GENERAL ASSEMBLY

INTRO TO DATA SCIENCE

LECTURE 6: SUPPORT VECTOR MACHINES

December 8, 2014

DAT11-SF

LAST TIME:

- LINEAR REGRESSION
- LOGISTIC REGRESSION

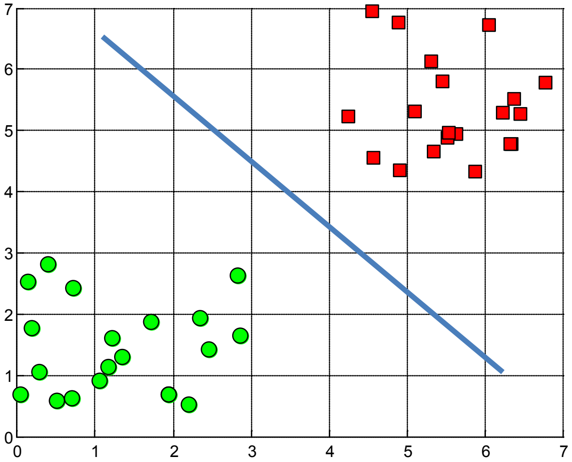
I. SUPPORT VECTOR MACHINES
II. MAXIMUM MARGIN HYPERPLANES
III. SOFT MARGINS
IV. NONLINEAR CLASSIFICATION

LAB:
V. SVM IN SCIKIT-LEARN

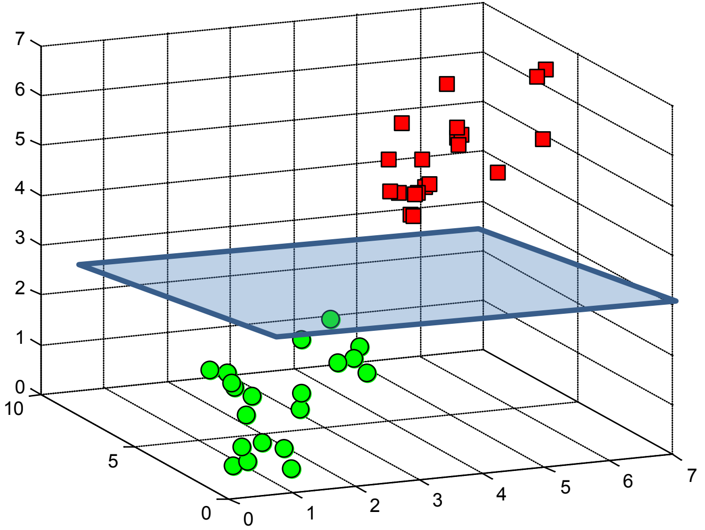
I. SUPPORT VECTOR MACHINES

LINEAR DECISION SURFACES

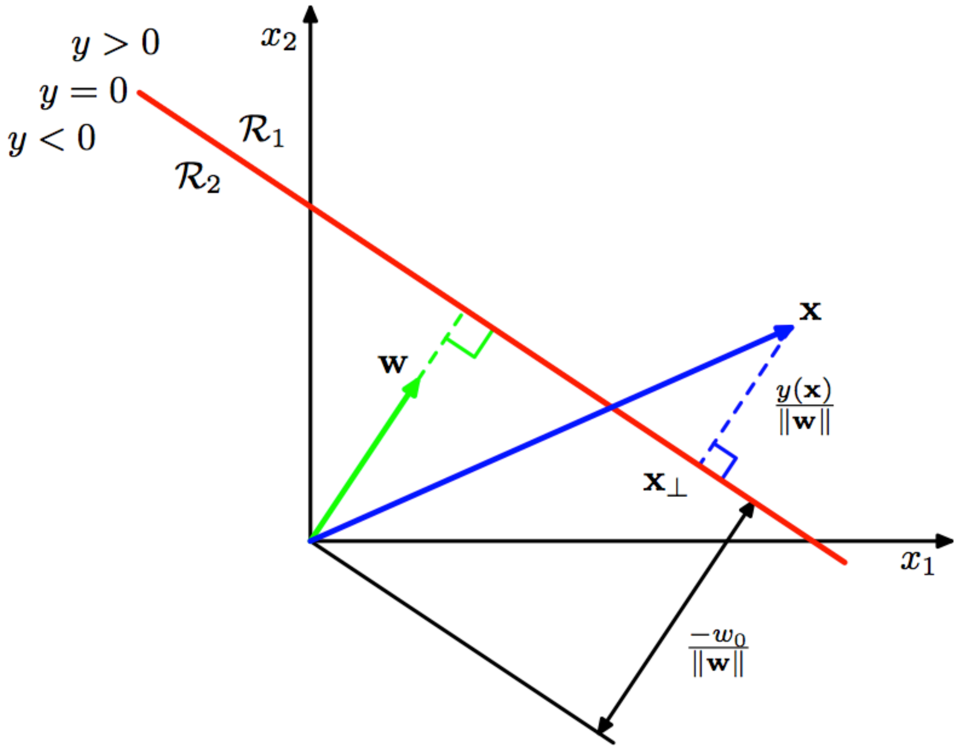
A hyperplane in $\mathbb{R}^2$ is a line



