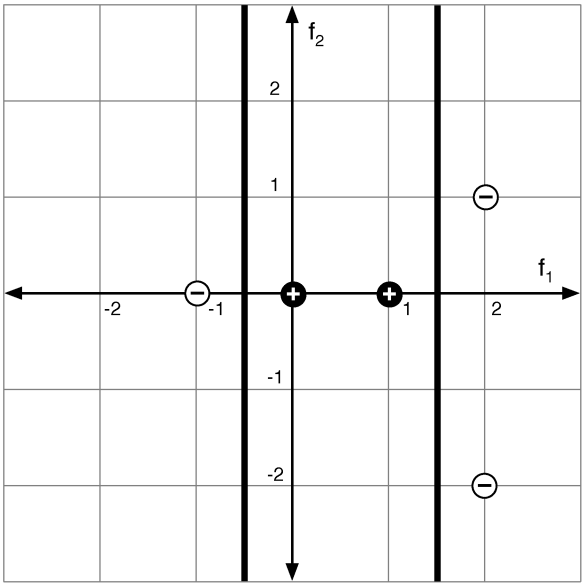
6.034 Quiz 2, Spring 2005
Open Book, Open Notes

Name:

Problem	Score
1 (13 pts)	
2 (8 pts)	
3 (7 pts)	
4 (9 pts)	
5 (8 pts)	
6 (16 pts)	
7 (15 pts)	
8 (12 pts)	
9 (12 pts)	
Total (100 pts)	

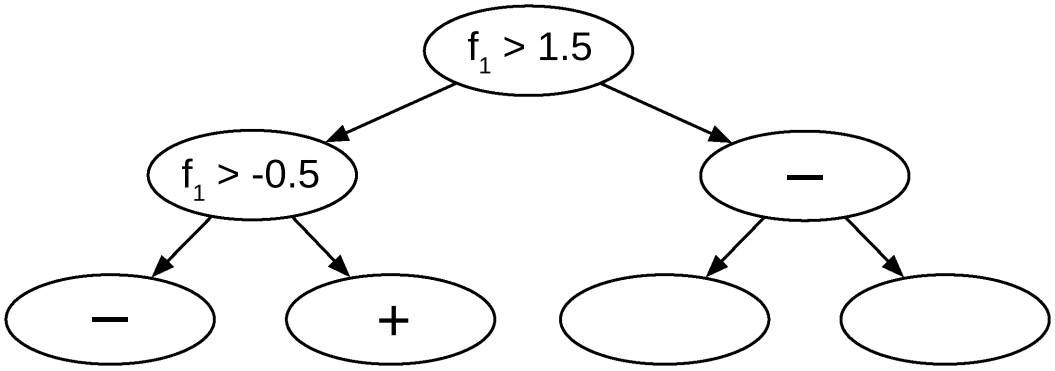
1 Decision Trees (13 pts)



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				

Pick the decision boundary which falls halfway between each pair of adjacent points in each dimension, and which produces the minimum average entropy

$$\begin{aligned} AE &= q_< H(p_<) + (1 - q_<) H(p_>) \\ H(p) &= -p \lg(p) - (1-p) \lg(1-p) \end{aligned}$$

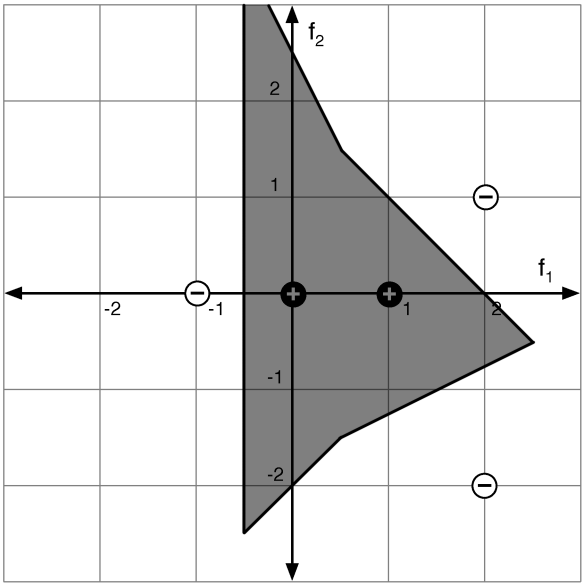
where $q_<$ is the fraction of points below the decision boundary, and $p_<$, $p_>$ are the fraction of positive (+) points below and above the decision boundary, respectively.

