CS534 - HW 3

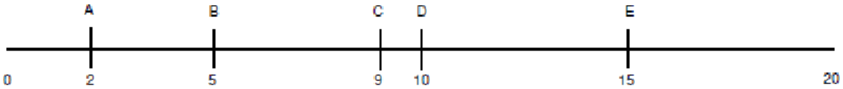
Keith Chester

Due date: July 10th 2022

Problem 1

In this problem we are exploring K-means clustering.

We have a set of data representing five customer ratings on a new car, with a scale from 0 to 20. The image below represents that data:



Part A

Assuming that $K = 2$, and the two initial centroids are 3 and 4, we will try to utilize the K-means algorithm to show all computational steps.

$$K = [2, 5, 9, 10, 15];\ C_0 = 3;\ C_1 = 4 \tag{1}$$

Next we find the distance (simple, since we're working with a singular dimension) from each point in order to assign it.

Data point	Distance from C_0	Distance from C_1
2	**1**	2
5	2	**1**
9	6	**5**
10	7	**6**
15	12	**11**

Here we've bolded the closer distance to the associated centroid. We will now recompute the resulting centroid position by taking the mean value of all points assigned.

$$C_i = \frac{1}{n} \sum_{x_i \to C_i} x_i \tag{2}$$

$$C_0 = \frac{1}{n} \sum_{x_i \to C_0} x_i = \frac{1}{1} \sum (2) = 2 \tag{3}$$

$$C_1 = \frac{1}{n} \sum_{x_i \to C_1} x_i = \frac{1}{4} \sum (5, 9, 10, 15) = 9.75 \tag{4}$$

This gives us two new centroids at $C_0 = 2$ and $C_1 = 9.75$. We continue this process until we no longer have any points change centroid assignments.

Data point	Distance from C_0	Distance from C_1
2	**0**	7.75
5	**3**	4.75
9	7	**0.75**
10	8	**0.25**
15	13	**5.25**

...since we have a new point on the first centroid, we run the calculations to get the new centroid averages again:

$$C_0 = \frac{1}{n} \sum_{x_i \to C_0} x_i = \frac{1}{2} \sum (2, 5) = 3.50 \tag{5}$$

$$C_1 = \frac{1}{n} \sum_{x_i \to C_1} x_i = \frac{1}{3} \sum (9, 10, 15) = 11.33 \tag{6}$$

...leaving us with $C_0 = 3.50$ and $C_1 = 11.33$. Continuing the process:

Data point	Distance from C_0	Distance from C_1
2	**0**	9.33
5	**3**	6.333
9	5.50	**2.33**
10	6.50	**0.67**
15	11.50	**3.67**

Since we see no point changes, so our centroids are finalized at $C_0 = 3.50$ and $C_1 = 11.33$.

To find the Silhouette Coefficient Index, we take a given cluster C_i, select a point x_i within it, and find the average distance distance between it and all other points within that cluster, which we'll mark as a_1. Then for each other cluster C_j, find the *minimum* average distance (a_j) between this point and the average distance from each point in the other clusters.
...choosing $x_i = x_0 = 2$ for our point

$$a_1 = 3 \tag{7}$$

...we need to test both x_0 and x_1 to find the minimum average distance to utilize for our calculation.

$$a_2 = \frac{1}{n} \sum d(x_0, C_1); = \frac{1}{3} \sum (7, 8, 13) = 9.33 \tag{8}$$

$$a_2 = \frac{1}{n} \sum d(x_1, C_1); = \frac{1}{3} \sum (4, 5, 10) = 6.33 \tag{9}$$

...which, since we have only the one other cluster, we can stop here. We'll use the smaller 6.33 average distance. If we had more clusters, we would take the minimum of the average distances from each point in other clusters, but we can stop here. We can now calculate the index S_1:

$$S_1 = 1 - \frac{a_1}{a_2} = 1 - \frac{3}{6.33} = 0.53 \tag{10}$$

...and then we do the same for the next cluster C_2. We use $x_i = x_3 = 9$.

$$a_1 = \frac{1}{n} \sum d(x_2, C_1); = \frac{1}{2}(1, 6) = 3.5 \tag{11}$$

...we need to test both x_2, x_3 and x_4 to find the minimum average distance to utilize for our calculation.

$$a_2 = \frac{1}{n} \sum d(x_2, C_0); = \frac{1}{2}(7, 4) = 5.5 \tag{12}$$

$$a_2 = \frac{1}{n} \sum d(x_3, C_0); = \frac{1}{2}(8, 5) = 6.5 \tag{13}$$

$$a_2 = \frac{1}{n} \sum d(x_4, C_0); = \frac{1}{2}(12, 10) = 11 \tag{14}$$

...using $a_2 = 5.5$ as our minimum.

$$S_2 = 1 - \frac{a_1}{a_2} = 1 - \frac{3.5}{5.5} = 0.37 \tag{15}$$

...so the Silhouette Coefficient Index for our clusters C_0 and C_1 are $S_1 = 0.76$ and $S_2 = 0.37$, respectively.

