perceptron

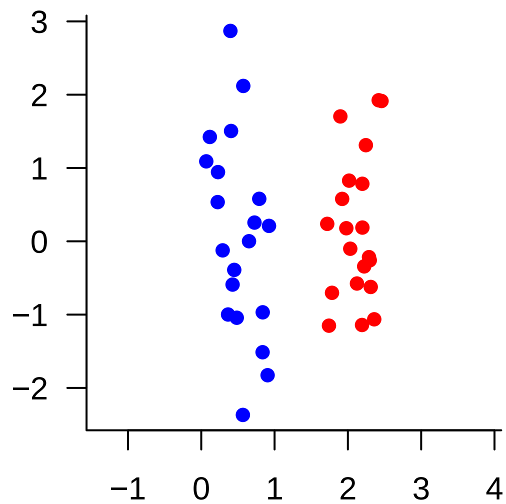
Joyce Robbins

8/13/2018

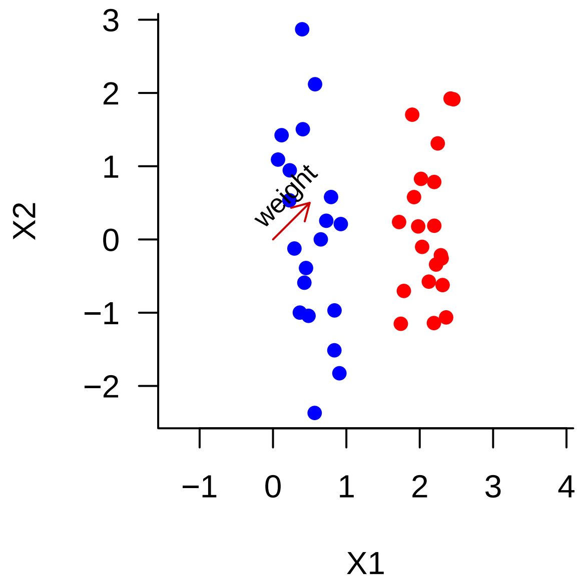
The perceptron is a simple algorithm that learns to classify inputs into two classes by adjusting the weights (w) in the equation $y_i = \text{sign}(w_i x_i)$ until all inputs in a training set are correctly classified. Here the steps of algorithm will be presented visually in two-dimensional space.

The basics

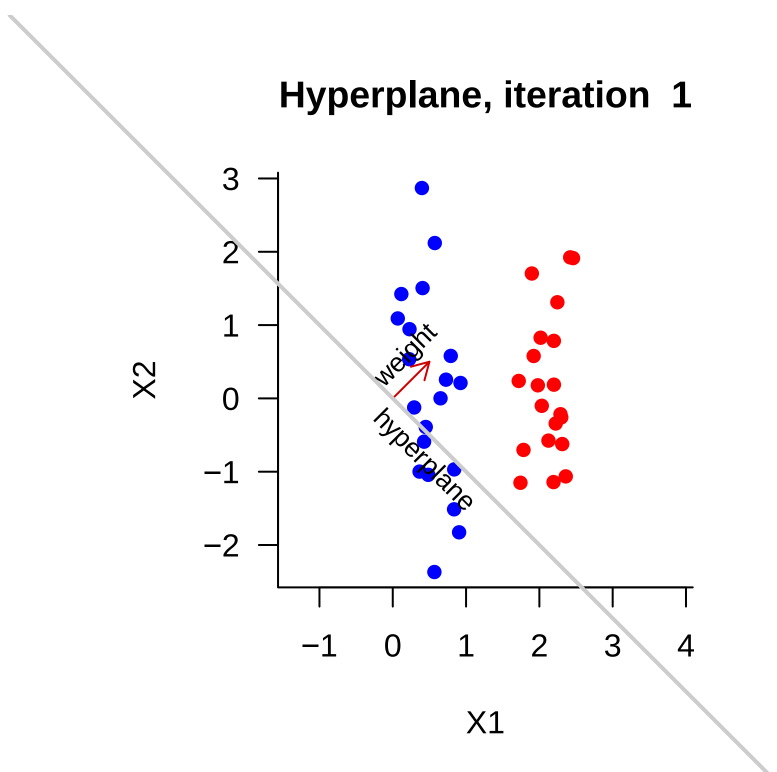
We start by plotting (x_1, x_2), coloring each point by class. Note that the points can be separated by a line; if this is not the case, the algorithm won't work.



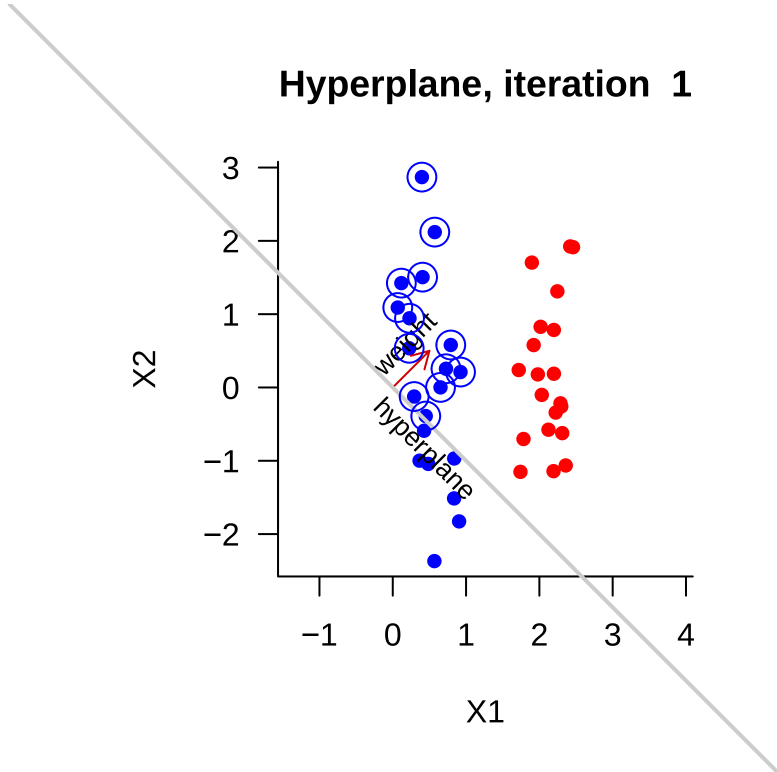
We start with an arbitrary weight vector, $(w0, w1, w2)$. Often $(0, 0, 0)$ is used, but we'll start with (0, 0.5, 0.5) so we can visualize it:



The decision boundary, or hyperplane, is the line orthogonal to the weight vector. For points on the line, the sign of $(w_i x_i)$ equals zero. On one side of the line, the sign of $(w_i x_i)$ is greater than zero whereas on the other side the sign of $(w_i x_i)$ is less than zero; hence the line serves to divide all points into two classes according to the perceptron logic.



Note the circled points – these are the misclassified points – the ones for which $y_i \neq \text{sign}(w_i x_i)$.



The Algorithm

The perceptron algorithm works by updating the weight vector based on a randomly selected misclassified point, calculating the new hyperplane, and repeating until the hyperplane separates all points into the two classes.

The formula for the new weight vector is:

$w_{t+1} = w_t + \eta y_i x_i$, where

$x_i =$ the misclassified point

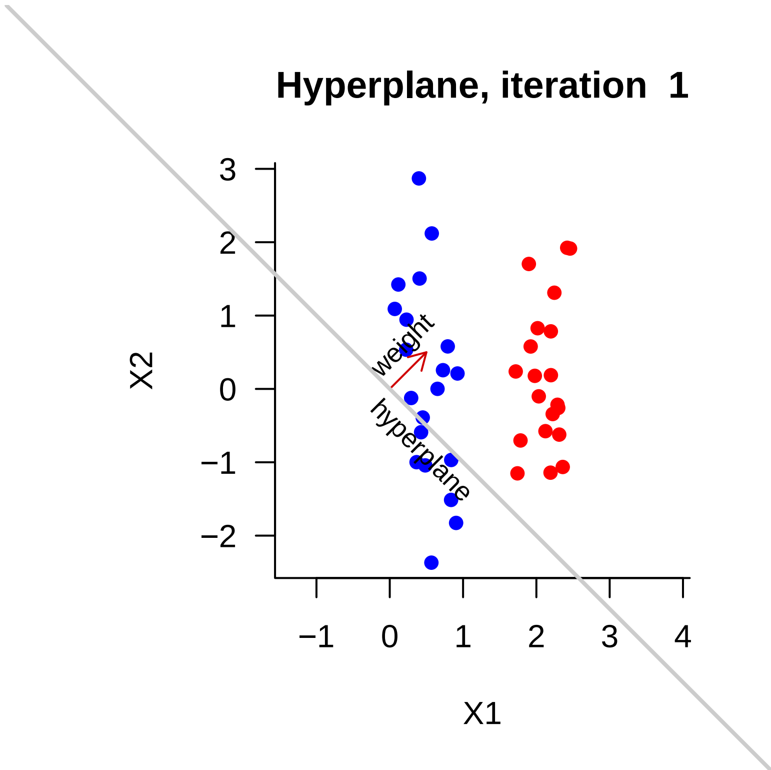
$y_i =$ the true label of the misclassified point (-1 or 1)