$$\begin{aligned} f_2 > \pm 0.5: &\quad AE = \frac{1}{5}(0) + \frac{4}{5}(1) = 0.8 \\ f_1 > 1.5: &\quad AE = \frac{3}{5} H\left(\frac{2}{3}\right) + \frac{2}{5}(0) = \frac{3}{5}(0.39 + 0.53) = 0.552 \\ f_1 > 0.5: &\quad AE = \frac{2}{5}(1) + \frac{3}{5} H\left(\frac{1}{3}\right) = \frac{2}{5} + \frac{3}{5}(0.39 + 0.53) = 0.952 \\ f_1 > -0.5: &\quad AE = \frac{1}{5}(0) + \frac{4}{5}(1) = 0.8 \end{aligned}$$

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

Positive (+)

2 Nearest Neighbors (8 pts)



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. Try to get the integer-valued coordinate points in the diagram on the correct side of the boundary lines.

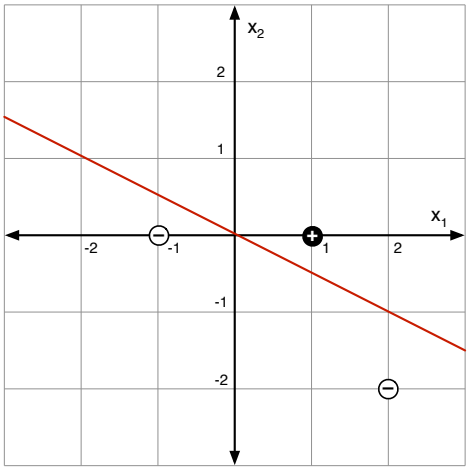
2. What class does 1-NN predict for the new point: (1, -1.01) Explain why.

   Positive (+) since this is the class of the closest data point (1,0).

3. What class does 3-NN predict for the new point: (1, -1.01) Explain why.

   Positive (+) since it is the majority class of the three closest data points (0,0), (1,0) and (2,-2).

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.

   The perceptron algorithm cycles through the augmented points, updating weights according to the update rule $w_{\text{new}} = w + y \cdot x$ after misclassifying points. The intermediate weights are given in the table below.

Test point	misclassified?	Updated weights
Initial weights		**0 0 0**
-: (1 -1 0)	**yes**	**-1 1 0**
-: (1 2 -2)	**yes**	**-2 -1 2**
+: (1 1 0)	**yes**	**-1 0 2**
-: (1 -1 0)	**no**	
-: (1 2 -2)	**no**	
+: (1 1 0)	**yes**	**0 1 2**
-: (1 -1 0)	**no**	
-: (1 2 -2)	**no**	
+: (1 1 0)	**no**	

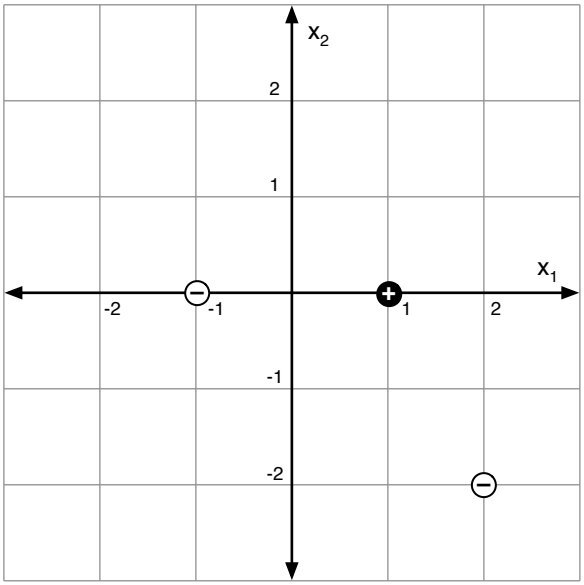
2. What class does this linear classifier predict for the new point: (2.0, -1.01)

   The margin of the point is -0.01, so it would be classified as negative.

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.

   The perceptron algorithm would not converge since the 5 point data set is not linearly separable.