We are now going to calculate the Davies-Bouldin Index. To find this, we will be first finding the average distance from a given cluster's C_i's points to its centroid (s_i), the distance between the centroids of two given clusters (d_{ij}), and then the max $R_{i,j}$ where...

$$R_{i,j} = \frac{s_1 + s_2}{d_{ij}} \tag{16}$$

...then we find the max $R_{i,j}$ (which will be the only one we calculate as we only have two clusters), and then take the average of those maxes - will again be just R_{ij} given our setup.

$$D_i = \max(R_{i,j}) \tag{17}$$

$$DB_{index} = \frac{1}{N} \sum_{i=1}^{N} D_i \tag{18}$$

...and now with our values:

$$s_1 = \frac{1}{2} \sum (1.5, 1.5) = 1.5 \tag{19}$$

$$s_2 = \frac{1}{3} \sum (2.33, 1.33, 3.67) = 2.44 \tag{20}$$

$$R_{0,1} = \frac{1.5 + 2.44}{7.83} = 0.50 \tag{21}$$

Finally we are moving onto the Calinski-Harabasz Index. To calculate this, we will be utilizing the following equation:

$$CH = \left[\frac{\sum_{k=1}^{K} n_k \|c_k - C\|^2}{K-1} \right] / \left[\frac{\sum_{k=1}^{K} \sum_{i=1}^{n_k} \|d_i - c_k\|^2}{N-K} \right] \tag{22}$$

...where CH is our index, C is the global centroid, K is the total number of clusters, N is the number of points, and d_i is a given data point within a cluster. n_k and c_k are the number of points within and the centroid of cluster k, respectively.

$$C = \frac{1}{N} \sum (d_i) = \frac{1}{5} \sum (2, 5, 9, 10, 15) = 8.2 \tag{23}$$

$$CH = \frac{\left[\frac{\sum(2\|3.5-8.2\|^2, 3\|11.33-8.2\|^2)}{1} \right]}{\left[\frac{\sum(\sum(\|2-3.5\|^2, \|5-3.5\|^2), \sum(\|9-11.33\|^2, \|10-11.33\|^2, \|15-11.33\|^2))}{5-2} \right]} \tag{24}$$

$$CH = \frac{\left[\frac{\sum(2*22.09, 3*9.80)}{1} \right]}{\left[\frac{\sum(\sum(2.25, 2.25), \sum(5.43, 1.77, 13.47))}{3} \right]} \tag{25}$$

$$CH = \frac{\left[\frac{\sum(44, 28.83)}{1} \right]}{\left[\frac{\sum(\sum(2.25, 2.25), \sum(5.43, 1.77, 13.47))}{3} \right]} \tag{26}$$

$$CH = \frac{\left[\frac{72.83}{1} \right]}{\left[\frac{\sum(4.5, 20.67)}{3} \right]} \tag{27}$$

$$CH = \frac{[72.83]}{\left[\frac{25.17}{3} \right]} \tag{28}$$

$$CH = 8.68 \tag{29}$$

Part B

We are doing that again, for some reason. This time, however, we'll be starting with the initial centroids of 11 and 12.

$$K = [2, 5, 9, 10, 15];\ C_0 = 11;\ C_1 = 12 \tag{30}$$

Next we find the distance (simple, since we're working with a singular dimension) from each point in order to assign it.

Data point	Distance from C_0	Distance from C_1
2	**9**	10
5	**6**	7
9	**2**	3
10	**1**	2
15	4	**3**

Here we've bolded the closer distance to the associated centroid. We will now recompute the resulting centroid position by taking the mean value of all points assigned.

$$C_i = \frac{1}{n} \sum_{x_i \to C_i} x_i \tag{31}$$

$$C_0 = \frac{1}{n} \sum_{x_i \to C_0} x_i = \frac{1}{4} \sum (2, 5, 9, 10) = 6.5 \tag{32}$$

$$C_1 = \frac{1}{n} \sum_{x_i \to C_1} x_i = \frac{1}{1} \sum (15) = 15 \tag{33}$$

This gives us two new centroids at $C_0 = 6.5$ and $C_1 = 15$. We continue this process until we no longer have any points change centroid assignments.

Data point	Distance from C_0	Distance from C_1
2	**4.5**	13
5	**1.5**	10
9	**2.5**	6
10	**3.5**	5
15	8.5	**0**

Since we see no point changes, so our centroids are finalized at $C_0 = 6.50$ and $C_1 = 15$.

