Introduction to Machine Learning

Exercise 3

1c

```
k = 10
Cluster 0 has 69 samples, and the most frequent label is 4 with frequency 31
Cluster 1 has 110 samples, and the most frequent label is 9 with frequency 44
Cluster 2 has 152 samples, and the most frequent label is 3 with frequency 71
Cluster 3 has 79 samples, and the most frequent label is 0 with frequency 75
Cluster 4 has 92 samples, and the most frequent label is 2 with frequency 69
Cluster 5 has 138 samples, and the most frequent label is 8 with frequency 42
Cluster 6 has 94 samples, and the most frequent label is 7 with frequency 49
Cluster 7 has 147 samples, and the most frequent label is 1 with frequency 98
Cluster 8 has 29 samples, and the most frequent label is 8 with frequency 14
Cluster 9 has 90 samples, and the most frequent label is 6 with frequency 69
The classification error is 0.438
```

If the most frequent label in cluster i is not the same as the label of the sample in cluster i, then it is a wrong classification.

The classification error is calculated as the sum of all the wrong classifications divided by the sample size.

1d

```
k = 10
Cluster 0 has 287 samples, and the most frequent label is 1 with frequency 30
Cluster 1 has 1 samples, and the most frequent label is 0 with frequency 1
Cluster 2 has 1 samples, and the most frequent label is 2 with frequency 1
Cluster 3 has 1 samples, and the most frequent label is 2 with frequency 1
Cluster 4 has 1 samples, and the most frequent label is 2 with frequency 1
Cluster 5 has 1 samples, and the most frequent label is 2 with frequency 1
Cluster 6 has 4 samples, and the most frequent label is 4 with frequency 4
Cluster 7 has 2 samples, and the most frequent label is 6 with frequency 2
Cluster 8 has 1 samples, and the most frequent label is 8 with frequency 1
Cluster 9 has 1 samples, and the most frequent label is 9 with frequency 1
The classification error is 0.8566666666666667
```

As we can see clearly from the classification error the k-means algorithm worked better for this problem.

1e

k-means:

```
k = 6
Cluster 0 has 184 samples, and the most frequent label is 4 with frequency 55
Cluster 1 has 218 samples, and the most frequent label is 1 with frequency 100
Cluster 2 has 214 samples, and the most frequent label is 3 with frequency 82
Cluster 3 has 86 samples, and the most frequent label is 0 with frequency 66
Cluster 4 has 48 samples, and the most frequent label is 0 with frequency 29
Cluster 5 has 250 samples, and the most frequent label is 7 with frequency 88
The classification error is 0.58
```

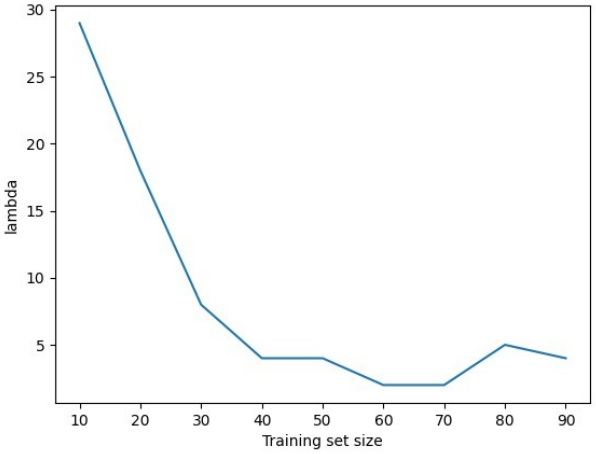
single-linkage:

```
k = 6
Cluster 0 has 1 samples, and the most frequent label is 0 with frequency 1
Cluster 1 has 1 samples, and the most frequent label is 6 with frequency 1
Cluster 2 has 1 samples, and the most frequent label is 6 with frequency 1
Cluster 3 has 1 samples, and the most frequent label is 8 with frequency 1
Cluster 4 has 295 samples, and the most frequent label is 1 with frequency 30
Cluster 5 has 1 samples, and the most frequent label is 8 with frequency 1
The classification error is 0.8833333333333333
```

When moving from k = 10 to k = 6 we expect the classification error to increase, and we can see that that is what happened especially in the k-means algorithm.

