Maximal Margin & Support Vector Classifiers.

W. Wang[1]

[1] Department of Mathematics
University of Houston

MATH 4323

Support Vector Machines.

Support vector machine (SVM) - classification approach developed in the computer science community back in the 1990s, which has been shown to be one of the best "standard" classifiers.

Mathematics associated with SVM can be difficult, but the basic concepts are understandable.

Support Vector Machines.

SVMs can be adapted for use with nearly any type of learning task, including both classification and numeric prediction.

Applications of SVMs include, but are not limited to

- Face detection (whether part of image is a face or not),
- Text categorization (classify documents into different categories),
- Image Classification (Is this a cat? Dog? Elephant?)
- Bioinformatics (e.g. cancer type classification based on genomic data)
- Handwriting recognition (recognize handwritten characters used widely)

Support Vector Machines.

Support vector machine (SVM) has been considered one of the best "out of the box" classifiers. Specifically, we will learn

- **maximal margin** classifier
- **support vector** classifier
- **support vector** machine

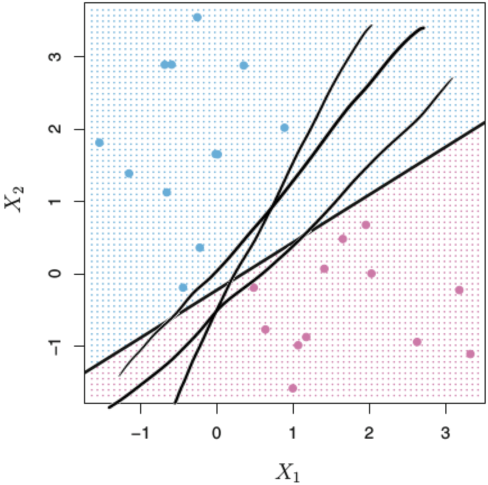
People often loosely refer to these approaches as "support vector machines."

Separable Hyperplanes

- Imagine a situation where you have a two-class classification problem with two predictors X_1 and X_2.
- Suppose that the two classes are “linearly separable” (i.e. one can draw a straight line in which all points on one side belong to the first class and points on the other side to the second class).
- Then a natural approach is to find the straight line that gives the biggest separation between the classes (i.e. the points are as far from the line as possible).

Maximal Margin Classifier.

Maximal Margin Classifier is based on optimal separating **hyperplane**.



What Is a Hyperplane? 2-Dimensional Space.

In a 2-dimensional space (of vectors $X = (X_1, X_2)^T \in \mathbb{R}^2$) a hyperplane is a subspace of dimension $(2 - 1) = 1$. In other words, for $2D$ case, a hyperplane is a line.

Mathematically, hyperplane for $2D$ space $X = (X_1, X_2)^T$ is defined as

$$\underline{\beta_0 + \beta_1 X_1 + \beta_2 X_2} = 0, \qquad (1) \longleftarrow \text{dimension} = 1$$

where $\beta_0, \beta_1, \beta_2$ are some fixed parameters.

That means: any such point $X = (X_1, X_2)^T$ for which equation (1) holds, belongs to that hyperplane.

Note: (1) is simply the equation of a line.

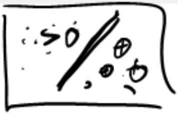
(X_1, X_2, X_3)

Question: What is a hyperplane for a $p = 3$-dimensional space?

dimension: $(p-1) = (3-1) = 2$

What Is a Hyperplane? 2-Dimensional Space.

$\beta_0 + \beta_1 X_1 + \beta_2 X_2 \neq 0$



Now, suppose that X does not satisfy (1), but

- $\beta_0 + \beta_1 X_1 + \beta_2 X_2 > 0, \implies X = (X_1, X_2)^T$ lies on one side of the hyperplane defined by (1).
- or $\beta_0 + \beta_1 X_1 + \beta_2 X_2 < 0, \implies X = (X_1, X_2)^T$ lies on the other side of the hyperplane defined by (1).

Hence, hyperplane defined by equation (1) literally **divides/separates** the 2D space into two halves.

One can easily determine on which side of the hyperplane a point lies by simply calculating the sign of the left hand side of equation (1).

