CLUSTERING AND CLASSIFICATION

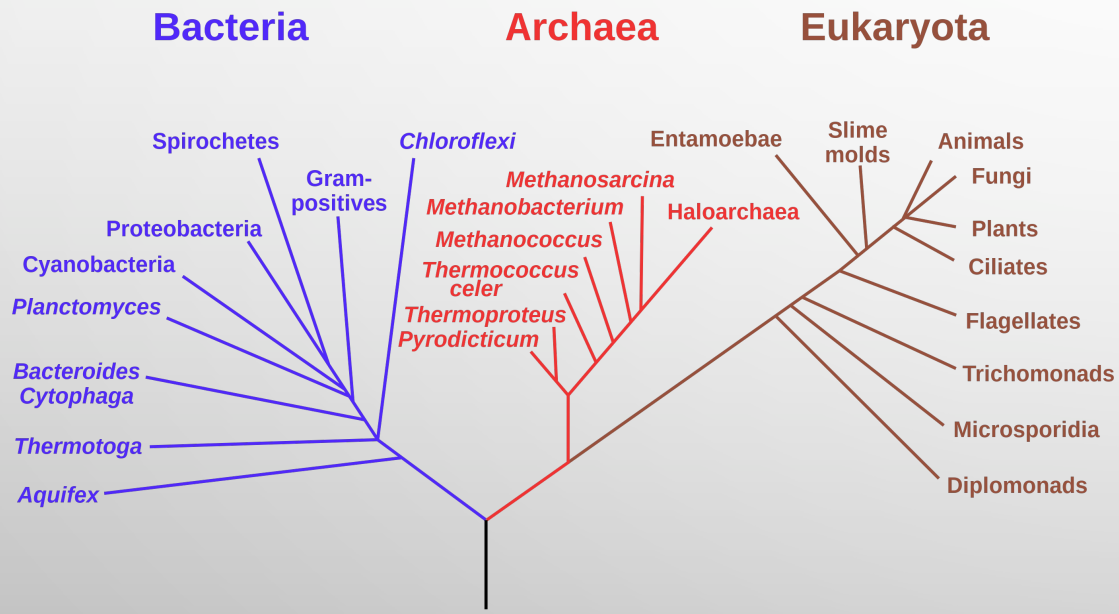
INTRODUCTION TO MACHINE LEARNING

2023

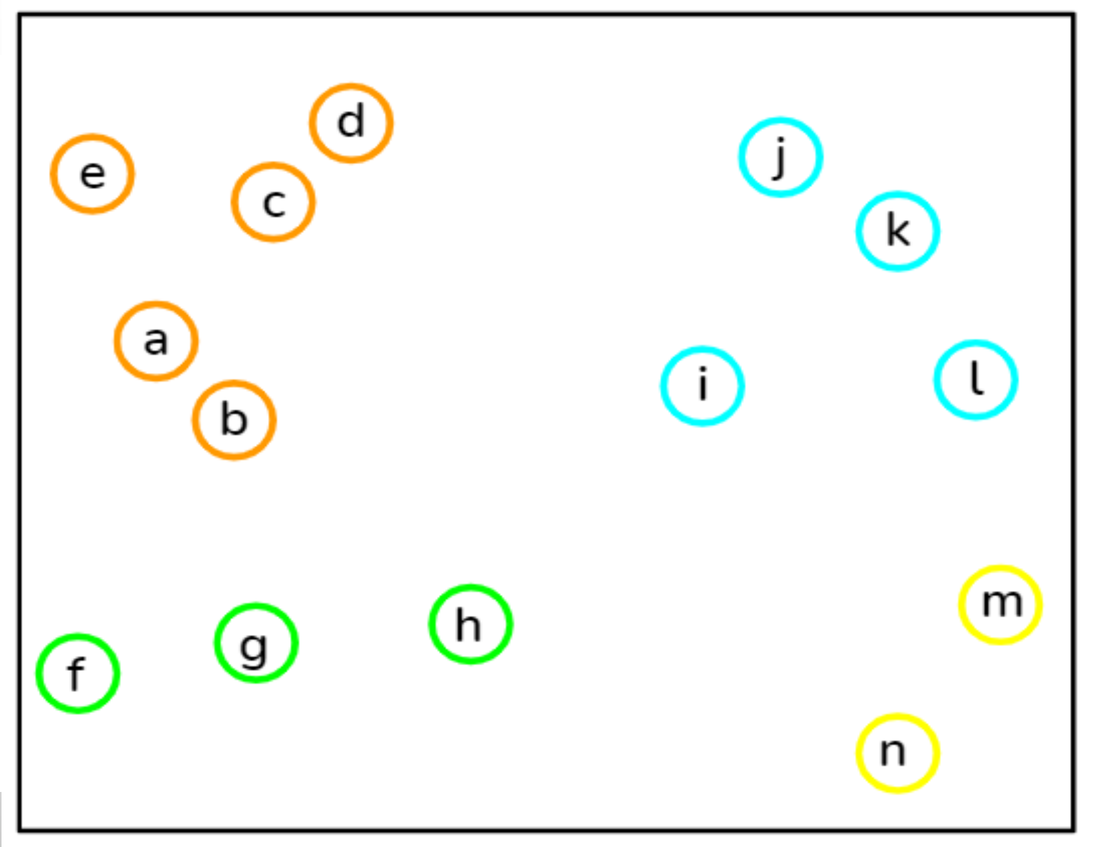
UNSUPERVISED HIERARCHICAL CLUSTERING: DENDROGRAM

UNSUPERVISED HIERARCHICAL CLUSTERING: DENDROGRAM

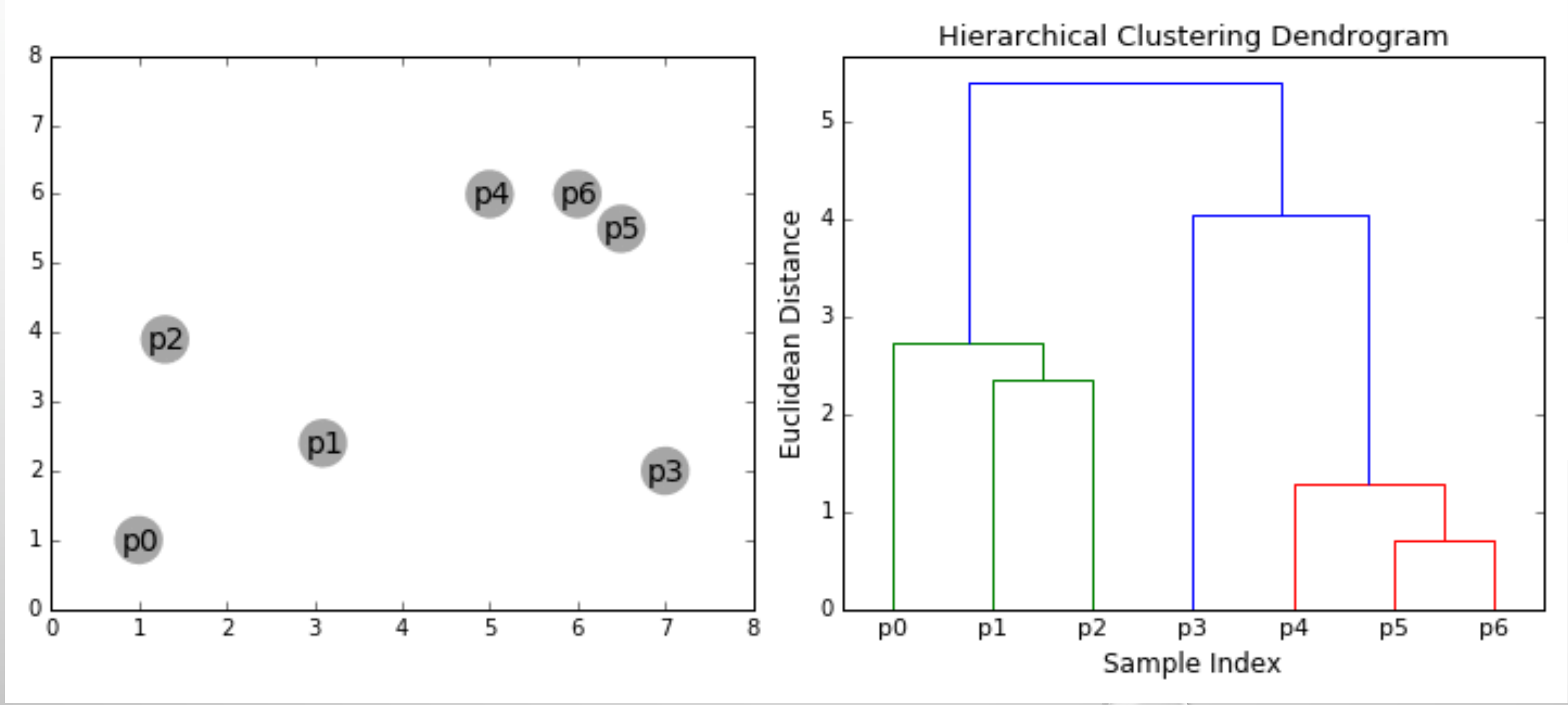
- AGLOMERATIVE



- DIVISIVE

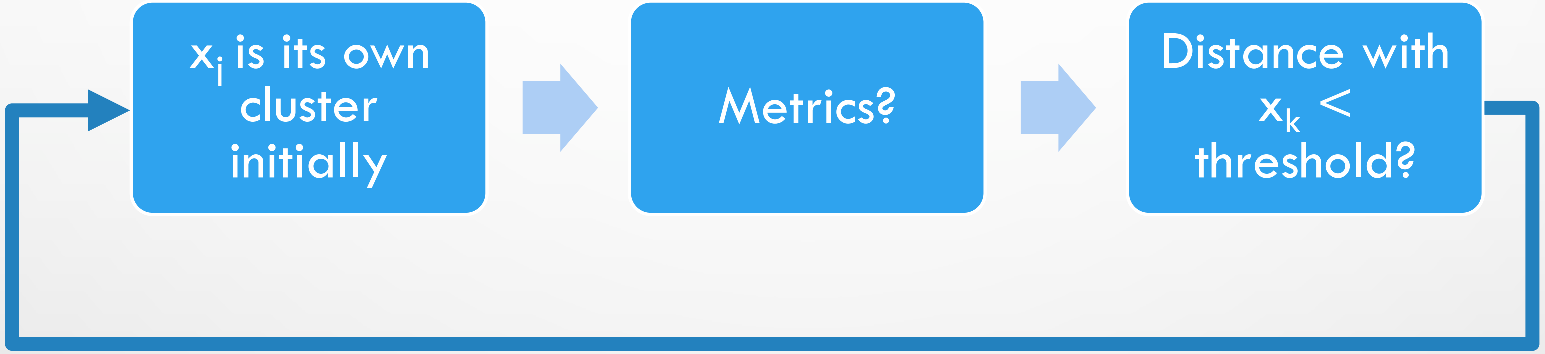


AGGLOMERATIVE HIERARCHICAL CLUSTERING AND THE DENDROGRAM



$\|\mathbf{x}_j - \mathbf{x}_k\|_2$

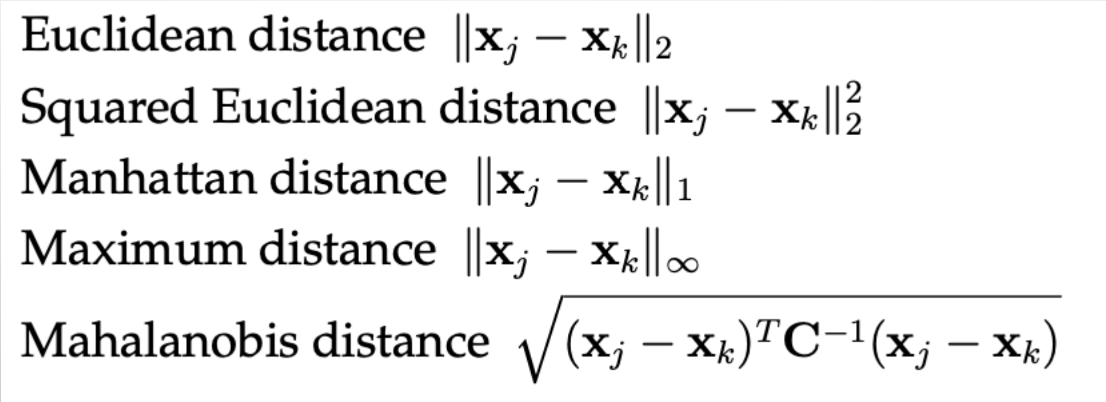
AGGLOMERATIVE HIERARCHICAL CLUSTERING AND THE DENDROGRAM



x_i is its own cluster initially

Metrics?

