Concordia University

Computer Science and Software Engineering Department

COMP 6321

Machine Learning

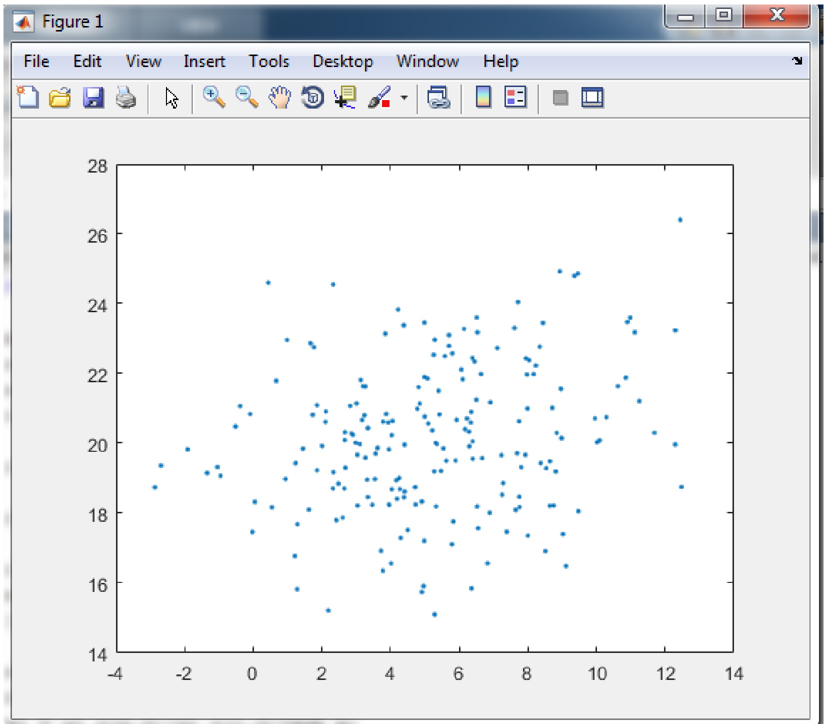
Assignment5

Student ID: 27162391

Question1 (a)

Please check file Question1a.m for the solution.

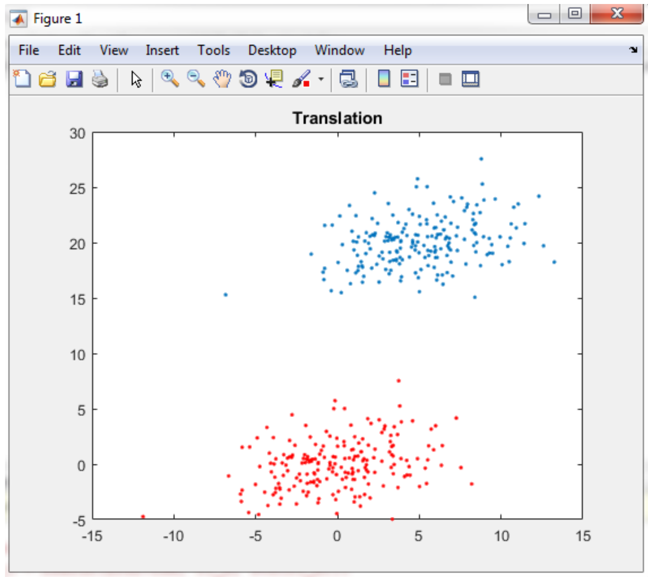
From the data, we can see that the principle component should be along the direction from down-left to upper-right.



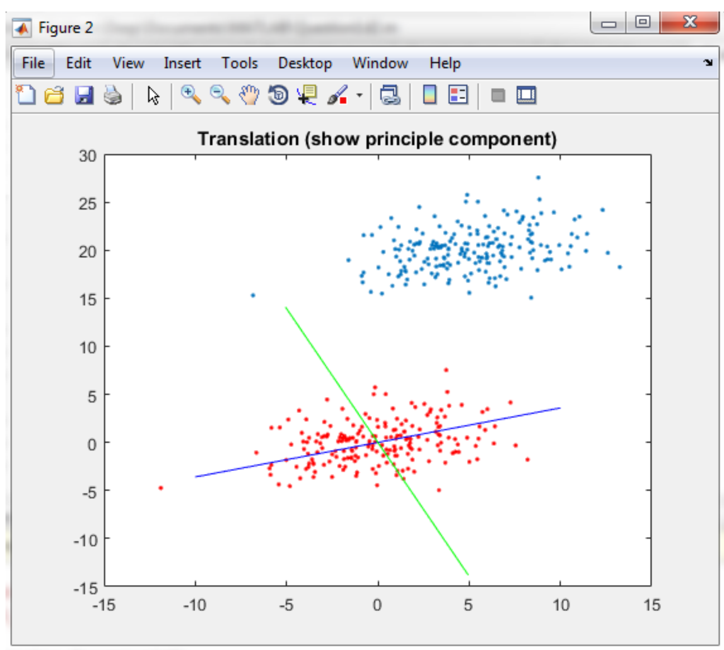
Question1 (b)

Please check file Question1b.m for the solution.