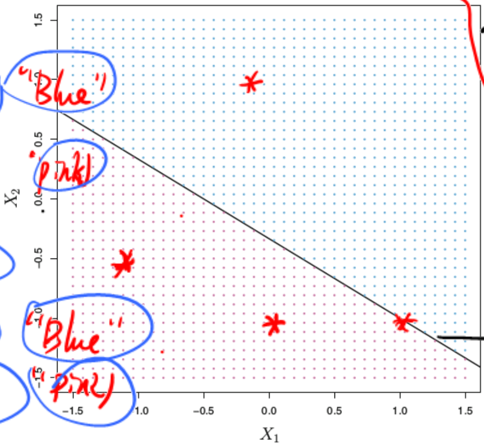
Example. Next slide shows the case of $\beta_0 = 1, \beta_1 = 2, \beta_2 = 3$.

What Is a Hyperplane? 2-Dimensional Space.

$1+2X_1+3X_2$

LHS

	LHS	
(0,1)	>0	"Blue"
(0,−1)	<0	"pink"
(1,−1)	=0	
(1,1)	>0	"Blue"
(−1,−0.5)	<0	"pink"



">0" "Blue"

"<0" "pink"

$\rightarrow \beta_0+\beta_1X_1+\beta_2X_2 = 0$

$\boxed{1+2X_1+3X_2} = 0$

X_1	X_2
0	−1
1	−1

FIGURE 9.1. *The hyperplane $1+2X_1+3X_2=0$ is shown. The blue region is the set of points for which $1+2X_1+3X_2>0$, and the purple region is the set of points for which $1+2X_1+3X_2<0$.*

What Is a Hyperplane? *p*-Dimensional Space.

$(X_1\ X_2 \cdot - - - X_p)$

All of the above can be extended to a general *p*-dimensional space:

In *p*-dimensional space, **hyperplane** is a subspace of dimension $p-1$.

Equation of a hyperplane in *p*-dimensional setting:

$$\longrightarrow \qquad \beta_0 + \beta_1 \underline{X_1} + \beta_2 \underline{X_2} + \cdots + \beta_p \underline{X_p} = 0 \qquad (2)$$

in the sense that if a point $X = (X_1, X_2, \ldots, X_p)^T \in \mathbb{R}^p$ satisfies (2), then X lies on the hyperplane.

Hyperplane (2) separates the *p*-dimensional space into two halves:

- $\underline{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p > 0}$, and
- $\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p < 0$

Classification Using a Separating Hyperplane.

Suppose that we have a

- $n \times p$ data matrix X, that consists of n training observations in p-dimensional space:

$$x_1 = (x_{11}, \ldots, x_{1p})^\intercal \equiv \begin{pmatrix} x_{11} \\ \vdots \\ x_{1p} \end{pmatrix}, \ \ldots, \ x_n = (x_{n1}, \ldots, x_{np})^\intercal$$

- each of these observations falls into one of two classes $\{-1, 1\}$:

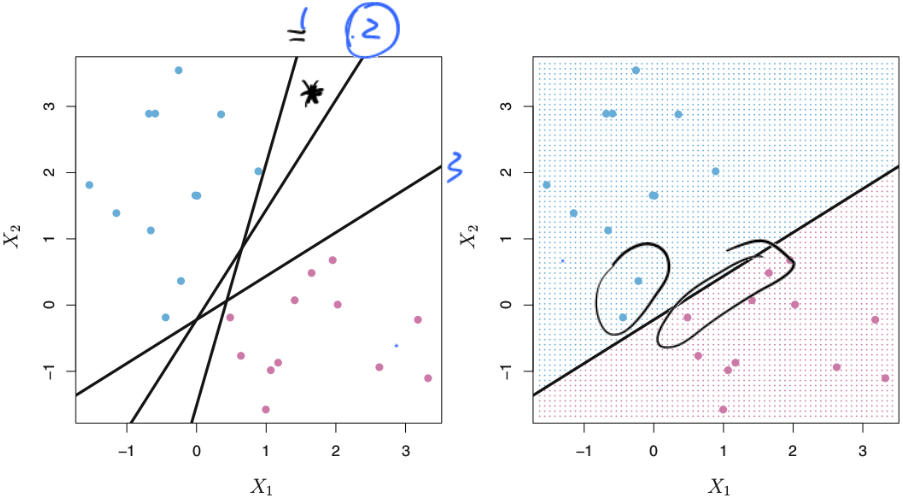
$$y_1, \ldots, y_n \in \{-1, 1\}$$

- Moreover, we also have a test observation $x^* = (x_1^*, \ldots, x_p^*)^\intercal$
- **Goal**: develop a classifier based on the training data that will correctly classify the test observation using its feature values x^*.