Distance with x_k < threshold?



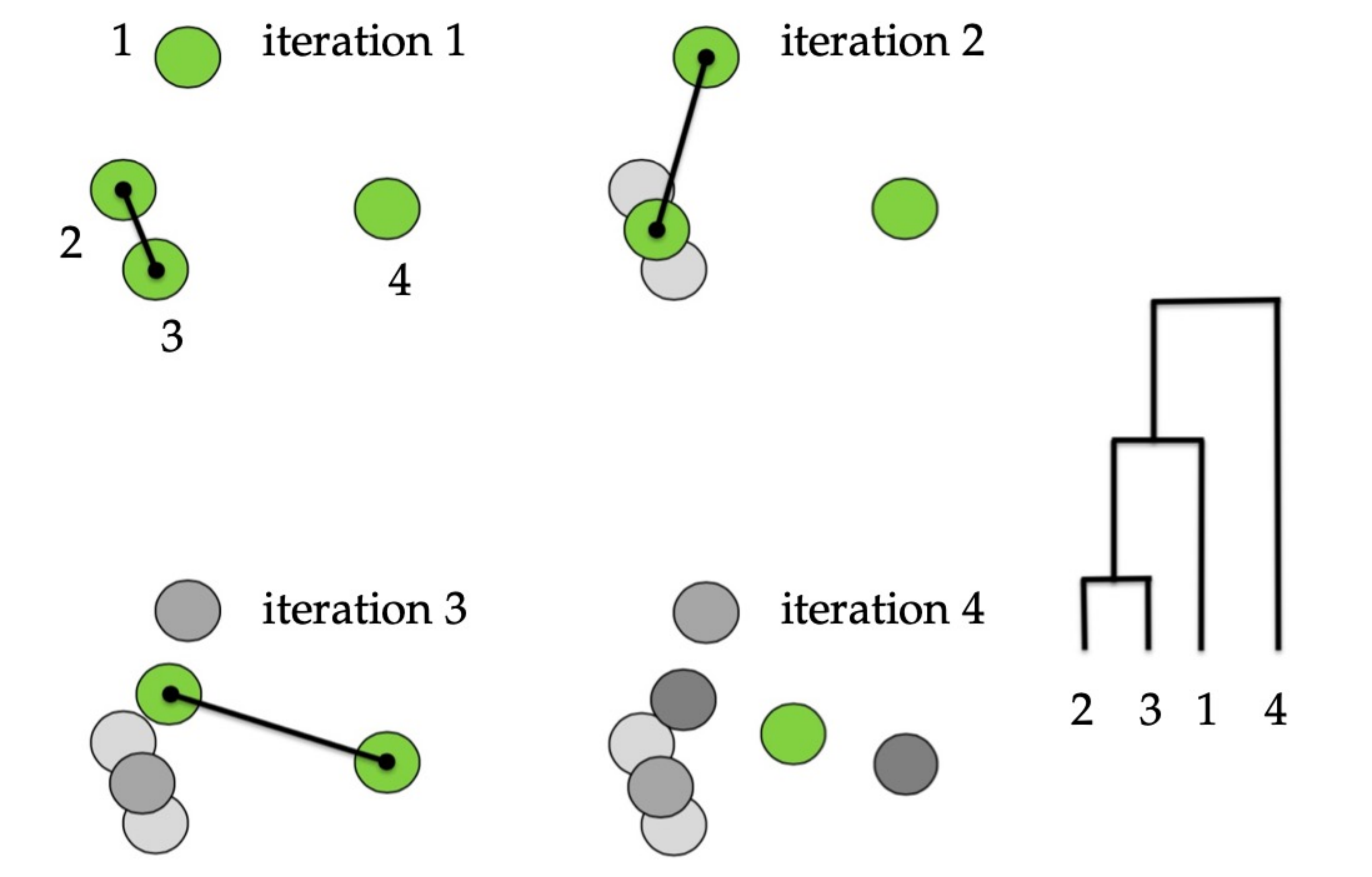
Euclidean distance $\|\mathbf{x}_j - \mathbf{x}_k\|_2$

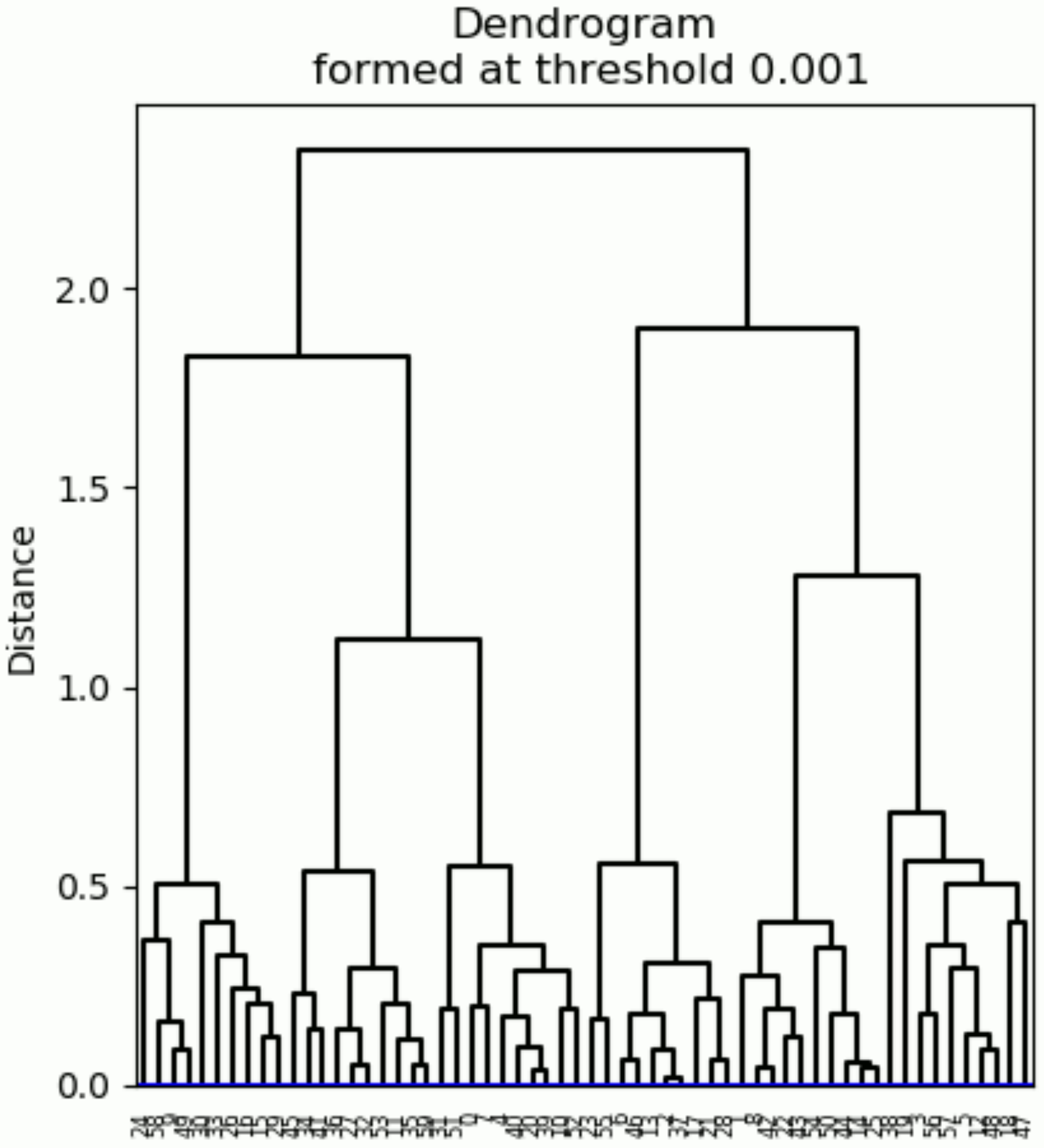
Squared Euclidean distance $\|\mathbf{x}_j - \mathbf{x}_k\|_2^2$

Manhattan distance $\|\mathbf{x}_j - \mathbf{x}_k\|_1$

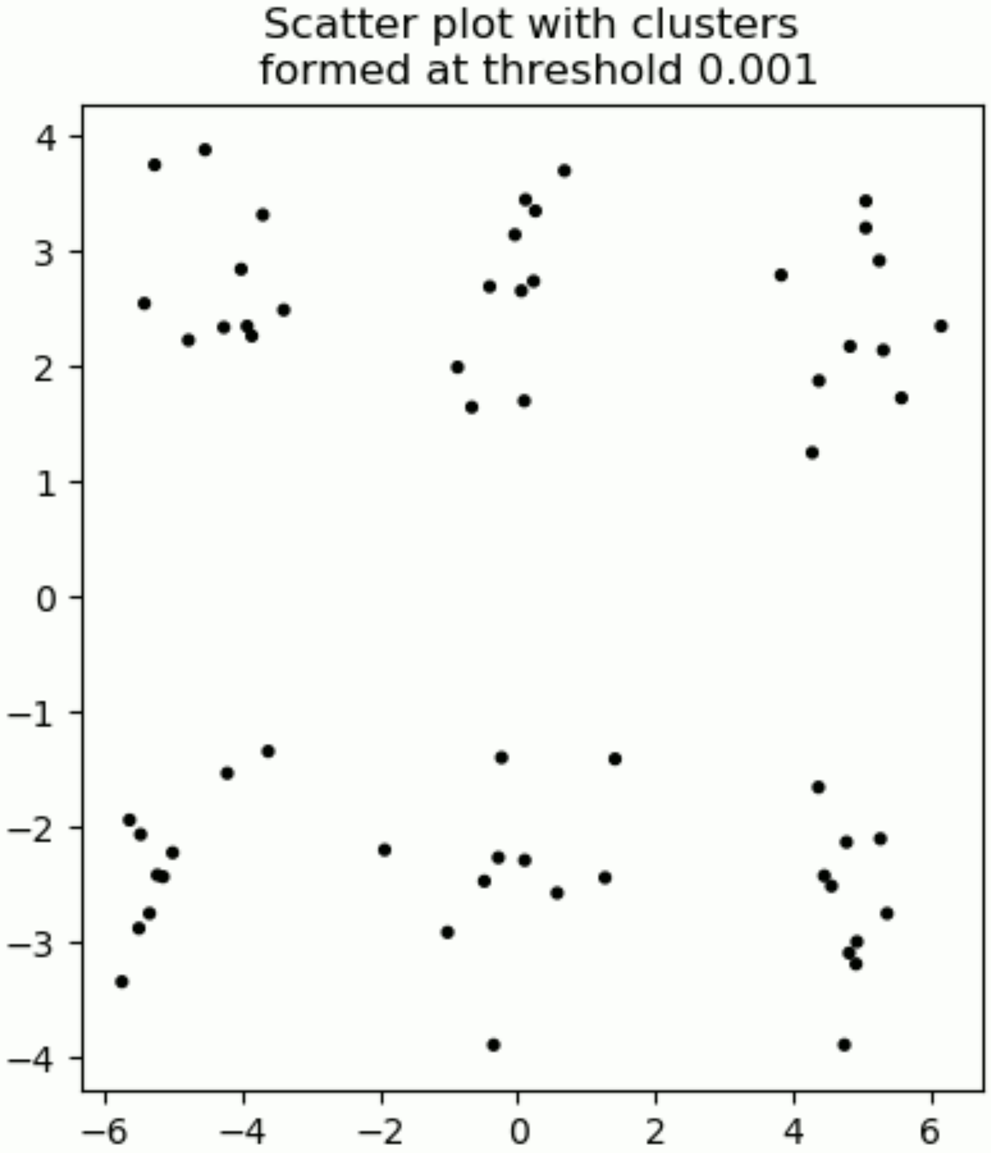
Maximum distance $\|\mathbf{x}_j - \mathbf{x}_k\|_\infty$