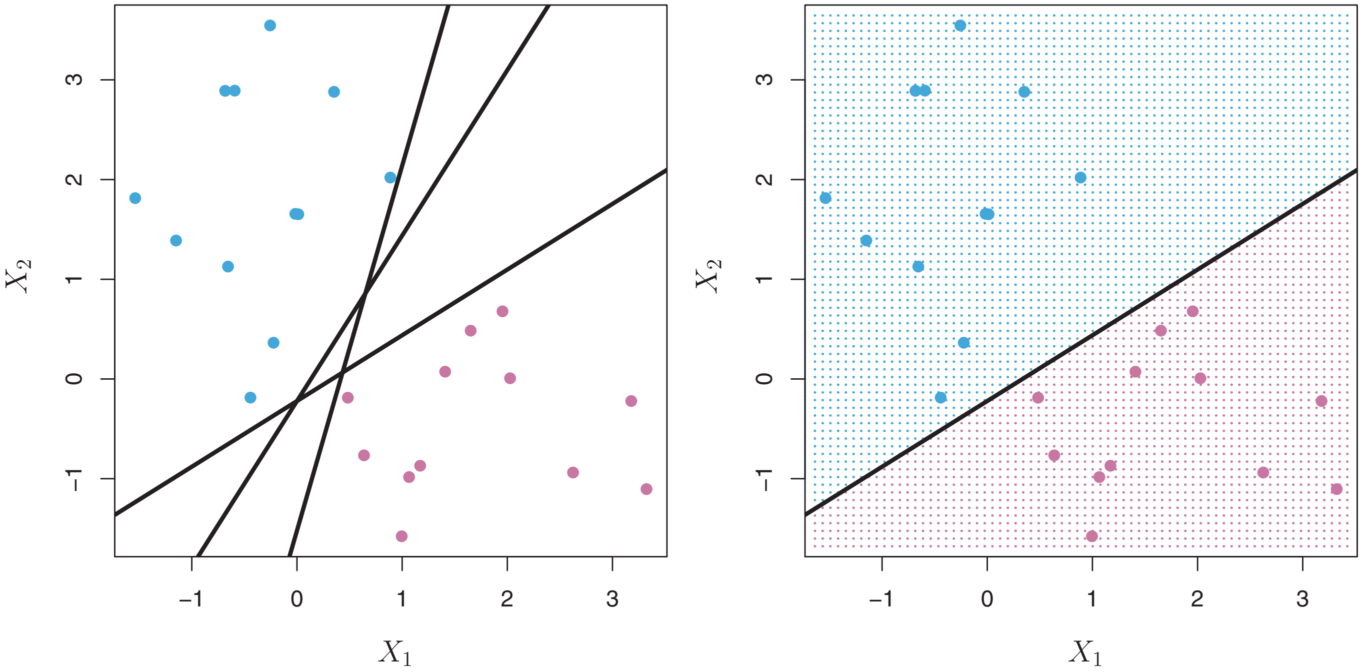
A hyperplane in $\mathbb{R}^3$ is a plane



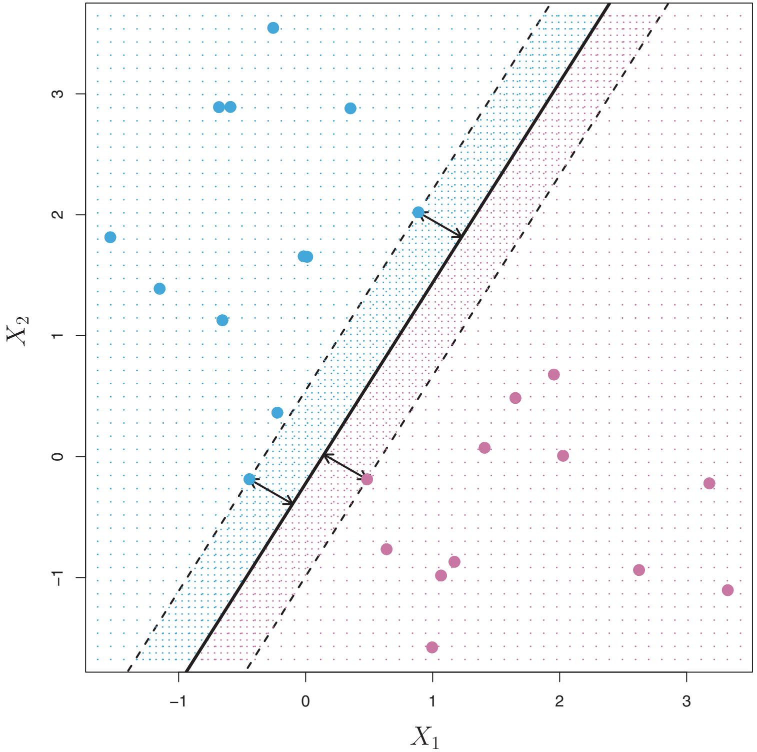
PLANE GEOMETRY



DECISION SURFACES



MAXIMAL MARGIN HYPERPLANE



Q: How is the decision boundary derived?
A: Using *geometric reasoning* (as opposed to the algebraic reasoning we've used to derive other classifiers).

