1. Consider the following four vectors:

(i) $\boldsymbol{x}^1 = \begin{bmatrix} -1 \\ 0.5 \\ 2 \\ 0 \end{bmatrix}$ (ii) $\boldsymbol{x}^2 = \begin{bmatrix} -0.5 \\ 0 \\ 1 \\ -2 \end{bmatrix}$ (iii) $\boldsymbol{x}^3 = \begin{bmatrix} 0 \\ 1 \\ -1 \\ 0.5 \end{bmatrix}$

(a) What is the magnitude of each vector?

(i) $\sqrt{{\boldsymbol{x}^1}^T \boldsymbol{x}^1} = \sqrt{-1^2 + 0.5^2 + 2^2 + 0^2} = \sqrt{1 + 0.25 + 4 + 0} = \sqrt{5.25} \approx \mathbf{2.3}$

(ii) $\sqrt{{\boldsymbol{x}^2}^T \boldsymbol{x}^2} = \sqrt{-0.5^2 + 0^2 + 1^2 + -2^2} = \sqrt{0.25 + 0 + 1 + 4} = \sqrt{5.25} \approx \mathbf{2.3}$

(iii) $\sqrt{{\boldsymbol{x}^3}^T \boldsymbol{x}^3} = \sqrt{0^2 + 1^2 + -1^2 + 0.5^2} = \sqrt{0 + 1 + 1 + 0.25} = \sqrt{2.25} = \mathbf{1.5}$

(b) What is the result of each dot product below?

$\mathbf{x}^{1\,\mathsf{T}}\,\mathbf{x}^2$

$\begin{bmatrix} -1 \\ 0.5 \\ 2 \\ 0 \end{bmatrix}^T \begin{bmatrix} -0.5 \\ 0 \\ 1 \\ -2 \end{bmatrix} = 0.5 + 0 + 2 + 0 = \mathbf{2.5}$

$\mathbf{x}^{3\,T}\,\mathbf{x}^{2}$

$$\begin{bmatrix} 0 \\ 1 \\ -1 \\ 0.5 \end{bmatrix}^{T} \begin{bmatrix} -0.5 \\ 0 \\ 1 \\ -2 \end{bmatrix} = 0 + 0 + (-1) + (-1) = \mathbf{-2}$$

$\mathbf{x}^{1\,T}\,\mathbf{x}^{3}$

$$\begin{bmatrix} -1 \\ 0.5 \\ 2 \\ 0 \end{bmatrix}^{T} \begin{bmatrix} 0 \\ 1 \\ -1 \\ 0.5 \end{bmatrix} = 0 + 0.5 - 2 + 0 = \mathbf{-1.5}$$

2. We wish to use a Bayesian classifier to distinguish between two classes of birds: y^i=D (for Duck) or y^i=G (for Goose). Each data point contains 5 features, measuring: motion speed, weight, size, number of daily hours-of-sleep, and typical depth-of-dive into water.

Presume we use a Gaussian Naïve Bayes classifier – we assume each $P(x^i_j \mid y^i)$ is Gaussian.

(a) How will we calculate the posterior probability: $P(y^i \mid x^i_{speed}, x^i_{weight}, x^i_{size}, x^i_{sleepHours}, x^i_{diveDepth})$?
(What other probabilities will we use for this calculation?)

To estimate the likelihood for each feature separately and then multiply the prior.
$P(y^i \mid x^i_{speed}, x^i_{weight}, x^i_{size}, x^i_{sleepHours}, x^i_{diveDepth}) \approx$
$P(x^i_{speed} \mid y^i)\, P(x^i_{weight} \mid y^i)\, P(x^i_{size} \mid y^i)\, P(x^i_{sleepHours} \mid y^i)\, P(x^i_{diveDepth} \mid y^i)\, P(y^i)$

Note: the actual complete posterior probability is:
$P(y \mid x_{speed}, x_{weight}, \ldots x_{diveDepth}) = P(x_{speed} \mid y) \ldots P(x_{diveDep} \mid y) P(y) \,/\, P(x_{speed}, x_{weight}, \ldots x_{diveDep})$

However, for the purposes of the Naïve Bayes classifier, we can estimate the posterior and avoid dividing by $P(x_{speed}, x_{weight}, \ldots x_{diveDep})$ because the x values for a single classification are constant. In classification we keep the feature values $\mathbf{x}^i$ constant and test different potential classes y^i.