Now let's calculate the Silhouette Coefficient Index. Choosing $x_i = x_0 = 2$ for our point

$$a_1 = \frac{1}{3} \sum ((5-2), (9-2), (10-2)) = 5.67 \tag{34}$$

...we need to test x_0, x_1, $x2$, and x_3 to find the minimum average distance to utilize for our calculation.

$$a_2 = \frac{1}{n} \sum d(x_0, C_1);\ = \frac{1}{1}(13) = 13 \tag{35}$$

$$a_2 = \frac{1}{n} \sum d(x_1, C_1);\ = \frac{1}{1}(10) = 10 \tag{36}$$

$$a_2 = \frac{1}{n} \sum d(x_2, C_1);\ = \frac{1}{1}(6) = 6 \tag{37}$$

$$a_2 = \frac{1}{n} \sum d(x_3, C_1);\ = \frac{1}{1}(5) = 5 \tag{38}$$

...which, since we have only the one other cluster, we can stop here. We'll use the minimum value of 5 average distance. If we had more clusters, we would take the minimum of the average distances from each point in other clusters, but we can stop here. We can now calculate the index S_1:

$$S_1 = 1 - \frac{a_1}{a_2} = 1 - \frac{5.67}{5} = -0.34 \tag{39}$$

...and then we do the same for the next cluster C_2. We use $x_i = x_4 = 15$.

$$a_1 = \frac{1}{1}\sum d(x_4, C_1); \; = \frac{1}{1}(0) = 0 \tag{40}$$

Since we have a 0, we can just stop here:

$$S_2 = 1 - \frac{a_1}{a_2} = 1 - \frac{0}{a_2} = 1 - 0 = 1 \tag{41}$$

...so the Silhouette Coefficient Index for our clusters C_0 and C_1 are $S_1 = -0.34$ and $S_2 = 1$, respectively.

We are now going to calculate the Davies-Bouldin Index for this cluster outcome.

$$R_{i,j} = \frac{s_1 + s_2}{d_{ij}} \tag{42}$$

...then we find the max $R_{i,j}$ (which will be the only one we calculate as we only have two clusters), and then take the average of those maxes - will again be just R_{ij} given our setup.

$$D_i = \max(R_{i,j}) \tag{43}$$

$$DB_{index} = \frac{1}{N}\sum_{i=1}^{N} D_i \tag{44}$$

...and now with our values:

$$s_1 = \frac{1}{4}\sum(4.5, 1.5, 2.5, 3.5) = 3.0 \tag{45}$$

$$s_2 = \frac{1}{3}\sum(0) = 0 \tag{46}$$

$$R_{0,1} = \frac{3.0 + 0}{8.5} = 0.35 \tag{47}$$

Finally we move onto our last Calinski-Harabasz Index. Again, we aim to calculate:

$$CH = \left[\frac{\sum_{k=1}^{K} n_k \|c_k - C\|^2}{K-1}\right] / \left[\frac{\sum_{k=1}^{K}\sum_{i=1}^{n_k} \|d_i - c_k\|^2}{N-K}\right] \tag{48}$$

$$C = \frac{1}{N}\sum(d_i) = \frac{1}{5}\sum(2, 5, 9, 10, 15) = 8.2 \tag{49}$$

$$CH = \frac{\left[\frac{\sum(4\|6.5-8.2\|^2, 1\|15-8.2\|^2)}{1}\right]}{\left[\frac{\sum(\sum(\|2-6.5\|^2, \|5-6.5\|^2, \|9-6.5\|^2, \|10-6.5\|^2), \sum(\|15-15\|^2))}{5-2}\right]} \tag{50}$$

$$CH = \frac{\left[\frac{\sum(4\|-1.7\|^2, 1\|6.2\|^2)}{1}\right]}{\left[\frac{\sum(\sum(\|-4.5\|^2, \|-1.5\|^2, \|2.5\|^2, \|3.5\|^2), \sum(\|0\|^2))}{3}\right]} \tag{51}$$

$$CH = \frac{\left[\frac{\sum(4*2.89, 1*38.44)}{1}\right]}{\left[\frac{\sum(\sum(20.25, 2.25, 6.25, 12.25), \sum(0))}{3}\right]} \tag{52}$$

$$CH = \frac{\left[\frac{\sum(11.56, 38.44)}{1}\right]}{\left[\frac{\sum(\sum(41.0), \sum(0))}{3}\right]} \tag{53}$$

$$CH = \frac{\left[\frac{50}{1}\right]}{\left[\frac{41}{3}\right]} \tag{54}$$

$$CH = 3.67 \tag{55}$$

Part C

In this quesiton, we are asked which initial clusters yielded better results. To determine this, we compare the indexes we calculated for each setup, which we'll label A and B.

Index	A	B
SI	0.76, 0.37	-0.34, 1
DHI	0.50	0.35
CHI	8.68	3.67

For the Silhouette Index, the higher the SCI, the more clustering improves. The SI for A is higher than one of the values of B but not both, and the negative index value is a poor sign. The lower DHI is better, so B wins out slightly here. A wins with the CHI index, since higher is better.

