ADVANCED CLASSIFICATION TECHNIQUES

David Kauchak
CS 159 – Spring 2023

1

Admin

Assignment 7

Next Monday: project proposal presentations
- informal
- 1 minute
- See the final project handout for details

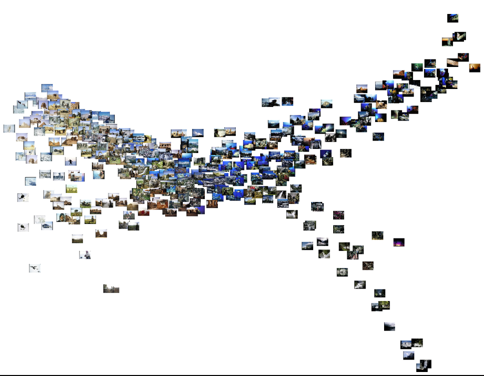
Hack week QA session from OpenAI engineer (Friday @ 12:30pm)
- https://5chack.com/#hack-week

2

Schedule for the rest of the semester

3

Machine Learning: A Geometric View



4

Apples vs. Bananas

Weight	Color	Label
4	Red	Apple
5	Yellow	Apple
6	Yellow	Banana
3	Red	Apple
7	Yellow	Banana
8	Yellow	Banana
6	Yellow	Apple

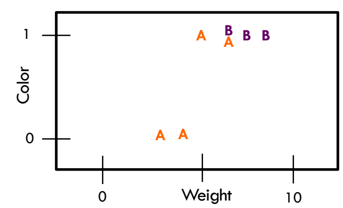
Can we visualize this data?

5

Apples vs. Bananas

Turn features into numerical values

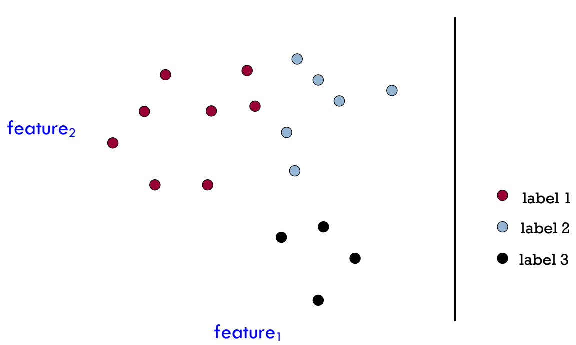
Weight	Color	Label
4	0	Apple
5	1	Apple
6	1	Banana
3	0	Apple
7	1	Banana
8	1	Banana
6	1	Apple



We can view examples as points in an *n*-dimensional space where *n* is the number of features called the feature space

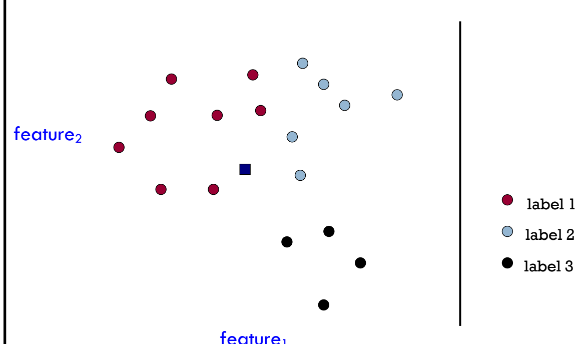
6

Examples in a feature space



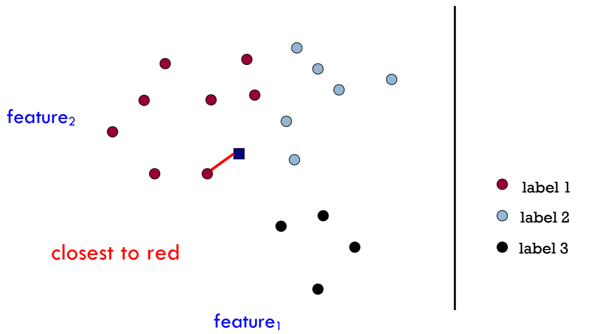
7

Test example: what class?



8

Test example: what class?

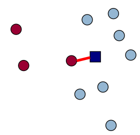


9

Another classification algorithm?

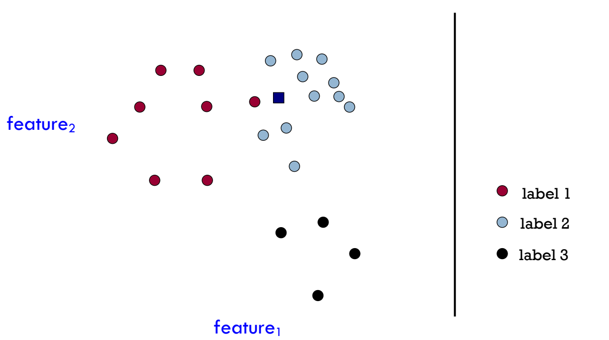
To classify an example ***d***:

Label ***d*** with the label of the closest example to ***d*** in the training set



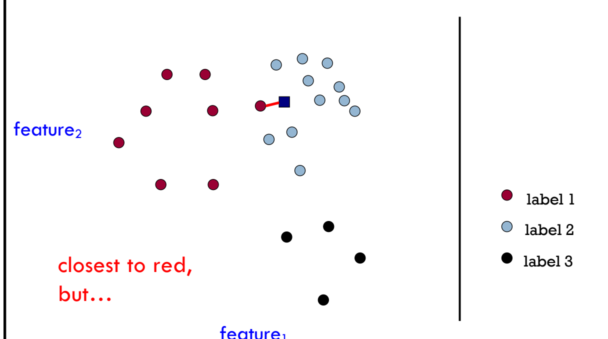
10

What about this example?



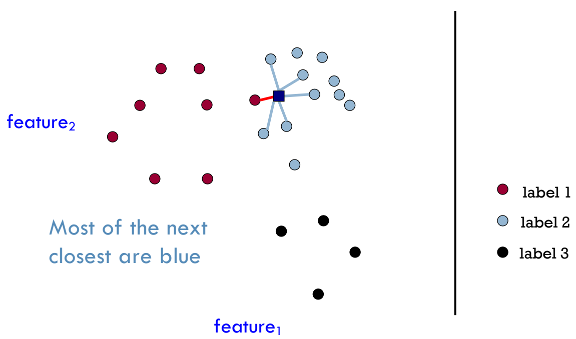
11

What about this example?



12

What about this example?



13

k-Nearest Neighbor (k-NN)

To classify an example ***d***:

- Find ***k*** nearest neighbors of ***d***
- Choose as the label the majority label within the ***k*** nearest neighbors

14

k-Nearest Neighbor (k-NN)

To classify an example ***d***:

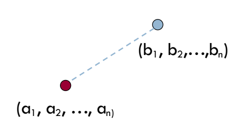
- Find ***k*** *nearest* neighbors of ***d***
- Choose as the label the majority label within the ***k*** nearest neighbors

How do we measure "nearest"?

15

Euclidean distance

Euclidean distance! (or L1 or cosine or …)

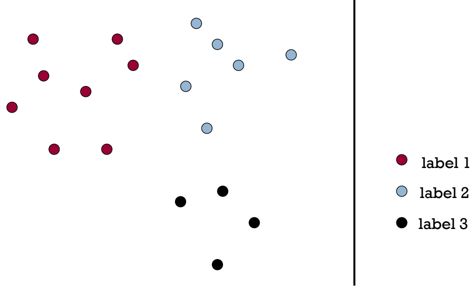


$$D(a,b) = \sqrt{(a_1-b_1)^2 + (a_2-b_2)^2 + \ldots + (a_n-b_n)^2}$$

16

Decision boundaries

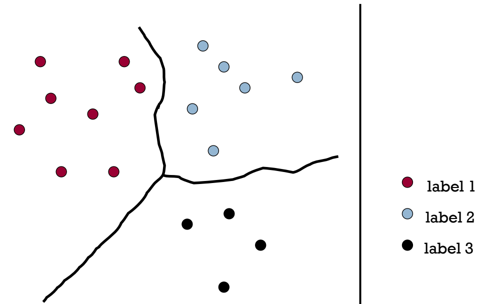
The **decision boundaries** are places in the features space where the classification of a point/example changes



Where are the decision boundaries for k-NN?

17

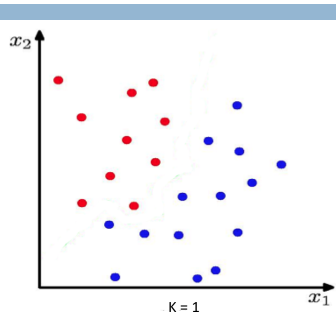
k-NN decision boundaries



k-NN gives locally defined decision boundaries between classes

18

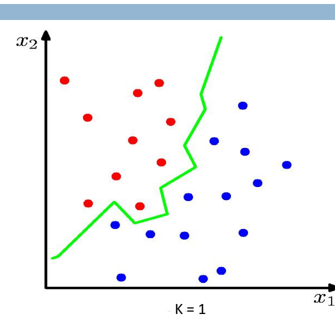
K Nearest Neighbour (k-NN) Classifier



What is the decision boundary for k-NN for this one?

19

K Nearest Neighbour (*k*NN) Classifier



20

Machine learning models

Some machine learning approaches make strong assumptions about the data

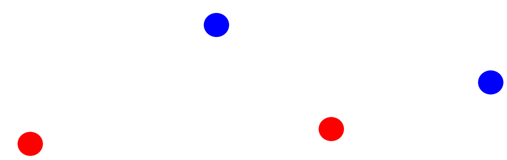
- If the assumptions are true this can often lead to better performance
- If the assumptions aren't true, they can fail miserably

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 (biasing too much to the training data)
- and generally require more training data

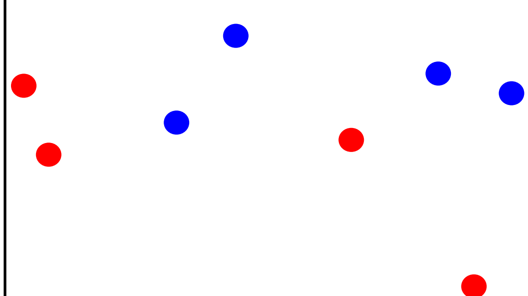
21

What is the data generating distribution?



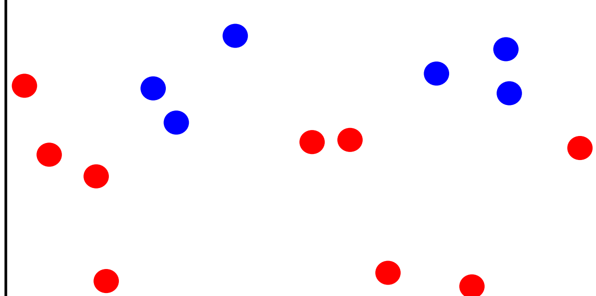
22

What is the data generating distribution?



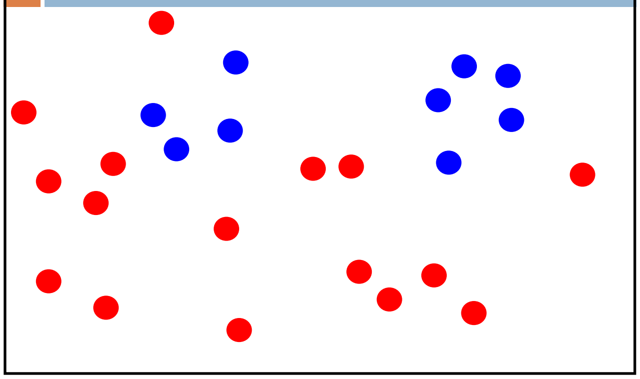
23

What is the data generating distribution?



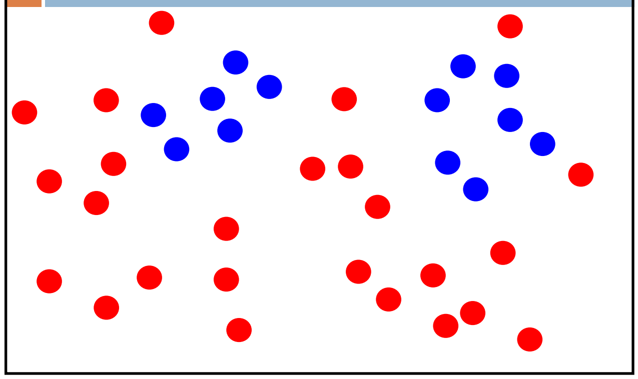
24

What is the data generating distribution?



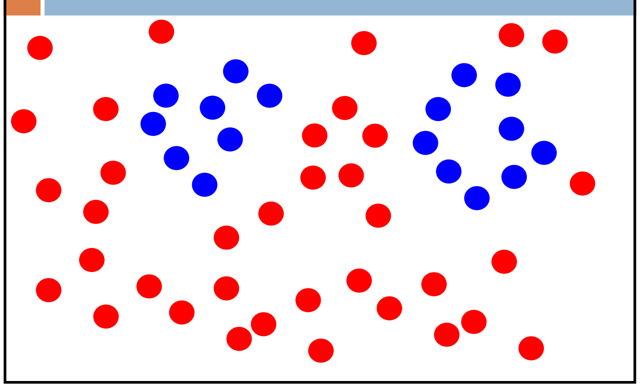
25

What is the data generating distribution?



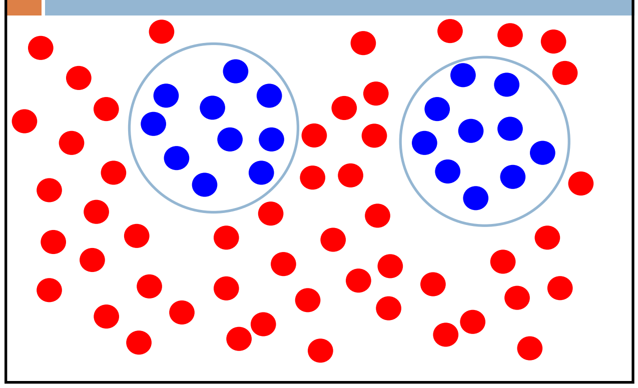
26

What is the data generating distribution?



27

Actual model



28

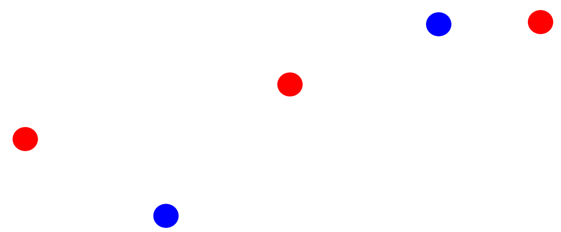
Model assumptions

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

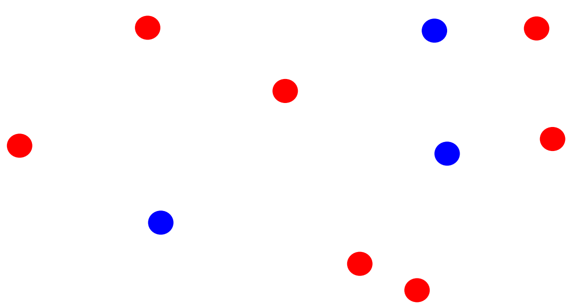
29

What is the data generating distribution?



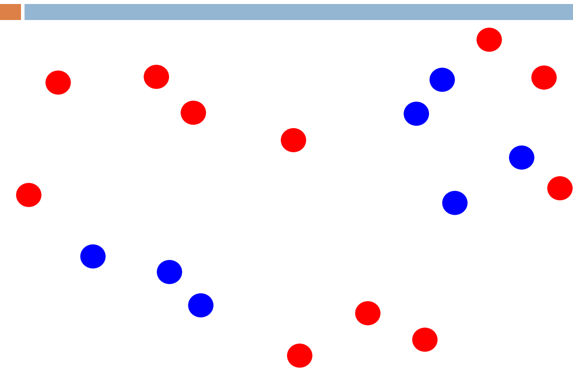
30

What is the data generating distribution?



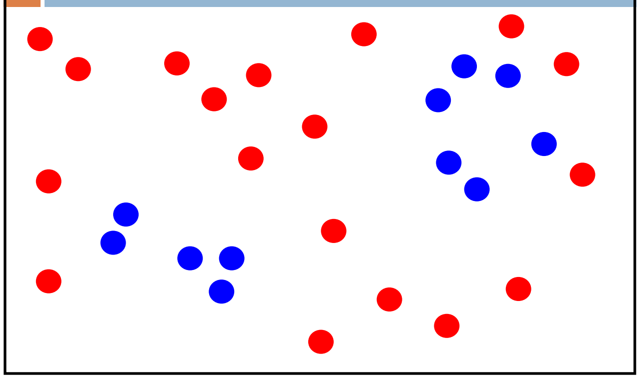
31

What is the data generating distribution?



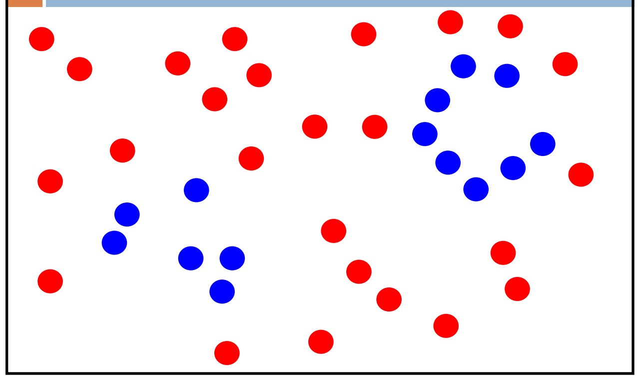
32

What is the data generating distribution?



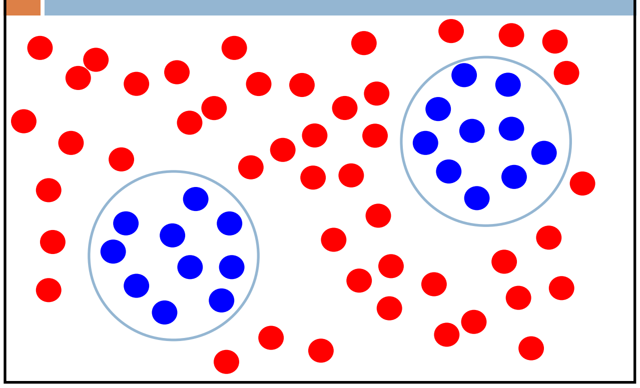
33

What is the data generating distribution?



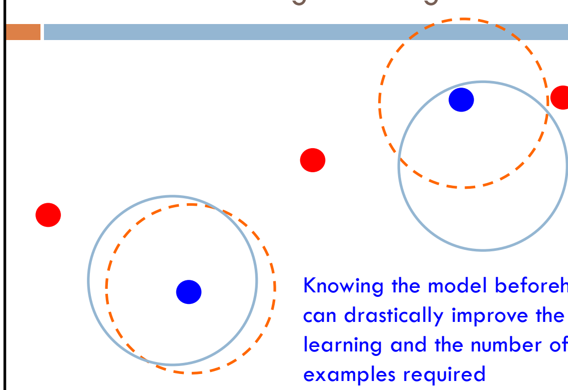
34

Actual model



35

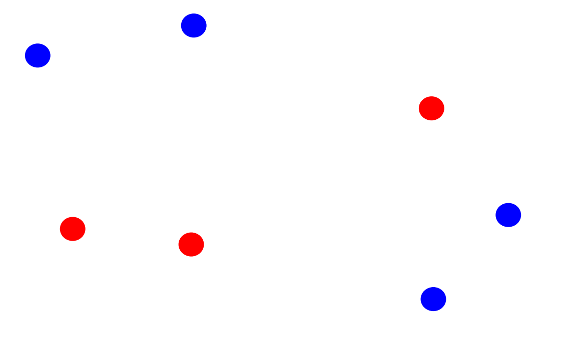
What is the data generating distribution?



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

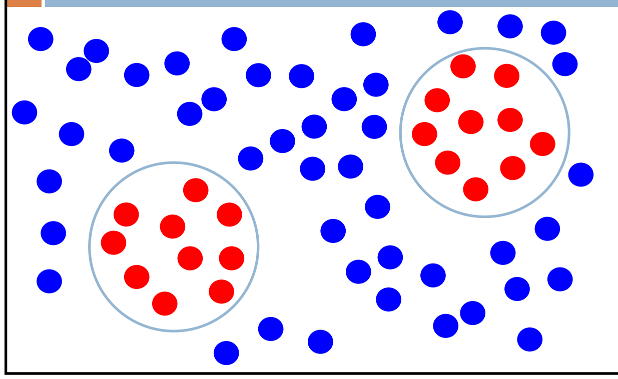
36

What is the data generating distribution?



37

Make sure your assumption is correct, though!



38

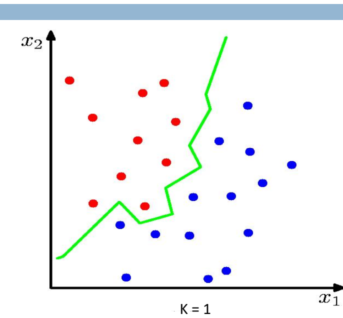
Machine learning models

What were the *model* assumptions (if any) that *k*-NN and NB made about the data?

Are there training data sets that could never be learned correctly by these algorithms?

39

k-NN model



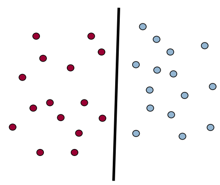
40

Linear models

A strong assumption is *linear separability*:

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

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



41

Hyperplanes

A hyperplane is line/plane in a high dimensional space

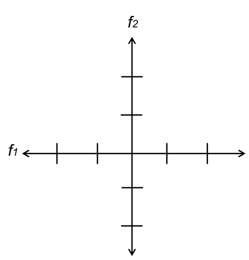
What defines a line?
What defines a hyperplane?

42

Defining a line

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

$$0 = w_1 f_1 + w_2 f_2$$



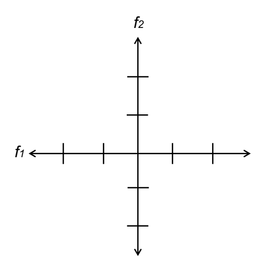
43

Defining a line

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

$$0 = w_1 f_1 + w_2 f_2$$

$$0 = 1 f_1 + 2 f_2$$



What does this line look like?

44

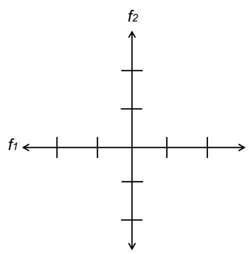
Defining a line

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



45

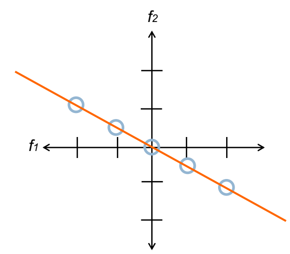
Defining a line

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



46

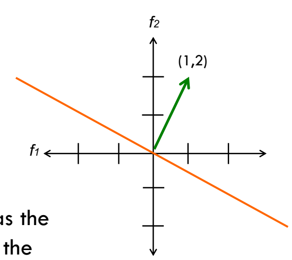
Defining a line

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)



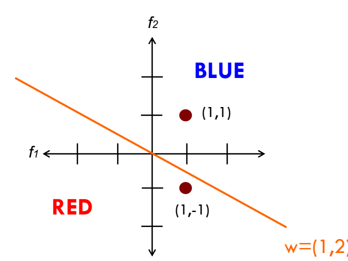
We can also view it as the line perpendicular to the *weight vector*

47

Classifying with a line

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

$$0 = 1 f_1 + 2 f_2$$



48

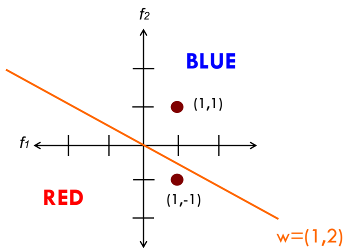
Classifying with a line

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

$$0 = 1f_1 + 2f_2$$

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

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



The sign indicates which side of the line

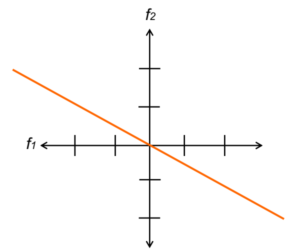
49

Defining a line

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

$$0 = w_1 f_1 + w_2 f_2$$

$$0 = 1f_1 + 2f_2$$



How do we move the line off of the origin?

50

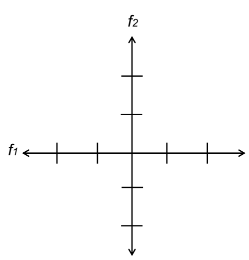
Defining a line

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

$$a = w_1 f_1 + w_2 f_2$$

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

-2
-1
0
1
2



51

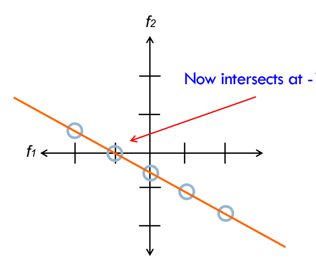
Defining a line

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

$$a = w_1 f_1 + w_2 f_2$$

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

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



52

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*

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

53

Classifying with a linear model

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

$f_1, f_2, \ldots, f_m$ ➡ classifier

$b + \sum_{j=1}^{m} w_j f_j > 0$ Positive example

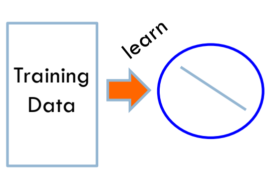
$b + \sum_{j=1}^{m} w_j f_j < 0$ Negative example

54

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

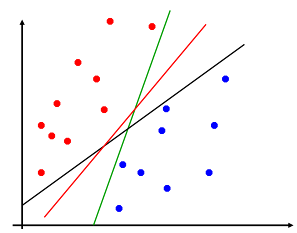


(data with labels)

How do we learn a linear model?

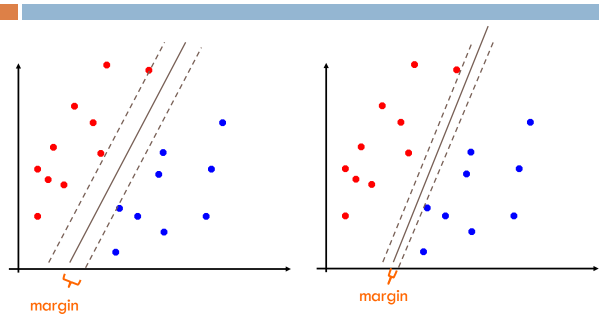
55

Which hyperplane would you choose?



56

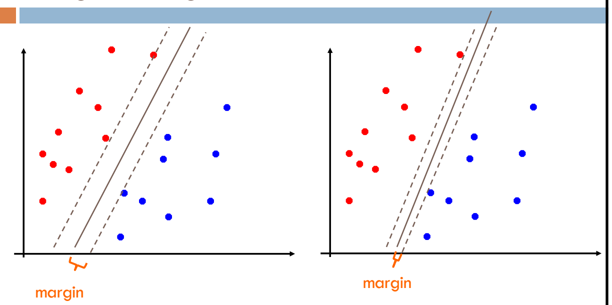
Large margin classifiers



Choose the line where the distance to the nearest point(s) is as large as possible

57

Large margin classifiers



The margin of a classifier is the distance to the closest points of either class

Large margin classifiers attempt to maximize this

58

Large margin classifier setup

Select the hyperplane with the largest margin where the points are classified correctly!

Setup as a constrained optimization problem:

$$\max_{w,b} \ \text{margin}(w,b)$$

subject to:

$$y_i(w \cdot x_i + b) > 0 \quad \forall i$$

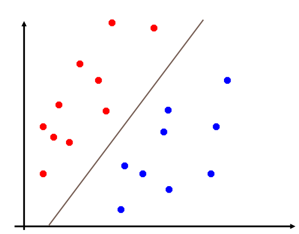
what does this say?

y_i: label for example i, either 1 (positive) or -1 (negative)
x_i: our feature vector for example i

59

Measuring the margin

How do we calculate the margin?

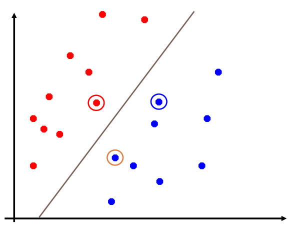


60

Support vectors

For any separating hyperplane, there exist some set of "closest points"

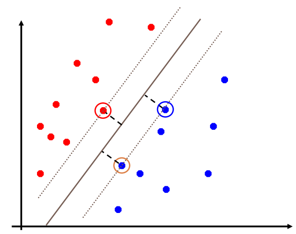
These are called the support vectors



61

Measuring the margin

The margin is the distance to the support vectors, i.e. the "closest points", on either side of the hyperplane



62

Support vector machine problem

Posed as a quadratic optimization problem

Maximize/minimize a quadratic function

Subject to a set of linear constraints

Many, many variants of solving this problem

One of the most successful classification approaches

63

Support vector machines

One of the most successful (if not the most successful) classification approach:

decision tree	About 2,240,000 results (**0.32** sec)
Support vector machine	About 2,180,000 results (**0.36** sec)
k nearest neighbor	About 844,000 results (**0.33** sec)
Naïve Bayes	About 71,300 results (**0.32** sec)

Google scholar

64

Other successful classifiers in NLP

Perceptron algorithm
- Linear classifier
- Trains "online"
- Fast and easy to implement
- Often used for tuning parameters (not necessarily for classifying)

Logistic regression classifier (aka Maximum entropy classifier)
- Probabilistic classifier
- Doesn't have the NB constraints
- Performs very well
- More computationally intensive to train than NB

71