Kai Wong (704451679)

CS 145 Fall 2018

Homework 2

1.1: Decision Tree Construction

Let class P: Democrat, class N: Republican

Vote for handicapped-infants	p_i	n_i	$I(p_i, n_i)$
Y	6	2	0.811
N	4	8	0.918

Vote for water-project	p_i	n_i	$I(p_i, n_i)$
Y	4	6	0.971
N	6	4	0.971

Vote for budget-resolution	p_i	n_i	$I(p_i, n_i)$
Y	9	2	0.684
N	1	8	0.503

Info(D) = $-\frac{10}{20}log_2\left(\frac{10}{20}\right)-\frac{10}{20}log_2\left(\frac{10}{20}\right)$ = 1.0

$\text{Info}_{\text{handicapped}}(D) = \frac{8}{20}I(6,2)+\frac{12}{20}I(4,8)$ = 0.875

$\text{Info}_{\text{water-project}}(D) = \frac{10}{20}I(4,6)+\frac{10}{20}I(6,4)$ = 0.971

$\text{Info}_{\text{budget-resolution}}(D) = \frac{11}{20}I(9,2)+\frac{9}{20}I(1,8)$ = 0.603

Gain(handicapped) = Info(D) - $\text{Info}_{\text{handicapped}}(D)$ = 0.125

Gain(water-project) = Info(D) - $\text{Info}_{\text{water-project}}(D)$ = 0.029

Gain(budget-resolution) = Info(D) - $\text{Info}_{\text{budget-resolution}}(D)$ = 0.397

$\text{Info}(D_{\text{budget-resolution=Y}}) = I(9,2)$ = 0.684

$\text{Info}_{\text{handicapped}}(D_{\text{budget-resolution=Y}}) = \frac{7}{11}I(6,1)+\frac{4}{11}I(3,1)$ = 0.672

$\text{Info}_{\text{water-project}}(D_{\text{budget-resolution=Y}}) = \frac{4}{11}I(3,1)+\frac{7}{11}I(6,1)$ = 0.672

$\text{Gain}_{\text{budget-resolution=Y}}$(handicapped) = 0.012

$\text{Gain}_{\text{budget-resolution=Y}}$(water-project) = 0.012

$\text{Info}(D_{\text{budget-resolution=N}}) = I(1,8)$ = 0.503

$\text{Info}_{\text{handicapped}}(D_{\text{budget-resolution=N}}) = \frac{1}{9}I(0,1)+\frac{8}{9}I(1,7)$ = 0.483

$\text{Info}_{\text{water-project}}(D_{\text{budget-resolution=N}}) = \frac{6}{9}I(1,5)+\frac{3}{9}I(0,3)$ = 0.433

$\text{Gain}_{\text{budget-resolution=N}}$(handicapped) = 0.020

$\text{Gain}_{\text{budget-resolution=N}}$(water-project) = 0.070

This is the resulting decision tree using information gain as the attribute selection heuristic as computed above. This tree is pruned (when all leaf nodes of a parent node evaluate to the same value V, subtree starting from parent node is pruned from main tree and parent node is replaced with V).

Tree pruning was performed for the following:

Budget -> N -> Republican:

Budget -> N -> Water -> N -> Republican

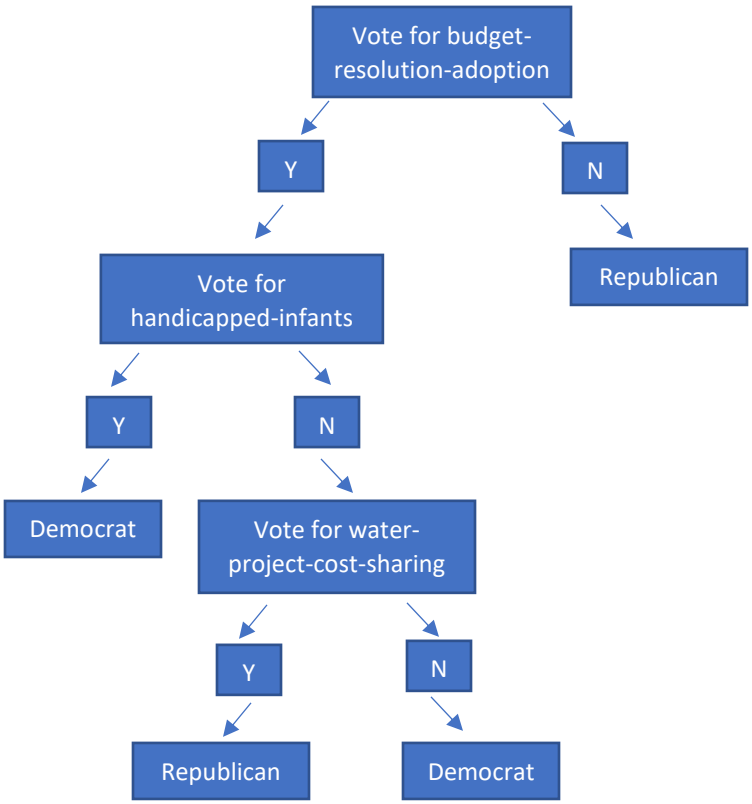
Budget -> N -> Water -> Y -> Handicap -> Y -> Republican