Since A wins two of three indexes, we would say that A is the better clustering attempt.

Problem 2

In this problem we are performing Association Analysis (AA) on the following supermarket customer transactions:

Transaction (C)	Milk (M)	Eggs (E)	Bread (B)
Customer 1	0	1	1
Customer 2	1	0	1
Customer 3	1	1	0
Customer 4	0	1	1
Customer 5	1	1	1
Customer 6	0	1	1
Customer 7	1	1	1

Part A

First we will generate all the possible itemsets:

Itemset	Size	Frequency	Support
M	1	0	$\frac{4}{7}$
E	1	0	$\frac{6}{7}$
B	1	0	$\frac{6}{7}$
ME	2	1	$\frac{3}{7}$
MB	2	1	$\frac{3}{7}$
EB	2	3	$\frac{5}{7}$
MEB	3	2	$\frac{2}{7}$

Part B

Assuming that our support threshold is 30%, here we will generate all possible associations; note that we're only including itemsets of size 2 and higher.

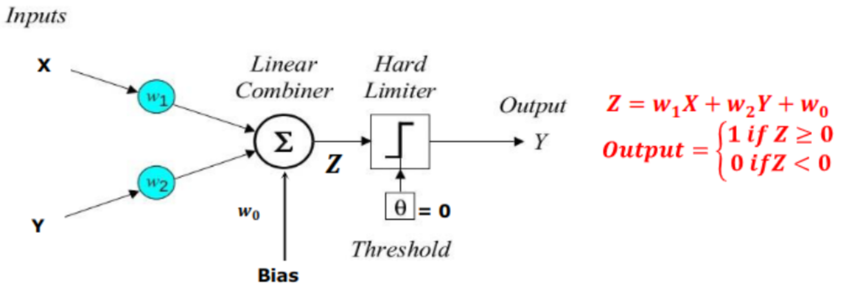
Antecedent	Consequent	Confidence
M	E	$\frac{support(M \cup E)}{support(M)} = \frac{3/7}{4/7} = 0.75$
M	B	$\frac{support(M \cup B)}{support(M)} = \frac{3/7}{4/7} = 0.75$
E	B	$\frac{support(E \cup B)}{support(E)} = \frac{5/7}{6/7} = 0.83$

Part C

If our confidence threshold is above 70%, then 1) Milk, therefore Eggs, 2) Milk, therefore Bread, and 3) Eggs, therefore Bread.

Problem 3

In this problem, we will be training a perceptron to properly classify the following instances:



X	Y	Output
0	0	0
0	1	1
1	0	1
1	1	1

We are making the assumption that the initial weights and bias are $w_1 = 0.75$, $w_2 = 0.45$, and $w_0 = -0.75$, for an initial equation of $z = 0.75x + 0.45y - 0.75$. Our learning rate is 0.1.

Our first step is to calculate the output for our given function with our weights.

$$z = 0.75x + 0.45y - 0.75 \tag{56}$$

X	Y	Output	Expected	Correct?
0	0	-0.75	0	Yes
0	1	-0.30	1	No
1	0	0	1	No
1	1	0.45	1	Yes

Now we look at our incorrect cases. The error for all incorrect answers must be 1 for our purposes because of our use of the output.

$$E = 1 \tag{57}$$

$$w_{i_{new}} = w_i + \alpha E * input \tag{58}$$

$$w_{1_{new}} = w_1 + \alpha E * 0 = 0.75 + 0.1 * 1 * 0 = 0.75 \quad (59)$$

$$w_{2_{new}} = w_2 + \alpha E * 1 = 0.45 + 0.1 * 1 * 1 = 0.55 \quad (60)$$

...and for our bias:

$$w_{bias_{new}} = w_{bias} + \alpha E \quad (61)$$

$$w_{0_{new}} = w_0 + \alpha E * 1 = -0.75 + 0.1 * 1 * 1 = -0.65 \quad (62)$$

Now we take our new weights and run it through our available data again.

$$z = 0.75x + 0.55y - 0.65 \quad (63)$$

X	Y	Output	Expected	Correct?
0	0	-0.65	0	Yes
0	1	-0.10	0	No
1	0	0.10	1	Yes
1	1	0.65	1	Yes

Here we see that we are still not quite right - let's go one more time.

$$w_{i_{new}} = w_i + \alpha E * input \quad (64)$$

$$w_{1_{new}} = w_1 + \alpha E * 0 = 0.75 + 0.1 * 1 * 0 = 0.75 \quad (65)$$

$$w_{2_{new}} = w_2 + \alpha E * 1 = 0.55 + 0.1 * 1 * 1 = 0.65 \quad (66)$$

...and for our bias:

$$w_{bias_{new}} = w_{bias} + \alpha E \quad (67)$$

$$w_{0_{new}} = w_0 + \alpha E * 1 = -0.65 + 0.1 * 1 * 1 = -0.55 \quad (68)$$

...and reapplying it to our testset:

$$z = 0.75x + 0.65y - 0.55 \quad (69)$$

X	Y	Output	Expected	Correct?
0	0	-0.55	0	Yes
0	1	0.10	1	Yes
1	0	0.20	1	Yes
1	1	0.85	1	Yes

...and with that, we've fully trained our perceptron.

