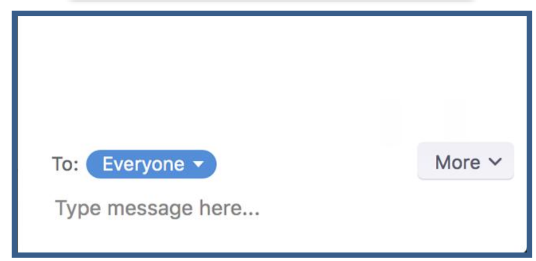
Interactions during interactive exercises

Draw solutions onto zoom window directly

Type it in @ask-a-question text channel

Type it in zoom chat box

Teaching assistants will reply live

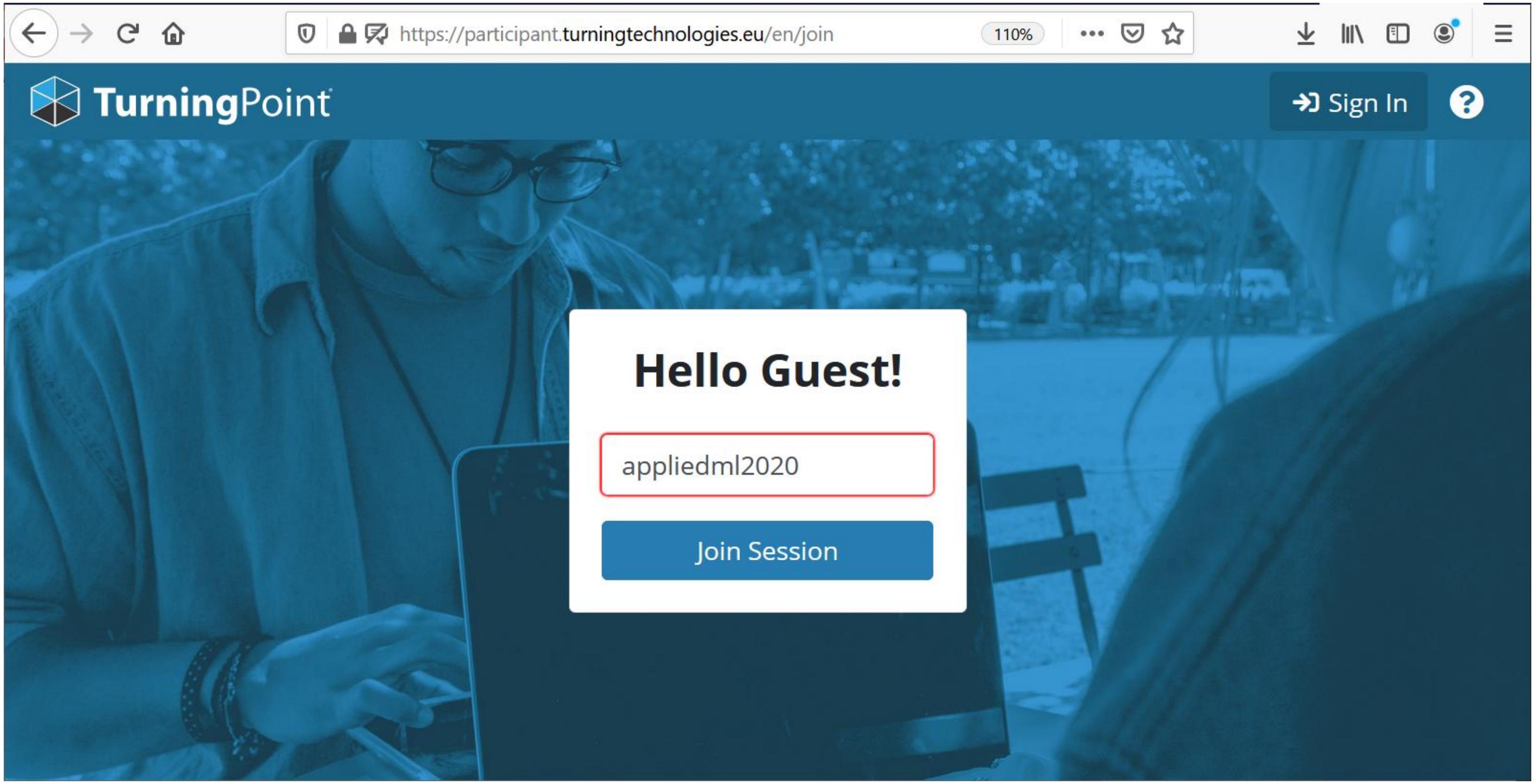


EPFL

Launch polling system

https://participant.turningtechnologies.eu/en/join

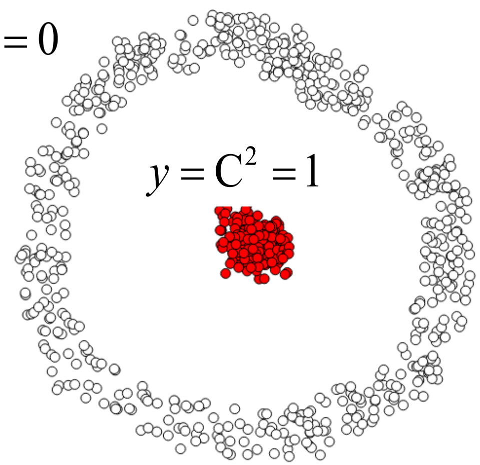
Acces as GUEST and enter the session id: *appliedml2020*



Can we classify the following dataset using one Gaussian to model each class?

$y = C^1 = 0$

$y = C^2 = 1$



Decision Boundary

$$p\left(y = C^1 \mid x\right) = p\left(y = C^2 \mid x\right)$$

$$p(x \mid y = c) \sim p\left(x \mid \mu^c, \Sigma^c\right)$$

μ^c, Σ^c : mean and covariance matrix

A. Yes

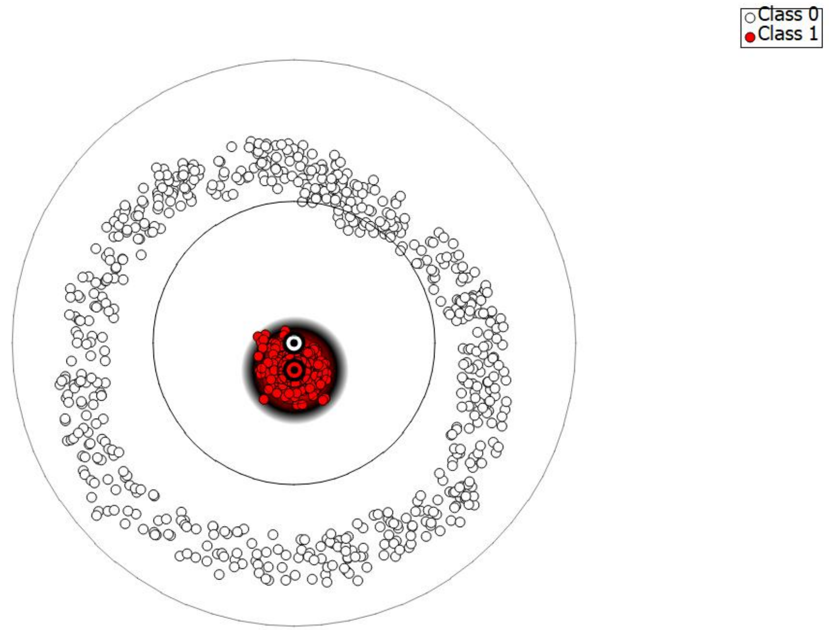
B. No

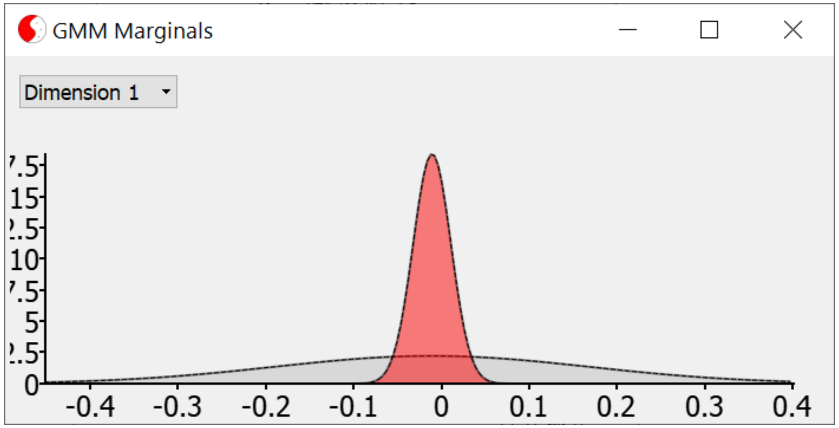
C. I do not know

EPFL

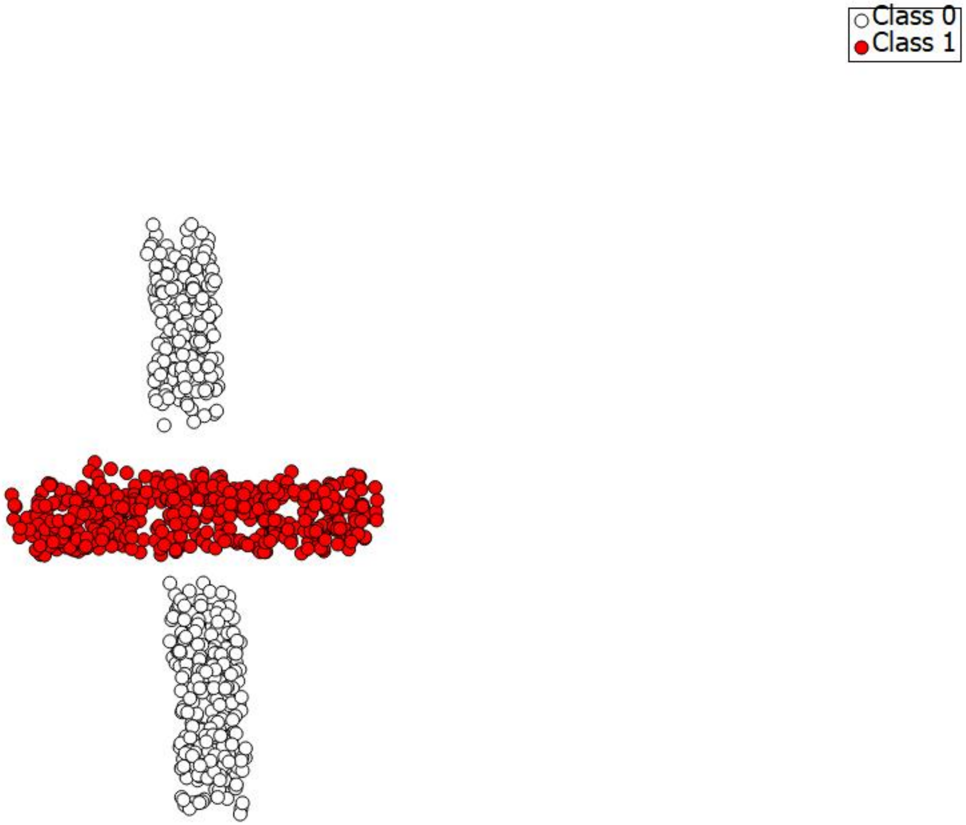
Can we classify the following dataset using one gaussian to model each class?

Answer YES





Can we classify the following dataset using one Gaussian to model each class?

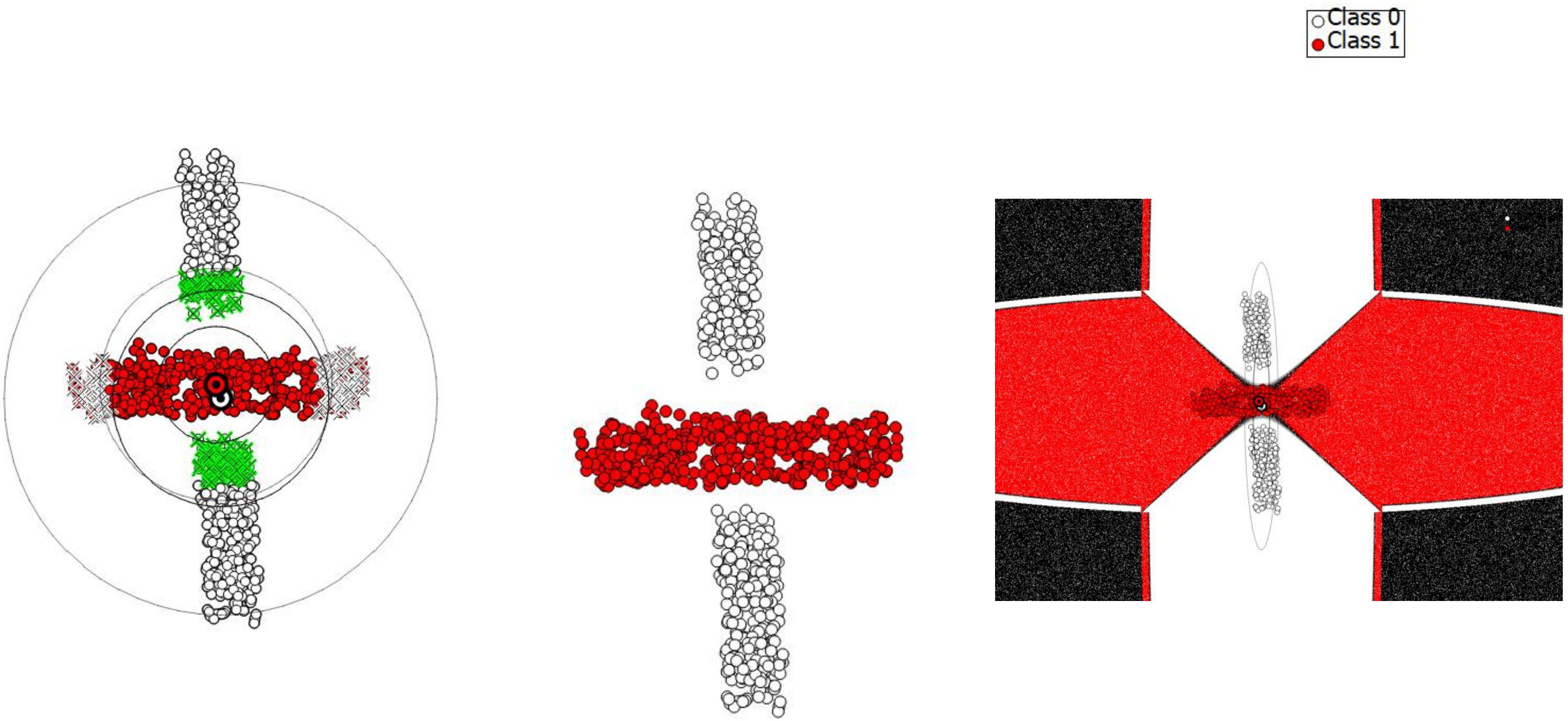


A. Yes

B. No

C. I do not know

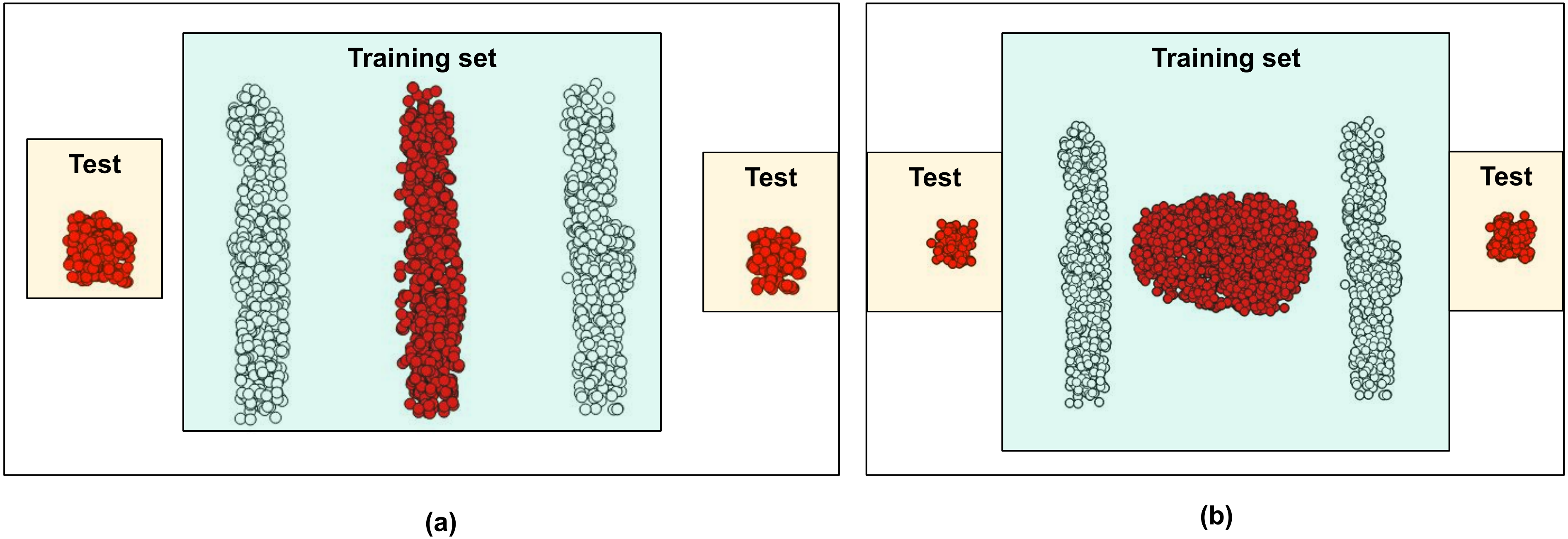
Can we classify the following dataset using one Gaussian to model each class?



The answer is yes but not with any model.

