SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

01.112 Machine Learning

HW 2

Barry Tee Wei Cong

1001549

WRITTEN QUESTIONS Qn 1

affine function : $y = Ax + c$
(does not need to fix 0 origin)
(unlike linear functions)

Question 1. Assume a function $f : \mathbb{R}^n \to R$ is continuous. As discussed in class, if it is a convex function, then it satisfies the following property:

Property A: $f\left(\frac{x_1 + x_2}{2}\right) \leq \frac{f(x_1) + f(x_2)}{2}$

for any $x_1, x_2 \in \mathbb{R}^n$.
Show that the following statements are true (with formal proofs):

(a) If two functions $f(x)$ and $g(x)$ both satisfy Property A, then the following function also satisfies Property A:

$$h(x) = f(x) + g(x)$$

(5 points)

(b) If two functions $f(x)$ and $g(x)$ both satisfy Property A, then the following function also satisfies Property A:

$$h(x) = \max(f(x), g(x))$$

(5 points)

(a) $d = v - u$ (1)

$[u, v] = \{ u + \lambda d : \lambda \in [0,1] \}$ (2)

Sub (1) into (2):

$= \{ u + \lambda(v-u) : \lambda \in [0,1] \}$

$= \{ (1-\lambda)u + \lambda v : \lambda \in [0,1] \}$

$C \in \mathbb{R}^n$ is convex if $\forall x, y \in \mathbb{R}$, $[x, y] \subseteq C$

$d = x_2 - x_1$ (1)

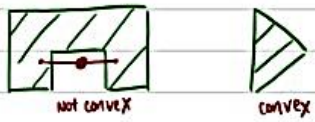
$[x_1, x_2] = \{ x_1 + \lambda x_2 : \lambda \in [0,1] \}$ (2)

Sub (1) into (2)

$= \{ x_1 + \lambda(x_2 - x_1) : \lambda \in [0,1] \}$

$= \{ (1-\lambda)x_1 + \lambda x_2 : \lambda \in [0,1] \}$

$C \in \mathbb{R}^n$ is convex if $\forall x, y \in \mathbb{R}$, $[x, y] \subseteq C$



Take $x_1, x_2 \in C$ $\exists\ \lambda_x, \lambda_y \in [0,1]$ st $f(x) = (1-\lambda_x)x_1 + \lambda_x x_2$

$g(x) = (1-\lambda_y)x_1 + \lambda_y x_2$

Proof

Take $h(x) \in [f(x), g(x)]$. Then $\exists\ \lambda \in [0,1]$ st $h(x) = (1-\lambda)x_1 + \lambda x_2$

$h(x) = \alpha[f(x)] + \beta[g(x)]$, $\alpha, \beta \geq 0$

$= \alpha(1-\lambda_x)x_1 + \alpha\lambda_x x_2 + \beta(1-\lambda_y)x_1 + \beta\lambda_y x_2$, $\alpha + \beta = 1$

$= (\alpha+\beta)x_1 - (\alpha\lambda_x + \beta\lambda_y)x_1 + (\alpha\lambda_x + \beta\lambda_y)x_2$, $\gamma = \alpha\lambda_x + \beta\lambda_y$

$= 1 \cdot x_1 - \gamma x_1 + \gamma x_2$

$= \underbrace{(1-\gamma)}_{\text{non-ve}} x_1 + \underbrace{\gamma}_{\text{non-ve}} x_2 \rightarrow h(x) \in C \rightarrow [x_1, x_2] \subseteq C$

→ $h(x)$ is also convex if $f(x)$ and $g(x)$ are both convex.

$h(x)$ in line segment b/w x_1 and x_2 and hence in C where C is convex.

↓ Aka intersection of 2 convex sets.