The k-means algorithm finds a centroid for each of the 6 clusters and the data points will be reassigned to the closest cluster. This means that some points that were previously in different clusters will now be grouped together in the same cluster, which may affect the most common label in the cluster, Thus increasing the number of misclassified labels.

2a



2b

When the training set size is small, the model may be more prone to overfitting, because the model may have learned specific patterns in the smaller training data that do not generalize to testing data.
So, to prevent the larger errors that will be caused from this overfitting, the optimal lambda may be larger in those smaller training sets, so that we get stronger regularization on the model which will prevent the model from overfitting and improve its generalization performance.
As the training set size increases, the model has access to more information about the underlying relationships in the data and may be able to learn more generalizable patterns. This can help the model to make more accurate predictions on unseen data, even without a strong regularization, so of course we expect the optimal lambda value to get smaller.

2c

Yes. This is generally the trend of the plot that we submitted. However, we can see that as we reached the biggest training set sizes the optimal lambda value actually increased a bit.
This can occur because the data we used was very complex or had a lot of noise , so we needed larger regularization to get smaller errors. This can also occur because the model we trained was too complex (e.g. had a large number of parameters) so was prone to overfitting, thus a stronger regularization(larger lambda) was needed to constrain the model and improve its generalization performance.

2d

To find the Bayes-optimal predictor, we need to minimize the expected loss over all possible predictions given the true distribution of the data.
The squared loss function is defined as $L(y,f(x)) = (y - f(x))^2$, where y is the true label and f(x) is the predicted label.
η is a standard Gaussian random variable with mean 0 and variance σ^2.
Using the definition of the squared loss function, we can calculate the expected loss as follows:
$E[L(y,f(x))] = E[(y - f(x))^2]= E[y^2 - 2y*f(x) + f(x)^2] = E[y^2] - 2E[y]E[f(x)] + E[f(x)^2]$
We'll calculate each of these expected values separately.
$E[y^2] = E[(<w, x> + \eta)^2] = E[<w, x>^2 + 2*<w, x>\eta + \eta^2] = = <w, x>^2 + 2<w, x>*0 + \sigma^2 = <w, x>^2 + \sigma^2$
And for $E[f(x)^2] = E[f(x)^2|x] = E[(<w, x> + \eta)^2|x] = E[<w, x>^2|x] + E[2*<w, x>\eta|x] + E[\eta^2|x] = <w, x>^2 + 2<w, x>*E[\eta|x] + \sigma^2$
Since η is independent of x, we can say that $E[\eta|x] = E[\eta] = 0$. So $E[f(x)^2]= <w, x>^2 +\sigma^2$

E[y] = <w, x> , E[f(x)] = <w, x> + E[η] = <w, x> so E[y]E[f(x)] = <w, x>^2
So E[y^2] - 2E[y]E[f(x)] + E[f(x)^2] = <w, x>^2 +σ^2 -2* <w, x>^2 + <w, x>^2 +σ^2 = 2*(σ^2)
So the expected loss does not depend on f(x) and is a constant value regardless of f.
So the best way to minimize the expected loss is to minimize the square difference between the true label and the predicted label, which is <w, x>.

The Bayes-optimal predictor for this problem with respect to the absolute loss is the median of the conditional distribution of y given x.
The absolute loss is defined as L(y,f(x)) = |y-f(x)|, which is the absolute difference between the true label y and the predicted label f(x).
For a given input x, the true label y is distributed according to a normal distribution with mean <w,x> and standard deviation σ, N(<w,x>,σ^2).
The median of a normal distribution is equal to its mean, so the median of the conditional distribution of y given x is <w,x>
Therefore, the Bayes-optimal predictor for this problem with respect to the absolute loss is <w,x>. This is because, the median of the conditional distribution of y given x is the point that minimize the expected absolute loss among all possible predictions, and <w,x> is the median of the conditional distribution of y given x.