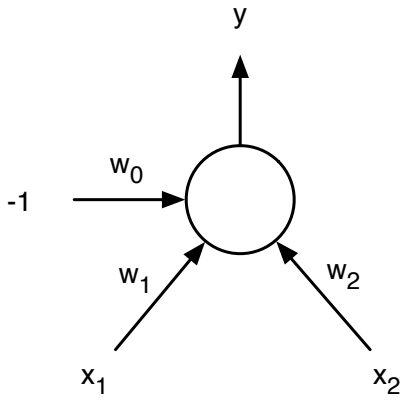
4 Neural Net (9 pts)



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) $x = (-1, 0)$, $y = \mathbf{s}(\mathbf{0} \cdot \mathbf{-1} + \mathbf{1} \cdot \mathbf{-1} + \mathbf{1} \cdot \mathbf{0}) = \mathbf{s}(\mathbf{-1}) = \mathbf{0.27}$

(b) $x = (2, -2)$, $y = \mathbf{s}(\mathbf{0} \cdot \mathbf{-1} + \mathbf{1} \cdot \mathbf{-2} + \mathbf{1} \cdot \mathbf{2}) = \mathbf{s}(\mathbf{0}) = \mathbf{0.5}$

(c) $x = (1, 0)$, $y = \mathbf{s}(\mathbf{0} \cdot \mathbf{-1} + \mathbf{1} \cdot \mathbf{1} + \mathbf{1} \cdot \mathbf{0}) = \mathbf{s}(\mathbf{1}) = \mathbf{0.73}$

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) $\Delta w_2 =$

Solution:

$$\begin{aligned}
\Delta w_2 &= -\eta \frac{\partial E}{\partial w_2} \\
&= -\eta \frac{\partial E}{\partial y} \frac{\partial y}{\partial z} \frac{\partial z}{\partial w_2} \\
&= -\eta (y - y^i) y (1 - y) x_2 \\
&= (-1)(0.5 + 0)(0.5)(0.5)(-2) \\
&= 0.25
\end{aligned}$$

Derivations:

$$\begin{aligned}
E &= \frac{1}{2}(y - y^i)^2 \\
y &= s(z) \\
z &= \sum_{i=0}^{2} w_i x_i \\
\frac{\partial E}{\partial y} &= y - y^i \\
\frac{\partial y}{\partial z} &= y(1 - y) \\
\frac{\partial z}{\partial w_2} &= x_2
\end{aligned}$$

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.

   - $\Pr(x_1 = 1|y = 0) = \frac{6+1}{6+2} = \frac{7}{8}$

   - $\Pr(x_2 = 1|y = 1) = \frac{0+1}{6+2} = \frac{1}{8}$

   - $\Pr(x_3 = 0|y = 0) = 1 - \frac{2+1}{6+2} = \frac{5}{8}$

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

   x_3**, because it has the biggest difference in the likelihood of being true for the two different classes. The other two features carry no information about the class.**

6 Learning algorithms (16 pts)

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.

   Nearest Neighbors is a good choice. The dimensionality is low and so appropriate for KNN. For KNN, training is very fast and since there are few classifications, the fact that this will be slow does not matter. With 1 million training examples, neural net and SVM will be extremely expensive to train. Naive Bayes is plausible on computational grounds but likely to be less accurate than KNN.

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.

   A Decision Tree is a good choice since the resulting classifier will need to be understandable to humans.

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.

   Naive Bayes is a good choice since it is fast to train and update. The dimensionality is high for Nearest Neighbors and Decision Trees. SVM's have to be re-trained from scratch if the data changes. Neural Nets could be trained incrementally but it will generally take a lot of iterations to change the current settings of the weights.

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.

   This is a regression problem; neural nets with linear output functions, regression trees or locally weighted nearest neighbors are all appropriate choices.

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

   A. It uses a fixed hypothesis class of linear separators. It is guaranteed to find a separator if one exists.

2. linear SVM

   B. It includes a complexity penalty in the error criterion (which is to maximize the margin while separating the data). It optimizes this criterion.

3. decision tree with fixed depth

   A. It uses a fixed hypothesis class, which is the class of fixed-depth trees. Implicitly, it tries to find the lowest-error tree within this class, but isn't guaranteed to optimize that criterion.

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

   A. It uses a fixed hypothesis class, which is determined by the wiring diagram of the network.

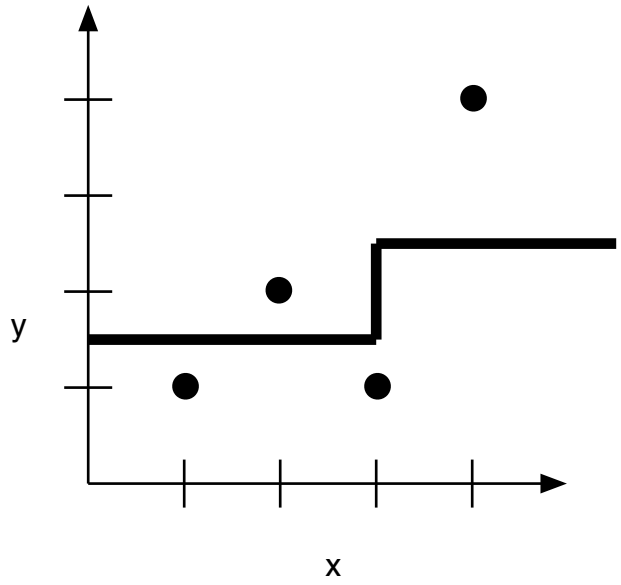
5. SVM (with arbitrary data and $c < \infty$)

   B. It includes a complexity penalty in the error criterion (which is to maximize the margin subject to assigning an $\alpha < c$ to each data point.