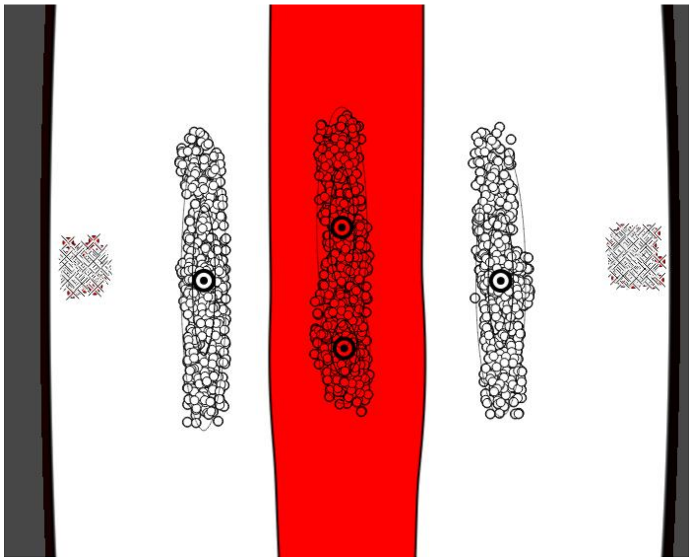
Only a diagonal or full covariance matrix will do.

Can we classify the outer datapoints using only the middle points for training?

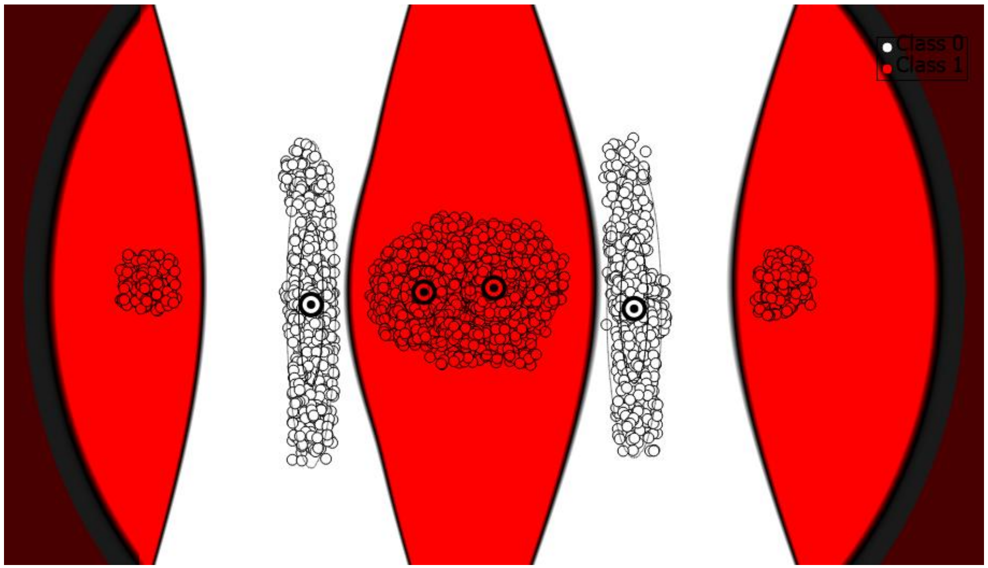


A. Yes for both a and b
B. No for both a and b
C. Yes for a only
D. Yes for b only
E. I do not know

Can we classify the outer datapoints using only the middle points for training?



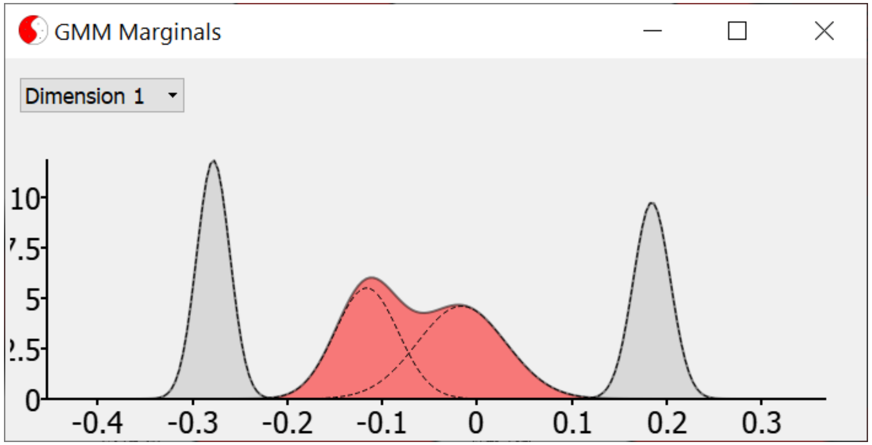
(a)



(b)

The answer is yes for (b) only.

Again, this is correct but not with any model. Only a diagonal or full covariance matrix will do and we need two Gauss functions at minimum.



Determining the boundary across two pdf-s

We must determine the class with class label c that is most likely to have generated the datapoint x: $p(y=C\,|\,x)$

Bayes's rule: $p(y=C\,|\,x)=\dfrac{p(y=C)\,p(x\,|\,y=C)}{p(x)}$

$p(y=C)$: Probability of class C

$p(x)$: Marginal on x

$p(x\,|\,y=C)$: class conditional distribution of x
~ how the samples are distributed within class C.

$$p\left(x\,|\,y=C^1\right)\sim p\left(x\,|\,\mu^1,\Sigma^1\right)=\frac{1}{(2\pi)^{N/2}\left|\Sigma^1\right|^{1/2}}e^{-\left(x-\mu^1\right)\left(\Sigma^1\right)^{-1}\left(x-\mu^1\right)^T}$$

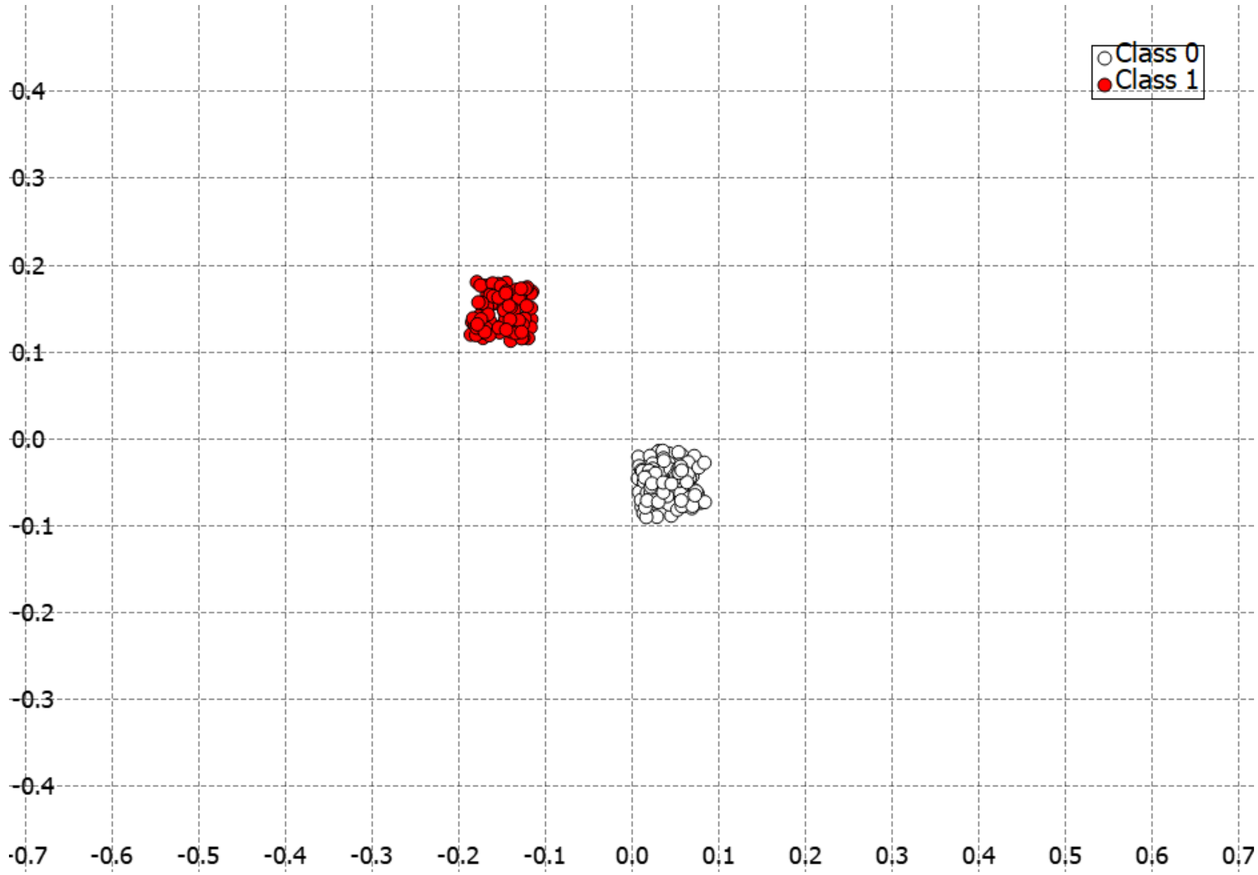
To determine the class label, compute optimal *Bayes classifier.*

A point x belongs to class C^1 if $p\left(y=C^1\,|\,x\right)>p\left(y=C^2\,|\,x\right)$

Assuming equal class distribution, $p\left(y=C^1\right)=p\left(y=C^2\right)$ & $\ln p\left(y=C^1\right)=\ln p\left(y=C^2\right)$

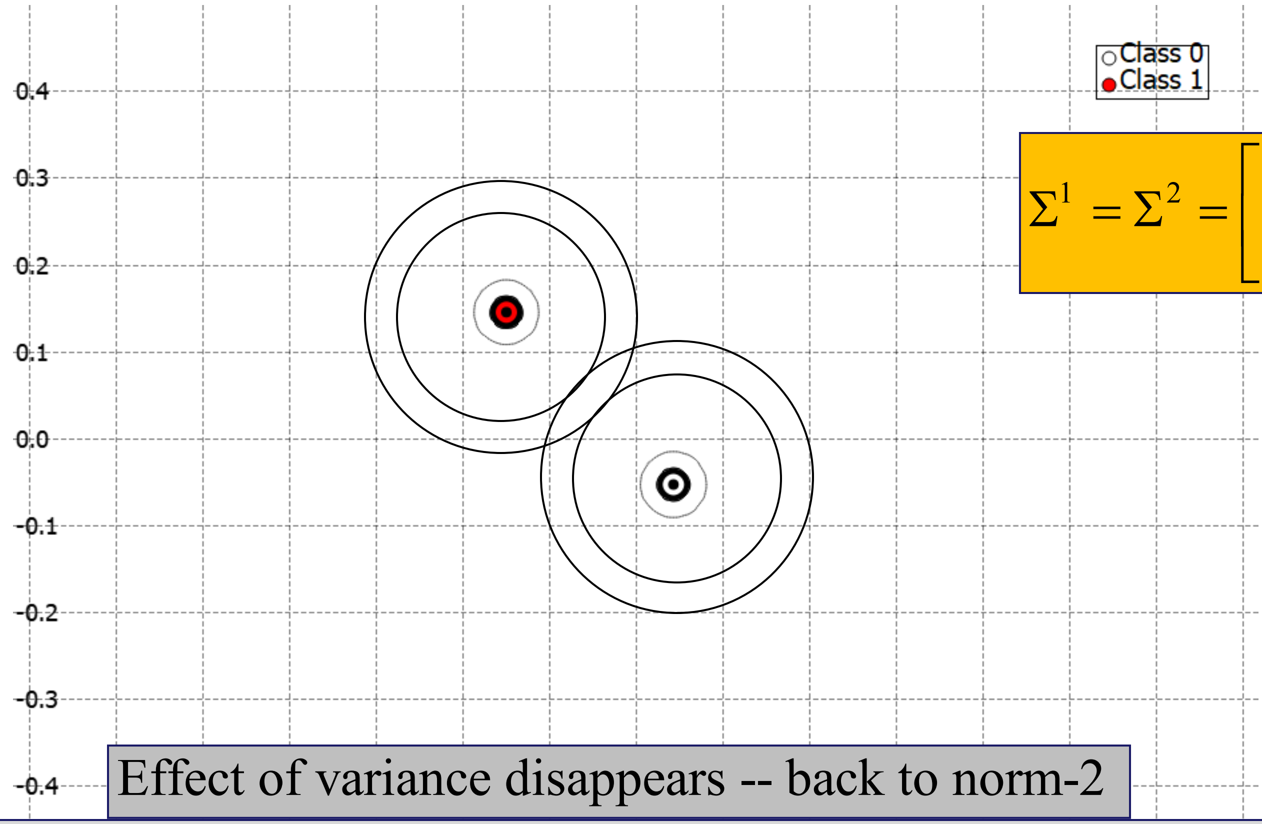
$$\Leftrightarrow\ \left(x-\mu^1\right)^T\left(\Sigma^1\right)^{-1}\left(x-\mu^1\right)+\log\left|\Sigma^1\right|<\left(x-\mu^2\right)^T\left(\Sigma^2\right)^{-1}\left(x-\mu^2\right)+\log\left|\Sigma^2\right|$$

Gaussian Discriminant Rule



$$\Leftrightarrow \quad \left(x-\mu^1\right)^T \left(\Sigma^1\right)^{-1} \left(x-\mu^1\right) + \log\left|\Sigma^1\right| < \left(x-\mu^2\right)^T \left(\Sigma^2\right)^{-1} \left(x-\mu^2\right) + \log\left|\Sigma^2\right|$$

Gaussian Discriminant Rule



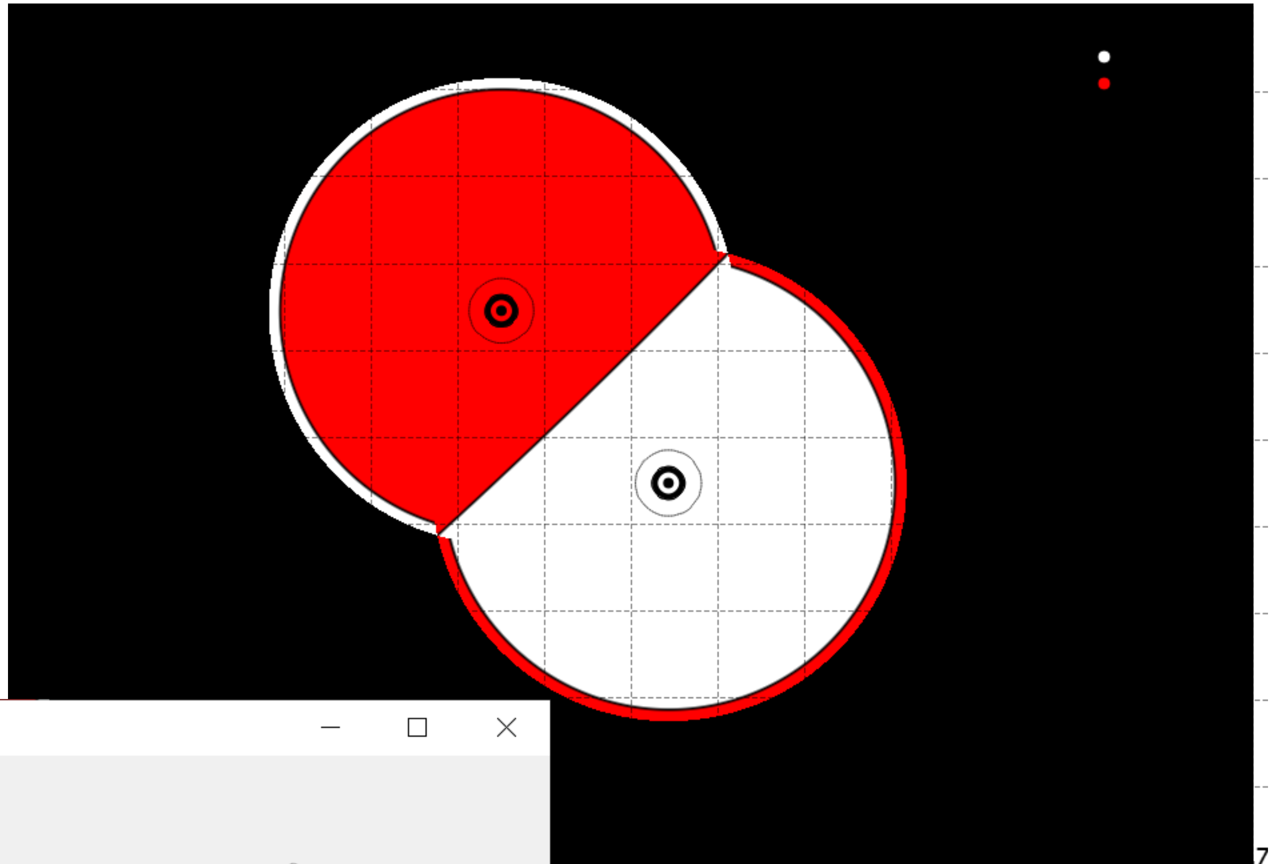
$$\Sigma^1 = \Sigma^2 = \begin{bmatrix} \sigma & 0 \\ 0 & \sigma \end{bmatrix}$$

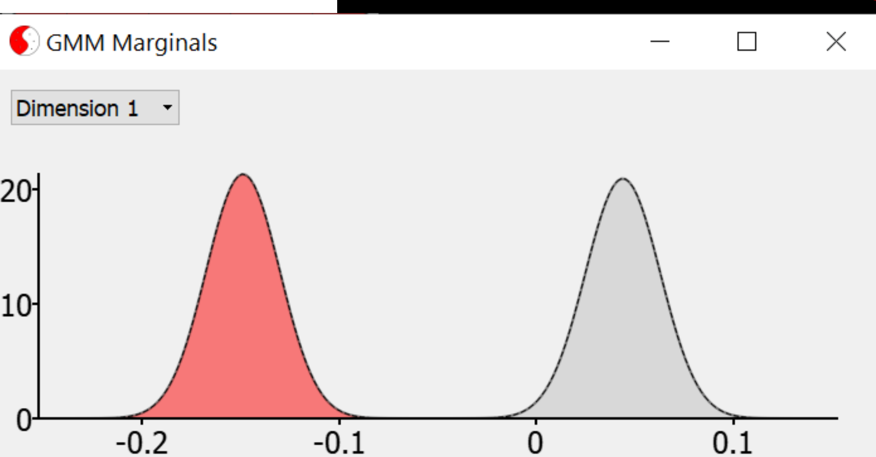
Effect of variance disappears -- back to norm-2

$$\Leftrightarrow \quad \left(x_j - \mu_j{}^1\right)^T (\sigma)^{-1} \left(x_j - \mu_j{}^1\right) < \left(x_j - \mu_j{}^2\right)^T (\sigma)^{-1} \left(x_j - \mu_j{}^2\right),\ j = \{1,2\}$$

$$\Leftrightarrow \quad \left(x - \mu^1\right)^T \left(\Sigma^1\right)^{-1} \left(x - \mu^1\right) + \log\left|\Sigma^1\right| < \left(x - \mu^2\right)^T \left(\Sigma^2\right)^{-1} \left(x - \mu^2\right) + \log\left|\Sigma^2\right|$$