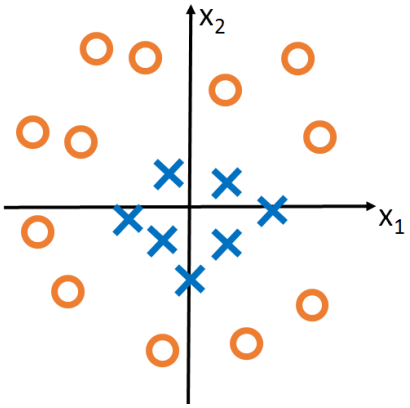
(b) How many parameters will we learn under the Naïve Bayes assumption?
2 parameters (mean and variance) for each feature likelihood, 2 classes, plus prior for y^i=Duck or (1-y^i=Duck) for y^i=Goose: 5x2x2+1 = **21**

(c) Let us now assume we will use a logistic classifier instead on the same data set. How many parameters must we learn to determine the separating hyperplane?

5 dimensions: we learn 5+1 = **6** parameters

3. For each example below, which of the following mapping functions will make these points linearly separable?
Possible mapping function: $\varphi_1 = ([x_1, x_2]) \rightarrow [(x_1+x_2)^2]$
$\varphi_2 = ([x_1, x_2]) \rightarrow [\cos(x_1), \cos(2x_1), \cos(3x_1)]$ $\varphi_3 = ([x_1, x_2]) \rightarrow [|x_1|, |x_2|]$

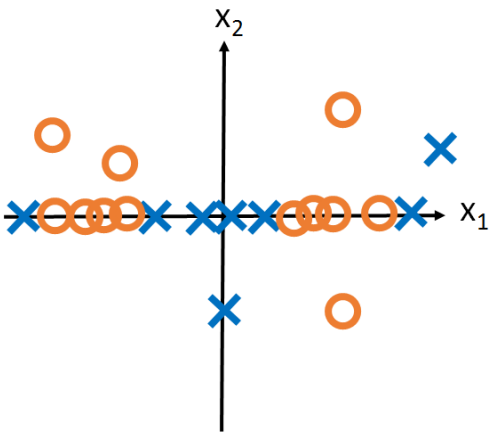
(a)



φ_3

I intended φ_1 to be correct also, but it would be correct only if $\varphi_1 = ([x_1,x_2]\text{->}[x_1^2+x_2^2])$

(b)



φ_2

4. Consider the following optimization covered in class:

$\min_{w,b} \mathbf{w}^T\mathbf{w} + C\sum_j \xi_j$

such that

$\mathbf{w}^T x^i + b \geq +1 - \xi_i$ if x^i is class +1

$\mathbf{w}^T x^i + b \leq -1 + \xi_i$ if x^i is class -1

(a) Which term(s) in the optimization is/are used to permit limited classification errors?

C and ξ_j

(b) Which term(s) in the optimization is/are used to maximize the margin?

$\mathbf{w}^T\mathbf{w}$

(c) Which term(s) in the optimization is/are used to encourage proper classification?

$\mathbf{w}^T\mathbf{x}^i + b \geq +1$ and $\mathbf{w}^T\mathbf{x}^i + b \leq -1$

5. Consider each set of support vector and find the resulting **w**

(a) $\boldsymbol{x^1} = \begin{bmatrix} 2 \\ 1 \\ -2 \end{bmatrix}$, $y^1 = +1$, $\alpha^1 = 0.5$ $\quad$ $\boldsymbol{x^2} = \begin{bmatrix} -3 \\ 3 \\ 0 \end{bmatrix}$, $y^2 = -1$, $\alpha^2 = 1$