Mahalanobis distance $\sqrt{(\mathbf{x}_j - \mathbf{x}_k)^T \mathbf{C}^{-1} (\mathbf{x}_j - \mathbf{x}_k)}$

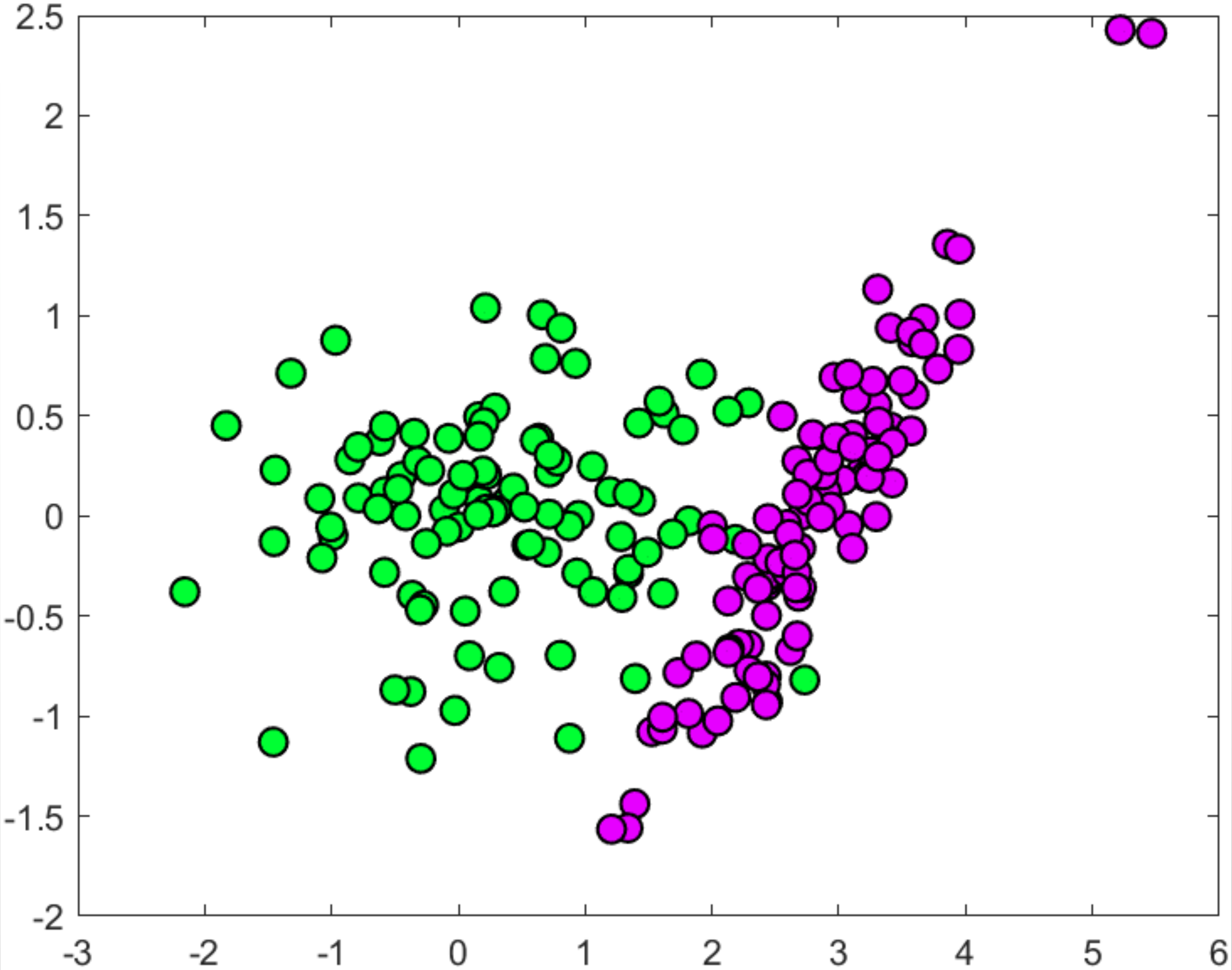

1
iteration 1
2
3
4
iteration 2
iteration 3
iteration 4
2
3
1
4

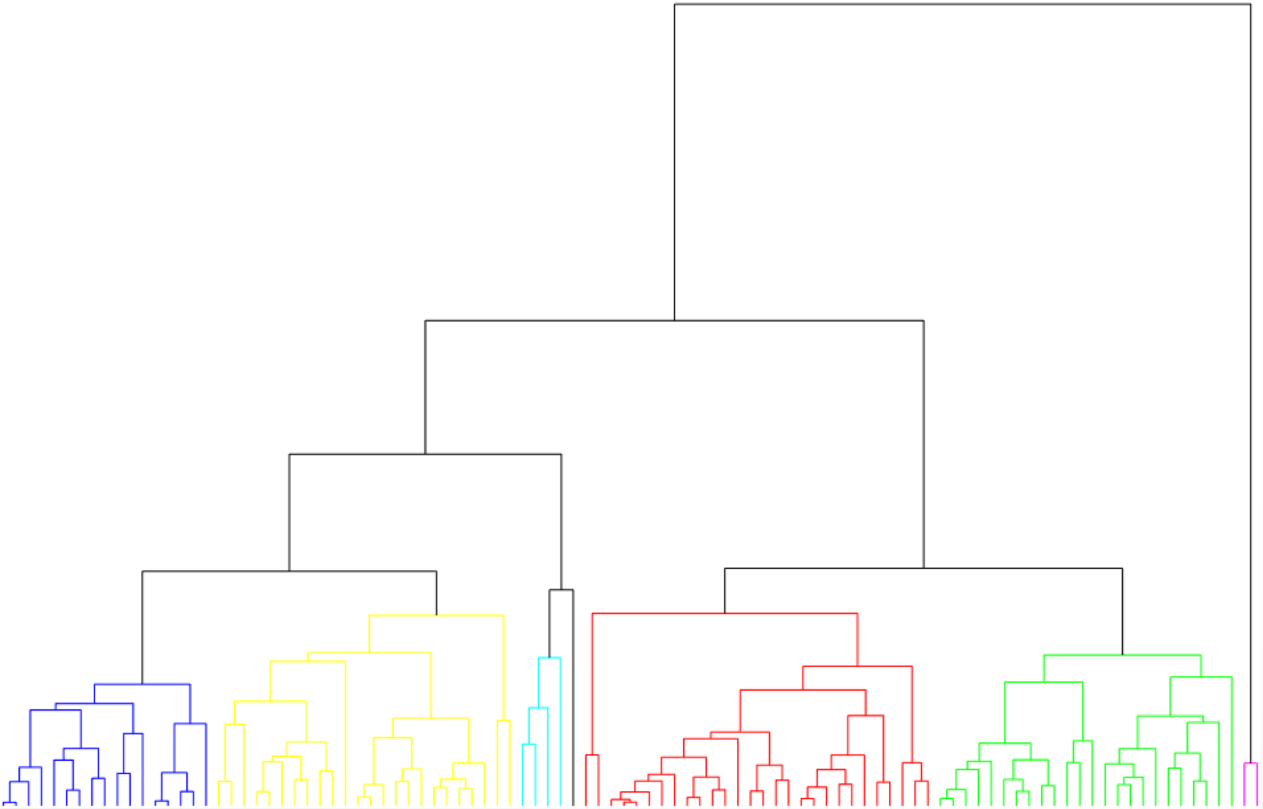


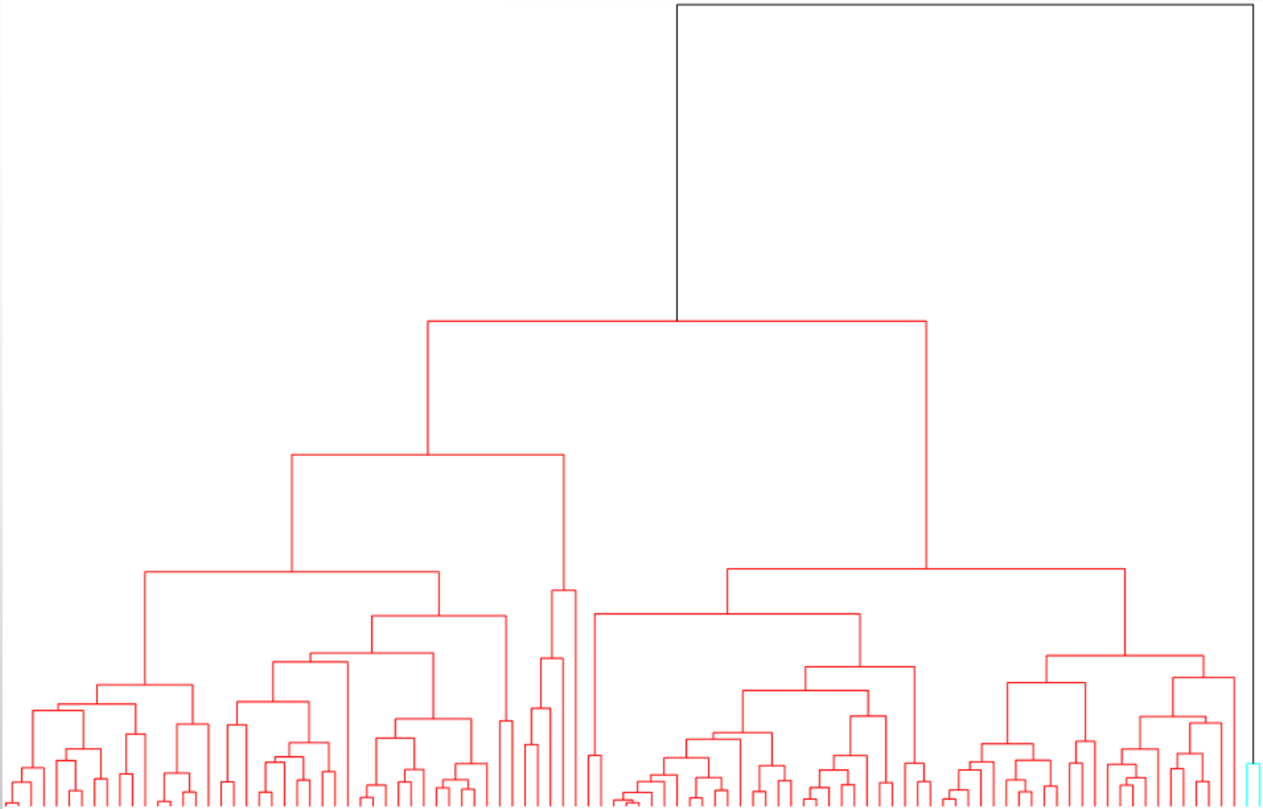
Dendrogram
formed at threshold 0.001

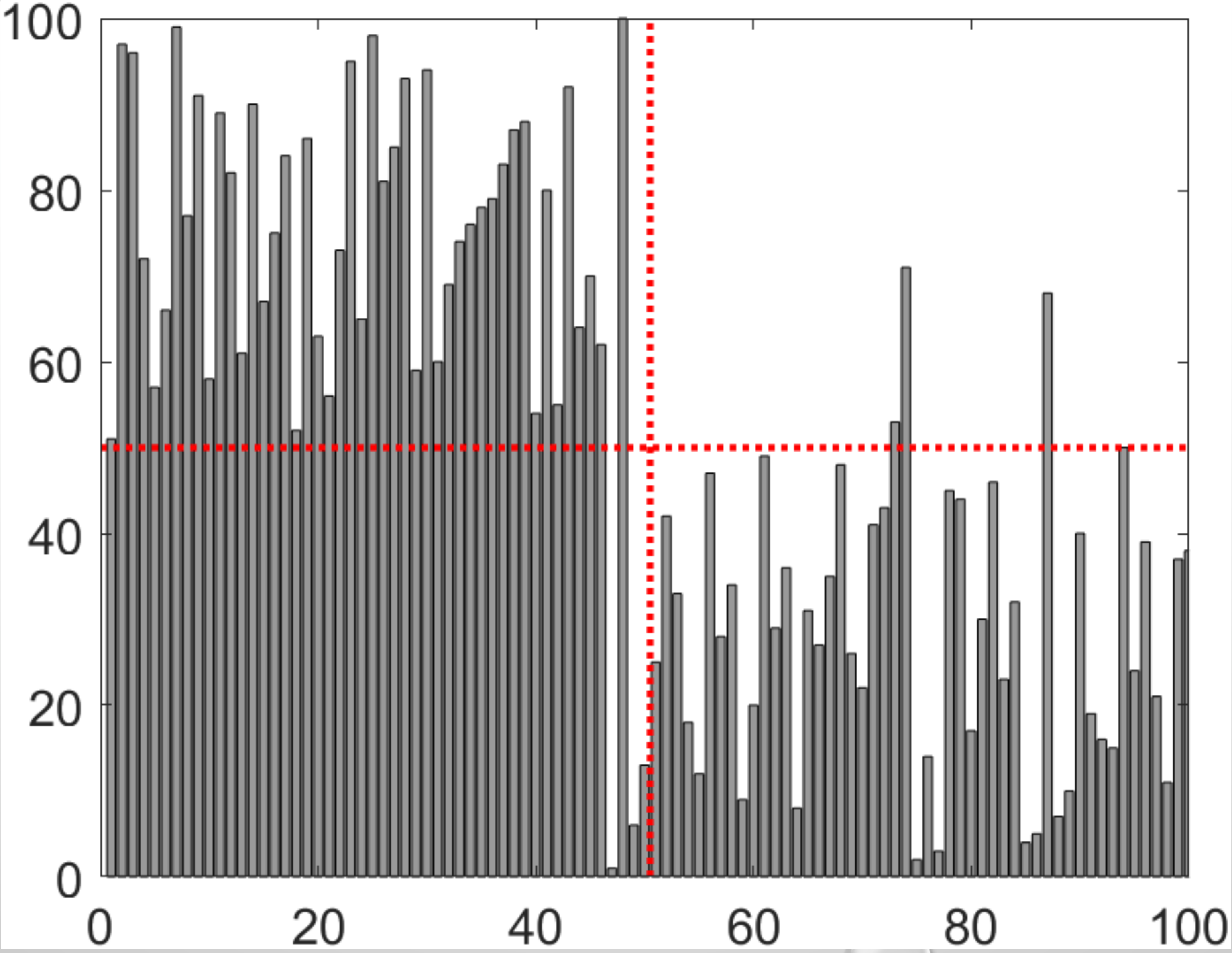


Scatter plot with clusters
formed at threshold 0.001


2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-3
-2
-1
0
1
2
3
4
5
6