Gaussian Discriminant Rule



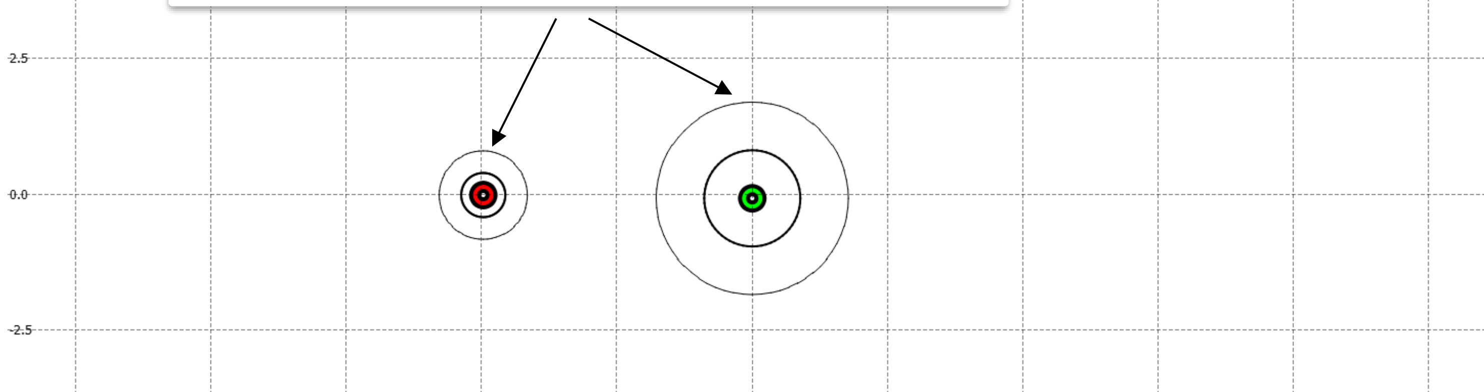


Exercise

Find the boundary when using GMM with one Gauss fct for each class

$$\Sigma^1 = \begin{bmatrix} \sigma^1 & 0 \\ 0 & \sigma^1 \end{bmatrix}, \Sigma^2 = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}, \sigma^2 > \sigma^1$$

Same value isoline correspond to 2 standard deviation

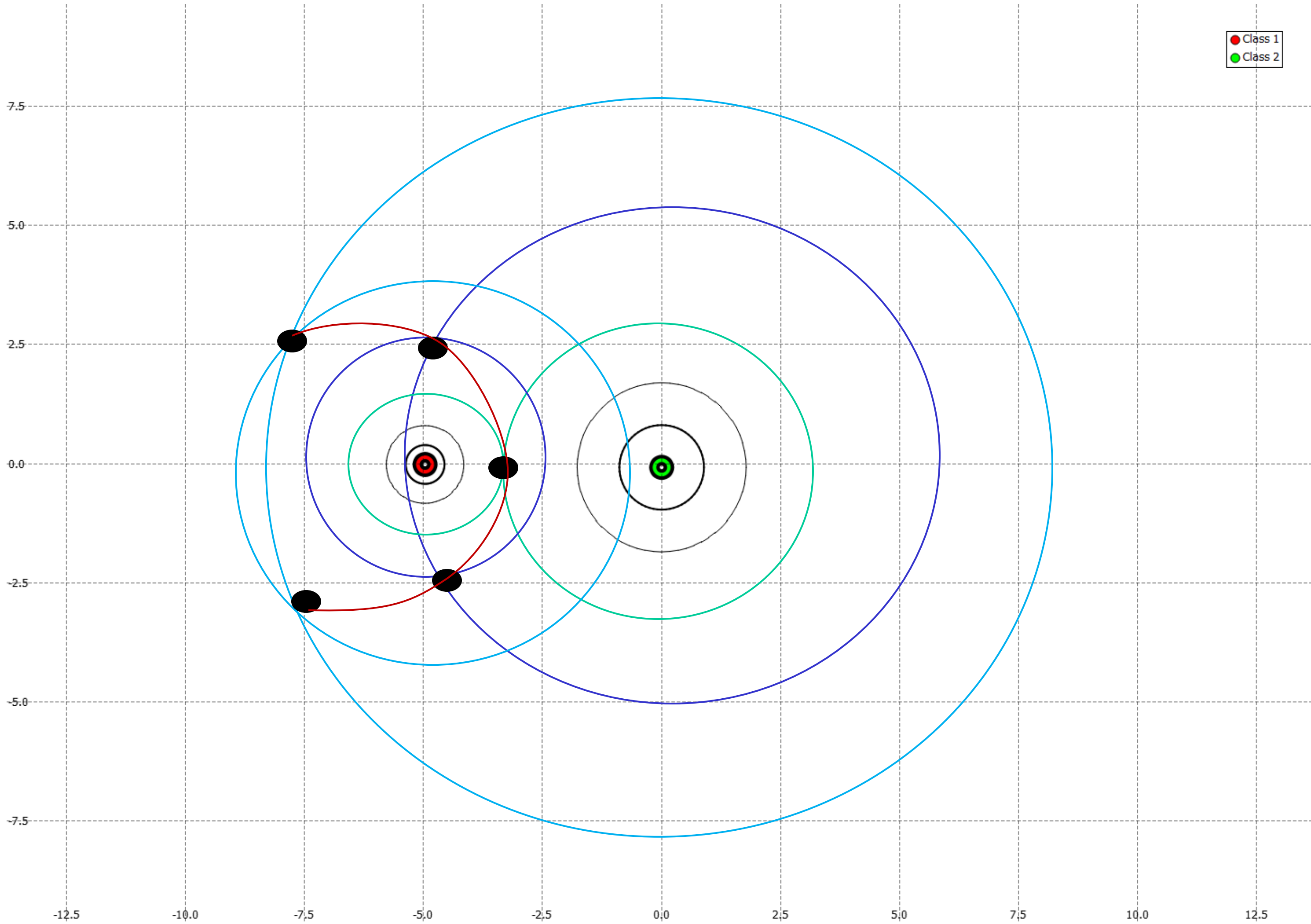


Look for intersection of each Gauss function's isolines of same value

Boundary at points x, s.t.

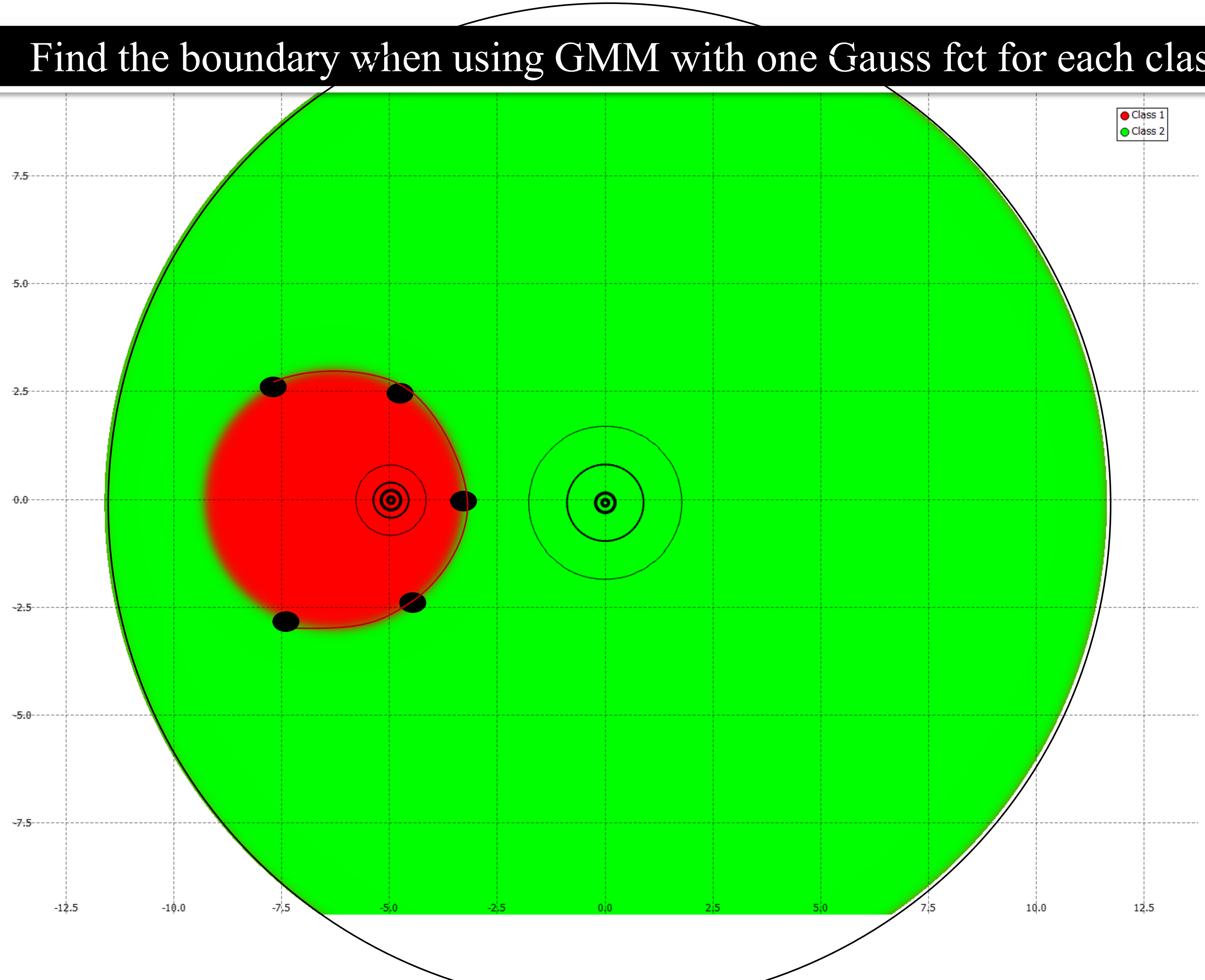
$$\left(x-\mu^1\right)^T \left(\Sigma^1\right)^{-1} \left(x-\mu^1\right) + \log\left|\Sigma^1\right| = \left(x-\mu^2\right)^T \left(\Sigma^2\right)^{-1} \left(x-\mu^2\right) + \log\left|\Sigma^2\right|$$

Exercise: Solution



Exercise: Solution

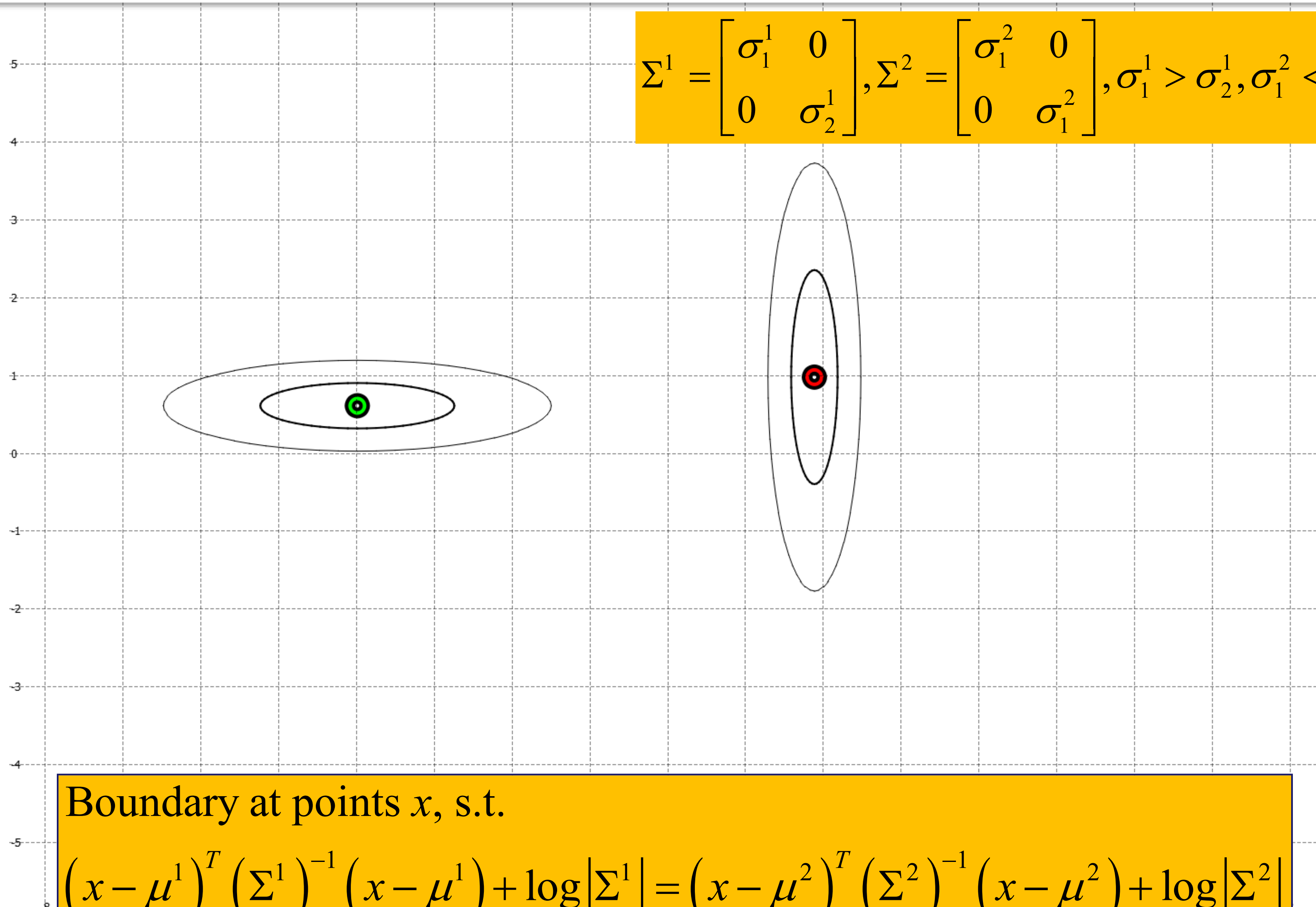
Find the boundary when using GMM with one Gauss fct for each class



Exercise

Find the boundary when using GMM with one Gauss fct for each class

$$\Sigma^1 = \begin{bmatrix} \sigma_1^1 & 0 \\ 0 & \sigma_2^1 \end{bmatrix}, \Sigma^2 = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_1^2 \end{bmatrix}, \sigma_1^1 > \sigma_2^1, \sigma_1^2 < \sigma_2^2,$$

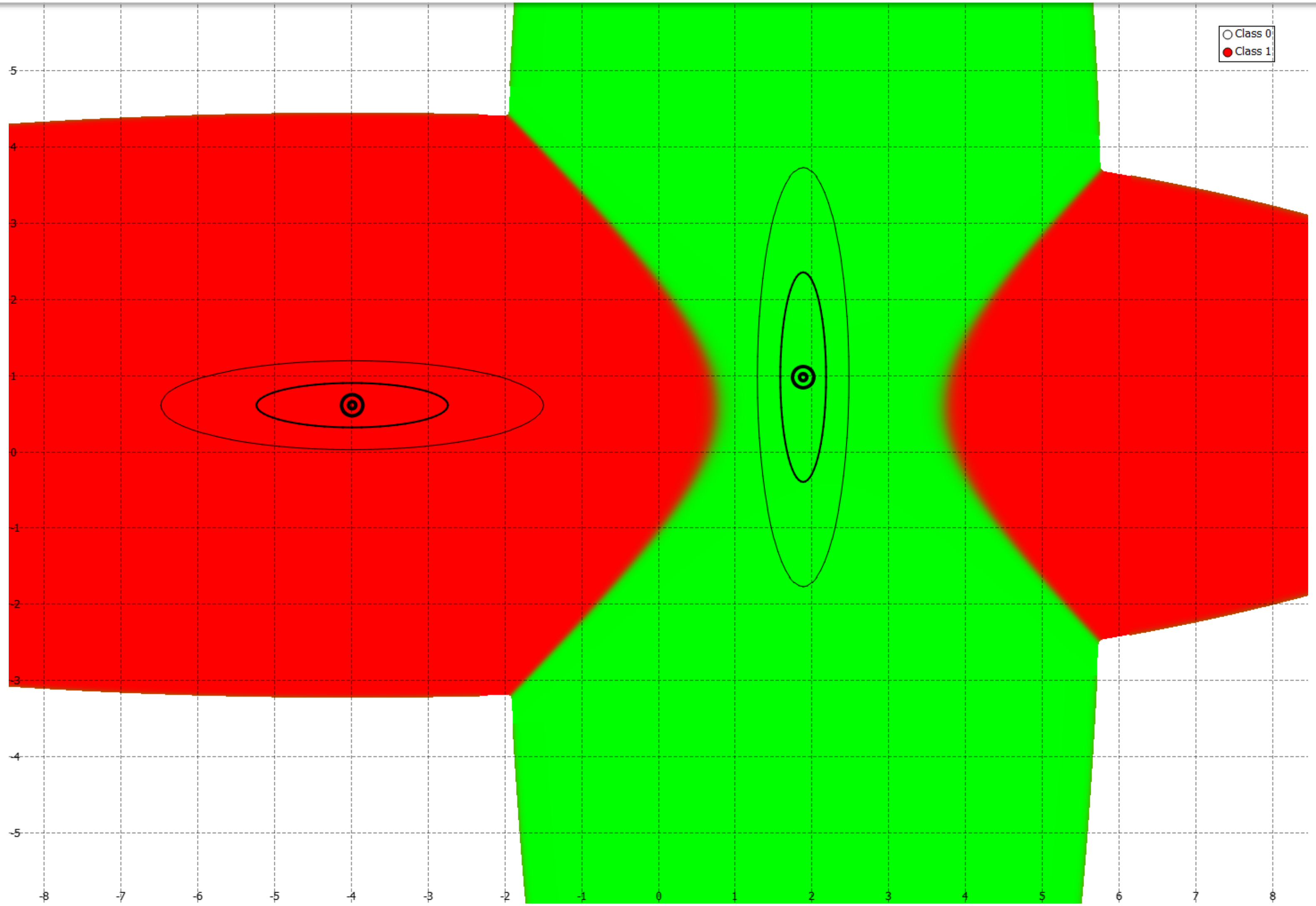


Boundary at points x, s.t.

