PERCEPTRON LEARNING

David Kauchak
CS 451 – Fall 2013

Admin

Assignment 1 solution available online

Assignment 2: Due date?

Due Sunday at midnight

Assignment 2 competition site setup later today

Machine learning models

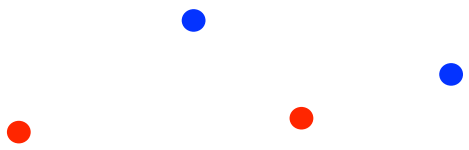
Some machine learning approaches make strong assumptions about the data

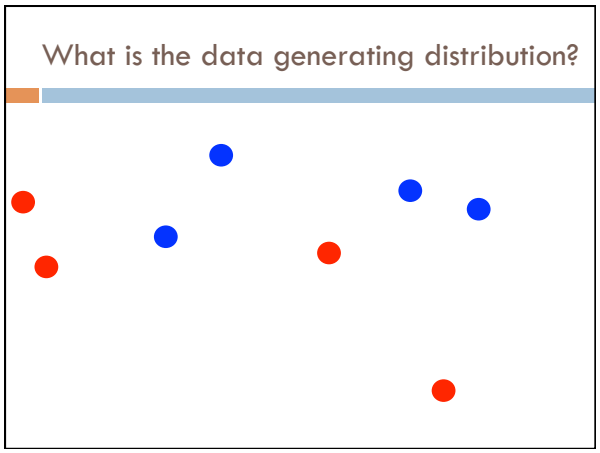
- If the assumptions are true this can often lead to better performance
- If the assumptions aren't true, they can fail miserably

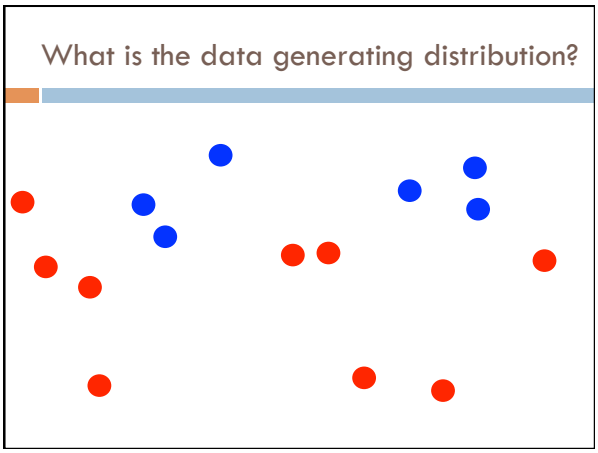
Other approaches don't make many assumptions about the data

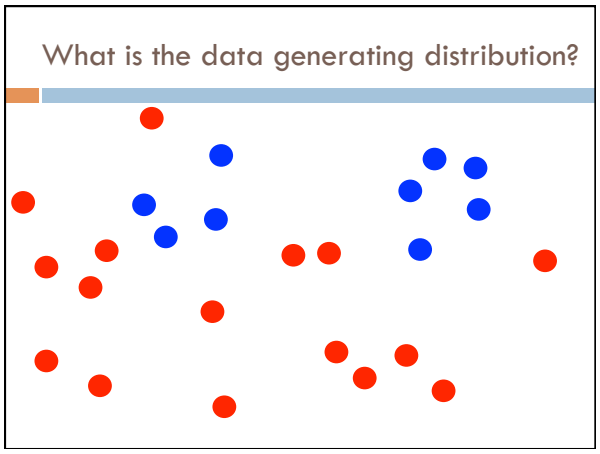
- This can allow us to learn from more varied data
- But, they are more prone to overfitting
- and generally require more training data

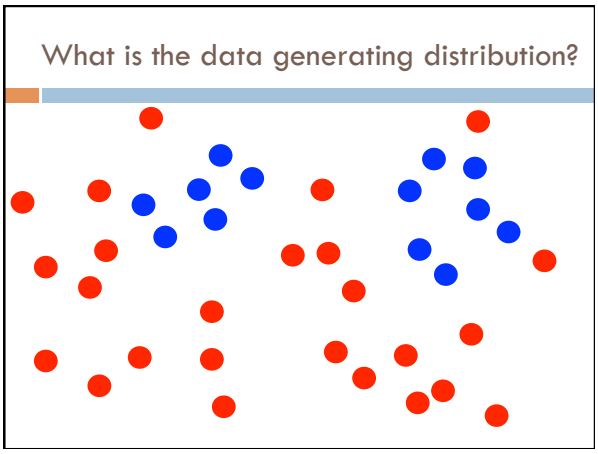
What is the data generating distribution?



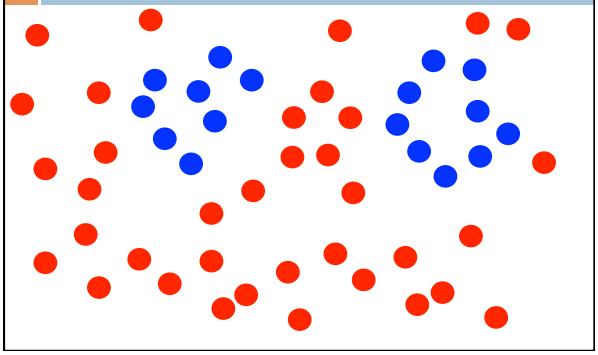

What is the data generating distribution?


What is the data generating distribution?

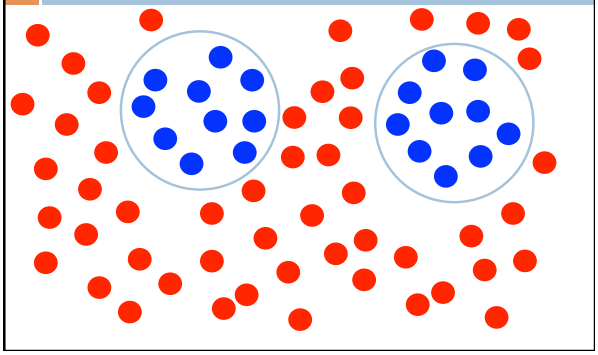

What is the data generating distribution?


What is the data generating distribution?

What is the data generating distribution?



Actual model

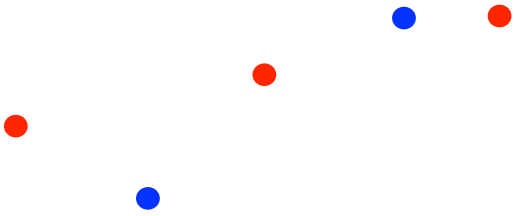


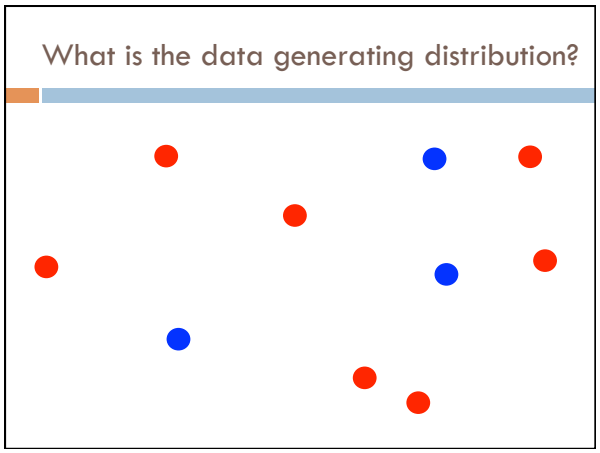
Model assumptions

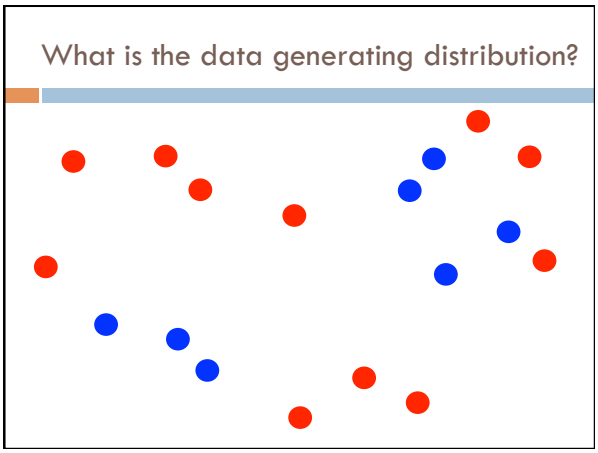
If you don't have strong assumptions about the model, it can take you a longer to learn

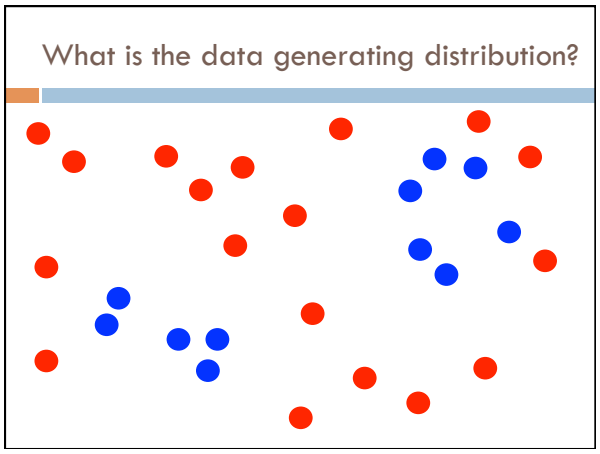
Assume now that our model of the blue class is two circles

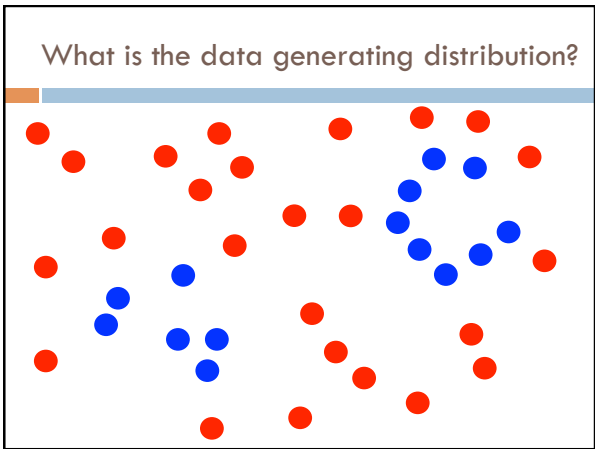
What is the data generating distribution?



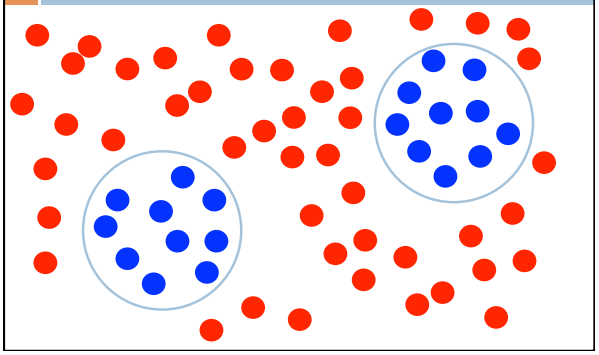

What is the data generating distribution?


What is the data generating distribution?

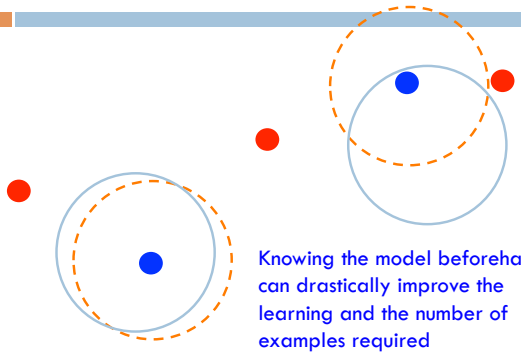

What is the data generating distribution?


What is the data generating distribution?

Actual model

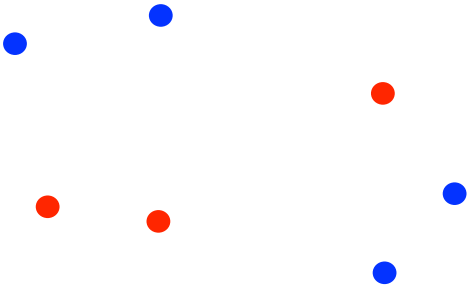


What is the data generating distribution?

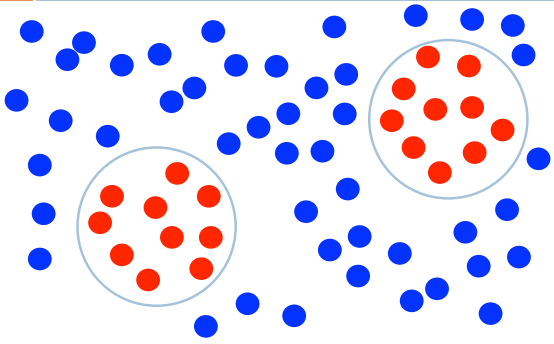


Knowing the model beforehand can drastically improve the learning and the number of examples required

What is the data generating distribution?



Make sure your assumption is correct, though!

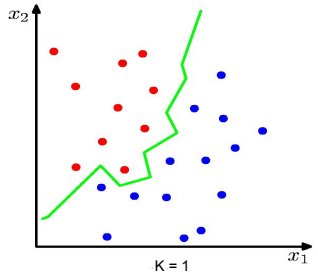


Machine learning models

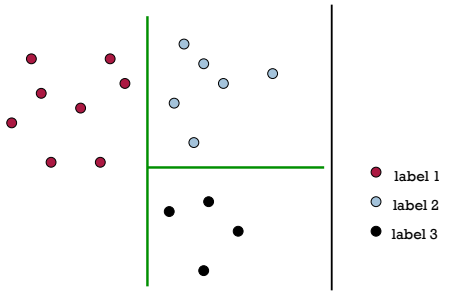
What were the *model* assumptions (if any) that *k*-NN and decision trees make about the data?

Are there data sets that could never be learned correctly by either?

k-NN model



Decision tree model



Axis-aligned splits/cuts of the data

Bias

The "bias" of a model is how strong the model assumptions are.

low-bias classifiers make minimal assumptions about the data (*k*-NN and DT are generally considered low bias

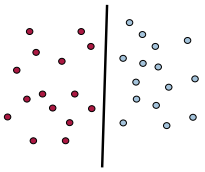
high-bias classifiers make strong assumptions about the data

Linear models

A strong high-bias assumption is *linear separability*:

- in 2 dimensions, can separate classes by a line
- in higher dimensions, need hyperplanes

A *linear model* is a model that assumes the data is linearly separable



Hyperplanes

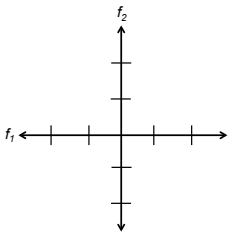
A hyperplane is line/plane in a high dimensional space

What defines a line?
What defines a hyperplane?

Defining a line

Any pair of values (w_1,w_2) defines a line through the origin:

$$0 = w_1 f_1 + w_2 f_2$$



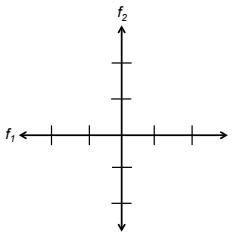
Defining a line

Any pair of values (w_1,w_2) defines a line through the origin:

$$0 = w_1 f_1 + w_2 f_2$$

$$0 = 1 f_1 + 2 f_2$$

-2	1
-1	0.5
0	0
1	-0.5
2	-1



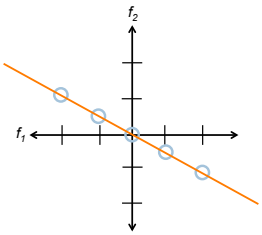
Defining a line

Any pair of values (w_1,w_2) defines a line through the origin:

$$0 = w_1 f_1 + w_2 f_2$$

$$0 = 1 f_1 + 2 f_2$$

-2	1
-1	0.5
0	0
1	-0.5
2	-1



Defining a line

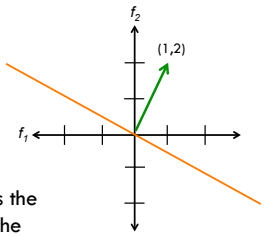
Any pair of values (w_1,w_2) defines a line through the origin:

$$0 = w_1 f_1 + w_2 f_2$$

$$0 = 1 f_1 + 2 f_2$$

w=(1,2)

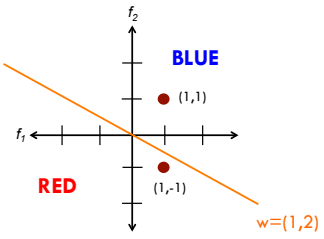
We can also view it as the line perpendicular to the *weight vector*



Classifying with a line

Mathematically, how can we classify points based on a line?

$$0 = 1 f_1 + 2 f_2$$



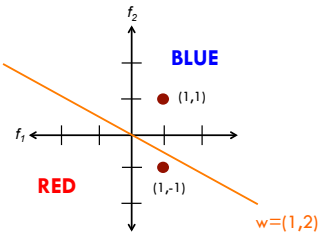
Classifying with a line

Mathematically, how can we classify points based on a line?

$$0 = 1 f_1 + 2 f_2$$

(1,1): $1*1+2*1=3$

(1,-1): $1*1+2*-1=-1$



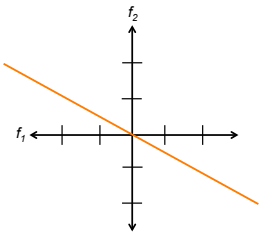
The sign indicates which side of the line

Defining a line

Any pair of values (w_1,w_2) defines a line through the origin:

$$0 = w_1 f_1 + w_2 f_2$$

$$0 = 1 f_1 + 2 f_2$$



How do we move the line off of the origin?

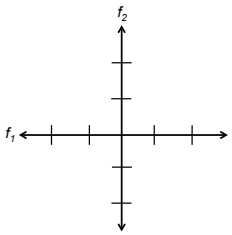
Defining a line

Any pair of values (w_1,w_2) defines a line through the origin:

$$a = w_1 f_1 + w_2 f_2$$

$$-1 = 1 f_1 + 2 f_2$$

-2
-1
0
1
2



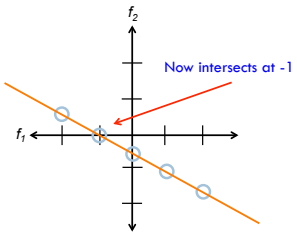
Defining a line

Any pair of values (w_1,w_2) defines a line through the origin:

$$a = w_1 f_1 + w_2 f_2$$

$$-1 = 1 f_1 + 2 f_2$$

-2	0.5
-1	0
0	-0.5
1	-1
2	-1.5



Linear models

A linear model in *n*-dimensional space (i.e. *n* features) is define by *n*+1 weights:

In two dimensions, a line:

$$0 = w_1 f_1 + w_2 f_2 + b$$ (where b = -a)

In three dimensions, a plane:

$$0 = w_1 f_1 + w_2 f_2 + w_3 f_3 + b$$

In *n*-dimensions, a *hyperplane*

$$0 = b + \sum_{i=1}^{n} w_i f_i$$

Classifying with a linear model

We can classify with a linear model by checking the sign:

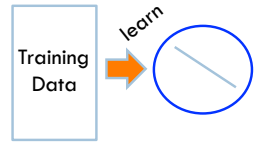
$f_1, f_2, \ldots, f_n$ → classifier → $b + \sum_{i=1}^{n} w_i f_i > 0$ Positive example

$b + \sum_{i=1}^{n} w_i f_i < 0$ Negative example

Learning a linear model

Geometrically, we know what a linear model represents

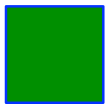
Given a linear model (i.e. a set of weights and b) we can classify examples



(data with labels)

How do we learn a linear model?

Positive or negative?



NEGATIVE

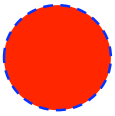
Positive or negative?

NEGATIVE

Positive or negative?

POSITIVE

Positive or negative?



NEGATIVE

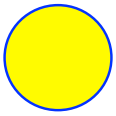
Positive or negative?

POSITIVE

Positive or negative?

POSITIVE

Positive or negative?



NEGATIVE

Positive or negative?

POSITIVE

A method to the madness

blue = positive

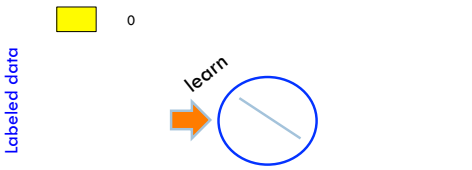
yellow triangles = positive

all others negative

How is this learning setup different than the learning we've done before?

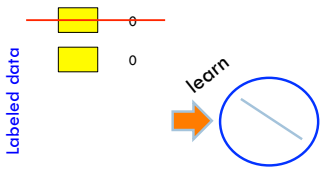
When might this arise?

Online learning algorithm



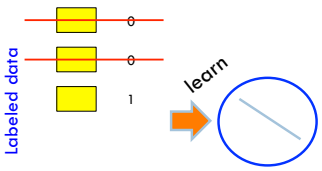
Only get to see one example at a time!

Online learning algorithm



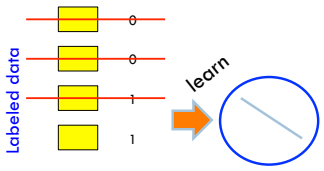
Only get to see one example at a time!

Online learning algorithm



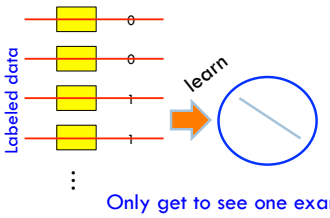
Only get to see one example at a time!

Online learning algorithm



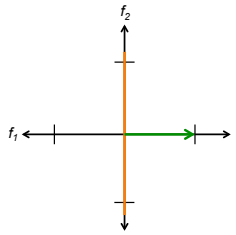
Only get to see one example at a time!

Online learning algorithm



Only get to see one example at a time!

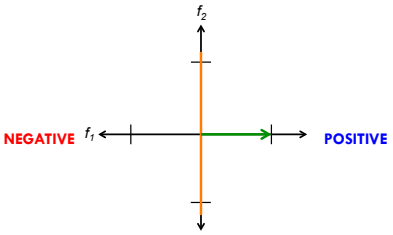
Learning a linear classifier



What does this model currently say?

w=(1,0)

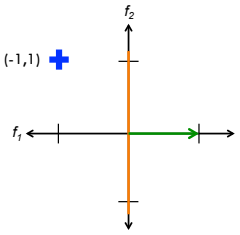
Learning a linear classifier



w=(1,0)

Learning a linear classifier

$$0 = w_1 f_1 + w_2 f_2$$



Is our current guess: right or wrong?

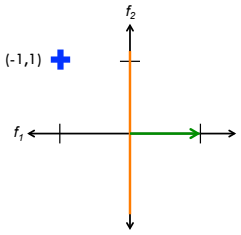
w=(1,0)

Learning a linear classifier

$$0 = w_1 f_1 + w_2 f_2$$

$$1 * f_1 + 0 * f_2 =$$

$$1 * -1 + 0 * 1 = -1$$



predicts negative, wrong

How should we update the model?

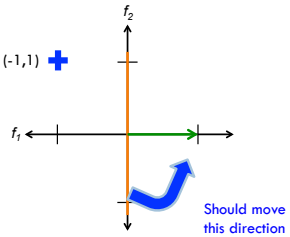
w=(1,0)

Learning a linear classifier

$$0 = w_1 f_1 + w_2 f_2$$

$$1 * f_1 + 0 * f_2 =$$

$$1 * -1 + 0 * 1 = -1$$



w=(1,0)

A closer look at why we got it wrong

w_1 w_2 (-1, 1, positive)

$$1 * f_1 + 0 * f_2 =$$

$$1 * -1 + 0 * 1 = -1$$ ← We'd like this value to be positive since it's a positive value

Which of these contributed to the mistake?

A closer look at why we got it wrong

w_1 w_2 (-1, 1, positive)

$$1 * f_1 + 0 * f_2 =$$

$$1 * -1 + 0 * 1 = -1$$ ← We'd like this value to be positive since it's a positive value

contributed in the wrong direction

could have contributed (positive feature), but didn't

How should we change the weights?

A closer look at why we got it wrong

w_1 w_2 (-1, 1, positive)

$$1 * f_1 + 0 * f_2 =$$

$$1 * -1 + 0 * 1 = -1$$ ← We'd like this value to be positive since it's a positive value

contributed in the wrong direction

could have contributed (positive feature), but didn't

decrease

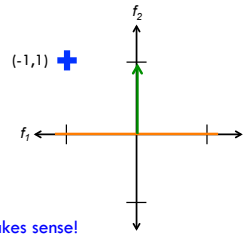
1 -> 0

increase

0 -> 1

Learning a linear classifier

$0 = w_1 f_1 + w_2 f_2$

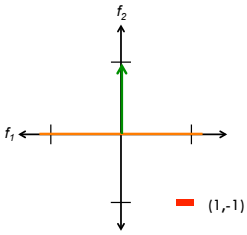


Graphically, this also makes sense!

w=(0,1)

Learning a linear classifier

$0 = w_1 f_1 + w_2 f_2$



Is our current guess: right or wrong?

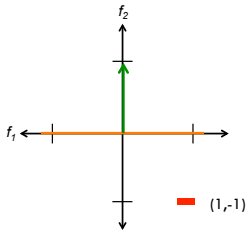
w=(0,1)

Learning a linear classifier

$0 = w_1 f_1 + w_2 f_2$

$0 * f_1 + 1 * f_2 =$

$0 * 1 + 1 * -1 = -1$



predicts negative, correct

How should we update the model?

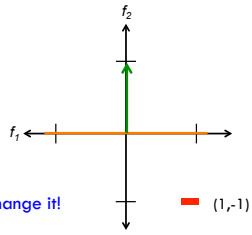
w=(0,1)

Learning a linear classifier

$0 = w_1 f_1 + w_2 f_2$

$0 * f_1 + 1 * f_2 =$

$0 * 1 + 1 * -1 = -1$

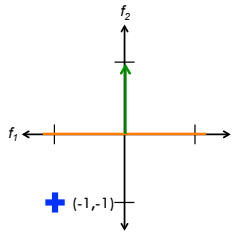


Already correct… don't change it!

w=(0,1)

Learning a linear classifier

$0 = w_1 f_1 + w_2 f_2$



Is our current guess: right or wrong?

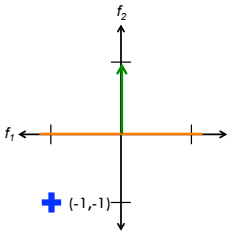
w=(0,1)

Learning a linear classifier

$0 = w_1 f_1 + w_2 f_2$

$0 * f_1 + 1 * f_2 =$

$0 * -1 + 1 * -1 = -1$



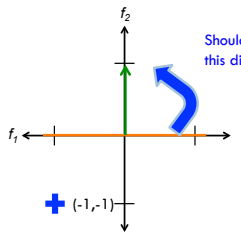
predicts negative, wrong

How should we update the model?

w=(0,1)

Learning a linear classifier

$0 = w_1 f_1 + w_2 f_2$



w=(0,1)

A closer look at why we got it wrong

w_1 w_2

(-1, -1, positive)

$0 * f_1 + 1 * f_2 =$

$0 * -1 + 1 * -1 = -1$

We'd like this value to be positive since it's a positive value

Which of these contributed to the mistake?

A closer look at why we got it wrong

w_1 w_2 (-1, -1, positive)

$0 * f_1 + 1 * f_2 =$

$0 * -1 + 1 * -1 = -1$ ← We'd like this value to be positive since it's a positive value

didn't contribute, but could have

contributed in the wrong direction

How should we change the weights?

A closer look at why we got it wrong

w_1 w_2 (-1, -1, positive)

$0 * f_1 + 1 * f_2 =$

$0 * -1 + 1 * -1 = -1$ ← We'd like this value to be positive since it's a positive value

didn't contribute, but could have

decrease

0 -> -1

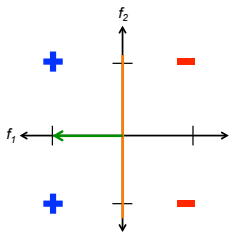
contributed in the wrong direction

decrease

1 -> 0

Learning a linear classifier

f_1, f_2, label
-1,-1, positive
-1, 1, positive
1, 1, negative
1,-1, negative



w=(-1,0)