3a

As we learned in class: $w_{t+1} \leftarrow w_t - \eta \nabla f(w_t)$.

We'll define the object function: $f(w,S) = \lambda ||w|| + \sum_{i=1}^{m} (< w, x_i > - y_i)^2$

And: $g(w) = \lambda ||w||$ so its gradient will be: $\nabla g(w) = \lambda(\frac{w_1}{||w||}, \dots, \frac{w_d}{||w||})$.

And: $h_i(w) = (< w, x_i > - y_i)^2$ so its gradient will be:

$\nabla h_i(w) = \nabla(< w, x_i>^2 - 2 < w, x_i > y_i + y_i^2) = 2 < w, x_i > - 2y_i x_i$.

So:

$w_{t+1} \leftarrow w_t - \eta(\lambda(\frac{w_1}{||w||}, \dots, \frac{w_d}{||w||}) + \sum_{i=1}^{m} 2 < w, x_i > - 2y_i x_i)$

3b

As we learned in class: $w_{t+1} \leftarrow w_t - \eta(\nabla g(w_t) + \nabla \sum_{i=1}^{m} h_i(w))$.

We'll define the object function: $f(w,S) = \lambda ||w|| + \sum_{i=1}^{m} (< w, x_i > - y_i)^2$

And: $g(w) = \lambda ||w||$ so its gradient will be: $\nabla g(w) = \lambda(\frac{w_1}{||w||}, \dots, \frac{w_d}{||w||})$.

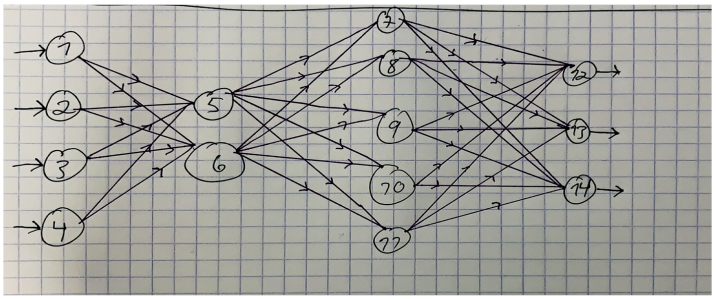
And: $h_i(w) = (< w, x_i > - y_i)^2$ so its gradient will be:

$\nabla h_i(w) = \nabla(< w, x_i>^2 - 2 < w, x_i > y_i + y_i^2) = 2 < w, x_i > - 2y_i x_i$.

So:

$w_{t+1} \leftarrow w_t - \eta(\lambda(\frac{w_1}{||w||}, \dots, \frac{w_d}{||w||}) + \sum_{i=1}^{m} 2 < w, x_i > - 2y_i x_i)$

4a



4b

$X \in R^4$

4c

Y is a group of size 3: $Y = \{1, 2, 3\}$

4d

$H = \{h_w(x) \mid w \in R^{33}\}$

$h_w((x_1, x_2, x_3, x_4)) = \Psi((o_{12}, o_{13}, o_{14})) = argmax\{o_{12}, o_{13}, o_{14}\} =$

$= argmax\{$

$$\sum_{i:(i,12)\in E} w_{i,12} \cdot o_i \quad , \quad \sum_{i:(i,13)\in E} w_{i,13} \cdot o_i \quad , \quad \sum_{i:(i,14)\in E} w_{i,14} \cdot o_i \quad \} =$$

$= argmax\{$

$$\sum_{i:(i,12)\in E} w_{i,12} \cdot sigmoid\Big(\sum_{j:(j,i)\in E} w_{j,i} \cdot o_j\Big) \quad , \quad \sum_{i:(i,13)\in E} w_{i,13} \cdot sigmoid\Big(\sum_{j:(j,i)\in E} w_{j,i} \cdot o_j\Big) \quad ,$$

$$\sum_{i:(i,14)\in E} w_{i,14} \cdot sigmoid\Big(\sum_{j:(j,i)\in E} w_{j,i} \cdot o_j\Big) \quad \} =$$