The goal of an SVM is to create the linear decision boundary with the largest margin. This is commonly called the **maximum margin hyperplane**.

Q: What is a support vector machine?

A: A binary linear classifier whose decision boundary is *explicitly* constructed to minimize generalization error.

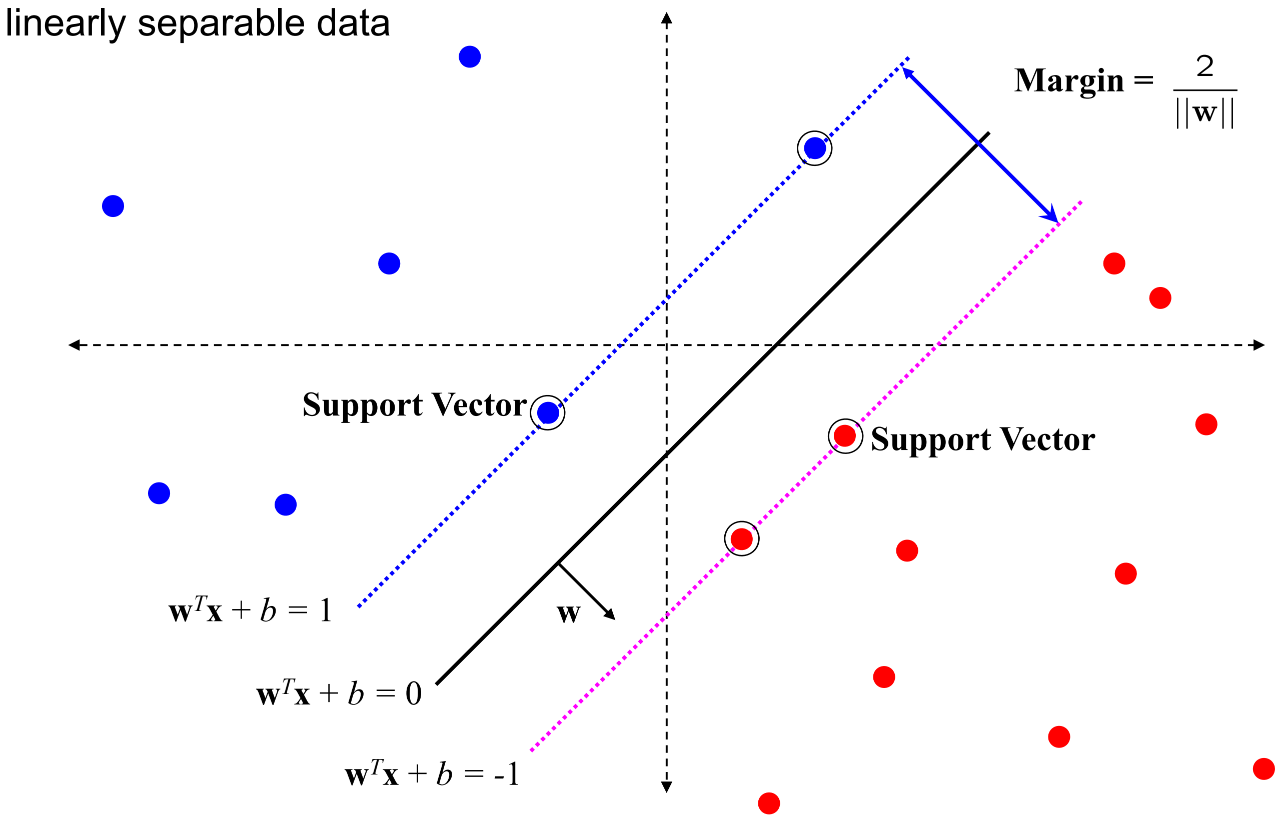
recall:

binary classifier – solves two-class problem

linear classifier – creates linear decision boundary (in 2d)

II. MAXIMUM MARGIN HYPERPLANES

MAXIMUM MARGIN CLASSIER


linearly separable data
Margin = $\frac{2}{||\mathbf{w}||}$
Support Vector
Support Vector
$\mathbf{w}$
$\mathbf{w}^T\mathbf{x} + b = 1$
$\mathbf{w}^T\mathbf{x} + b = 0$
$\mathbf{w}^T\mathbf{x} + b = -1$

All of the decision boundaries we've seen so far have split the data perfectly; eg, the data are **linearly separable**, and therefore the training error is 0.