Budget -> N -> Water -> Y -> Handicap -> N -> Republican (By majority voting)

Budget -> Y -> Handicap -> Y -> Democrat:

Budget -> Y -> Handicap -> Y -> Water -> Y -> Democrat

Budget -> Y -> Handicap -> Y -> Water -> N -> Democrat (By majority voting)



1.2(a): Information Gain

```
C:\Users\kyles\Desktop\CS145\hw2\HW2 Programming\DecisionTree
λ python DecisionTree.py 0
Attribute Selection Criterion:  0
best_feature is:  legs
best_feature is:  fins
best_feature is:  toothed
best_feature is:  eggs
best_feature is:  hair
best_feature is:  hair
best_feature is:  toothed
best_feature is:  aquatic
{'legs': {0: {'fins': {0.0: {'toothed': {0.0: 7.0, 1.0: 3.0}},
                       1.0: {'eggs': {0.0: 1.0, 1.0: 4.0}}}},
          2: {'hair': {0.0: 2.0, 1.0: 1.0}},
          4: {'hair': {0.0: {'toothed': {0.0: 7.0, 1.0: 5.0}}, 1.0: 1.0}},
          6: {'aquatic': {0.0: 6.0, 1.0: 7.0}},
          8: 7.0}}
Test accuracy:  0.8571428571428571
```

1.2(b): Gain Ratio

```
C:\Users\kyles\Desktop\CS145\hw2\HW2 Programming\DecisionTree
λ python DecisionTree.py 1
Attribute Selection Criterion:  1
best_feature is:  feathers
best_feature is:  backbone
best_feature is:  airborne
best_feature is:  predator
best_feature is:  milk
best_feature is:  fins
best_feature is:  legs
{'feathers': {0: {'backbone': {0.0: {'airborne': {0.0: {'predator': {0.0: 6.0,
                                                                     1.0: 7.0}},
                                                  1.0: 6.0}},
                               1.0: {'milk': {0.0: {'fins': {0.0: {'legs': {0.0: 3.0,
                                                                            4.0: 5.0}},
                                                             1.0: 4.0}},
                                             1.0: 1.0}}}},
              1: 2.0}}
Test accuracy:  0.8095238095238095
```

The accuracy when using information gain as the attribute selection heuristic was 0.857 whereas using gain ratio gave an accuracy of 0.810. A possible reason for the decreased performance when using gain ratio is because while information gain is typically biased towards multivalued attributes, almost all of the attributes in the dataset were binary except for 'legs', hence it would not be necessary to apply gain ratio to overcome these biases from using information gain.

Furthermore, gain ratio tends to prefer unbalanced splits in which one partition is much smaller than the others (since smaller partitions would have a smaller SplitInfo => larger gain ratio). As a result, it may overcompensate and choose an attribute just because its intrinsic information is low despite having lower than average information gain. This can be seen in the feature selection choice: information gain first selected 'legs' as the best feature whereas gain ratio selected 'feathers' when intuitively, it would not be hard to imagine why partitioning animals based on their number of legs first would be a better choice over if they had feathers or not. Hence, information gain would be a better attribute selection measure than gain ratio for this dataset.

2.1(a): Please point out the support vectors in the training points

Each non-zero α_i indicates that the corresponding x_i is a support vector.

