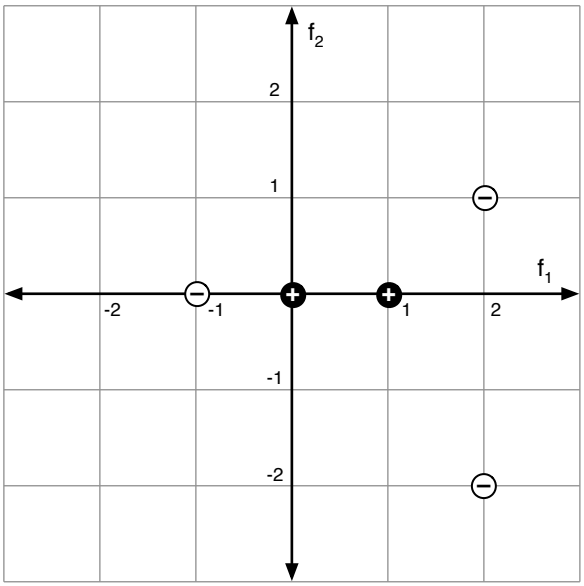
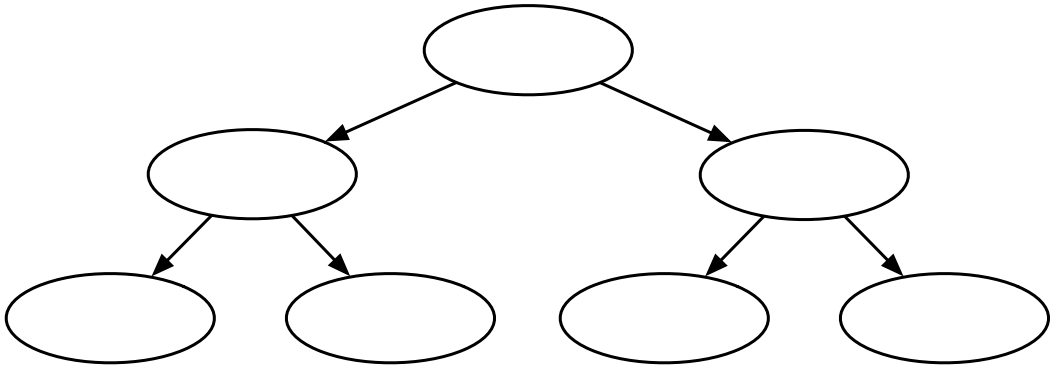
1 Decision Trees



Data points are: Negative: (-1, 0) (2, 1) (2, -2) Positive: (0, 0) (1, 0)

Construct a decision tree using the algorithm described in the notes for the data above.

1. Show the tree you constructed in the diagram below. The diagram is more than big enough, leave any parts that you don't need blank.



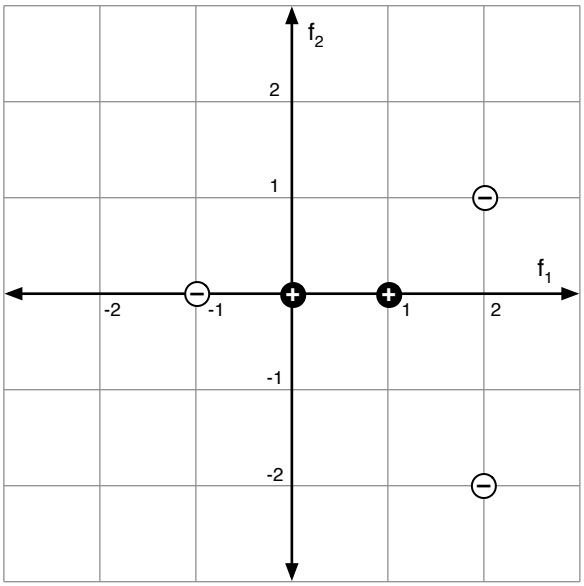
2. Draw the decision boundaries on the graph at the top of the page.

3. Explain how you chose the top-level test in the tree. The following table may be useful.

x	y	-(x/y)*lg(x/y)		x	y	-(x/y)*lg(x/y)
1	2	0.50		1	5	0.46
1	3	0.53		2	5	0.53
2	3	0.39		3	5	0.44
1	4	0.50		4	5	0.26
3	4	0.31				

4. What class does the decision tree predict for the new point: (1, -1.01)

2 Nearest Neighbors



Data points are: Negative: (-1, 0) (2, 1) (2, -2) Positive: (0, 0) (1, 0)

1. Draw the decision boundaries for 1-Nearest Neighbors on the graph above. Your drawing should be accurate enough so that we can tell whether the integer-valued coordinate points in the diagram are on the boundary or, if not, which region they are in.
2. What class does 1-NN predict for the new point: (1, -1.01) Explain why.
3. What class does 3-NN predict for the new point: (1, -1.01) Explain why.

5 Naive Bayes (8 pts)

Consider a Naive Bayes problem with three features, $x_1 \ldots x_3$. Imagine that we have seen a total of 12 training examples, 6 positive (with $y = 1$) and 6 negative (with $y = 0$). Here is a table with some of the counts:

	$y = 0$	$y = 1$
$x_1 = 1$	6	6
$x_2 = 1$	0	0
$x_3 = 1$	2	4

1. Supply the following estimated probabilities. Use the Laplacian correction.

   - $\Pr(x_1 = 1|y = 0)$

   - $\Pr(x_2 = 1|y = 1)$

   - $\Pr(x_3 = 0|y = 0)$

2. Which feature plays the largest role in deciding the class of a new instance? Why?

6 Learning algorithms

For each of the learning situations below, say what learning algorithm would be best to use, and why.

1. You have about 1 million training examples in a 6-dimensional feature space. You only expect to be asked to classify 100 test examples.

2. You are going to develop a classifier to recommend which children should be assigned to special education classes in kindergarten. The classifier has to be justified to the board of education before it is implemented.

3. You are working for Am*z*n as it tries to take over the retailing world. You are trying to predict whether customer X will like a particular book, as a function of the input which is a vector of 1 million bits specifying whether each of Am*z*n's other customers liked the book. You will train a classifier on a very large data set of books, where the inputs are everyone else's preferences for that book, and the output is customer X's preference for that book. The classifier will have to be updated frequently and efficiently as new data comes in.

4. You are trying to predict the average rainfall in California as a function of the measured currents and tides in the Pacific ocean in the previous six months.

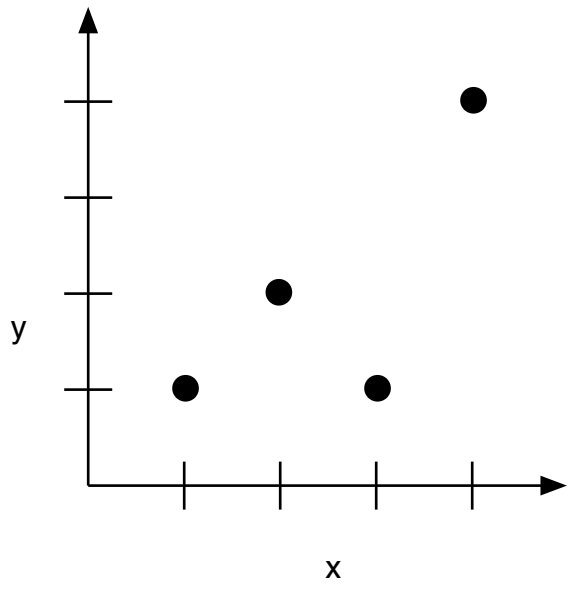
4. neural network (no weight decay or early stopping)

5. SVM (with arbitrary data and $c < \infty$)

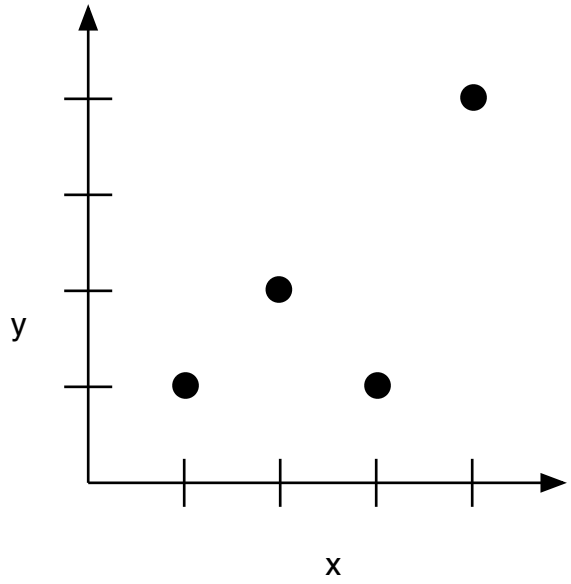
8 Regression

Consider a one-dimensional regression problem (predict y as a function of x). For each of the algorithms below, draw the approximate shape of the output of the algorithm, given the data points shown in the graph.