We have already covered a K-Nearest Neighbor approach for such task, now we will show an approach based on separating hyperplane.

Classification Using a Separating Hyperplane.

Suppose that it is possible to construct a hyperplane that separates the training observations perfectly according to their class labels.



Classification Using a Separating Hyperplane.

We can label the observations from the

- blue class as $\underline{y_i = 1}$,
- purple class as $\underline{y_i = -1}$.

not 0, vs 1.
but 1 vs −1

Then a separating hyperplane has the property that, $\forall i = 1, \ldots, n$:

- $\underline{\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}} > 0$ if $y_i = \underline{1}$,
- $\underline{\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}} < 0$ if $y_i = -1$,

Equivalently, a separating hyperplane has the property that

(−1) (< 0) > 0

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) > 0, \ i = 1, \ldots, n$$

1 (> 0) > 0

If a separating hyperplane exists, a natural classifier is: a test observation is assigned to a class depending on which side of the hyperplane it is located. E.g. see next slide.

$$y_i(\beta_0+\beta_1 x_{i1}+\cdots\cdots)>0$$

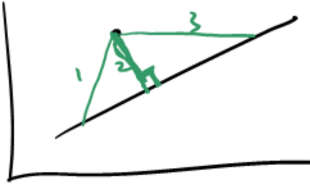
$$\Longleftrightarrow \frac{|\beta_0+\beta_1 x_{i1}+\cdots+\beta_p x_{ip}|}{\sqrt{\beta_1^2+\beta_2^2+\cdots+\beta_p^2}}$$

$(x_1 - x_p)$

dist. from a point in p dimensional space to a line $\beta_0+\beta_1 x_1+\cdots+\beta_p x_p=0$

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 = 0$$

$$(X_1, X_2)$$



$$\mathrm{dist}(X_0, \mathrm{line}) = \frac{|\beta_0 + \beta_1 X_1 + \beta_2 X_2|}{\sqrt{\beta_1^2 + \beta_2^2}}$$

Classification Using a Separating Hyperplane.

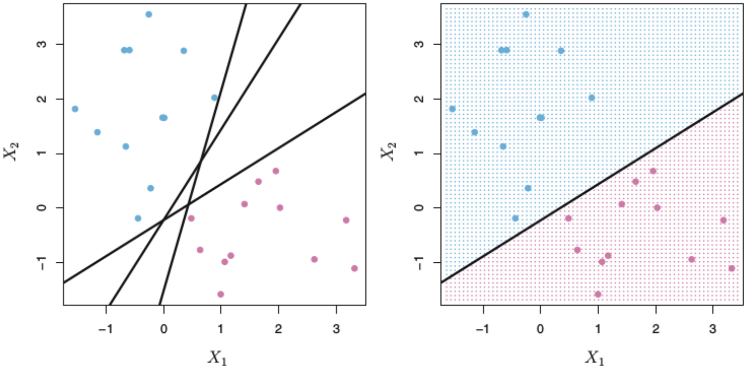


FIGURE 9.2. Left: *There are two classes of observations, shown in blue and in purple, each of which has measurements on two variables. Three separating hyperplanes, out of many possible, are shown in black.* Right: *A separating hyperplane is shown in black. The blue and purple grid indicates the decision rule made by a classifier based on this separating hyperplane: a test observation that falls in the blue portion of the grid will be assigned to the blue class, and a test observation that falls into the purple portion of the grid will be assigned to the purple class.*

Maximal Margin Classifier.

Issue: If our data can be perfectly separated using a hyperplane, then there will in fact exist an infinite number of such hyperplanes.

Hence, to construct a classifier we need to decide which separating hyperplane to use.

A natural choice is the **maximal margin** hyperplane - separating hyperplane that is **farthest** from the training observations.

Next slide demonstrates

- how the distance from observations to hyperplane is measured (perpendicular lines), and
- what is meant by **margin** (distance from hyperplane to the closest training observation).

Task: Compare the margin of the separating hyperplane on the next slide, with the margin of three separating hyperplanes on the left figure on previous slide.