$\eta =$ the learning rate, which we'll set to 1 for the sake of simplicity

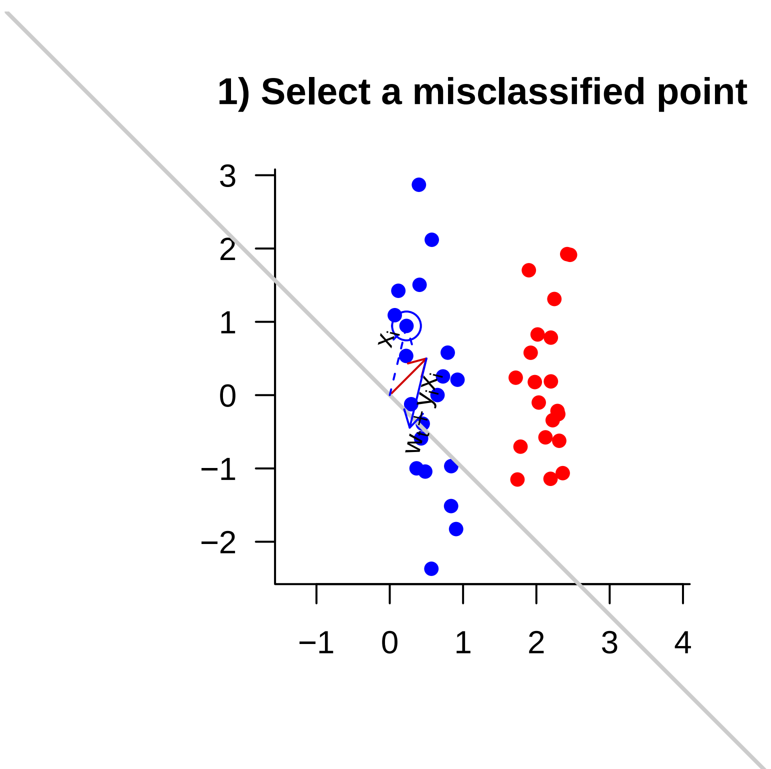
Visually, the new weight vector, w_{t+1}, is determined by adding $y_i x_i$ to w_t and then shifting by the offset $w_0/||w||_2$.

We'll go through the algorithm one step at a time.

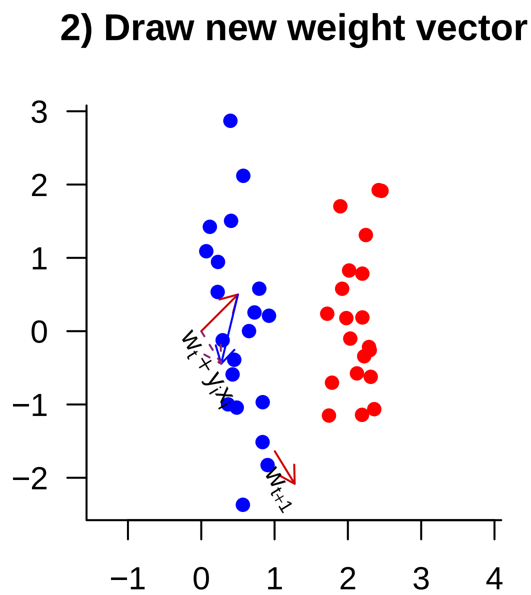
We begin with our original weight vector and hyperplane:



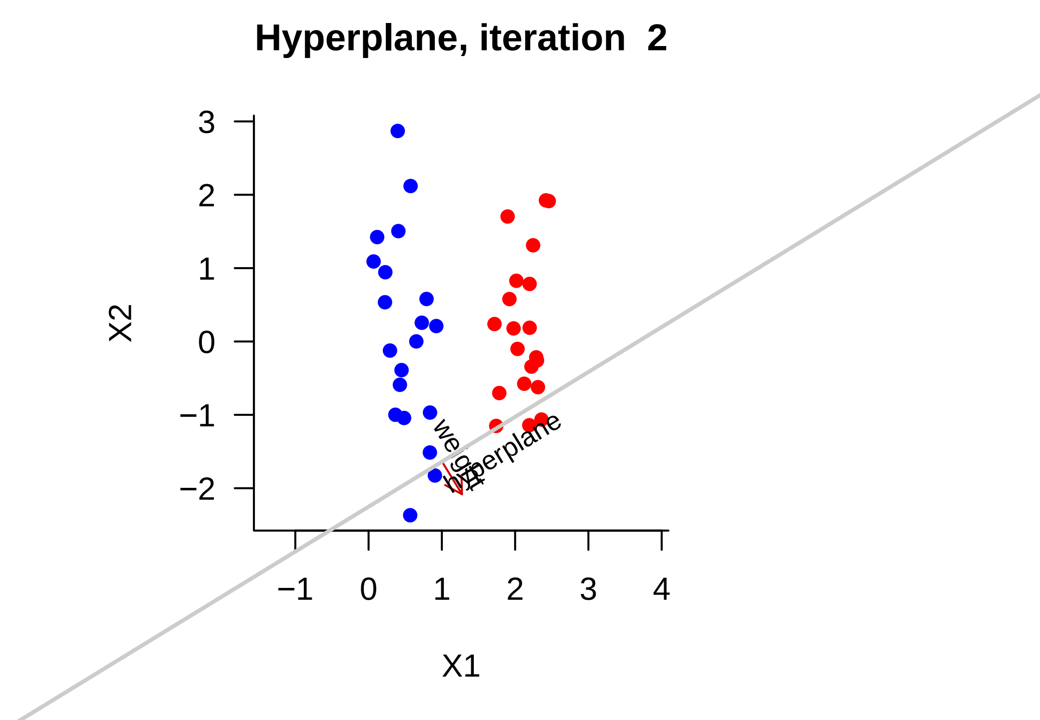
Next we randomly select a misclassified point. In the diagram below, x_i is shown as a **dashed blue arrow**, and $y_i x_i$ added to w_t as a **solid blue arrow**:



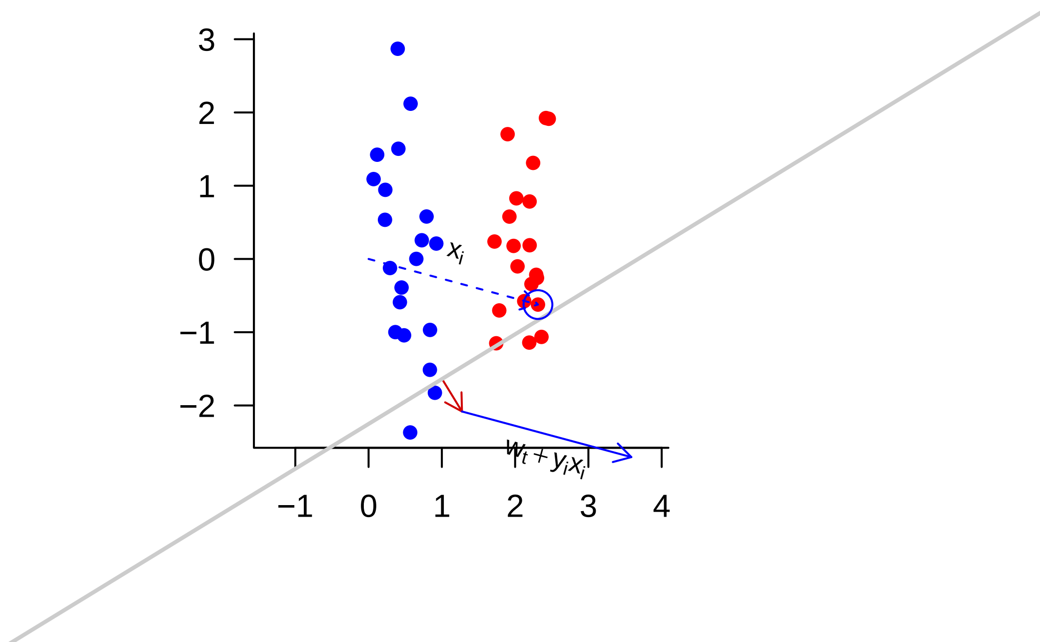
Next we determine the new weight vector by shifting the vector sum by $w_0/||w||_2$:



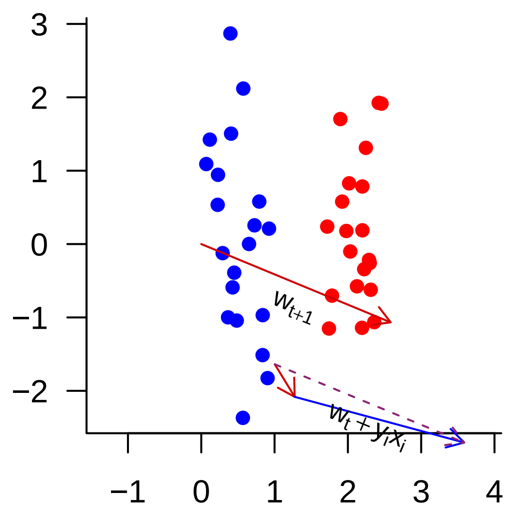
Finally, we draw the new hyperplane:

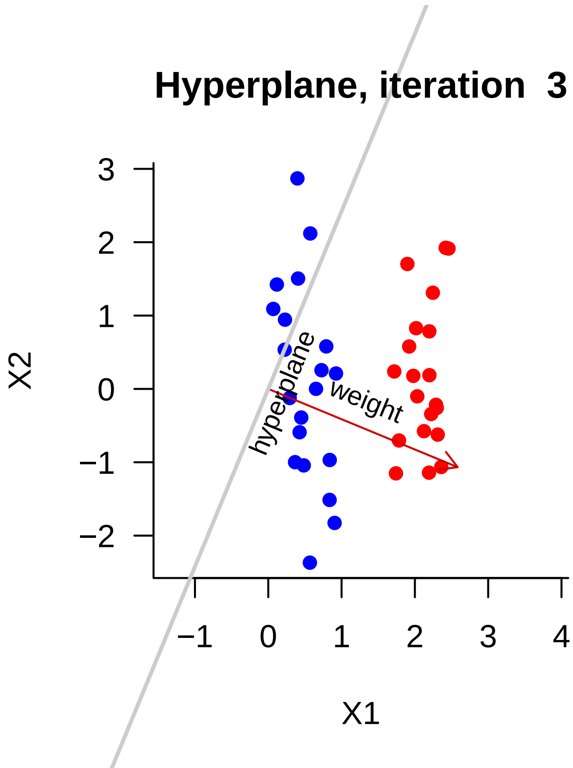


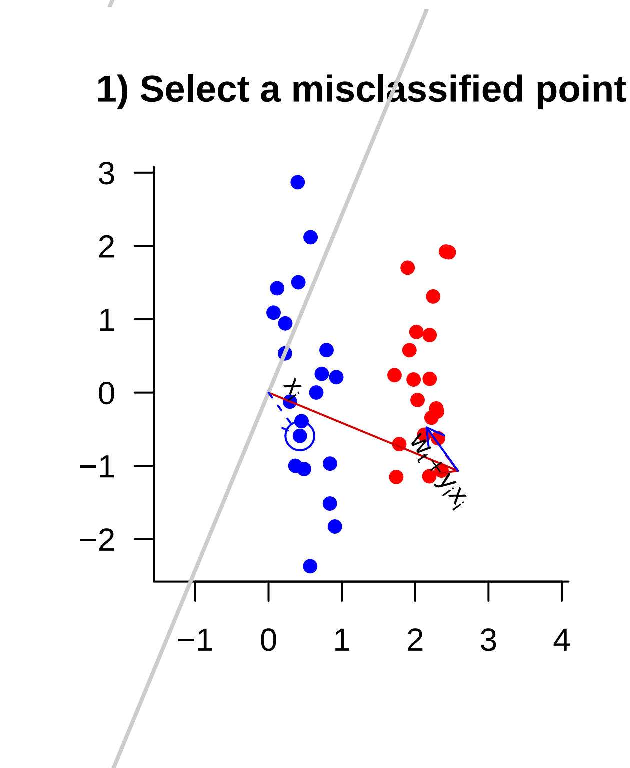
1) Select a misclassified point

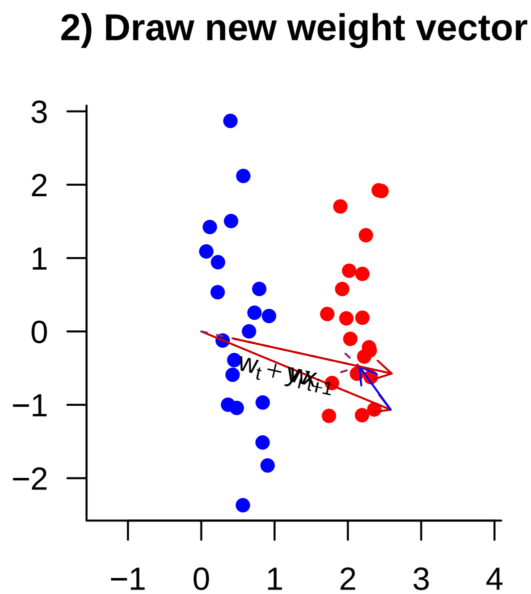


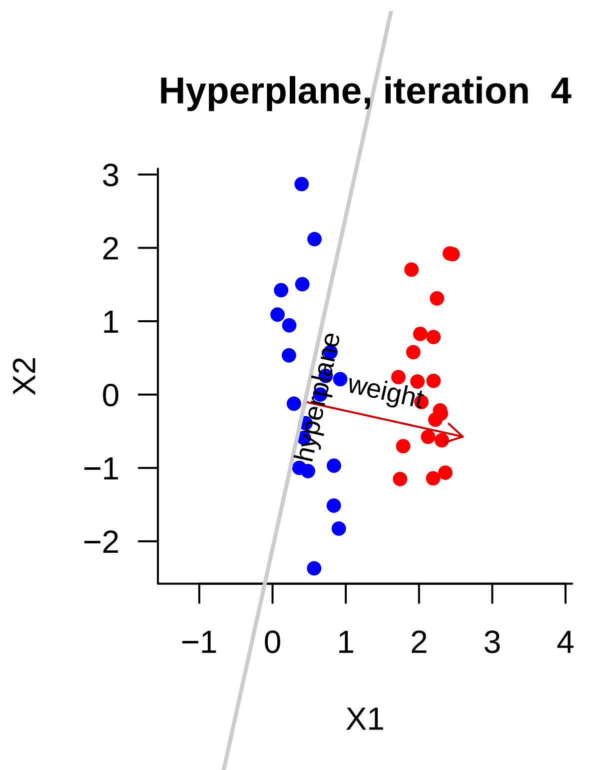
2) Draw new weight vector

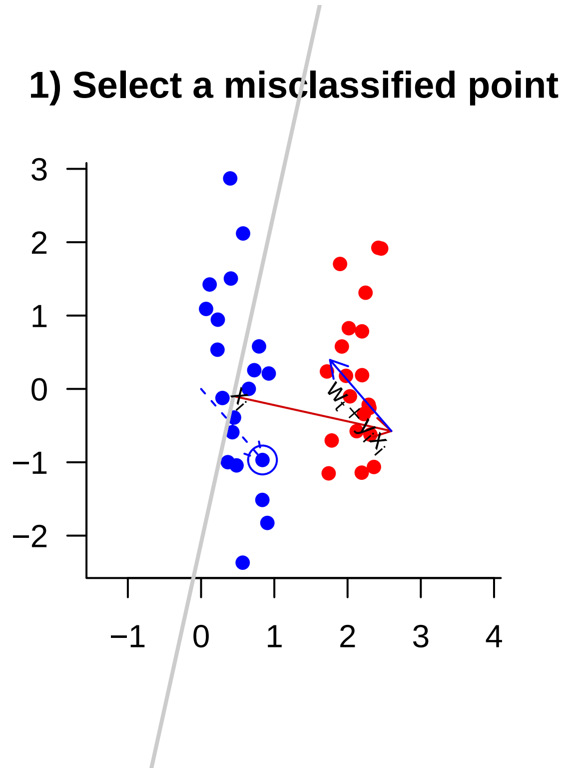


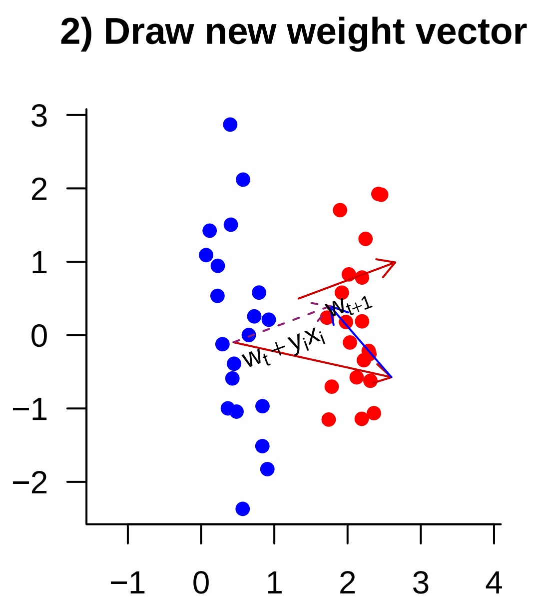

Hyperplane, iteration 3
X2
X1
3
2
1
0
−1
−2
−1
0
1
2
3
4
hyperplane
weight

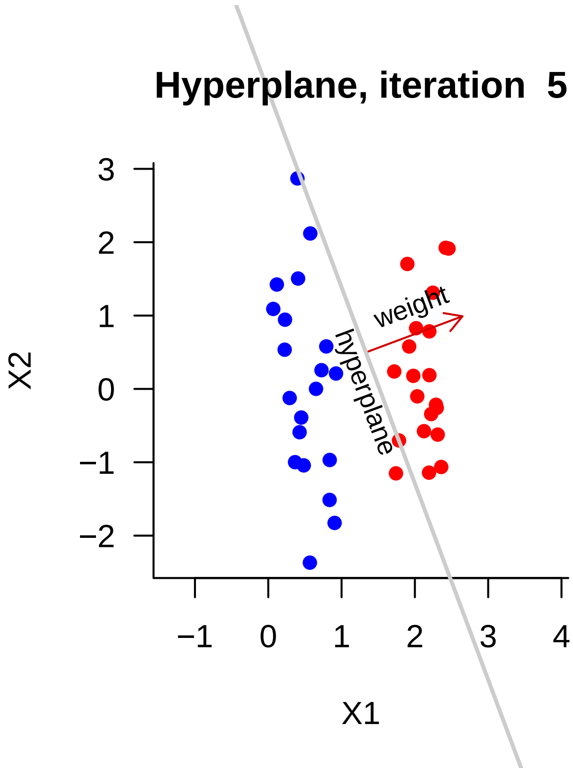

1) Select a misclassified point
3
2
1
0
−1
−2
−1
0
1
2
3
4

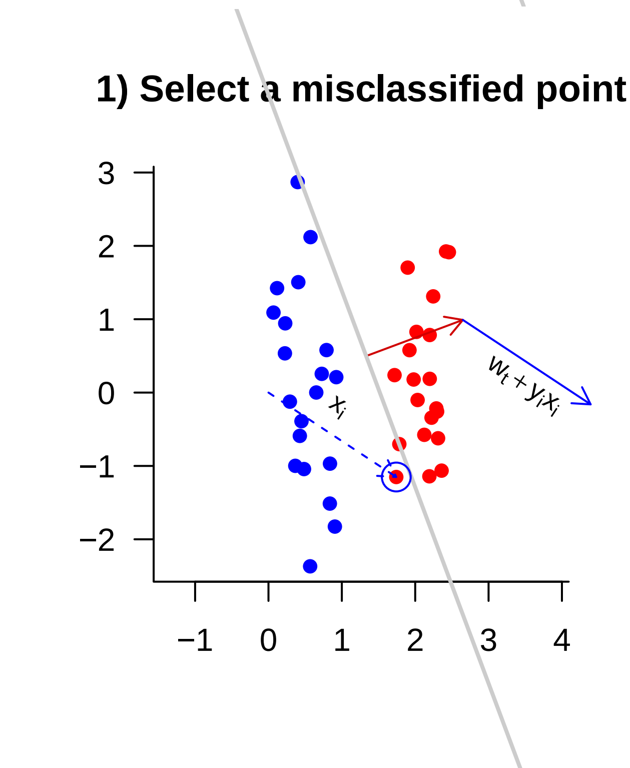

2) Draw new weight vector
3
2
1
0
−1
−2
−1
0
1
2
3
4

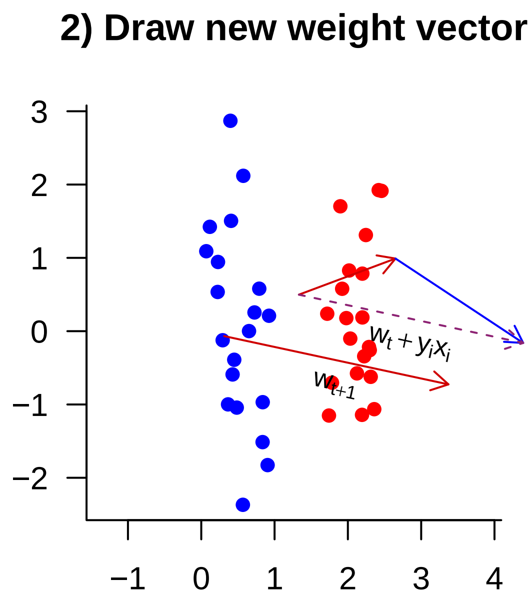

Hyperplane, iteration 4