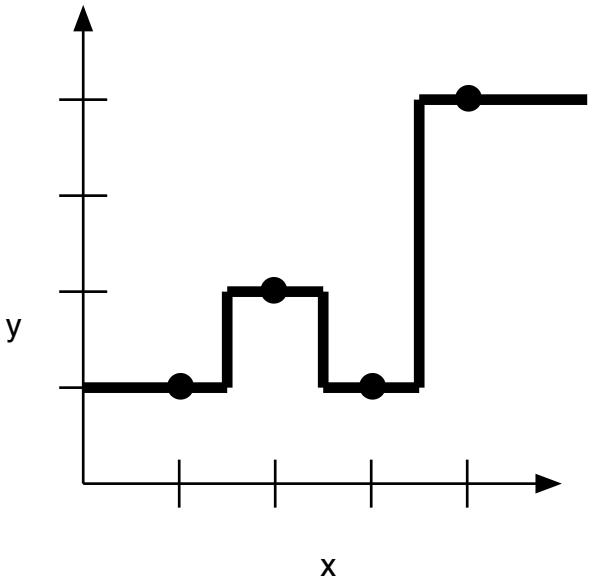
8 Regression (12 pts)

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.

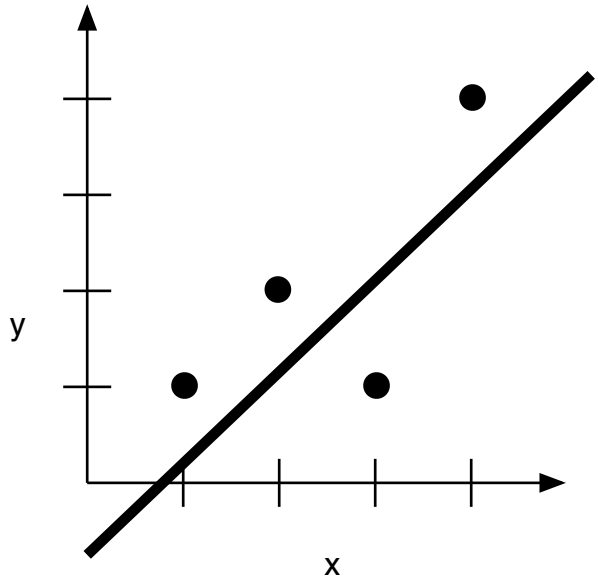
1. 2-nearest-neighbor (equally weighted averaging)



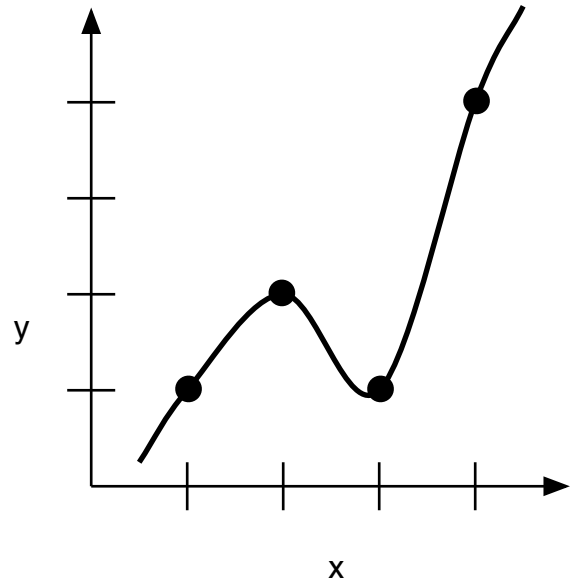
2. regression trees (with leaf size 1)