$$\left(x-\mu^1\right)^T \left(\Sigma^1\right)^{-1} \left(x-\mu^1\right) + \log\left|\Sigma^1\right| = \left(x-\mu^2\right)^T \left(\Sigma^2\right)^{-1} \left(x-\mu^2\right) + \log\left|\Sigma^2\right|$$

Exercise: Solution

Find the boundary when using GMM with one Gauss fct for each class



EPFL

Nonlinearity of the Decision Boundary

Recall: to determine the class label, compute optimal *Bayes classifier.*

$$\boxed{\text{A point } x \text{ belongs to class } C^1 \text{ if } p\left(y=C^1 \mid x\right) > p\left(y=C^2 \mid x\right)}$$

$$\Rightarrow \ln p\left(y=C^1 \mid x\right) > \ln p\left(y=C^2 \mid x\right)$$

Consider the univariate case (1D data) and classes equally likely $p\left(y=C^1\right)=p\left(y=C^2\right)$

$$\frac{(x-\mu^1)^2}{2\left(\sigma^1\right)^2}+\ln\sqrt{2\pi}\sigma^1=\frac{(x-\mu^1)^2}{2\left(\sigma^2\right)^2}+\ln\sqrt{2\pi}\sigma^2$$

$$\frac{x^2-2x\mu^1+\left(\mu^1\right)^2}{2\left(\sigma^1\right)^2}-\frac{x^2-2x\mu^2+\left(\mu^2\right)^2}{2\left(\sigma^2\right)^2}+\ln\sqrt{2\pi}(\sigma^1-\sigma^2)=0$$

$$\boxed{\left(\frac{\left(\sigma^2\right)^2}{2\left(\sigma^1\right)^2\left(\sigma^2\right)^2}-\frac{\left(\sigma^1\right)^2}{2\left(\sigma^1\right)^2\left(\sigma^2\right)^2}\right)x^2-\left(\frac{\left(\sigma^2\right)^2\left(\mu^1\right)^2-\left(\sigma^1\right)^2\left(\mu^2\right)^2}{\left(\sigma^1\right)^2\left(\sigma^2\right)^2}\right)x+\frac{\left(\sigma^2\right)^2\left(\mu^1\right)^2-\left(\sigma^1\right)^2\left(\mu^2\right)^2}{2\left(\sigma^1\right)^2\left(\sigma^2\right)^2}+\ln\sqrt{2\pi}(\sigma^1-\sigma^2)=0}$$

Quadratic equation → Nonlinear boundary

EPFL

Nonlinearity of the Decision Boundary

The decision boundary has the form:

$ax^2 + bx + c = 0 \rightarrow$ in the univariate case
$\mathrm{x}^T A x + b^T x + c = 0 \rightarrow$ in the multivariate case

} Quadratic Discriminant Analysis

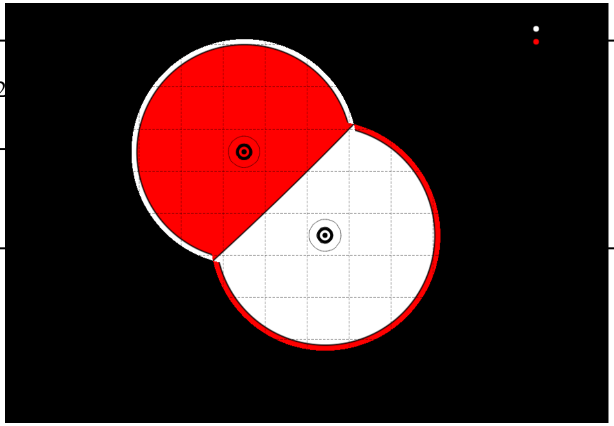
$bx + c = 0 \rightarrow$ in the univariate case
$b^T x + c = 0 \rightarrow$ in the multivariate case

} Linear Discriminant Analysis

Linear equation → Linear boundary

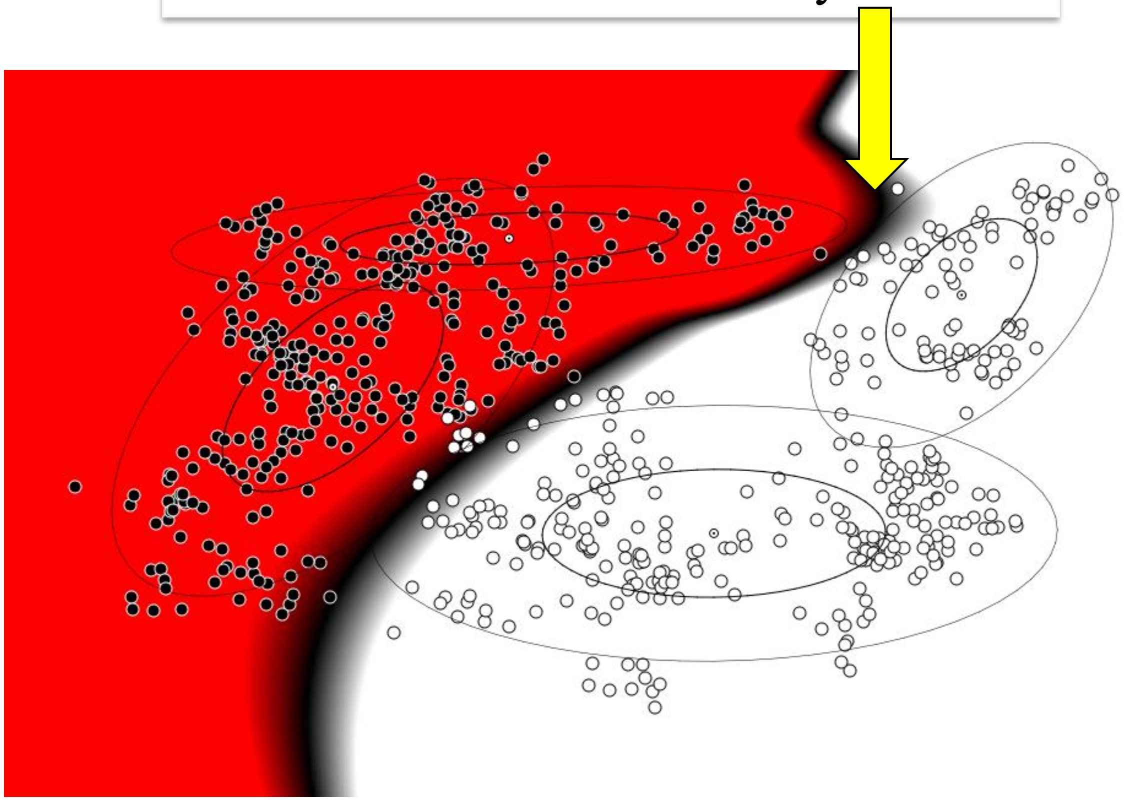
In the case where $\sigma_1^2 = \sigma_2^2$ or $\Sigma^1 = \Sigma^2$ for the multivariate case

$$\underbrace{\left(\frac{(\sigma^2)^2}{2(\sigma^1)^2(\sigma^2)^2} - \frac{(\sigma^1)^2}{2(\sigma^1)^2(\sigma^2)^2} \right)}_{0} x^2 - \left(\frac{(\sigma^2)^2(\mu^1)^2 - (\sigma^1)^2(\mu^2)^2}{(\sigma^1)^2(\sigma^2)^2} \right) x + \frac{(\sigma^2)^2}{} \quad = 0$$



Classification with multiple Gauss fcts (GMM-s)

Where is the boundary?



Example of binary classification using
2 Gaussian Mixture Models with 2 Gauss functions each

Maximum Likelihood Discriminant Rule for Multi-Class Classification

The maximum likelihood (ML) discriminant rule predicts the class of an observation x using:

$$c(x) = \arg\max_{k=1\ldots K} p_k(x) \qquad \text{for } K \text{ classes}$$

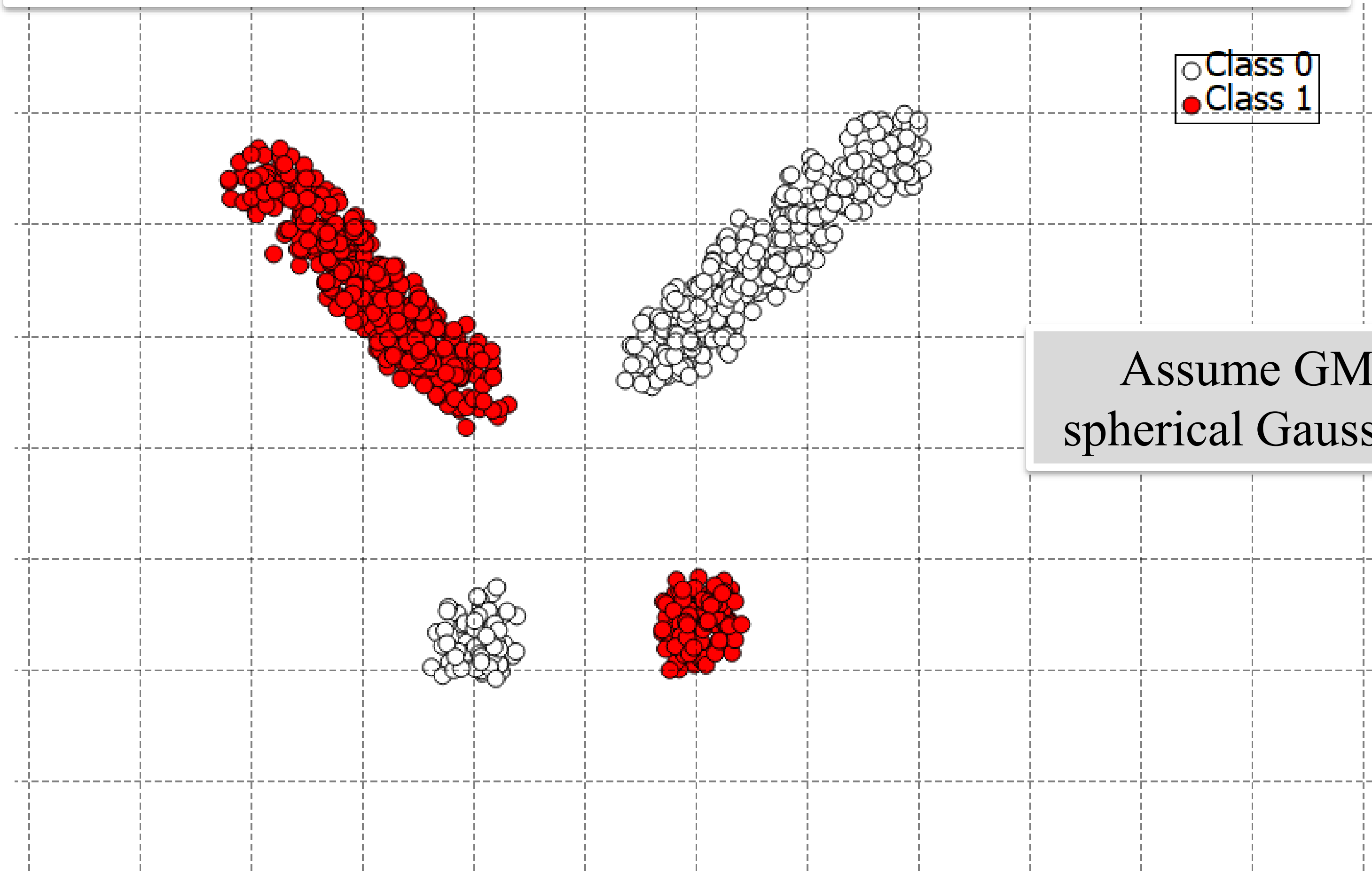
ML discriminant rule is minimum of minus the log-likelihood (equiv. to maximizing the likelihood):

$$C^k(x) = \arg\min_k \left\{ \left(x-\mu^k\right)\left(\Sigma^k\right)^{-1}\left(x-\mu^k\right)^T + \log\left|\Sigma^k\right| \right\}$$

Even though there are two classes, the groups being distinct, the boundaries can be thought of as a 4-class classification problem.

Exercise

Find the boundaries for the two distributions below

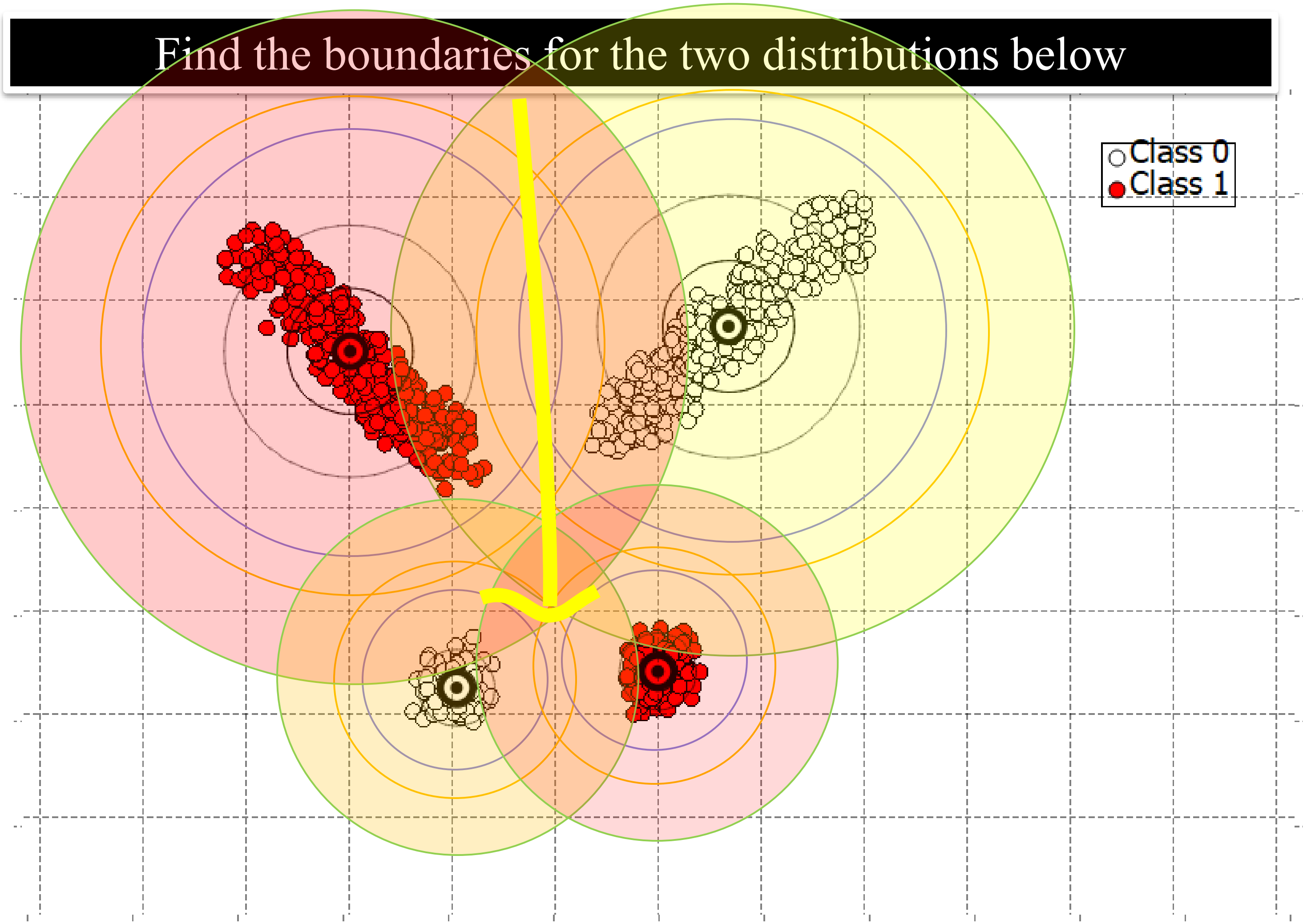


Assume GMM with spherical Gauss functions

$$C^k(x) = \arg\min_k \left\{ \left(x-\mu^k\right)\left(\Sigma^k\right)^{-1}\left(x-\mu^k\right)^T + \log\left|\Sigma^k\right| \right\}$$