$\boldsymbol{x^3} = \begin{bmatrix} 1 \\ 4 \\ -4 \end{bmatrix}$, $y^3 = +1$, $\alpha^3 = 0.5$

$$\mathbf{w} = \sum_i \alpha^i y^i \mathbf{x}^i \qquad \boldsymbol{w^1} = 0.5 \times \begin{bmatrix} 2 \\ 1 \\ -2 \end{bmatrix} - 1 \times \begin{bmatrix} -3 \\ 3 \\ 0 \end{bmatrix} + 0.5 \times \begin{bmatrix} 1 \\ 4 \\ -4 \end{bmatrix} = \begin{bmatrix} 1+3+0.5 \\ 0.5-3+2 \\ -1+0-2 \end{bmatrix} = \begin{bmatrix} \mathbf{4.5} \\ \mathbf{-0.5} \\ \mathbf{-3} \end{bmatrix}$$

(b) $\boldsymbol{x^1} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$, $y^1 = +1$, $\alpha^1 = 1.0$ $\quad$ $\boldsymbol{x^2} = \begin{bmatrix} 0 \\ -3 \\ -1 \end{bmatrix}$, $y^2 = -1$, $\alpha^2 = 0.7$

$\boldsymbol{x^3} = \begin{bmatrix} 2 \\ 0 \\ 2 \end{bmatrix}$, $y^3 = +1$, $\alpha^3 = 0.5$ $\quad$ $\boldsymbol{x^4} = \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}$, $y^4 = -1$, $\alpha^4 = 0.8$

$$\mathbf{w} = \sum_i \alpha^i y^i \mathbf{x}^i \qquad \boldsymbol{w^2} = 1 \times \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} - 0.7 \times \begin{bmatrix} 0 \\ -3 \\ -1 \end{bmatrix} + 0.5 \times \begin{bmatrix} 2 \\ 0 \\ 2 \end{bmatrix} - 0.8 \times \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 1+0+1-0.8 \\ 1+2.1+0+0.8 \\ 0+0.7+1+0 \end{bmatrix} = \begin{bmatrix} \mathbf{1.2} \\ \mathbf{3.9} \\ \mathbf{1.7} \end{bmatrix}$$

(c) $\boldsymbol{x^1} = \begin{bmatrix} -3 \\ -1 \\ 4 \end{bmatrix}$, $y^1 = +1$, $\alpha^1 = 1.0$ $\quad$ $\boldsymbol{x^2} = \begin{bmatrix} 2 \\ 3 \\ -1 \end{bmatrix}$, $y^2 = -1$, $\alpha^2 = 0.7$

$\boldsymbol{x^3} = \begin{bmatrix} -4 \\ -2 \\ 1 \end{bmatrix}$, $y^3 = +1$, $\alpha^3 = 0.5$ $\quad$ $\boldsymbol{x^4} = \begin{bmatrix} 3 \\ 1 \\ 0 \end{bmatrix}$, $y^4 = -1$, $\alpha^4 = 0.8$

$$\mathbf{w} = \sum_i \alpha^i y^i \mathbf{x}^i \qquad \boldsymbol{w^3} = 1 \times \begin{bmatrix} -3 \\ -1 \\ 4 \end{bmatrix} - 0.7 \times \begin{bmatrix} 2 \\ 3 \\ -1 \end{bmatrix} + 0.5 \times \begin{bmatrix} -4 \\ -2 \\ 1 \end{bmatrix} - 0.8 \times \begin{bmatrix} 3 \\ 1 \\ 0 \end{bmatrix}$$
$$= \begin{bmatrix} -3 - 1.4 - 2 - 2.4 \\ -1 - 2.1 - 1 - 0.8 \\ 4 + 0.7 + 0.5 + 0 \end{bmatrix} = \begin{bmatrix} \mathbf{-8.8} \\ \mathbf{-4.9} \\ \mathbf{5.2} \end{bmatrix}$$

6. List two useful applications for logarithms in Machine Learning (they can be either practical engineering uses or mathematical derivation uses).

Application 1: using logarithms can make differentiation easier for derivation of learning rules.
Application 2: using logarithms results in multiplied probabilities not being rounded to 0, e.g., probability of 0.000000004 would be rounded to 0 by a computer but log(prob)=-8.4 , won't be rounded to 0.