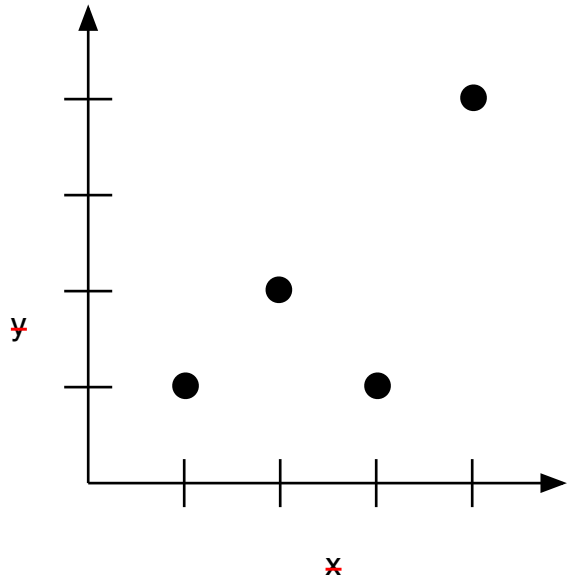
1. 2-nearest-neighbor (equally weighted averaging)



2. regression trees (with leaf size 1)



~~3. one-layer neural network~~



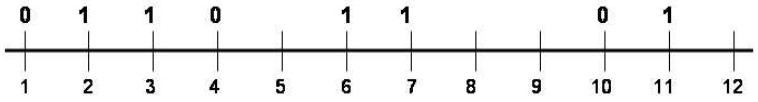
~~4. multi-layer neural network~~

4 Machine Learning — Continuous Features (20 points)

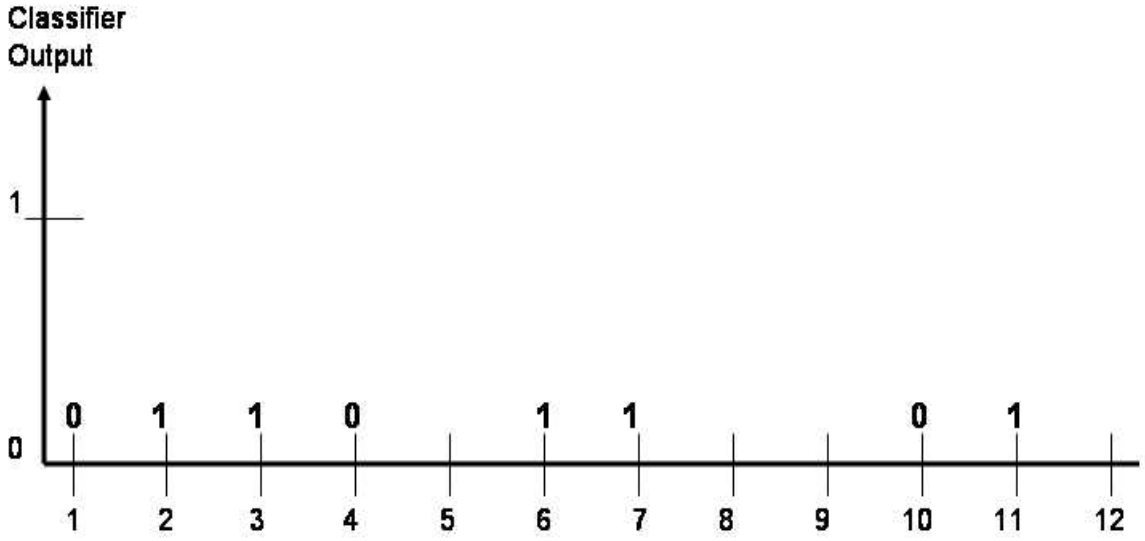
In all the parts of this problem we will be dealing with one-dimensional data, that is, a set of points (x^i) with only one feature (called simply x). The points are in two classes given by the value of y^i. We will show you the points on the x axis, labeled by their class values; we also give you a table of values.

4.1 Nearest Neighbors

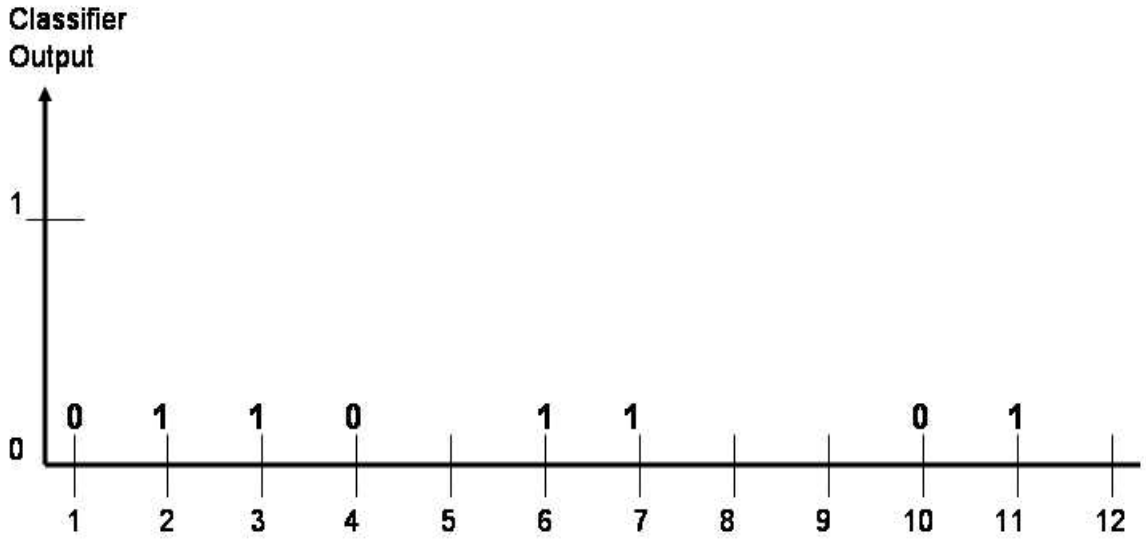
i	x^i	y^i
1	1	0
2	2	1
3	3	1
4	4	0
5	6	1
6	7	1
7	10	0
8	11	1



1. In the figure below, draw the output of a 1-Nearest-Neighbor classifier over the range indicated in the figure.

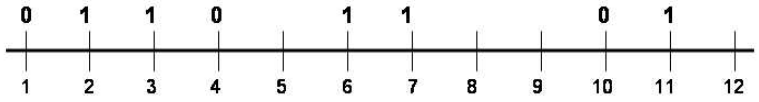


2. In the figure below, draw the output of a 5-Nearest-Neighbor classifier over the range indicated in the figure.



4.2 Decision Trees

Answer this problem using the same data as in the Nearest Neighbor problem above.



Which of the following three tests would be chosen as the top node in a decision tree?

$$x \leq 1.5 \qquad x \leq 5 \qquad x \leq 10.5$$

Justify your answer.

You may find this table useful.

x	y	-(x/y)*lg(x/y)		x	y	-(x/y)*lg(x/y)
1	2	0.50		1	8	0.38
1	3	0.53		3	8	0.53
2	3	0.39		5	8	0.42
1	4	0.50		7	8	0.17
3	4	0.31		1	9	0.35
1	5	0.46		2	9	0.48
2	5	0.53		4	9	0.52
3	5	0.44		5	9	0.47
4	5	0.26		7	9	0.28
1	6	0.43		8	9	0.15
2	6	0.53		1	10	0.33
5	6	0.22		3	10	0.52
1	7	0.40		7	10	0.36
2	7	0.52		9	10	0.14
3	7	0.52				
4	7	0.46				
5	7	0.35				
6	7	0.19				

6 Pruning Trees (20 points)

Following are some different strategies for pruning decision trees. We assume that we grow the decision tree until there is one or a small number of elements in each leaf. Then, we prune by deleting individual leaves of the tree until the score of the tree starts to get worse. The question is how to score each possible pruning of the tree.

For each possible definition of the score below, explain whether or not it would be a good idea and give a reason why or why not.

1. The score is the percentage correct of the tree on the training set.

2. The score is the percentage correct of the tree on a separate validation set.

3. The score is the percentage correct of the tree, computed using cross validation.

4. The score is the percentage correct of the tree, computed on the training set, minus a constant C times the number of nodes in the tree.

   C is chosen in advance by running this algorithm (grow a large tree then prune in order to maximize percent correct minus C times number of nodes) for many different values of C, and choosing the value of C that minimizes training-set error.

5. The score is the percentage correct of the tree, computed on the training set, minus a constant C times the number of nodes in the tree.

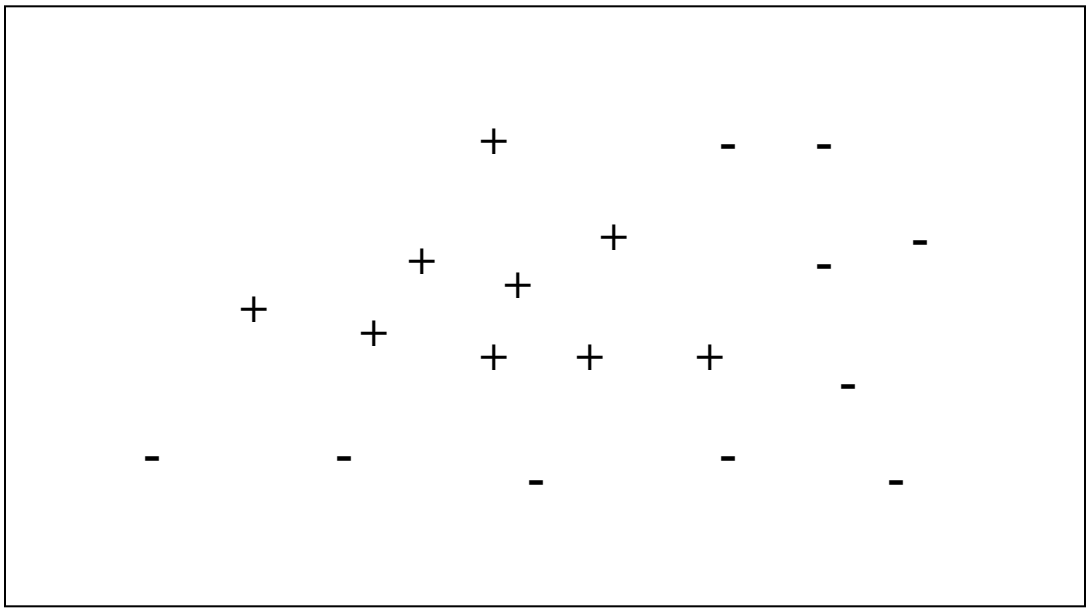
   C is chosen in advance by running cross-validation trials of this algorithm (grow a large tree then prune in order to maximize percent correct minus C times number of nodes) for many different values of C, and choosing the value of C that minimizes cross-validation error.