Problem 4

Part A

Here we are considering a 1dimensional time-series with values $2, 1, 3, 4, 7$. We are going to perform a convolution with a $1, 0, 1$ filter and zero padding.
Our first batch, due to zero padding, is $0, 2, 1$. So we have $0 * 1 + 0 * 2 + 0 * 1 = 0$
Convolving over one we then look at batch $2, 1, 3$, for $1 * 2 + 0 * 1 + 1 * 3 = 5$.
Over to $1, 3, 4$ we get $1 * 1 + 0 * 3 + 1 * 4 = 5$.
Moving further to $3, 4, 7$ we have $1 * 3 + 0 * 4 + 1 * 7 = 10$
...and finally, due to zero padding, $4, 7, 0$, we get $1 * 4 + 0 * 7 + 1 * 0 = 4$.
So our final result is $0, 5, 5, 10, 4$

Part B

Here we are utilizing the following tensor and filter, and are performing a stride of 1 convolution with no padding.

$$\begin{array}{ccccccc}
6 & 3 & 4 & 5 & 5 & 0 & 3 \\
4 & 7 & 0 & 4 & 4 & 0 & 4 \\
7 & 0 & 3 & 4 & 4 & 5 & 2 \\
3 & 7 & 0 & 3 & 3 & 0 & 7 \\
5 & 8 & 2 & 5 & 5 & 4 & 2 \\
8 & 0 & 0 & 6 & 6 & 0 & 0 \\
6 & 4 & 3 & 0 & 0 & 4 & 5
\end{array}$$

$$\begin{array}{ccc}
1 & 1 & 1 \\
0 & 0 & 0 \\
-1 & -1 & -1
\end{array}$$

: What follows is the resulting convolution with this horizontal edge detection filter.

$$1 * 6 + 1 * 3 + 1 * 4 + 0 * 4 + 0 * 7 + 0 * 4 + -1 * 7 + -1 * 0 + -1 * 2 = 1 \tag{70}$$

$$1 * 3 + 1 * 4 + 1 * 4 + 0 * 7 + 0 * 4 + 0 * 0 + -1 * 0 + -1 * 2 + -1 * 3 = 6 \tag{71}$$

$$1 * 4 + 1 * 4 + 1 * 5 + 0 * 7 + 0 * 4 + 0 * 4 + -1 * 2 + -1 * 3 + -1 * 4 = 4 \tag{72}$$

$$1 * 4 + 1 * 5 + 1 * 0 + 0 * 0 + 0 * 4 + 0 * 0 + -1 * 3 + -1 * 4 + -1 * 5 = -3 \tag{73}$$

$$1 * 5 + 1 * 0 + 1 * 3 + 0 * 4 + 0 * 0 + 0 * 4 + -1 * 4 + -1 * 5 + -1 * 2 = -1 \tag{74}$$

...then we move down a row...

$$1 * 4 + 1 * 7 + 1 * 4 + 0 * 7 + 0 * 0 + 0 * 2 + -1 * 3 + -1 * 7 + -1 * 5 = 0 \tag{75}$$

$$1 * 7 + 1 * 4 + 1 * 0 + 0 * 0 + 0 * 2 + 0 * 3 + -1 * 7 + -1 * 5 + -1 * 0 = -1 \tag{76}$$

$$1 * 4 + 1 * 0 + 1 * 4 + 0 * 2 + 0 * 3 + 0 * 4 + -1 * 5 + -1 * 0 + -1 * 3 = 0 \tag{77}$$

$$1 * 0 + 1 * 4 + 1 * 0 + 0 * 3 + 0 * 4 + 0 * 5 + -1 * 0 + -1 * 3 + -1 * 0 = 1 \tag{78}$$

$$1 * 4 + 1 * 0 + 1 * 4 + 0 * 4 + 0 * 5 + 0 * 2 + -1 * 3 + -1 * 0 + -1 * 7 = -2 \tag{79}$$

