Tutorial 3 (CS 412)

Note: Question [x.y] refers to question y of the exercises in Chapter x.

• Decision tree

Question 1. The loans department of a bank has the following past loan processing records each containing an applicant's income, credit history, debt, and the final approval decision. These records can serve as training examples to build a decision tree for a loan advisory system.

Income	Credit History	Debt	Decision
$ 0 – $ 5K	Bad	Low	Reject
$ 0 – $ 5K	Good	Low	Approve
$ 0 – $ 5K	Unknown	High	Reject
$ 0 – $ 5K	Unknown	Low	Approve
$ 0 – $ 5K	Unknown	Low	Approve
$ 0 – $ 5K	Unknown	Low	Reject
$ 5K – $ 10K	Bad	High	Reject
$ 5K – $ 10K	Good	High	Approve
$ 5K – $ 10K	Unknown	High	Approve
$ 5K – $ 10K	Unknown	Low	Approve
Over $10K	Bad	Low	Reject
Over $10K	Good	Low	Approve

Use the above training examples to construct a decision tree based on information gain. Let the total impurity (I'_m in Eq 9.8 on page 218) after splitting based on A_i be denoted as *Remainder*(A_i). Then to decide which attribute to base each split on, we need to compute *Remainder*(A_i) for each attribute A_i, and select the attribute that provides the minimal remaining information. If the impurity before splitting is I_m as in Eq 9.3 on page 216, we define the information **gain** as $I_m - I'_m$. **Note: Remainder is sufficient for picking the attribute. But we also compute the full information gain for illustration.**

Solution: $I(\frac{5}{12},\frac{7}{12}) = -\frac{5}{12}\log_2\frac{5}{12} - \frac{7}{12}\log_2\frac{7}{12} = 0.980$

Remainder(Income)

$$= \frac{6}{12}(-\frac{3}{6}\log_2\frac{3}{6} - \frac{3}{6}\log_2\frac{3}{6}) + \frac{4}{12}(-\frac{1}{4}\log_2\frac{1}{4} - \frac{3}{4}\log_2\frac{3}{4}) + \frac{2}{12}(-\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2}) = 0.937$$

Gain(Income) = 0.980 – 0.937 = 0.043

Remainder(Credit History)

$$= \frac{3}{12}(-\frac{3}{3}\log_2\frac{3}{3} - 0\log_2 0) + \frac{3}{12}(-\frac{3}{3}\log_2\frac{3}{3} - 0\log_2 0) + \frac{6}{12}(-\frac{2}{6}\log_2\frac{2}{6} - \frac{4}{6}\log_2\frac{4}{6}) = 0.459$$

Gain(Credit History) = 0.980 – 0.459 = 0.521

Remainder(Debt)

$$= \frac{8}{12}(-\frac{3}{8}\log_2\frac{3}{8} - \frac{5}{8}\log_2\frac{5}{8}) + \frac{4}{12}(-\frac{2}{4}\log_2\frac{2}{4} - \frac{2}{4}\log_2\frac{2}{4}) = 0.970$$

Gain(Debt) = 0.980 – 0.970 = 0.010

Since Credit History has the highest gain, choose it as the root, which has three values, i.e., "Bad", "Good", and "Unknown". Since all examples for "Bad" have the same classification (i.e., "Reject") and all examples for "Good" have the same classification (i.e., "Approve"), both nodes have no further subtree. For "Unknown", a subtree for the following subset of examples is to be constructed:

Since Credit History has the highest gain, choose it as the root, which has three values, i.e., "Bad", "Good", and "Unknown". Since all examples for "Bad" have the same classification (i.e., "Reject") and all examples for "Good" have the same classification (i.e., "Approve"), both nodes have no further subtree. For "Unknown", a subtree for the following subset of examples is to be constructed:

Income	Debt	Decision
$ 0 – $ 5K	High	Reject
$ 0 – $ 5K	Low	Approve
$ 0 – $ 5K	Low	Approve
$ 0 – $ 5K	Low	Reject
$ 5K – $ 10K	High	Approve
$ 5K – $ 10K	Low	Approve

$$I(\frac{2}{6},\frac{4}{6}) = -\frac{2}{6}\log_2\frac{2}{6} - \frac{4}{6}\log_2\frac{4}{6} = 0.918$$

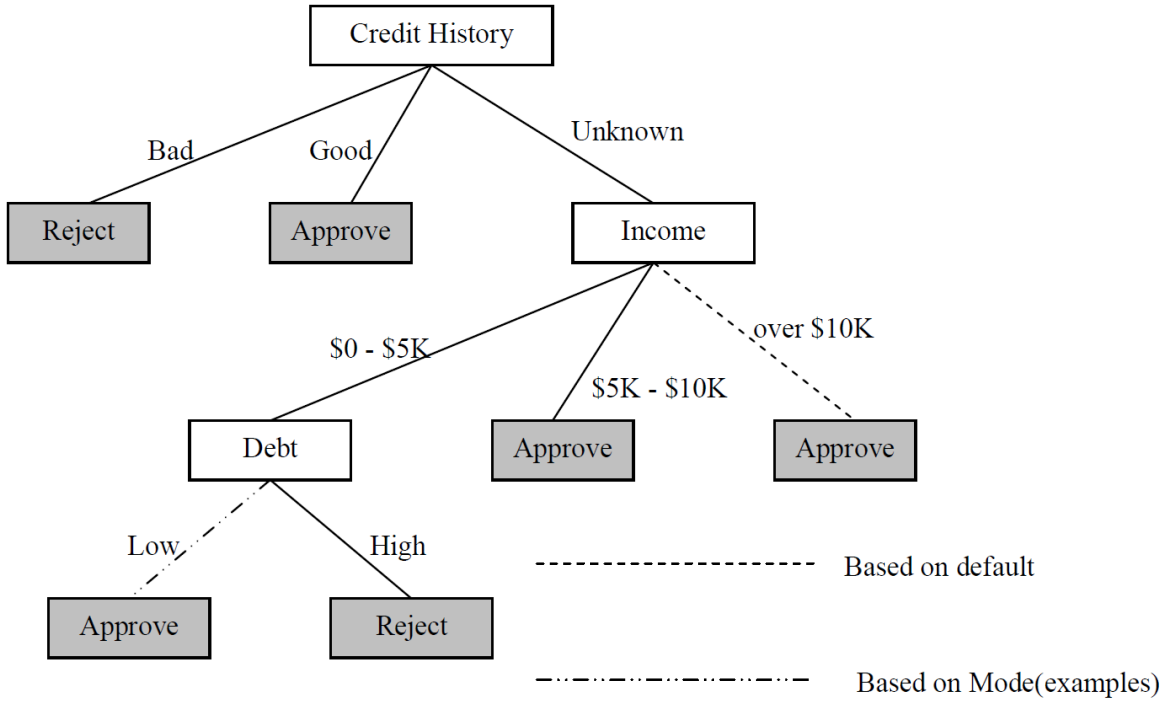
$$\text{Remainder(Income)} = \frac{4}{6}(-\frac{2}{4}\log_2\frac{2}{4} - \frac{2}{4}\log_2\frac{2}{4}) + \frac{2}{6}(-\frac{2}{2}\log_2\frac{2}{2} - 0\log_2 0) = 0.667$$

$$\text{Gain(Income)} = 0.918 - 0.667 = 0.251$$

$$\text{Remainder(Debt)} = \frac{2}{6}(-\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2}) + \frac{4}{6}(-\frac{1}{4}\log_2\frac{1}{4} - \frac{3}{4}\log_2\frac{3}{4}) = 0.874$$

$$\text{Gain(Debt)} = 0.918 - 0.874 = 0.044$$

Since Income has a higher gain than Debt, Income is chosen as the root of the subtree under Credit History=Unknown.