The optimization problem that this SVM solves is:

$$\begin{aligned} &\underset{\mathbf{w},b}{\text{minimize}} && \tfrac{1}{2}||\mathbf{w}||^2 \\ &\text{subject to:} && y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 \quad i = 1, \ldots, n\,. \end{aligned}$$

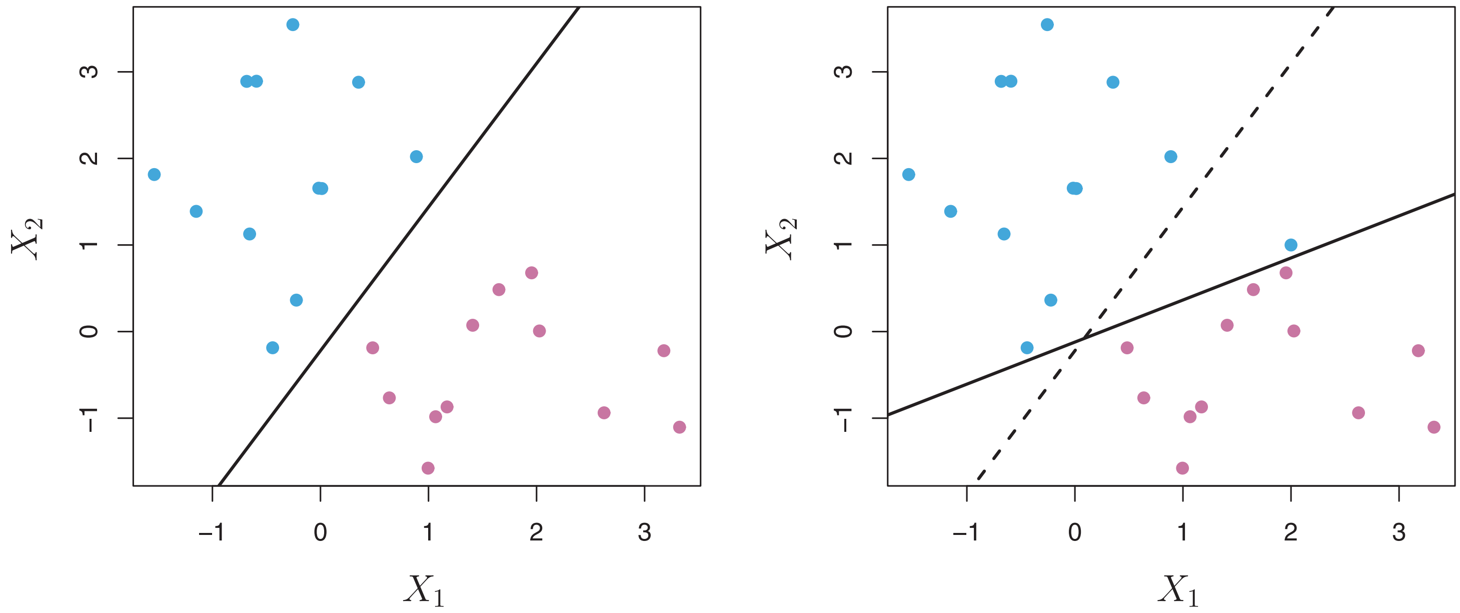
Notice that the margin depends only on a *subset* of the training data; namely, those points that are nearest to the decision boundary.

These points are called the **support vectors**.

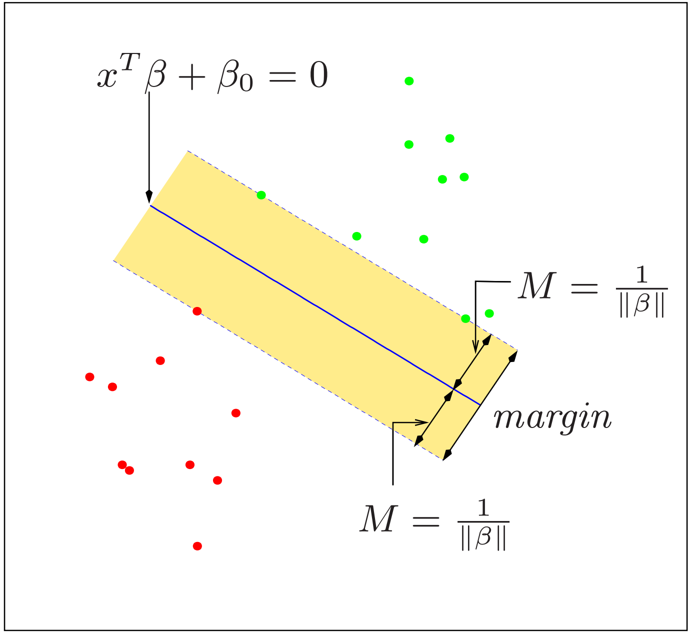
The other points (far from the decision boundary) don’t affect the construction of the mmh at all!

