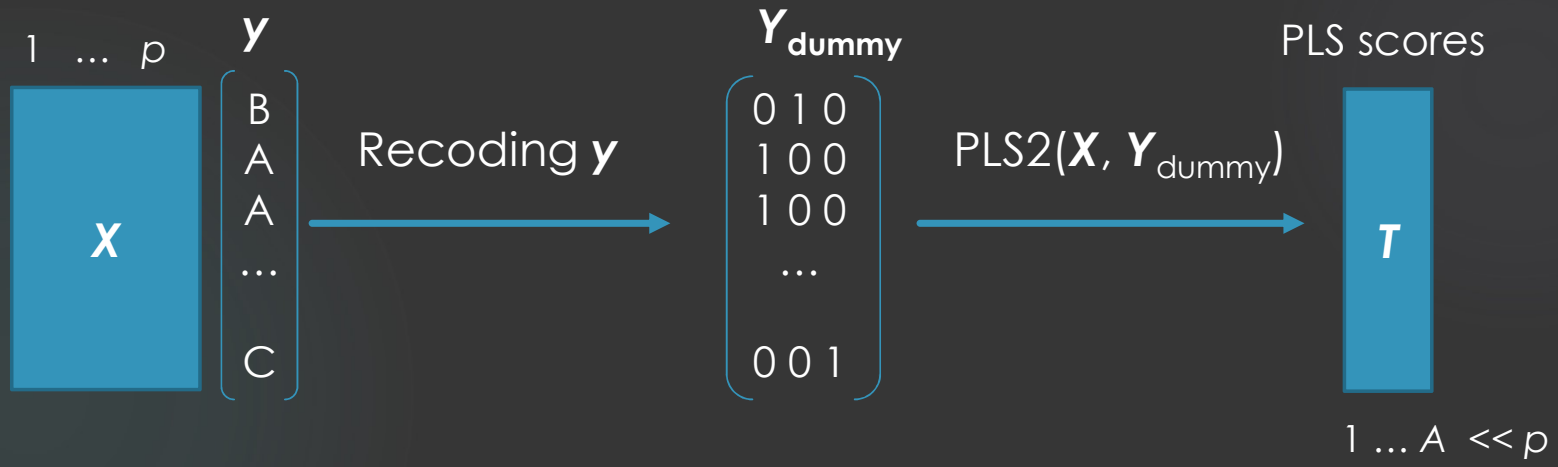
A non parametric PLSDA

matthieu.lesnoff@cirad.fr

Seminar ChemHouse Montpellier 19 June 2023

"PLSDA" PLS discriminant analysis

Step 1) Dimension reduction



Step 2) Several alternatives

A) Regression $\boldsymbol{Y}_{\mathbf{dummy}}$ on $\boldsymbol{T}$ $\Rightarrow$ PLSR-DA = usual "PLSDA"

$\widehat{Y}_{dummy}$

-.2	3.7	-1.5
.3	.4	-1.7
.9	-.1	.2
	…	
-.7	-.1	1.8

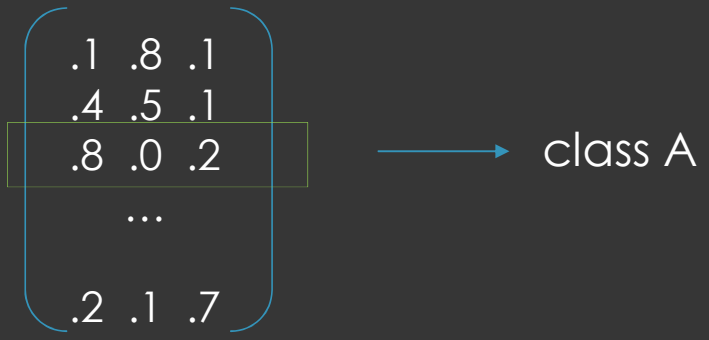
→ class A (highest prediction)

~ **unbounded** estimates of class-membership probabilities

(**Rk:** The same prediction table is usually returned by deep learning pipelines. In general, they transform the predictions by applying a final softmax function to come back to the scale of a probability [0, 1] but, in principle, it is not mandatory)

B) or Probabilistic DA on ***T***

Objective: $\hat{P}(\boldsymbol{y}_i = Class\ \ j \mid \boldsymbol{x}_i)$



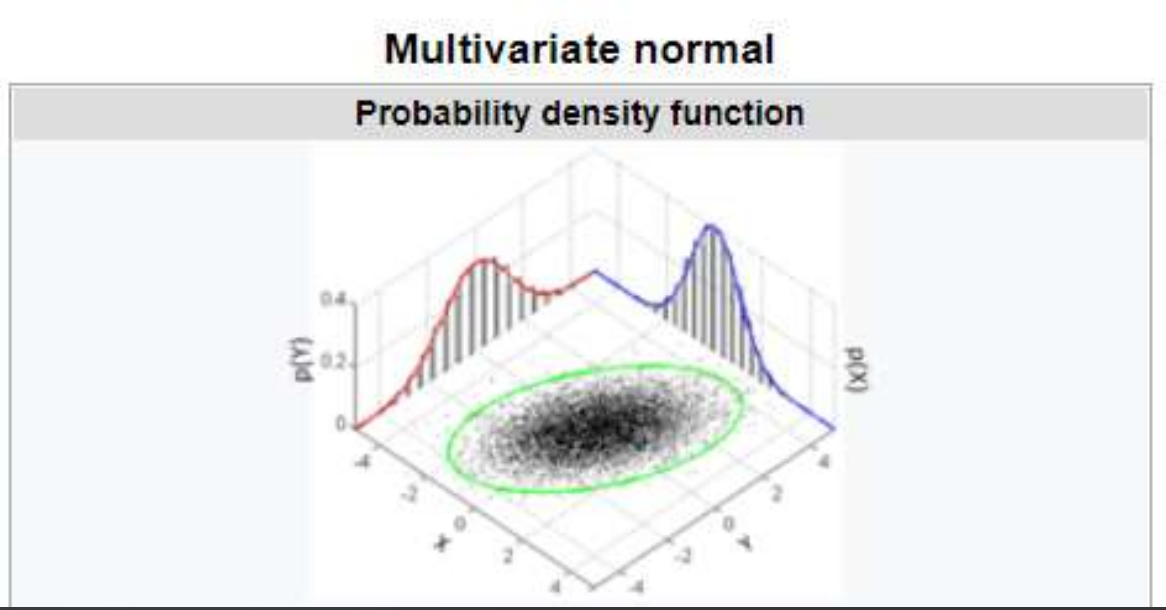
a) Parametric hypothesis on the distribution of ***T***
- e.g. **Gaussian** distribution(s)
  - LDA ⇒ PLS-LDA
  - QDA ⇒ PLS-QDA