Problem 4: Learning (25 points)

Part A: (5 Points)

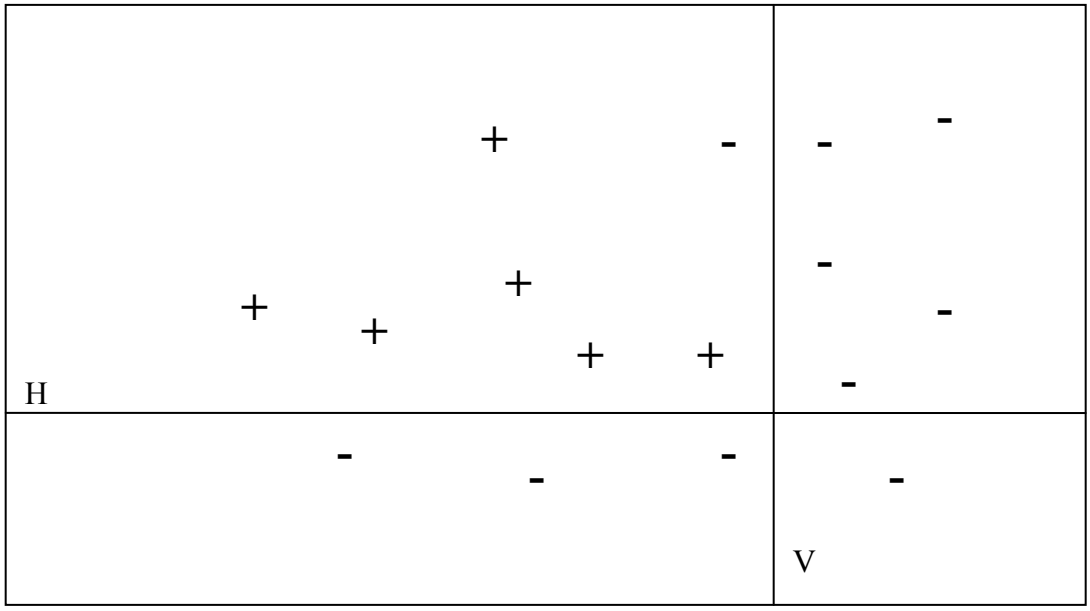
Since the cost of using a nearest neighbor classifier grows with the size of the training set, sometimes one tries to eliminate redundant points from the training set. These are points whose removal does not affect the behavior of the classifier for any possible new point.

1. In the figure below, sketch the decision boundary for a 1-nearest-neighbor rule and circle the redundant points.



2. What is the general condition(s) required for a point to be declared redundant for a 1-nearest-neighor rule? Assume we have only two classes (+, -). Restating the definition of redundant ("removing it does not change anything") is not an acceptable answer. Hint – think about the neighborhood of redundant points.

Part B: (5 Points)

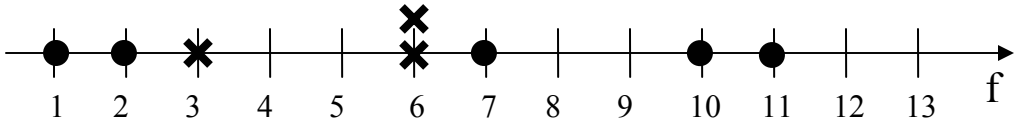


Which of H or V would be preferred as an initial split for a decision (identification) tree? Justify your answer numerically.

x	y	-(x/y)*lg(x/y)
1	2	0.50
1	3	0.53
2	3	0.39
1	4	0.50
3	4	0.31
1	5	0.46
2	5	0.53
3	5	0.44
4	5	0.26
1	6	0.43
2	6	0.53
5	6	0.22
1	7	0.40
2	7	0.52
3	7	0.52
4	7	0.46
5	7	0.35
6	7	0.19

x	y	-(x/y)*lg(x/y)
1	8	0.38
3	8	0.53
5	8	0.42
7	8	0.17
1	9	0.35
2	9	0.48
4	9	0.52
5	9	0.47
7	9	0.28
8	9	0.15
1	10	0.33
3	10	0.52
7	10	0.36
9	10	0.14

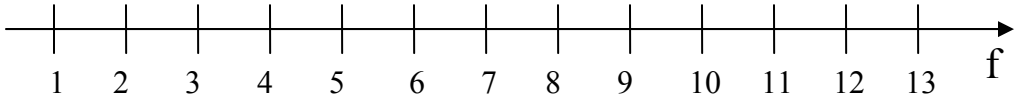
Problem 1: Classification (40 points)



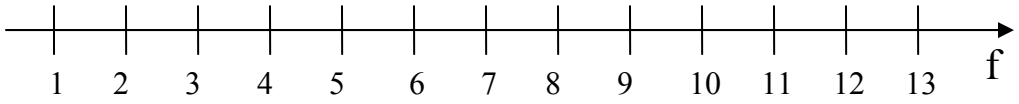
The picture above shows a data set with 8 data points, each with only one feature value, labeled f. Note that there are two data points with the same feature value of 6. These are shown as two X's one above the other, but they really should have been drawn as two X's on top of each other, since they have the same feature value.

Part A: (10 Points)

1. Consider using 1-Nearest Neighbors to classify unseen data points. On the line below, darken the segments of the line where the 1-NN rule would predict an O given the training data shown in the figure above.

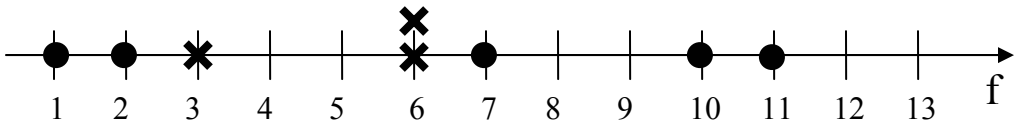


2. Consider using 5-Nearest Neighbors to classify unseen data points. On the line below, darken the segments of the line where the 5-NN rule would predict an O given the training data shown in the figure above.



3. If we do 8-fold cross-validation using 1-NN on this data set, what would be the predicted performance? Settle ties by choosing the point on the left. Show how you arrived at your answer.

Part B: (8 Points)

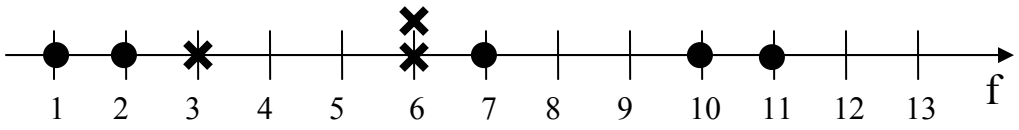


Using this same data set, show the decision tree that would be built from this data. Assume that the tests in the tree are of the form $f \leq c$. For each test show the approximate value of the average disorder for that test. To help you compute this, there's a small table of values of –(x/y)*log(x/y) for small integer x and y.

x	y	-(x/y)*lg(x/y)
1	2	0.50
1	3	0.53
2	3	0.39
1	4	0.50
3	4	0.31
1	5	0.46
2	5	0.53
3	5	0.44
4	5	0.26
1	6	0.43
2	6	0.53
5	6	0.22
1	7	0.40
2	7	0.52
3	7	0.52
4	7	0.46
5	7	0.35
6	7	0.19

x	y	-(x/y)*lg(x/y)
1	8	0.38
3	8	0.53
5	8	0.42
7	8	0.17
1	9	0.35
2	9	0.48
4	9	0.52
5	9	0.47
7	9	0.28
8	9	0.15
1	10	0.33
3	10	0.52
7	10	0.36
9	10	0.14

Part C: (12 Points)



Construct the **simplest** neural net (using sigmoid units) that can accurately classify this data. Pick a set of appropriate weights for the network. Assume that we will predict O when the network's output is less than 0.5 and X when the output is above 0.5.
Whenever possible use weights that are either 1, -1, 10 or -10.

Problem 2: Overfitting (20 points)

For each of the supervised learning methods that we have studied, indicate how the method could overfit the training data (consider both your design choices as well as the training) and what you can do to minimize this possibility. There may be more than one mechanism for overfitting, make sure that you identify them all.

Part A: Nearest Neighbors (5 Points)

1. How does it overfit?

2. How can you reduce overfitting?

Part B: Decision Trees (5 Points)

1. How does it overfit?

2. How can you reduce overfitting?

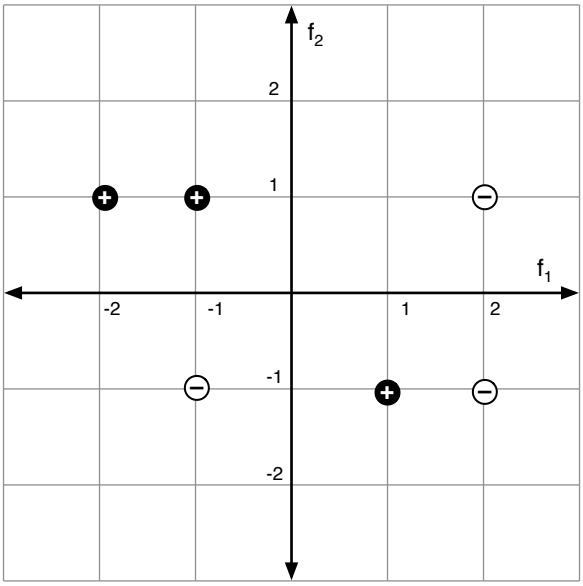
Problem 3: Spaminator (10 points)

Suppose that you want to build a program that detects whether an incoming e-mail message is spam or not. You decide to attack this using machine learning. So, you collect a large number of training messages and label them as spam or not-spam. You further decide that you will use the presence of individual words in the body of the message as features. That is, you collect every word found in the training set and assign to each one an index, from 1 to N. Then, given a message, you construct a feature vector with N entries and write in each entry a number that indicates how many times the word appears in that message.

Part A: (6 Points)

If you had to choose between a Nearest Neighbor implementation or an Decision Tree implementation, which would you choose? Justify your answer briefly both in terms of expected accuracy and efficiency of operation. Indicate the strength and weaknesses of each approach.

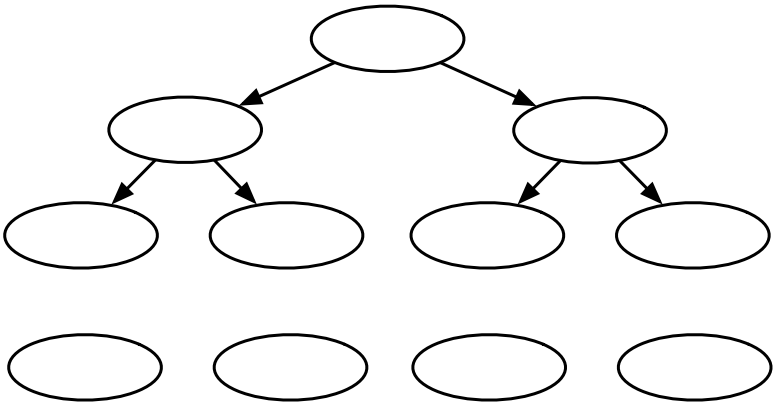
4 Decision Trees (20 points)



Data points are: Negative: (-1, -1) (2, 1) (2, -1) Positive: (-2, 1) (-1, 1) (1,-1)

Construct a decision tree using the algorithm described in the notes for the data above.