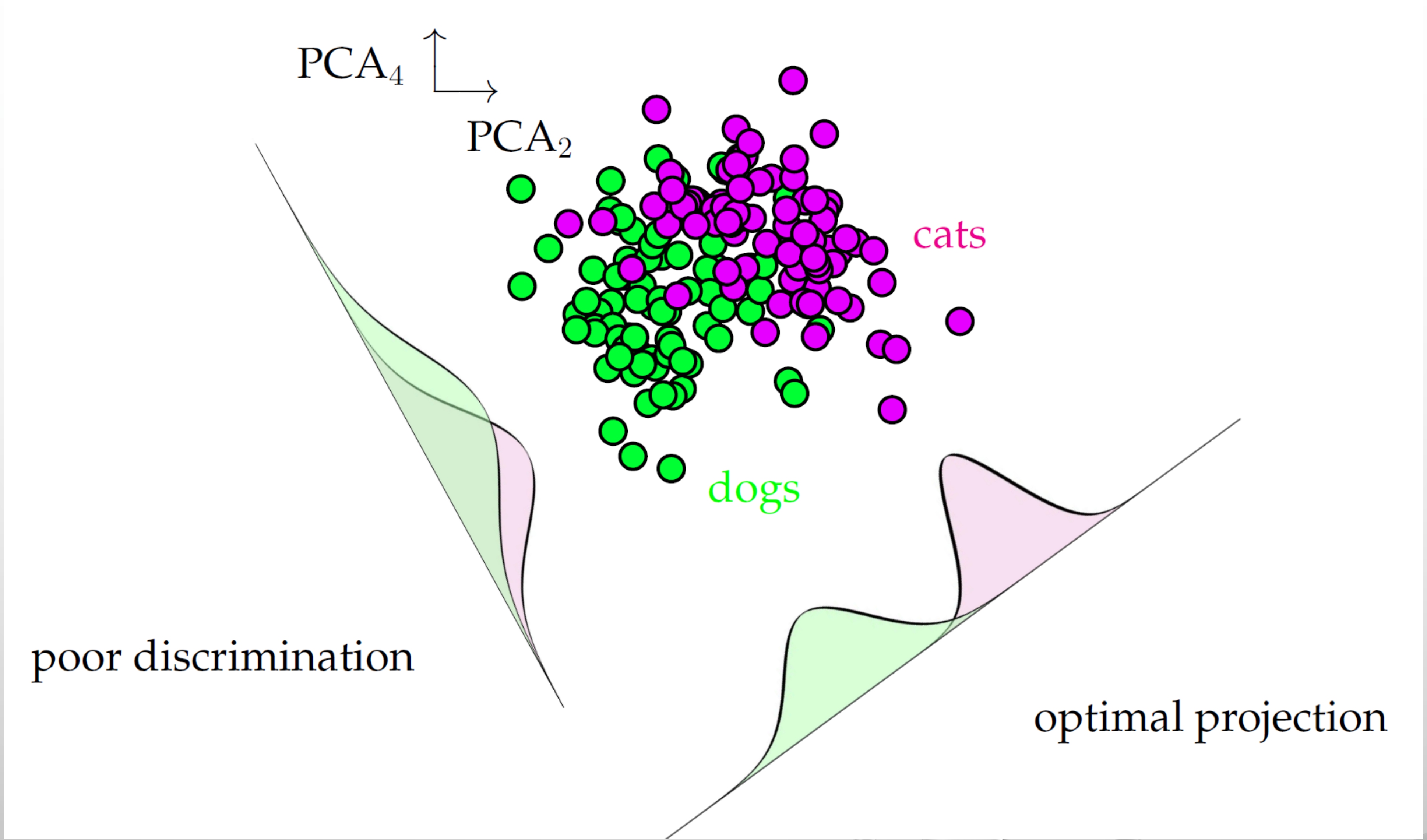



100
80
60
40
20
0
0
20
40
60
80
100

SUPERVISED LEARNING AND LINEAR DISCRIMINANTS

LINEAR DISCRIMINANT ANALYSIS (LDA)

THE GOAL OF THESE ALGORITHMS IS TO FIND A LINEAR COMBINATION OF FEATURES THAT CHARACTERIZES OR SEPARATES TWO OR MORE CLASSES OF OBJECTS OR EVENTS IN THE DATA


PCA_4
PCA_2
cats
dogs
poor discrimination
optimal projection

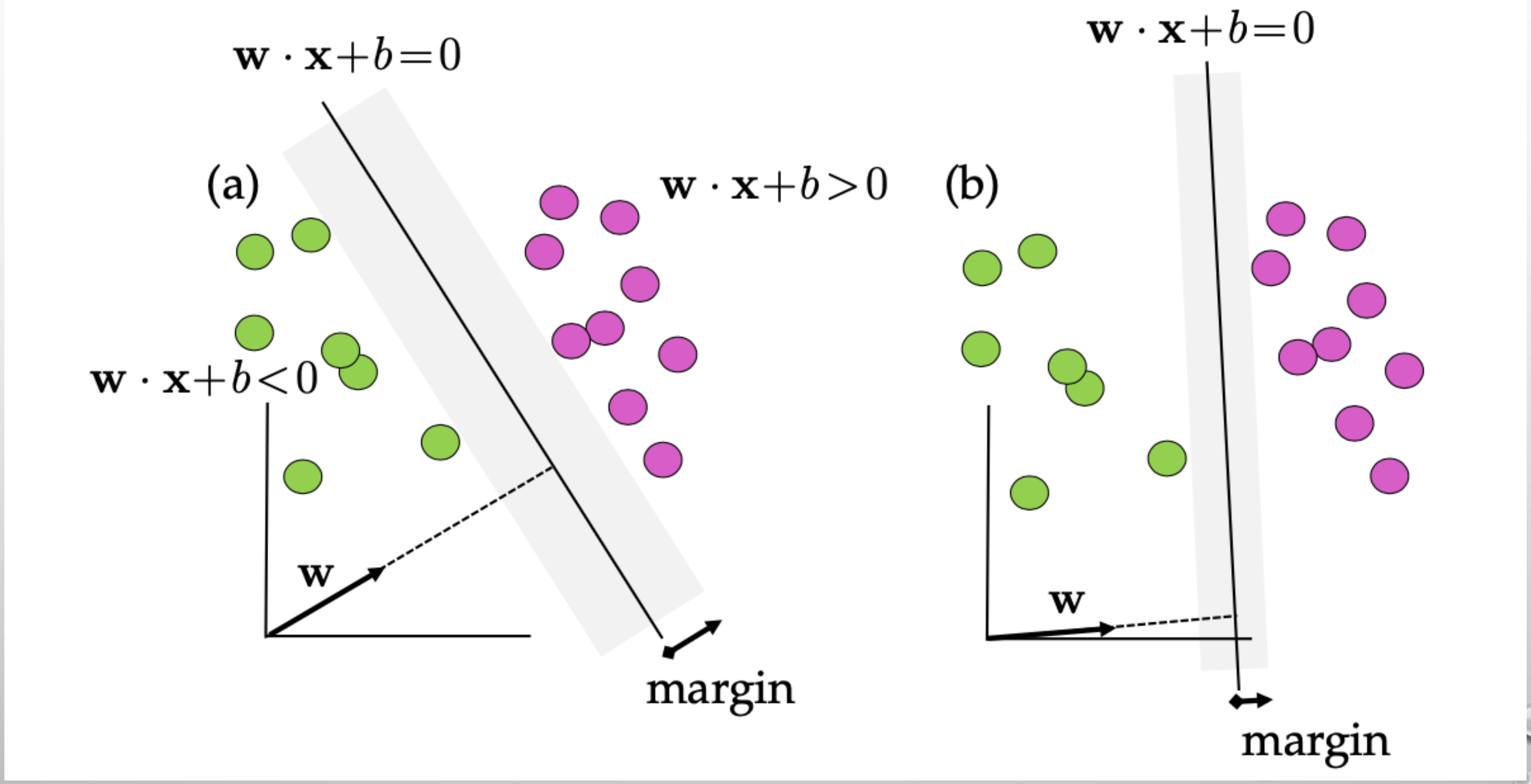
SUPPORT VECTOR MACHINES

ONE OF THE MOST SUCCESSFUL DATA MINING METHODS DEVELOPED TO DATE IS THE *SUPPORT VECTOR MACHINE* (SVM)