3
2
1
0
−1
−2
−1
0
1
2
3
4
X2
X1
hyperplane
weight

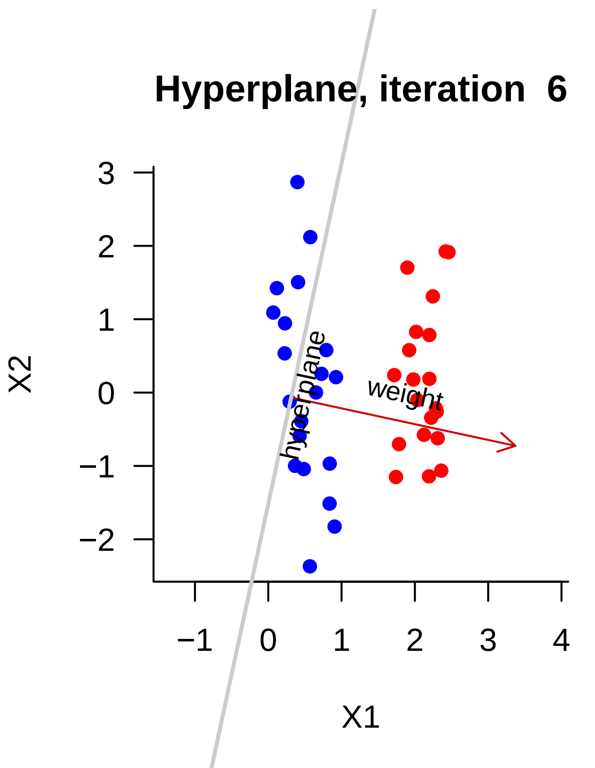

1) Select a misclassified point
3
2
1
0
−1
−2
−1
0
1
2
3
4
xi
wt + yixi


2) Draw new weight vector
3
2
1
0
−1
−2
−1
0
1
2
3
4
wt+1
wt + yixi

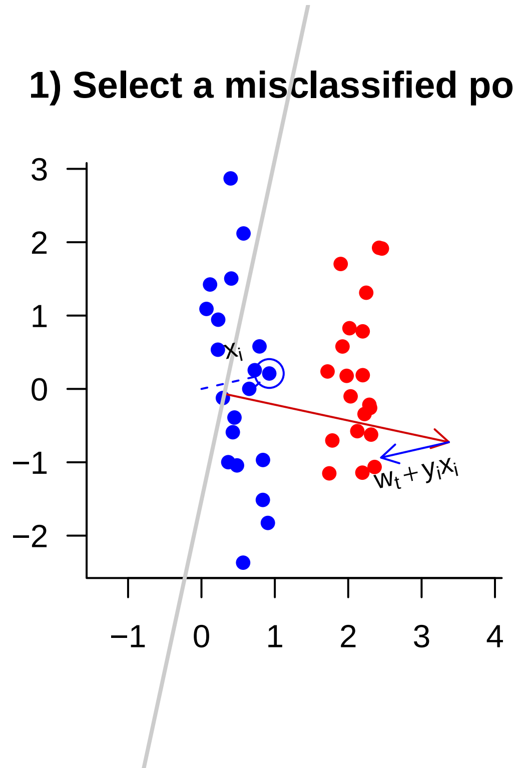

Hyperplane, iteration 5
X2
X1
weight
hyperplane
3
2
1
0
−1
−2
−1
0
1
2
3
4


1) Select a misclassified point
x_i
$w_t + y_i x_i$
3
2
1
0
−1
−2
−1
0
1
2
3
4

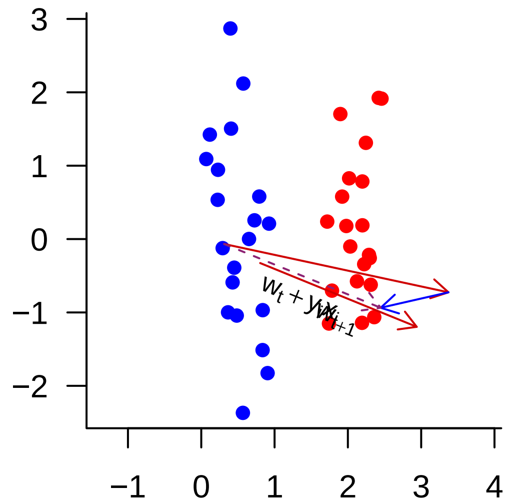

2) Draw new weight vector
3
2
1
0
−1
−2
−1
0
1
2
3
4
$w_t + y_i x_i$
w_{t+1}

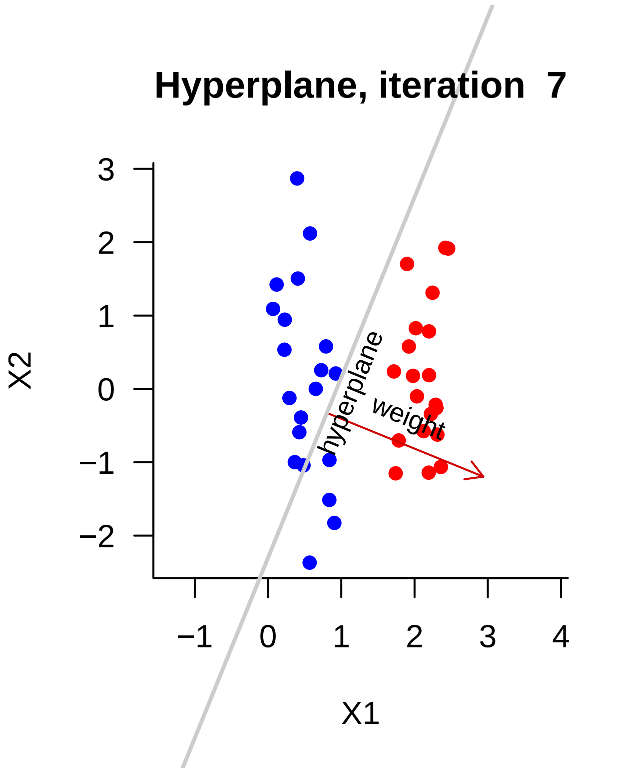

Hyperplane, iteration 6
3
2
1
0
−1
−2