7. For the following functions, set the derivative with respect to h to 0:

(a)

f(h) = $\frac{h^2-3}{5x^3}$ (Presume $x \neq 0$)

$\frac{2h}{5x^3} = 0$ -> **h=0**

(b)

f(h) = $\prod_i 5h^{2i}e^{-3h}$

Take log, then apply derivative

FIXED OCT 17, 7:45pm

log(f(h)) = $\sum_i(\log 5 + 2i \log h - 3h)$ -> $\sum_i\left(\left(\frac{2i}{h}\right) - 3\right) = 0$

$\sum_i 2i - h\sum_i 3 = 0$

$\sum_i \frac{\mathbf{2i}}{\mathbf{3}} = \mathbf{h}$

8. Compute AIC given the following log-likelihoods and number of parameters
AIC: log(L) - k

(a) log(L)=-42 # params= 12
-42-12 = **-54**

(b) log(L)=-44 # params= 9
-44-9 = **-53**

(c) log(L)=-33 # params=10
-33-10 = **-43**

9. Let us define a logistic regression classifier with initial weight w^0=[1.2 -2.3 0.5]. In this question we will not worry about b, though we normally should.

Let us begin by optimizing for "maximum likelihood", i.e., assuming no prior constraints. Also, assume $\varepsilon = 0.01$

$$\boldsymbol{w^{new}} \leftarrow \boldsymbol{w^{old}} + \varepsilon \boldsymbol{x_j^i}\left(y^i - g(\boldsymbol{w^T x^i})\right)$$

Note: showing your work allows you to get partial credit if you make a mistake!
You will receive the majority of points by following the correct process.

What will be the update to w^0 if we see the data point:
x^1=[3 2 1], y^1=1

Compute $\boldsymbol{w^T x^1} = [1.2 \quad -2.3 \quad 0.5]\begin{bmatrix}3\\2\\1\end{bmatrix} = 3.6 - 4.6 + 0.5 = -0.5$

Compute $g(\boldsymbol{w}^T\boldsymbol{x}^1) = g(-0.5) = \frac{1}{1+e^{+0.5}} \approx 0.38$

$$\boldsymbol{w}_1^{new} \leftarrow \boldsymbol{w}_1^{old} + \varepsilon \boldsymbol{x}_1^i \left(y^i - g(\boldsymbol{w}^T\boldsymbol{x}^i)\right) = 1.2 + 0.01 \times 3 \times (1 - 0.38) = 1.2 + 0.01 \times 3 \times 0.62$$
$$= 1.2 + 0.0186 \approx 1.22$$

$$\boldsymbol{w}_2^{new} \leftarrow \boldsymbol{w}_2^{old} + \varepsilon \boldsymbol{x}_2^i \left(y^i - g(\boldsymbol{w}^T\boldsymbol{x}^i)\right) = -2.3 + 0.01 \times 2 \times (1 - 0.38)$$
$$= -2.3 + 0.01 \times 2 \times 0.62 = -2.3 + 0.0122 \approx -2.29$$

$$\boldsymbol{w}_3^{new} \leftarrow \boldsymbol{w}_3^{old} + \varepsilon \boldsymbol{x}_3^i \left(y^i - g(\boldsymbol{w}^T\boldsymbol{x}^i)\right) = 0.5 + 0.01 \times 1 \times (1 - 0.38) = 0.5 + 0.01 \times 1 \times 0.62$$
$$= 0.5 + 0.0062 \approx 0.51$$

Final answer: $\boldsymbol{w}^{new} = \begin{bmatrix} \mathbf{1.22} \\ \mathbf{-2.29} \\ \mathbf{0.51} \end{bmatrix}$

What will be the update to w^0 if we see the data point:
x^1=[3 2 1], y^1=0

$$\boldsymbol{w^{new} \leftarrow w^{old} + \varepsilon x_j^i \left(y^i - g(w^T x^i)\right)}$$

Compute $\boldsymbol{w^T x^1} = [1.2 \quad -2.3 \quad 0.5]\begin{bmatrix}3\\2\\1\end{bmatrix} = 3.6 - 4.6 + 0.5 = -0.5$

Compute $g(\boldsymbol{w^T x^1}) = g(-0.5) = \frac{1}{1+e^{+0.5}} \approx 0.38$

$\boldsymbol{w_1^{new} \leftarrow w_1^{old} + \varepsilon x_1^i \left(y^i - g(w^T x^i)\right)} = 1.2 + 0.01 \times 3 \times (0 - 0.38) = 1.2 - 0.01 \times 3 \times 0.38$
$= 1.2 - 0.0114 \approx 1.19$

