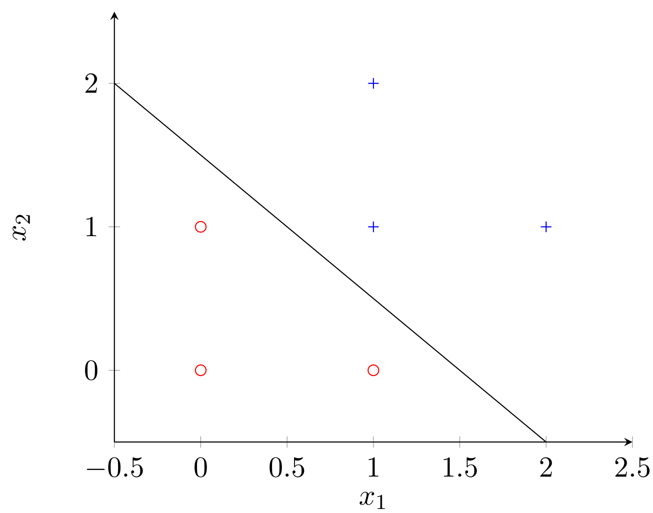
CMPS 142 - Homework 4

Emily Hockel - echockel@ucsc.edu
Christopher Hsiao - chhsiao@ucsc.edu

May 29, 2017

1 Determine Support Vectors



By observing the plot, we see that the support vectors are:

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

We use the formula $y^{(i)}(w \cdot x^{(i)} + b) = 1$ to find the maximum margin separating plane. We solve for w and b to find $x_2 = 1.5 - x_1$, where $w = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$ and $b = -3$.

We use the formula $\frac{y^{(i)}(w \cdot x^{(i)} + b)}{||w||}$ to find that the geometric margin for the support vectors is $\frac{\sqrt{2}}{4}$.

2 Support Vector Machine Kernels

For each of the following problems, we used 10-fold cross validation.

2.1 RBF Kernel

C \ γ	0.01	0.1	1	10
0.1	73.1363	87.5027	89.3719	64.2469
1	87.2202	92.3712	93.0667	85.1771
10	92.1539	93.6753	93.2189	85.8726
100	93.3493	93.8926	92.5668	85.8292
1000	93.5884	93.4362	91.4584	85.6336

We see that the values $C = 1000$ and $\gamma = 0.01$ give us the highest accuracy of 93.59%. For this data set, a narrow gaussian and a high margin penalty give us the best results.

2.2 Polynomial Kernel

C \ γ	0.01	1	10	100
0.1				
1		81.1997	91.2193	
10		87.394	92.7842	91.9365
100	60.5955	91.2193	92.4364	91.2628
1000			91.9365	

With this different kernel, we get just slightly lower accuracy than the RBF kernel. Parameter values of $C = 100$ and $\gamma = 10$ gives us an accuracy of 92.44%.

2.3 Linear SVM

C	
0.1	85.1989
1	90.3282
10	92.2408
100	92.9363
1000	93.0233
10000	92.9363

With a linear SVM, we get our highest accuacy of 93.02% with a value of $C = 1000$.

3 Greedy Decision Trees

We built a dataset based on the logical statement:

$$(A \land B) \lor (\neg A \land C) \tag{1}$$

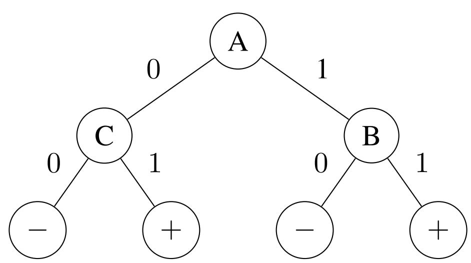
3.1 The Dataset

Note that the last entry in the dataset has been repeated to demonstrate the pitfalls of the greedy top-down decision tree generation algorithm.

A	B	C	label
0	0	0	$-$
1	0	0	$-$
0	1	0	$-$
0	0	1	$+$
1	1	0	$+$
1	0	1	$-$
0	1	1	$+$
1	1	1	$+$
1	1	0	$+$

3.2 The Optimal Tree

First, let us take a look at the optimal decision tree.



This tree has seven nodes, with four leaf nodes representing the predictions.