X2
−1
0
1
2
3
4
X1
hyperplane
weight

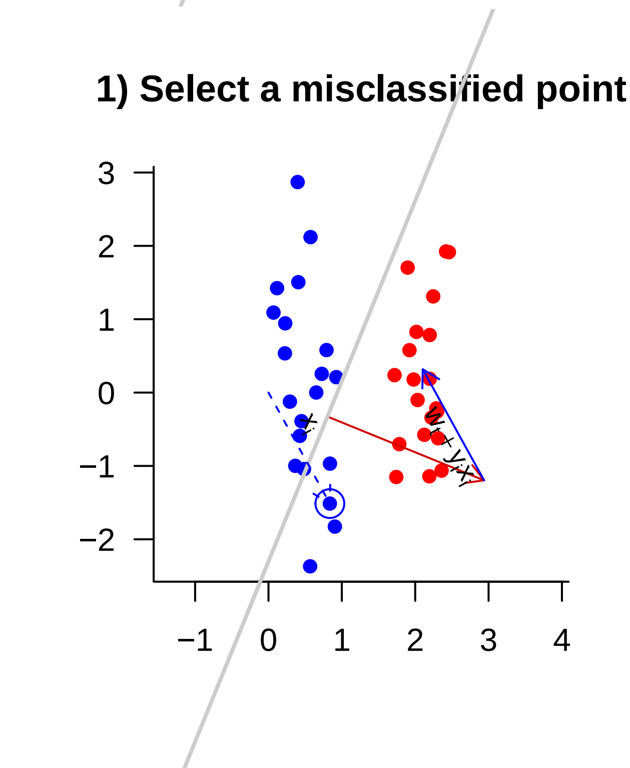
1) Select a misclassified point

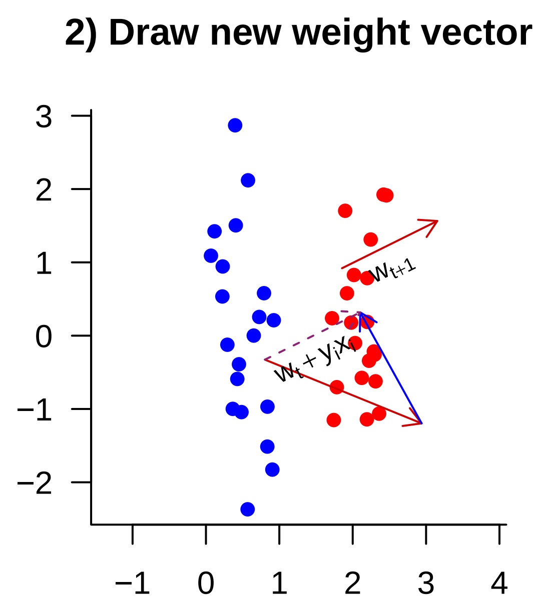


2) Draw new weight vector

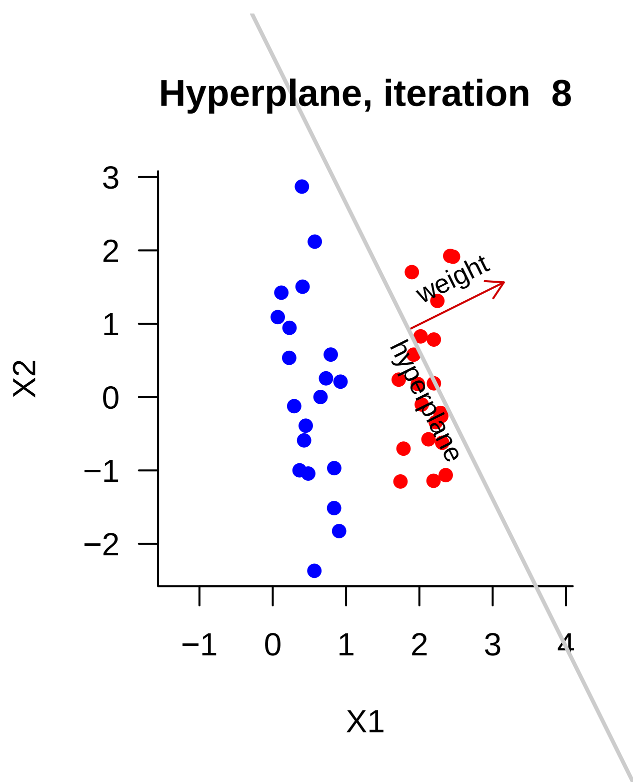



Hyperplane, iteration 7
X1
X2
hyperplane
weight
−1 0 1 2 3 4
−2 −1 0 1 2 3


1) Select a misclassified point
−1 0 1 2 3 4
−2 −1 0 1 2 3

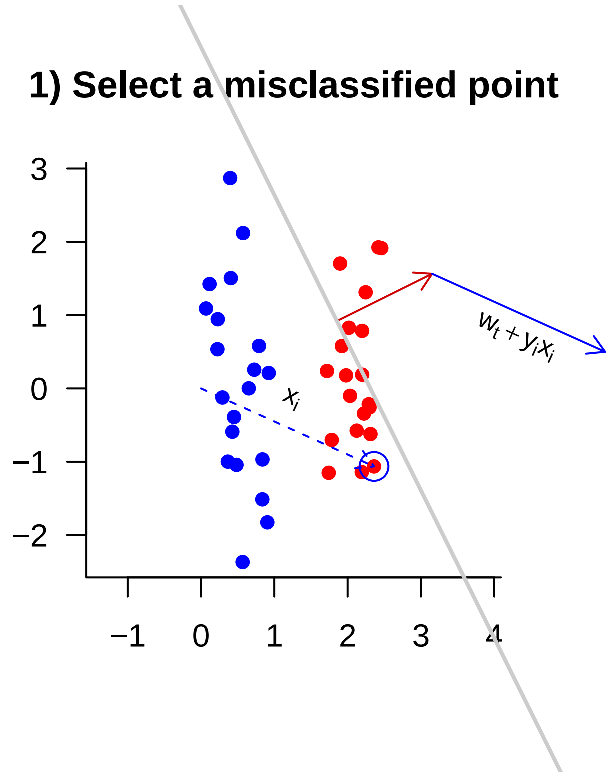


2) Draw new weight vector
w_{t+1}
$w_t + y_i x_i$
3
2
1
0
−1
−2
−1
0
1
2
3
4

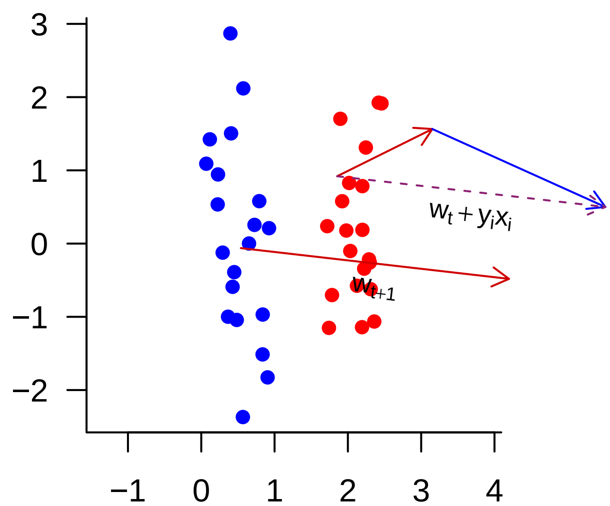


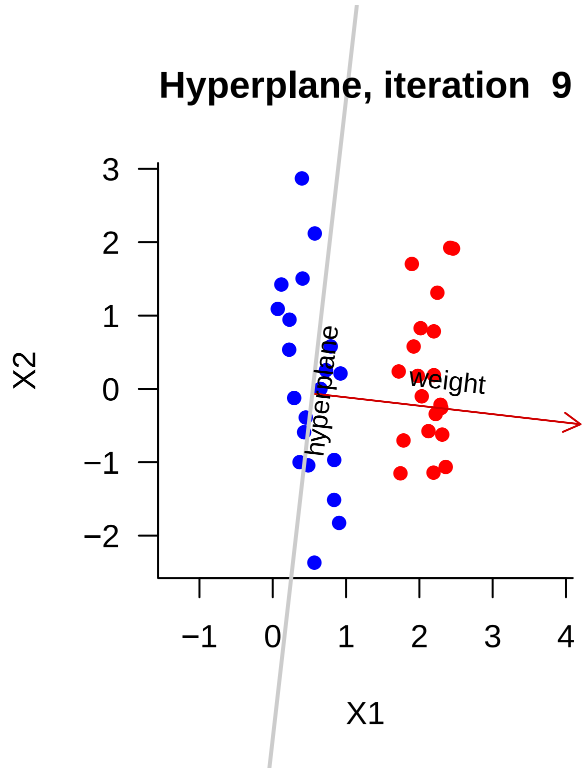
Hyperplane, iteration 8
weight
hyperplane
X2
X1
3
2
1
0
−1
−2
−1
0
1
2
3
4