$\boldsymbol{w_2^{new} \leftarrow w_2^{old} + \varepsilon x_2^i \left(y^i - g(w^T x^i)\right)} = -2.3 + 0.01 \times 2 \times (0 - 0.38)$
$= -2.3 - 0.01 \times 2 \times 0.38 = -2.3 - 0.0075 \approx -2.31$

$\boldsymbol{w_3^{new} \leftarrow w_3^{old} + \varepsilon x_3^i \left(y^i - g(w^T x^i)\right)} = 0.5 + 0.01 \times 1 \times (0 - 0.38) = 0.5 - 0.01 \times 1 \times 0.38$
$= 0.5 - 0.0038 \approx 0.5$

Final answer: $\boldsymbol{w^{new}} = \begin{bmatrix}\mathbf{1.19}\\\mathbf{-2.31}\\\mathbf{0.5}\end{bmatrix}$

What will be the update to w^0 if we see the data point:
x^1=[1 2 3], y^1=1

$$\boldsymbol{w^{new} \leftarrow w^{old} + \varepsilon x_j^i \left(y^i - g(w^T x^i)\right)}$$

Compute $\boldsymbol{w^T x^1} = [1.2 \quad -2.3 \quad 0.5]\begin{bmatrix}1\\2\\3\end{bmatrix} = 1.2 - 4.6 + 1.5 = -1.9$

Compute $g(\boldsymbol{w^T x^1}) = g(-1.9) = \frac{1}{1+e^{+1.9}} \approx 0.13$

$\boldsymbol{w_1^{new} \leftarrow w_1^{old} + \varepsilon x_1^i \left(y^i - g(w^T x^i)\right)} = 1.2 + 0.01 \times 1 \times (1 - 0.13) = 1.2 + 0.01 \times 1 \times 0.87$
$= 1.2 + 0.0087 \approx 1.21$

$\boldsymbol{w_2^{new} \leftarrow w_2^{old} + \varepsilon x_2^i \left(y^i - g(w^T x^i)\right)} = -2.3 + 0.01 \times 2 \times (1 - 0.13)$
$= -2.3 + 0.01 \times 2 \times 0.87 = -2.3 + 0.0174 \approx -2.28$

$\boldsymbol{w_3^{new} \leftarrow w_3^{old} + \varepsilon x_3^i \left(y^i - g(w^T x^i)\right)} = 0.5 + 0.01 \times 3 \times (1 - 0.13) = 0.5 + 0.01 \times 3 \times 0.87$
$= 0.5 + 0.0261 \approx 0.53$

Final answer: $\boldsymbol{w^{new}} = \begin{bmatrix}\mathbf{1.21}\\\mathbf{-2.28}\\\mathbf{0.53}\end{bmatrix}$

What will be a potential effect of decreasing ε.

Decreasing ε will decrease the amount each element of the weight matrix is updated for each new data point.

How will we change the optimization process if we include L2 regularization?
Two possible interpretations to this question. Based on my question phrasing, either interpretation is valid:
The technical answer: **We add $-\frac{w_j}{\lambda}$ during weight updates**
The "big picture" answer: **Prevents magnitude of weights in any feature dimension from growing too large.**

We are trying to determine the probability that Microsoft stock will go up (M=yes) based on the following yes/no features:
Recent increase in laptop sales (L=yes)
Recent increase in silicon price (S=yes)
Recent decrease in rain in Seattle, Washington (R=yes)

For the rest of this section: Let us assume R is independent of L and S, given M. However, Assume L and S are NOT independent given M.