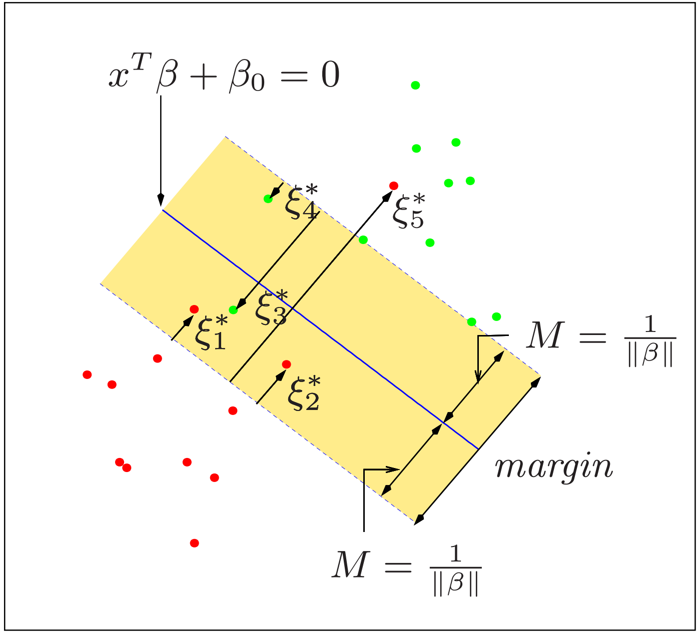
III. SOFT MARGINS

SOFT MARGIN CLASSIER



SOFT MARGIN CLASSIER





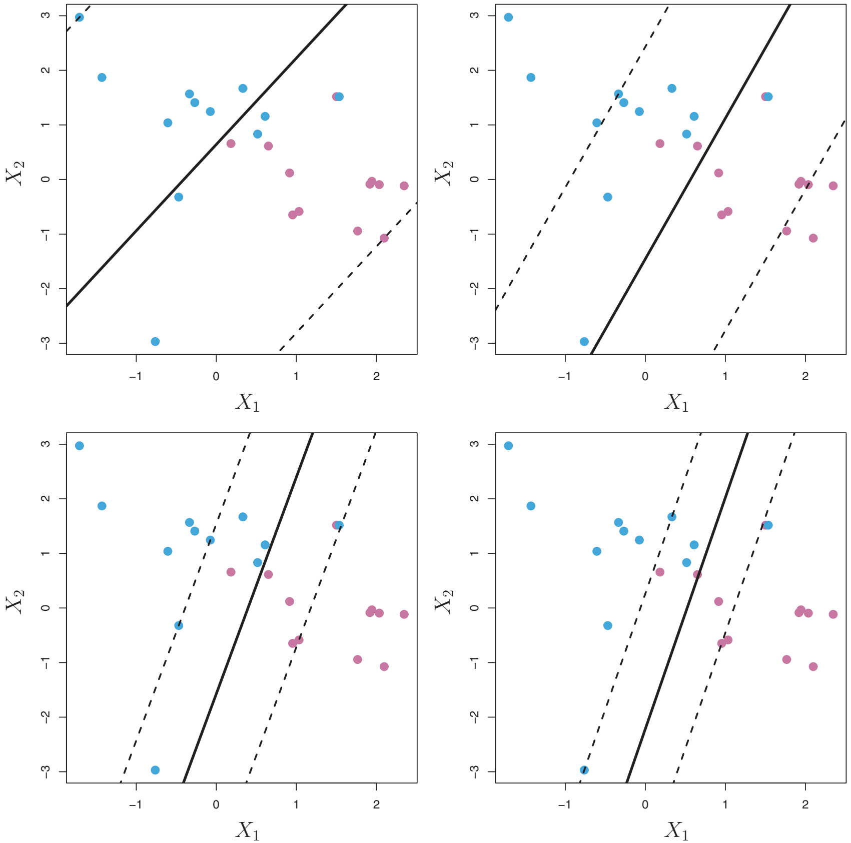
SLACK VARIABLES

Slack variables ξ_i generalize the optimization problem to permit some misclassified training records (which come at a cost C).