b) Non parametric: No hypothesis on the distribution of ***T***
- e.g. **Kernel density estimation** (KDE) ⇒ PLS-KDE-DA

a) Parametric Gaussian probability density

PDF	$(2\pi)^{-k/2}\det(\boldsymbol{\Sigma})^{-1/2}\,\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right),$ exists only when $\boldsymbol{\Sigma}$ is positive-definite

Σ = ***X*** covariance matrix



- **LDA** Same Σ for all classes
- **QDA** One Σ per class

https://en.wikipedia.org/wiki/Multivariate_normal_distribution

b) Non parametric

Univariate KDE

$$\hat{f}_K(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) = \frac{1}{n}\sum_{i-1}^{n} K_h(x - x_i)\,,$$

Density estimate at obs. **x**

Bandwidth (= smoothing) Parameter to tune

Kernel function (integral sums to 1) e.g. Gaussian pdf

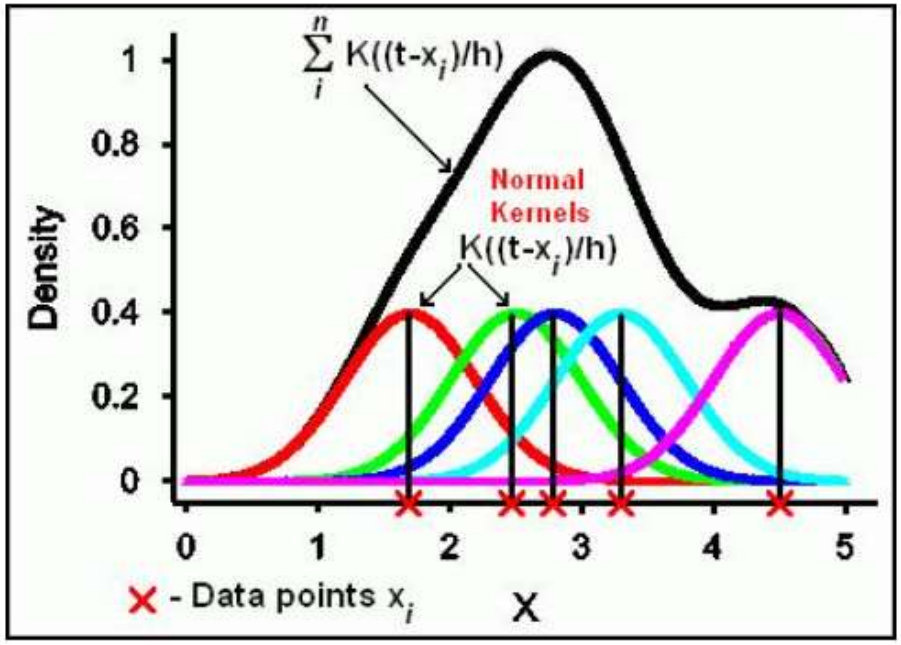
~ similarity index to **x**

Scott DW, Sain SR. - Multidimensional Density Estimation. In: Rao CR, Wegman EJ, Solka JL, eds. Handbook of Statistics. Vol 24. Data Mining and Data Visualization. Elsevier; 2005:229-261. doi:10.1016/S0169-7161(04)24009-3.

$$fh(t) = \text{the sum of } (K(t-x_i)/h)/(nh) \text{ from } i = 1...n$$

where n denotes the sample size. The choice of kernel function K is not very critical for the resulting estimate fh(t) and so a Gaussian kernel is used.

The following graph showing the sum of the normal kernels at 5 data points illustrates the ideas behind the kernel density estimation.



Univariate KDE

Source: https://genstat.kb.vsni.co.uk/knowledge-base/kernel-density-estimation

Gaussian univariate KDE

$$\hat{f}(x) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{h\sqrt{2\pi}} exp\left\{-\frac{1}{2}\left(\frac{x - x_i}{h}\right)^2\right\}$$

$$= \frac{1}{nh}\sum_{i=1}^{n} K(u_i)$$

with $u_i = \frac{x - x_i}{h}$

and $K(u_i) = \frac{1}{\sqrt{2\pi}} exp\left\{-\frac{1}{2}\, u_i^2\right\}$ **pdf Normal(0, 1)**

Gaussian multivariate KDE

The extension of the kernel estimator to vector-valued data, $\mathbf{x} \in \Re^d$, is straightforward for a normal kernel, $K \sim N(0, \Sigma)$:

$$\hat{f}(\mathbf{x}) = \frac{1}{n(2\pi)^{d/2}|\Sigma|^{1/2}} \sum_{i=1}^{n} \exp[-\frac{1}{2}(\mathbf{x} - \mathbf{x}_i)'\Sigma^{-1}(\mathbf{x} - \mathbf{x}_i)] \,. \qquad (16)$$

Bandwidth matrix

$p \times p$ Positive-definite, symmetric

To tune

Scott & Sain 2005

$$\hat{f}(\mathbf{x}) = \frac{1}{n(2\pi)^{d/2}|\Sigma|^{1/2}} \sum_{i=1}^{n} \exp[-\frac{1}{2}(\mathbf{x} - \mathbf{x}_i)'\Sigma^{-1}(\mathbf{x} - \mathbf{x}_i)] \,. \qquad (16)$$