Question 2. Attributes with many different possible values can cause problems with the information gain. Such attributes tend to split the examples into numerous small classes or even singleton classes, thereby appearing to be highly relevant according to the gain measure. To address this problem, the gain ratio criterion selects attributes according to the ratio between their gain and their intrinsic information content – that is, the amount of information contained in the answer to the question, "What is the value of this attribute?" The gain ratio criterion therefore tries to measure how efficiently an attribute provides information on the correct classification of an example.

Mathematically, Gain Ratio is defined as:

$$GainRatio(E, A) = \frac{InformationGain(E, A)}{SplitInformation(E, A)}$$

where

$$SplitInformation(E, A) = -\sum_{k=1}^{d} \frac{|E_k|}{|E|} \log_2 \frac{|E_k|}{|E|}$$

Here E_1, …, E_d are the d subsets of examples resulting from partitioning E by the d-valued attribute A.

Consider the following Weather dataset.

Day	Temperature	Outlook	Humidity	Windy	Play Golf?
07-05	hot	sunny	high	false	no
07-06	hot	sunny	high	true	no
07-07	hot	overcast	high	false	yes
07-09	cool	rain	normal	false	yes
07-10	cool	overcast	normal	true	yes
07-12	mild	sunny	high	false	no
07-14	cool	sunny	normal	false	yes
07-15	mild	rain	normal	false	yes
07-20	mild	sunny	normal	true	yes
07-21	mild	overcast	high	true	yes
07-22	hot	overcast	normal	false	yes
07-23	mild	rain	high	true	no
07-26	cool	rain	normal	true	no
07-30	mild	rain	high	false	yes

Compute the Gain Ratio of the attributes Day, Temperature, Outlook, Humidity, and Windy.

Solution: There are 9 rows of yes and 5 rows of no. So the entropy is
$-9/14*\log_2(9/14) - 5/14*\log_2(5/14) = 0.940$.
SplitInfo(Day) = $-1/14*\log_2(1/14) * 14 = 3.807$. GainRatio(Day) = 0.940 / 3.807 = 0.246.

Outlook		Temperature	
Info:	0.693	Info:	0.911
Gain: 0.940-0.693	0.247	Gain: 0.940-0.911	0.029
Split info: info([5,4,5])	1.577	Split info: info([4,6,4])	1.557
Gain ratio: 0.247/1.577	0.157	Gain ratio: 0.029/1.557	0.019
Humidity		Windy	
Info:	0.788	Info:	0.892
Gain: 0.940-0.788	0.152	Gain: 0.940-0.892	0.048
Split info: info([7,7])	1.000	Split info: info([8,6])	0.985
Gain ratio: 0.152/1	0.152	Gain ratio: 0.048/0.985	0.049

Question 3. Which of the following are good ways to limit overfitting in the Decision Tree classifier?

(a) Prune some of the branches of an overfit tree;

(b) Stop growing the tree at nodes with a small number of examples;

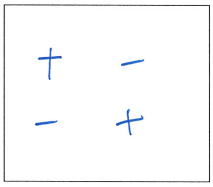
(c) Use a subset of the training data to construct the decision tree;

(d) Randomize the predictions in the tree leaves.

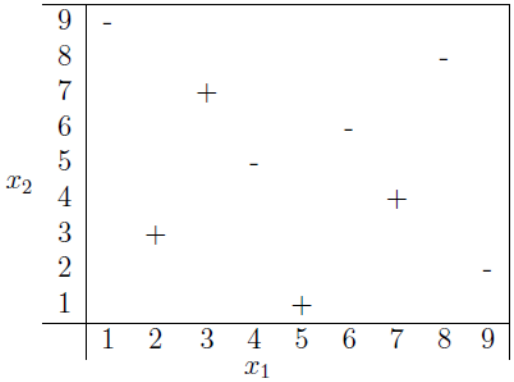
Solution: a (post-pruning) and b (pre-pruning).

Question 4. **[True or False]** Consider a decision tree with input features X_1, X_2, X_n and class label Y. Is it true that if X_2 is independent of Y, then no decision based on X_2 will appear in the decision tree?

Solution: False. This about the XOR problem. Both X1 (horizontal axis) and X2 (vertical axis) are independent on Y. But surely a decision tree will use both X1 and X2.



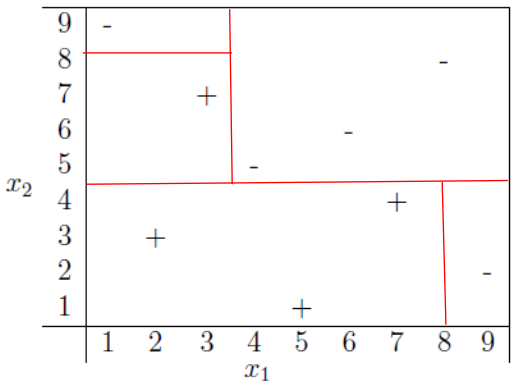
Question 5. Consider the dataset of positive (+') and negative (`-') examples:



a. Draw the decision boundaries for a decision tree that is greedily selected using (all of the following):

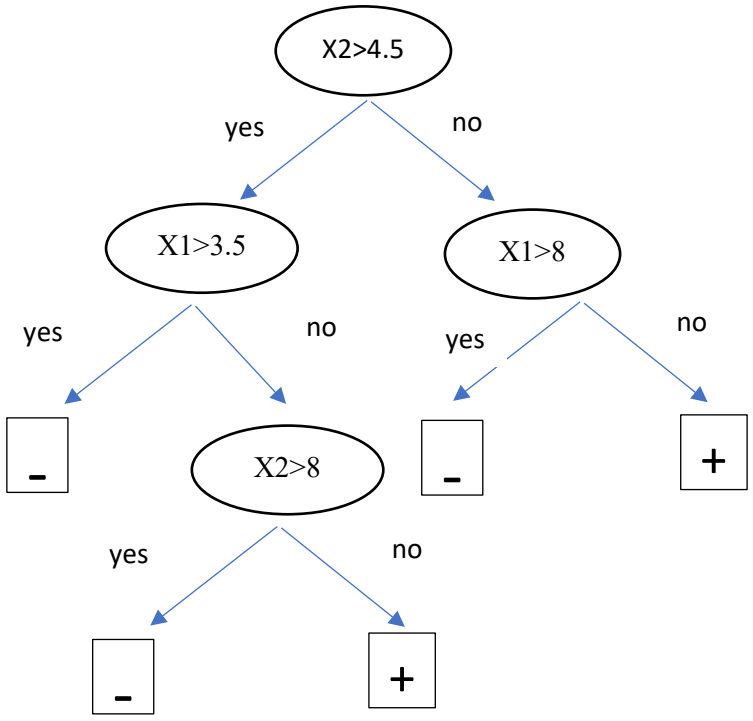
- classification accuracy as the decision criterion rule;
- ties can be broken as you wish;
- thresholds in either feature dimension for defining the decision splits; and
- continues until reaching perfect classification accuracy.