Below is the original generated data (in blue) compared its counterpart (in red) that have been translated to the origin.



I draw the two principle component of the data. That's shown in the blue and green segments. Note that they are orthogonal to each other, but due to different scaling of x and y-axis, it is confusing to believe so. Blue segment is the No.1 principle component, and green is the second principle component. Consider below figure as output.



Compare with the blue data cluster, the principle components have not changed after translation. This can be seen both from the diagram, and from the numerical analysis.

For original data, the two component directions are: $\begin{bmatrix}-0.9525\\-0.3046\end{bmatrix}$ and $\begin{bmatrix}-0.3046\\0.9525\end{bmatrix}$.

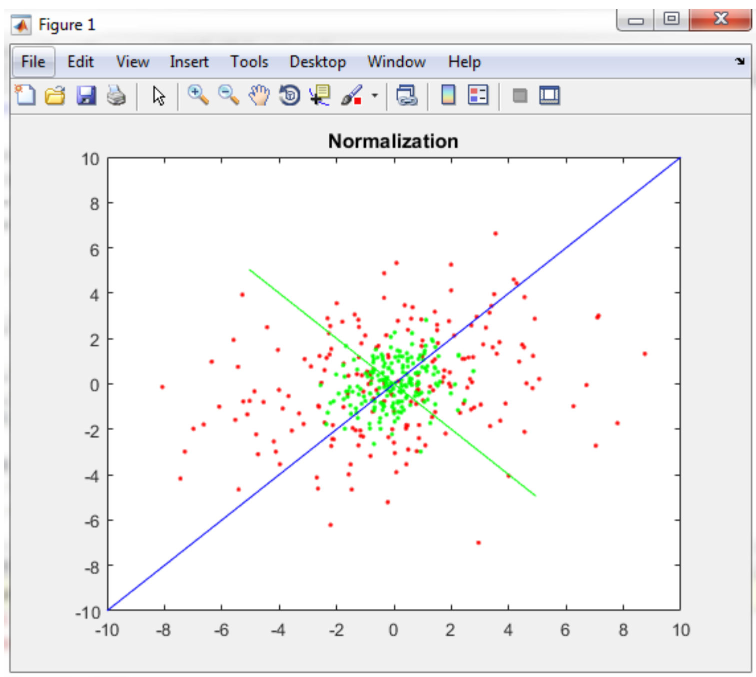
This remains unchanged when data are translated to the origin. Corresponding eigenvalues are 12.1535 and 3.8066. This remains unchanged as well.

Question1(c)

Please check file Question1c.m for the solution.

Consider below figure as output.

The green points are normalized counterpart of red points. Blue segment is the first principle component, and green is the second. From the diagram we can tell that the first two components are orthogonal and, they are along the diagonal direction.



After normalization by dividing the result from Question1 (b) with deviation, the principle components are now changed to two diagonal directions.

This can also be seen from analytical analysis, that after normalization, the two component directions are: $\begin{bmatrix}-0.7071\\-0.7071\end{bmatrix}$ and $\begin{bmatrix}-0.7071\\0.7071\end{bmatrix}$.

Corresponding eigenvalues are 1.3354 and 0.6646. This has changed from Question1 (a) and Question1 (b) as well.

Question1 (d)

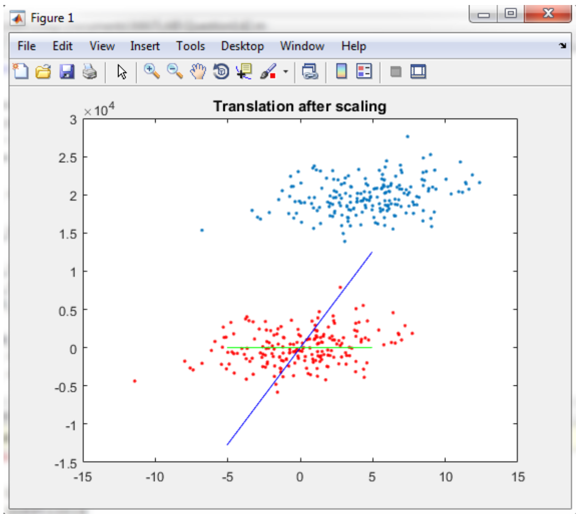
After manipulating the original data by multiplying second coordinate with 1000, below is what happens under the two scenarios:

(1) Translating the origin:

Please check file Question1d1.m for the solution.

Consider below figure as output for Question1 (d) (1).