3.3 The Greedy Top-Down Tree

For the first iteration of the algorithm, let us see the split for each of the three attributes.
Note that we are using the error rate function, which is the fraction of predictions that are **not** in the majority of predicted values.

3.3.1 First Iteration Predictions

Below, we have the first iteration predictions and error rates for each attribute value.

- **A = 0**: $\{-, -, +, +\}$, error: $\frac{2}{4}$

- **A = 1**: $\{-, -, +, +, +\}$, error: $\frac{2}{5}$
- **B = 0**: $\{-, -, -, +\}$, error: $\frac{1}{4}$
- **B = 1**: $\{-, +, +, +, +\}$, error: $\frac{1}{5}$
- **C = 0**: $\{-, -, -, +, +\}$, error: $\frac{2}{5}$
- **C = 1**: $\{-, +, +, +\}$, error: $\frac{1}{4}$

The total error across the three attributes are:

$$\textbf{A} = 0.9; \textbf{B} = 0.45; \textbf{C} = 0.65$$

Since **B** has the lowest error, we'll choose **B** as our attribute to split on.

3.3.2 Second Iteration Predictions: Split on B

Now, we find the second iteration predictions and error rates for the remaining attribute values **A** and **C**.

- **B = 0; A = 0**: $\{-, +\}$, error: $\frac{1}{2}$
- **B = 0; A = 1**: $\{-, -\}$, error: 0
- **B = 1; A = 0**: $\{-, +\}$, error: $\frac{1}{2}$
- **B = 1; A = 1**: $\{+, +, +\}$, error: 0
- **B = 0; C = 0**: $\{-, -\}$, error: 0
- **B = 0; C = 1**: $\{+, -\}$, error: $\frac{1}{2}$
- **B = 1; C = 0**: $\{-, +, +\}$, error: $\frac{1}{3}$
- **B = 1; C = 1**: $\{+, +\}$, error: 0

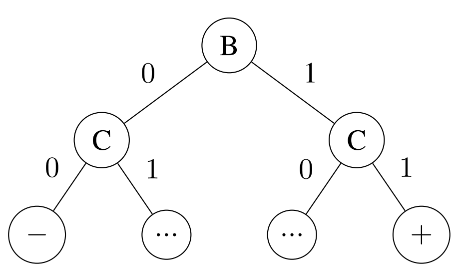
The total error across the two attributes are:

$$\textbf{A} = 1; \textbf{C} = 0.83$$

Since **C** has the lowest error, we'll choose **C** as our attribute to split on. Note that we split on both branches of **B**, so we create two **C** nodes.

3.3.3 The Final Iteration

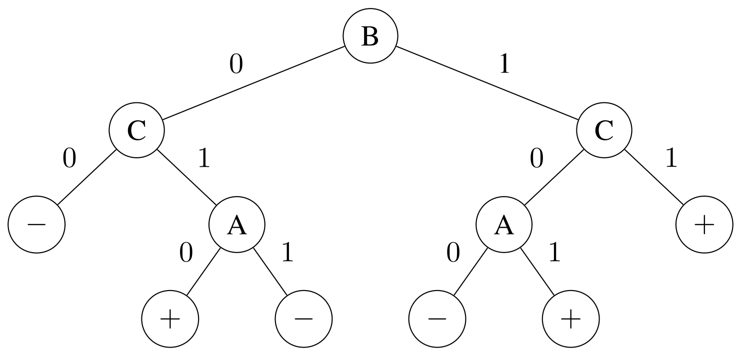
This is what the tree now looks like, after the second iteration:



Now all that is left is to add the splits on **A**, for the last four sequences:

$$\{B = 0; C = 1; A = 0\} = +$$
$$\{B = 0; C = 1; A = 1\} = -$$
$$\{B = 1; C = 0; A = 0\} = -$$
$$\{B = 1; C = 0; A = 1\} = +$$

Adding these last few nodes to our tree, we get the full decision tree:



This is the decision tree output by the greedy top-down algorithm. This tree has 11 nodes, and 6 leaf nodes. Recall from **3.2** that the optimal tree has 7 nodes, with 4 leaf nodes. Clearly, this tree is not optimal.