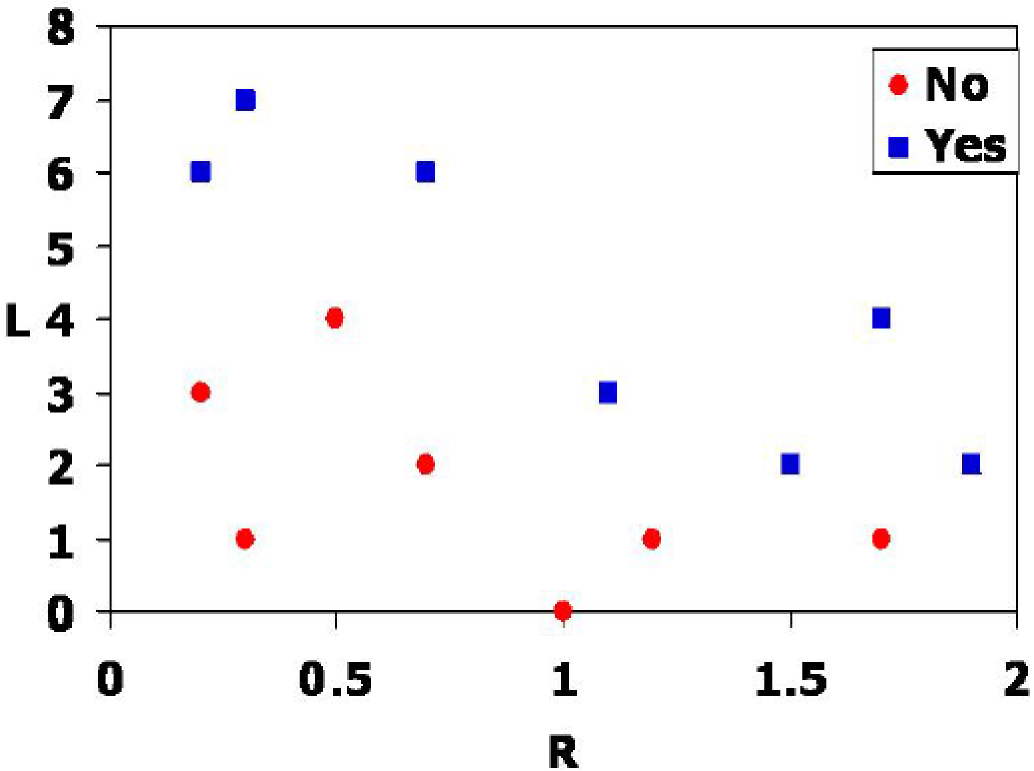
Linear Classifiers I: Perceptron

Bankruptcy example

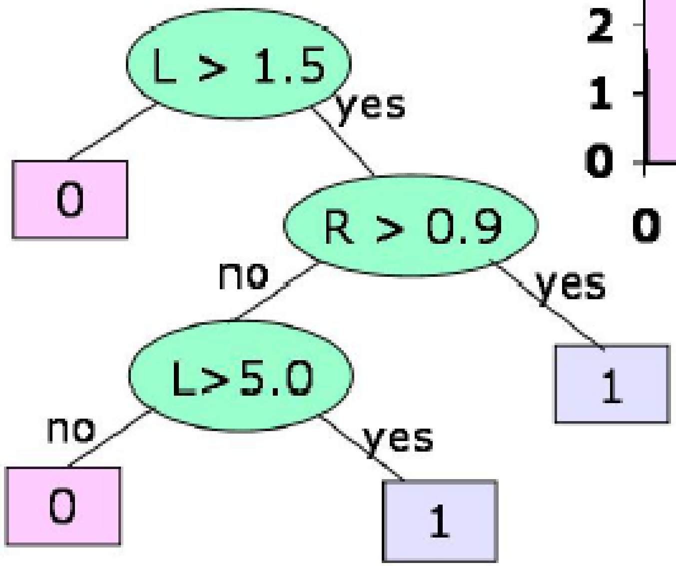
R is the ratio of earnings to expenses