Solution:



b. For the decision boundaries in a., draw the corresponding decision tree with decisions (e.g., $x_1 < 2.5$) in each node and prediction labels for each decision tree leaf.

Solution:

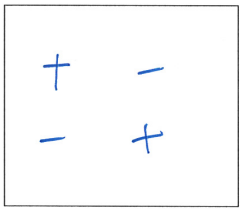


c. (beyond the scope of this course) Draw a binary dataset in the two-dimensional feature space for which greedily choosing decisions based on classification accuracy will produce bad results, while choosing decisions based on the impurity of the decision split will perform significantly better. Explain why this is the case.

Solution: See https://sebastianraschka.com/faq/docs/decisiontree-error-vs-entropy.html

Question 6. Draw a set of positive ('+') and negative ('-') examples in the two-dimensional feature space for which the best decision tree of depth two makes no errors, while the best decision tree of depth one (one decision node, two leaves) makes as many errors as the best decision tree of depth zero (a single leaf).

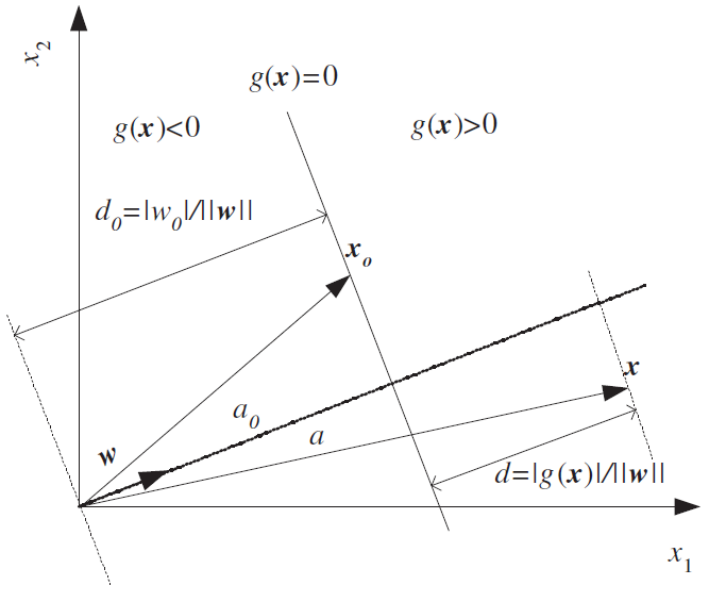
Solution: the two trees are obvious for the following dataset:



Question [9.2, 9.9, 9.10]: solution available on the textbook.

- **Linear discrimination**

Question [10.2]



Solution: Given figure 10.1, first let us take input $\boldsymbol{x}_0$ on the hyperplane. The angle between $\boldsymbol{x}_0$ and $\boldsymbol{w}$ is $\boldsymbol{a}_0$ and because it is on the hyperplane

$g(\boldsymbol{x}_0) = 0$. Then

$$\begin{aligned} g(\boldsymbol{x}_0) &= \boldsymbol{w}^T\boldsymbol{x}_0 + w_0 = \|\boldsymbol{w}\|\|\boldsymbol{x}_0\|\cos a_0 + w_0 = 0 \\ d_0 &= \|\boldsymbol{x}_0\|\cos a_0 = \frac{|w_0|}{\|\boldsymbol{w}\|} \end{aligned}$$

For any $\boldsymbol{x}$ with angle a to $\boldsymbol{w}$, similarly we have

$$\begin{aligned} g(\boldsymbol{x}) &= \boldsymbol{w}^T\boldsymbol{x} + w_0 = \|\boldsymbol{w}\|\|\boldsymbol{x}\|\cos a + w_0 \\ d &= \|\boldsymbol{x}\|\cos a - \frac{w_0}{\|\boldsymbol{w}\|} = \frac{g(\boldsymbol{x}) - w_0 + w_0}{\|\boldsymbol{w}\|} = \frac{g(\boldsymbol{x})}{\|\boldsymbol{w}\|} \end{aligned}$$

Question [10.3]

Solution: Given that $y_i = \frac{\exp a_i}{\sum_j \exp a_j}$,

for $i = j$, we have

$$\begin{aligned} \frac{\partial y_i}{\partial a_i} &= \frac{\exp a_i \left(\sum_j \exp a_j\right) - \exp a_i \exp a}{\left(\sum_j \exp a_j\right)^2} \\ &= \frac{\exp a_i}{\sum_j \exp a_j}\left(\frac{\sum_j \exp a_j - \exp a_i}{\sum_j \exp a_j}\right) \\ &= y_i(1 - y_i) \end{aligned}$$

for $i \neq j$, we have

$$\begin{aligned} \frac{\partial y_i}{\partial a_j} &= \frac{-\exp a_i \exp a_j}{\left(\sum_j \exp a_j\right)^2} \\ &= -\left(\frac{\exp a_i}{\sum_j \exp a_j}\right)\left(\frac{\sum_j \exp a_j}{\sum_j \exp a_j}\right) \\ &= y_i(0 - y_j) \end{aligned}$$

So we can combine in one equation as $\frac{\partial y_i}{\partial a_j} = y_i(\delta_{ij} - y_j)$

Question [10.4, 10.7, 10.8, 10.9]: solution available on the textbook.

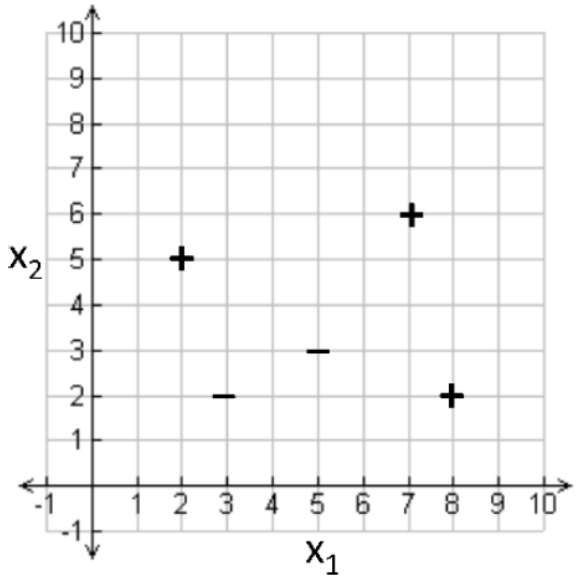
<u>Question 1</u>.