Write the expression for the likelihood of L, S, and R, given M. Simplify the expression by including the fewest number of variables possible in each probability term.
Complex likelihood: **P(L,S,R|M)**
Simplified likelihood: **P(L,S|M) P(R|M)**

Let us say we have the following data from past days. Y means yes, N means no and ? means "data not available." Whenever data is unavailable for a given feature, it is not used in estimating probabilities relating to that feature. For example, the first data point:

	L	S	R	M
Day 1:	Y	?	N	Y

can be used in the estimation of P(L=yes) and P(M=yes | R=no), but it cannot be used in the estimation of P(L=yes,S=yes|M=yes)

Data:

	L	S	R	M
Day 1:	Y	?	N	Y
Day 2:	Y	Y	N	N
Day 3:	?	Y	N	Y
Day 4:	?	?	Y	N
Day 5:	N	N	?	Y
Day 6:	N	?	?	Y
Day 7:	Y	?	Y	Y
Day 8:	?	N	Y	Y

Which of the probabilities below have Maximum Likelihood Estimate of 0? What are the non-zero values?

P(M=yes)

Count M=yes and all available M data points

$\frac{6}{8} = \mathbf{0.75}$

P(L=yes,S=yes|M=no)

Count all data points where L=yes and S=yes and M=no,

Count all data points where data available for L and S, and where M=no

$\frac{1}{1} = \mathbf{1}$

P(L=no,S=no,R=yes|M=yes)
Due to independence of R, P(L,S,R|M) is calculated as P(L,S|M) x P(R|M)

P(R=yes|M=yes): Count number of data points where R=yes and M=yes, count number of data points for R and M where M=yes
P(R=yes|M=yes) = $\frac{2}{4} = 0.5$

P(L=no,S=no|M=yes): Count number of data points where L=no, S=no, and M=yes, count number of data points for L, S, and M where M=no
P(L=no,S=no|M=yes) = $\frac{1}{1} = 1$

P(L=no,S=no,R=yes|M=yes) = P(L=no,S=no|M=yes) x P(R=no|M=yes) = 1x0.5 = **0.5**

P(L=no,S=yes|M=yes)
P(L=no,S=yes|M=yes)=0 … no data points for L=no, S=yes, and M=no at the same time

To calculate the a posteriori probability of P(L=no,S=no|M=yes), how do we incorporate a prior belief that there is a 20% chance Microsoft stock will fall or stay the same?
$\frac{\#D(L=no \wedge S=no \wedge M=yes)+80}{\#D(M=yes)+100}$ **… i.e., add $\beta_{rise} = 80$ and $\beta_{fall\_or\_stay\_same} = 20$**

Let us consider a binary classification problem. For several objects in an online store, we wish to train a classifier to predict the label: “Does this object make people happy?” (variable H={yes,no}) We will use five features:

How big is it?
How old is it?
How expensive is it?
How familiar is it?
How natural is it?

Each feature will take on an integer value from 1 to 10, inclusive.

Assuming all features are independent, how many parameters must be learned for the classifier?

$$2 \times 5 \times (10 - 1) = 2 \times 5 \times 9 = \mathbf{90}$$

How many features do we learn if we wish to categorize each object into three different "make people happy" classes (including scores 1 – really makes people sad, 2 – neutral in affecting happiness, and 3 – really makes people happy).

$$3 \times 5 \times (10 - 1) = 3 \times 5 \times 9 = \mathbf{135}$$

Let us assume a single-feature data set in which each data point comes from one of two distributions:

For class 0: a **uniform** distribution starting x and ending x =a, $f(x) = \begin{cases} \frac{1}{a} & 0 \le x \le a \\ 0 & otherwise \end{cases}$

Class 1: a **triangle** distribution, starting at x=0 and ending at x=b:

$$h(x) = \begin{cases} \frac{2}{b}\left(1 - \frac{x}{b}\right) & 0 \le x \le b \\ 0 & otherwise \end{cases}$$

Given N data points (x^i,y^i) in a training set, what is the formula for the likelihood, given the parameters a and b? (You may express your answer in terms of f(x) and h(x).)

$$P(D|\theta) = \prod_{i=1}^{N} \boldsymbol{f(x)}^{(1-y^i)} \boldsymbol{h(x)}^{y^i}$$

Assuming all data points are positive, what is the maximum likelihood estimate for a and b?
Calculus will not help you here. You actually have to use your intuition!