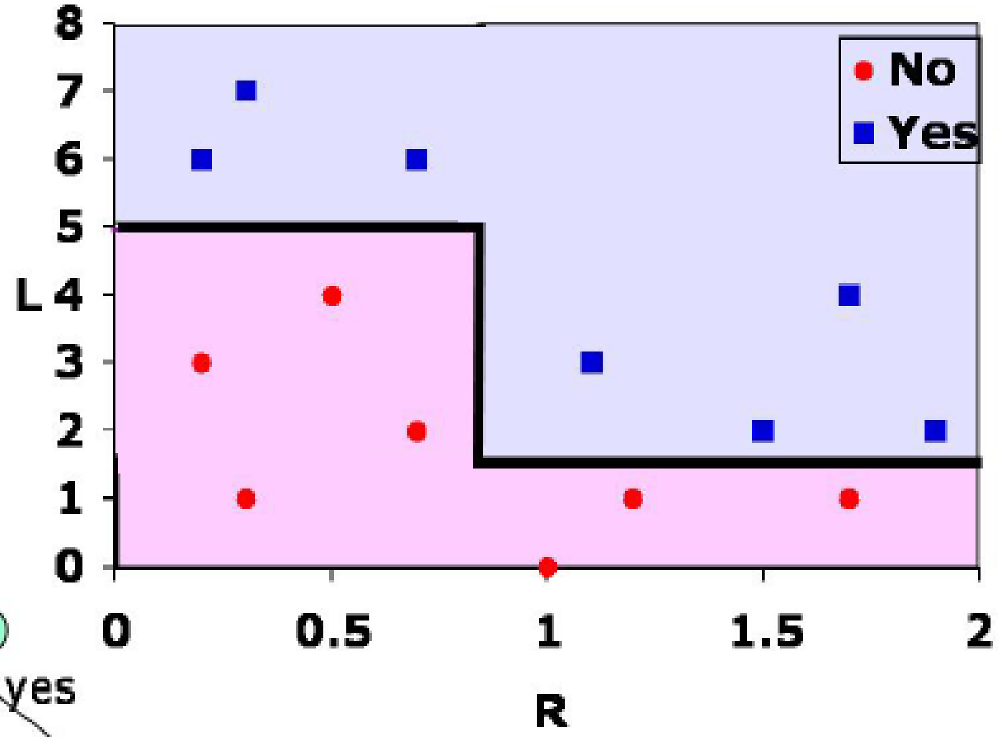
L is the number of late payments on credit cards over the past year.

We would like to draw a **linear separator**, and get so a classifier.