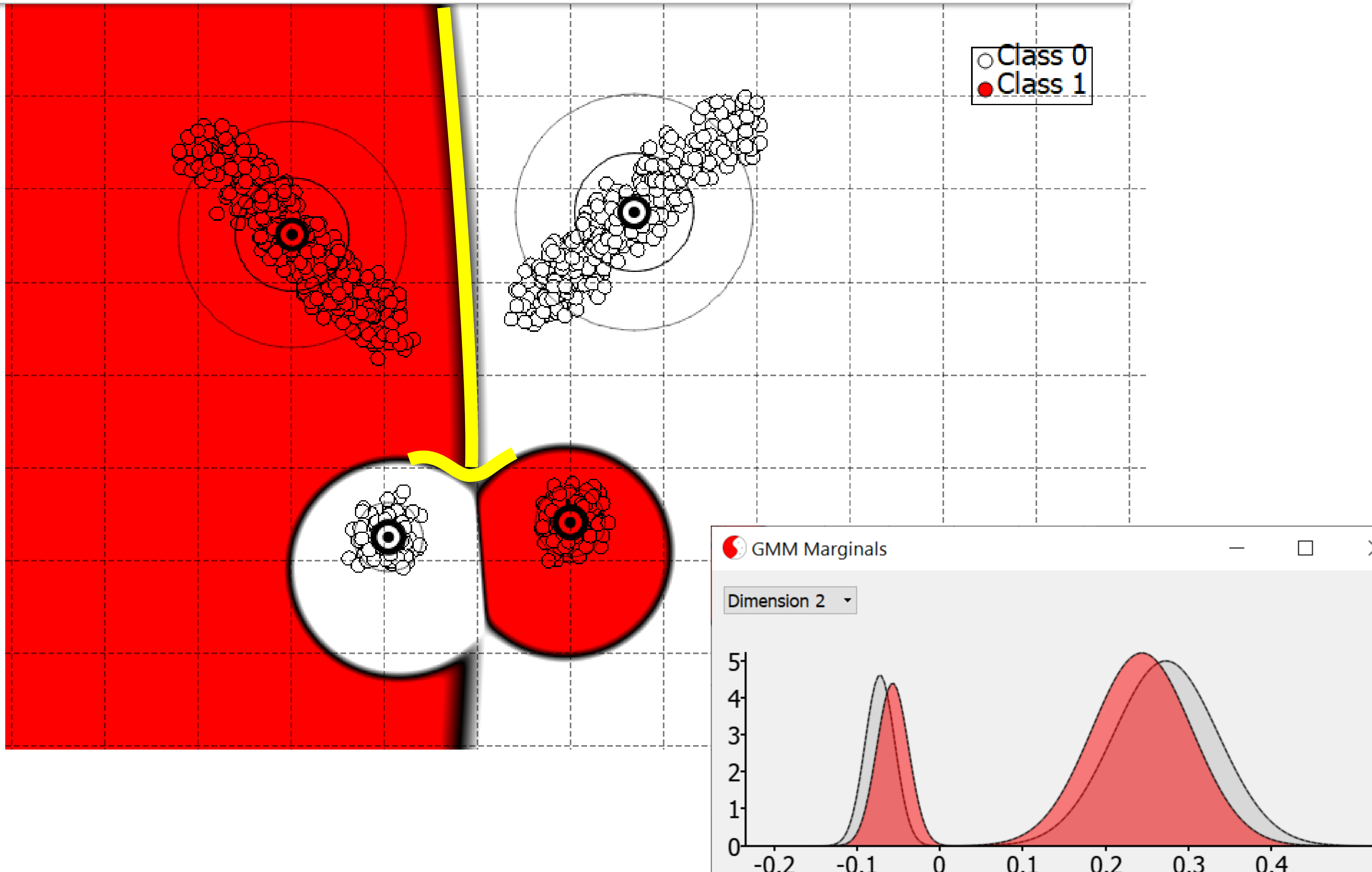
Exercise

Find the boundaries for the two distributions below



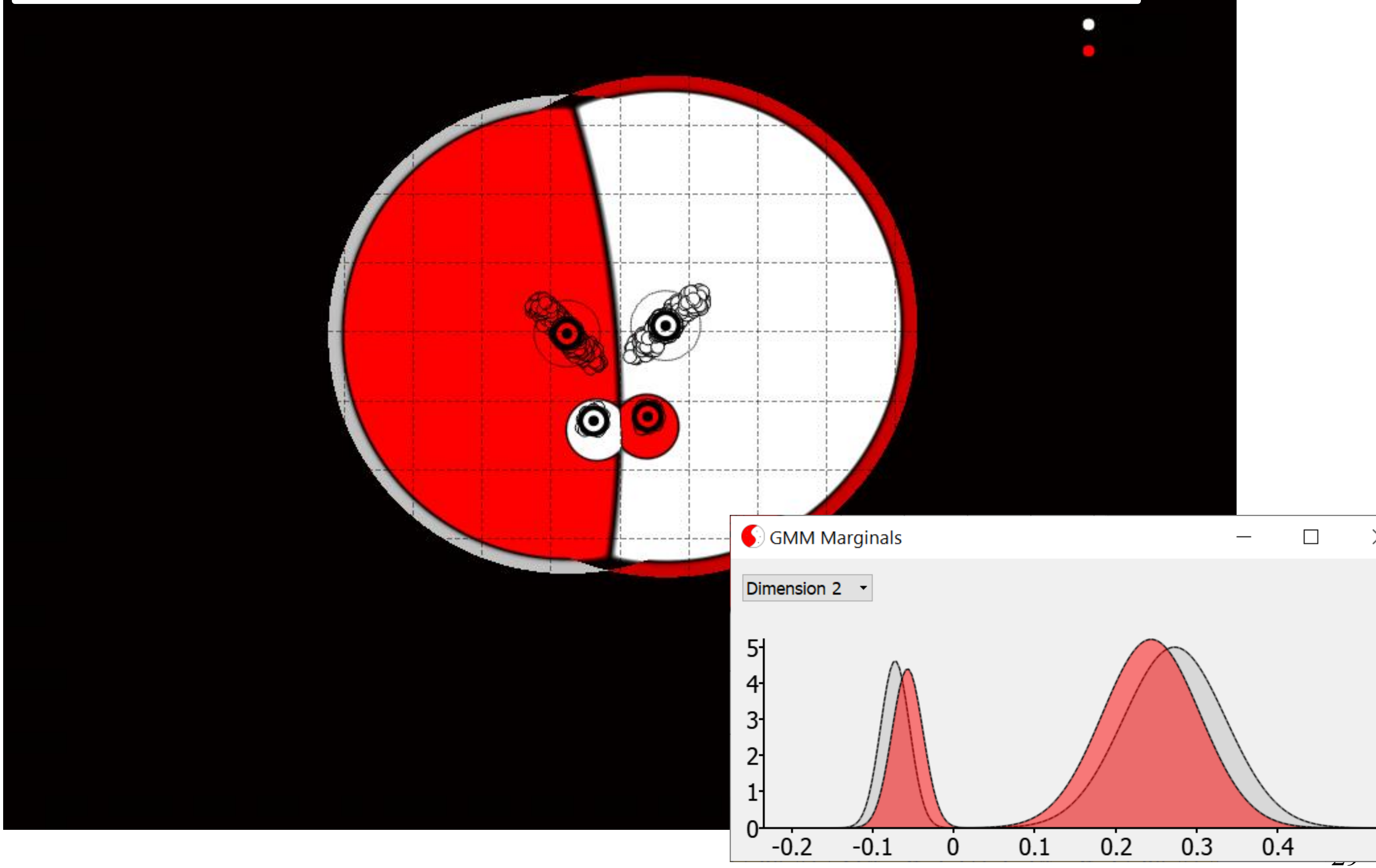
Exercise

Find the boundaries for the two distributions below



Exercise

Find the boundaries for the two distributions below



EPFL

Gaussian ML Discriminant Rule and LDA

When all class densities have the same covariance matrix, $\Sigma^k = \Sigma$, the discriminant rule is linear. This can be turned into a single optimization problem for supervised learning when the class labels are known. This is known as Linear discriminant analysis (LDA).

$$0 < \left(x-\mu^1\right)(\Sigma)^{-1}\left(x-\mu^1\right)^T < \left(x-\mu^2\right)(\Sigma)^{-1}\left(x-\mu^2\right)^T$$

or equivalently

$$c_k(x) = \arg\min_{k=\{1,2\}} \left\{\left(x-\mu^k\right)(\Sigma)^{-1}\left(x-\mu^k\right)^T\right\}$$

LDA can be used as projection technique, like PCA, using as objective function the discriminant rule.

It finds the projection that separates best the two classes.

See Lecture Notes for description of LDA.

LDA

- Linear Discriminant Analysis (LDA) **combines concepts of PCA and clustering** to determine projections along which a dataset can be best separated into two distinct classes.
- LDA aims to find the optimal transformation A such that the class structure of the original high-dimensional space is preserved in the low dimensional space

$$A : x_i \in \mathbb{R}^N \rightarrow y_i = Ax_i \in \mathbb{R}^q \left(q \leq N\right)$$

- LDA finds the matrix A solution of the following optimization (Sw and Sb measure the within-class and between-class distribution respectively).

$$J\left(A\right) = \frac{\left|A^T S_b A\right|}{\left|A^T S_w A\right|}$$

$$S_w = \sum_{k=1}^{K} \sum_{x \in C^k} \left(x - \mu^k\right)\left(x - \mu^k\right)^T$$

$$S_b = \sum_{k=1}^{K} n^k \left(\mu^k - \bar{\mu}\right)\left(\mu^k - \bar{\mu}\right)^T$$

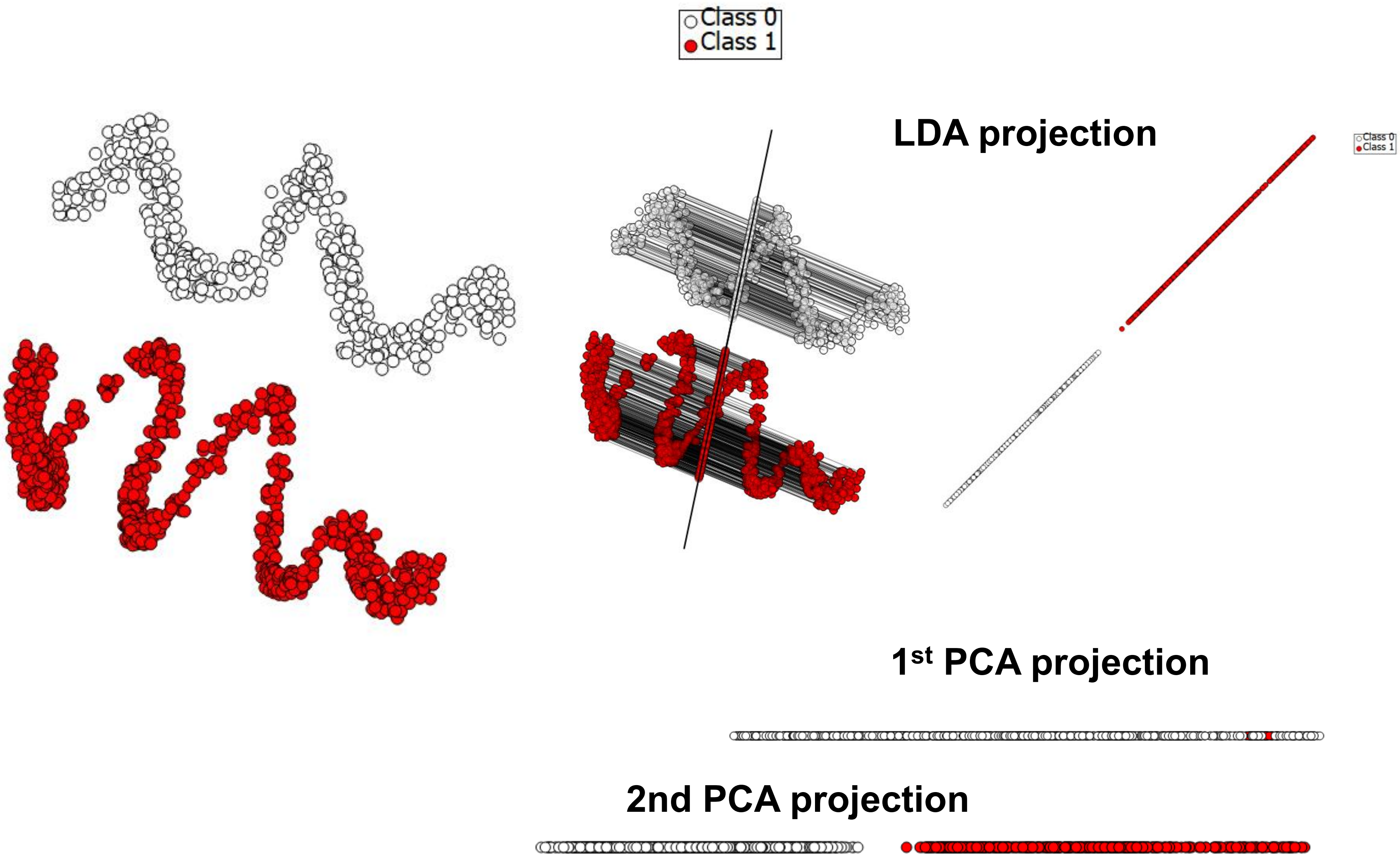
- As for PCA, the solution is found **analytically as an eigenvalue problem**

$$S_b x = \lambda S_w x, \quad \lambda \neq 0$$

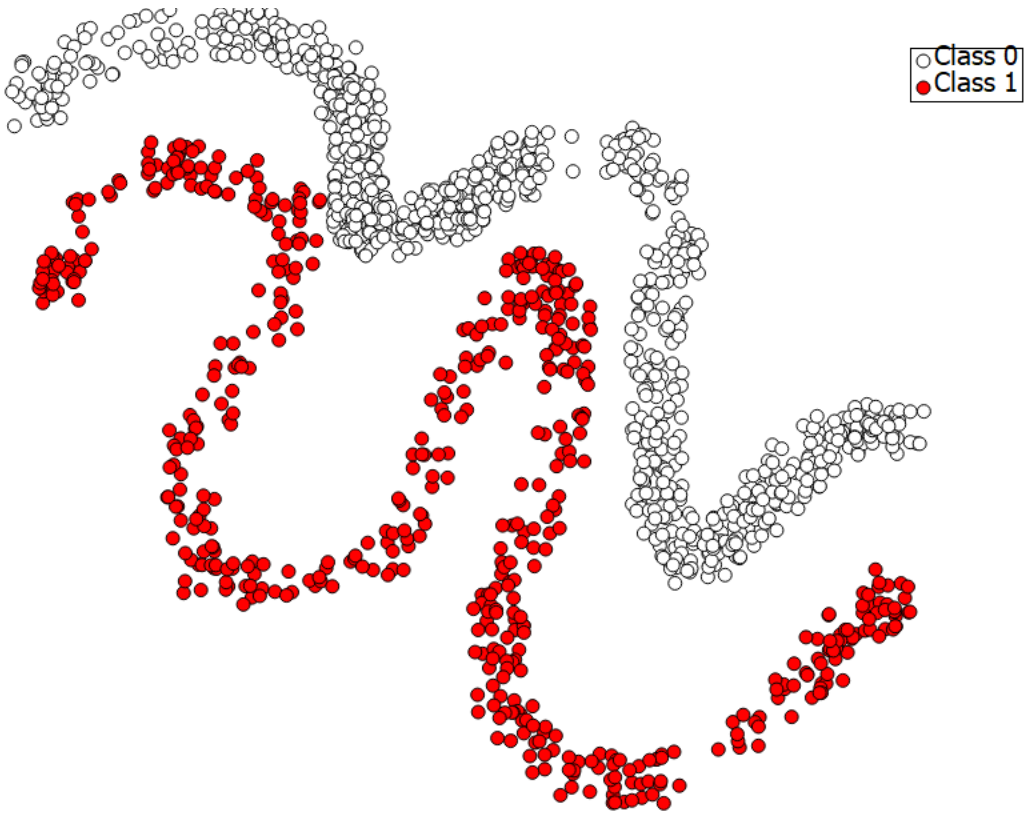
- LDA assumes that **all classes have equal class covariance** (otherwise, the elements of the within-class matrix should be normalized by the covariance on the set of data

See Lecture Notes for description of LDA.