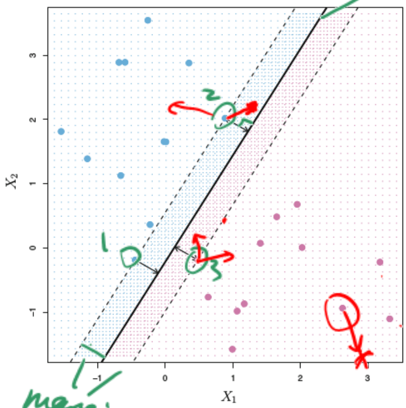
Maximal Margin Classifier.

no obs. between the margin.

seperating hyperplane.

$\hat{\beta}_0$ $\hat{\beta}_1$

1, 2, 3 support vectors



margin.

FIGURE 9.3. *There are two classes of observations, shown in blue and in purple. The maximal margin hyperplane is shown as a solid line. The margin is the distance from the solid line to either of the dashed lines. The two blue points and the purple point that lie on the dashed lines are the support vectors, and the distance from those points to the margin is indicated by arrows. The purple and blue grid indicates the decision rule made by a classifier based on this separating hyperplane.*

Maximal Margin Classifier.

On previous slide, we see that three training observations are equidistant from the maximal margin hyperplane - these are **support vectors**, since

1. They are **vectors** in *p*-dimensional space. E.g. for our case: $p = 2$, and $X = (X_1, X_2) \in \mathbb{R}^2$.
2. They **support** the maximal margin hyperplane, because they are closest to the hyperplane $\Longrightarrow$ they dictate, or "support", its location.

Construction of the Maximal Margin Classifier.

Presume we have

- a set of n training observations $x_1, \ldots, x_n \in \mathbb{R}^p$, and
- associated class labels $y_1, \ldots, y_n \in \{-1, 1\}$.

Maximal margin hyperplane is the solution to the optimization problem

$$\underset{\beta_0,\beta_1,\ldots,\beta_p}{\text{maximize}}\ \boxed{M} \tag{3}$$

subject to

$$\boxed{\sum_{j=1}^{p} \beta_j^2 = 1,} \quad \text{and} \longleftarrow \tag{4}$$

$$\boxed{y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip})} \geq M \quad \forall i = 1, \ldots, n \tag{5}$$

$1 + 2x_1 + 3x_2 = 0$

$2 + 4x_1 + 6x_2 = 0$

$\frac{1}{\sqrt{13}} + \frac{2x_1}{\sqrt{13}} + \frac{3x_2}{\sqrt{13}} = 0$

$+$

$$\frac{|\beta_0+\beta_1X_{i1}+\cdots+\beta_pX_{ip}|}{\sqrt{\beta_1^2+\cdots\beta_p^2}} \geq \underline{\underline{M}}.$$

LHS → dist (a training obs, seperating hyper).

Construction of the Maximal Margin Classifier.

Breaking down the optimization problem (3) – (5):

3. *M* corresponds to the margin to be maximized.

   It should be maximized with respect to parameters β_0, β_1,. . . , β_p of the separating hyperplane:

   $$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} > 0.$$

   So the part (3) of our optimization problem:

   $$\underset{\beta_0,\beta_1,\ldots,\beta_p}{\text{maximize}}\ M$$

   can be read as

   "Find such a hyperplane $\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} = 0$ that the margin *M* is maximized."

Construction of the Maximal Margin Classifier.

4. (4) just provides context for (5), because with this constraint the perpendicular distance from observation $x_i = (x_{i1}, \ldots, x_{ip})^\intercal$ to the hyperplane is given by

$$y_i(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip})$$

In algebraic notation:

$$\sum_{j=1}^{p} \beta_j^2 = 1 \implies \text{dist}(x_i, \text{hyperplane}) \equiv \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}$$

Proof: Given

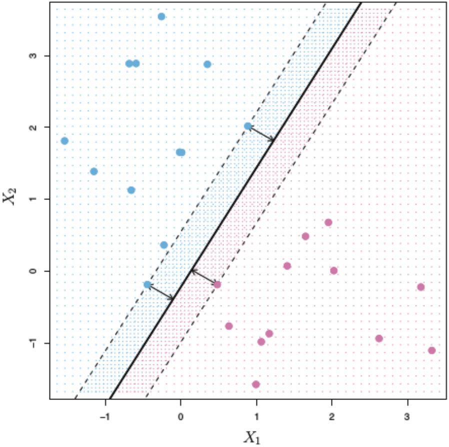
- Line with equation: $\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p = 0$;
- Point x_i with coordinates: $(x_{i1}, \ldots, x_{ip})^T$;

By distance formula: $\text{dist}(x_i, \text{line}) = \frac{\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}}{\sqrt{\beta_0^2 + \beta_1^2 + \cdots + \beta_p^2}}$

$\sum_{j=1}^{p} \beta_j^2 = 1 \implies \text{dist}(x_i, \text{line}) = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}$.