$= argmax\{$

$$\sum_{i:(i,12)\in E} w_{i,12} \cdot sigmoid\Big(\sum_{j:(j,i)\in E} w_{j,i} \cdot sigmoid\Big(\sum_{t:(t,j)\in E} w_{t,j} \cdot o_t\Big)\Big) \quad ,$$

$$\sum_{i:(i,13)\in E} w_{i,13} \cdot sigmoid\Big(\sum_{j:(j,i)\in E} w_{j,i} \cdot sigmoid\Big(\sum_{t:(t,j)\in E} w_{t,j} \cdot o_t\Big)\Big) \quad ,$$

$$\sum_{i:(i,14)\in E} w_{i,14} \cdot sigmoid\Big(\sum_{j:(j,i)\in E} w_{j,i} \cdot sigmoid\Big(\sum_{t:(t,j)\in E} w_{t,j} \cdot o_t\Big)\Big) \quad \}$$

$= argmax\{$

$$\sum_{i:(i,12)\in E} w_{i,12} \cdot sigmoid\Big(\sum_{j:(j,i)\in E} w_{j,i} \cdot sigmoid\Big(\sum_{t:(t,j)\in E} w_{t,j} \cdot x_t\Big)\Big) \quad ,$$

$$\sum_{i:(i,13)\in E} w_{i,13} \cdot sigmoid\Big(\sum_{j:(j,i)\in E} w_{j,i} \cdot sigmoid\Big(\sum_{t:(t,j)\in E} w_{t,j} \cdot x_t\Big)\Big) \quad ,$$

$$\sum_{i:(i,14)\in E} w_{i,14} \cdot sigmoid\Big(\sum_{j:(j,i)\in E} w_{j,i} \cdot sigmoid\Big(\sum_{t:(t,j)\in E} w_{t,j} \cdot x_t\Big)\Big) \quad \}$$

$for\ all\colon\ 1 \le t \le 4\ \ and\ \ x_t\ is\ one\ input\ layer\ neuron$

5a

Each x has d coordinates and for each coordinate we can ask if $x_i \geq \theta$ for $\theta \in \{\frac{1}{4}, \frac{1}{2}, \frac{3}{4}\}$, thus each internal node has $3d$ options.

A leaf has 2 options for the 2 possible labels.

For a tree with a depth of at most n, the maximum number of nodes is 2^{n+1}.

Since that: $|H_n| \le (3d+2)^{2^{n+1}}$

5b

No, Danny is not correct.

ID3 tries to find a tree with a small sample error, though it's not always the case and the ID3 algorithm may not return the optimal decision tree for distribution D, even with a large sample size m.

The error depends on D, which is unknown and the chosen ϵ, δ that may be too strict.

Since that, the different between $err(T_s, D)$ and $inf_{T\in H_k} err(T, D)$ may be too small to be achieved and since ϵ, δ are given, it is not always true with probability of 1-x.

6a

The Bayes assumption does not hold for this distribution.

$P(x = (-1, 1) \mid Y = -1) = 0$

But:

$P(x_1 = -1 \mid Y = -1) \cdot P(x_2 = 1 \mid Y = -1) = (\frac{5}{60} + 0) \cdot (\frac{4}{60} + 0) \neq 0$

Thus we don't have coordinates independency.

6b

The predictor we would get from this algorithm will be:

$(\frac{-1}{-1}) \rightarrow +1$

$(\frac{-1}{1}) \rightarrow +1$

$(\frac{1}{-1}) \rightarrow -1$

$(\frac{1}{1}) \rightarrow +1$

The Naive-Bayes algorithm will return the label that has the hightest probability to appear.

Since the probability in the sample given is the same as the distribution, the predictor will return the label with the higher probability, as above.

7a

In the experiment we have m vectors in R^4. $x_i = (x_i(1), x_i(2), x_i(3), x_i(4))$

As given in the question, for each i : $x_i(3) = 3x_i(1) + x_i(2)$ and $x_i(4) = 2x_i(2) - 4x_i(1)$.