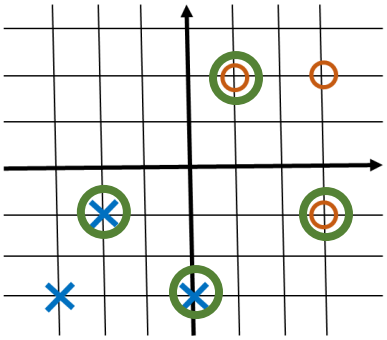
The value for a should be the maximum value of all x^i such that y^i=0, i.e., $\boldsymbol{a = \max(x^i)^{(1-y^i)}}$

This ensures 1/a is as large possible while $x^i \le a$ for all x^i in class 0.

The value for b should be the maximum value of all x^i such that y^i=1, i.e., $\boldsymbol{b = \max(x^i)^{y^i}}$

This ensures $\frac{2}{b}\left(1 - \frac{x}{b}\right)$ is as large possible while $x^i \le b$ for all x^i in class 1.

Consider points below and select four as likely support vectors for a linear separator. Draw your estimate of the separator and calculate the w value, assuming all support vectors have alpha=1.



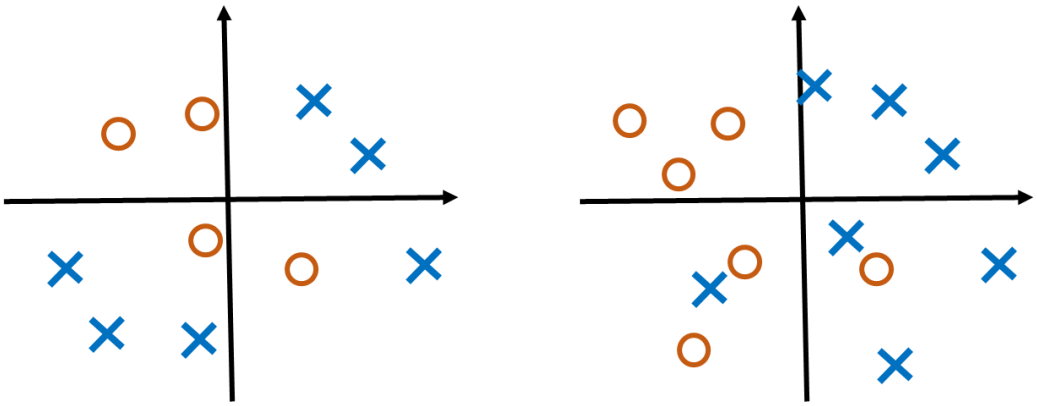
$$\boldsymbol{w} = \sum_i \alpha^i \boldsymbol{x^i} y^i = +1 \times \begin{bmatrix} 1 \\ 2 \end{bmatrix} \times 1 + 1 \times \begin{bmatrix} 3 \\ -1 \end{bmatrix} + 1 \times \begin{bmatrix} -2 \\ -1 \end{bmatrix} \times -1 + 1 \times \begin{bmatrix} 0 \\ -3 \end{bmatrix} \times -1$$

$$= \begin{bmatrix} 1+3+2+0 \\ 2-1+1+3 \end{bmatrix} = \begin{bmatrix} \mathbf{6} \\ \mathbf{5} \end{bmatrix}$$

What is the purpose of a slack variable?

The slack variable is used to control the degree of penalty given for mis-classified data points while learning a linear separator in an SVM.

Suggest the two classification methods from the list below most fitting the following data:



Bayes classifier
Logistic classifier
Linear SVM
Kernel SVM (i.e., SVM with dimension mapping function)
SVM with slack variables

Left hand plot:
Bayes classifier
Kernel SVM

Right hand plot:
Bayes classifier
SVM with slack variables

What is overfitting? What are potential causes?

Overfitting is caused by over-learning classifier parameters based on training data to focus on patterns specific to training data that does not generalize to outside testing data. This leads to poor classifier performance on testing sets.

Potential causes are performing too many learning iterations on the same data set, and fitting too many parameters to insufficient data.

We wish to use a Maximum Likelihood Bayesian classifier to determine whether we are at a fruit stand (variable F) based on the presence of bananas (variable B), milk (variable M), and cash registers (variable C). We assume B, M, and C are all independent of one another given the value of F. Based on training data, we have found the following probabilities:

P(B=yes|F=yes) = 0.8	P(M=yes|F=yes)=0.1	P(C=yes|F=yes)=0.7
P(B=yes|F=no)=0.2	P(M=yes|N=no)=0.5	P(C=yes|F=no)=0.6

We observe cash registers, but no bananas or milk. Does the Maximum Likelihood classifier conclude we are in a fruit stand?