The resulting **soft margin classifier** is given by:

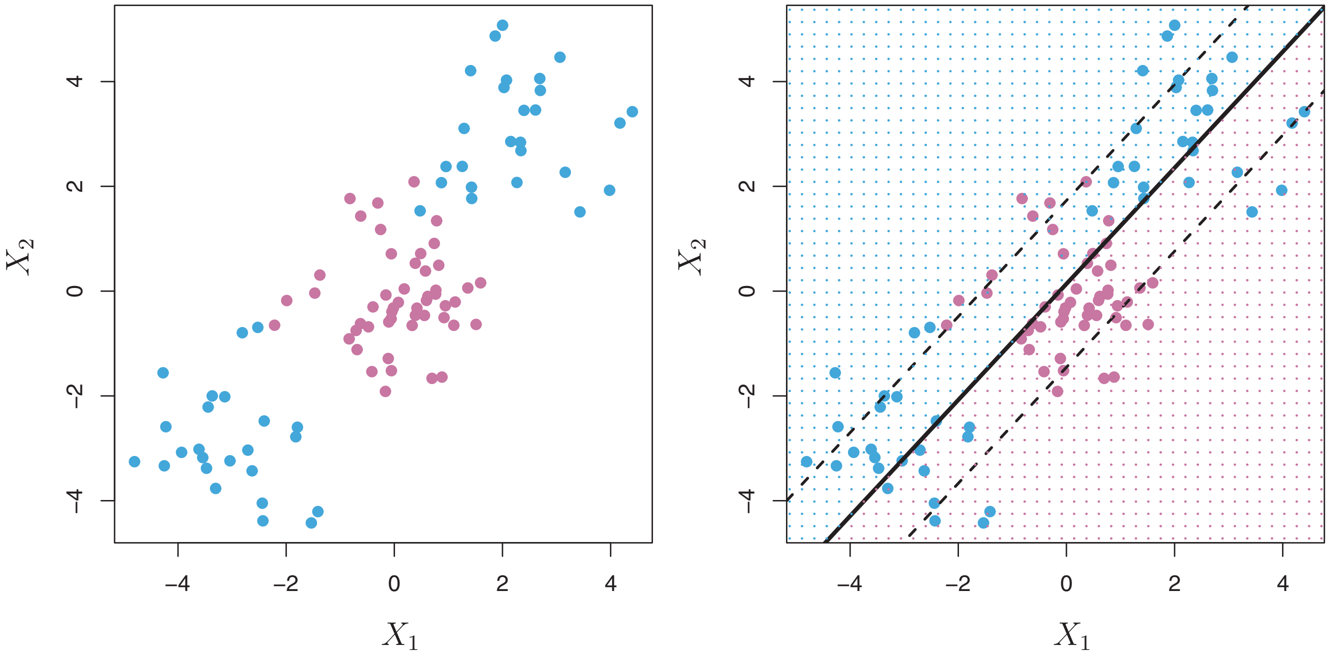
$$\begin{aligned} \underset{\mathbf{w},b}{\text{minimize}} \quad & \tfrac{1}{2}||\mathbf{w}||^2 + C\textstyle\sum_{i=1}^{n}\xi_i \\ \text{subject to:} \quad & y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0. \end{aligned}$$