(b) $h((1-\lambda)x_1 + \lambda x_2) = \max\left(f((1-\lambda_x)x_1 + \lambda_x x_2),\ g((1-\lambda_y)x_1 + \lambda_y x_2)\right)$

$\leq \max((1-\lambda_x)f(x_1),\ \lambda_x f(x_2)) + \max((1-\lambda_y)f(x_1),\ \lambda_y f(x_2))$

$\leq (1-\lambda)x_1 + \lambda x_2$

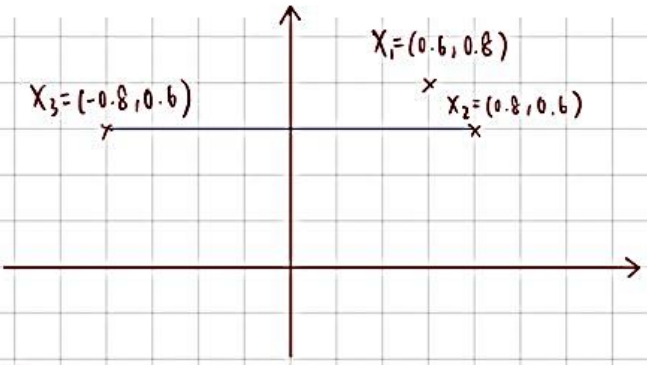
2(a) Code, Input & Instructions Attached

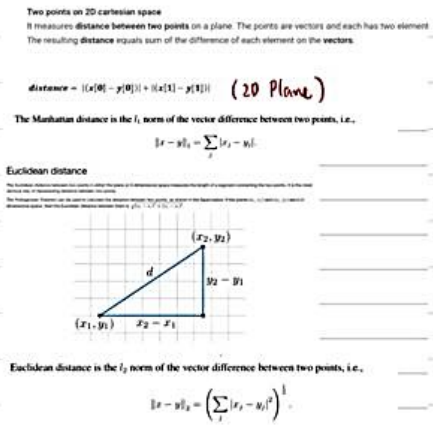
2(b) (b) Explain in English how could you use the validation set to select the model (with the parameters θ, θ_0) to use on the test set? *(2 points)*

As shown in 2(a), mean squared error would decrease until ≈ 5500 iteration and start to increase thereafter. This means that the error increases after that number of iterations. This means that is an ideal number of iterations for tuning the parameters when setting the number of iterations to tune the weights based on the training set.

Qn 3a

(a) Consider a set of points $X = (0.6, 0.8), (0.8, 0.6), (-0.8, 0.6)$. Compute the value of z that minimizes $\sum_{x \in X} d(x, z)$ when $d(x, z)$ is defined as follows respectively: 1) the Euclidean distance between x and z, and 2) the Manhattan distance between x and z. *(5 points)*




Two points on 2D cartesian space

It measures **distance between two points** on a plane. The points are vectors and each has two element. The resulting **distance** equals sum of the difference of each element on the **vectors**.

$distance = |(x[0] - y[0])| + |(x[1] - y[1])|$ (2D Plane)

The Manhattan distance is the l_1 norm of the vector difference between two points, i.e.,

$$\|x - y\|_1 = \sum_i |x_i - y_i|.$$

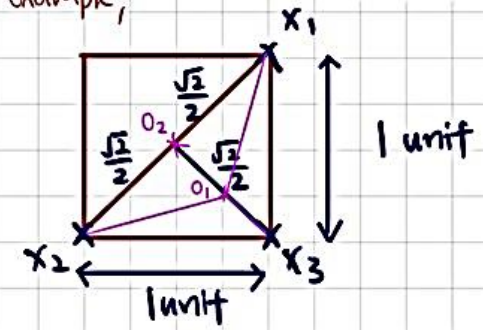
Euclidean distance

Euclidean distance is the l_2 norm of the vector difference between two points, i.e.,

$$\|x - y\|_2 = \left(\sum_i |x_i - y_i|^2\right)^{\frac{1}{2}}.$$

3(a)