1. Show the tree you constructed in the diagram below. The diagram is more than big enough, leave any parts that you don't need blank. If you need to connect in the extra four nodes in the last row, add the connections to the parent nodes.



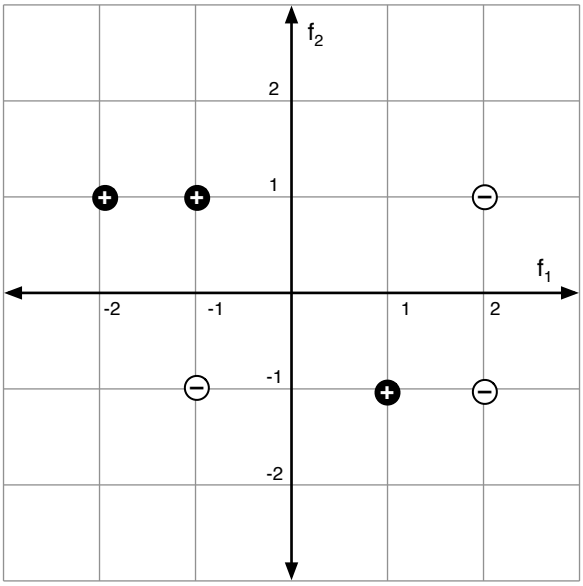
2. Draw the decision boundaries on the graph at the top of the page.

3. Explain how you chose the top-level test in the tree. The following table may be useful.

x	y	-(x/y)*lg(x/y)		x	y	-(x/y)*lg(x/y)
1	2	0.50		1	5	0.46
1	3	0.53		2	5	0.53
2	3	0.39		3	5	0.44
1	4	0.50		4	5	0.26
3	4	0.31				

4. What class does the decision tree predict for the new point: (1, 1)

5 Nearest Neighbors (20 points)



Data points are: Negative: (-1, -1) (2, 1) (2, -1) Positive: (-2, 1) (-1, 1) (1,-1)

1. Draw the decision boundaries for 1-Nearest Neighbors on the graph above. Your drawing should be accurate enough so that we can tell whether the integer-valued coordinate points in the diagram are on the boundary or, if not, which region they are in.

2. What class does 1-NN predict for the new point: (1, 1) Explain why.

3. What class does 1-NN predict for the new point: (1, 0) Explain why.

4. What class does 3-NN predict for the new point: (1, 0) Explain why.

5. In general, how would you select between two alternative values of k for use in k-nearest neighbors?

6 Naive Bayes (15 points)

Consider a Naive Bayes problem with three features, $x_1 \ldots x_3$. Imagine that we have seen a total of 12 training examples, 6 positive (with $y = 1$) and 6 negative (with $y = 0$). Here are the actual points:

x_1	x_2	x_3	y
0	1	1	0
1	0	0	0
0	1	1	0
1	1	0	0
0	0	1	0
1	0	0	0
1	0	1	1
0	1	0	1
1	1	1	1
0	0	0	1
0	1	0	1
1	0	1	1

Here is a table with the summary counts:

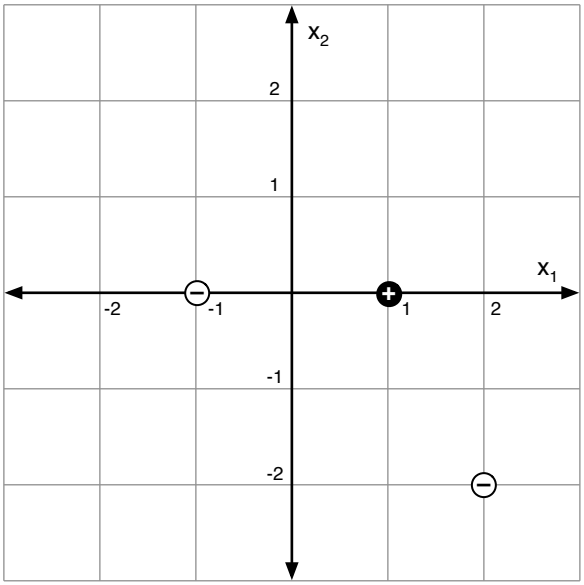
	$y = 0$	$y = 1$
$x_1 = 1$	3	3
$x_2 = 1$	3	3
$x_3 = 1$	3	3

1. What are the values of the parameters $R_i(1, 0)$ and $R_i(1, 1)$ for each of the features i (using the Laplace correction)?

2. If you see the data point $1, 1, 1$ and use the parameters you found above, what output would Naive Bayes predict? Explain how you got the result.

3. Naive Bayes doesn't work very well on this data, explain why.

3 Perceptron (7 pts)



Data points are: Negative: (-1, 0) (2, -2) Positive: (1, 0). Assume that the points are examined in the order given here.

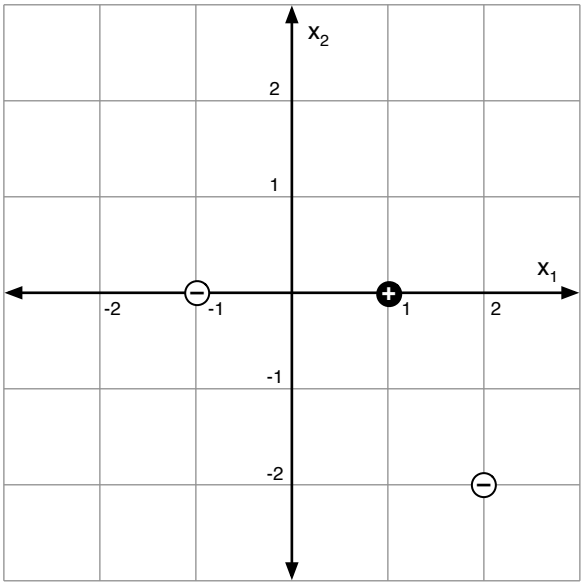
Recall that the perceptron algorithm uses the extended form of the data points in which a 1 is added as the 0th component.

1. The linear separator obtained by the standard perceptron algorithm (using a step size of 1.0 and a zero initial weight vector) is (0 1 2). Explain how this result was obtained.

2. What class does this linear classifier predict for the new point: (2.0, -1.01)

3. Imagine we apply the perceptron learning algorithm to the 5 point data set we used on Problem 1: Negative: (-1, 0) (2, 1) (2, -2), Positive: (0, 0) (1, 0). Describe qualitatively what the result would be.

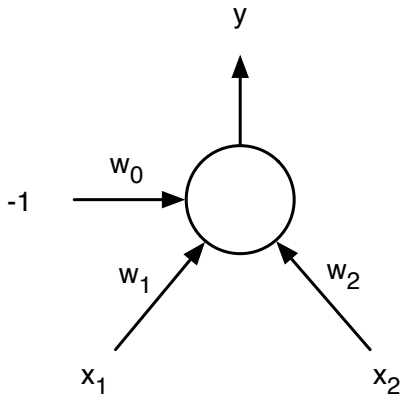
4 Neural Net (9 pts)



Data points are: Negative: (-1, 0) (2, -2) Positive: (1, 0)

Recall that for neural nets, the negative class is represented by a desired output of 0 and the positive class by a desired output of 1.

1. Assume we have a single sigmoid unit:



Assume that the weights are $w_0 = 0, w_1 = 1, w_2 = 1$. What is the computed y value for each of the points on the diagram above?

(a) $x = (-1, 0)$, $y =$

(b) $x = (2, -2)$, $y =$

(c) $x = (1, 0)$, $y =$

Hint: Some useful values of the sigmoid $s(z)$ are $s(-1) = 0.27$ and $s(1) = 0.73$.

2. What would be the change in w_2 as determined by backpropagation using a step size (η) of 1.0? Assume that the input is $x = (2, -2)$ and the initial weights are as specified above. Show the formula you are using as well as the numerical result.

   (a) $\Delta w_2 =$

7 Error versus complexity (15 pts)

Most learning algorithms we have seen try to find a hypotheses that minimizes error. But how do they attempt to control complexity? Here are some possible approaches:

A: Use a fixed-complexity hypothesis class

B: Include a complexity penalty in the measure of error

C: Nothing

For each of the following algorithms, specify which approach it uses and say what hypothesis class it uses (including any restrictions) and what complexity criterion (if any) is included in the measure of error. If the algorithm attempts to optimize the error measure, say whether it is guaranteed to find an optimal solution or just an approximation.

1. perceptron

2. linear SVM

3. decision tree with fixed depth

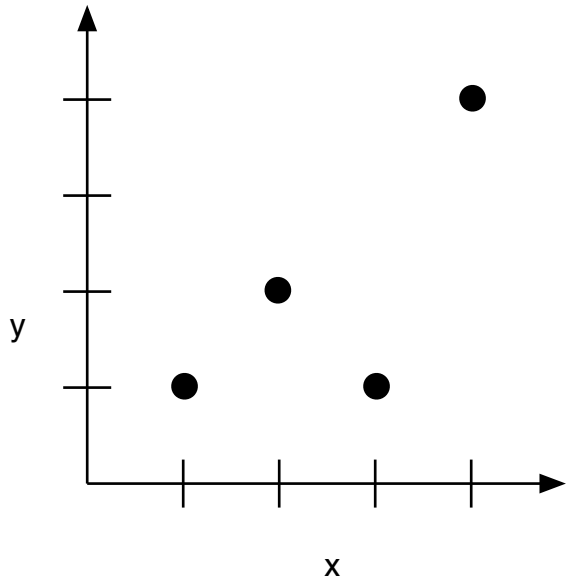
4. neural network (no weight decay or early stopping)

5. SVM (with arbitrary data and $c < \infty$)

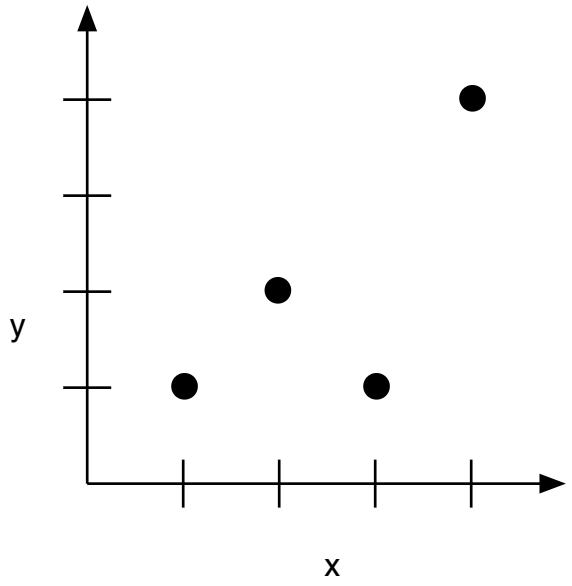
8 Regression (12 pts)

Consider a one-dimensional regression problem (predict y as a function of x). For each of the algorithms below, draw the approximate shape of the output of the algorithm, given the data points shown in the graph.