Consider the logistic regression model with parameters w_0, w_1, w_2:

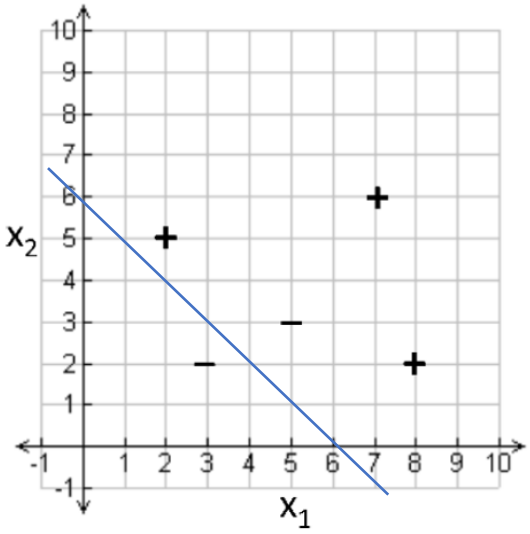
$$P(y = 1|x_1, x_2) = \frac{2^{w_2x_2 + w_1x_1 + w_0}}{1 + 2^{w_2x_2 + w_1x_1 + w_0}}. \quad (1)$$

Note, base 2 is employed rather than e for computational convenience.

a. For parameter weights $w_0 = -6$, $w_1 = 1$, $w_2 = 1$, draw the decision boundary on the following plot:



Solution: The decision boundary is P($y = 1|x_1, x_2$) = 0.5. Plug the w_0, w_1, and w_2 values into P($y = 1|x_1, x_2$) = 0.5, we get $w_2 x_2 + w_1 x_1 + w_0 = 0$, i.e., $x_2 + x_1 - 6 = 0$. See the figure below.



b. What is the log likelihood of the negative datapoint at $(x_1 = 5, x_2 = 3)$, $\log_2 P(Y = 0|x_1 = 5, x_2 = 3)$ in the logistic regression model from subquestion a.?

$$\log_2 P(y=0|x_1,x_2) = \log_2 \frac{1}{1+2^{w_2x_2+w_1x_1+w_0}} = -\log(1+2^{w_2x_2+w_1x_1+w_0})$$

$$1\cdot 3 + 1\cdot 5 - 6 = 2$$

$$-\log_2(5)$$

c. What is the gradient of this datapoint?

$\frac{\partial}{\partial w_0} \log_2 P(Y = 0|x_1 = 5, x_2 = 3) =$

$\frac{\partial}{\partial w_1} \log_2 P(Y = 0|x_1 = 5, x_2 = 3) =$

$\frac{\partial}{\partial w_2} \log_2 P(Y = 0|x_1 = 5, x_2 = 3) =$

$$\frac{\partial}{\partial w_i} \log_2 P(y=0|x_1,x_2) = -\frac{\partial}{\partial w_i} \log\left(1+2^{w_2x_2+w_1x_1+w_0}\right)$$

$$= -\frac{1}{1+2^{w_2x_2+w_1x_1+w_0}} \cdot 2^{w_2x_2+w_1x_1+w_0} \cdot \frac{\partial}{\partial w_i}\left(w_2x_2+w_1x_1+w_0\right)$$

$$P(y=1|x_1=5,x_2=3)$$

So the three numbers required in this question are

$-\frac{4}{5}$

$-\frac{4}{5} \times 5 = -4$

$-\frac{4}{5} \times 3 = 2.4$

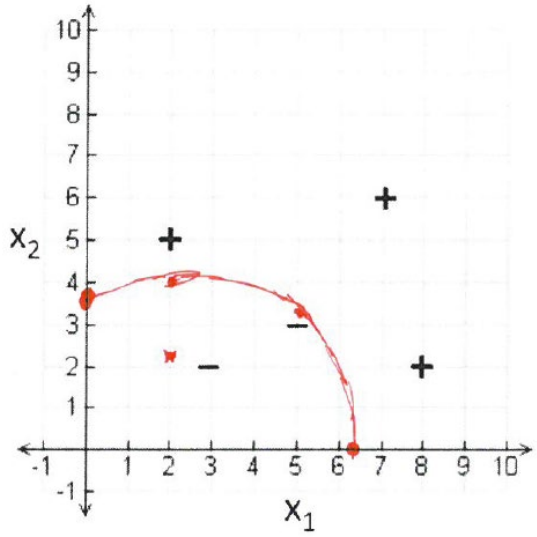
d. Consider adding an additional input feature, $x_1^2 + x_2^2$.

What weights w_3, w_2, w_1, w_0 provide a good fit to the data?

Solution: the solution is not unique. To be good, we just need to make the probability on the right side of the threshold of 0.5:

$$+ (2,5): \quad 29w_3 + 5w_2 + 2w_1 + w_0 > 0$$
$$- (3,2): \quad 13w_3 + 2w_2 + 3w_1 + w_0 \leq 0$$
$$- (5,3): \quad 34w_3 + 3w_2 + 5w_1 + w_0 \leq 0$$
$$+ (7,6): \quad 75w_3 + 6w_2 + 7w_1 + w_0 > 0$$
$$+ (8,2): \quad 68w_3 + 2w_2 + 8w_1 + w_0 > 0$$

Draw the new decision boundary for this modified logistic regression model on the following plot:



Question 2. [True or False] A logistic regression model with L1 regularization will have lower training log loss / higher log likelihood than the same logistic regression model without regularization.

Solution: **False**. Since we are talking about training loss, regularization will be only harmful although it does help with the test error (generalization).

Question 3. Which model is more prone to overfitting: a decision tree of depth *k* or a logistic regression model with *k* features? Why?

Solution: decision tree of depth *k*. This is because such a tree can have 2^k branches, which can lead to much more complicated separating boundaries.

Question 4 Consider a binary logistic regression model, $P(Y = 1|x) = \frac{2^{w_2 x_2 + w_1 x_1 + w_0}}{2^{w_2 x_2 + w_1 x_1 + w_0} + 1}$, that provides predictions of: $P(Y = 1|X_1 = 2, X_2 = 0) = 0.5$, $P(Y = 1|X_1 = 0, X_2 = 3) = 0.5$, and $P(Y = 1|X_1 = 0, X_2 = 0) = \frac{2}{3}$. What are the values of w_0, w_1, and w_2?

$w_2 \cdot 0 + w_1 \cdot 2 + w_0 = 0$

$w_2 \cdot 3 + w_1 \cdot 0 + w_0 = 0$

$w_2 \cdot 0 + w_1 \cdot 0 + w_0 = 1$

$w_0 = 1$

$w_1 = -1/2$

$w_2 = -1/3$

Question: why do $|w|$ and $|w_0|$ in Figure 10.7 converge to infinity as gradient descent converges?

Solution: Figure 10.7 shows the change of parameter w and w_0 (which are both real numbers encoding the slope and intercept of the straight line respectively), as the gradient descent method (Figure 10.6 on page 252 of textbook) processes to optimize w and w_0 in logistic regression. We want to find out their optimal value for the discriminate function $wx + w_0$ when the gradient descent converges.

Think about a simpler case: the training set consists of a single **positive** example ($x^1 \in C_1$, $r^1=1$) and a single **negative** example ($x^2 \in C_2$, $r^2=0$). Assume the training set is linearly separable.