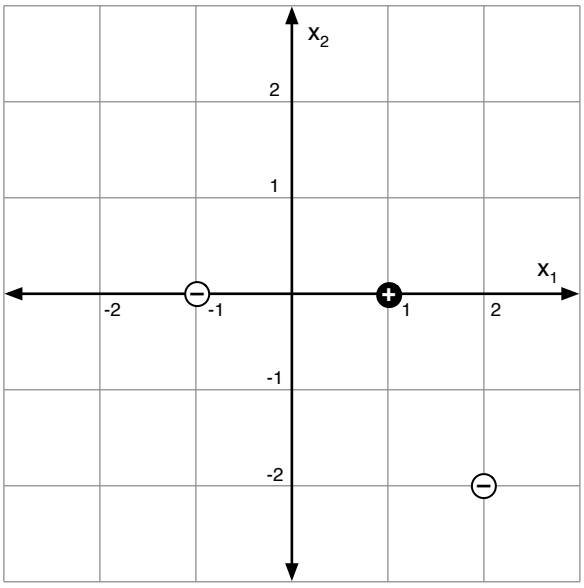
3. one linear neural-network unit



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) $x_1 + x_2 = 0$

   Goes through the (2,-2) point so obviously not maximal margin.

   (b) $x_1 + 1.5x_2 = 0$

   Yes. All three points are support vectors, with margin = 1.

   (c) $x_1 + 2x_2 = 0$

   No. Three points are needed to define a line, with two support vectors there is no unique maximal margin line.

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

   No. The margin for the points is 2, not 1

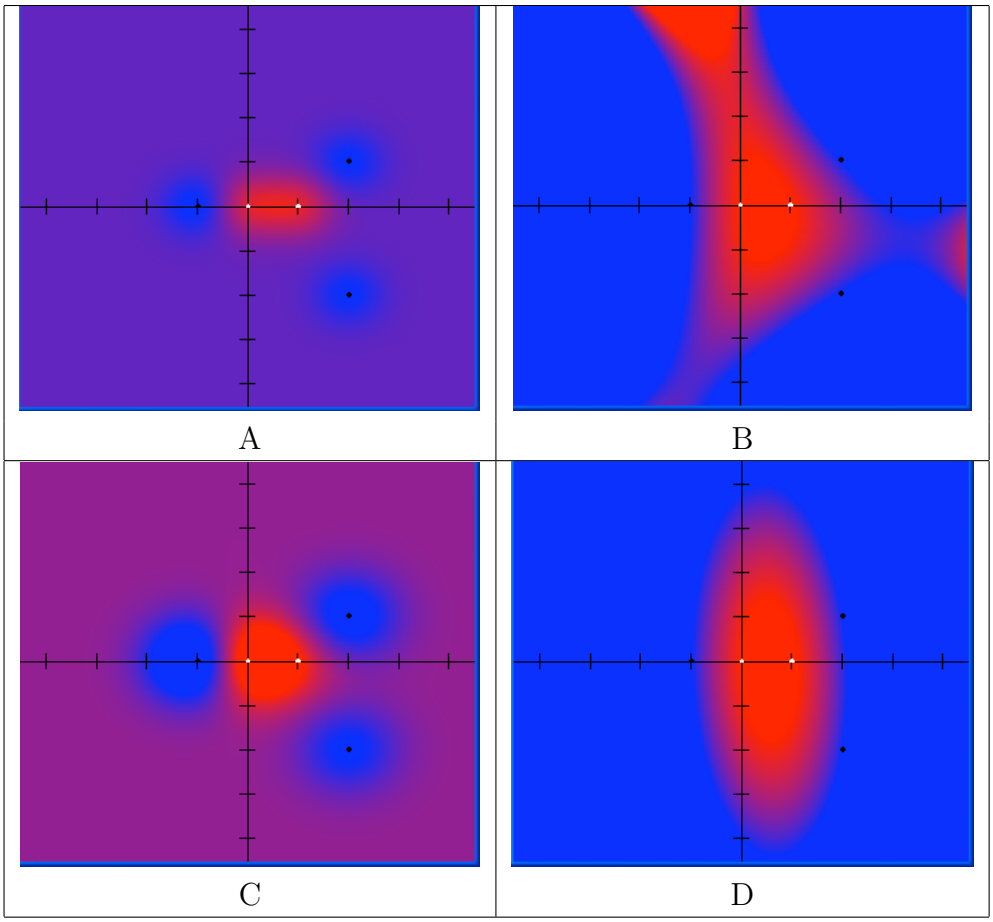
2. For each of the kernel choices below, find the decision boundary diagram (on the next page) that best matches. In these diagrams, the brightness of a point represents the magnitude of the SVM output; red means positive output and blue means negative. The black circles are the negative training points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

   (a) Polynomial kernel, degree 2 : **D**

   (b) Polynomial kernel, degree 3 : **B**

   (c) Radial basis kernel, sigma = 0.5 : **A**

   (d) Radial basis kernel, sigma = 1.0 : **C**


A
B
C
D