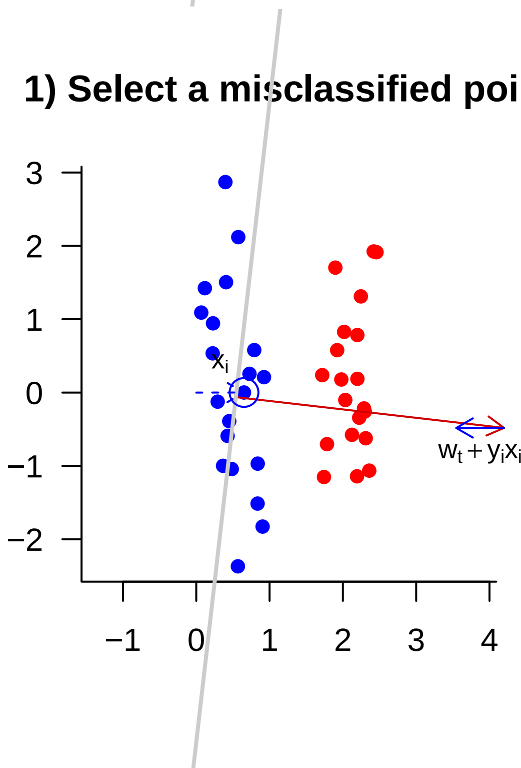
1) Select a misclassified point



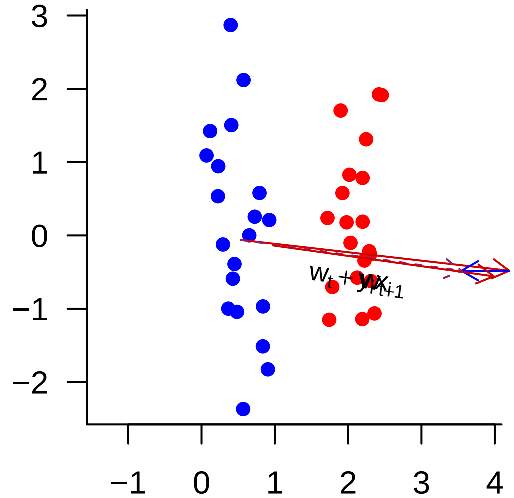
2) Draw new weight vector



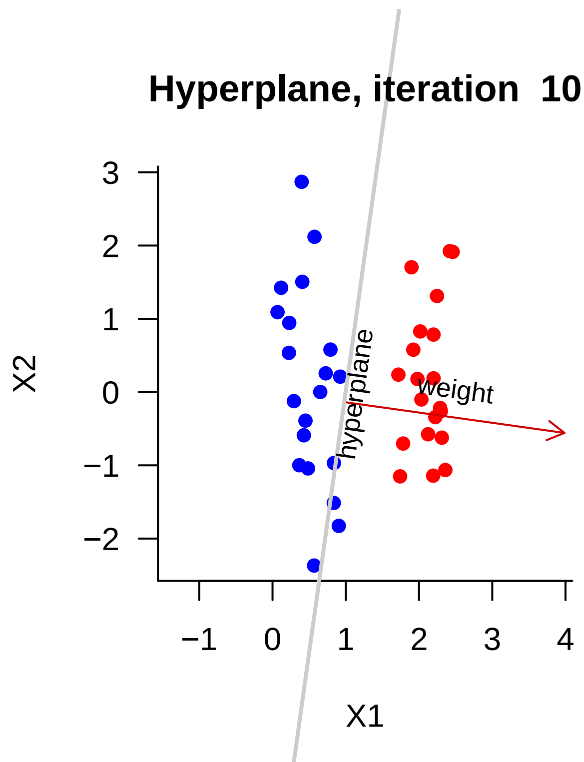

Hyperplane, iteration 9
3
2
1
0
−1
−2
X2
−1
0
1
2
3
4
X1
hyperplane
weight

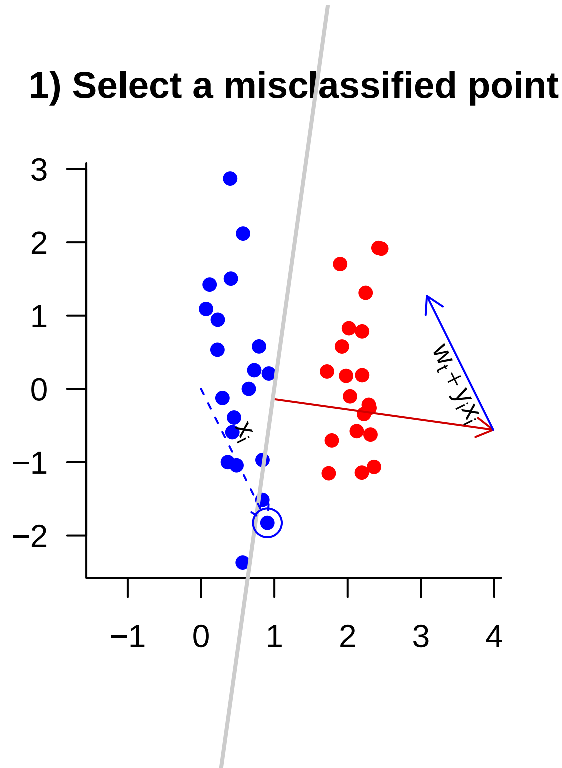

1) Select a misclassified point
3
2
1
0
−1
−2
−1
0
1
2
3
4
x_i
$w_t + y_i x_i$

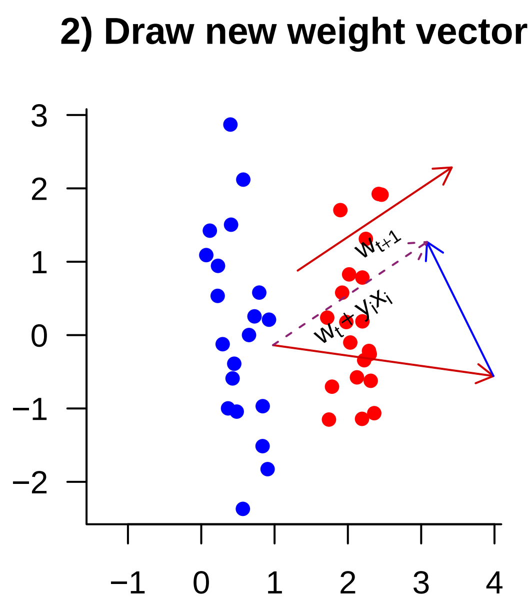
2) Draw new weight vector



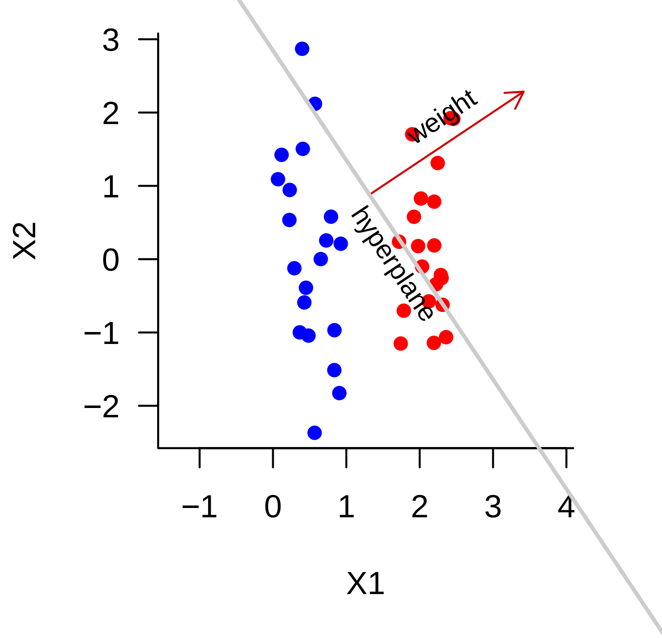
Hyperplane, iteration 10



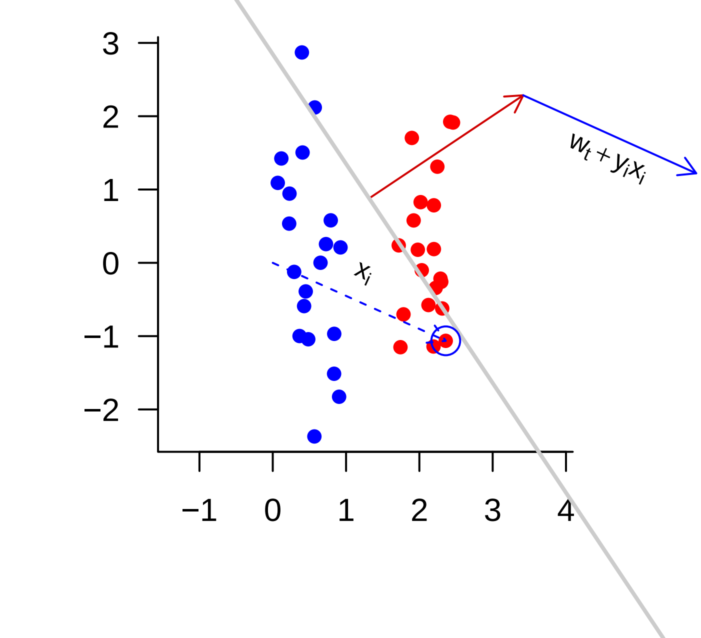

1) Select a misclassified point
3
2
1
0
−1
−2
−1
0