Since there is a dependency between $x_i(3)$,$x_i(4)$ and $x_i(1)$ $x_i(2)$ the degree of the matrix A is maximum 2. And because of that we have 2 eigenvalues $\lambda_1, \lambda_2 = 0$.

The distortion of the PCA will be the sum of the $d - k$ smallest eigenvalues.

$d = 4,\ k = 2,\ d - k = 2.$

Since we have 2 eigenvalues that are 0 the distortion will be 0.

7b

$m = 4$, $x_3 = x_1^2 + x_2^3$, $x_4 = (x_3 - x_1)^2$

$x_1 = (1, 0, 1, 0)$

$x_2 = (0, 1, 1, 1)$

$x_3 = (1, 1, 2, 1)$

$x_4 = (2, 1, 5, 9)$

$A = \sum_{i=1}^{4} x_i \cdot x_i^T = A_1 + A_2 + A_3 + A_4 =$

6	3	13	19
3	3	8	11
13	8	31	48
19	11	48	83

After row reduction the matrix will look like this:

1	0	1	0
0	1	1	1
0	1	0	3
0	0	1	4

And finally:

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

We can see that the degree of the matrix A is bigger than 2.

So the distortion must be bigger than 0.

The eigenvectors for the matrix A are - $\lambda_1 \sim 1.111, \lambda_2 = 0, \lambda_3 \sim 3.842, \lambda_4 = 118.047.$

The distortion is the sum of the 2 smallest eigenvectors $= 1.111$.

8a

As we learned in class:

$$P_{S\sim D_\theta^m}[S' = S] = \prod_{i=1}(\theta_1 I[x_i = 1] + \theta_2 I[x_i = 2] + \theta_3 I[x_i = 3]) =$$

$$= \theta_1^{\frac{1}{2}\Sigma(2-x_i)(3-x_i)}\theta_2^{-1\Sigma(1-x_i)(3-x_i)}\theta_3^{\frac{1}{2}\Sigma(2-x_i)(1-x_i)}$$

The Log-likelihood:

$$L(S;\theta) = log\ P_{S\sim D_\theta^m}[S' = S] =$$

$$= \frac{1}{2}\Sigma(2-x_i)(3-x_i)\ log(\theta_1) - 1\Sigma(1-x_i)(3-x_i)\ log(\theta_2) + \frac{1}{2}\Sigma(2-x_i)(1-x_i)\ log(\theta_3)$$

The value of the maximum likelihood estimator $\hat{\theta}$ is $\hat{\theta} = argmax\ \ L(S;\theta)$

So:

$$\hat{\theta} = argmax\ \ L(S;\theta) = \frac{\partial L(S;\theta)}{\partial\ \theta} = \frac{\sum_i(2-x_i)(3-x_i)}{2\theta_1} - \frac{\sum_i(1-x_i)(3-x_i)}{\theta_2} - \frac{2\sum_i(2-x_i)(1-x_i)}{1-4\theta_3} = 0$$

8b

X is distributed by $D(p_1,\ldots,p_k,\sigma_1,\ldots,\sigma_k)$, we can say that X is distributed by $\sum_{j=1}^{k} p_j * N(1,\sigma_j^2)$.

So we can say that $\Theta = \{(p_1,\ldots,p_k,\sigma_1,\ldots,\sigma_k) | \sum_{j=1}^{k} p_j, \sigma_j > 0\}$

and that Z is distributed by Multinomial(m,1,...,k).

Thus we can conclude $p(Z = (z_1,\ldots,z_m)) = \binom{m}{z_1,\ldots,z_m} * \prod_{i=1}^{m} p_i^{z_1}$

So the augmented log-likelihood $L(S,Z,p_1,\ldots,p_k,\sigma_1,\ldots,\sigma_k)$ is retrieved by:

$$\sum_{j=1}^{m} log(P_{z_j} * \frac{1}{\sigma_j * \sqrt{2\Pi}} * e^{\frac{-(x_j-1)^2}{2\sigma_j}})$$