Assuming Σ to be diagonal in (16)
simplifies a lot the computations and tuning

$\Rightarrow$ Multiplicative Gaussian KDE

= product of univariate KDEs

Illustration on iris data

X

Row	sepal_length Float64	sepal_width Float64	petal_length Float64	petal_width Float64
1	5.1	3.5	1.4	0.2
2	4.9	3.0	1.4	0.2
3	4.7	3.2	1.3	0.2

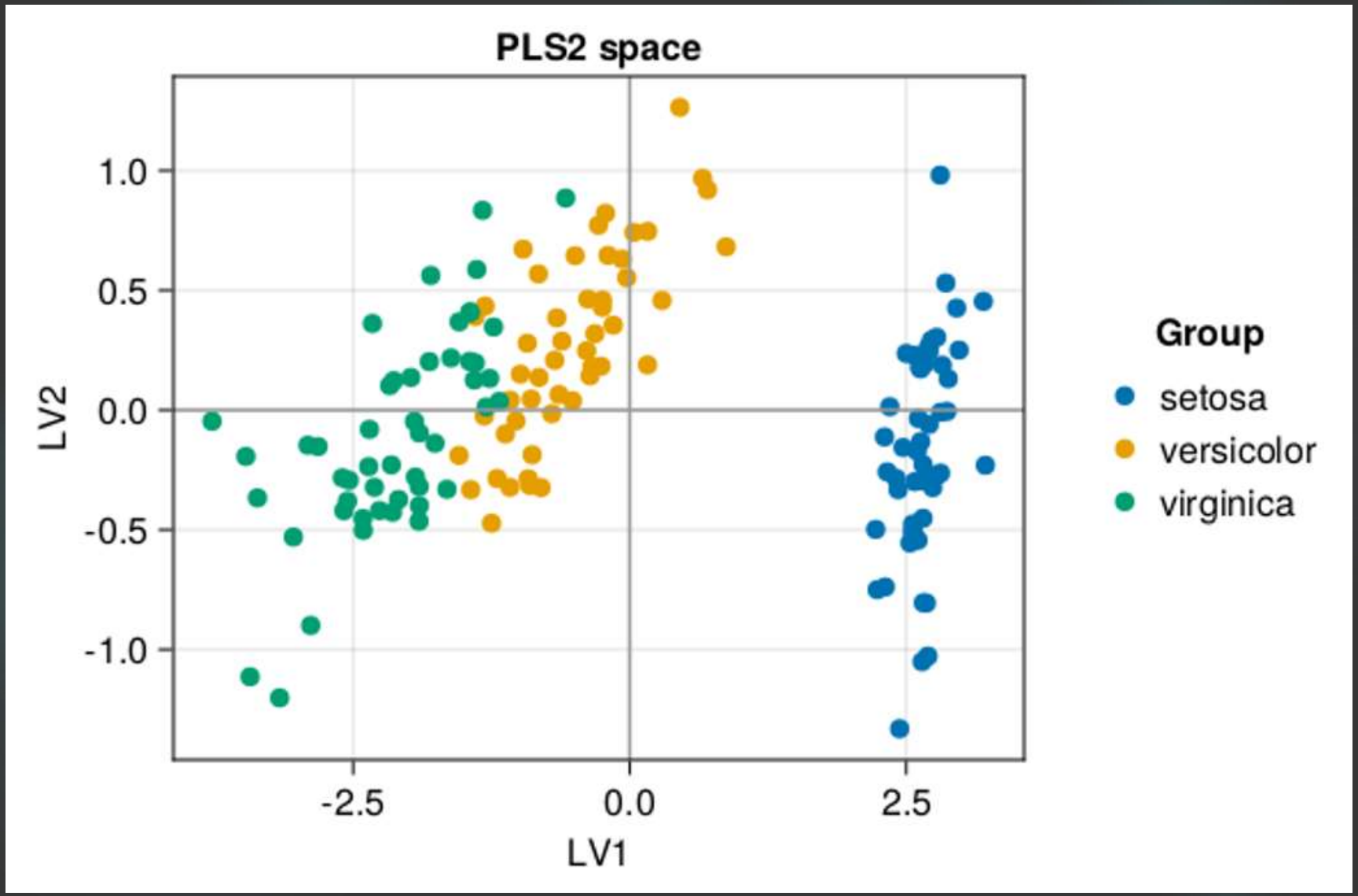
... (150, 4)