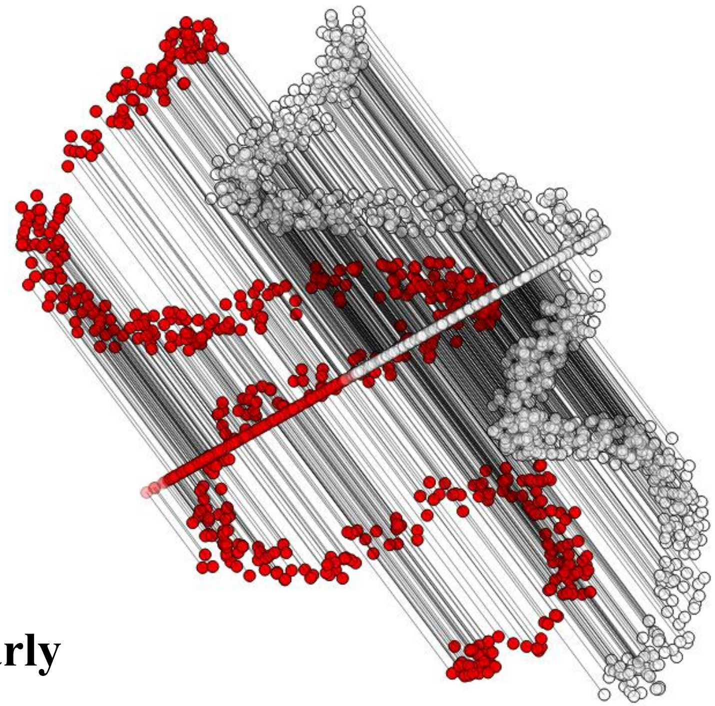
Gaussian ML Discriminant Rule and LDA



Gaussian ML Discriminant Rule and LDA

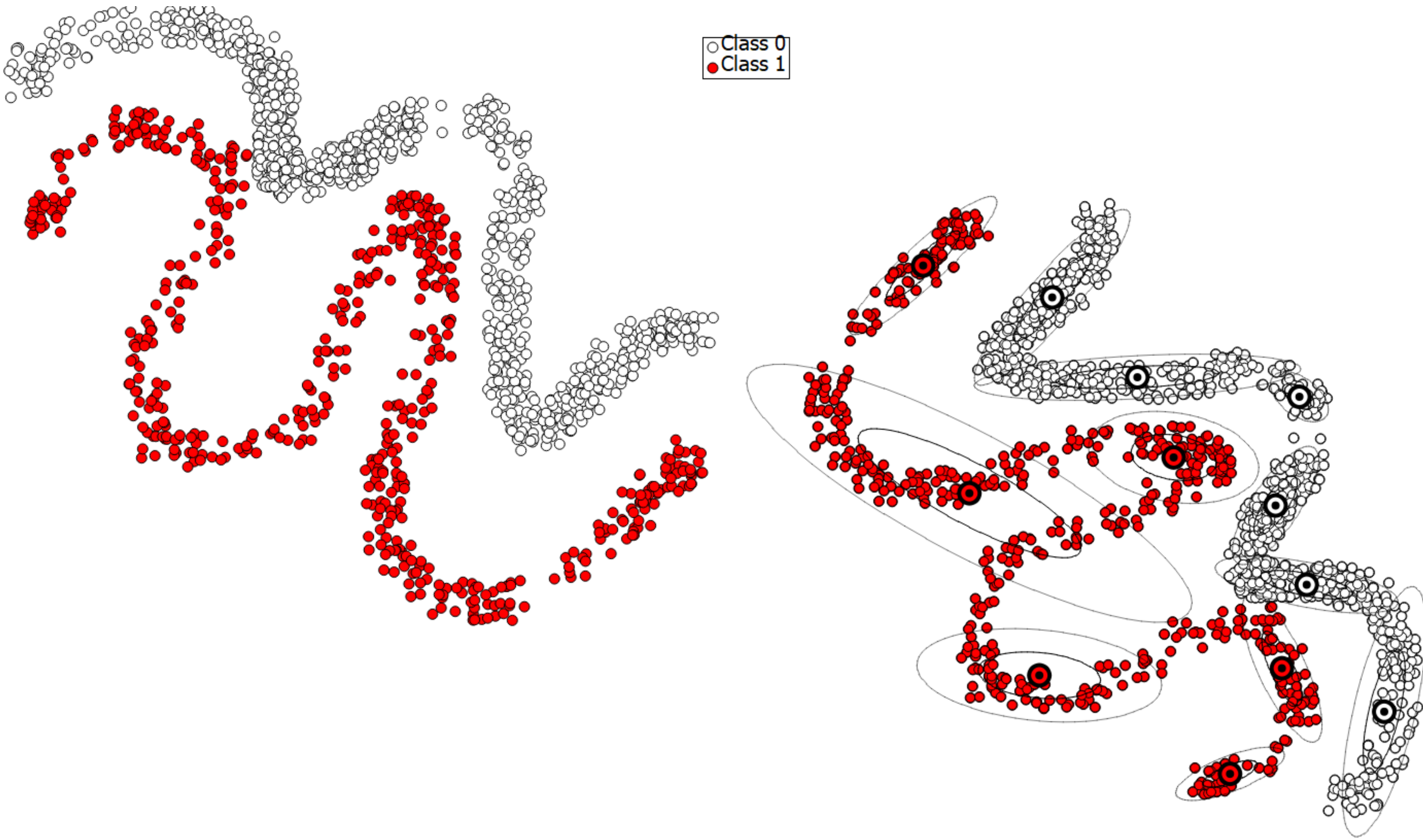


LDA projection

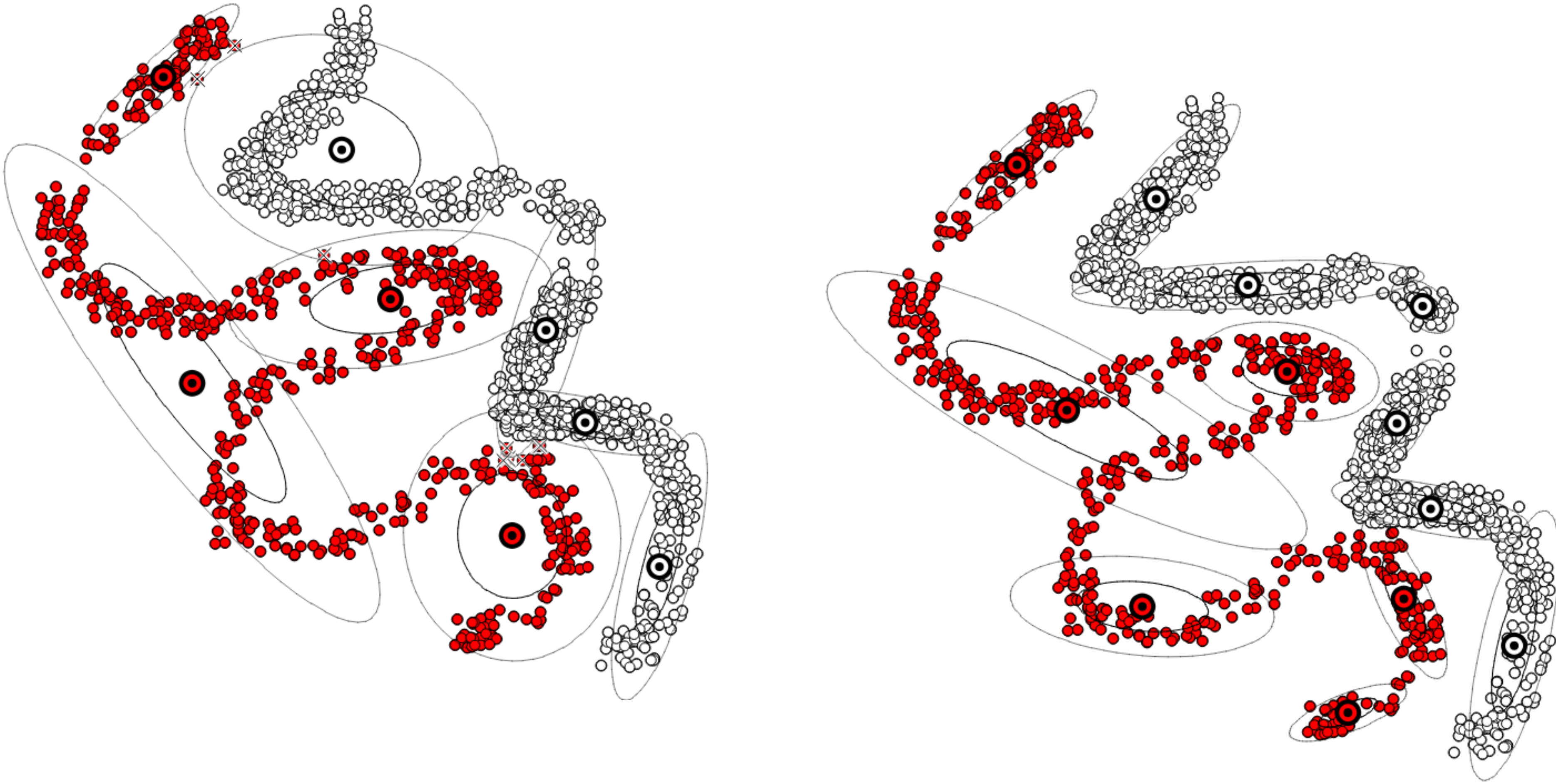


LDA fails when the classes cannot be separated linearly

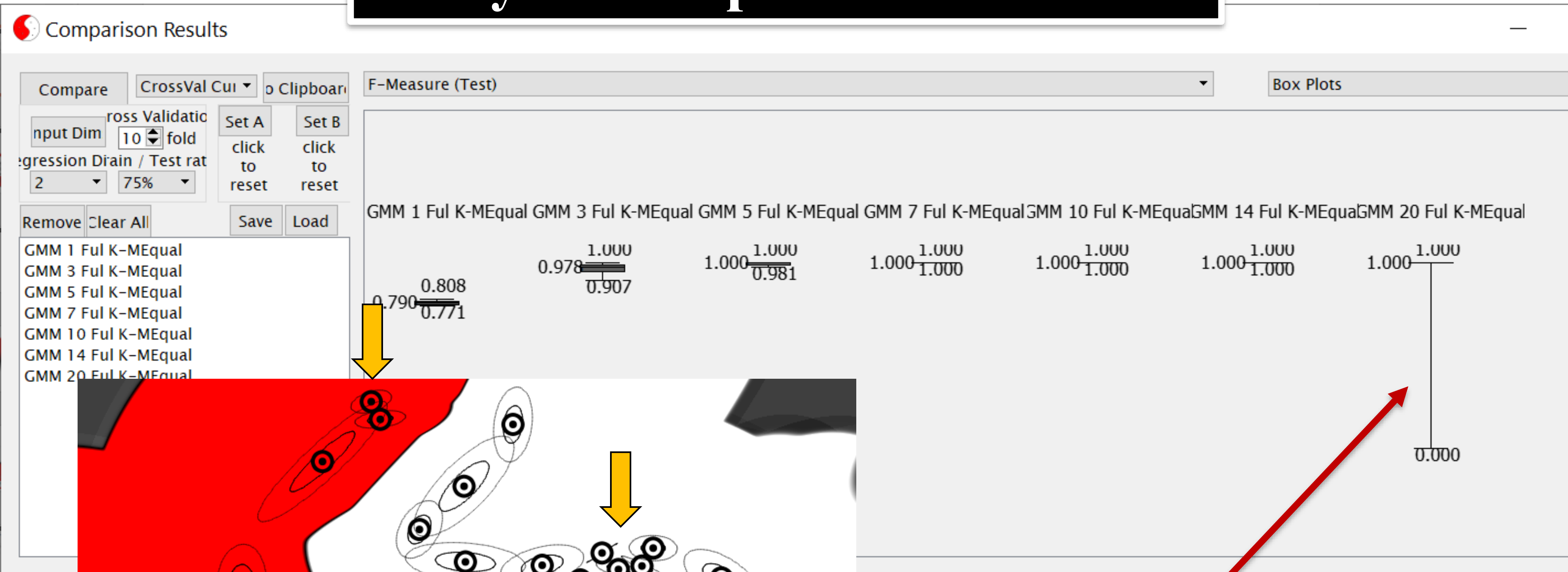
GMM with multiple Gauss functions for each class

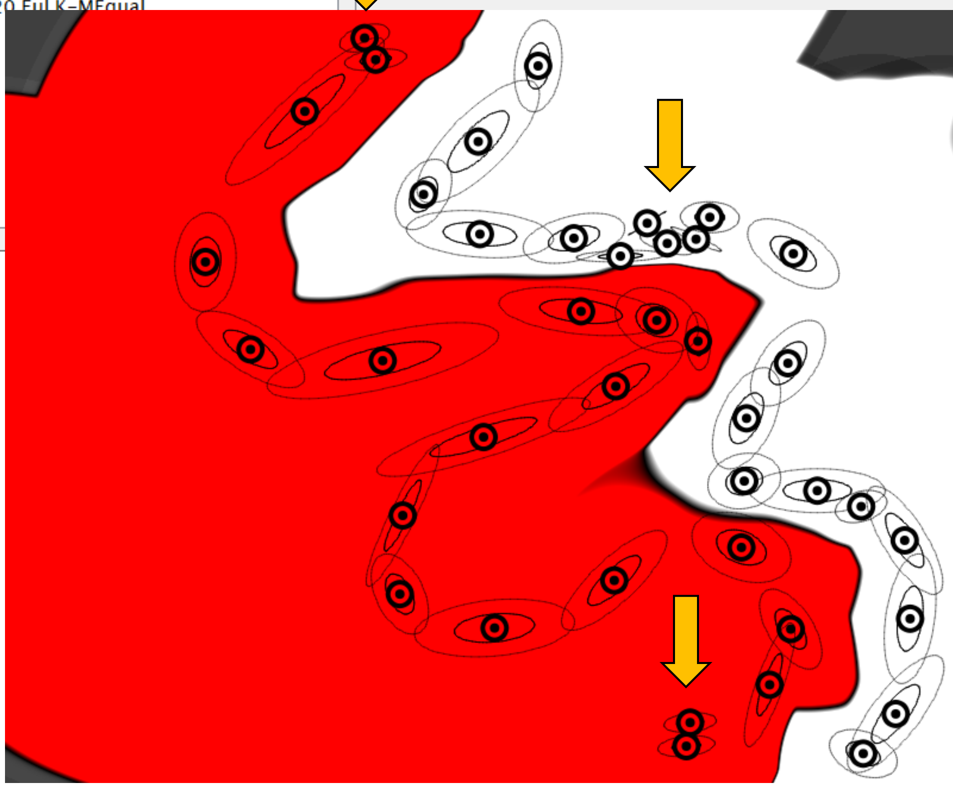


Which model?



Can you interpret these results?

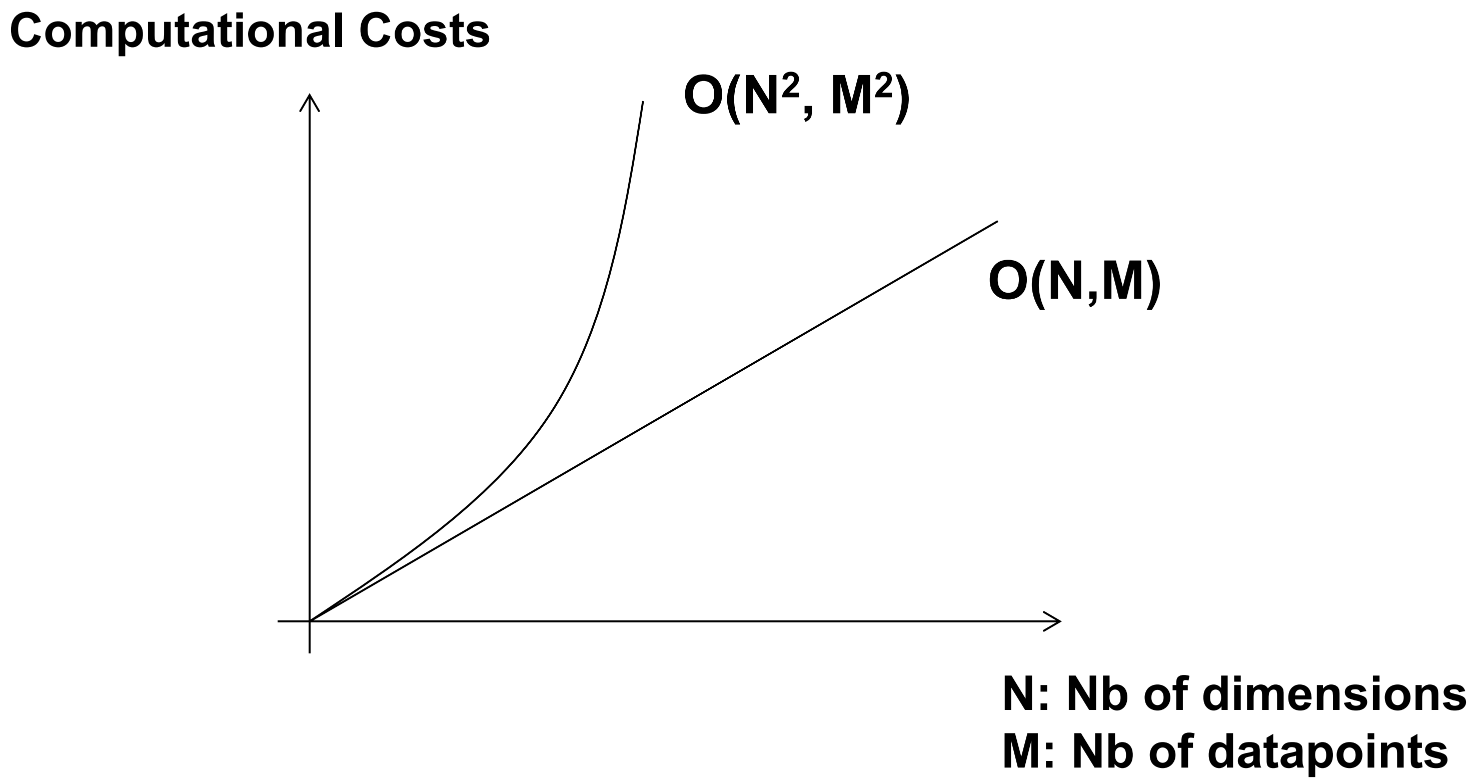




What happened?

Fits single Gauss fct with too few datapoints

Curse of Dimensionality

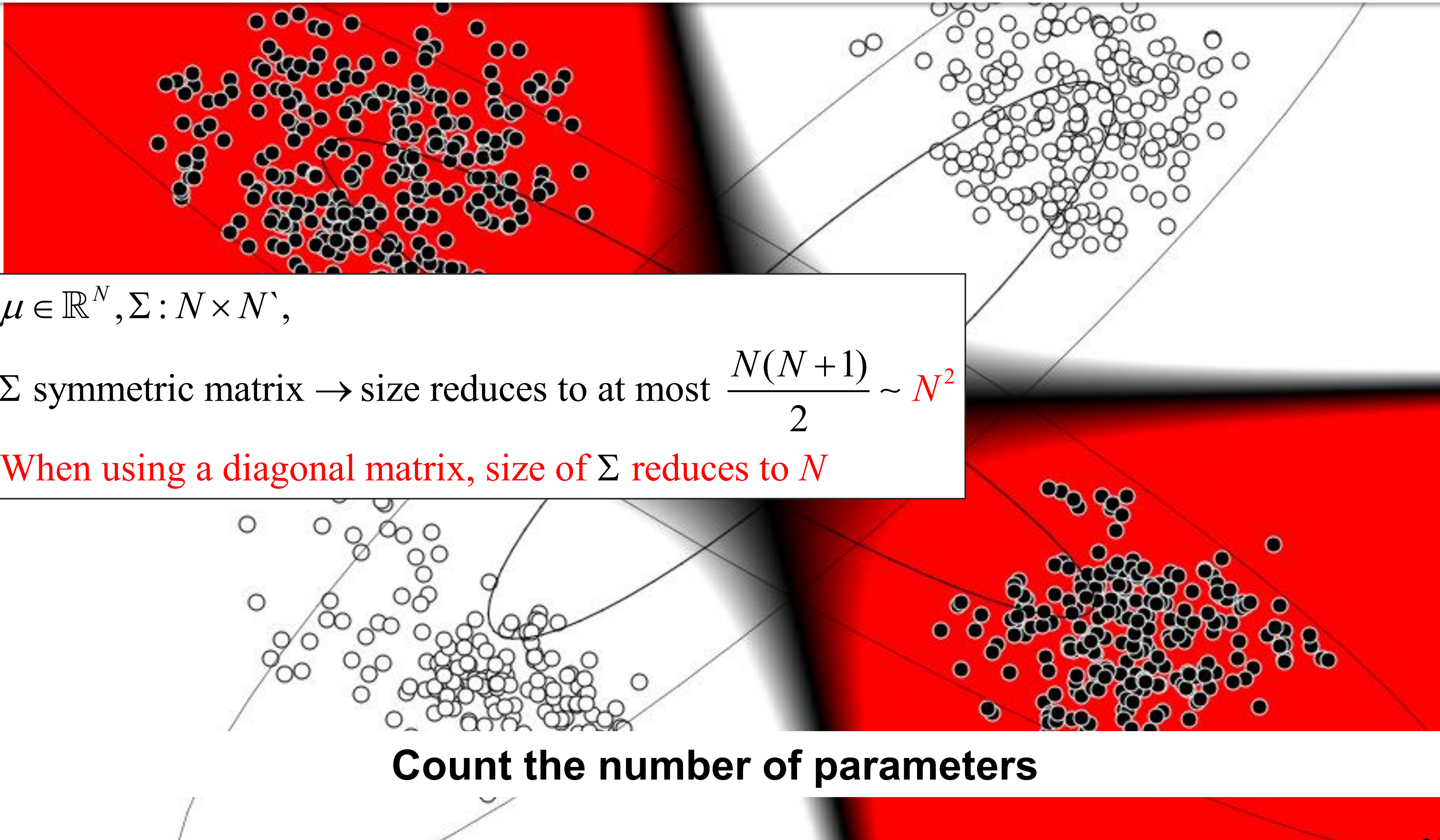


Computational costs may grow as a function of number of dimensions or of number of datapoints