Hence, the support vectors are points 2, 6, and 18

2.1(b): Calculate the normal vector of the hyperplane

$$\omega = \Sigma a_i y_i x_i$$
$$= \alpha_2 y_2 x_2 + \alpha_6 y_6 x_6 + \alpha_{18} y_{18} x_{18}$$
$$= (0.5084 \times 0.91, 0.5084 \times 0.32)^T + (0.4625 \times 0.41, 0.4625 \times 2.04)^T + (-0.9709 \times 2.05, -0.9709 \times 1.54)^T$$
$$= (-1.3381, -0.3890)^T$$

2.1(c): Calculate the bias b

$$b = (y_2 - \omega^T x_2)/3 + (y_6 - \omega^T x_6)/3 + (y_{18} - \omega^T x_{18})/3$$
$$= 2.3421$$

2.1(d): Write the learned decision boundary function

$$f(x) = (-1.3381, -0.3890)\mathrm{x} + 2.3421$$

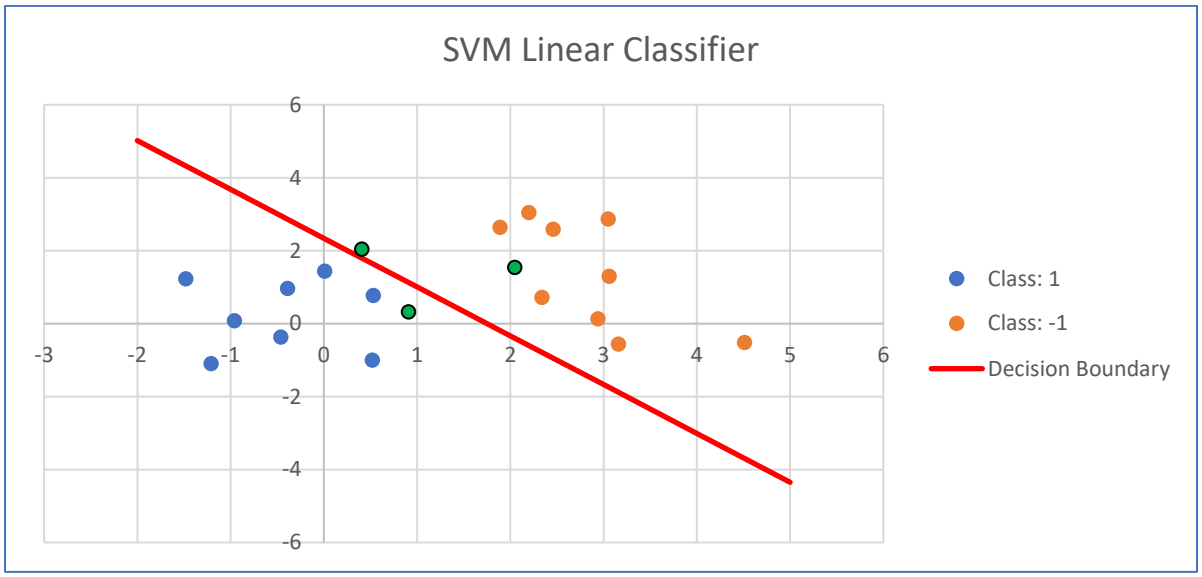
2.1(e): Predict class label of new data point

$$f(x) = (-1.3381, -0.3890)(-1,2)^T + 2.3421$$

$$= 2.9022 > 0$$

Hence, the class label for the new data point is 1.

2.1(f): Plot data points and decision boundary line



Green data points are the support vectors.

2.2(a): Hard margin and soft margin SVM for linear classifier

```
C:\Users\kyles\Desktop\CS145\hw2\HW2 Programming\SVM
λ python svm.py 0 0
Margin:  0
kernel Type:  0
     pcost       dcost       gap    pres   dres
 0: -1.4438e+02 -3.3911e+02  5e+03  6e+01  2e+00
 1: -3.8214e+02 -5.6098e+02  3e+03  4e+01  1e+00
 2: -6.6231e+02 -9.2563e+02  3e+03  4e+01  1e+00
 3: -1.3110e+03 -1.7230e+03  3e+03  4e+01  1e+00
 4: -2.7029e+03 -3.4114e+03  4e+03  3e+01  1e+00
 5: -9.7014e+03 -1.1663e+04  4e+03  3e+01  1e+00
 6: -3.6797e+04 -4.2248e+04  6e+03  3e+01  1e+00
 7: -1.1615e+05 -1.2907e+05  1e+04  3e+01  1e+00
 8: -5.5869e+05 -5.9758e+05  4e+04  3e+01  1e+00
 9: -1.7063e+06 -1.8091e+06  1e+05  3e+01  1e+00
10: -2.8334e+07 -2.8527e+07  2e+05  3e+01  1e+00
11: -3.5391e+09 -3.5429e+09  4e+06  3e+01  1e+00
12: -7.1603e+09 -7.1677e+09  7e+06  3e+01  1e+00
13: -9.3453e+09 -9.3547e+09  9e+06  3e+01  1e+00
14: -1.6524e+10 -1.6540e+10  2e+07  3e+01  1e+00
15: -2.7493e+10 -2.7518e+10  3e+07  3e+01  1e+00
Terminated (singular KKT matrix).
1098 support vectors out of 1098 points
Test accuracy:  0.5547445255474452
```