Using a 1 x1 unit as an example,



Using K-Means Algo,

From O_1 to X_2, Euclidean distance = $\sqrt{0.75^2 + 0.25^2} \approx 0.7906$

O_1 to X_1, Euclidean distance = " ≈ 0.7906

O_1 to X_3, Euclidean distance = $\sqrt{0.25^2 + 0.25^2} \approx 0.3536$

Total Euclidean distance = 1.9348

From O_1 to X_2, Manhattan distance = $0.75 + 0.25 = 1$

O_1 to X_1, Manhattan distance = $0.75 + 0.25 = 1$

O_1 to X_3, Manhattan distance = $0.25 + 0.25 = 0.5$

Total Manhattan distance = $1 + 1 + 0.5 = 2.5$

From O_2 to $X_1/X_2/X_3$, Euclidean distance = $\sqrt{2}/2 \approx 0.707$

Total Euclidean distance $\approx$ 2.121

From O_2 to X_2, Manhattan distance = $0.707 + 0.5 = 1.207$

O_2 to X_1, Manhattan distance = " $= 1.207$

O_2 to X_3, Manhattan distance = " $= 1.207$

Total Manhattan distance = 3.621

Using K-Mediods Algo, X_3 as representative

→ Total Euclidean distance = $1 + 1 = 2$

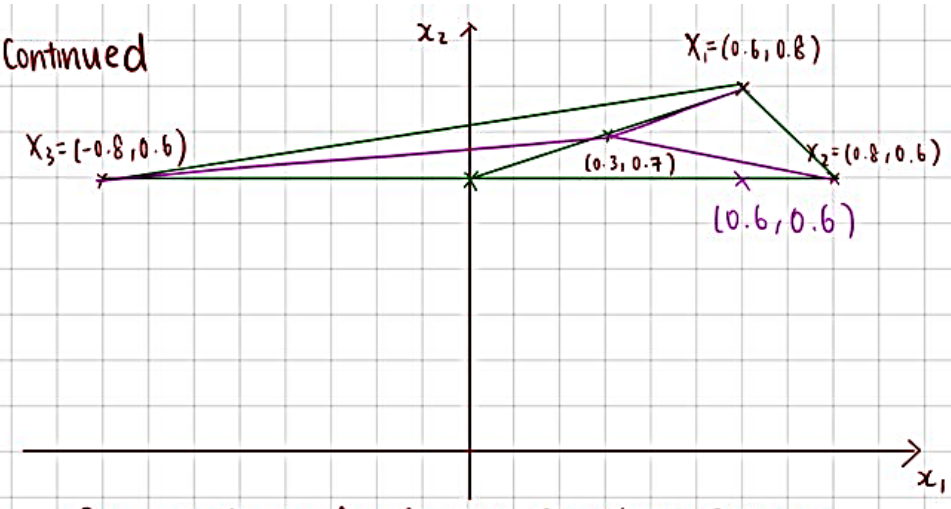
→ Total Manhattan distance = $1 + 1 = 2$

Conclusion: K-means Algo produce shortest Euclidean Distance compared to K-Mediods Algo

cont. → 3a.

Qn 3a Continued

3a. Continued



1. Centre Cordinate for Shortest Euclidean Distance

$$= \left(\frac{\left(\frac{0.8+(-0.8)}{2}\right) + 0.6}{2} , \frac{\frac{0.6+(0.6)}{2} + 0.8}{2} \right) = (0.3, 0.7)$$

Euclidean Distance

$$= \sqrt{(0.6-0.3)^2 + (0.8-0.7)^2} + \sqrt{(0.8-0.3)^2 + (0.6-0.7)^2} + \sqrt{(-0.8-0.3)^2 + (0.6-0.7)^2}$$

$$= 1.931$$

2. For Manhattan Distance, if use X_1 as representative

Distance from X_3 to X_1 $= |-0.8-0.6| + |0.6-0.8| = 1.6$

X_2 to X_1 $= |0.8-0.6| + |0.6-0.8| = 0.4$

Total Manhattan Distance $= 2.0$

For Manhattan Distance, if use X_2 as representative

Distance from X_3 to X_2 $= |-0.8-0.8| + |0.6-0.6| = 1.6$

X_1 to X_2 $= |0.8-0.6| + |0.6-0.8| = 0.4$

Total Manhattan Distance $= 2.0$

x_1	x_2
-0.8	0.6
0.6	0.6
0.8	0.8

For Manhattan Distance, choose medium x_1, x_2 coordinates

Set $C_2(0.6, 0.6)$ as representative

Distance from X_3 to C_2 $= |-0.8-0.6| + |0.6-0.6| = 1.4$

X_2 to C_2 $= |0.8-0.6| + |0.6-0.6| = 0.2$

X_1 to C_2 $= |0.6-0.6| + |0.8-0.6| = 0.2$

Total Manhattan Distance $= 1.8$

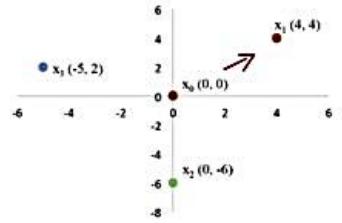
In Conclusion, $d(x,z) = (0.3, 0.4)$ for shortest Euclidean Distance

$d(x,z) = (0.6, 0.6)$ for shortest Manhattan Distance.