Given w and w_0, the linear discriminate boundary is $g(x|w, w_0) = wx + w_0$, and the posterior is

$$y = P(r = 1|x) = \frac{1}{1 + \exp(-wx - w_0)}.$$

In order to optimize w_1 and w_0, we maximize likelihood function:

$$L = \prod_{t=1}^{2} (y^t)^{r^t} (1 - y^t)^{(1-r^t)}$$

In this simplified case, likelihood function equals to

$$\frac{1}{1+\exp(-wx^1-w_0)} \times \left(1 - \frac{1}{1+\exp(-wx^2-w_0)}\right) \quad (1)$$

Here the first item is the posterior probability that x^1 is positive $P(r^1 = 1|x^1)$, and the second item is the posterior probability that x^2 is negative $P(r^2 = 0|x^2)$.

Since the training set is linearly separable and gradient descent converges to the global optimal, it can produce a value of w and w_0 such that $P(r^1 = 1|x^1) > 0.5$ and $P(r^2 = 0|x^2) > 0.5$ (corresponding to slant lines in Figure 10.7). That is,

$$P(r^1 = 1|x^1; w, w_0) = \frac{1}{1 + \exp(-wx^1 - w_0)} > 0.5 \quad \Rightarrow \quad -wx^1 - w_0 < 0$$

$$P(r^2 = 0|x^2; w, w_0) = 1 - \frac{1}{1 + \exp(-wx^2 - w_0)} > 0.5 \quad \Rightarrow \quad -wx^2 - w_0 > 0$$

Now suppose we magnify w and w_0 by 10 times. The two inequalities above imply that

$$-10wx^1 - 10w_0 < -wx^1 - w_0 \;\Rightarrow\; P(r^1 = 1|x^1; 10w, 10w_0) > P(r^1 = 1|x^1; w, w_0)$$

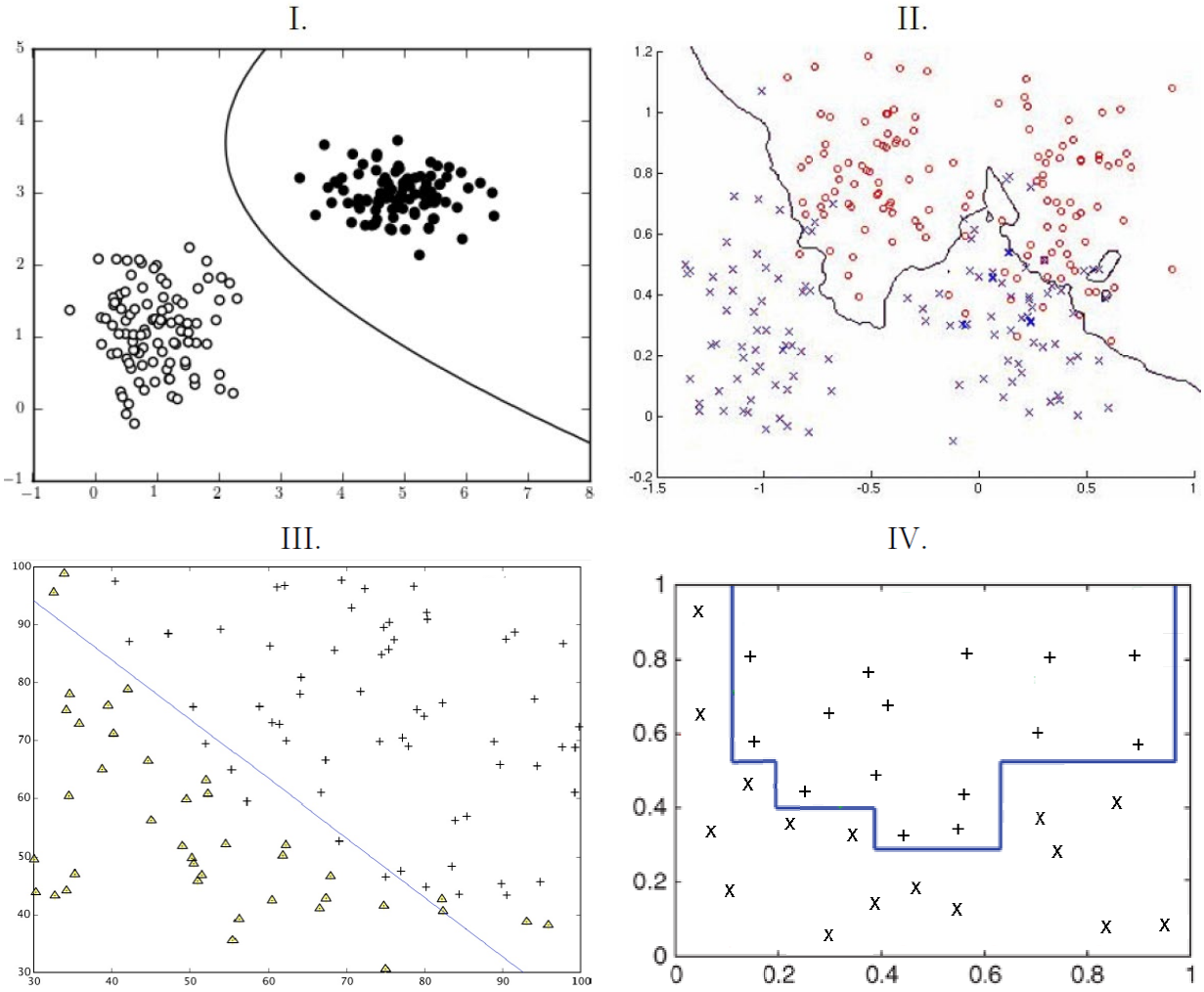
$$-10wx^2 - 10w_0 > -wx^2 - w_0 \;\Rightarrow\; P(r^2 = 0|x^2; 10w, 10w_0) > P(r^2 = 0|x^2; w, w_0)$$

In other words, the posterior probability of the correct class for both training examples gets improved by multiplying w and w_0 by 10. Keep multiplying by 10, these posterior probabilities will converge to

1, which is the perfect result (recall gradient converges to the global optimal solution) attained as both $|w|$ and $|w_0|$ approach infinity.

- **Support vector machines (SVMs) and kernel methods**

Question 1. Consider the following predictors: decision tree, k-nearest neighbor, Gaussian naive Bayes, logistic regression for linear discrimination, and linear support vector machines. You will need to explain which predictor(s) could produce a particular decision boundary and the parameters of the predictor that would do so (how).



a. What predictor(s) produces the decision boundary from figure I and how?

Solution: Gaussian naive Bayes. When the covariance matrix of the two classes are not tied, we can get a quadratic boundary.

b. What predictor(s) produces the decision boundary from figure II and how?

Solution: k-means. Obvious as the boundary is so ragged.

Solution: it can be either logistic regression for linear discrimination or linear support vector machines. The reason is because the separation boundary is linear and not parallel to axes.

d. What predictor(s) produces the decision boundary from figure IV and how?

Solution: decision tree, because as the separation boundary is not linear and all segments are axis parallel (either vertical or horizontal).