LINEAR SUPPORT VERCTOR MACHINES

The key idea of the linear SVM method is to construct a hyperplane: **w·x** + b = 0

where the vector **w** and constant b parametrize the hyperplane



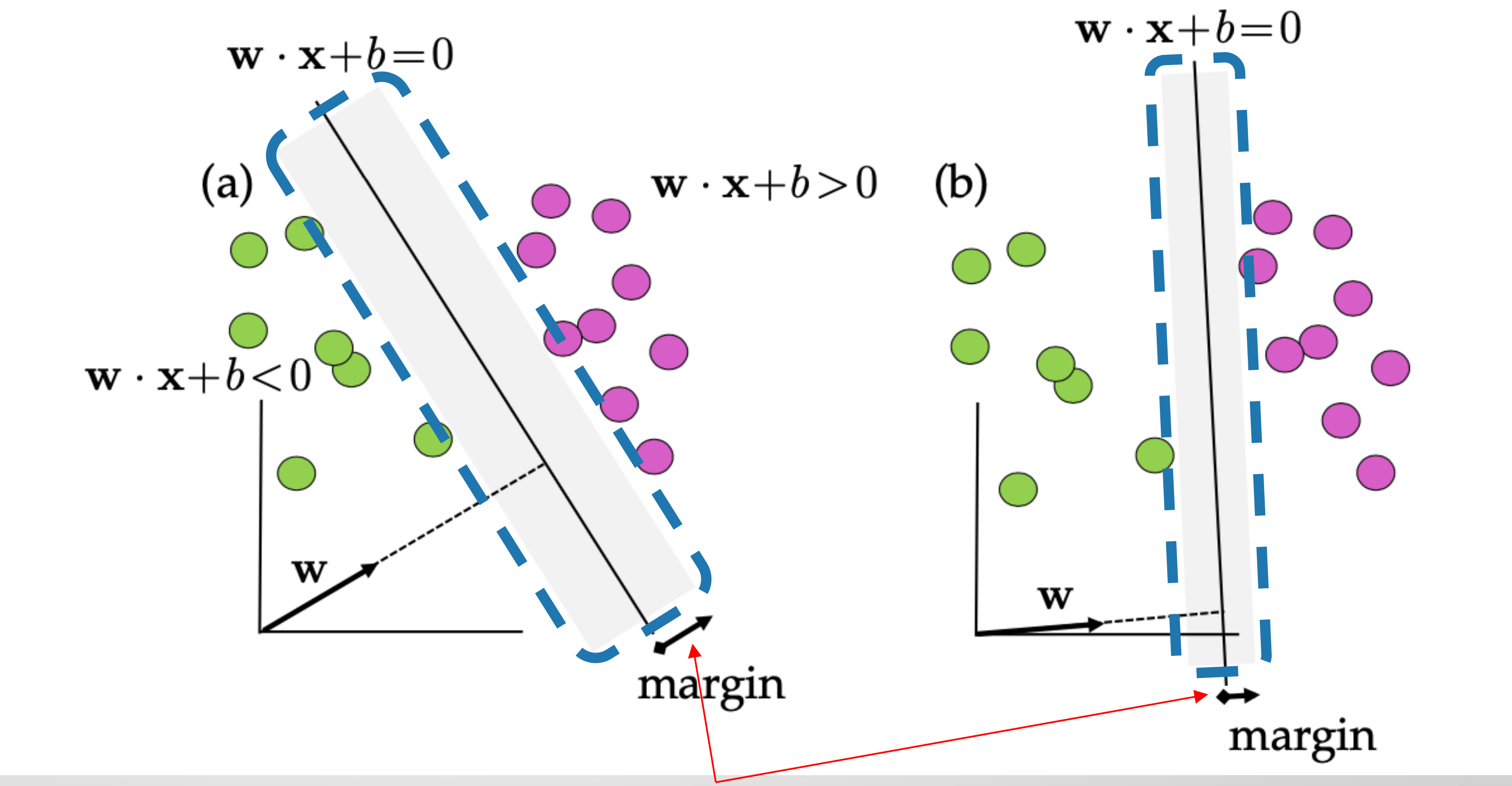
$\mathbf{w}\cdot\mathbf{x}+b=0$
(a)
$\mathbf{w}\cdot\mathbf{x}+b>0$
$\mathbf{w}\cdot\mathbf{x}+b<0$
$\mathbf{w}$
margin
$\mathbf{w}\cdot\mathbf{x}+b=0$
(b)
$\mathbf{w}$
margin

LINEAR SUPPORT VERCTOR MACHINES

- Each has a different value of w and constant b.

- The optimization problem associated with SVM is to not only optimize a decision line which makes the fewest labeling errors for the data,

- but also optimizes the largest margin between the data, shown in the gray region

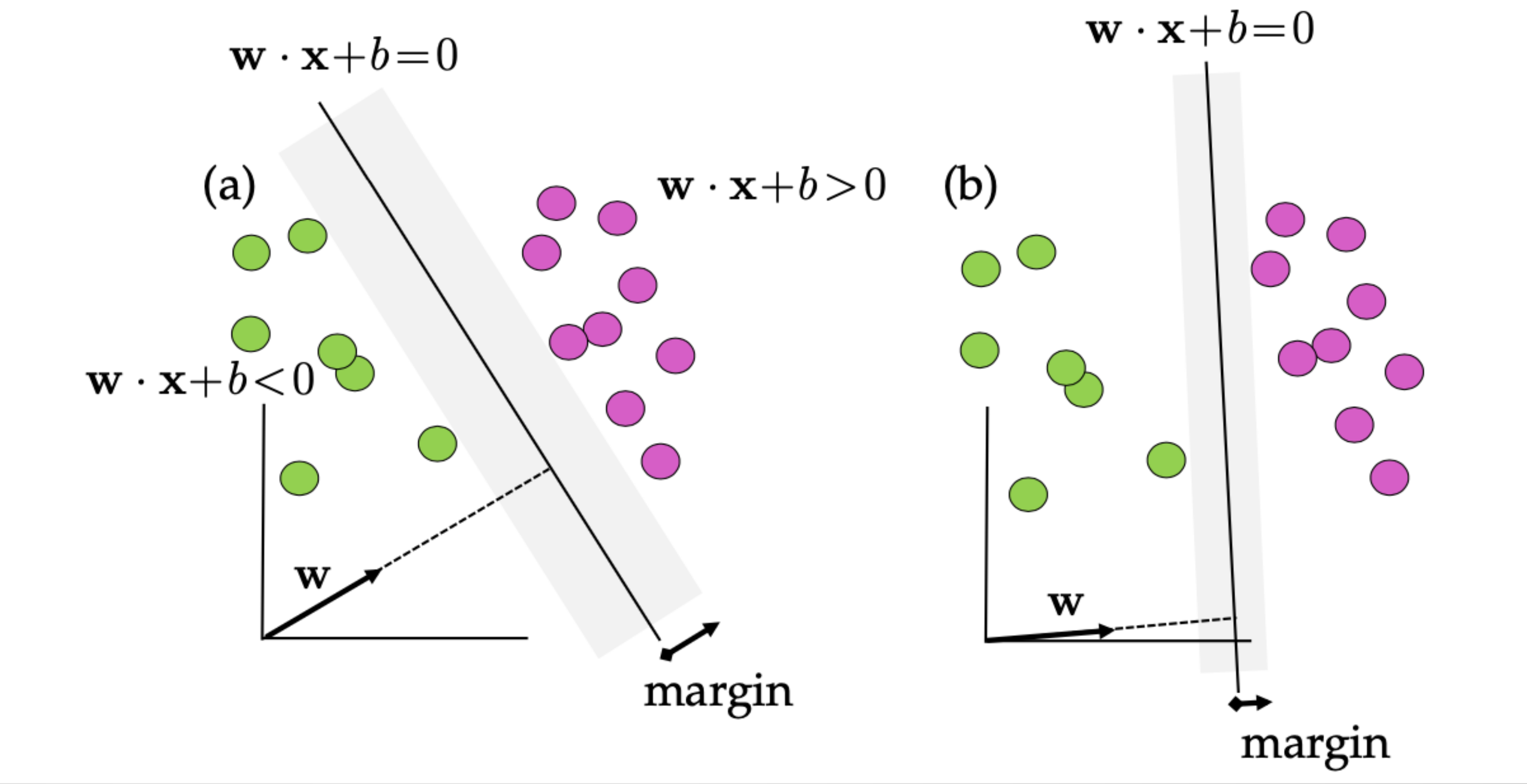


$\mathbf{w}\cdot\mathbf{x}+b=0$
(a)
$\mathbf{w}\cdot\mathbf{x}+b>0$
$\mathbf{w}\cdot\mathbf{x}+b<0$
w
margin
$\mathbf{w}\cdot\mathbf{x}+b=0$
(b)
w
margin

The vectors that determine the boundaries of the margin, i.e. the vectors touching the edge of the gray regions, are termed the *support vectors*

LINEAR SUPPORT VERCTOR MACHINES

Given the hyperplane, a new data point x_j can be classified by simply computing the sign of ($w \cdot x_j + b$)



$\mathbf{w} \cdot \mathbf{x} + b = 0$
(a)
$\mathbf{w} \cdot \mathbf{x} + b > 0$
$\mathbf{w} \cdot \mathbf{x} + b < 0$
$\mathbf{w}$
margin
$\mathbf{w} \cdot \mathbf{x} + b = 0$
(b)
$\mathbf{w}$
margin