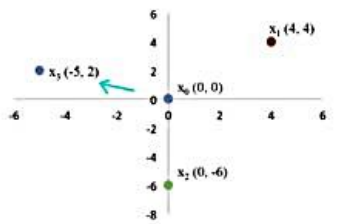
Qn 3b

(b) The following figures (points in the same cluster have the same color) are produced by the k-medoids algorithm for $k = 3$ clusters using l_1 , l_2 , and l_∞ distance measures. Indicate which distance measure is used for each figure. *(5 points)*

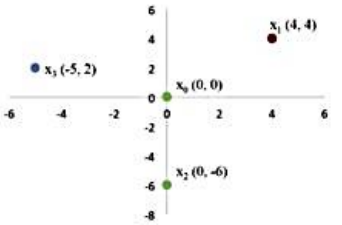
A.



B.



C.



Two points on 2D cartesian space

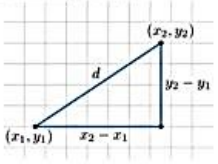
It measures **distance between two points** on a plane. The points are vectors and each has two elements.
The resulting **distance** equals sum of the difference of each element on the **vectors**.

$distance = |(x[0] - y[0])| + |(x[1] - y[1])|$ (2D Plane)

The Manhattan distance is the l_1 norm of the vector difference between two points, i.e.,

$$\|x - y\|_1 = \sum_j |x_j - y_j|.$$

Euclidean distance



Euclidean distance is the l_2 norm of the vector difference between two points, i.e.,

$$\|x - y\|_2 = \left(\sum_j |x_j - y_j|^2\right)^{\frac{1}{2}}.$$

The l_∞ distance is the maximum absolute element in the vector difference between two points, i.e.,

$$\|x - y\|_\infty = \max_j |x_j - y_j|.$$

3(b) Distance Measure

For x_3, Manhattan Dist = 5 + 2 = 7
Euclidean Dist = $\sqrt{5^2 + 2^2}$ = $\sqrt{25+4}$ = $\sqrt{29}$ ≈ 5.39 B.
l_∞ Dist = max (x_1, x_2) = max (5, 2) = 5

For x_2, Manhattan Dist = 0 + 6 = 6 A.
Euclidean Dist = $\sqrt{0^2 + 6^2}$ = $\sqrt{6^2}$ = $\sqrt{36}$ = 6
l_∞ Dist = max (x_1, x_2) = max (0, 6) = 6

For x_1, Manhattan Dist = 4 + 4 = 8
Euclidean Dist = $\sqrt{4^2 + 4^2}$ = $\sqrt{16+16}$ = $\sqrt{32}$ ≈ 5.66 C.
l_∞ Dist = max (x_1, x_2) = max (4, 4) = 4

Rationale: Find the shortest distance from point to origin using the distance method.
ie Euclidean Distance, $X_1 = 5.66$, $X_2 = 6$, $X_3 = 5.39$
↑ Shortest Distance
Hence B is labelled blue with origin.

In conclusion Fig A uses l_1
Fig B uses l_2
Fig C uses l_∞

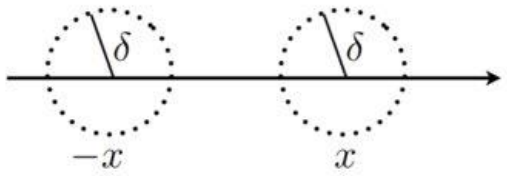
Qn 4

Question 4. Each iteration of the k-means algorithm consists of two steps: assigning points to centroids, and updating the centroids based on the points assigned to them. Assume that the number of clusters $k = 2$.

(a) If the centroids are initialized to be the means of two *well-separated* clusters, will the centroids change after the first iteration? (A yes/no answer suffices.) *(1 point)*

(b) If the centroids are initialized by setting each to a random point from each of the two well-separated clusters, how many iterations does it take for k-means to converge? Explain your answer. *(4 point)*

(c) In the figure below, we have two spherical (circle-like) clusters of radius δ that are centered at locations $-x$ and x. For what values of x would the k-means algorithm fail to find the centers of the two clusters regardless of the initialization? Explain your answer. *(5 points)*



(A) No

(B) Since the clusters are well-separated, the initialization will not change assignment of points to centroids. The centroids become cluster means after the first iteration.