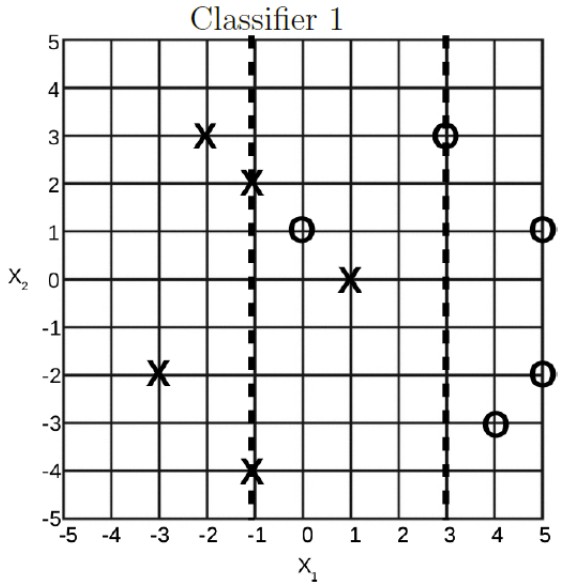
<u>Question 2 (this question is beyond the scope of this course)</u>: Consider the linear SVM where *M* is the distance from decision boundary to margin boundary and C being a "slack budget" for the classifier:

$$\max_{\rho, w_0, w_1, w_2, \xi} \rho$$

$$\text{such that: } y_i(w_2 x_{i,2} + w_1 x_{i,1} + w_0) \geq \rho(1 - \xi_i), \text{ and } \xi_i \geq 0, \forall i \in \{1, \ldots, n\};$$

$$w_1^2 + w_2^2 = 1; \text{and} \sum \xi_i \leq C;$$

on the following dataset, where 'X' class is defined as positive ($y_i = 1$) and 'O' class is defined as negative ($y_i = -1$).



For the support vector classifier with margin boundaries (dashed lines) shown above:

a. Which datapoints are support vectors? Please circle them on the figure.

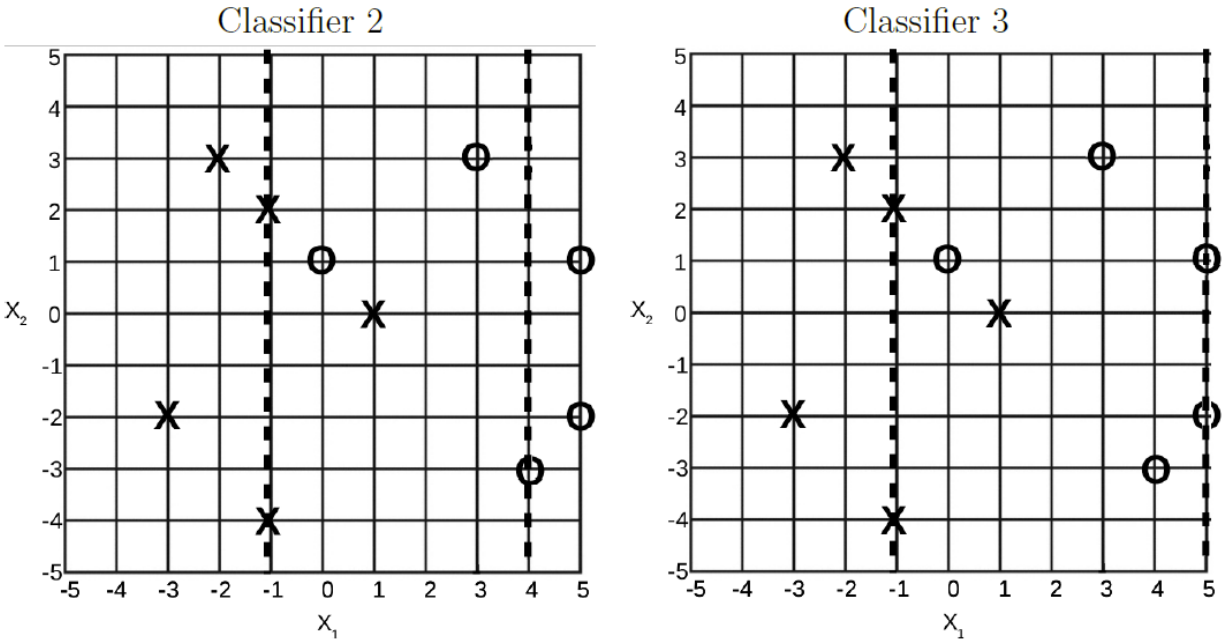
Solution: The support vectors are the three crosses and two circles lying between the two dotted lines, including on the lines.

b. What is the value of ρ for this classifier?

c. What are the values of w_2, w_1, w_0 for this classifier?

Solution: The decision boundary is x_1 = 1, also considering that 'O' class is defined as negative ($y_i = -1$). Therefore $w_2 = 0, w_1 = -1, w_0 = 1$.

d. Consider the classifier when the margin boundary at $x_1 = 3$ is instead moved to $x_1 = 4$ (Classifier 2) or moved to $x_1 = 5$ (Classifier 3). When will the optimization above prefer Classifier 2 or Classifier 3 (in terms slack budget parameters C) instead of Classifier 1? (Hint: what is the margin/sum of slacks for each?)



Solution: As the slack budget parameters C increases, the classifier has more freedom to tolerate classification error.

In Classifier 2, the margin ρ =2.5 and decision boundary $x_1 = 1.5$, plug it into the optimization problem, we can obtain C = 14/5 (only points between the margin boundary contributes to the slackness C)

In Classifier 3, the margin ρ =3 and decision boundary $x_1 = 2$, plug it into the optimization problem, we can obtain C = 10/3.

Question 3. Consider binary classification on a large dataset using a support vector machine. Let Φ be a mapping from the original feature space to a higher-dimensional feature space.

a. What are the advantages of using SVM in the projected feature space rather than SVM in the original feature space?

b. (this question is beyond the requirement of the course). What are the advantages of using kernel methods for the SVM rather than linear SVM applied to the expanded feature space?

Solution: Kernel methods for the SVM is not equivalent to linear SVM applied to the expanded feature space, because it does not allow the weights in the higher-dimensional feature space to be freely chosen. In fact, the weight vector has to be in the span of $\Phi(x_i)$ where x_i are training examples.

Question 4. [True or False] If the non-support examples of a trained SVM are removed from the training set and the SVM is retrained from the smaller data, the decision boundary is guaranteed to be the same as the original SVM's decision boundary.

Solution: True.

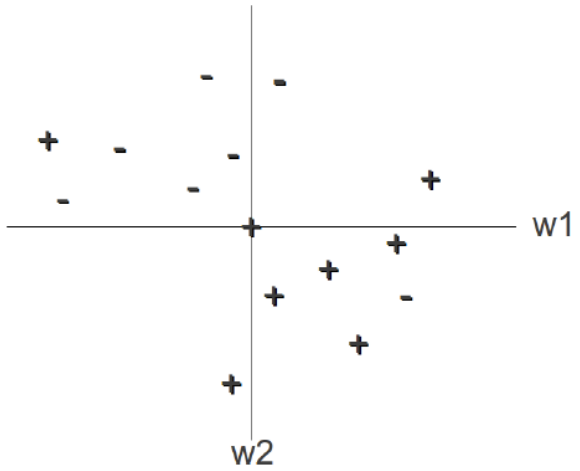
Question 5. [True or False] Consider a linear SVM with k support vectors. The SVM using polynomial kernel with degree $n > 2$ trained from the same data must have at least k support vectors.

Solution: False. It depends on the data and the regularization parameter C.