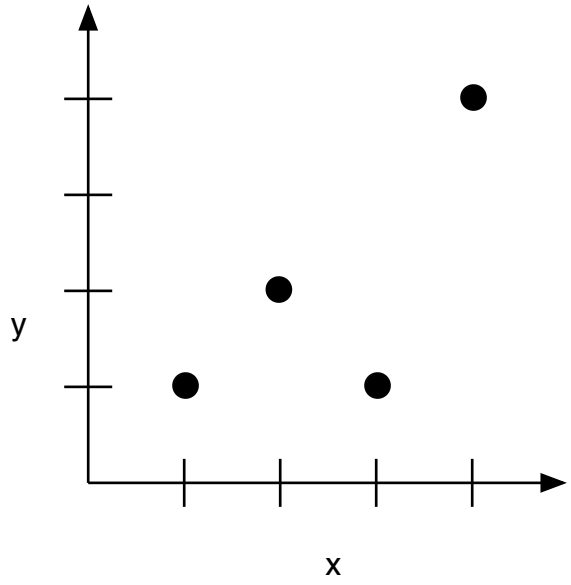
1. 2-nearest-neighbor (equally weighted averaging)



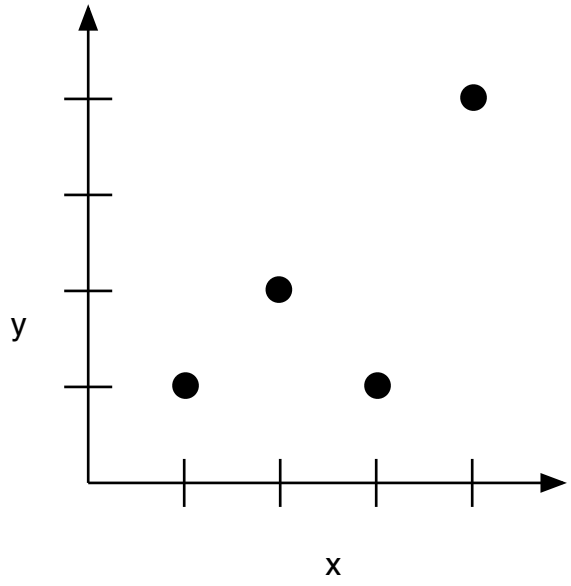
2. regression trees (with leaf size 1)



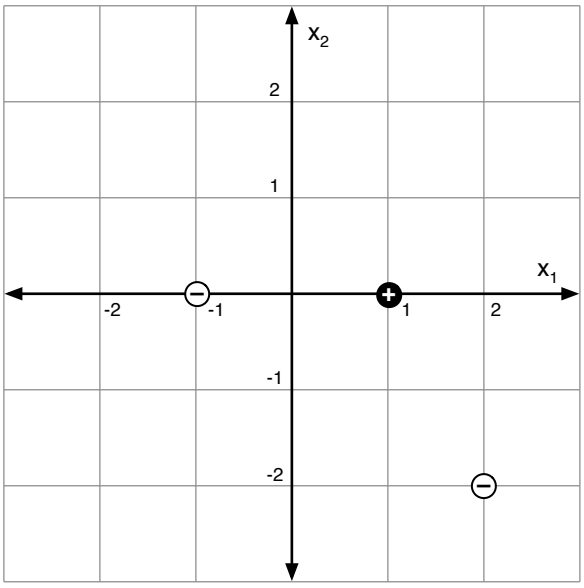
3. one linear neural-network unit



4. multi-layer neural network (with linear output unit)



9 SVM



Data points are: Negative: (-1, 0) (2, -2) Positive: (1, 0)

Recall that for SVMs, the negative class is represented by a desired output of -1 and the positive class by a desired output of 1.

1. For each of the following separators (for the data shown above), indicate whether they satisfy all the conditions required for a support vector machine, assuming a linear kernel. Justify your answers very briefly.

   (a) $x_1 + x_2 = 0$

   (b) $x_1 + 1.5x_2 = 0$

   (c) $x_1 + 2x_2 = 0$

   (d) $2x_1 + 3x_2 = 0$

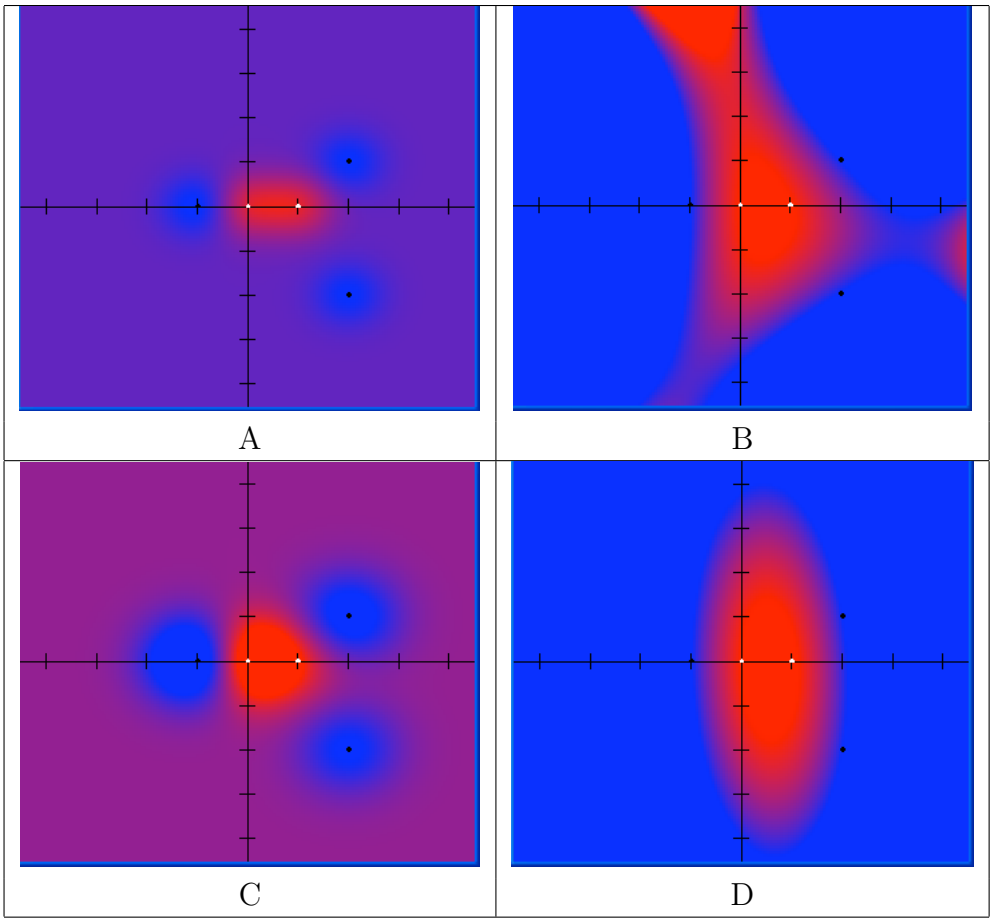
2. For each of the kernel choices below, find the decision boundary diagram (on the next page) that best matches. In these diagrams, the brightness of a point represents the magnitude of the SVM output; red means positive output and blue means negative. The black circles are the negative training points and the white circles are the positive training points.

   (a) Polynomial kernel, degree 2

   (b) Polynomial kernel, degree 3

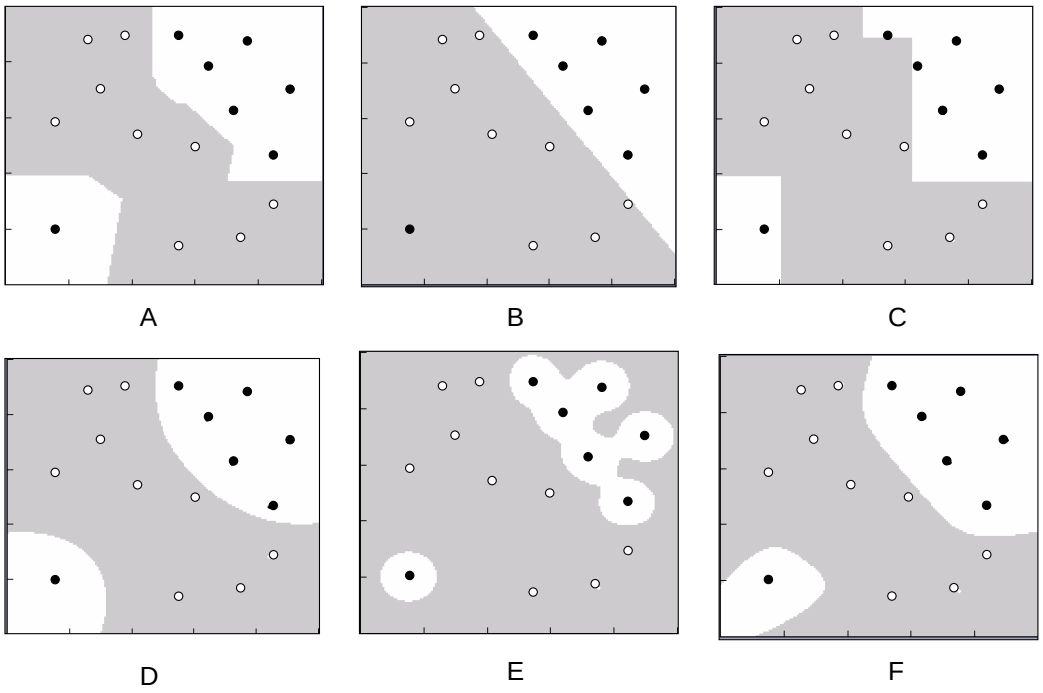
   (c) Radial basis kernel, sigma = 0.5

   (d) Radial basis kernel, sigma = 1.0


A
B
C
D

5 Learning hypothesis classes (16 points)

Consider a classification problem with two real-valued inputs. For each of the following algorithms, specify all of the separators below that it could have generated and explain why. If it could not have generated any of the separators, explain why not.