Example : Classification with 2 GMMs

(1 Gaussian per model, full covariance matrix)

Computational costs in GMM grow quadratically with N and linearly with M at training and quadratically with N at testing



$\mu \in \mathbb{R}^N, \Sigma : N \times N`,$

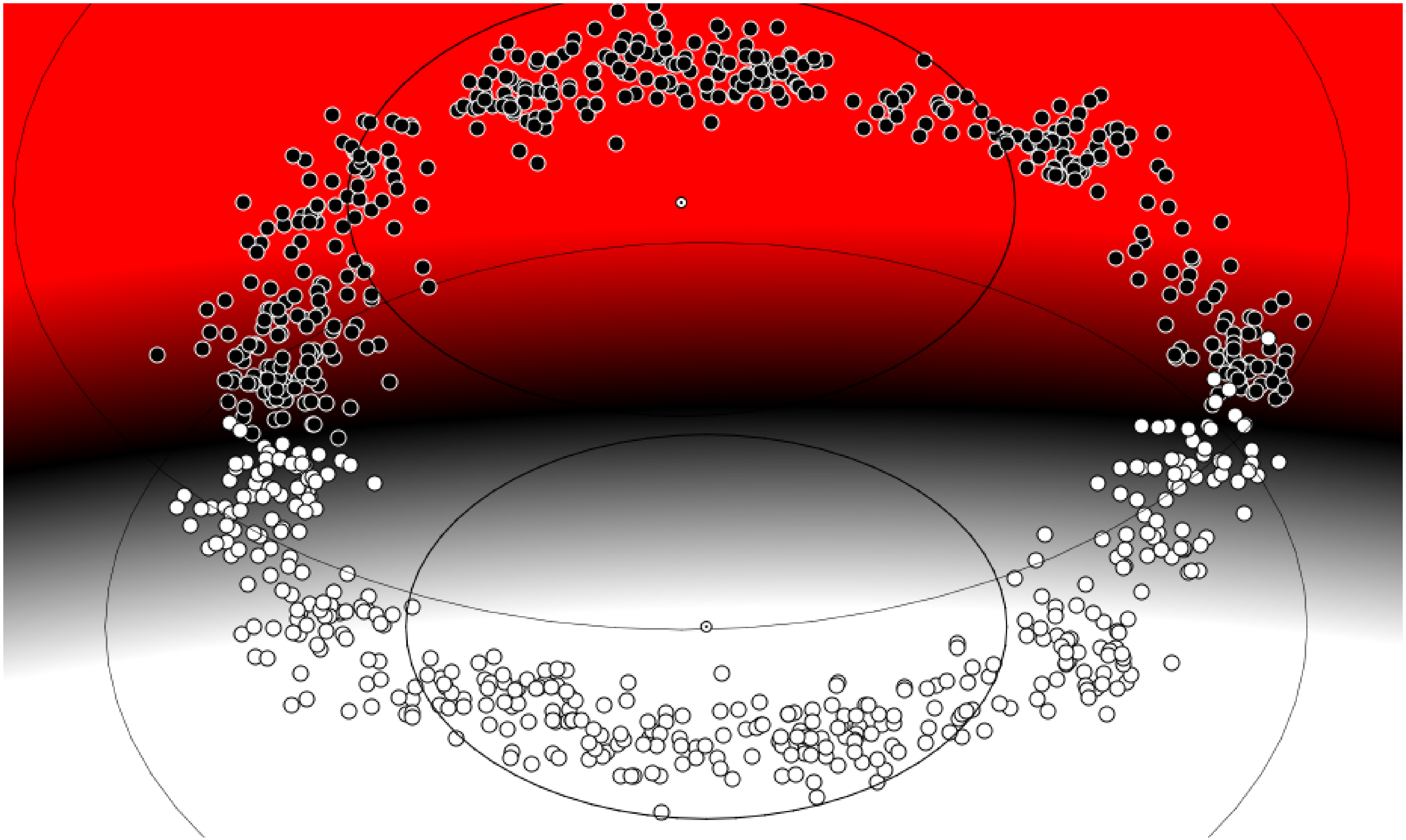
Σ symmetric matrix $\rightarrow$ size reduces to at most $\dfrac{N(N+1)}{2} \sim N^2$

When using a diagonal matrix, size of Σ reduces to N

Count the number of parameters

Example : Classification with 2 GMMs

(1 Gaussian per model, spherical covariance matrix)



Count the number of parameters

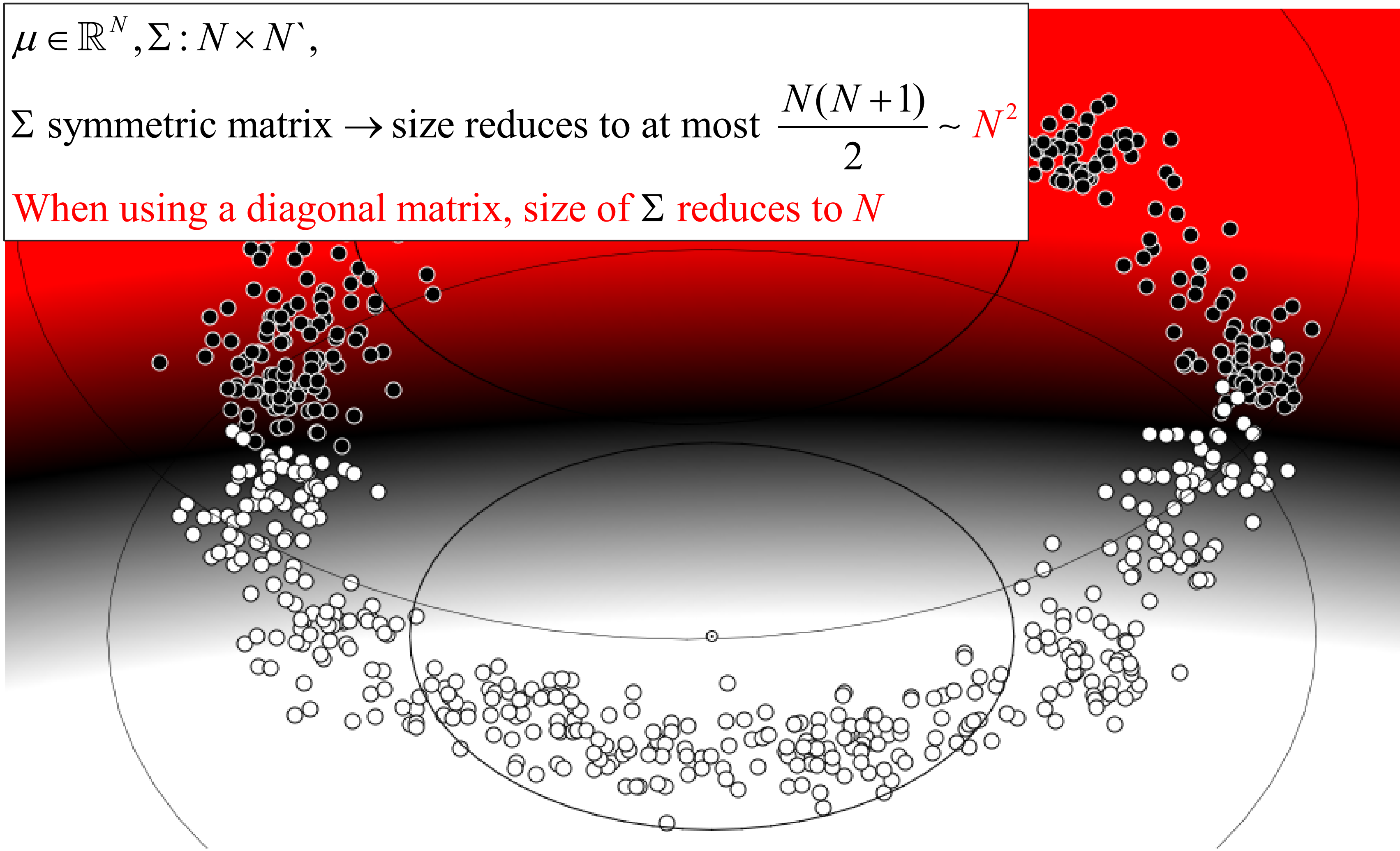
Example : Classification with 2 GMMs

(1 Gaussian per model, spherical covariance matrix)

$\mu \in \mathbb{R}^N, \Sigma : N \times N`,$

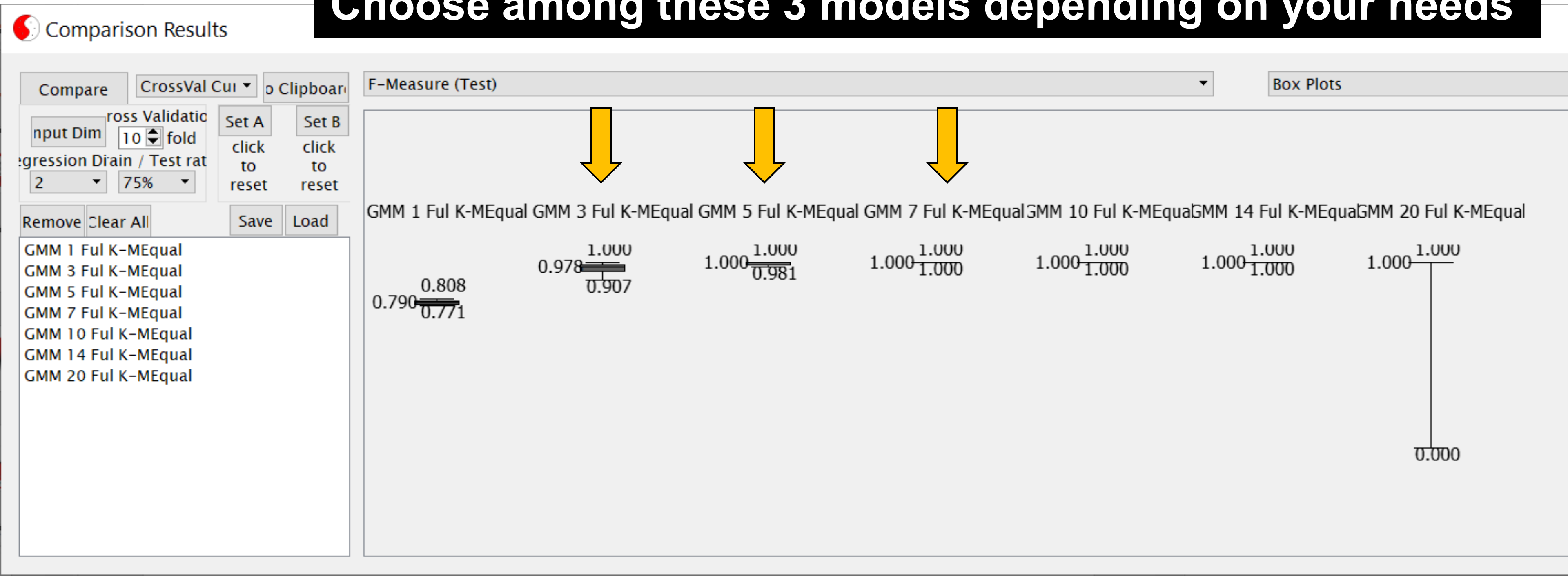
Σ symmetric matrix $\rightarrow$ size reduces to at most $\dfrac{N(N+1)}{2} \sim N^2$

When using a diagonal matrix, size of Σ reduces to N

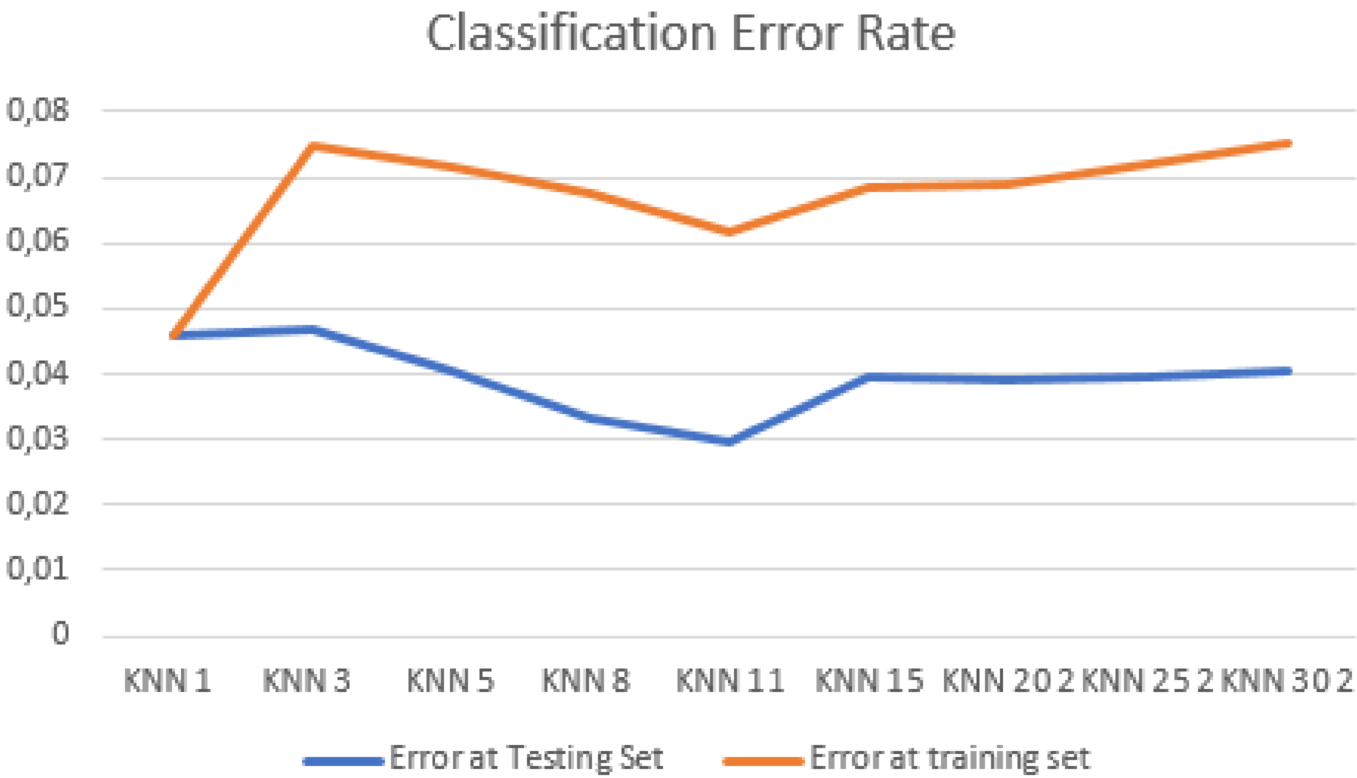


Count the number of parameters

Choose among these 3 models depending on your needs



Classification Error on training and testing sets

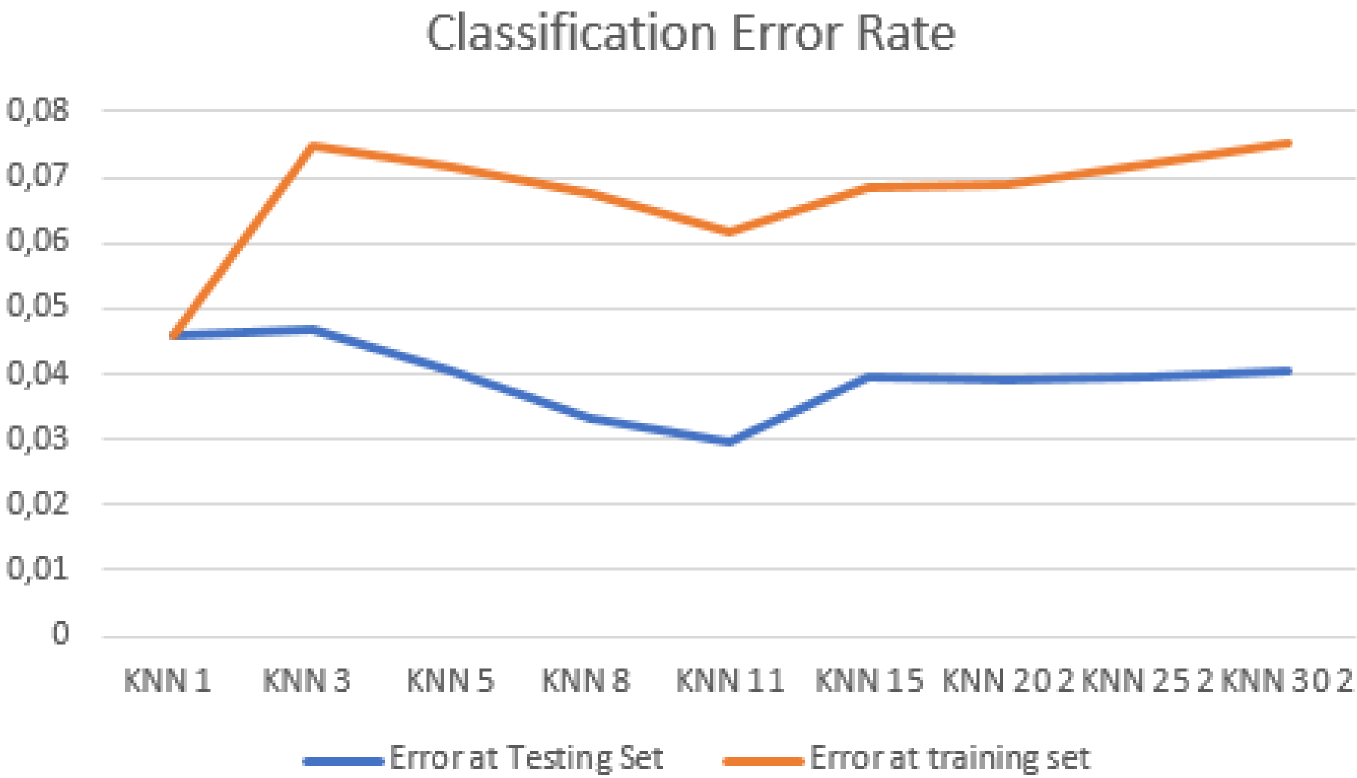


Which of these three combination is overfitting?