Data in red are after translation, and in blue are the original data. Blue segment is the first principle component, and green is the second. (This is not clear however, due to the big difference of axis scaling).



Express the principle components in vectors are: $\begin{bmatrix} 5.3e-4 \\ 1 \end{bmatrix}$ and $\begin{bmatrix} 1 \\ -5.3e-4 \end{bmatrix}$.

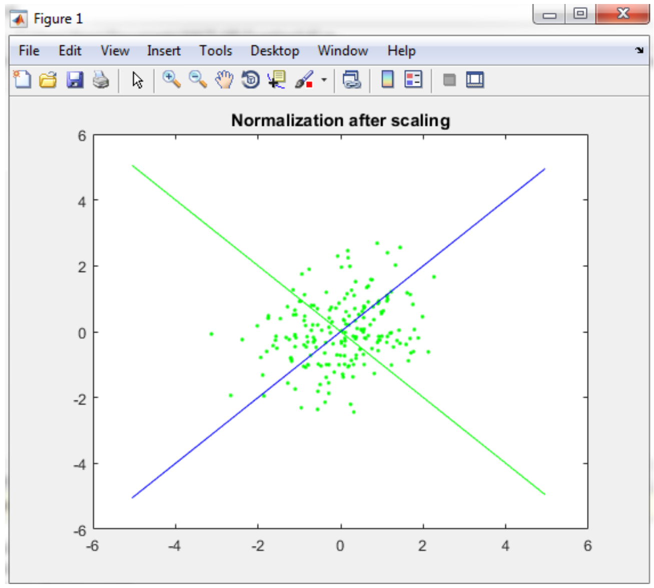
They are orthogonal, and corresponding eigenvalues are: 4.58e+6, and 10.10. This has changed dramatically from original data.

(2) Divide the data with its standard deviation

Please check file Question1d2.m for the solution.

Consider below figure as output for Question 1 (d) (2)

In figure standard deviation is shown as green dots. Blue segment is the first principle component; green segment is the second principle component.



Express in vectors, blue and green segment are $\begin{bmatrix} -0.7071 \\ -0.7071 \end{bmatrix}$ and $\begin{bmatrix} -0.7071 \\ 0.7071 \end{bmatrix}$ respectively, with corresponding eigenvalues 1.3353 and 0.6646. This is the same with the data before scaling the second coordinate.

Conclusion:

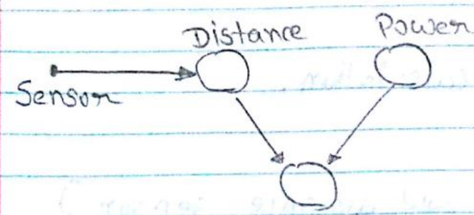
1. Translating data will not change the principle components (and their eigenvalues);

2. Normalization (by dividing standard deviation) will change the principle component into diagonal ones;

3. Scaling part of the data will change principle components and their eigenvalues; however, if this data is then normalized by dividing standard deviation, the principle components will always be diagonal lines. Eigenvalues will not change either, that's they are the same as 2.

Question 2 (a)

=)



Distance/Power	low	High
close	70%	90%
far	10%	50%

Sensor over Distance

Right	80%
Wrong	20%

Question 2 (b)

The probability that he will hit the Borg is 0.58.

Below is the calculation.

P (hit = yes | power = low, distance, sensor)

= P (hit = yes | power = low, distance) *

$\sum_{sensor}$ P (distance | sensor) * P (sensor)

= P (hit = yes | power = low }, distance = close) *

P (sensor = right) + P (hit = yes | power = low, distance = close) * P (sensor = wrong)

= ~~70 % * 80 % + 10 % * 20 %~~

= ~~58 %~~

= 0.

= 70 * 80 + 10 * 20

= 5800

= 58 %

Question 3 (a)

The parameters of this model are :

$P(S_0)$, $P(U_0)$, $P(S_{t-1} \mid S_t, U_t)$, $P(Y_t \mid S_t)$,

$P(U_{t+1} \mid U_t, S_t)$, $P(Z_t \mid U_t)$

Question 3 (b)

The joint probability is

$$P(y_0, z_0, y_1, z_1, \ldots, y_t, z_t)$$

$$= \sum_{\substack{s_0, \ldots, s_t \\ u_0, \ldots, u_t}} P\left(y_0, \ldots, y_t, z_0, \ldots, z_t, s_0, \ldots, s_t, u_0, \ldots, u_t\right)$$

$$= \sum_{\substack{s_0, \ldots, s_t \\ u_0, \ldots, u_t}} P(s_0) P(u_0) \left(\prod_{t=0}^{T} P(y_t \mid s_t) P(z_t \mid u_t) \right) \cdot \left(\prod_{t=1}^{T} P(s_t \mid s_{t-1}, u_{t-1}) P(s_{t-1}, u_{t-1}) \right)$$