1
2
3
4
x_i
$w_t + y_i x_i$

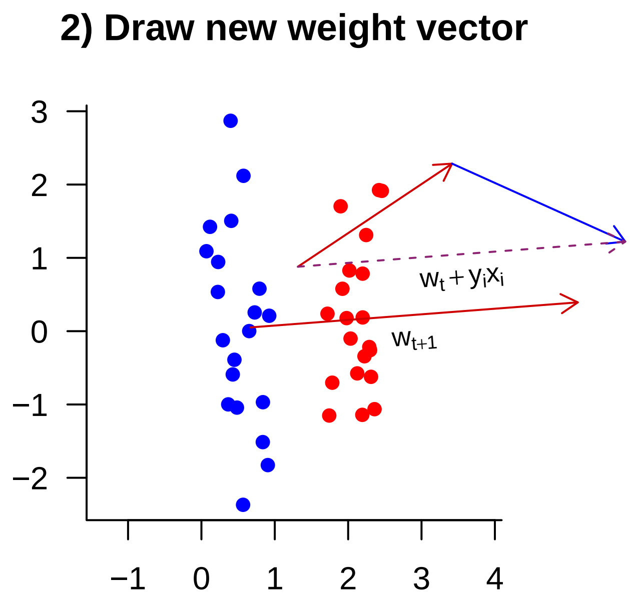

2) Draw new weight vector
3
2
1
0
−1
−2
−1
0
1
2
3
4
w_{t+1}
$w_t + y_i x_i$

Hyperplane, iteration 11

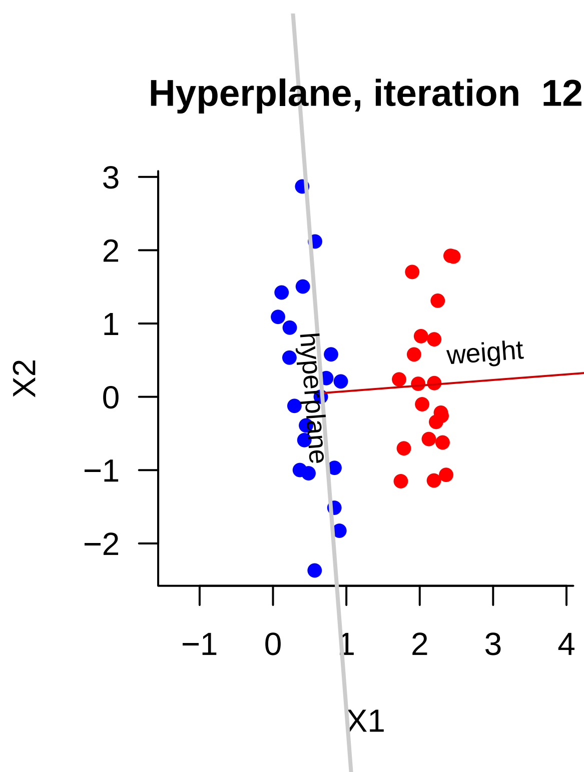


1) Select a misclassified point

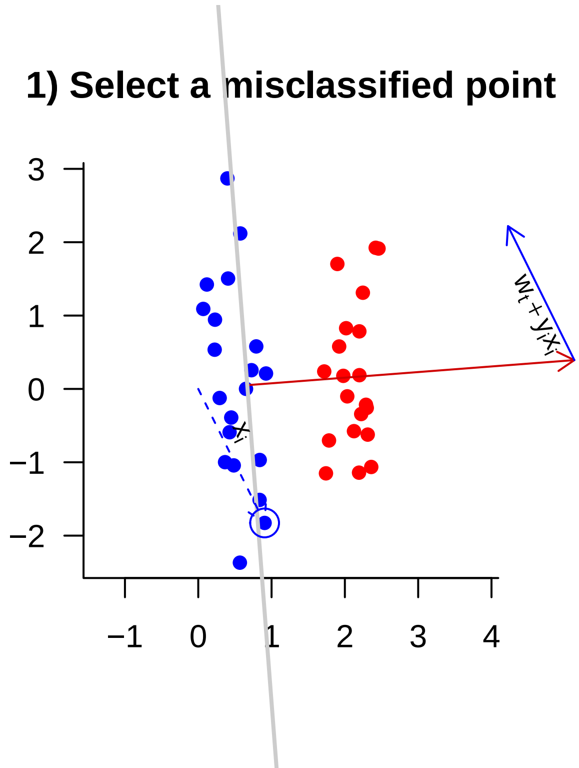


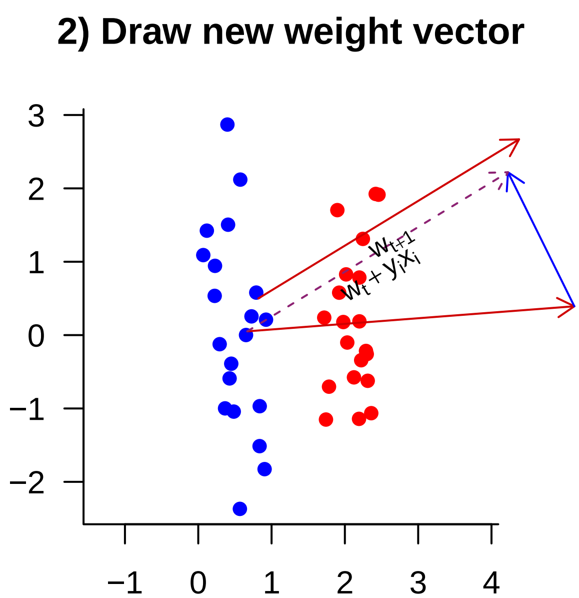


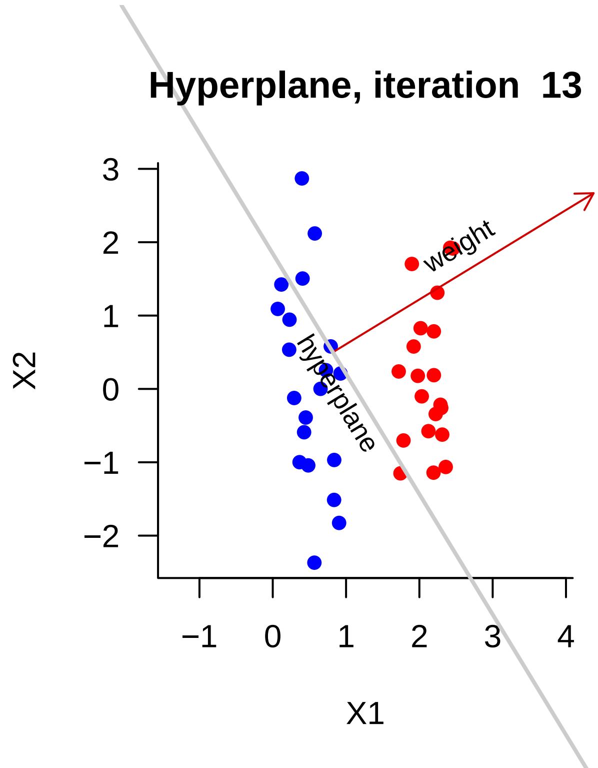
2) Draw new weight vector
3
2
1
0
−1
−2
−1
0
1
2
3
4
$w_t + y_i x_i$
w_{t+1}

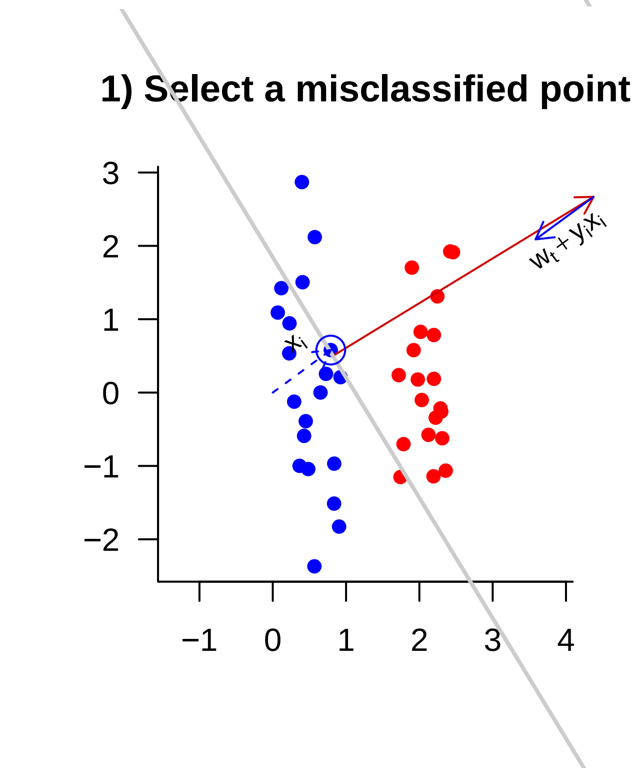


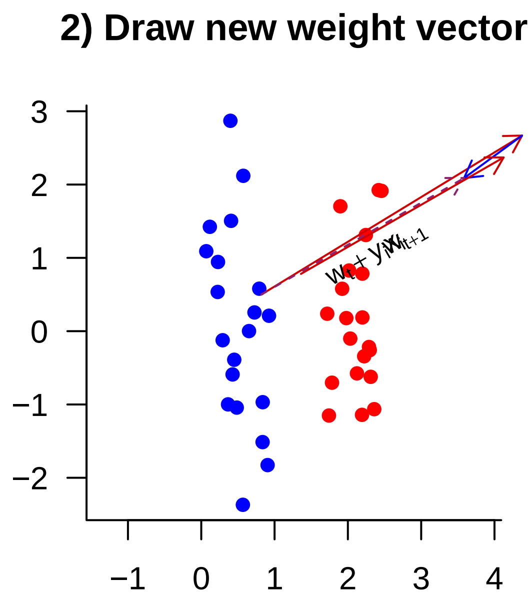
Hyperplane, iteration 12
3
2
1
0
−1
−2
−1
0
1
2
3
4
X2
X1
hyperplane
weight