A. a
B. b
C. c

	Training Error	Testing Error
a	Low	High
b	Low	Low
c	High	High

Classification Error on training and testing sets



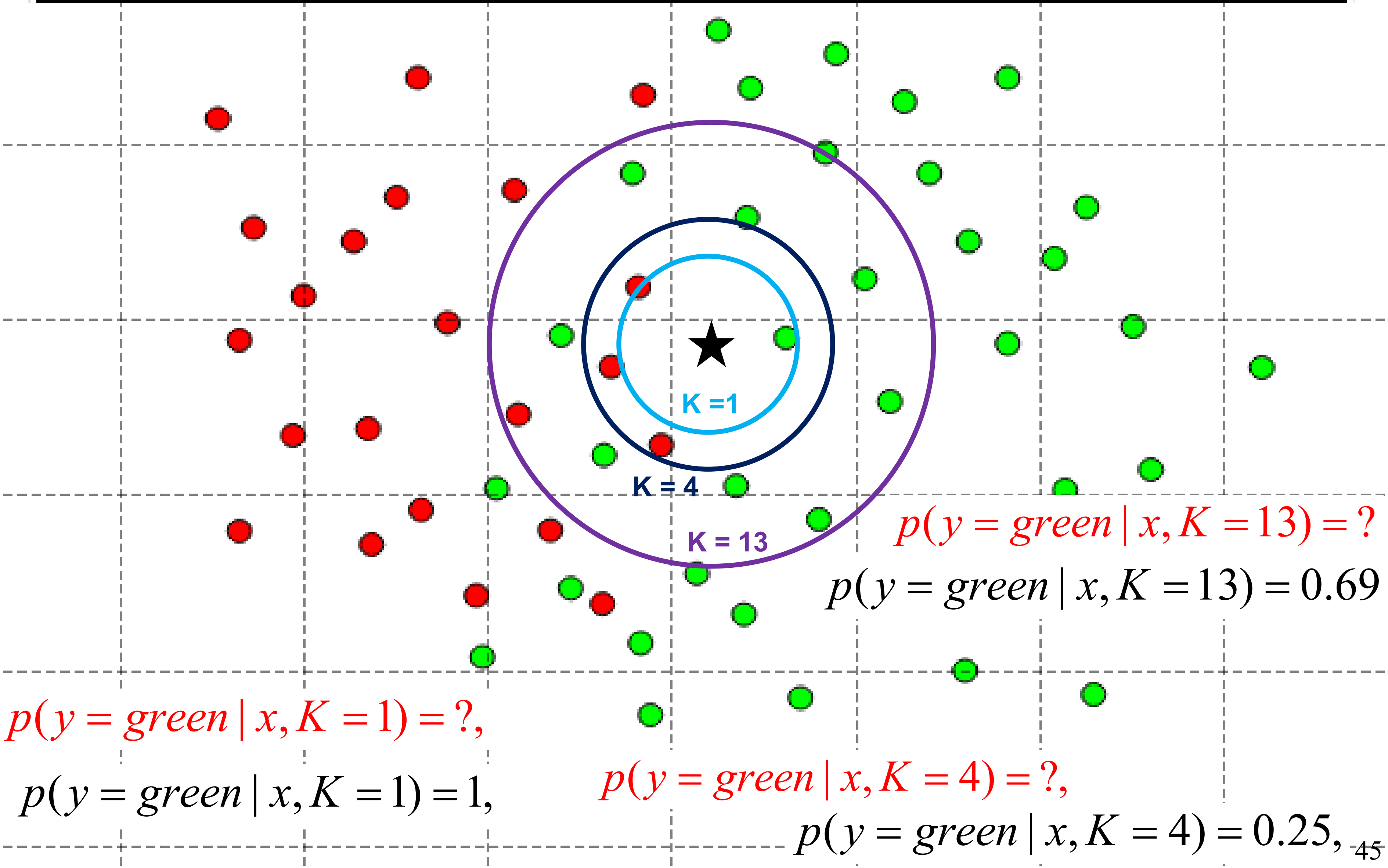
Which of these three combination is overfitting?

	Training Error	Testing Error
Overfitting	Low	High
Good fit	Low	Low
Underfitting	High	High

Crossvalidation & Choice of training/testing ratio

- Avoid **overfitting** (i.e. fitting too well all datapoints including noise)
  - → Train the classifier with a small sample of all datapoints
  - → Test the classifier with the remaining datapoints.
- Typical choice of training/testing set ratio is 2/3rd training, 1/3rd testing.
- The smaller the ratio, the more robust the classification
- **Several-fold crossvalidation**
- Typical choice is 10-fold crossvalidation. However, this depends on how many datapoints you have in your dataset!

Classification with K- Nearest Neighbors (K-NN)



$p(y = green \mid x, K = 13) = ?$

$p(y = green \mid x, K = 13) = 0.69$

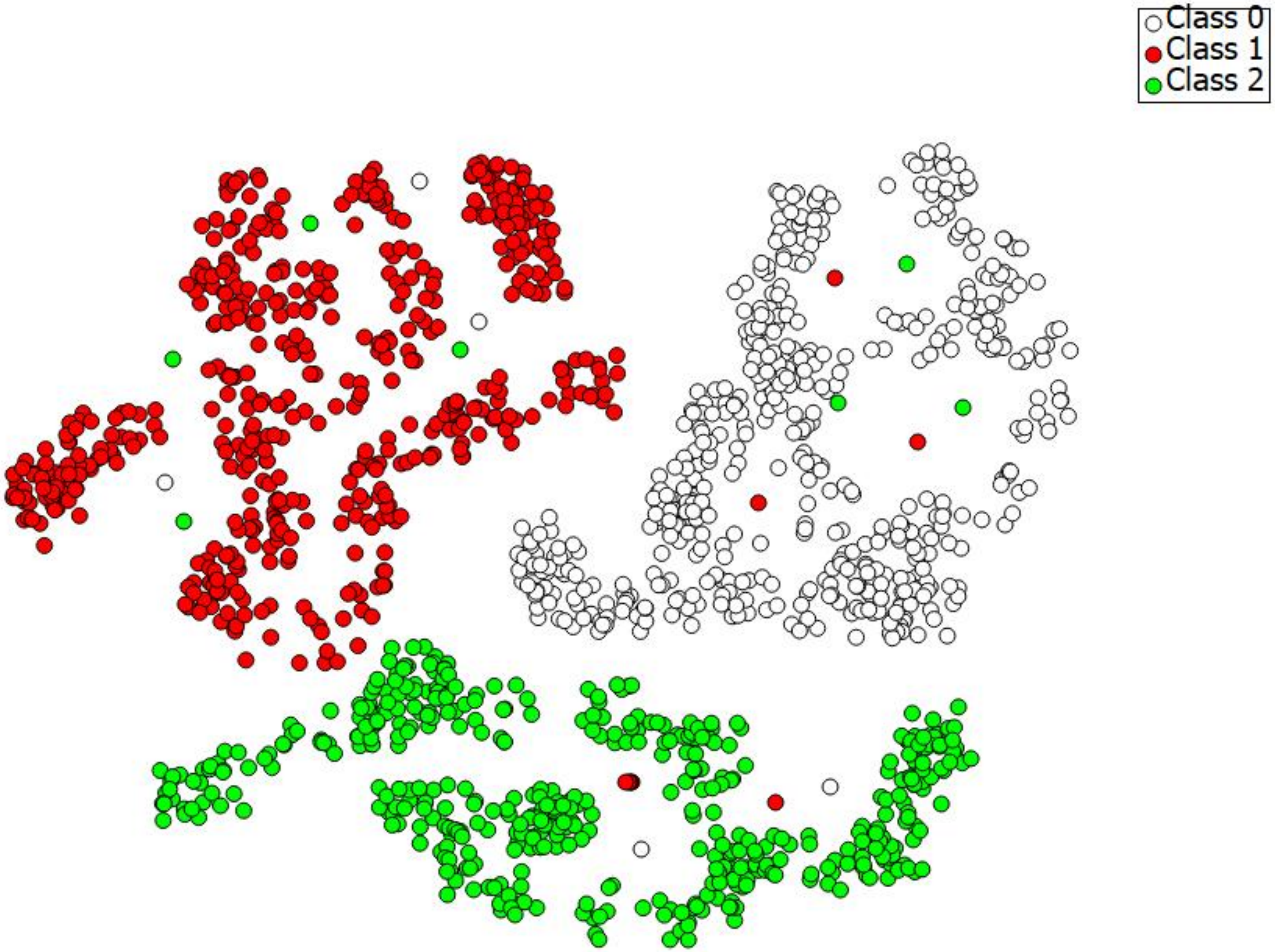
$p(y = green \mid x, K = 1) = ?,$

$p(y = green \mid x, K = 1) = 1,$

$p(y = green \mid x, K = 4) = ?,$

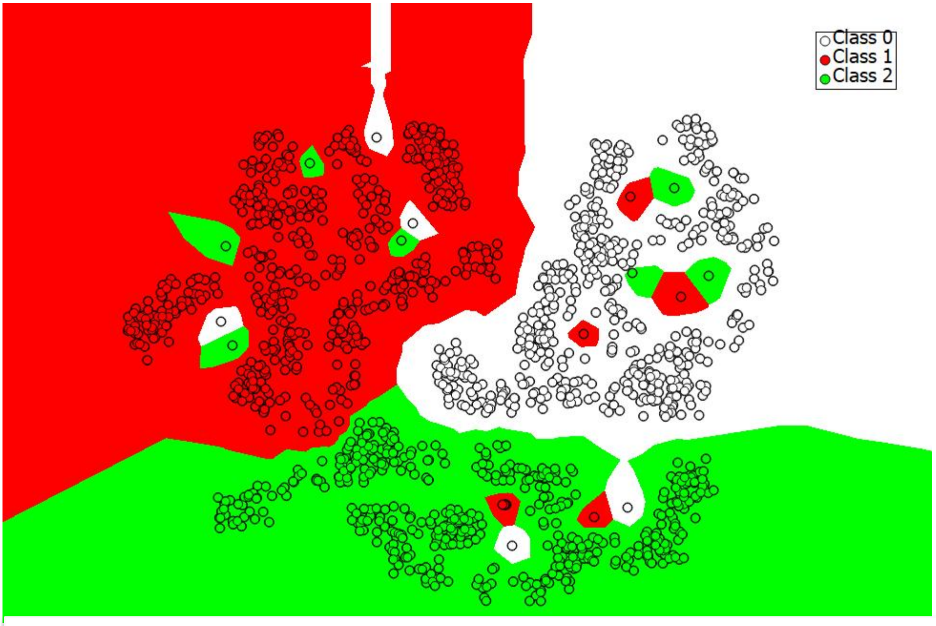
$p(y = green \mid x, K = 4) = 0.25,$

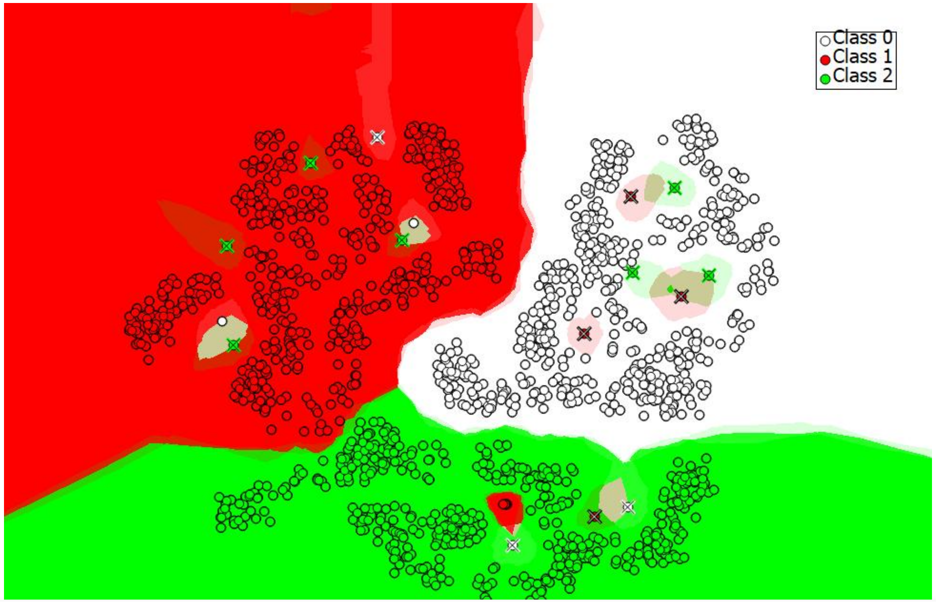
Draw the boundaries found by KNN for different K



K=1, K=3, K=20

Draw the boundaries found by KNN for different K

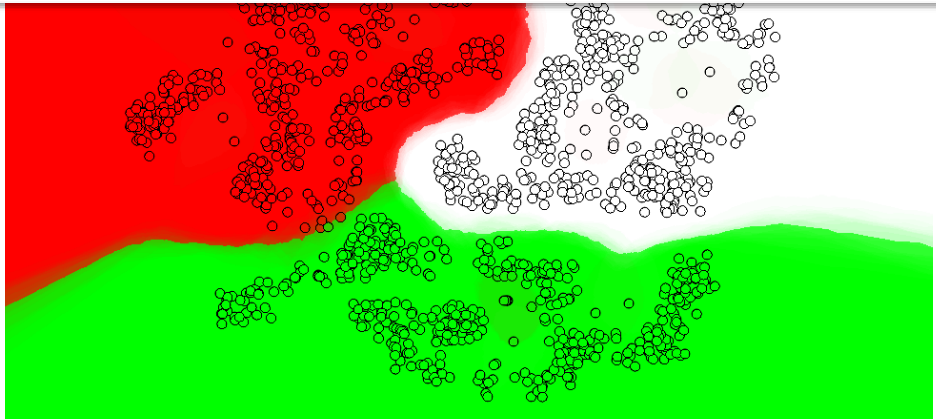




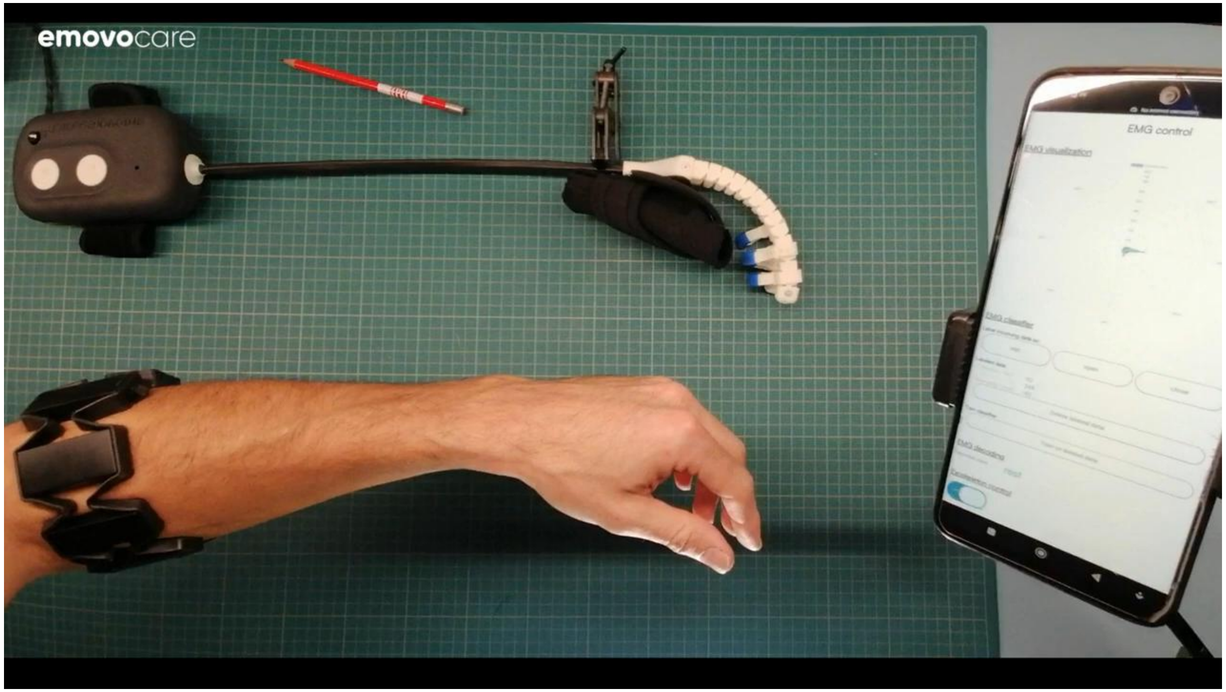
Advantages: **Simple, fast, no need for long training period**

Disadvantages: **Curse of dimensionality (*stores the entire dataset)***

→ ***Can be used for real-time learning of low-dimensional classification problems***



KNN application – Training EMG controller



EMG control App : provides visualization and a way to collect labelled data rapidly

Training phase (~30 sec) : User input from 8D EMG -> labelled into 3 classes (closed, open, rest) using the app

Control phase : Real-time KNN classification to control exoskeleton accordingly