Compute and compare P(C=yes,B=no,M=no|F=yes) and P(C=yes,B=no,M=no|F=no)

P(C,B,M|F)=P(C|F)P(B|F)P(M|F)

P(C=yes,B=no,M=no|F=yes)= 0.7 x (1-0.8) x (1-0.1) = 0.7 x 0.2 x 0.9 = 0.126

P(C=yes,B=no,M=no|F=no)= 0.6 x (1-0.2) x (1-0.5) = 0.6 x 0.8 x 0.5 = 0.24

P(C=yes,B=no,M=no|**F=no**)> P(C=yes,B=no,M=no|**F=yes**) **We ARE NOT at a fruit stand**

We observe bananas, but no milk or cash registers. Does the Maximum Likelihood classifier conclude we are in a fruit stand?

Compute and compare P(B=yes,M=no,C=no|F=yes) and P(B=yes,M=no,C=no|F=no)

P(B,M,C|F)= P(B|F)P(M|F)P(C|F)

P(B=yes,M=no,C=no|F=yes)= 0.8x (1-0.1) x (1-0.7) = 0.8 x 0.9 x 0.3 = 0.216

P(B=yes,M=no,C=no|F=no)= 0.2x(1-0.5) x (1-0.6) = 0.2 x 0.5 x 0.4 = 0.04

P(B=yes,M=no,C=no|**F=yes**)> P(B=yes,M=no, C=no|**F=no**) **We ARE at a fruit stand**

Consider the principal components:

$$\boldsymbol{u}^1 = \begin{bmatrix} 0.5 \\ 0 \\ -0.4 \\ 0.8 \end{bmatrix}, \boldsymbol{u}^2 = \begin{bmatrix} 0 \\ -0.7 \\ 0.3 \\ 0.6 \end{bmatrix}, \boldsymbol{u}^3 = \begin{bmatrix} 0.6 \\ 0.3 \\ 0.7 \\ 0 \end{bmatrix}$$

Find the weights for the first two components z_1^1 and z_2^1 for the data point $\boldsymbol{x}^1 = \begin{bmatrix} 1 \\ 1.5 \\ 4 \\ -2 \end{bmatrix}$

$\boldsymbol{z_1^1 = -2.7}$ $\boldsymbol{z_2^1 = -1.1}$

Compute the sum squared error for the reconstruction of $\boldsymbol{x^1}$ using the first two principal components.

$$\widetilde{x^1} = -2.7 \begin{bmatrix} 0.5 \\ 0 \\ -0.4 \\ 0.8 \end{bmatrix} - 1.1 \begin{bmatrix} 0 \\ -0.7 \\ 0.3 \\ 0.6 \end{bmatrix} = \begin{bmatrix} -1.35 \\ 0.77 \\ 0.75 \\ -2.82 \end{bmatrix}$$

Error: $\sum_j \left(x_j^i - \widetilde{x_j^i}\right)^2 = (1 + 1.35)^2 + (1.5 - 0.77)^2 + (4 - 0.75)^2 + (-2 + 2.82)^2 = 2.35^2 + 0.73^2 + 3.25^2 + 0.82^2 =$ **17.29**

The following vectors were learned as components. Could they have been learned from ICA, PCA, or both?

$$\boldsymbol{u}^1 = \begin{bmatrix} 0.5 \\ -0.5 \\ 0 \\ 0.7 \end{bmatrix}, \boldsymbol{u}^2 = \begin{bmatrix} -0.8 \\ 0 \\ 0.4 \\ 0.4 \end{bmatrix}, \boldsymbol{u}^3 = \begin{bmatrix} 0.3 \\ 0.3 \\ -0.4 \\ -0.8 \end{bmatrix}, \boldsymbol{u}^4 = \begin{bmatrix} 0 \\ 0.2 \\ -0.4 \\ 0.9 \end{bmatrix}$$

ICA (not orthogonal)