Question 3 (c)

⟹ The probability of hidden states given a sequence of ~~obvers~~ observations is :

$$P(S_t = s, U_t = u \mid y_0, \ldots, y_t, z_0, \ldots, z_t)$$

$$= \frac{P(S_t = s, U_t = u, y_0, \ldots, y_t, z_0, \ldots, z_t) * P(y_0, \ldots, y_t, z_0, \ldots, z_t \mid S_t = S, U_t = u)}{P(y_0, \ldots, y_t, z_0, \ldots, z_t)}$$

- The most likely sequence is :

$$\arg\max P(S_t = s, U_t = u \mid y_0, \ldots, y_t, z_0, \ldots, z_t).$$

To solve this one, we need to compute the numerator and denomitor.

However, the denominator is merely normalization so we need just solve this numerator.

For the first half, we can use forward algorithm.

Denote $\alpha_t(s, u) = P(S_t = s, U_t = u, y_0, \ldots, y_t, z_0, \ldots, z_t)$

To solve $\alpha_t(s,u)$, here is the step:

(I) Recursion:

$$P(S_{t+1}=s, U_{t+1}=u, y_0, \cdots, y_{t+1}, z_0, \cdots, z_{t+1})$$

$$= P(y_{t+1}, z_{t+1} \mid S_{t+1}=s, U_{t+1}=u) *$$

$$\sum_{s',u'} \{ P(S_{t+1}=s \mid S_t=s', U_t=u') P(U_{t+1}=u \mid S_t=s', U_t=u')$$

$$* P[S_t \mid s', U_t=u' \mid y_0, \cdots, y_{t+1}, z_0, \cdots, z_{t+1}] \} \quad \cdots\cdots (1)$$

so, if we denote

$$b_{t+1}(s,u) = P(S_{t+1}=s, U_{t+1}=u, y_0, \cdots, y_{t+1}, z_0, \cdots, z_{t+1})$$

$$q_{\substack{sy_{t+1}\\uz_{t+1}}} = P(y_{t+1}, z_{t+1} \mid S_{t+1}=s, U_{t+1}=u)$$

Then each time, just update:

$$b_{t+1}(s,u) = q_{\substack{sy_{t+1}\\uz_{t+1}}} \sum_{s',u'} q_{s's}\, q_{u'u}\, b_t(s',u')$$

(2) Initialization:

$b_1(s, u)$

$= P(S_1 = s, U_1 = u \mid y_0, z_0)$

$= \dfrac{P(S_1 = s, U_1 = u, y_0, z_0)}{P(y_0, z_0)}$

$$= \frac{b_0(s)\, b_0(u)\, q_{sy_0}\, q_{uz_0}}{\sum_{s'} \sum_{u'} b_0(s')\, b_0(u')\, q_{s'y_0}\, q_{u'z_0}},$$

This is the forward algorithm.

Question 3 (d)

If it is a loosely coupled network, the inference algorithm going to change mainly for the updating part, that is equation 1 from Question 3 (c) is changed to below:

$$P(S_{t+1} = s, U_{t+1} = u, y_0, \ldots, y_{t+1}, z_0, \ldots, z_{t+1})$$

$$= P(y_{t+1}, z_{t+1} \mid S_{t+1} = s, U_{t+1} = u) *$$

$$\sum_{s', u'} P(S_{t+1} = s \mid S_t = s', U_{t-k} = u')\, P(U_{t+1} = u \mid S_{t-k} = s', U_t = u')$$

$$P(S_t = s', U_t = u' \mid y_0, \ldots, y_{t+1}, z_0, \ldots, z_{t+1})$$

That's conditional probability of $S_{t+1} = s$ and $U_{t+1} = u$ are changed.

Question 3 (e)

This can be solved using Baum-Welch algorithm. This is an Expectation - Maximization algorithm for fitting HMM parameter.

For HMMs, the missing data is the state sequence, so we start with an initial guess about the model parameters and alternate the following steps:

(1) Estimate the probability of the state sequence given the observation sequence,

For every s, u, t, : $P(S_t = s, U_t = u \mid y_0, \dots, y_t, z_0, \dots, z_t)$

For every s, s', u, u', t :

$$P(S_t = s, S_{t+1} = s', U_t = u, U_{t+1}, u' \mid y_0, \dots, y_t, z_0, \dots, z_t)$$

(2)

Fit new model parameters based on the completed data (using the maximum likelihood algorithm).

References:

- Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006
- Kevin P. Murphy, *Machine Learning. A Probabilistic Perspective,* MIT Press, 2012.
- https://en.wikipedia.org/wiki/Eigenvalues_and_eigenvectors
- http://www.cs.cmu.edu/~dgovinda/pdf/recog/brand-cvpr97.pdf
- https://www.norsys.com/tutorials/netica/secA/tut_A1.htm