y: 3 classes

```
"setosa"     => 50
"versicolor" => 50
"virginica"  => 50
```

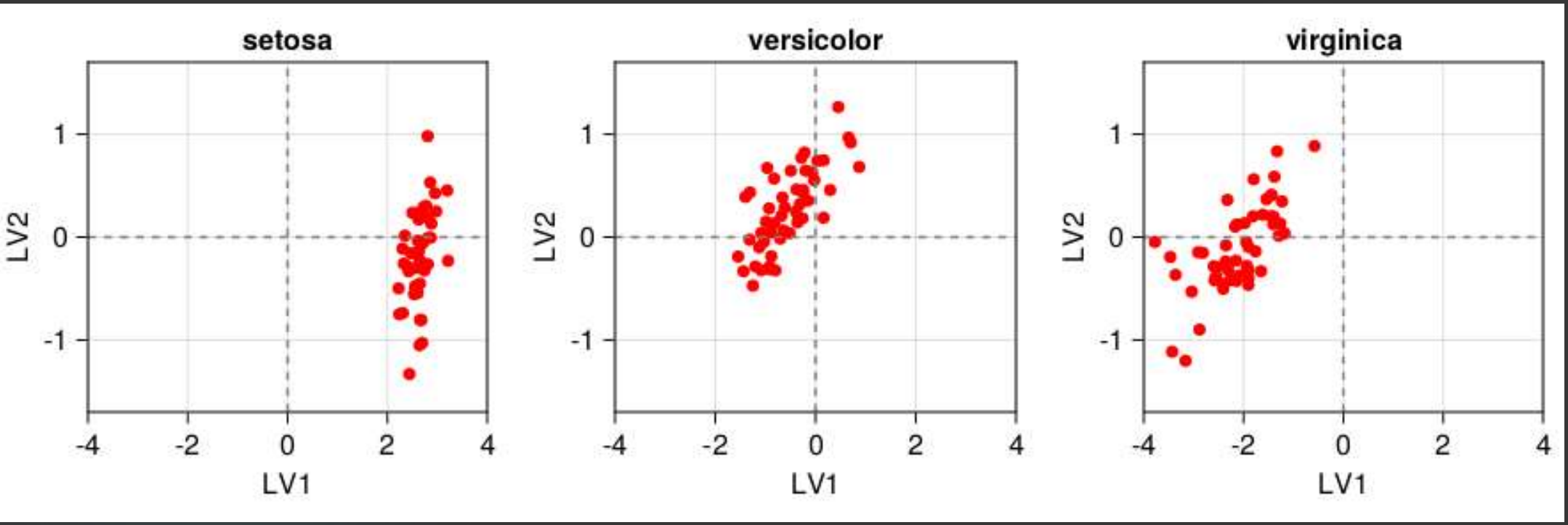
PLS2($\boldsymbol{X}$, $\boldsymbol{Y}_{\text{dummy}}$) nb. LVs = 2

$\Rightarrow$ $\mathbf{T}$ ($n \times 2$)

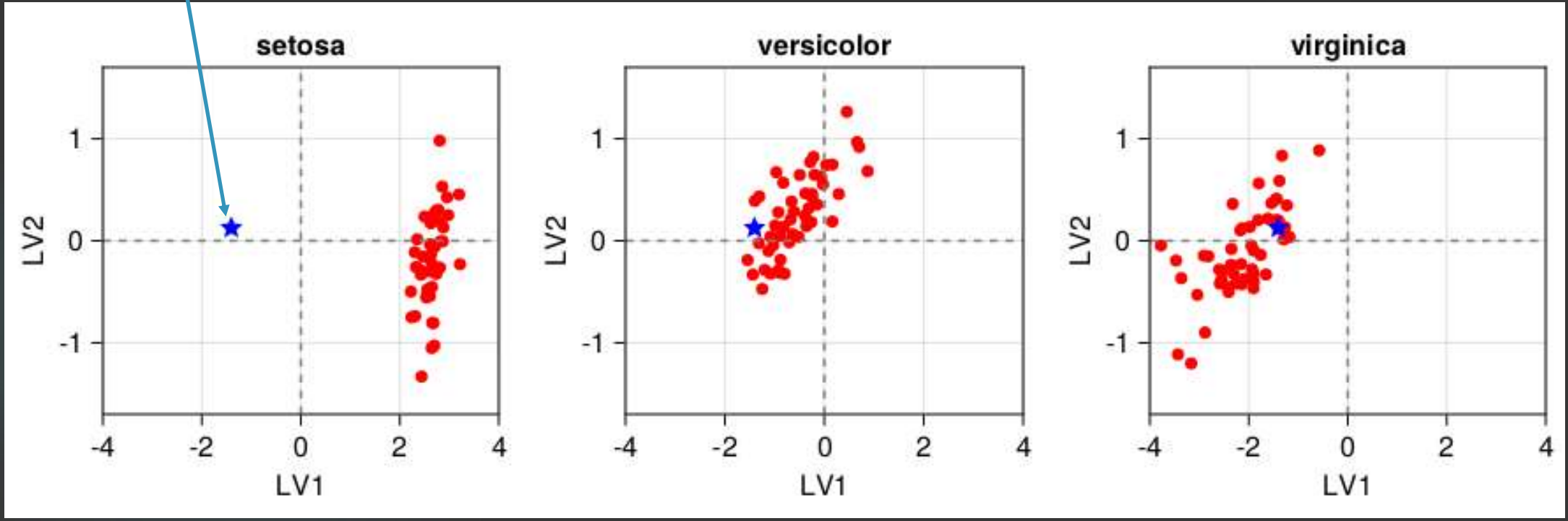

PLS2 space
1.0
0.5
0.0
-0.5
-1.0
LV2
-2.5
0.0
2.5
LV1
Group
setosa
versicolor
virginica

Probabilistic DA

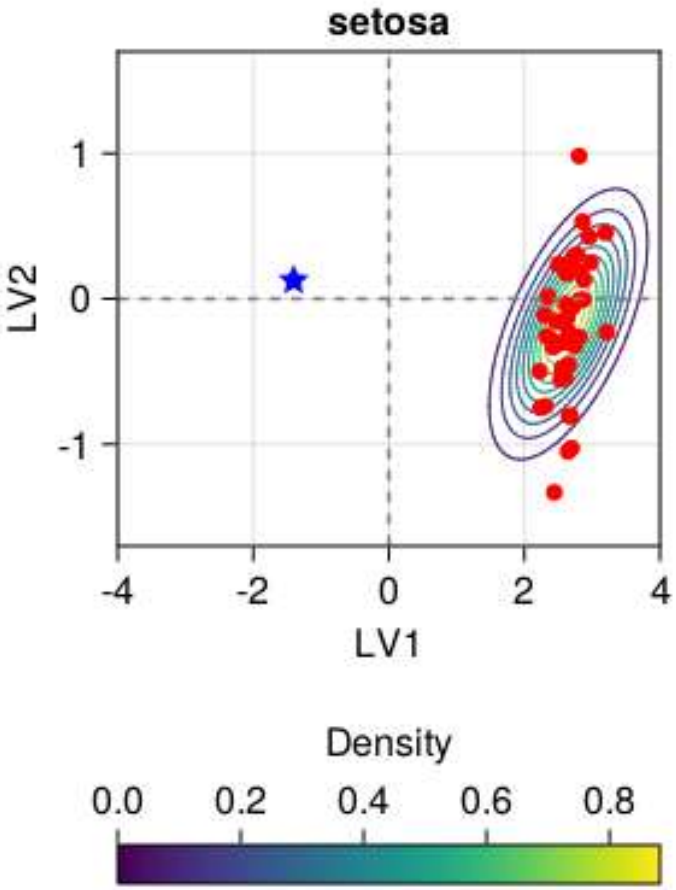
⇒ Estimate the multivariate **probability density function (pdf)** of ***T*** in each class