...and onto the next row:

$$1 * 7 + 1 * 0 + 1 * 2 + 0 * 3 + 0 * 7 + 0 * 5 + -1 * 5 + -1 * 8 + -1 * 1 = -5 \tag{80}$$

$$1 * 0 + 1 * 2 + 1 * 3 + 0 * 7 + 0 * 5 + 0 * 0 + -1 * 8 + -1 * 1 + -1 * 2 = -6 \tag{81}$$

$$1 * 2 + 1 * 3 + 1 * 4 + 0 * 5 + 0 * 0 + 0 * 3 + -1 * 1 + -1 * 2 + -1 * 5 = 1 \tag{82}$$

$$1 * 3 + 1 * 4 + 1 * 5 + 0 * 0 + 0 * 3 + 0 * 0 + -1 * 2 + -1 * 5 + -1 * 4 = 1 \tag{83}$$

$$1 * 4 + 1 * 5 + 1 * 2 + 0 * 3 + 0 * 0 + 0 * 7 + -1 * 5 + -1 * 4 + -1 * 2 = 0 \tag{84}$$

...and again:

$$1*3+1*7+1*5+0*5+0*8+0*1+-1*8+-1*0+-1*1=6 \tag{85}$$

$$1*7+1*5+1*0+0*8+0*1+0*2+-1*0+-1*1+-1*0=11 \tag{86}$$

$$1*5+1*0+1*3+0*1+0*2+0*5+-1*1+-1*0+-1*6=1 \tag{87}$$

$$1*0+1*3+1*0+0*2+0*5+0*4+-1*0+-1*6+-1*0=-3 \tag{88}$$

$$1*3+1*0+1*7+0*5+0*4+0*2+-1*6+-1*0+-1*0=4 \tag{89}$$

Exhaustingly, one more row left:

$$1*5+1*8+1*1+0*8+0*0+0*1+-1*6+-1*4+-1*1=1 \tag{90}$$

$$1*8+1*1+1*2+0*0+0*1+0*0+-1*4+-1*1+-1*3=3 \tag{91}$$

$$1*1+1*2+1*5+0*1+0*0+0*6+-1*1+-1*3+-1*0=4 \tag{92}$$

$$1*2+1*5+1*4+0*0+0*6+0*0+-1*3+-1*0+-1*4=4 \tag{93}$$

$$1*5+1*4+1*2+0*6+0*0+0*0+-1*0+-1*4+-1*5=2 \tag{94}$$

Finally, our results are:

1	6	4	-3	-1
0	-1	0	1	-2
-5	-6	1	1	0
6	11	1	-3	4
1	3	4	4	2

Part C

Now we seek to perform a $4x4$ pooling with stride 1 from the above input. Since no pooling type is specified, we shall perform a max pooling.
Our first passover:

$$\max(6,3,4,4,4,7,4,0,7,0,2,3,3,7,5,0)=7 \tag{95}$$

$$\max(3,4,4,5,7,4,0,4,0,2,3,4,7,5,0,3)=7 \tag{96}$$

$$\max(4,4,5,0,4,0,4,0,2,3,4,5,5,0,3,0)=5 \tag{97}$$

$$\max(4,5,0,3,0,4,0,4,3,4,5,2,0,3,0,7)=7 \tag{98}$$

...next row:

$$\max(4,7,4,0,7,0,2,3,3,7,5,0,5,8,1,2)=7 \tag{99}$$

$$\max(7,4,0,4,0,2,3,4,7,5,0,3,8,1,2,5)=7 \tag{100}$$

$$\max(4,0,4,0,2,3,4,5,5,0,3,0,1,2,5,4)=5 \tag{101}$$

$$\max(0,4,0,4,3,4,5,2,0,3,0,7,2,5,4,2) = 5 \tag{102}$$

...and the next

$$\max(7,0,2,3,3,7,5,0,5,8,1,2,8,0,1,0) = 8 \tag{103}$$

$$\max(0,2,3,4,7,5,0,3,8,1,2,5,0,1,0,6) = 8 \tag{104}$$

$$\max(2,3,4,5,5,0,3,0,1,2,5,4,1,0,6,0) = 6 \tag{105}$$

$$\max(3,4,5,2,0,3,0,7,2,5,4,2,0,6,0,0) = 7 \tag{106}$$

...and finally the last row:

$$\max(3,7,5,0,5,8,1,2,8,0,1,0,6,4,1,3) = 8 \tag{107}$$

$$\max(7,5,0,3,8,1,2,5,0,1,0,6,4,1,3,0) = 8 \tag{108}$$

$$\max(5,0,3,0,1,2,5,4,1,0,6,0,1,3,0,4) = 6 \tag{109}$$

$$\max(0,3,0,7,2,5,4,2,0,6,0,0,3,0,4,5) = 7 \tag{110}$$

...for our resulting pooled output:

$$\begin{matrix} 7 & 7 & 5 & 7 \\ 7 & 7 & 5 & 5 \\ 8 & 8 & 6 & 7 \\ 8 & 8 & 6 & 7 \end{matrix}$$

Problem 5

In this problem, we are looking to demonstrate transfer learning using AlexNet on a provided dataset of plants. For our purposes (and due to limitations of training hardware) we have limited ourselves to only 8 categories from this dataset.