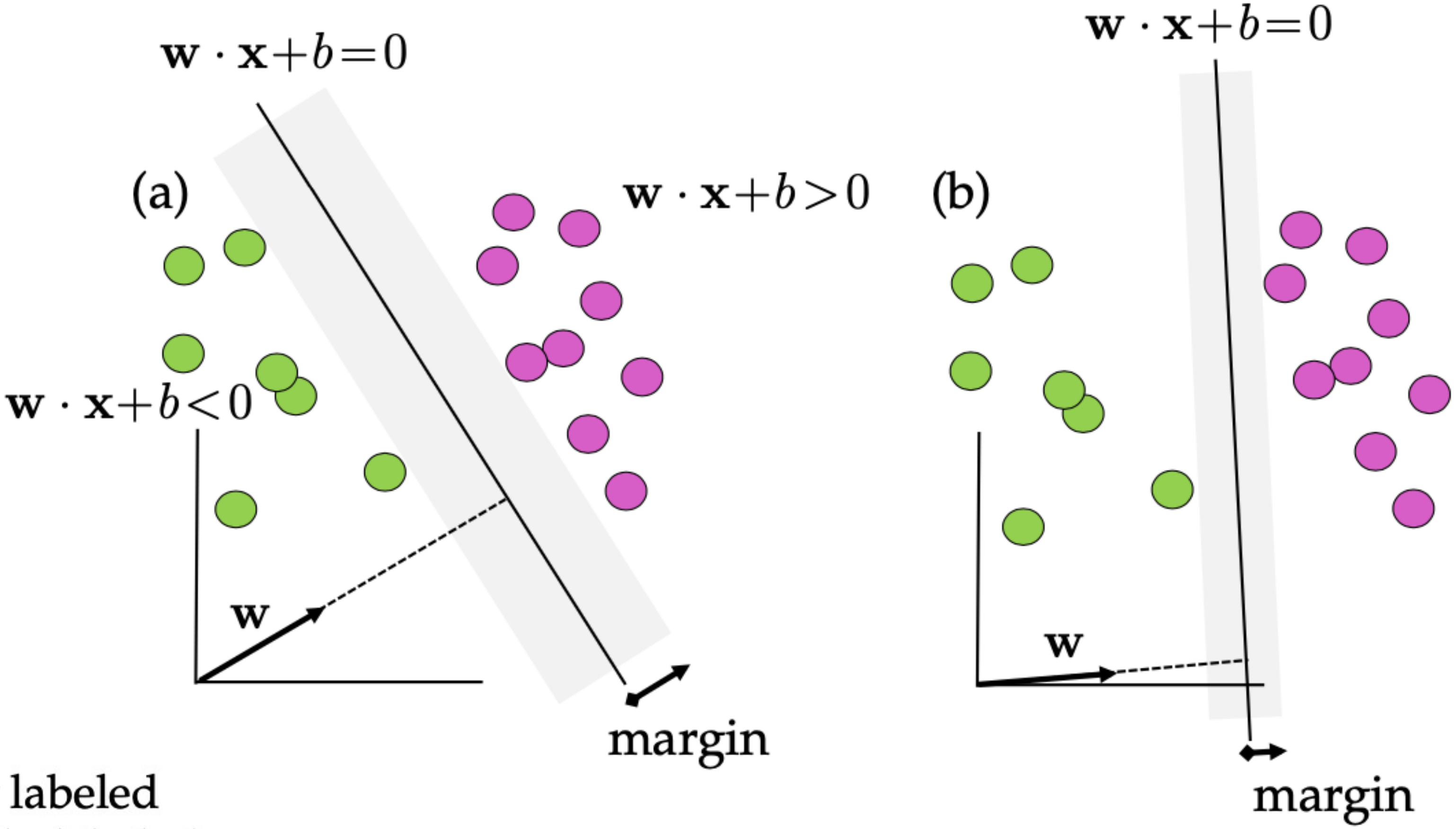
$$\mathbf{y}_j(\mathbf{w} \cdot \mathbf{x}_j + b) = \text{sign}(\mathbf{w} \cdot \mathbf{x}_j + b) = \begin{cases} +1 & \text{magenta ball} \\ -1 & \text{green ball.} \end{cases}$$

LINEAR SUPPORT VERCTOR MACHINES

Critical to the success of the SVM is determining w and b in a principled way.

The optimization is aimed at both minimizing the number of misclassified data points as well as creating the largest margin possible.

To construct the optimization objective function, we define a loss function:



$\mathbf{w}\cdot\mathbf{x}+b=0$
(a)
$\mathbf{w}\cdot\mathbf{x}+b>0$
$\mathbf{w}\cdot\mathbf{x}+b<0$
w
margin
(b)
$\mathbf{w}\cdot\mathbf{x}+b=0$
w
margin

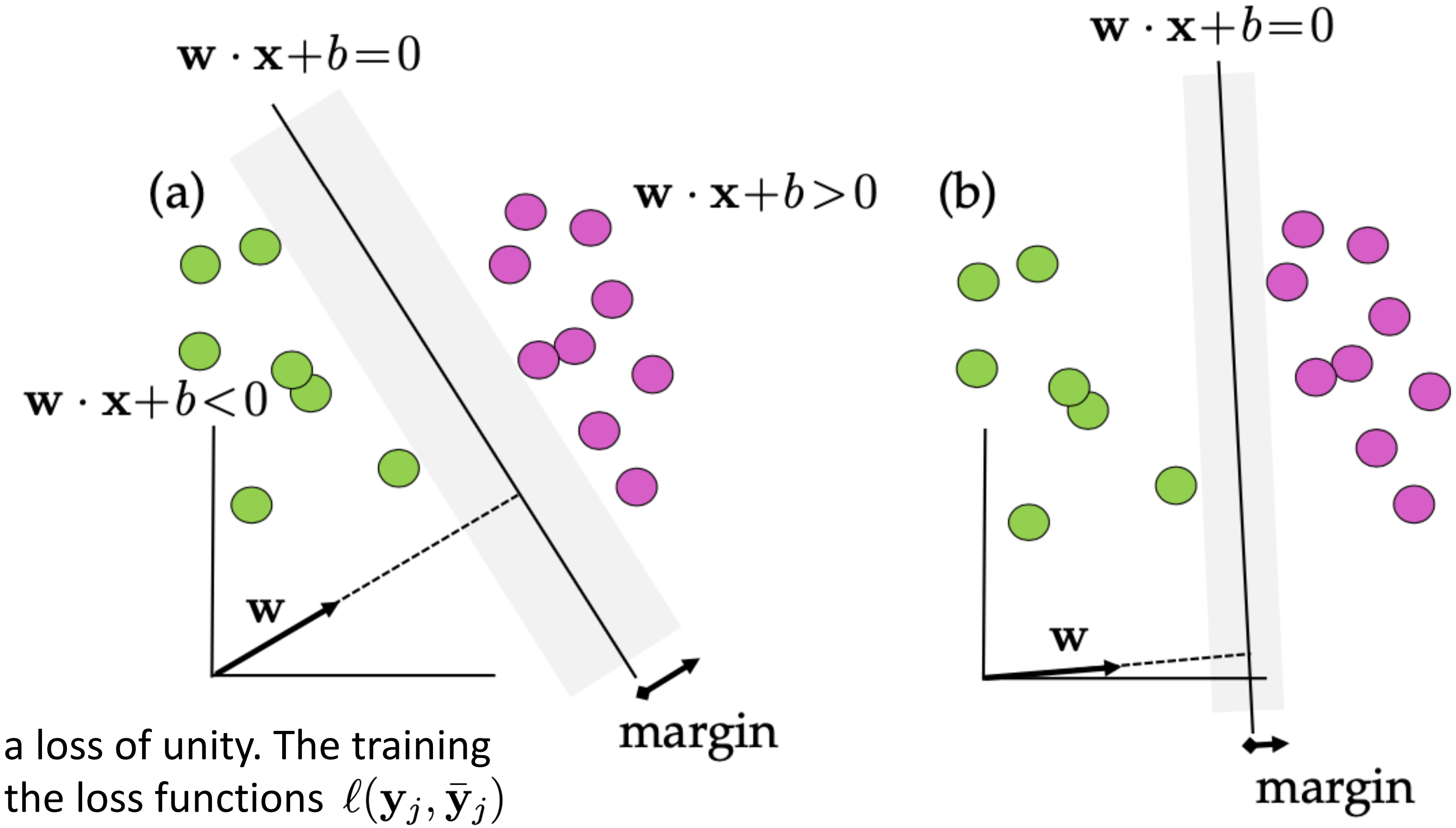
$$\ell(\mathbf{y}_j,\bar{\mathbf{y}}_j)=\begin{cases}0 & \text{if data is correctly labeled}\\ +1 & \text{if data is incorrectly labeled}\end{cases}$$

$$\ell(\mathbf{y}_j,\bar{\mathbf{y}}_j)=\ell(\mathbf{y}_j,\text{sign}(\mathbf{w}\cdot\mathbf{x}_j+b))=\begin{cases}0 & \text{if } \mathbf{y}_j=\text{sign}(\mathbf{w}\cdot\mathbf{x}_j+b)\\ +1 & \text{if } \mathbf{y}_j\neq\text{sign}(\mathbf{w}\cdot\mathbf{x}_j+b)\end{cases}$$

LINEAR SUPPORT VERCTOR MACHINES

Critical to the success of the SVM is determining w and b in a principled way.

The optimization is aimed at both minimizing the number of misclassified data points as well as creating the largest margin possible.
To construct the optimization objective function, we define a loss function:


$\mathbf{w}\cdot\mathbf{x}+b=0$
(a)
$\mathbf{w}\cdot\mathbf{x}+b>0$
$\mathbf{w}\cdot\mathbf{x}+b<0$
$\mathbf{w}$
margin
$\mathbf{w}\cdot\mathbf{x}+b=0$
(b)
$\mathbf{w}$
margin

Thus, each mislabeled point produces a loss of unity. The training error over m data points is the sum of the loss functions $\ell(\mathbf{y}_j, \bar{\mathbf{y}}_j)$

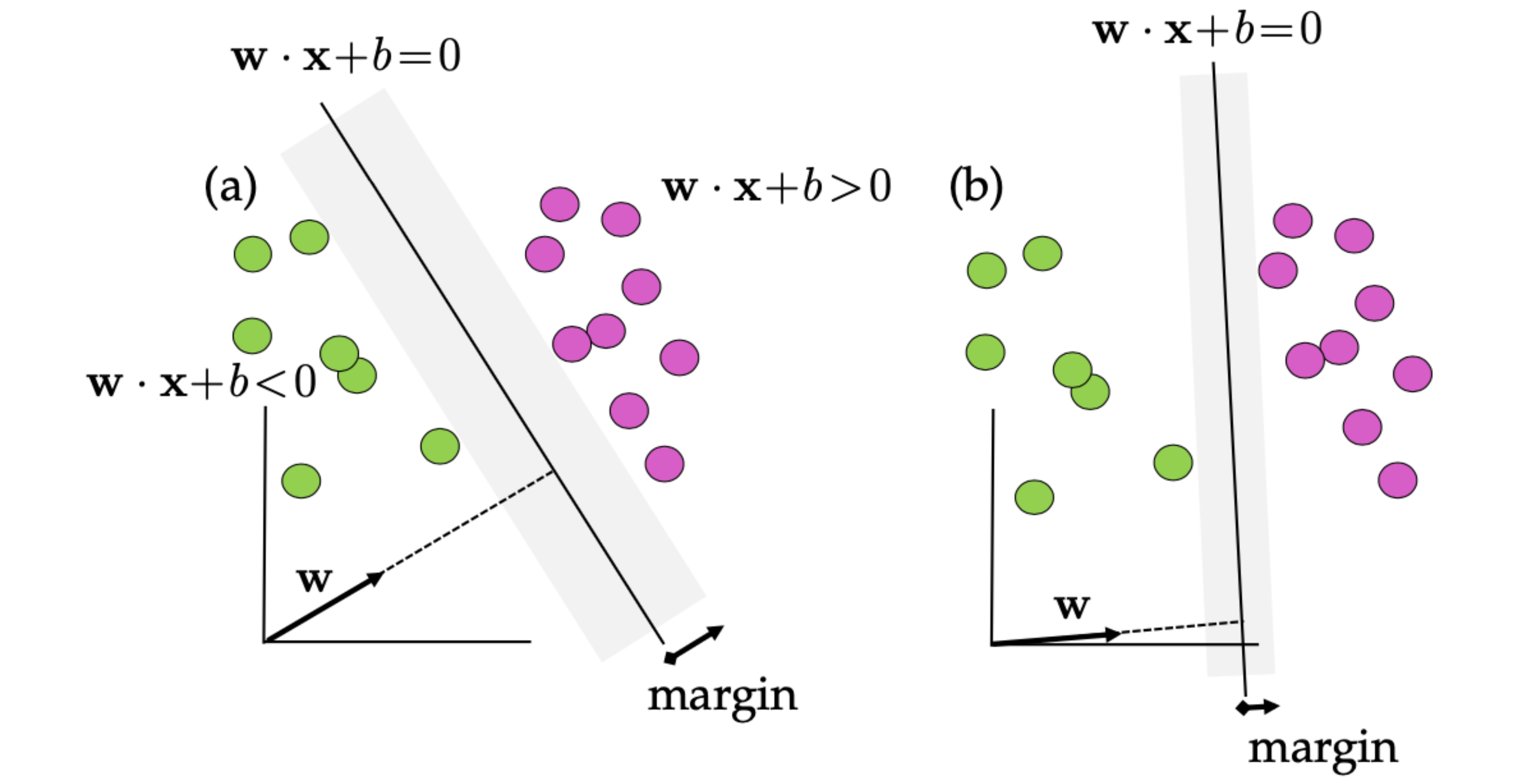
$$\ell(\mathbf{y}_j, \bar{\mathbf{y}}_j) = \ell(\mathbf{y}_j, \text{sign}(\mathbf{w}\cdot\mathbf{x}_j + b)) = \begin{cases} 0 & \text{if } \mathbf{y}_j = \text{sign}(\mathbf{w}\cdot\mathbf{x}_j + b) \\ +1 & \text{if } \mathbf{y}_j \neq \text{sign}(\mathbf{w}\cdot\mathbf{x}_j + b) \end{cases}$$