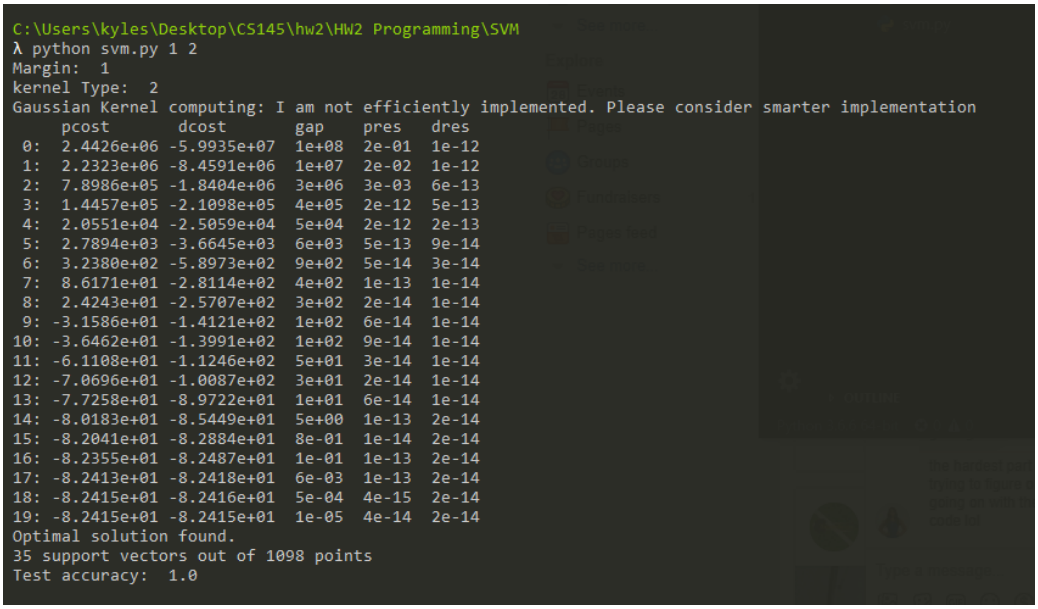
Hard margin SVM with a linear classifier terminates before finding an optimal solution because the training data is not linearly separable. As a result, a singular KKT matrix is generated by the cvxopt quadratic programming solvers which cannot be used to find the Lagrange multipliers.

```
C:\Users\kyles\Desktop\CS145\hw2\HW2 Programming\SVM
λ python svm.py 1 0
Margin:  1
kernel Type:  0
     pcost       dcost       gap    pres   dres
 0: -1.7838e+04 -1.4216e+08  4e+08  6e-01  2e-10
 1:  2.0347e+04 -3.4085e+07  6e+07  8e-02  2e-10
 2:  3.5993e+04 -9.5151e+06  2e+07  2e-02  1e-10
 3:  4.9596e+04 -6.0614e+06  1e+07  1e-02  7e-11
 4:  5.3650e+04 -3.2836e+06  5e+06  6e-03  5e-11
 5:  5.2427e+04 -1.2117e+06  2e+06  1e-03  4e-11
 6:  2.1260e+04 -2.6704e+05  3e+05  3e-06  4e-11
 7: -1.0283e+03 -9.9499e+04  1e+05  9e-07  4e-11
 8: -4.8524e+03 -4.9722e+04  4e+04  4e-07  4e-11
 9: -5.2399e+03 -4.9623e+04  4e+04  4e-07  4e-11
10: -6.3385e+03 -2.2126e+04  2e+04  9e-08  4e-11
11: -6.8781e+03 -2.2626e+04  2e+04  9e-08  4e-11
12: -7.6981e+03 -2.1745e+04  1e+04  7e-08  4e-11
13: -6.7996e+03 -2.0773e+04  1e+04  7e-08  4e-11
14: -8.1136e+03 -1.6840e+04  9e+03  2e-08  5e-11
15: -8.7965e+03 -1.6725e+04  8e+03  1e-08  5e-11
16: -8.6221e+03 -1.6538e+04  8e+03  1e-08  4e-11
17: -8.8831e+03 -1.6609e+04  8e+03  9e-09  5e-11
18: -9.2486e+03 -1.5360e+04  6e+03  6e-09  4e-11
19: -9.2427e+03 -1.4163e+04  5e+03  3e-09  5e-11
20: -9.4498e+03 -1.2471e+04  3e+03  2e-09  5e-11
21: -9.5107e+03 -1.2150e+04  3e+03  1e-09  5e-11
22: -9.5965e+03 -1.1545e+04  2e+03  4e-10  5e-11
23: -9.5977e+03 -1.1531e+04  2e+03  4e-10  5e-11
24: -9.7482e+03 -1.1573e+04  2e+03  2e-10  5e-11
25: -1.0001e+04 -1.0944e+04  9e+02  7e-11  5e-11
26: -1.0111e+04 -1.0701e+04  6e+02  3e-11  5e-11
27: -1.0304e+04 -1.0347e+04  4e+01  9e-13  5e-11
28: -1.0323e+04 -1.0324e+04  5e-01  8e-12  5e-11
29: -1.0324e+04 -1.0324e+04  5e-03  4e-12  5e-11
Optimal solution found.
34 support vectors out of 1098 points
Test accuracy:  0.9890510948905109
```