1. 1-nearest neighbor

2. decision trees on real-valued inputs

3. standard perceptron algorithm

4. SVM with linear kernel

5. SVM with Gaussian kernel ($\sigma = 0.25$)

6. SVM with Gaussian kernel ($\sigma = 1$)

7. neural network with no hidden units and one sigmoidal output unit, run until convergence of training error

8. neural network with 4 hidden units and one sigmoidal output unit, run until convergence of training error

6 Perceptron (8 points)

The following table shows a data set and the number of times each point is misclassified during a run of the perceptron algorithm, starting with zero weights. What is the equation of the separating line found by the algorithm, as a function of x_1, x_2, and x_3? Assume that the learning rate is 1 and the initial weights are all zero.

x_1	x_2	x_3	y	times misclassified
2	3	1	+1	12
2	4	0	+1	0
3	1	1	-1	3
1	1	0	-1	6
1	2	1	-1	11

7 SVMs (12 points)

Assume that we are using an SVM with a **polynomial kernel of degree 2**. You are given the following support vectors:

x_1	x_2	y
-1	2	+1
1	2	-1

The α values for each of these support vectors are equal to 0.05.

1. What is the value of b? Explain your approach to getting the answer.

2. What value does this SVM compute for the input point $(1, 3)$

8 Neural networks (18 points)

A physician wants to use a neural network to predict whether patients have a disease, based on the results of a battery of tests. He has assigned a cost of c_{01} to false positives (generating an output of 1 when it ought to have been 0), and a cost of c_{10} to generating an output of 0 when it ought to have been 1. The cost of a correct answer is 0.

The neural network is just a single sigmoid unit, which computes the following function:

$$g(\bar{x}) = s(\bar{w} \cdot \bar{x})$$

with $s(z)$ being the usual sigmoid function.

1. Give an error function for the whole training set, $E(\bar{w})$ that implements this error metric, for example, for a training set of 20 cases, if the network predicts 1 for 5 cases that should have been 0, predicts 0 for 3 cases that should have been 1 and predicts another 12 correctly, the value of the error function should be: $5c_{01} + 3c_{10}$.

2. Would this be an appropriate error criterion to use for a neural network? Why or why not?

3. Consider the following error function for the whole training set:

$$E(\bar{w}) = c_{10} \sum_{\{i|y^i=1\}} (g(\bar{x}^i) - y^i)^2 + c_{01} \sum_{\{i|y^i=0\}} (g(\bar{x}^i) - y^i)^2$$

   Describe, in English that is not simply a direct paraphrase of the mathematics, what it measures.

4. What is the gradient of this E with respect to $\bar{w}$?

5. Give a complete algorithm for training a single sigmoid unit using this error function.

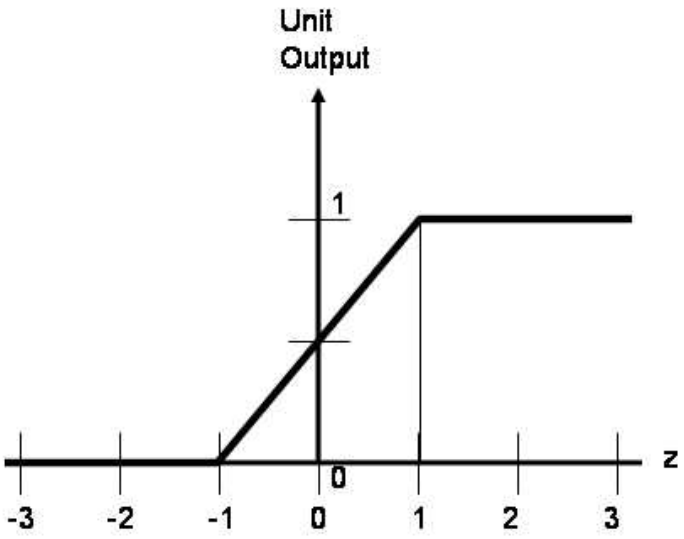
4.3 Neural Nets

Assume that each of the units of a neural net uses one of the the following output functions of the total activation (instead of the usual sigmoid $s(z)$)

- **Linear:** This just outputs the total activation:

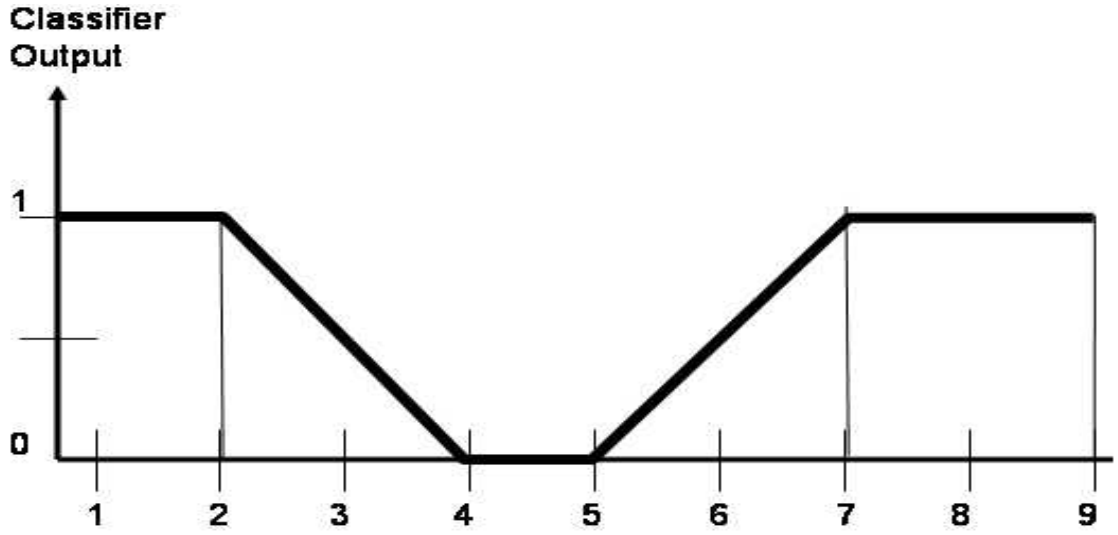
$$l(z) = z$$

- **Non-Linear:** This looks like a linearized form of the usual sigmoid funtion:



$$\begin{aligned} f(z) &= 0 \quad \text{if } z < -1 \\ f(z) &= 1 \quad \text{if } z > 1 \\ f(z) &= 0.5(z+1) \quad \text{otherwise} \end{aligned}$$

Consider the following output from a neural net made up of units of the types described above.



1. Can this output be produced using only linear units? Explain.

2. Construct the simplest neural net out of these two type of units that would have the output shown above. When possible, use weights that have magnitude of 1. Label each unit as either Linear or Non-Linear.

4.4 SVM

What are the values for the α_i and the offset b that would give the maximal margin linear classifier for the two data points shown below? You should be able to find the answer without deriving it from the dual Lagrangian.

i	x^i	y^i
1	0	1
2	4	-1

5 Machine Learning (20 points)

Grady Ent decides to train a single sigmoid unit using the following error function:

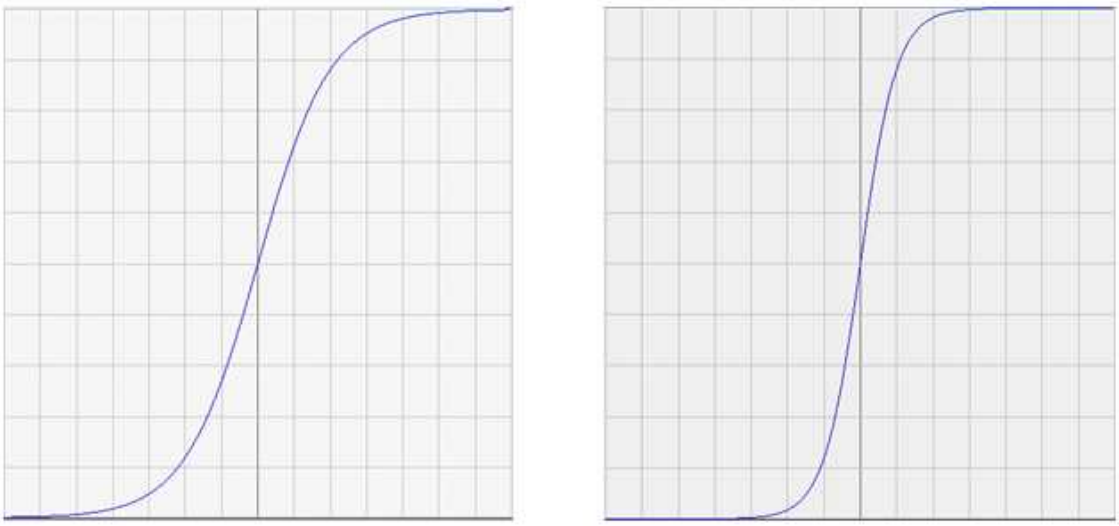
$$E(\mathbf{w}) = \frac{1}{2}\sum_i (y(\mathbf{x}^i, \mathbf{w}) - y^{i*})^2 + \frac{1}{2}\beta \sum_j w_j^2$$

where $y(\mathbf{x}^i, \mathbf{w}) = s(\mathbf{x}^i \cdot \mathbf{w})$ with $s(z) = \frac{1}{1+e^{-z}}$ being our usual sigmoid function.

1. Write an expression for $\frac{\partial E}{\partial w_j}$. Your answer should not involve derivatives.

2. What update should be made to weight w_j given a single training example $< \mathbf{x}, y^* >$. Your answer should not involve derivatives.