(C) Given radius to be labelled delta units, if |x| < delta and there lies points between both intersected circles, then the k-means algorithm would fail to find the centers of the two clusters regardless of the initialization. This is because the two circle-like clusters that overlap causes the points within the overlap to be unable to be assigned just one membership. If x = delta, then the point that lie on both circumference would also face the same problem causing the k-means algorithm to fail.

CODING QUESTIONS Qn 2a

Instructions on How to Run the Code for Qn 2a

1) Download Anaconda. Highly recommended to download Anaconda's latest Python 3 version (currently Python 3.6).
2) Install the version of Anaconda which you downloaded, following the instructions on the download page. Please ensure that Anaconda installer is installed via ***Run By Administrator***. This is to prevent an error in installing Jupyter Notebook and access the local directory.
3) **Unzip the Folder named HW2_1001549_BarryTeeWeiCong**
4) Open Jupyter Notebook and access the folder to open Homework2.2final.ipynb.
5) Press Play Button once to obtain the answer for Qn2a.
6) Feel free to call Barry Tee at +65 81393748 if there are any errors running the code.
7) Below is my code with the answers to each question

Qn 2a

```
1   import csv
2   import os
3   import numpy as np
4   import pandas as pd
5   from matplotlib import pyplot as plt
6
7   cost_train = []
8   cost_test = []
9   cost_valid = []
10  cost_total = []
11  theta = []
12
13  def get_data_train(path):    # path to read data from
14      raw_panda_data = pd.read_csv(path, header = None) #pandas
15      raw_panda_data.insert(0, 'Ones', 1) # python, append a column of ones to the front of the data set for normalization
16      num_columns = raw_panda_data.shape[1]                    # numpy, (num_rows, num_columns)
17      panda_X = raw_panda_data.iloc[: , 0:num_columns-1]          # pandas, [ slice_of_rows, slice_of_columns 0 to 21 ]
18      panda_y = raw_panda_data.iloc[: , num_columns-1:num_columns]  # pandas, [ slice_of_rows, slice_of_columns 21 to 22 ]
19
20      X1 = np.matrix(panda_X.values)   # numpy, pandas.DataFrame -> numpy.ndarray -> numpy.matrix
21      y1 = np.matrix(panda_y.values)   # numpy, pandas.DataFrame -> numpy.ndarray -> numpy.matrix
22      #print (X1)
23      return X1, y1
24
25  def get_data_test(path):    # path to read data from
26      raw_panda_data = pd.read_csv(path, header = None) #pandas
27      raw_panda_data.insert(0, 'Ones', 1) # python, append a column of ones to the front of the data set for normalization
28      num_columns = raw_panda_data.shape[1]                    # numpy, (num_rows, num_columns)
29      panda_X = raw_panda_data.iloc[:,0:num_columns-1]          # pandas, [ slice_of_rows, slice_of_columns 0 to 21 ]
```

```
30      panda_y = raw_panda_data.iloc[:,num_columns-1:num_columns]  # pandas, [ slice_of_rows, slice_of_columns 21 to 22 ]
31
32      X2 = np.matrix(panda_X.values)   # numpy, pandas.DataFrame -> numpy.ndarray -> numpy.matrix
33      y2 = np.matrix(panda_y.values)   # numpy, pandas.DataFrame -> numpy.ndarray -> numpy.matrix
34      return X2, y2
35
36  def get_data_valid(path):    # path to read data from
37      raw_panda_data = pd.read_csv(path, header = None) #pandas
38      raw_panda_data.insert(0, 'Ones', 1) # python, append a column of ones to the front of the data set for normalization
39      num_columns = raw_panda_data.shape[1]                    # numpy, (num_rows, num_columns)
40      panda_X = raw_panda_data.iloc[:,0:num_columns-1]          # pandas, [ slice_of_rows, slice_of_columns 0 to 21 ]
41      panda_y = raw_panda_data.iloc[:,num_columns-1:num_columns]  # pandas, [ slice_of_rows, slice_of_columns 21 to 22 ]
42
43      X3 = np.matrix(panda_X.values)   # numpy, pandas.DataFrame -> numpy.ndarray -> numpy.matrix
44      y3 = np.matrix(panda_y.values)   # numpy, pandas.DataFrame -> numpy.ndarray -> numpy.matrix
45      return X3, y3
46
47  def compute_mean_square_error(X, y, theta):
48      summands = np.dot(X * theta.T - y, 2)  #numpy, (X*T -y)^2
49      mse = np.sum(summands) / (2 * len(X))
50      return mse
51
52  def gradient_descent(X1, y1, X2, y2, X3, y3, learning_rate, num_iterations):
53      num_sample = num_iterations / 100
54      num_parameters = X1.shape[1]                        # numpy, dim theta = 21 columns
55      theta = np.matrix([0.0 for i1 in range(num_parameters)])    # numpy, init theta matrix with 21 columns
56
57      for i3 in range(num_iterations):
58          error_train = np.repeat((X1 * theta.T) - y1, num_parameters, axis=1)
59          error_derivative_train = np.sum(np.multiply(error_train, X1), axis=0)
60          if i3 % 100 == 0 :
```

```
            if i3 == 0:
                print ('')
            if i3 > 0:
                #print(i3)
                theta = theta - (learning_rate / len(y1)) * error_derivative_train
                print (i3 , 'iteration, weights =' , theta )
                cost = [0.0 for i2 in range(num_iterations)]
                cost[i3] = abs(compute_mean_square_error(X1, y1, theta))
                cost_train.append(cost[i3])
                cost[i3] = abs(compute_mean_square_error(X2, y2, theta))
                cost_test.append(cost[i3])
                cost[i3] = abs(compute_mean_square_error(X3, y3, theta))
                cost_valid.append(cost[i3])
    cost_total.append(cost_train)
    cost_total.append(cost_test)
    cost_total.append(cost_valid)
    #print(cost_total)
    return theta, cost_total

def plot_line(cost_ref):
    x_values = [i4 for i4 in range(len(cost_train))] # plotting from 0 to 99
    #print(len(cost_train))
    #y1_values = cost_total[0]
    #y2_values = cost_total[1]
    #y3_values = cost_total[2]
    #plt.plot(x_values, y1_values, 'b')
    #plt.plot(x_values, y2_values, 'r')
    #plt.plot(x_values, y3_values, 'g')

    for i5 in range(len(cost_total)):
        #print (i5)
```

```
92              plt.plot( x_values, cost_total[i5] )
93          plt.title("Graph for Training, Testing and Validation Set")
94          plt.xlabel("No. of Iterations")
95          plt.ylabel("Mean Square Errors")
96          plt.legend(('error_train','error_test','error_valid'))
97          plt.show()

98      X1, y1 = get_data_train(os.getcwd() + '/train_warfarin.csv')
99      X2, y2 = get_data_test(os.getcwd() + '/test_warfarin.csv')
100     X3, y3 = get_data_valid(os.getcwd() + '/validation_warfarin.csv')
101     theta, cost = gradient_descent(X1, y1, X2, y2, X3, y3, 0.00015, 10000)

102     plot_line(cost_total)
```