SOFT MARGIN CLASSIER

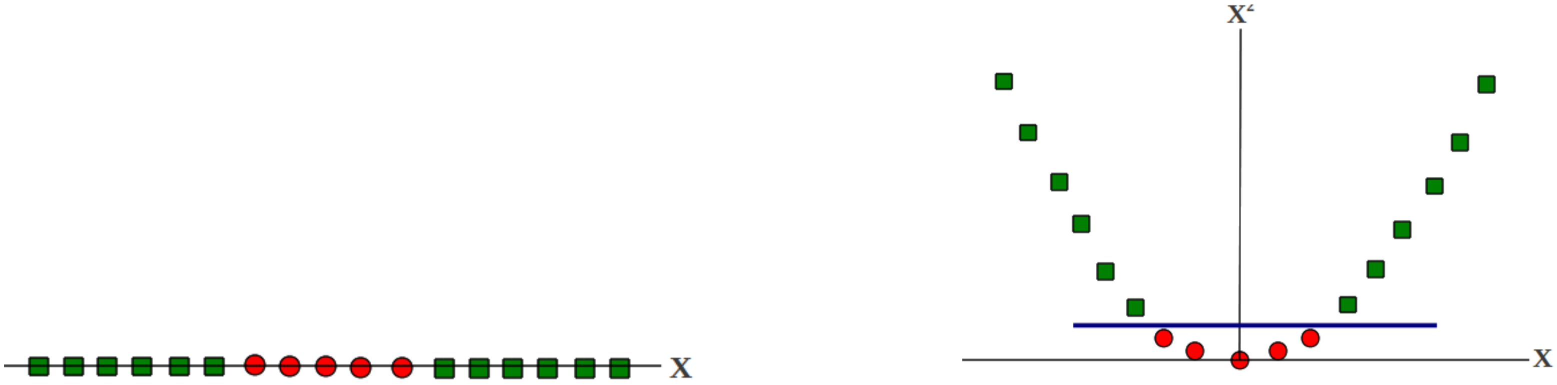


IV. NONLINEAR CLASSIFICATION

NON-LINEAR BOUNDARIES

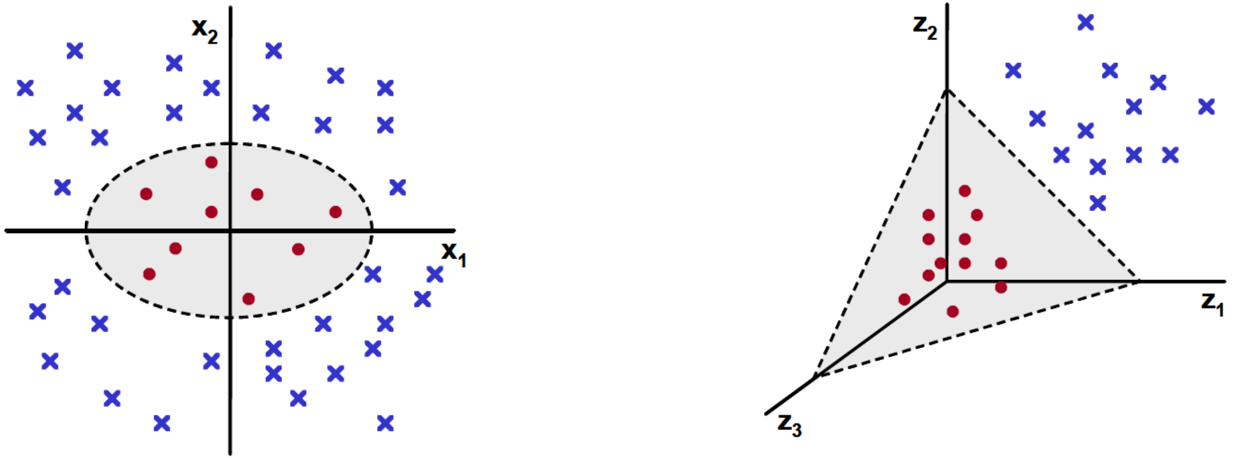


NON - LINEAR BOUNDARIES



$$x \to \{x, x^2\}$$

NON LINEAR BOUNDARIES



$$\mathbf{x} = \{x_1, x_2\} \rightarrow \mathbf{z} = \{x_1^2, \sqrt{2}x_1x_2, x_2^2\}$$

Let's hang on to the logic of the previous example, namely:

- remap the feature vectors x_i into a higher-dimensional space K'
- create a linear decision boundary in K'
- back out the nonlinear decision boundary in K from the result

This linear decision boundary will be mapped to a nonlinear decision boundary in the original feature space.

The inner product is an operation that takes two vectors and returns a real number.
We we can rewrite the optimization problem in terms of the inner product and *we don’t actually have to do any calculations* in the feature space K.

We can replace this with a generalization of the inner product called a **kernel function** that maps two vectors in a higher-dimensional feature space *K’* into $\mathbb{R}$

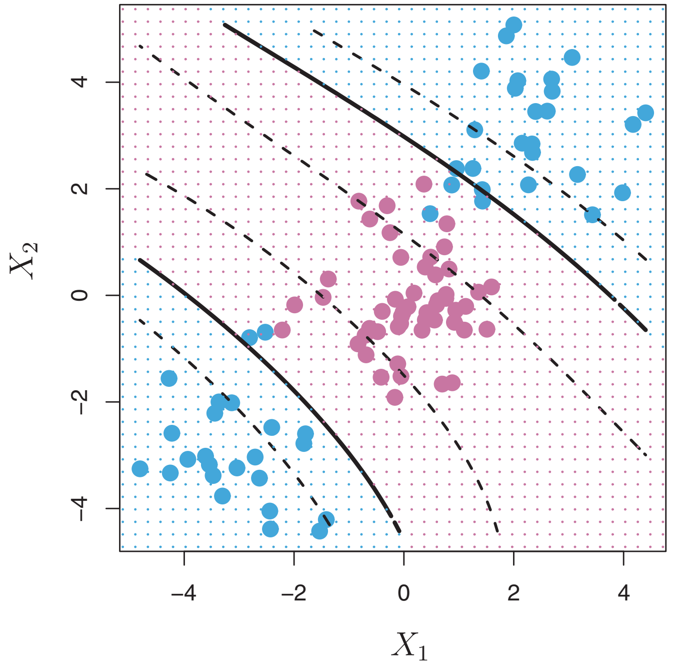
Formally, we can think of the inner product as a map that sends two vectors in the feature space K into the real line $\mathbb{R}$.

We can replace this with a generalization of the inner product called a **kernel function** that maps two vectors in a higher-dimensional feature space K' into $\mathbb{R}$.