Question 6. Consider the linear support vector machine,

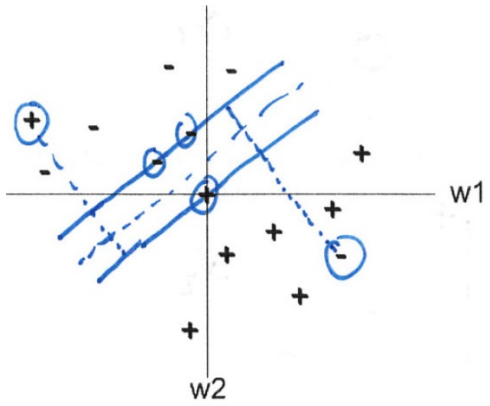
$$\min_{\mathbf{w},b,\xi\geq 0} ||\mathbf{w}||^2 + C\sum_{i=1}^{n}\xi_i \text{ such that: } y_i(\mathbf{w}\cdot\mathbf{x}_i - b) \geq 1-\xi_i \qquad \forall i \in \{1,...,n\},$$

on the following dataset:



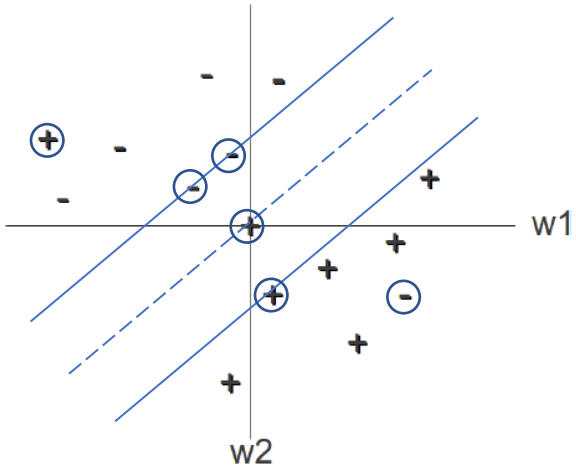
a. Plot the decision boundary and margin boundaries when C is very large (as C goes to infinity) and circle the examples serving as support vectors.

Solution: When *C* is large, it really tries to minimize the hinge loss (i.e. putting the points on the right side of the boundary), and ignore the regularization (i.e. margin). But there are still one + on the far left and one – on the bottom-right which can't be correctly classified:



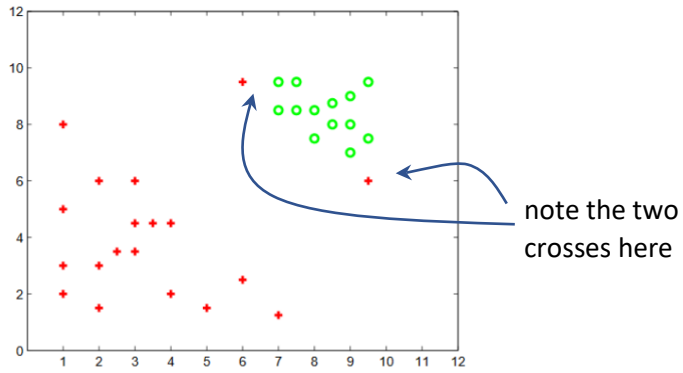
b. Plot the decision boundary and margin boundaries when *C* is more moderate and circle the examples serving as support vectors.

Solution: now SVM will care more about the margin, not minding misclassifying the + at the origin.



Question 7. The goal of this section is to correctly classify test data points, given a training data set. **You have been warned, however, that the training data comes from sensors which can be error-prone, so you should avoid trusting any specific point too much**.

For this problem, assume that we are training an SVM with a **quadratic kernel** $(x \cdot y + 1)^2$-- that is, a polynomial kernel of degree 2. The dataset is presented the figure below. The slack penalty *C* (weight for the slack variables in SVM) determines the location of the separating hyperplane.



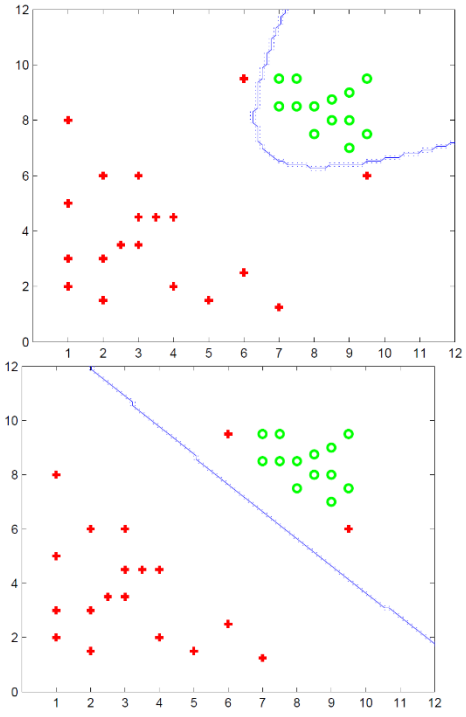
Please answer the following questions **qualitatively**. Give a one sentence answer/justification for each, and draw your solution in the appropriate part of the figure below the problems.

(a) [3 pt] Where would the decision boundary be for very large values of C (i.e., $C \rightarrow \infty$)? Remember that we are using an SVM with a quadratic kernel. Draw on the figure below (right). Justify your answer.

Answer: For large values of C, the penalty for misclassifying points is very high, so the decision boundary will perfectly separate the data if possible. See right for the boundary learned using libSVM and C = 100000.

(b) [3 pt] For C $\approx$ 0, indicate in the figure below (right), where you would expect the decision boundary to be? Justify your answer.

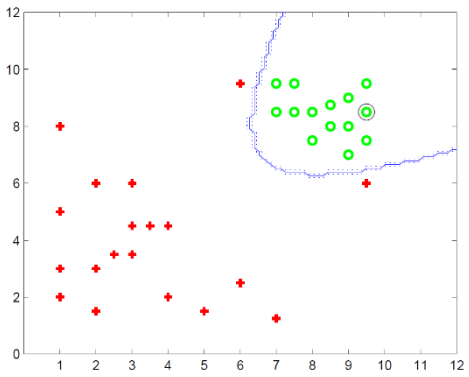
Answer: The classifier can maximize the margin between most of the points, while misclassifying a few points, because the penalty is so low. See right for the boundary learned by libSVM with C = 0.00005. (you only need to be qualitative)



(c) [3 pt] Which of the two cases above would you expect to work better in the classification task? Why?

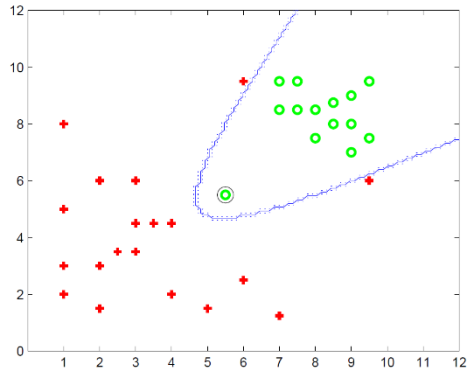
Answer: We were warned not to trust any specific data point too much, so we prefer the solution where C $\approx$ 0, because it maximizes the margin between the dominant clouds of points.

(d) [3 pt] Draw a data point (either cross or circle) which will not change the decision boundary learned for very large values of *C*. Justify your answer.