We seek to demonstrate the difference of accuracy of a model trained on different size datasets - specifically, we will be limiting each cateogry to 100, 150, 200, 250, and 300 images.

During training, we tracked not just the running average loss per epoch, but also tracked a ”full loss” at the end of each epoch. This is calculated at the end of each epoch, disabling gradient training on the model, and then running the model through the entire dataset to calculate average loss, instead of utilizing the limited dataset.

We trained through 25 epochs with a learning rate of 0.01. These are our results:

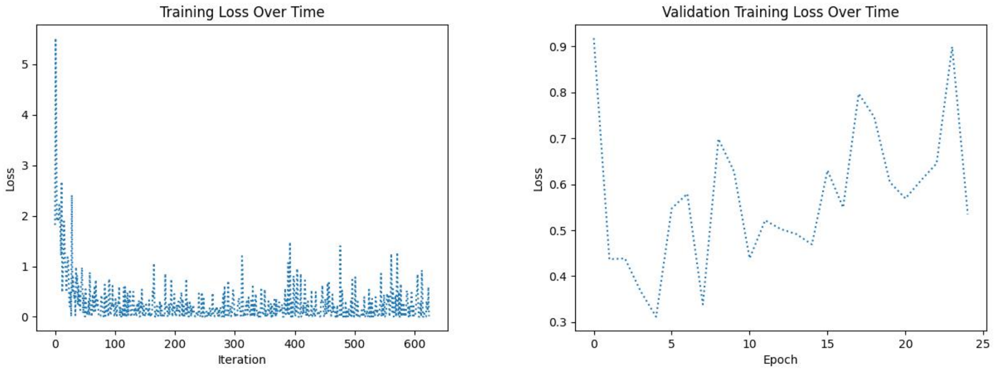


Figure 1: Training and Validation Loss w/ Dataset of 100 Images Per Category

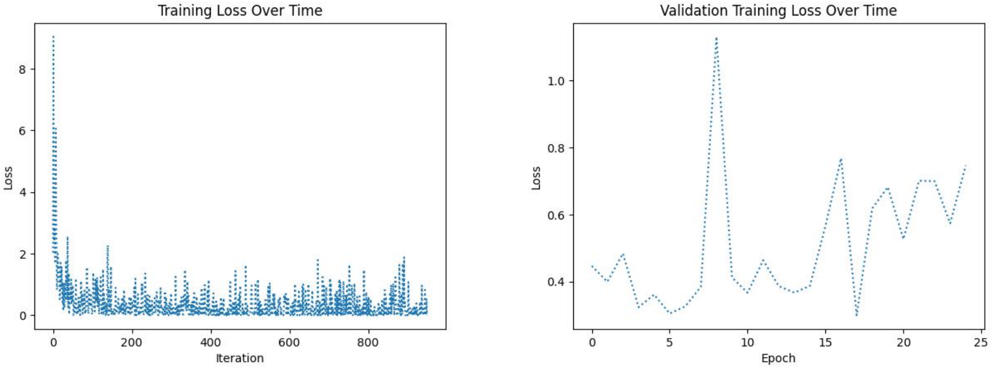