Construction of the Maximal Margin Classifier.

5. (5) guarantees that each observation is
   - on the correct side of the hyperplane, and
   - is removed by at least $M > 0$ ($\equiv$ max. margin) from the hyperplane.



Construction of the Maximal Margin Classifier.

To recap,

- *M* represents the margin of our hyperplane $\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p = 0$.
- Constraints (4) and (5) ensure that each observation is
  - on the correct side of the hyperplane, and
  - at least a distance *M* from the hyperplane.
- Part (3) of the optimization problem chooses $\beta_0, \beta_1, \ldots, \beta_p$ to maximize *M*.

This is exactly the definition of the maximal margin hyperplane!

The problem (3) $-$ (5) can be solved efficiently, but details of this optimization are outside of the scope of this book.

Issue #1: Non-Separable Case.

Issue #1: Maximal margin classifier is a natural way to perform classification, if a separating hyperplane exists.

However, in many cases there's no separating hyperplane $\Longrightarrow$ no maximal margin classifier. In particular, optimization problem (3) $-$ (5) has no solution with $M > 0$.

See example on next slide.

Issue #1: Non-Separable Case.

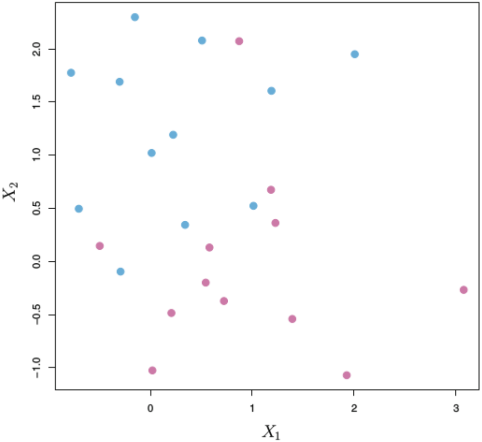


FIGURE 9.4. *There are two classes of observations, shown in blue and in purple. In this case, the two classes are not separable by a hyperplane, and so the maximal margin classifier cannot be used.*

Issue #2: Overfitting.

Issue #2: But even when a separating hyperplane **does** exist, sometimes it might **not** desirable: it will perfectly classify all of the training observations, which potentially leads to

Sensitivity to individual observations,

$$\Longrightarrow$$

Overfitting.

Example. On the next slide, see how dramatically the maximal marginal hyperplane changes after just one new observation (which might be noise) is introduced.

Moreover, the resulting margin is **tiny**.

Issue #2: Overfitting.

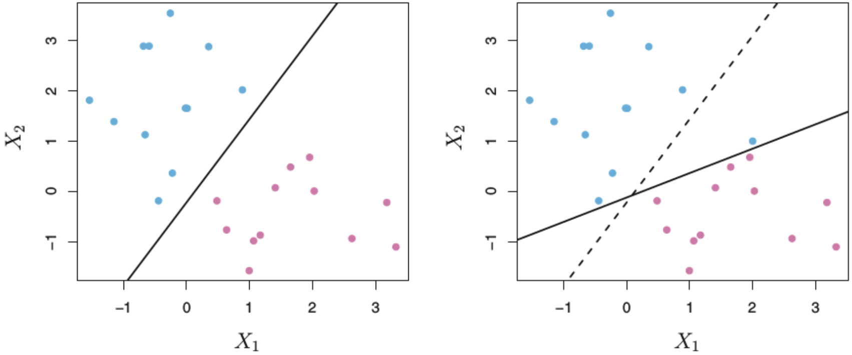


FIGURE 9.5. Left: *Two classes of observations are shown in blue and in purple, along with the maximal margin hyperplane.* Right: *An additional blue observation has been added, leading to a dramatic shift in the maximal margin hyperplane shown as a solid line. The dashed line indicates the maximal margin hyperplane that was obtained in the absence of this additional point.*

Solution: Support Vector Classifiers.

We might want a classifier based on a hyperplane that does **not** perfectly separate the two classes, in the interest of

- Greater robustness to individual observations (less variance), and
- Better classification of most of the training observations.

Meaning: we could sacrifice misclassifying a few training observations, in order to better classify the rest of training observations.