On the other hand, soft margin SVM with a linear classifier rectifies this problem by adding slack variables which supports misclassification of difficult or noisy training data. These slack variables allow misclassified data points to be 'moved' to their correct classes, but at a cost. As a result, the algorithm can find an optimal solution.

2.2(b): Different kernels for soft margin SVM

```
C:\Users\kyles\Desktop\CS145\hw2\HW2 Programming\SVM
λ python svm.py 1 1
Margin:  1
kernel Type:  1
     pcost       dcost       gap    pres   dres
 0: -8.9993e+02 -7.6966e+07  2e+08  4e-01  2e-06
 1:  2.0423e+04 -1.4910e+07  3e+07  5e-02  2e-06
 2:  2.6926e+04 -1.6663e+06  3e+06  5e-03  7e-07
 3:  1.3337e+04 -2.0412e+05  4e+05  5e-04  1e-07
 4:  4.4128e+03 -5.4525e+04  9e+04  1e-04  2e-08
 5:  1.2674e+03 -1.3679e+04  2e+04  2e-05  4e-09
 6:  4.1532e+02 -3.8486e+03  6e+03  5e-06  9e-10
 7:  1.4911e+02 -1.3070e+03  2e+03  1e-06  3e-10
 8:  5.6834e+01 -4.4006e+02  6e+02  4e-07  1e-10
 9:  1.8764e+01 -1.0151e+02  1e+02  7e-08  3e-11
10:  5.8413e+00 -2.3618e+01  3e+01  1e-08  7e-12
11:  1.1303e+00 -2.7148e+00  4e+00  2e-16  1e-12
12:  1.4499e-01 -5.6055e-01  7e-01  2e-16  6e-13
13: -2.4286e-02 -4.2777e-01  4e-01  1e-16  3e-13
14: -1.1167e-01 -2.2497e-01  1e-01  2e-16  2e-13
15: -1.3095e-01 -1.9550e-01  6e-02  2e-16  1e-13
16: -1.4060e-01 -1.7556e-01  3e-02  2e-16  8e-14
17: -1.4784e-01 -1.6171e-01  1e-02  2e-16  1e-13
18: -1.4954e-01 -1.5895e-01  9e-03  2e-16  1e-13
19: -1.5208e-01 -1.5520e-01  3e-03  2e-16  1e-13
20: -1.5314e-01 -1.5374e-01  6e-04  2e-16  1e-13
21: -1.5336e-01 -1.5345e-01  9e-05  2e-16  1e-13
22: -1.5340e-01 -1.5340e-01  1e-06  2e-16  1e-13
23: -1.5340e-01 -1.5340e-01  1e-08  2e-16  1e-13
Optimal solution found.
19 support vectors out of 1098 points
Test accuracy:  0.927007299270073
```



I have the following accuracy results:

Soft margin SVM with linear kernel: 0.989

Soft margin SVM with polynomial kernel: 0.927

Soft margin SVM with gaussian kernel: 1.000

Based on this data, it appears that using the gaussian kernel gives the best model for this dataset and is thus the model I would want to choose. However, 100% accuracy is practically unrealistic and may indicate overfitting or insufficient size/complexity of the training dataset. Nonetheless, for the purposes of this assignment soft margin SVM with gaussian kernel is the best model for this dataset.