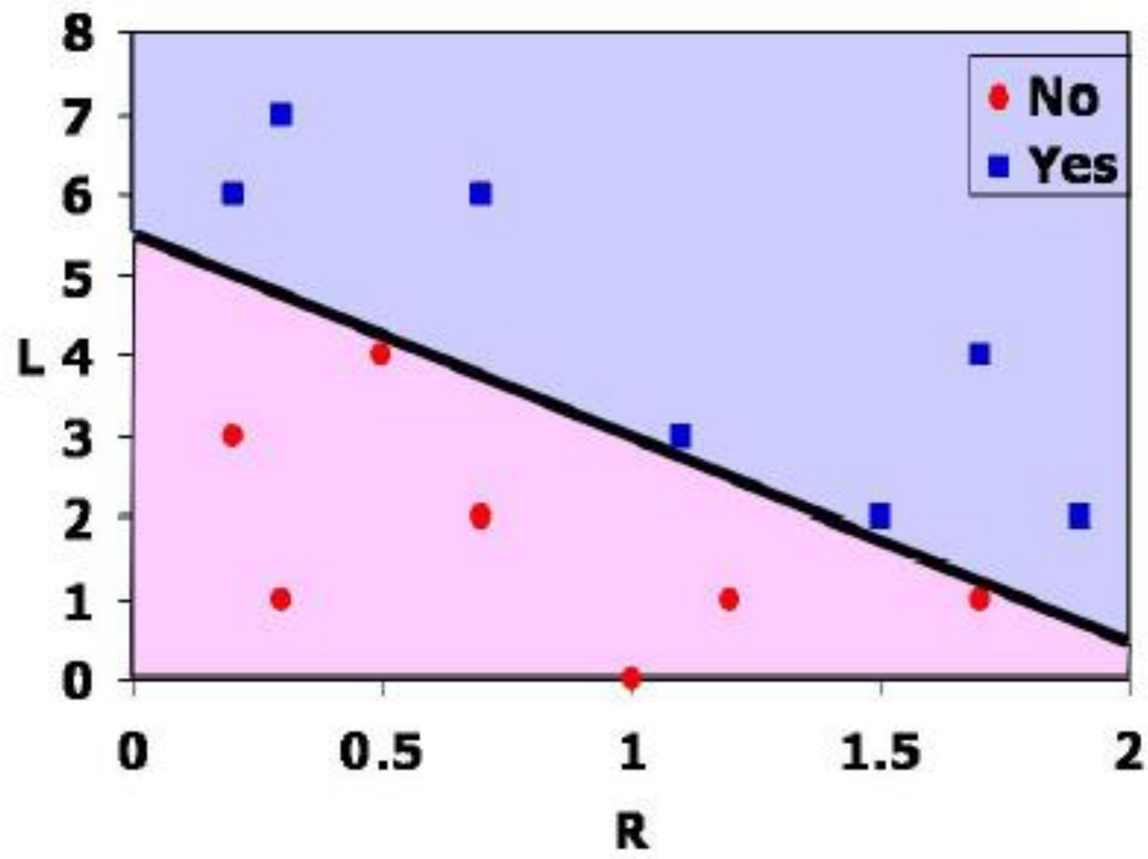


Classification as Boundary: E.g. Decision Tree Boundary





Simple Linear Boundary



Linear Hypothesis Class

- Line equation (assume 2D first):
  $w_2x_2+w_1x_1+b=0$

- **Fact1**: All points (x_1, x_2) lying on the line make the equation true.

- **Fact2**: The line separates the plane in two half-planes.

- **Fact3**: The points (x_1, x_2) in one half-plane give us an inequality with respect to 0, which has the same direction for each of the points in the half-plane.

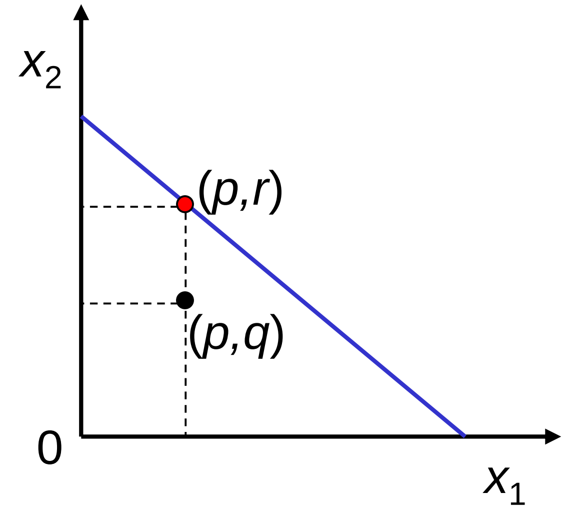
- **Fact4**: The points (x_1, x_2) in the other half-plane give us the reverse inequality with respect to 0.

Fact 3 proof

$w_2x_2+w_1x_1+b=0$

We can write it as:

$$x_2 = -\frac{w_1}{w_2}x_1 - \frac{b}{w_2}$$



(p,r) is on the line so:

$$r = -\frac{w_1}{w_2}p - \frac{b}{w_2}$$

For $q<r$, so we have: $q < r = -\frac{w_1}{w_2}p - \frac{b}{w_2}$ *i.e.*

$$w_2q + w_1p + b < 0 \qquad \text{if} \quad w_2 > 0$$

$$w_2q + w_1p + b > 0 \qquad \text{if} \quad w_2 < 0$$

Since (p,q) was an arbitrary point in the half-plane, we say that the same direction of inequality holds for any other point of the half-plane. Fact 4 is similar.

Linear classifier

$$h(\mathbf{x}, \mathbf{w}, b) = sign(\mathbf{w} \cdot \mathbf{x} + b)$$

Which outputs +1 or –1.

Say:

+1 corresponds to blue, and

-1 to red, or vice versa.