Support vector classifier, does exactly this:

it allows some observations to be

- on the incorrect side of the **margin** (be within margins, rather than outside - see **left figure on next slide**), or
- even on the incorrect side of the **hyperplane** ($\Longrightarrow$ some training observations are misclassified, which is inevitable in a non-linearly separable case - see **right figure on next slide**)

Support Vector Classifiers.

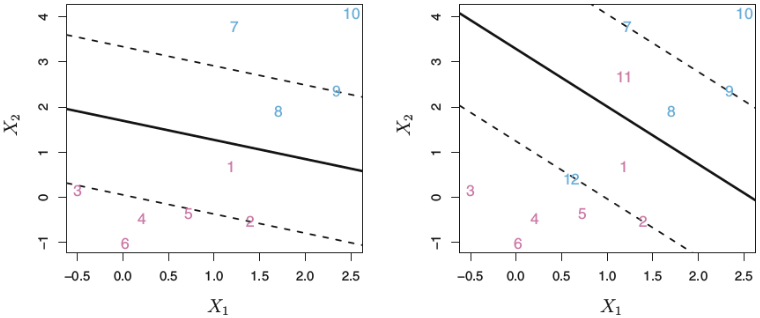


FIGURE 9.6. Left: *A support vector classifier was fit to a small data set. The hyperplane is shown as a solid line and the margins are shown as dashed lines.* Purple observations: *Observations* 3, 4, 5, *and* 6 *are on the correct side of the margin, observation* 2 *is on the margin, and observation 1 is on the wrong side of the margin.* Blue observations: *Observations* 7 *and* 10 *are on the correct side of the margin, observation* 9 *is on the margin, and observation* 8 *is on the wrong side of the margin. No observations are on the wrong side of the hyperplane.* Right: *Same as left panel with two additional points,* 11 *and* 12. *These two observations are on the wrong side of the hyperplane and the wrong side of the margin.*

Support Vector Classifier $\equiv$ "Soft" Margin Classifier.

Support vector classifier, for the aforementioned reasons, is sometimes called a **soft** margin classifier.

"Soft" margin means that it **can be violated** by some training observations.

Example. From the images on the previous slide:

- **Left**: which observations violate the margin?
- **Right**: which observations violate the margin?
- Taking it a step further, which observations (on the left and right images, respectively) violate the **hyperplane**?

Constructing Support Vector Classifier.

To find the coefficients $\beta_0, \beta_1, \ldots, \beta_p$ of hyperplane equation for the support vector classifier

$$\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p = 0$$

we need to solve the following optimization task

$$\underset{\beta_0,\beta_1,\ldots,\beta_p,\epsilon_1,\ldots,\epsilon_n}{\text{maximize}} \quad M \tag{6}$$

subject to

$$\sum_{j=1}^{p} \beta_j^2 = 1 \tag{7}$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) \geq M(1 - \epsilon_i), \tag{8}$$

$$\epsilon_i \geq 0, \; \sum_{i=1}^{n} \epsilon_i \leq C \tag{9}$$

Support Vector Classifier: Slack Variables.

As before,

- M is the margin to be maximized.
- (7) gives context to (8), having it be about distances of observations from separating hyperplane.

Among the newcomers, we have

- $\epsilon_1, \dots, \epsilon_n$, which are **slack** variables. They allow individual observations to be on the wrong side of the margin or the hyperplane. In particular,
  - if $\epsilon_i = 0$, then observation i is on the correct side of the margin;
  - if $0 < \epsilon_i \leq 1$, then observation i is on the wrong side of the margin, but on the correct side of the hyperplane;
  - if $\epsilon > 1$, then observation i is on the wrong side of both margin & hyperplane ($\Leftrightarrow$ it is misclassified).

Question. Back on slide 25, for both **left** and **right** figures: Which observations have $\epsilon_i = 0$? $\epsilon_i \in (0, 1]$? $\epsilon_i > 1$?

Support Vector Classifier: Tuning Parameter C.

- C in (9) is a nonnegative tuning parameter that determines the number and severity of the violations to the margin (& hyperplane) that we will tolerate.