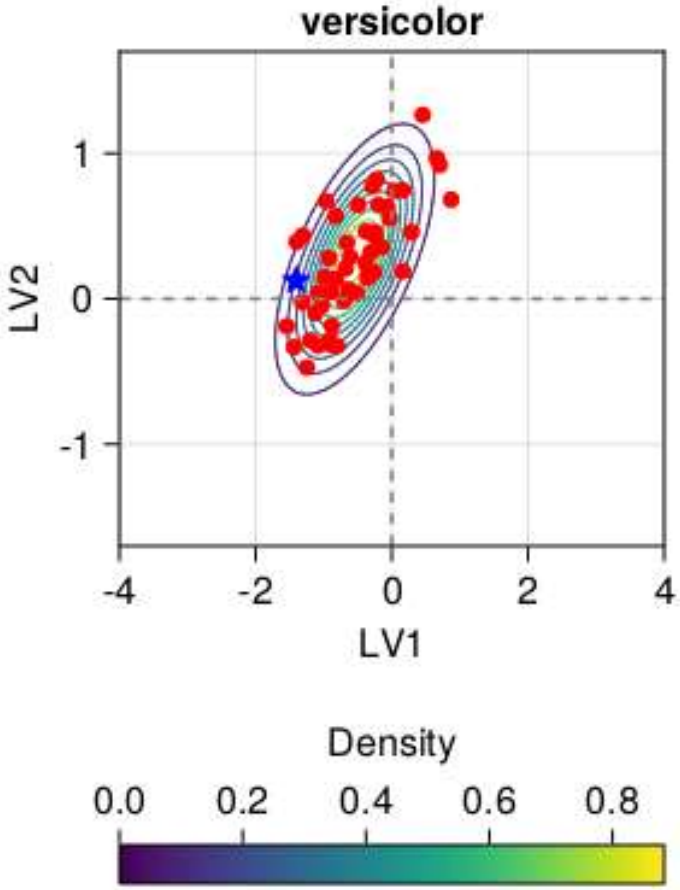


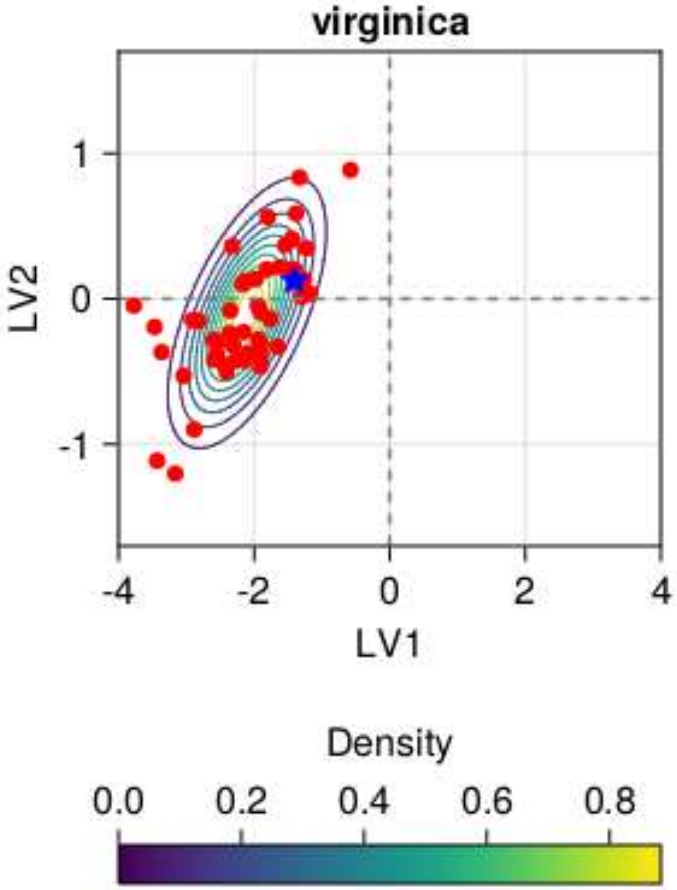
New observation to predict
= which class? ⇔ where is located the new obs. compared to the class pdf