One small change

$$h(\mathbf{x}, \mathbf{w}, b) = \operatorname{sign}\left(\left(\sum_{i=1}^{m} w_i x_i\right) + b\right)$$

$$h(\mathbf{x}, \mathbf{w}) = \operatorname{sign}\left(\left(\sum_{i=1}^{m} w_i x_i\right) + w_0\right)$$

$$w_0 = b$$

$$\mathbf{x} = [1, x_1, .., x_m]$$

$$\mathbf{w} = [w_0, w_1, .., w_m]$$

$$h(\mathbf{x}, \mathbf{w}) = \operatorname{sign}\left(\sum_{i=0}^{m} w_i x_i\right)$$

$$= \operatorname{sign}(\mathbf{w} \cdot \mathbf{x})$$

$$= \operatorname{sign}(\mathbf{w}^T \mathbf{x})$$

Learning Algorithm

Start with random w's

$$h(\mathbf{x}, \mathbf{w}) = \text{sign}(\mathbf{w}^T \mathbf{x})$$

A misclassified tuple will have $y.\mathbf{w}.\mathbf{x}<0$, i.e. either $y=1$ and $\mathbf{w}.\mathbf{x}<0$, or $y=-1$ and $\mathbf{w}.\mathbf{x}>0$.

BTW, $\mathbf{w}.\mathbf{x} = \mathbf{w}^T\mathbf{x}$.

w.x is dot product.

$\mathbf{w}^T\mathbf{x}$ is matrix multiplication of a row matrix with a column matrix. The result is the same.

Training tuples

$\mathbf{x}^1, y^1$

$\mathbf{x}^2, y^2$

...

$\mathbf{x}^n, y^n$

For each misclassified training tuple, e.g.

$$\text{sign}(\mathbf{w}^T \mathbf{x}^k) \neq y^k$$

Update **w**

$$\mathbf{w} = \mathbf{w} + \eta \cdot y^k \mathbf{x}^k$$

In the original algorithm, **w** is updated and used immediately for the next training tuple.

In the Excel example, we wait until we process all the tuples before updating **w**.

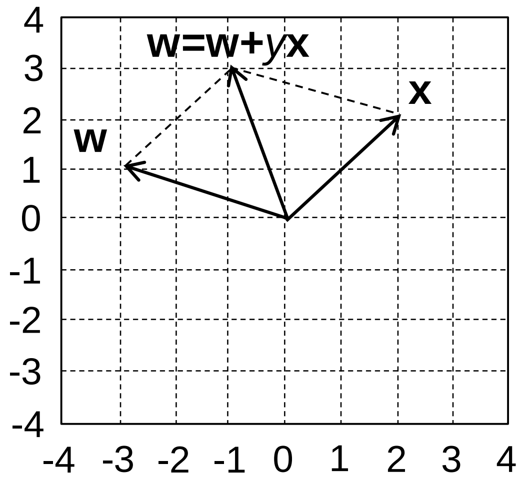
Well, why is this a good rule?

It can be shown that if the data is linearly separable, and we repeat this procedure many times, we will get a line that separates the training tuples.

η is the learning rate, $0<\eta<=1$

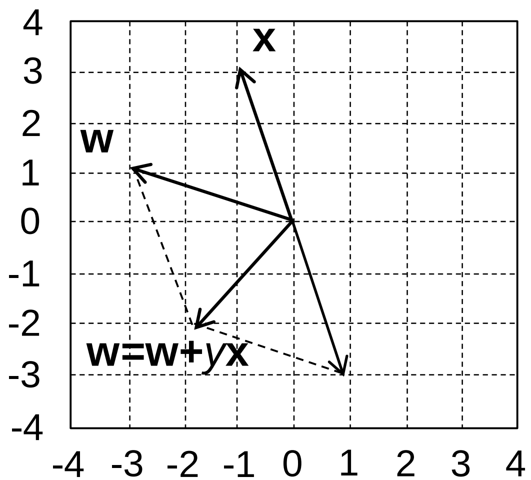
Sign of dot product and **misclassification**

y=+1
w.x<0
y.**w.x**<0



In the first case, the sum w=w+yx will bring w closer, angle-wise, to x, and hopefully the angle becomes acute, and thus the dot product becomes positive, same as class y=+1, obtaining y.w.x>0.

y=-1
w.x>0
y.**w.x**<0



In the second case, the sum w=w+yx will take w farther, angle-wise, from x, and hopefully the angle becomes obtuse, and thus the dot product becomes negative, same as class y=-1, obtaining y.w.x>0.

Two vectors a,b.
Facts:

a.b>0 iff angle between a and b is **acute**.

a.b<0 iff angle between a and b is **obtuse**.