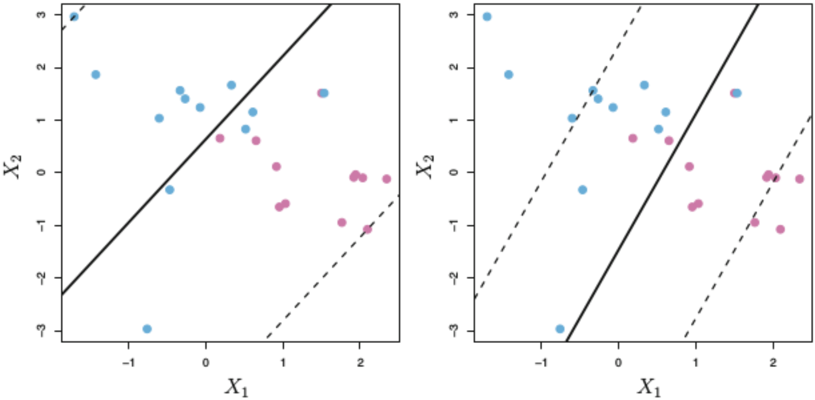
  We can think of C as a "budget" for margin violations by n observations:

  - $C = 0 \implies \epsilon_1 = \cdots = \epsilon_n$, no budget for violation. All observations should be on correct side of the margin $\Leftrightarrow$ maximal margin classifier (only exists when classes are **linearly separable**).
  - $C > 0 \implies$ violations are allowed, but no more than C observations can be on the wrong side of the hyperplane (as $\implies \epsilon_i = 1$, and (9) requires $\sum_i \epsilon_i \leq C$).
  - As $C \Uparrow$, we are more tolerant of violations $\implies$ margin widens.
  - As $C \Downarrow$, we are less tolerant of violations $\implies$ margin narrows.

Support Vector Classifier: Tuning Parameter C.

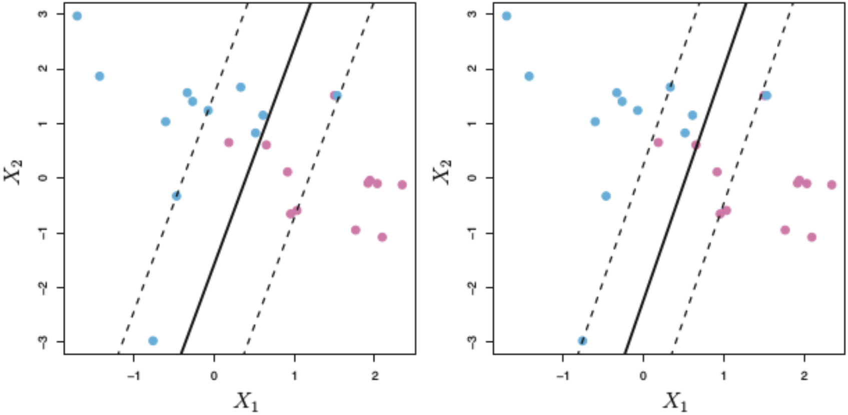
Below we show examples of resulting support vector classifiers on the same data set for

1. **Left**: large $C \implies$ high tolerance of violations, wide margin.
2. **Right**: moderate $C \implies$ moderate tolerance, smaller margin.



Support Vector Classifier: Tuning Parameter C.

- **Left**: Smaller $C \implies$ low tolerance of violations, small margin.
- **Right**: Tiny $C \implies$ nearly no tolerance, tiny margin .



Selecting C Value: Cross-Validation.

Natural question: How do we set the value of tuning parameter C?

It is similar to selection of K in K-Nearest Neighbor models: we can use **cross-validation** to come up with optimal value of C. You could

1. Try a grid of values for C, e.g.

$$\{0.001, 0.01, 0.1, 0.5, 1, 5, 10, 100\}$$

2. For each value C on that grid, calculate the test error via cross-validation.
3. Pick the value C yielding the **lowest** test error.

Support Vectors. Similarity to KNN.

Support vector classifiers are only affected by observations that lie

- directly on the margin, or
- on the wrong side of the margin for their class,

and those are known as **support vectors**.

Hence, large $C \implies$ more observations violating the margin & defining the hyperplane (more support vectors).

Another similarity between *K* in KNN and tuning parameter *C*:

- Large value of $C \Rightarrow$ many observations determine the hyperplane
  High K in KNN $\Rightarrow$ many observations ("neighbors") determine the predictions. Much less sensitive to single observations.
- Small value of $C \Rightarrow$ few observations determine the hyperplane
  Low K in KNN $\Rightarrow$ few observations ("neighbors") determine predictions. Much more sensitive to single observations.