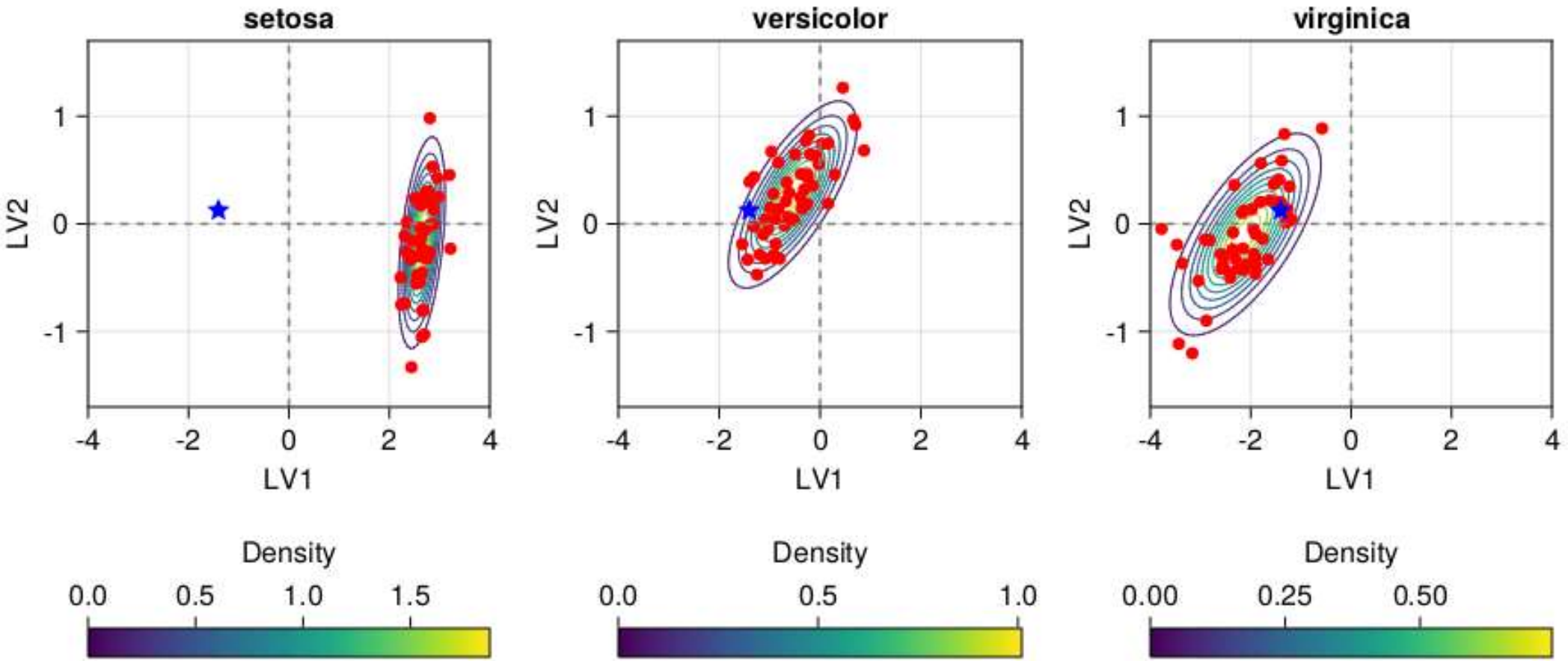
LDA **Same covariance matrix Σ** estimated for all classes