3. Here are two graphs of the output of the sigmoid unit as a function of a single feature x. The unit has a weight for x and an offset. The two graphs are made using different values of the magnitude of the weight vector ($\|\mathbf{w}\|^2 = \sum_j w_j^2$).



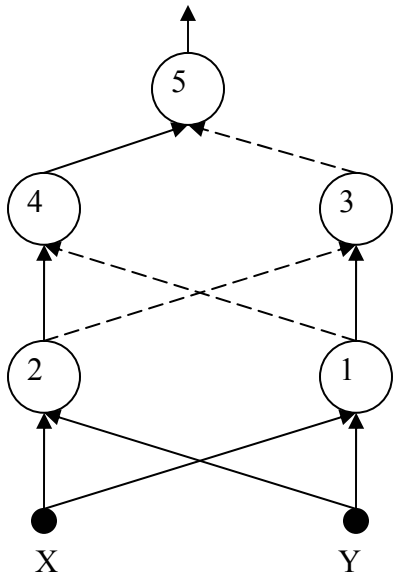
A B

Which of the graphs is produced by the larger $\|\mathbf{w}\|^2$? Explain.

4. Why might penalizing large $\|\mathbf{w}\|^2$, as we could do above by choosing a positive β, be desirable?

5. How might Grady select a good value for β for a particular classification problem?

Part C: (10 Points)



In this network, **all the units are sigmoid except unit 5 which is linear** (its output is simply the weighted sum of its inputs). All the bias weights are zero. The dashed connections have weights of -1, all the other connections (solid lines) have weights of 1.

1. Given X=0 and Y=0, what are the output values of each of the units?

Unit 1 =
Unit 2 =
Unit 3 =
Unit 4 =
Unit 5 =

2. What are the δ values for each unit (as computed by backpropagation defined for squared error) assume that the desired output for the network is 4.

Unit 1 =
Unit 2 =
Unit 3 =
Unit 4 =
Unit 5 =

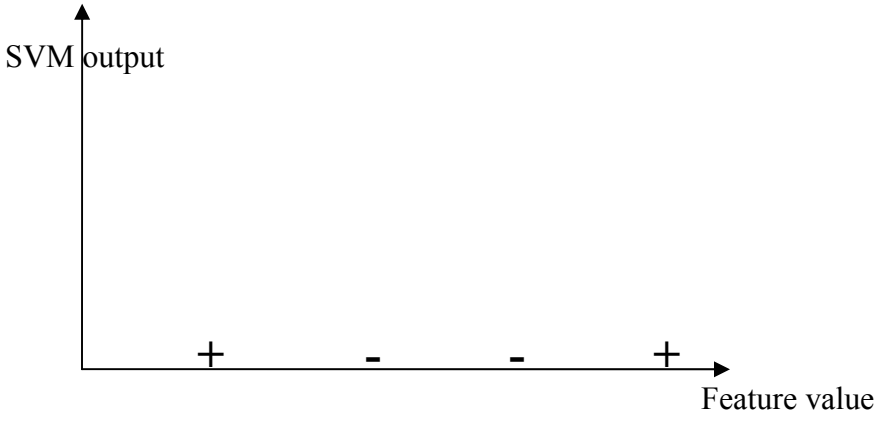
3. What would be the new value of the weight connecting units 2 and 3 assuming that the learning rate for backpropagation is set to 1?

Part D: (10 Points)

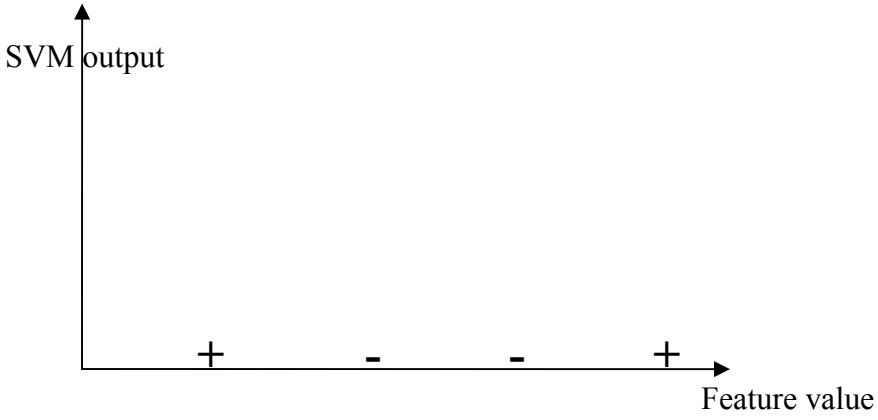
1. Consider the simple one-dimensional classification problem shown below. Imagine attacking this problem with an SVM using a radial-basis function kernel. Assume that we want the classifier to return a positive output for the + points and a negative output for the – points.

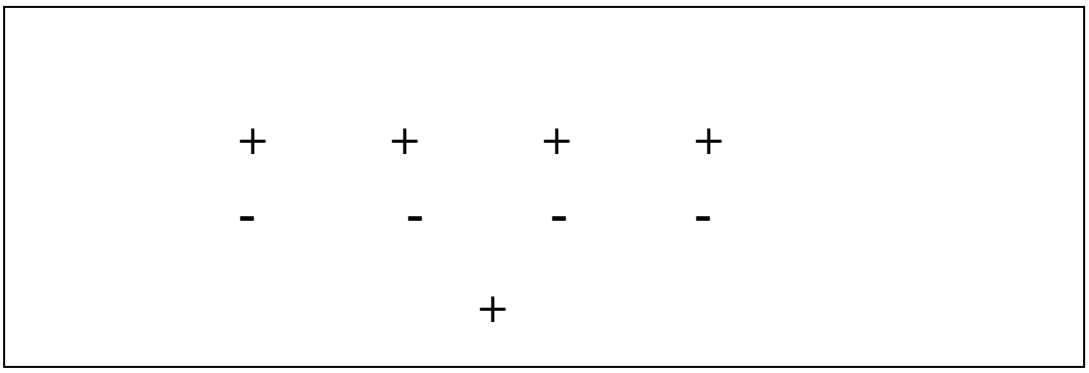
   Draw a plausible classifier output curve for a trained SVM, indicating the classifier output for every feature value in the range shown. Do this twice, once assuming that the standard deviation (σ) is very small relative to the distance between adjacent training points and again assuming that the standard deviation (σ) is about double the distance between adjacent training points.

   Small standard deviation (σ):



   Large standard deviation (σ):





2. Would you expect that a polynomial kernel with d=1 would be successful in carrying out the classification shown above? Explain.

3. Assume we use an SVM with a radial-basis function kernel to classify these same data points. We repeat the classification for different values of the the standard deviation (σ) used in the kernel. What would you expect to be the relationship between the standard deviation used in the kernel and the value of the largest Lagrange multiplier (a_i) needed to carry out the classification? That is, would you expect that the max a_i would increase or decrease as the standard deviation decreases? Explain your answer.

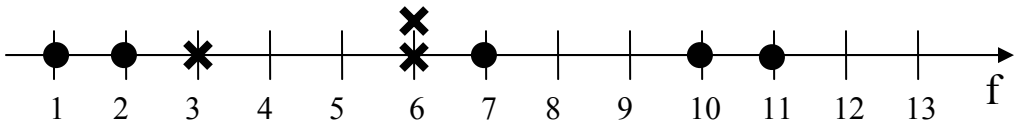
Part E: (5 Points)

Given a validation set (a set of samples which is separate from the training set), explain how it should be used in connection with training different learning functions (be specific about the problems that are being addressed):

1. For a neural net

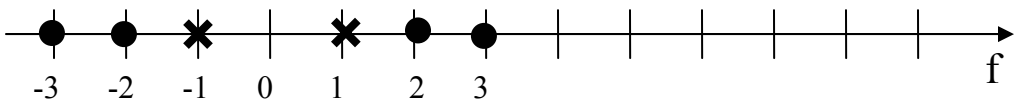
For a decision (identification) tree

Part C: (12 Points)



Construct the **simplest** neural net (using sigmoid units) that can accurately classify this data. Pick a set of appropriate weights for the network. Assume that we will predict O when the network's output is less than 0.5 and X when the output is above 0.5.
Whenever possible use weights that are either 1, -1, 10 or -10.

Part D: (10 Points)



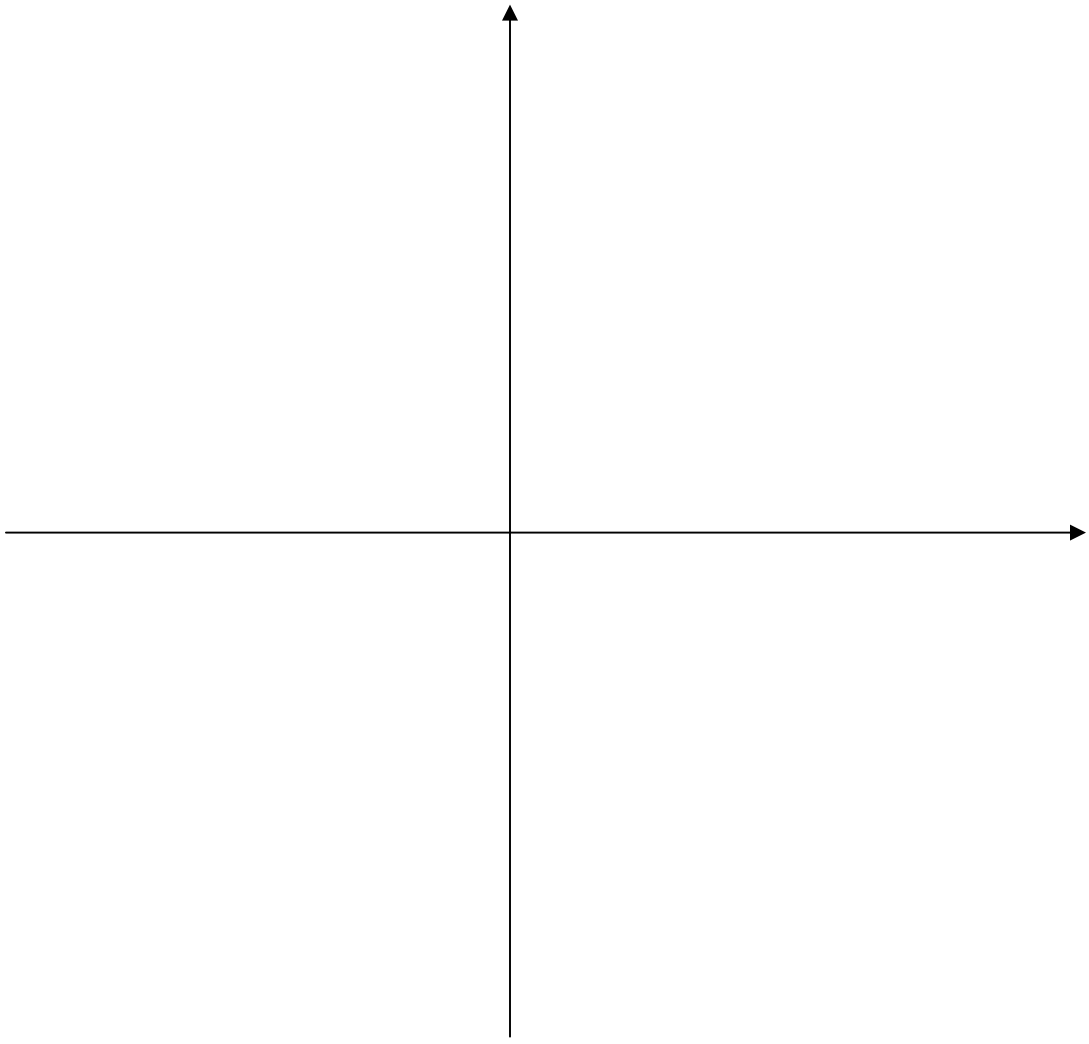
Consider the simplified data set above and consider using an SVM with a polynomial kernel with d=2. Let's say the data points are specified as:

X1 = [-2] Y1= -1
X2 = [-1] Y2= 1
X3 = [1] Y3= 1
X4 = [2] Y4= -1
X5=[-3] Y5= -1
X6=[3] Y6= -1

1. What are the kernel values?

K(x1,x1)	
K(x1,x2)	
K(x2,x3)	
K(x3,x4)	

2. Show a reasonably accurate picture of the **transformed** feature space.
    a. label the axes,
    b. label the data points,
    c. show the separating line that would be found by the SVM,
    d. circle the support vectors.



Problem 2: Overfitting (20 points)

For each of the supervised learning methods that we have studied, indicate how the method could overfit the training data (consider both your design choices as well as the training) and what you can do to minimize this possibility. There may be more than one mechanism for overfitting, make sure that you identify them all.

Part A: Nearest Neighbors (5 Points)

1. How does it overfit?

2. How can you reduce overfitting?

Part B: Decision Trees (5 Points)

1. How does it overfit?

2. How can you reduce overfitting?

Part C: Neural Nets (5 Points)

1. How does it overfit?

2. How can you reduce overfitting?

Part D: SVM [Radial Basis and Polynomial kernels] (5 Points)

1. How does it overfit?

2. How can you reduce overfitting?

Problem 5: Backprop (10 points)

Suppose we want to do regression instead of classification and so we change the (final) output unit of a neural net to be a linear unit, which simply outputs the weighted sum of its inputs (no sigmoid):

$$y = \sum_i w_i x_i$$

All the other units in the network would still retain their sigmoids.

How would this change the backpropagation equations? Derive only the needed changes, do not repeat anything in the usual derivation that does not change.

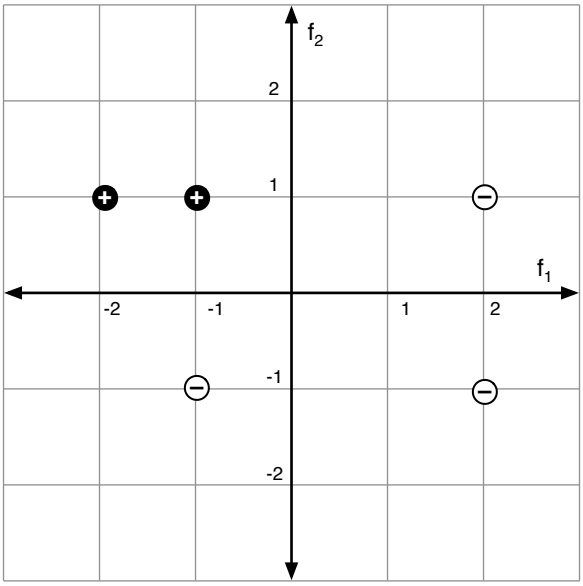
Part B: (4 Points)

Assume that you wanted to reduce the size of the feature vectors (during training and classification), for each of the approaches below indicate why it might be a good or bad idea.

1. Use only the words that appear in spam messages.

2. Eliminate words that are very common in the whole data set.

1 Perceptron (20 points)



Data points are: Negative: (-1, -1) (2, 1) (2, -1) Positive: (-2, 1) (-1, 1)

Recall that the perceptron algorithm uses the extended form of the data points in which a 1 is added as the 0th component.

1. Assume that the initial value of the weight vector for the perceptron is $[0, 0, 1]$, that the data points are examined in the order given above and that the rate (step size) is 1.0. Give the weight vector after one iteration of the algorithm (one pass through all the data points):

2. Draw the separator corresponding to the weights after this iteration on the graph at the top of the page.

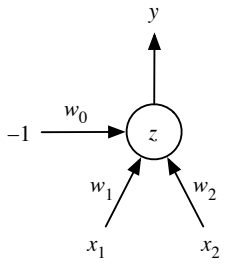
3. Would the algorithm stop after this iteration or keep going? Explain.

4. If we add a positive point at (1,-1) to the other points and retrain the perceptron, what would the perceptron algorithm do? Explain.

2 Neural Nets (20 points)

For this problem, we will consider the simple type of unit shown below. The output of the unit, y, is computed as follows:

$$y = g(z)$$
$$g(z) = z \text{ if } |z| < 1 \text{ and } sign(z) \text{ otherwise}$$
$$z = -w_0 + w_1 x_1 + w_2 x_2$$



We can use this type of unit to classify our inputs by assigning any input for which the output is greater than or equal to 0 as positive and for which the output is less than 0 to negative.

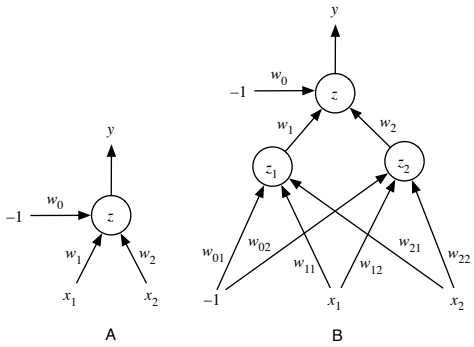
1. Given the four data points: Positive: (0,0), (0,1) Negative: (1,0), (1,1), choose weights for one of the units defined above so that can separate these points.

   (a) $w_0 =$

   (b) $w_1 =$

   (c) $w_2 =$

2. Given the four **different** data points: Positive: (0,0), (1,1) Negative: (0,1), (1,0) and the following two networks, using the type of unit defined above.

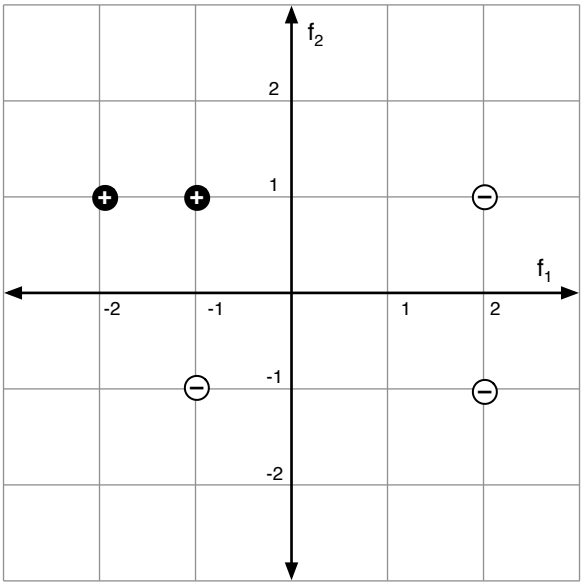


   Can either of the two networks above successfully classify them? Explain why (and which ones) or why not.

3. Derive the on-line learning gradient-descent rule for a **single** unit of this type. The on-line learning rule is incremental, meaning it updates the weights for each individual training point. Thus, we will consider the error at a single training example, i. Assume that the input vector has been augmented so that x_0^i is -1. That way we don't need a separate rule for the constant weight terms. Assume we use the usual quadratic error.

   Note that the derivative of $g(z)$ is discontinuous, which is undesirable, but answer the question ignoring this issue.

3 Maximal Margin Linear Separator (20 points)



Data points are: Negative: (-1, -1) (2, 1) (2, -1) Positive: (-2, 1) (-1, 1)

1. Give the equation of a linear separator that has the maximal geometric margin for the data above. Hint: Look at this geometrically, don't try to derive it formally.

   (a) $w =$

   (b) $b =$

2. Draw your separator on the graph above.
3. What is the value of the smallest geometric margin for any of the points?
4. Which are the support vectors for this separator? Mark them on the graph above.

4 SVM (20 points)

Assume that our training data is four 1-dimensional points, as follows:

index	x	y
1	-2	-1
2	-0.1	-1
3	0.1	1
4	1	1

1. Find the values of all the α_i that would be found by the SVM training algorithm, using a linear kernel. You should be able to do this without going through the Lagrangian minimization procedure. Think about the conditions for the optimization directly.

2. What would the offset be for these values of α_i?

3. What if the value of C were set to 1? What would happen to the values of α_i and the offset? Explain.

5 Machine Learning (20 points)

For each of the statements below, indicate whether they are True or False and **briefly** justify your answer.

1. Given a data set and two alternative sets of weights for a neural network, we can pick between them using cross-validation.
   True or **False** Explain.

2. Given a data set and two values of C for an SVM, we can pick between them using cross-validation.
   True or **False** Explain.

3. The Gaussian (RBF) kernel for SVMs is usually a better choice than the linear kernel.
   True or **False** Explain.

4. We should train a neural net until its error on the training set is as low as possible.
   True or **False** Explain.

5. There is no value of K for which locally weighted averaging will produce exactly y^i when given x^i.
   True or **False** Explain.