Figure 2: Training and Validation Loss w/ Dataset of 150 Images Per Category

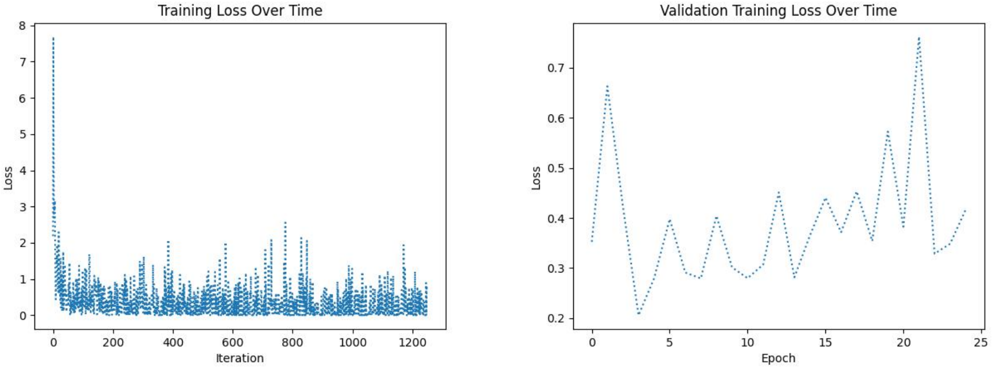


Figure 3: Training and Validation Loss w/ Dataset of 200 Images Per Category

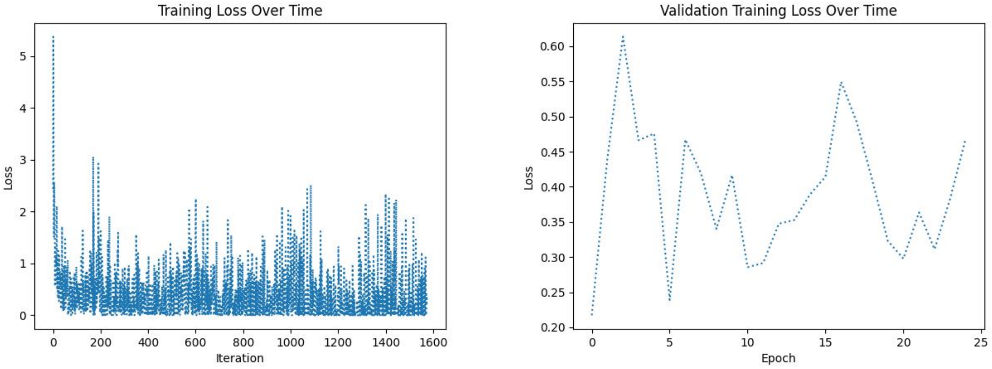


Figure 4: Training and Validation Loss w/ Dataset of 250 Images Per Category

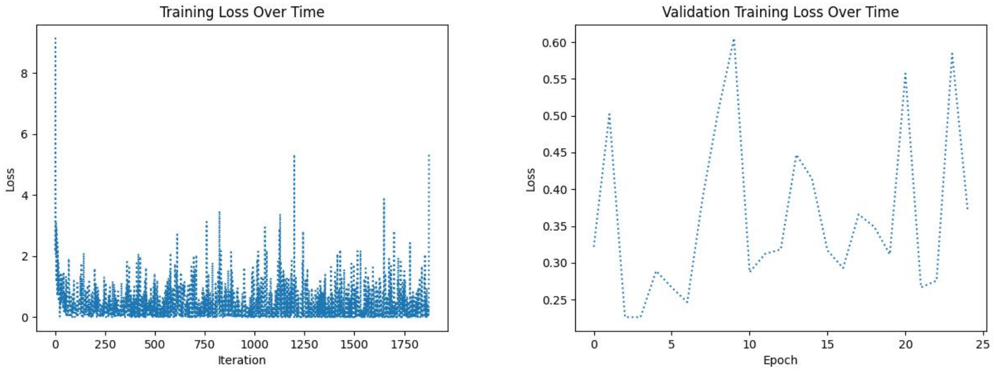


Figure 5: Training and Validation Loss w/ Dataset of 300 Images Per Category

noindent Finally, we present the accuracy of our final models, when compared against the full dataset:

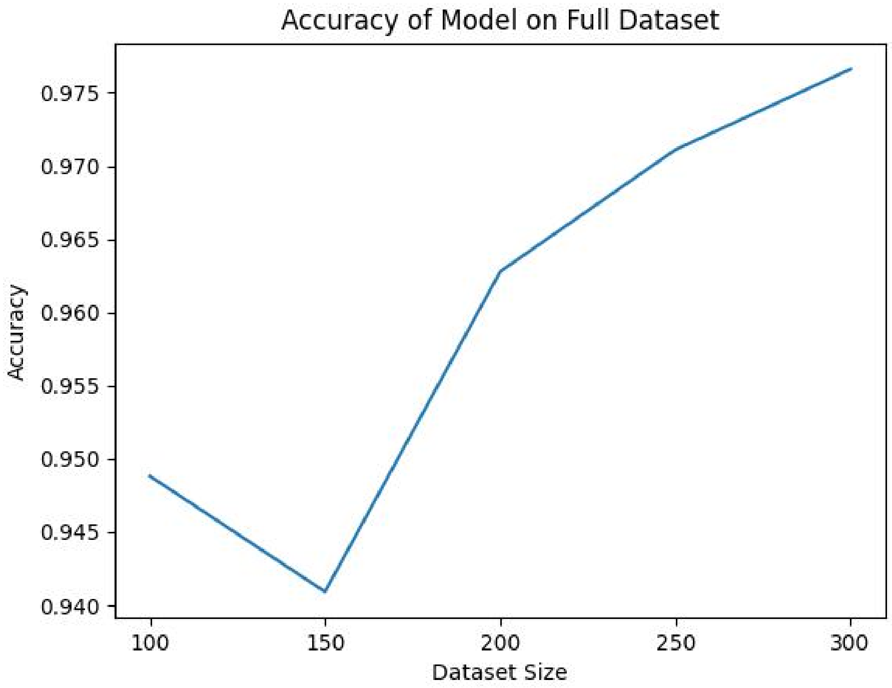


Figure 6: Model Accuracy

Here we see a clear increase in accuracy of the transfer-learned model as we increase the amonut of data available to the network to train upon. We ranged from 95% to 98% with 100 to 300 images per category, respectively.