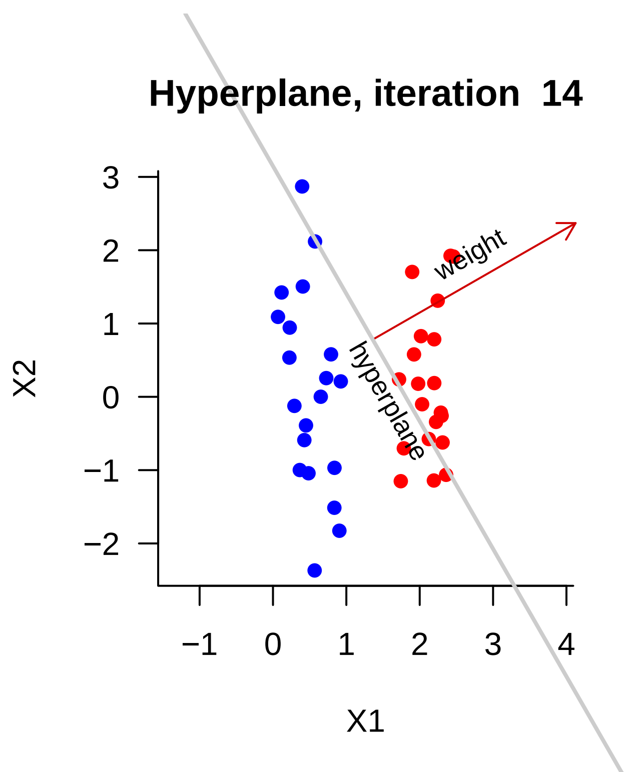

1) Select a misclassified point
3
2
1
0
−1
−2
−1
0
1
2
3
4
$w_t+y_ix_i$
x_i


2) Draw new weight vector
3
2
1
0
−1
−2
−1
0
1
2
3
4
w_{t+1}
$w_t+y_ix_i$

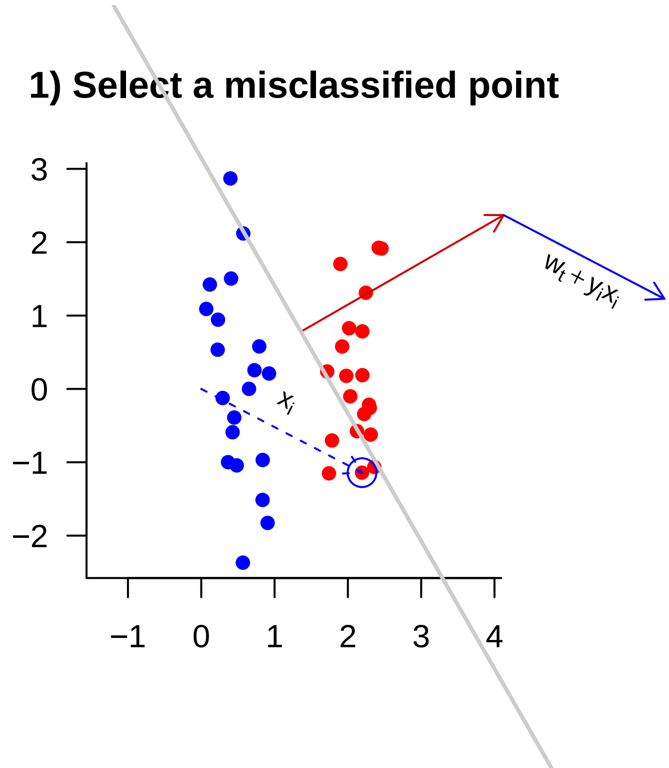

Hyperplane, iteration 13
weight
hyperplane
X1
X2


1) Select a misclassified point
x_i
$w_t + y_i x_i$

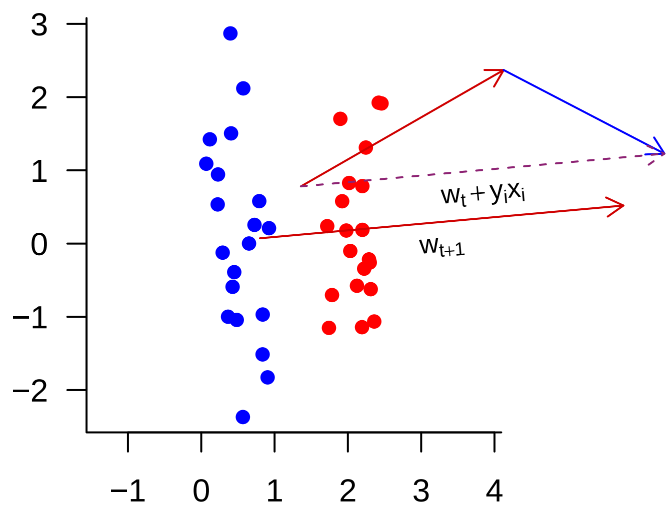

2) Draw new weight vector
wt + yixi
wt+1

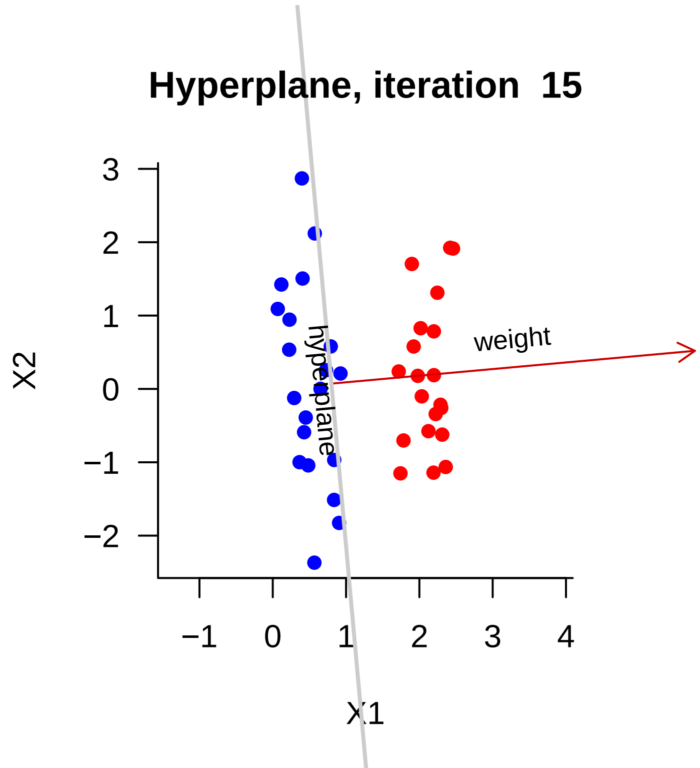

Hyperplane, iteration 14
weight
hyperplane
X1
X2

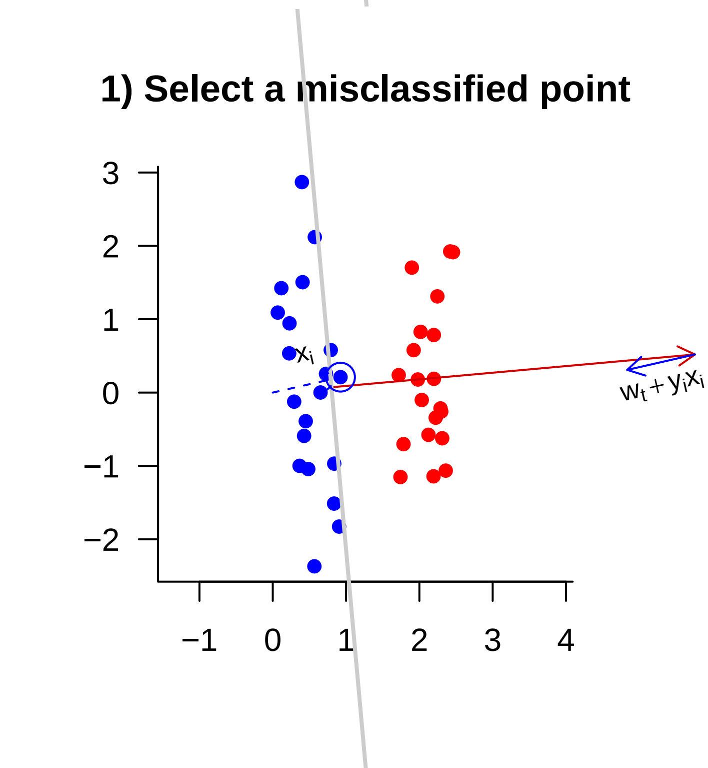
1) Select a misclassified point

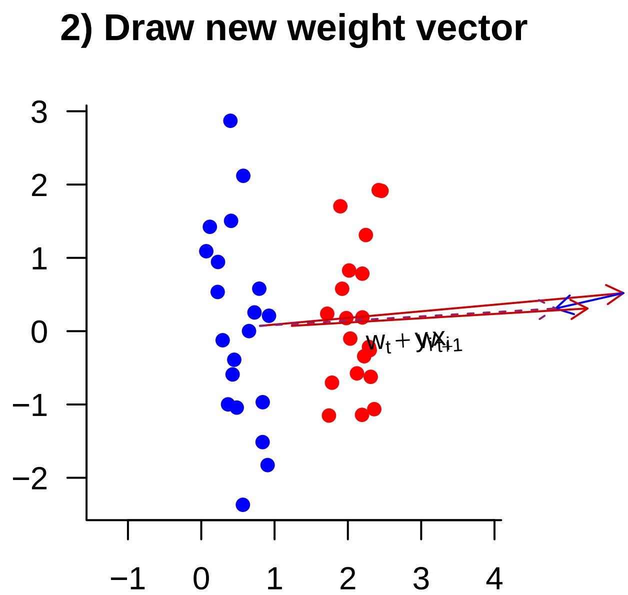


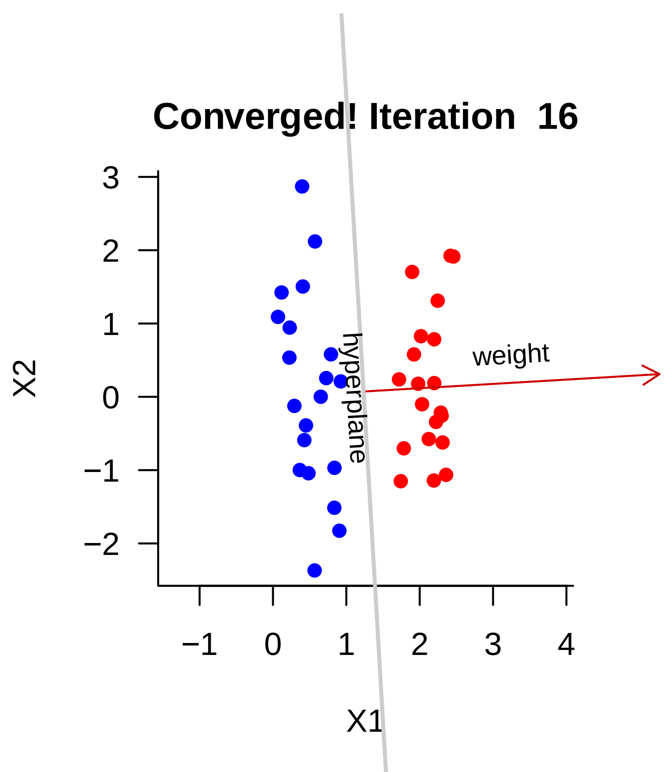
2) Draw new weight vector




Hyperplane, iteration 15
weight
hyperplane
X1
X2


1) Select a misclassified point
x_i
$w_t + y_i x_i$


2) Draw new weight vector
3
2
1
0
−1
−2
−1
0
1
2
3
4
$w_t + yx_i$
w_{t+1}


Converged! Iteration 16
3
2
1
0
−1
−2
−1
0
1
2
3
4
X2
X1
hyperplane
weight

Summary of 16 iterations: