

ADVANCED CLASSIFICATION
TECHNIQUES

David Kauchak
CS 159 – Fall 2024

1

Admin

Assignment 7 out (last one!!)

Thursday:

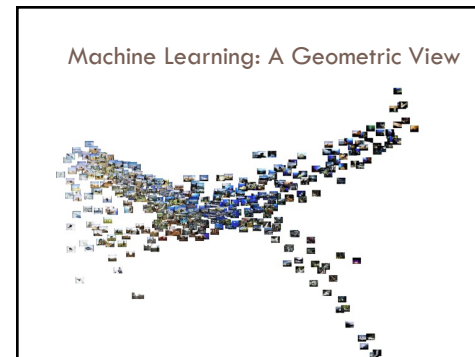
- 30 minute class discussing final project details
- Take Quiz 3 sometime during the day

Grading

2

Schedule for the rest of the semester

3



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

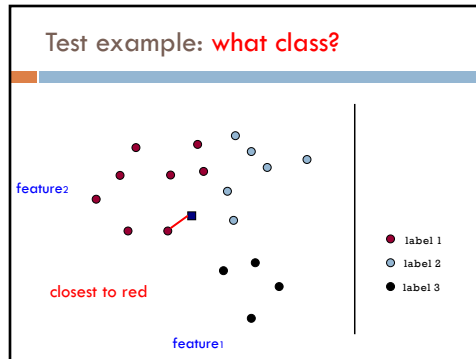
- label 1
- label 2
- label 3

7

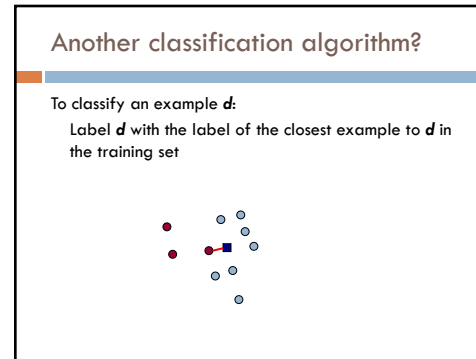
Test example: what class?

- label 1
- label 2
- label 3

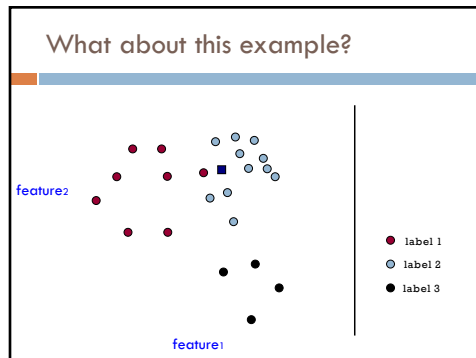
8



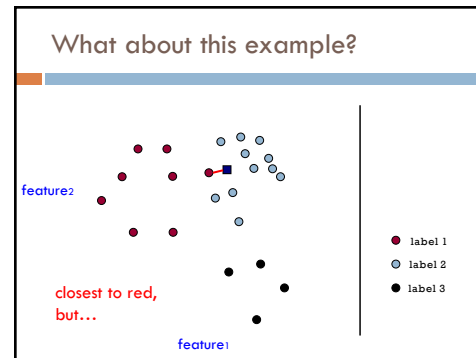
9



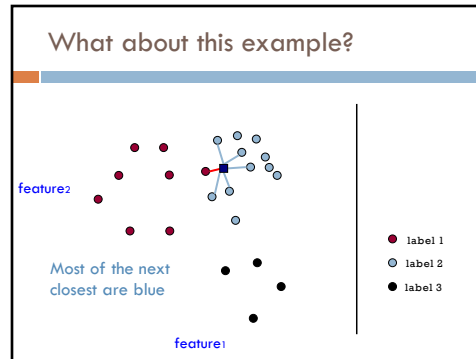
10



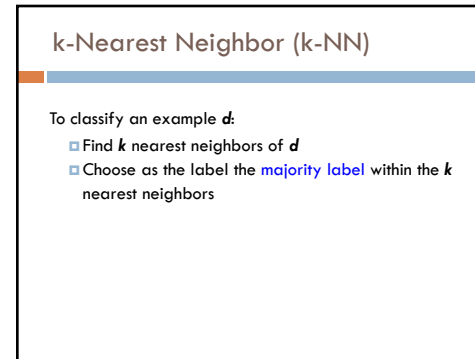
11



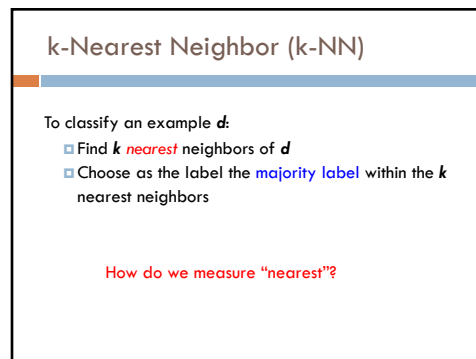
12



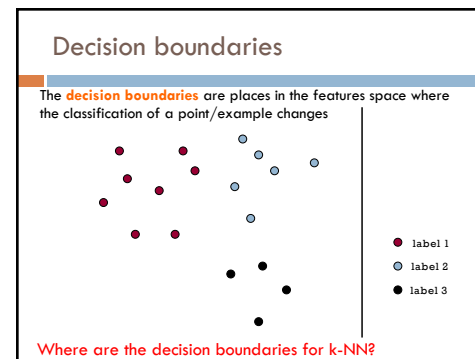
13



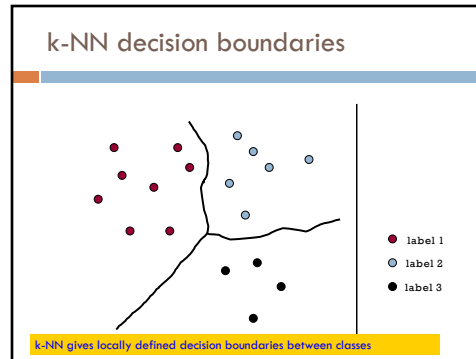
14



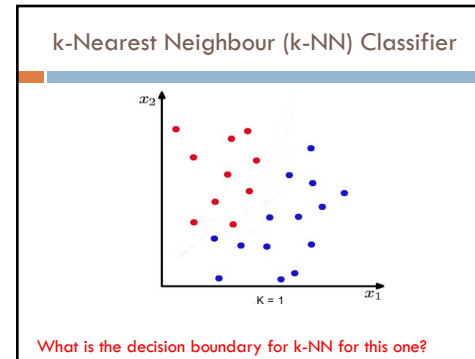
15



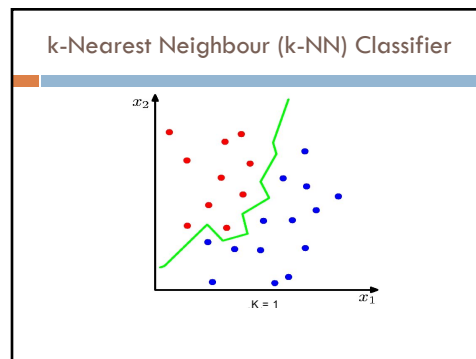
16



17



18



19

Machine learning models

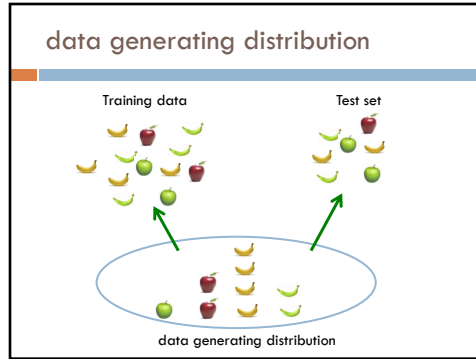
Some machine learning approaches make strong assumptions about the data

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

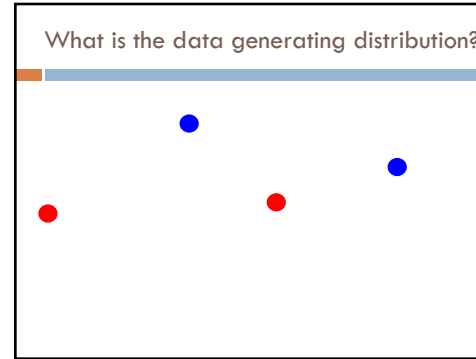
Other approaches don't make many assumptions about the data

- 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

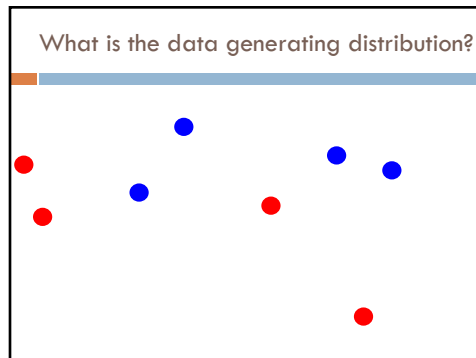
20



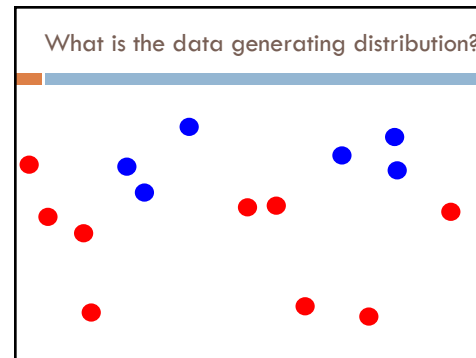
21



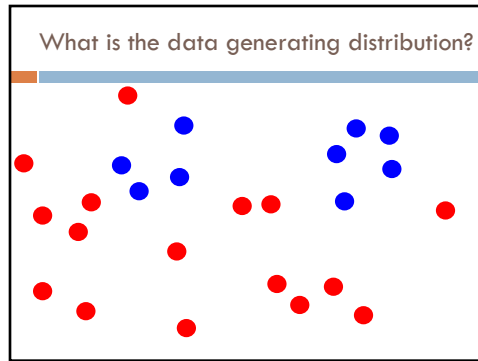
22



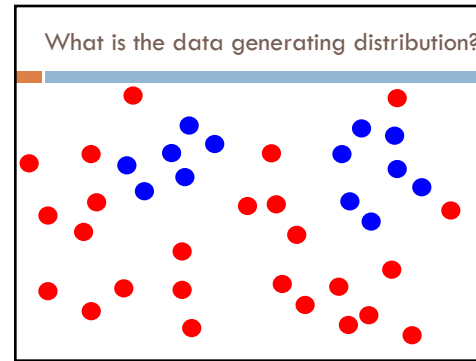
23



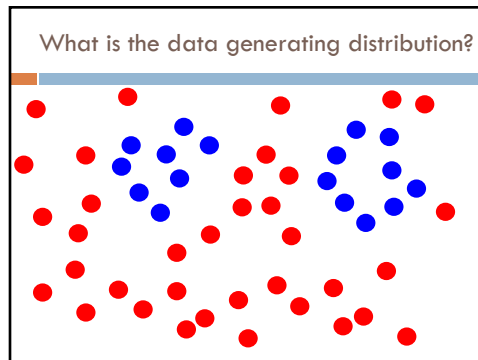
24



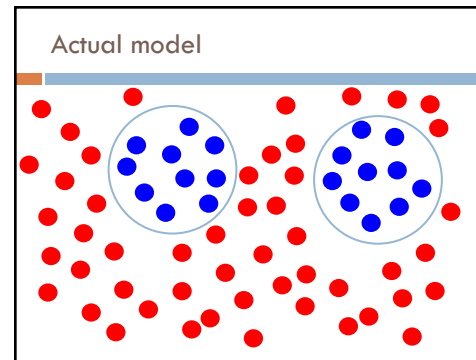
25



26



27



28

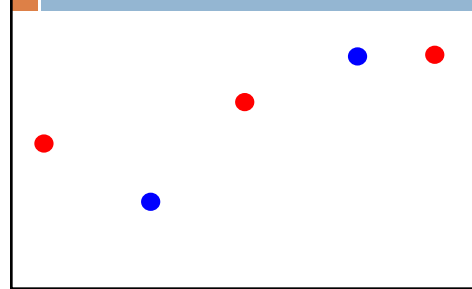
Model assumptions

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

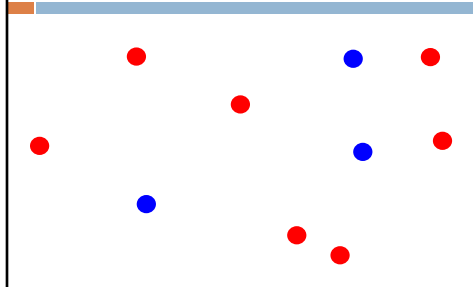
29

What is the data generating distribution?



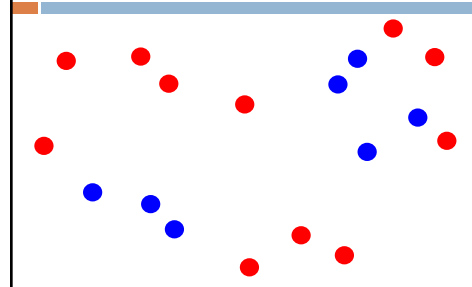
30

What is the data generating distribution?

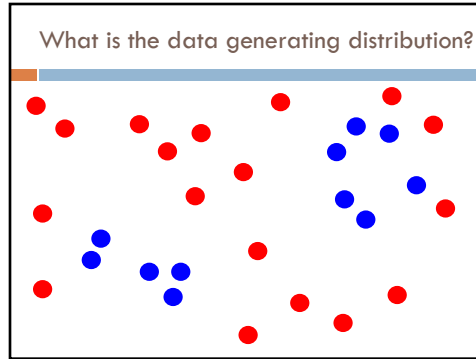


31

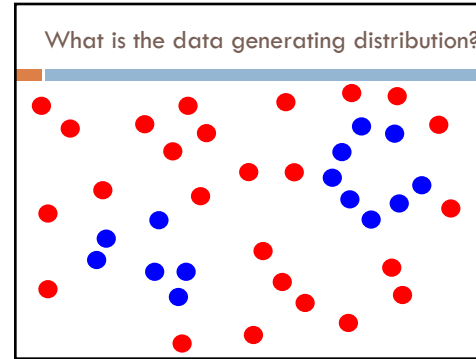
What is the data generating distribution?



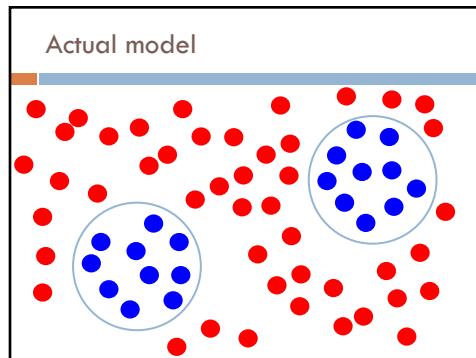
32



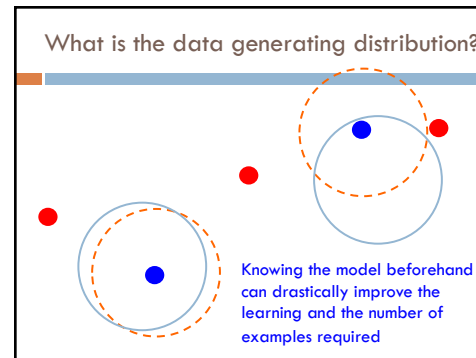
33



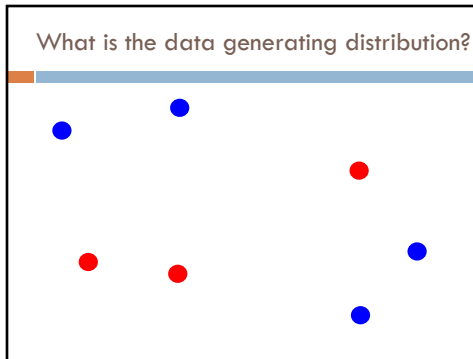
34



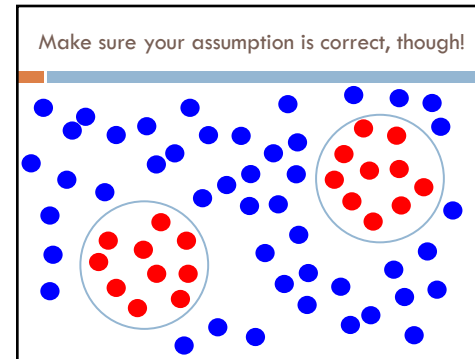
35



36



37



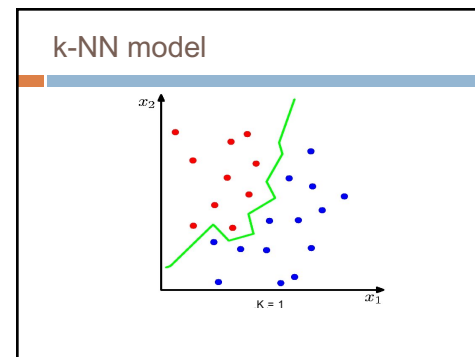
38

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



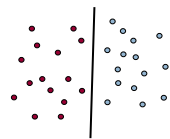
40

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?

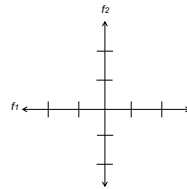
What defines a hyperplane?

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

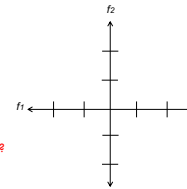
Defining a line

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

$$0 = w_1 f_1 + w_2 f_2$$

$$0 = 1f_1 + 2f_2$$

What does this line look like?



44

Defining a line

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

$$0 = w_1 f_1 + w_2 f_2$$

$$0 = 1f_1 + 2f_2$$

-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:

$$0 = w_1 f_1 + w_2 f_2$$

$$0 = 1f_1 + 2f_2$$

-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:

$$0 = w_1 f_1 + w_2 f_2$$

$$0 = 1f_1 + 2f_2$$

$w = (1, 2)$

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?

$$0 = 1f_1 + 2f_2$$

48

Classifying with a line

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

$$0 = 1f_1 + 2f_2$$

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

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

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:

$$0 = w_1f_1 + w_2f_2$$

$$0 = 1f_1 + 2f_2$$

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:

$$a = w_1f_1 + w_2f_2$$

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

-2
-1
0
1
2

51

Defining a line

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

$$a = w_1f_1 + w_2f_2$$

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

-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 defined 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, \dots, f_m \rightarrow 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

An aside: dot product

$$b + \sum_{j=1}^m w_j f_j = b + w \cdot f$$

$w = w_1, w_2, w_3, \dots, w_m$

$f = f_1, f_2, f_3, \dots, f_m$

55

Learning a linear model

Geometrically, we know what a linear model represents

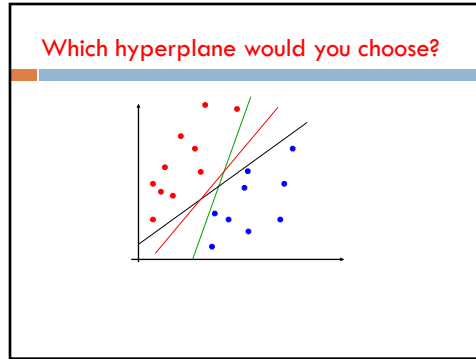
Given a linear model (i.e. a set of weights and b) we can classify examples

Training Data \rightarrow learn \rightarrow classifier

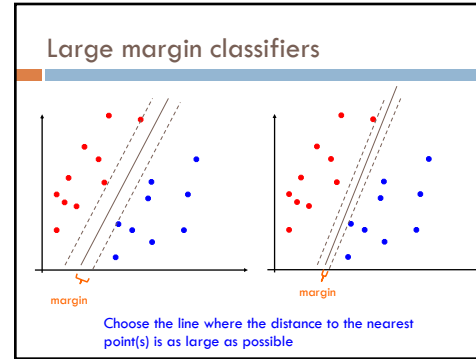
(data with labels)

How do we learn a linear model?

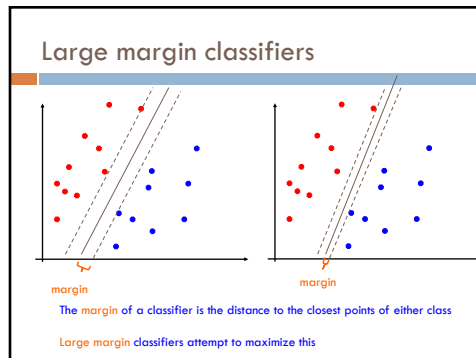
56



57



58



59

Large margin classifier setup

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

Setup as a **constrained optimization problem**:

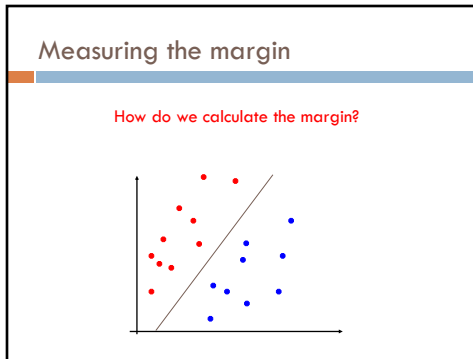
$$\max_{w,b} \text{margin}(w,b)$$

subject to:

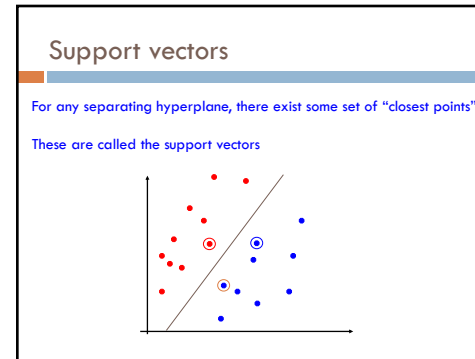
$$y_i(w \cdot x_i + b) > 0 \quad \forall i \quad \text{what does this say?}$$

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

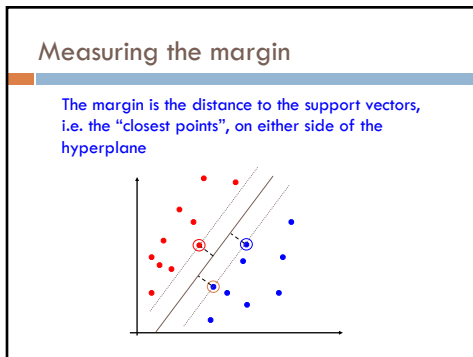
60



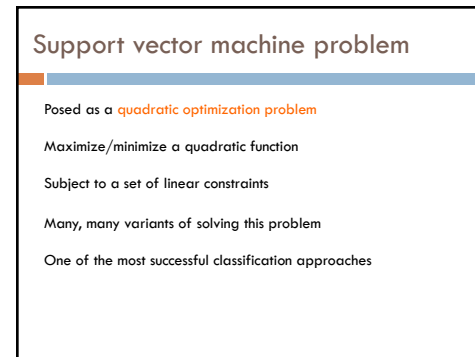
61



62



63




64

Support vector machines

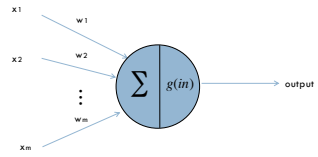
One of the most successful (if not the most successful) classification approaches:

decision tree	About 4,080,000 results (0.07 sec)
Support vector machine (SVM)	About 4,130,000 results (0.10 sec) About 3,080,000 results (0.06 sec)
k nearest neighbor	About 3,460,000 results (0.05 sec)
Naive Bayes	About 596,000 results (0.05 sec)



65

NN decision boundary

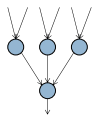


What does the decision boundary of a perceptron look like?

Line (linear set of weights)

66

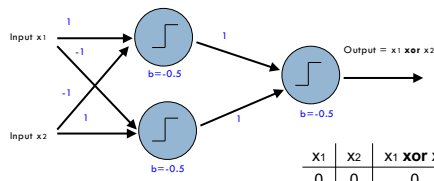
NN decision boundary



What does the decision boundary of a 2-layer network look like?
Is it linear?
What types of things can and can't it model?

67

XOR



Output = $x_1 \text{ XOR } x_2$

X1	X2	X1 XOR X2
0	0	0
0	1	1
1	0	1
1	1	0

$$\text{output} = \begin{cases} 1 & \text{if } \sum w_i x_i + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

68

What does the decision boundary look like?

X1	X2	X1 XOR X2
0	0	0
0	1	1
1	0	1
1	1	0

69

What does the decision boundary look like?

What does this perceptron's decision boundary look like?

X1	X2	X1 XOR X2
0	0	0
0	1	1
1	0	1
1	1	0

70

NN decision boundary

(without the bias)

71

NN decision boundary

72

What does the decision boundary look like?

Input x_1 and Input x_2 feed into two hidden nodes, each with bias $b = -0.5$. The hidden nodes feed into an output node with bias $b = -0.5$. The output is $x_1 \text{ XOR } x_2$.

x_1	x_2	$x_1 \text{ XOR } x_2$
0	0	0
0	1	1
1	0	1
1	1	0

What does this perceptron's decision boundary look like?

73

NN decision boundary

Input x_1 and Input x_2 feed into a single neuron with bias $b = -0.5$. The graph shows a coordinate system with x_1 and x_2 axes. A red line represents the decision boundary $x_2 = x_1 + 1$. A green vector points to the point $(1, -1)$.

(without the bias)

74

NN decision boundary

Input x_1 and Input x_2 feed into a single neuron with bias $b = -0.5$. The graph shows a coordinate system with x_1 and x_2 axes. A red line represents the decision boundary $x_2 = x_1 + 1$. A green vector points to the point $(1, -1)$.

75

What does the decision boundary look like?

Input x_1 and Input x_2 feed into two hidden nodes, each with bias $b = -0.5$. The hidden nodes feed into an output node with bias $b = -0.5$. The output is $x_1 \text{ XOR } x_2$.

x_1	x_2	$x_1 \text{ XOR } x_2$
0	0	0
0	1	1
1	0	1
1	1	0

What operation does this perceptron perform on the result?

76

Fill in the truth table

out1	out2	
0	0	?
0	1	?
1	0	?
1	1	?

77

OR

out1	out2	
0	0	0
0	1	1
1	0	1
1	1	1

78

What does the decision boundary look like?

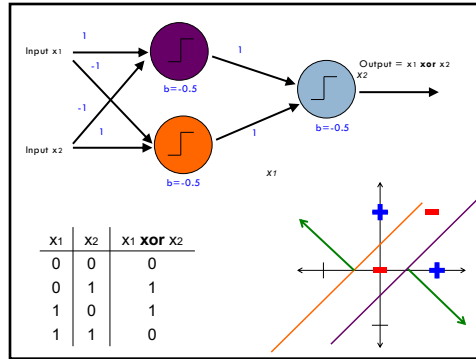
X1	X2	X1 XOR X2
0	0	0
0	1	1
1	0	1
1	1	0

79

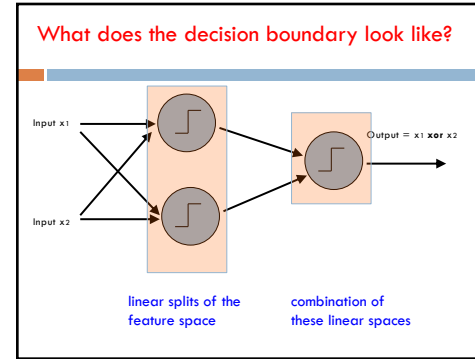
Output = $x_1 \text{ XOR } x_2$

If either predicts positive, example is positive

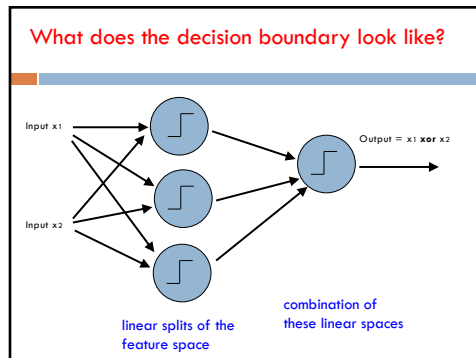
80



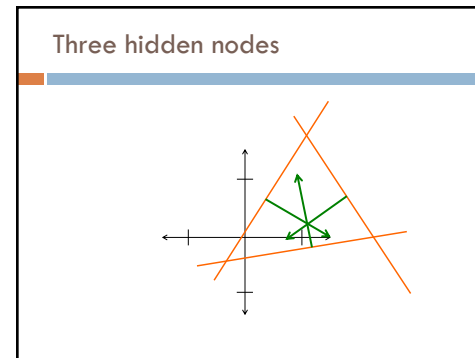
81



82



83



84

NN decision boundaries

Theorem 9 (Two-Layer Networks are Universal Function Approximators). *Let F be a continuous function on a bounded subset of D -dimensional space. Then there exists a two-layer neural network \hat{F} with a finite number of hidden units that approximate F arbitrarily well. Namely, for all x in the domain of F , $|F(x) - \hat{F}(x)| < \epsilon$.*

Put simply: **two-layer networks can approximate any function**

85

Compiling Java programs

86

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

93