LINEAR SUPPORT VERCTOR MACHINES

In addition to minimizing the loss function, the goal is also to make the margin as large as possible. We can then frame the linear SVM optimization problem as:

$$\underset{\mathbf{w},b}{\operatorname{argmin}} \sum_{j=1}^{m} \ell(\mathbf{y}_j, \bar{\mathbf{y}}_j) + \frac{1}{2}\|\mathbf{w}\|^2$$

subject to $\min_j |\mathbf{x}_j \cdot \mathbf{w}| = 1.$



$\mathbf{w}\cdot\mathbf{x}+b=0$
(a)
$\mathbf{w}\cdot\mathbf{x}+b>0$
$\mathbf{w}\cdot\mathbf{x}+b<0$
$\mathbf{w}$
margin
$\mathbf{w}\cdot\mathbf{x}+b=0$
(b)
$\mathbf{w}$
margin

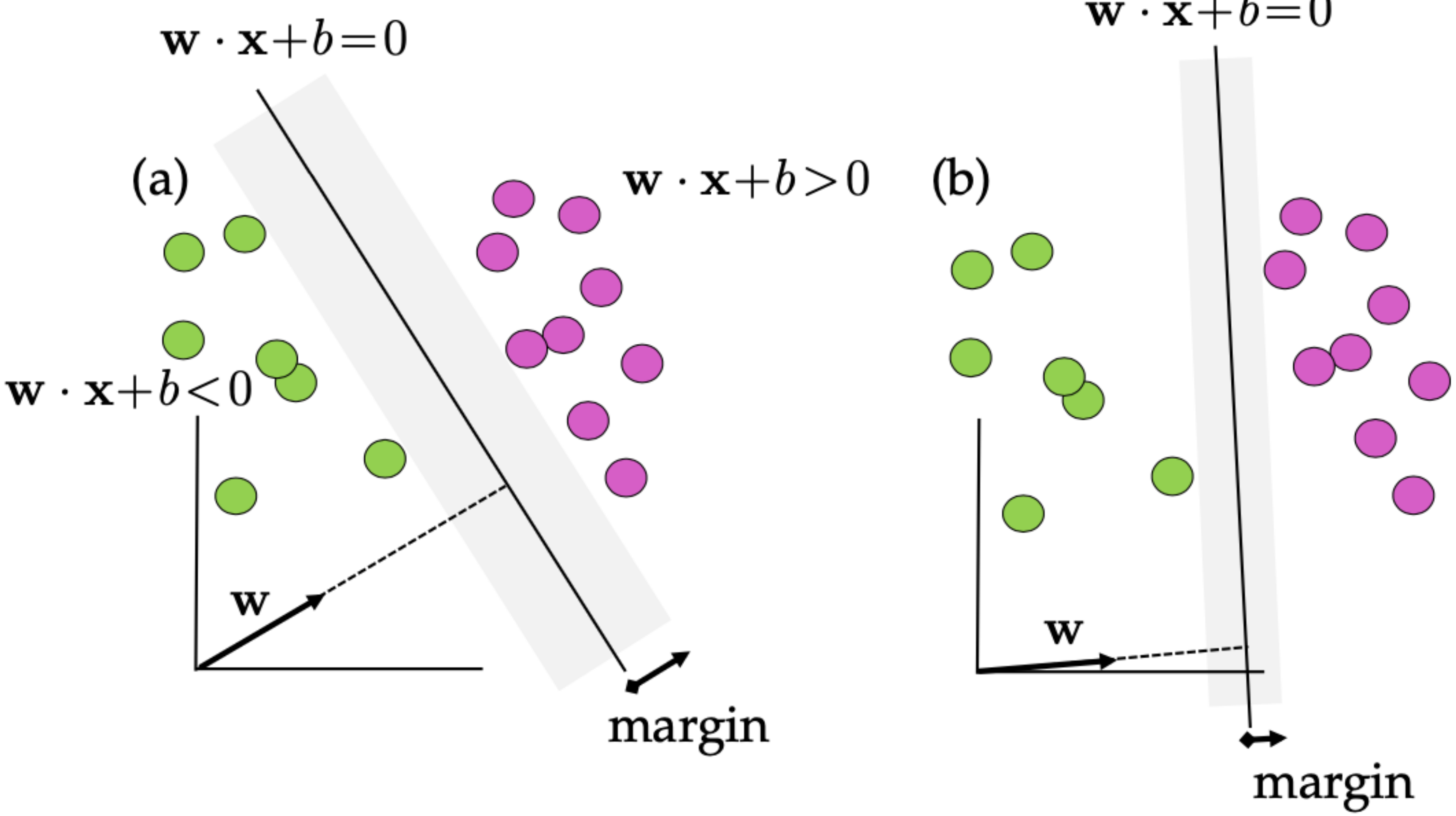
Although this is a concise statement of the optimization problem, the fact that the loss function is discrete and constructed from ones and zeros makes it very difficult to optimize.
Most optimization algorithms are based on some form of gradient descent which requires smooth objective functions in order to compute derivatives or gradients to update the solution.

LINEAR SUPPORT VERCTOR MACHINES

A more common formulation then is given by:

$$\underset{\mathbf{w},b}{\operatorname{argmin}} \sum_{j=1}^{m} H(\mathbf{y}_j, \bar{\mathbf{y}}_j) + \frac{1}{2}\|\mathbf{w}\|^2$$

Subject to $\min_{j} |\mathbf{x}_j \cdot \mathbf{w}| = 1.$



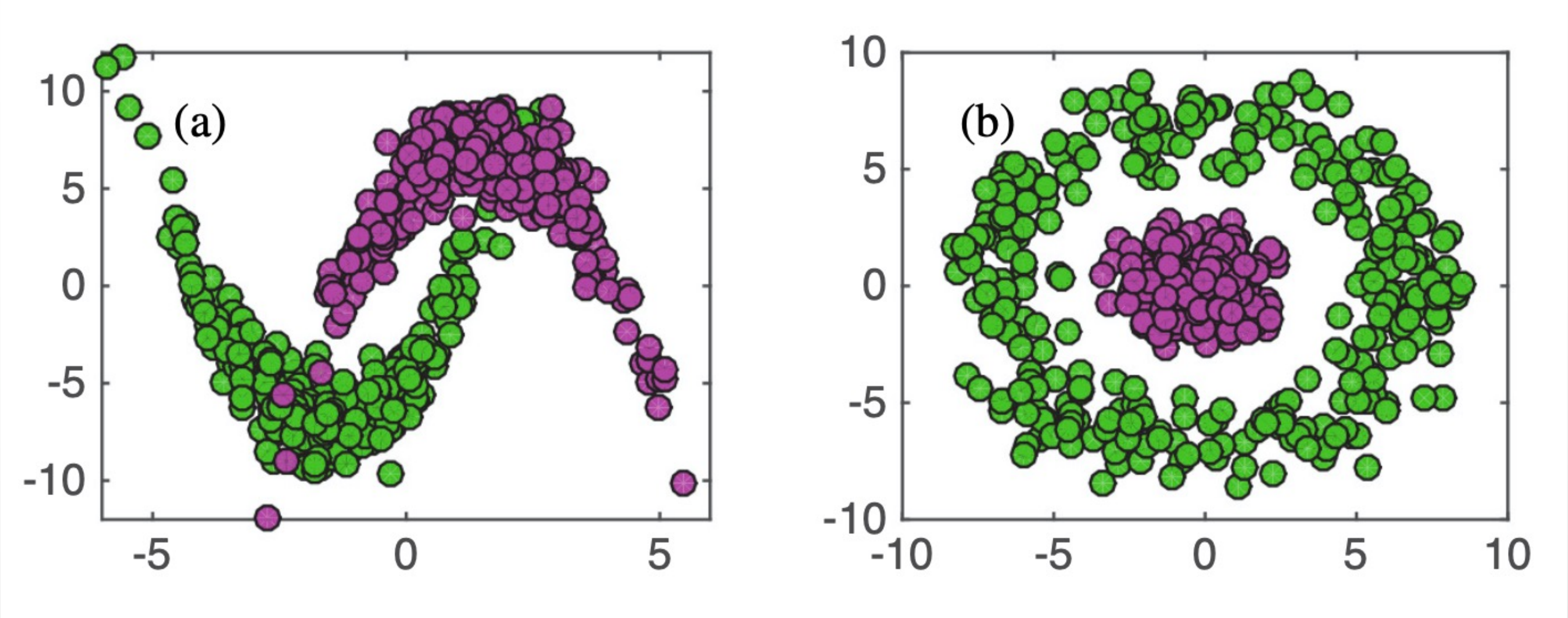
$\mathbf{w}\cdot\mathbf{x}+b=0$
(a)
$\mathbf{w}\cdot\mathbf{x}+b>0$
$\mathbf{w}\cdot\mathbf{x}+b<0$
$\mathbf{w}$
margin
$\mathbf{w}\cdot\mathbf{x}+b=0$
(b)
$\mathbf{w}$
margin

Where

$$H = (y_i, \bar{y}_i) = \max(0, 1 - y_i \cdot \bar{y}_i)$$

is called a Hinge loss function. This is a smooth function that counts the number of errors in a linear way and that allows for piecewise differentiation so that standard optimization routines can be employed.

NONLINEAR SUPPORT VECTOR MACHINES



How to classify this sort of data?