Answer: We add the point circled on the right, which is correctly classified by the original classifier, and will not be a support vector.

(e) [3 pt] Draw a data point (either cross or circle) which will significantly change the decision boundary learned for very large values of C. Justify your answer.



Answer: Since C is very large, adding a point that would be incorrectly classified by the original boundary will force the boundary to move.

Question [13.1]. Solution available in the textbook.

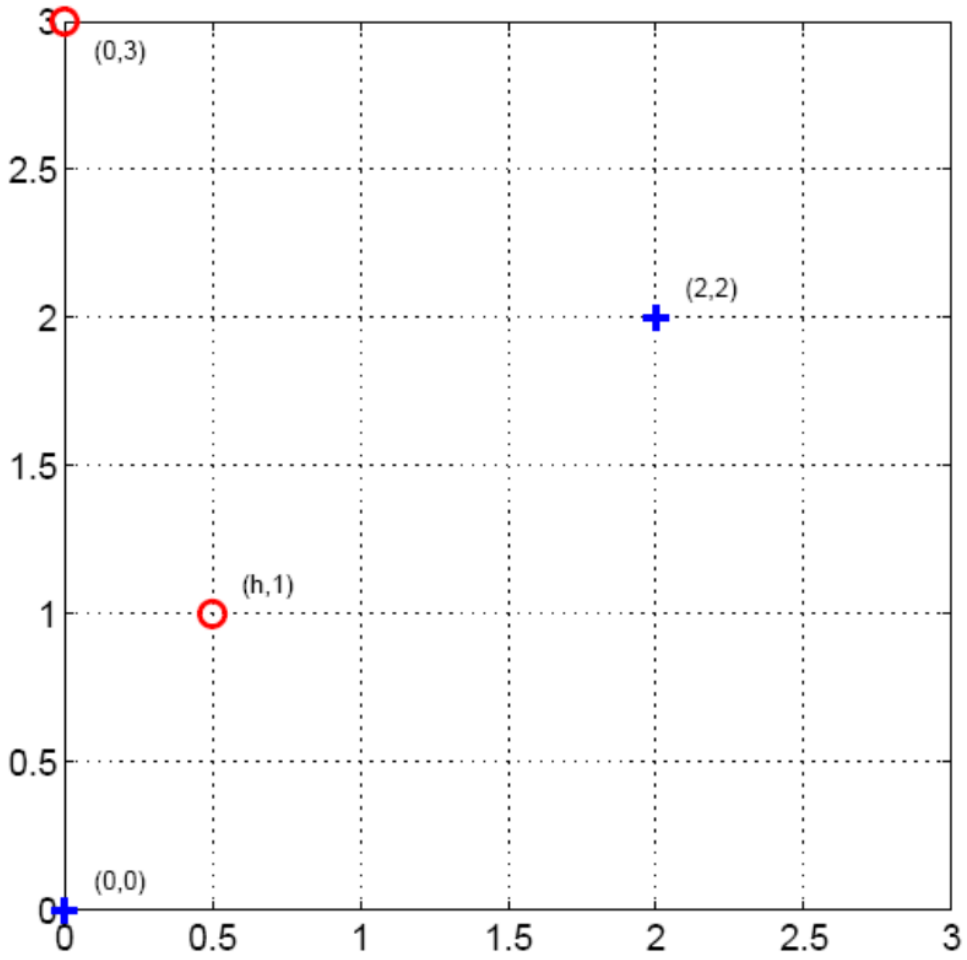
Question [13.3]. Solution available in the textbook. This question is beyond the scope of the course, but could be interesting for some.

Question 8.

Suppose we only have four training examples in two dimensions (see the figure on the next page): positive examples at x1 = [0, 0] and x2 = [2, 2], and negative examples at x3 = [h, 1] , x4 = [0, 3], where we treat $0 \leq h \leq 3$ as a parameter.

(a) [5 pt] What is the range of h (as a subset of $[0, 3]$) so that the training points are still linearly separable?

Solution: [0, 1).



(b) [5 pt] Does the orientation (i.e., normal direction) of the maximum margin decision boundary change as a function of h when the points are separable? **(in addition, what if h < 0 or h > 3)**

Solution: No, it does not. The maximum margin decision boundary keeps parallel to the line segment connecting (0, 0) and (2, 2). That is, slope = 1.

(c) [4 pt] What is the margin achieved by the maximum margin boundary as a function of h?

Hint: It turns out that the margin as a function of h is an affine function.

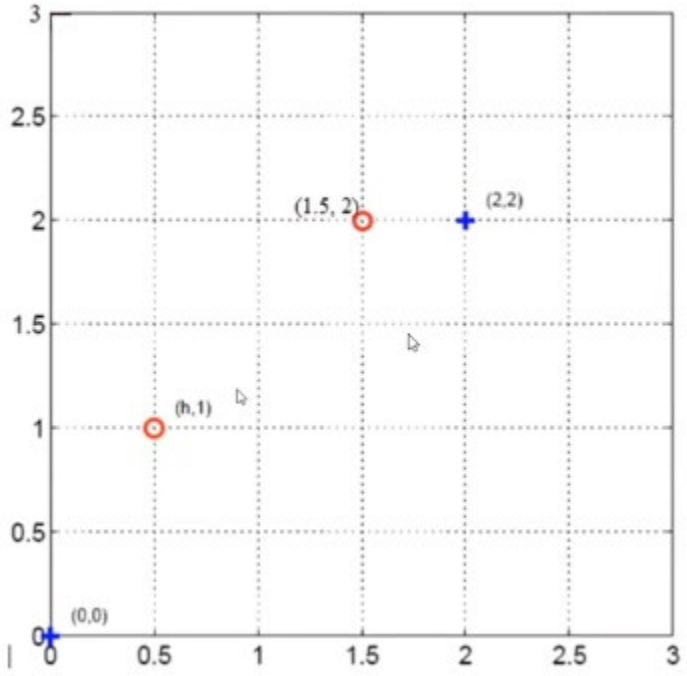
(d) [5 pt] Assume that we can only observe the second component of the input vectors. Without the other component, the labeled training points reduce to (0, y = 1), (2, y = 1), (1, y = –1), and (3, y = –1). What is the lowest order p of polynomial kernel that would allow us to correctly classify these points?

Solution. The classes of the points on the x2-projected line observe the order 1, –1, 1, –1. Therefore, we need a cubic polynomial. So $p = 3$.

(e) [5 pt] With h = 0.5, write out the max-margin hyperplane and also the marginal hyperplanes.

(f) Suppose we change the red circles to (0.5, 1) and (1.5, 2). Draw the data points above, and sketch the maximum-margin hyperplane and also the marginal hyperplanes (the hyperplanes parallel to the maximum-margin hyperplane which pass the nearest points to the maximum-margin hyperplane). Write down the margin ρ (i.e., the distance from the decision boundary to the margin boundary).

(g) Choose three support vectors, write out the system of equations for those support vector data points: $y_i(w_0 + w_1 x_{i,1} + w_2 x_{i,2}) = 1$, and solve for w_0, w_1, and w_2.



(g) Will the boundary change if we make the following changes:

i) move (0, 0) to (1, 0)

ii) move (0, 0) to (-0.25, 0)