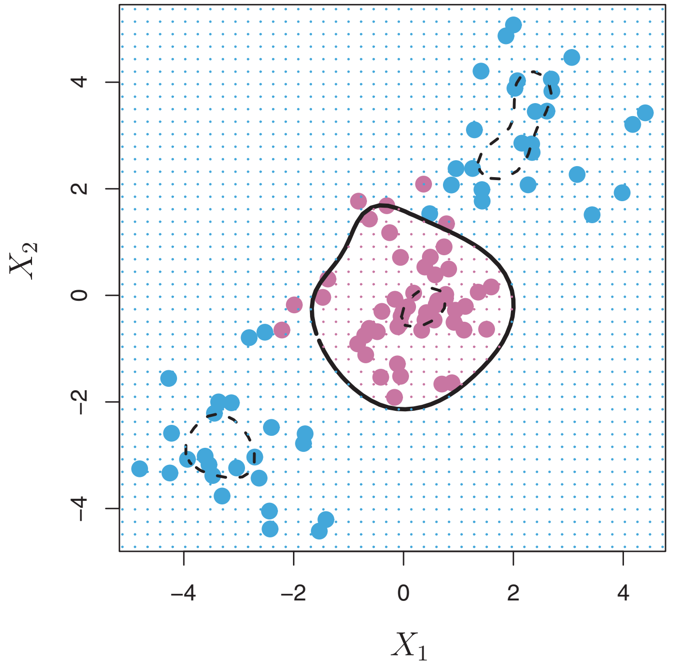
some popular kernels:

linear kernel	$k(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x}, \mathbf{x}' \rangle$
polynomial kernel	$k(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^\top \mathbf{x}' + 1)^d$
Gaussian kernel	$k(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \lvert\lvert \mathbf{x} - \mathbf{x}' \rvert\rvert^2)$

KERNEL SVM

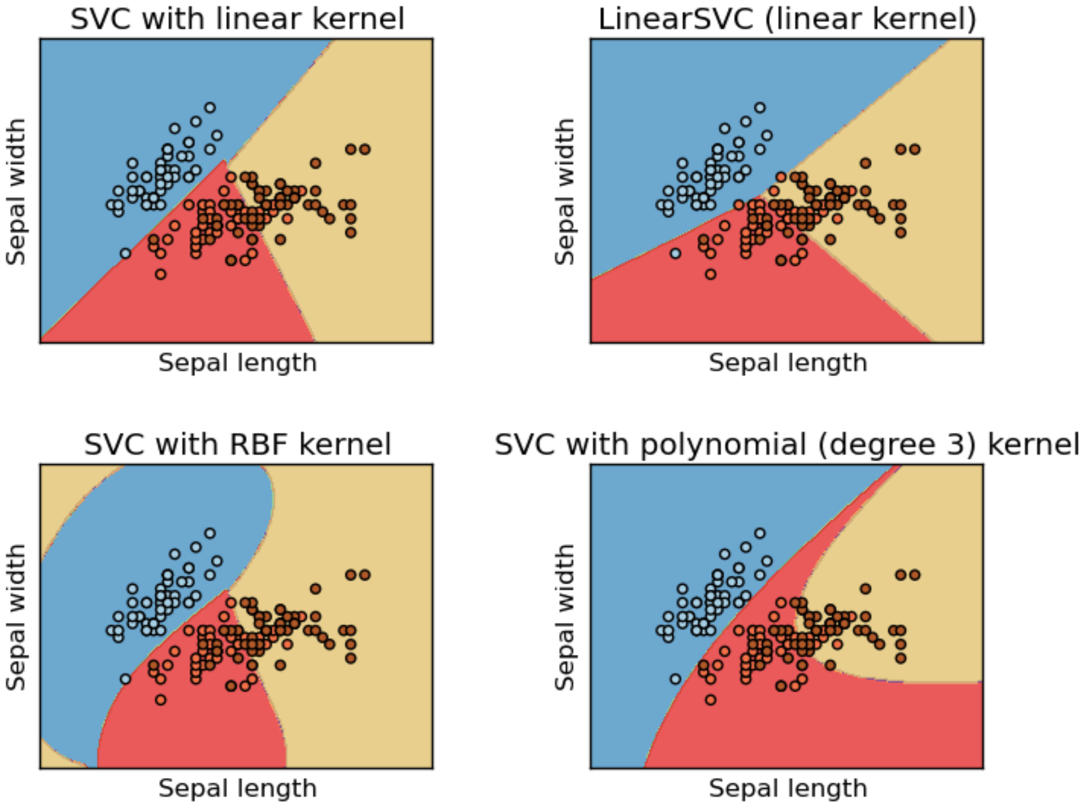


Polynomial kernel



RBF kernel

KERNEL SVM


SVC with linear kernel
LinearSVC (linear kernel)
SVC with RBF kernel
SVC with polynomial (degree 3) kernel
Sepal width
Sepal length

SVM STRENGTHS & WEAKNESSES

SVMs (and **kernel methods** in general) are versatile, powerful, and popular techniques that can produce accurate results for a wide array of classification problems.

The main disadvantage of SVMs is the lack of intuition they produce. These models are truly black boxes!

LAB: SVM IN SCIKIT-LEARN