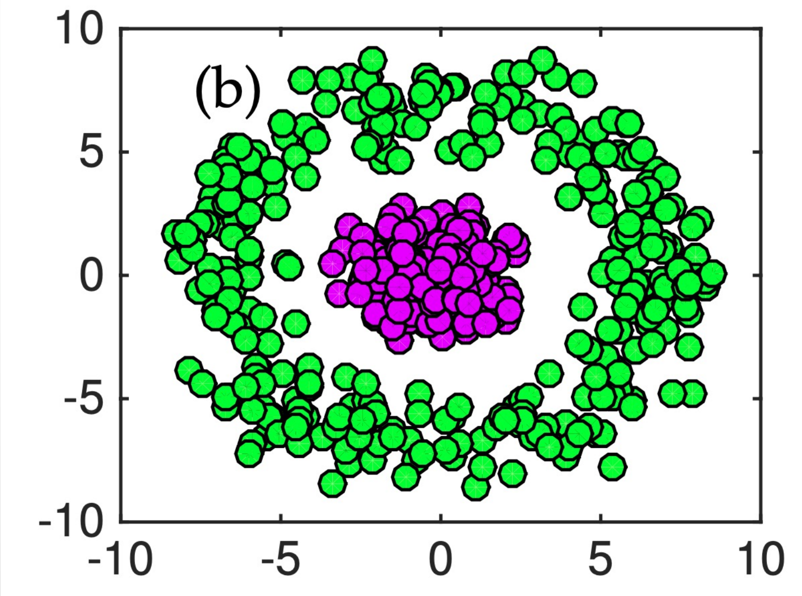
We must enrich, i.e, maps the data into a nonlinear, higher-dimensional space

$$\mathbf{x} \mapsto \mathbf{\Phi}(\mathbf{x}).$$

We can call them, new ***observables*** of the data

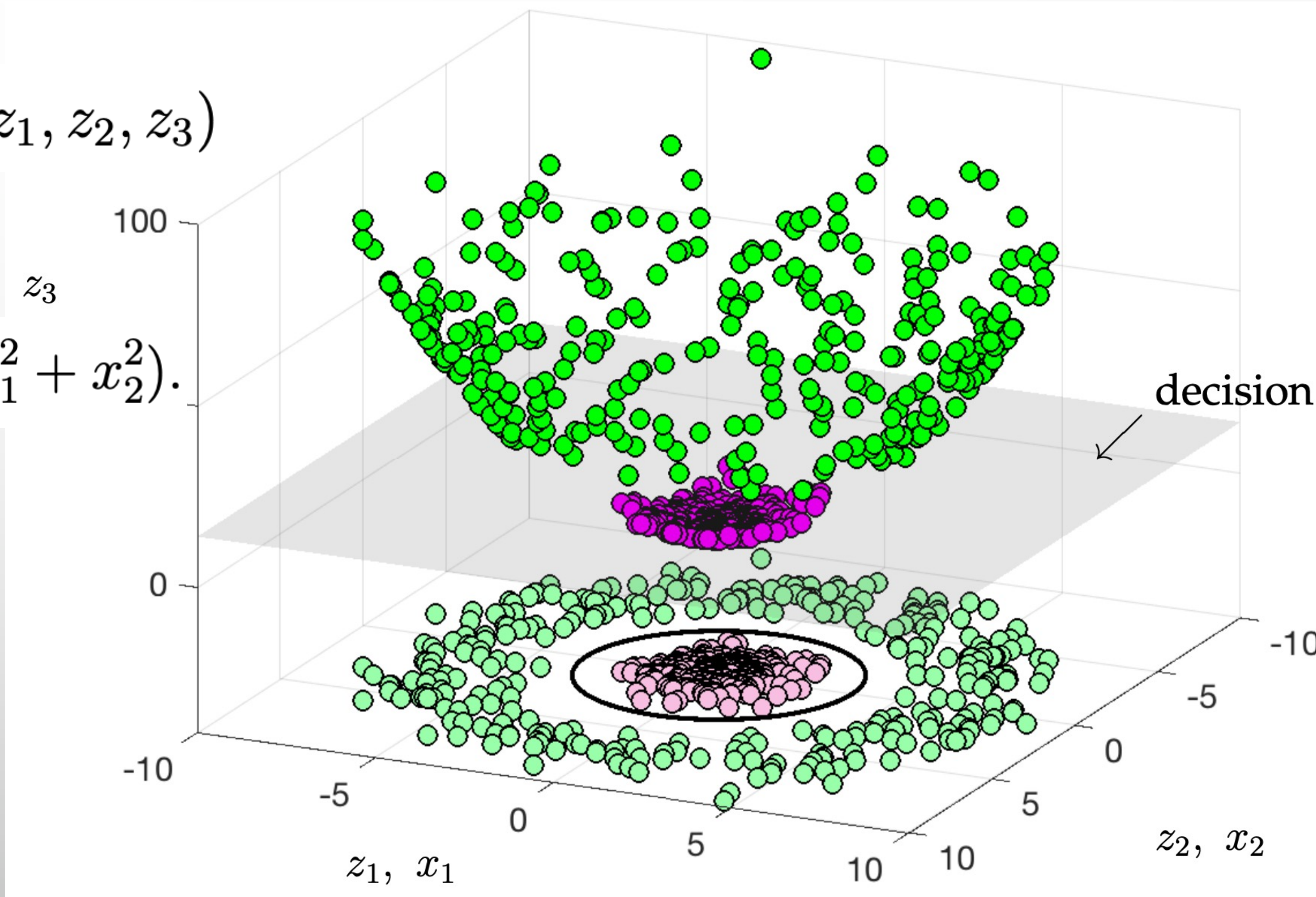
NONLINEAR SUPPORT VECTOR MACHINES

- The SVM algorithm now learns the hyperplanes that optimally split the data into distinct clusters in a new space. Thus, one now considers the hyperplane function: $f(\mathbf{x}) = \mathbf{w} \cdot \boldsymbol{\Phi}(\mathbf{x}) + b,$



$$(x_1, x_2) \mapsto (z_1, z_2, z_3)$$

$$:= (x_1, x_2, x_1^2 + x_2^2).$$



NONLINEAR SUPPORT VERCTOR MACHINES

The ability of SVM to embed in higher-dimensional nonlinear spaces makes it one of the most successful machine learning algorithms developed.

The underlying optimization algorithm

$$\underset{\mathbf{w},b}{\operatorname{argmin}} \sum_{j=1}^{m} H(\mathbf{y}_j, \bar{\mathbf{y}}_j) + \frac{1}{2}\|\mathbf{w}\|^2$$

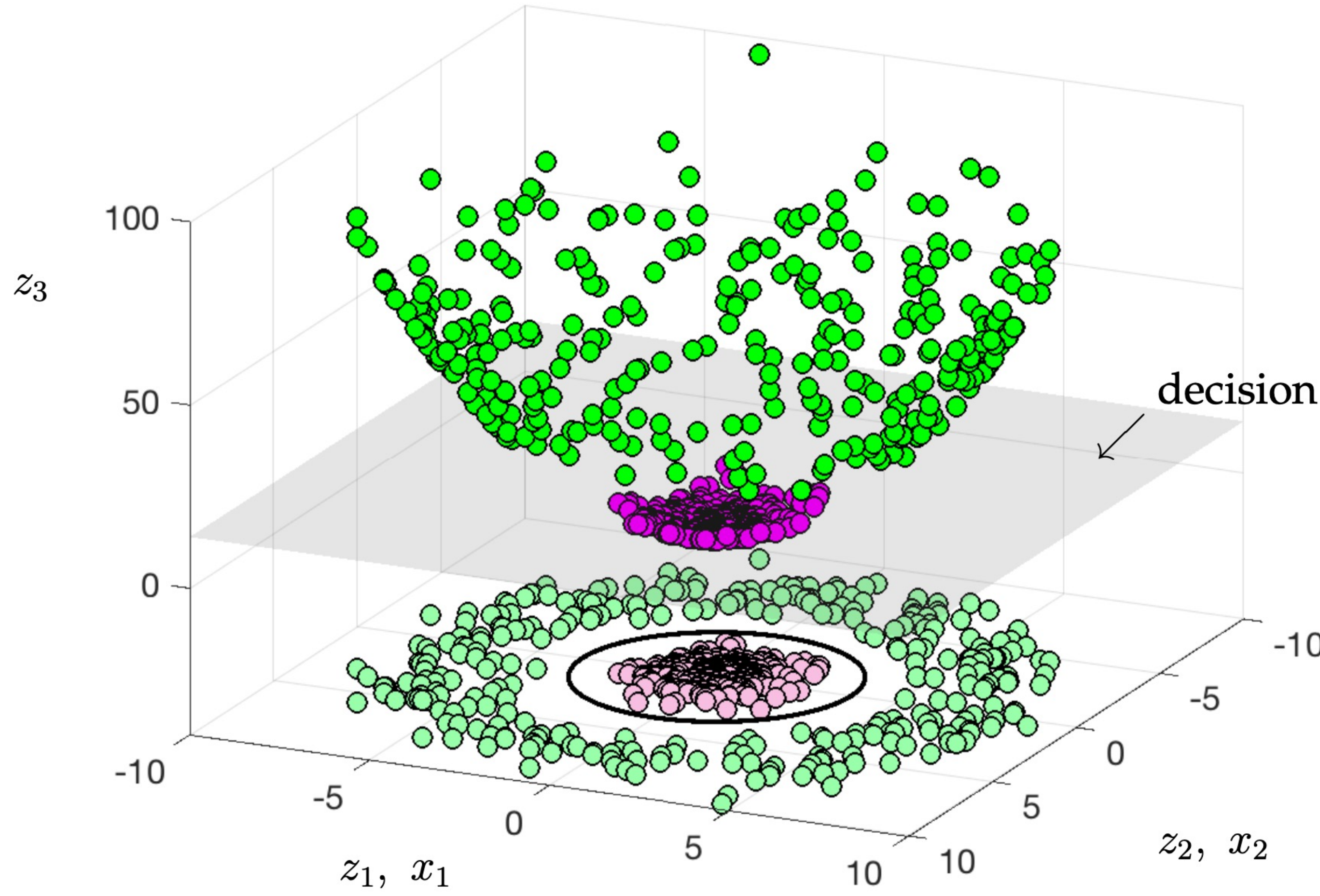
subject to $$\min_j |\mathbf{x}_j \cdot \mathbf{w}| = 1.$$

remains unchanged, except that the previous labeling function, $\bar{y}_j = sign(w \cdot x_j + b)$, is now:

$$\bar{\mathbf{y}}_j = \mathbf{sign}(\mathbf{w} \cdot \Phi(\mathbf{x}_j) + b).$$

The function Φ(x) specifies the enriched space of observables.

As a rule, more features are better for classification.



KERNEL METHODS FOR SUPPORT VECTOR MACHINES

KERNEL METHODS FOR SVM

From Linear SVM

$$\operatorname*{argmin}_{\mathbf{w},b} \sum_{j=1}^{m} H(\mathbf{y}_j, \bar{\mathbf{y}}_j) + \frac{1}{2}\boxed{\|\mathbf{w}\|^2} \quad \textbf{subject to} \quad \min_j |\mathbf{x}_j \cdot \mathbf{w}| = 1$$

$$\mathbf{w} = \sum_{j=1}^{m} \alpha_j \Phi(\mathbf{x}_j)$$

From nonlinear SVM

$$f(\mathbf{x}) = \mathbf{w} \cdot \Phi(\mathbf{x}) + b \quad \Rightarrow \quad f(\mathbf{x}) = \sum_{j=1}^{m} \alpha_j \Phi(\mathbf{x}_j) \cdot \Phi(\mathbf{x}) + b.$$

If we define the KERNEL function as:

$$K(\mathbf{x}_j, \mathbf{x}) = \Phi(\mathbf{x}_j) \cdot \Phi(\mathbf{x})\,.$$

$$\operatorname*{argmin}_{\boldsymbol{\alpha},b} \sum_{j=1}^{m} H(\mathbf{y}_j, \bar{\mathbf{y}}_j) + \frac{1}{2}\|\sum_{j=1}^{m} \alpha_j \Phi(\mathbf{x}_j)\|^2 \quad \textbf{subject to} \quad \min_j |\mathbf{x}_j \cdot \mathbf{w}| = 1$$

Radial basis functions (RBF): $K(\mathbf{x}_j, \mathbf{x}) = \exp\left(-\gamma \|\mathbf{x}_j - \mathbf{x}\|^2\right)$

Polynomial kernel: $K(\mathbf{x}_j, \mathbf{x}) = (\mathbf{x}_j \cdot \mathbf{x} + 1)^N$