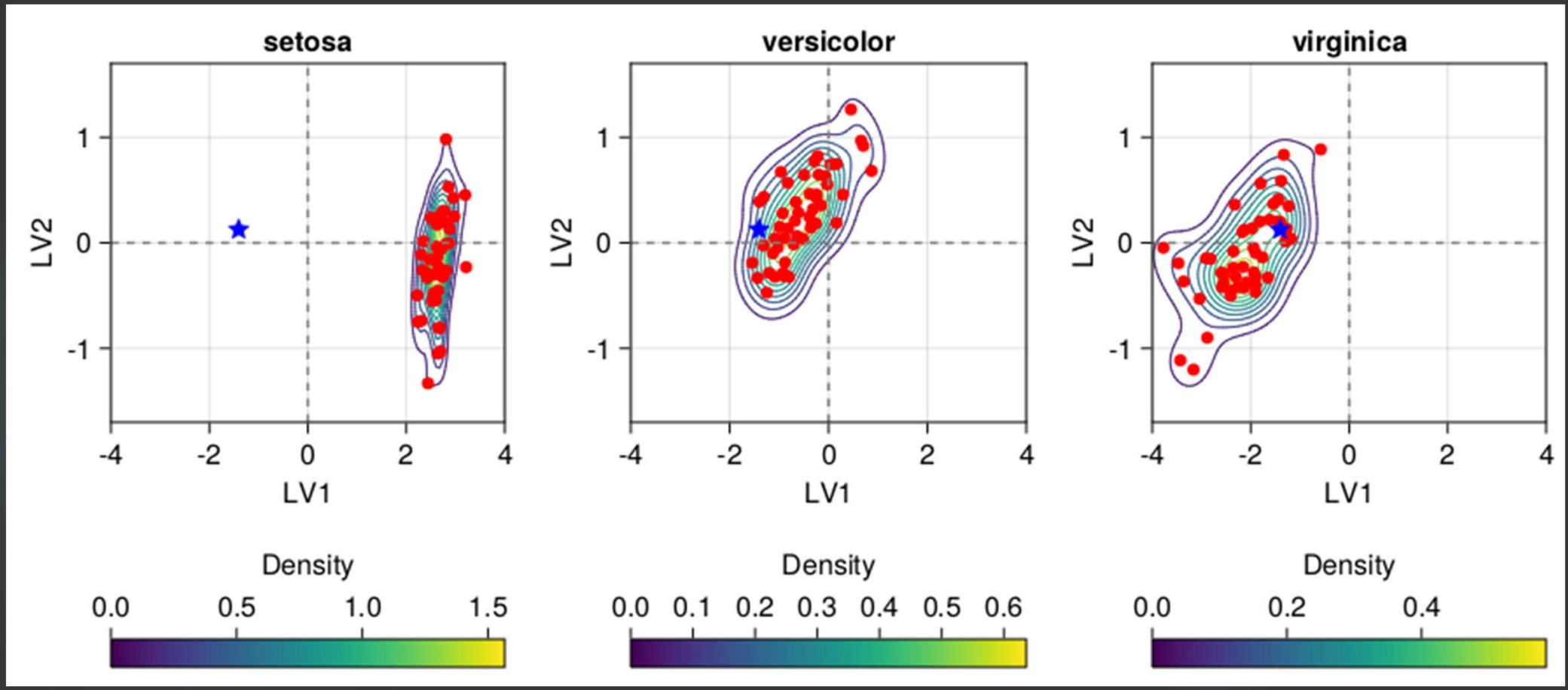


QDA **One covariance matrix** Σ_j estimated per class



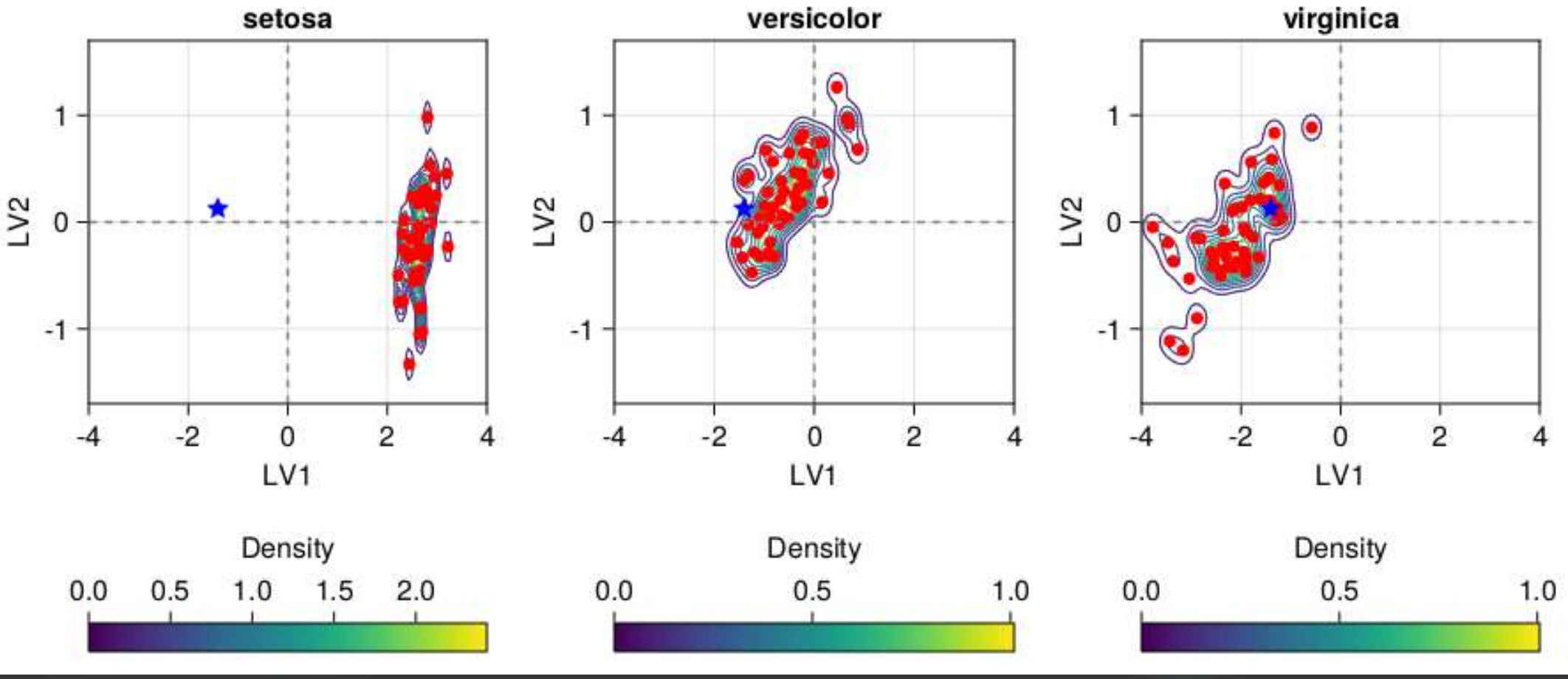
Multiplicative Gaussian KDE

Smoothing level $a = 1$

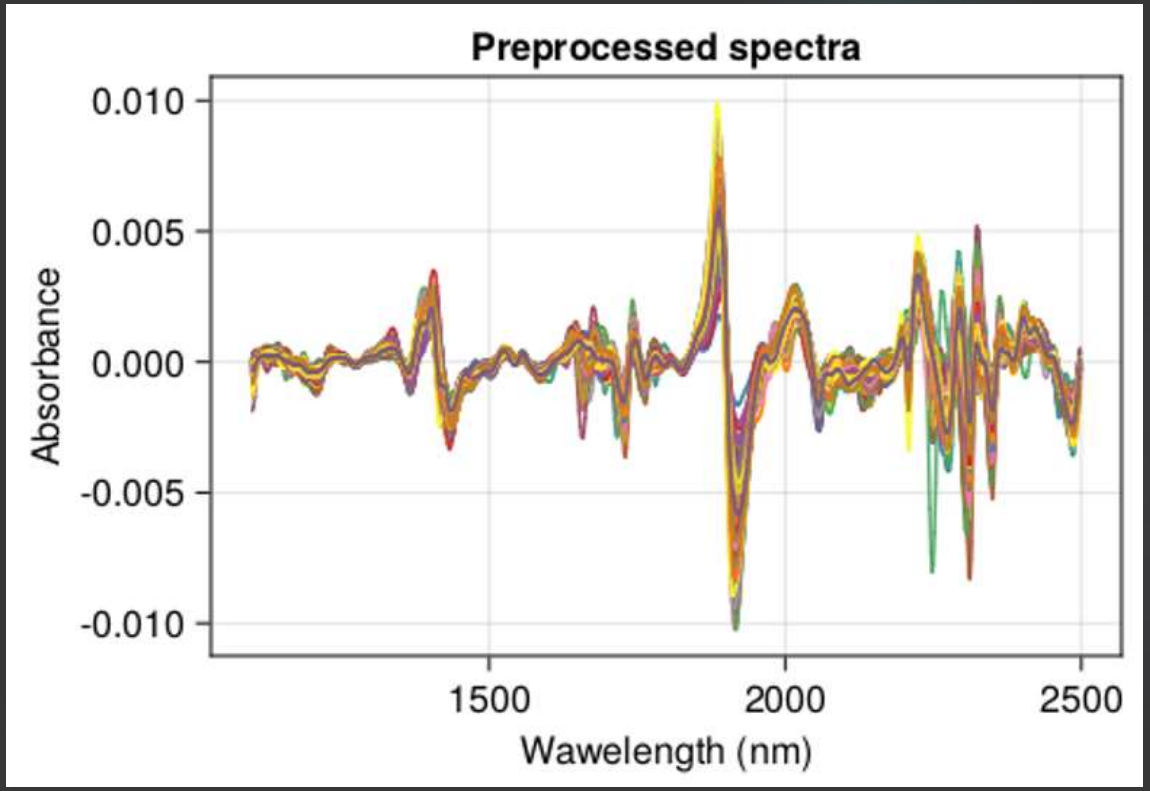


Multiplicative Gaussian KDE

Smoothing level $\alpha = .5$



Another illustration
NIR Forages

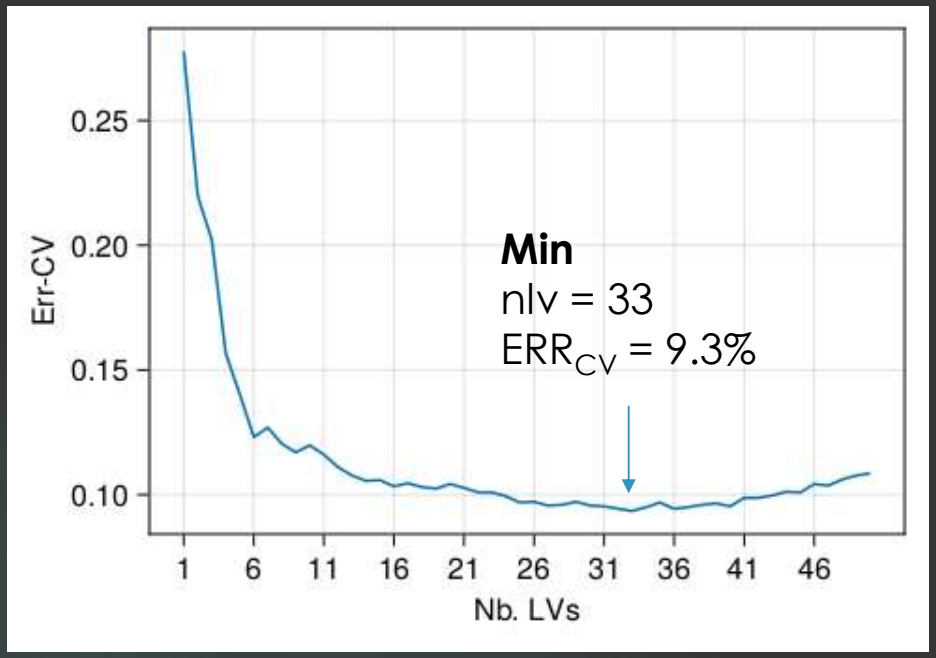


ntot = 485
ntrain = 323 (CV)
ntest = 162

y: 3 classes

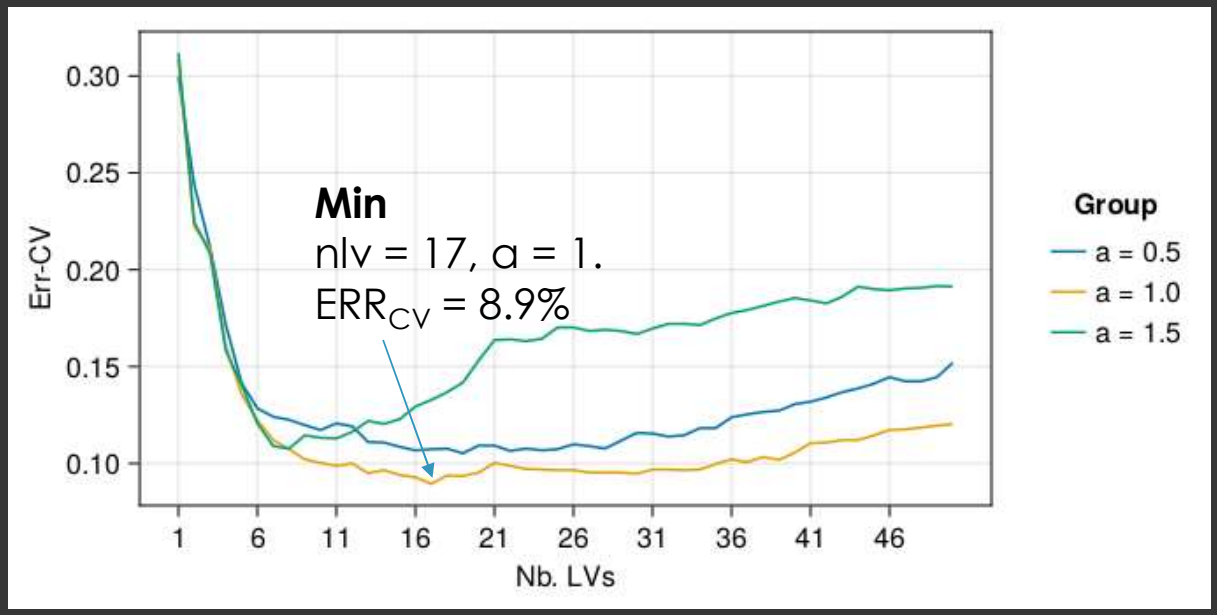
```
"Legume forages"
"Forage trees"
"Cereal and grass forages"
```

PLS-LDA



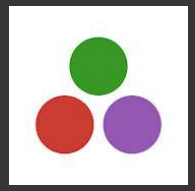
ERR_{Test} = 8%

PLS-KDE-DA



ERR_{Test} = 7%

Often, PLS-KDE-DA has a performance close to those of parametric methods, but it can overperform if some classes have clear internal multimodal distributions (eg. when a class contains several sub-groups that do not span the same sub-spaces of the PLS space)



julia

Function **plskdeda** available in package Jchemo
https://github.com/mlesnoff/Jchemo.jl
https://github.com/mlesnoff/JchemoDemo

```
model = plskdeda(nlv = 15, a = .5)
fit!(model, Xtrain, ytrain)
pred = predict(model, Xtest).test
```