Output for Python Code for Qn2a

```
100 iteration, weights = [[  7.20000000e-05   1.24270242e-03   3.80250000e-
05   8.25750000e-05
    5.76300000e-05   1.95000000e-05   2.55000000e-05   1.05000000e-05
    1.65000000e-05   1.35000000e-05   1.05000000e-05   1.05000000e-05
    7.50000000e-06   1.05000000e-05   1.20000000e-05   7.50000000e-06
    1.95000000e-05   2.10000000e-05   1.20000000e-05   1.95000000e-05
    3.60000000e-05]]
200 iteration, weights = [[  1.41292363e-04   2.44185514e-03   7.45780688e-
05   1.62029734e-04
    1.13110794e-04   3.84299436e-05   5.01977545e-05   2.04284515e-05
    3.22362134e-05   2.66264718e-05   2.04289866e-05   2.07421590e-05
    1.46761522e-05   2.05315157e-05   2.36112556e-05   1.46758222e-05
    3.84616228e-05   4.11128717e-05   2.34325146e-05   3.82853541e-05
    7.03804399e-05]]
300 iteration, weights = [[  2.07972118e-04   3.59898645e-03   1.09710866e-
04   2.38473719e-04
    1.66517815e-04   5.68098368e-05   7.41214125e-05   2.98054221e-05
    4.72354470e-05   3.93925201e-05   2.98070044e-05   3.07355299e-05
    2.15398251e-05   3.01109898e-05   3.48474070e-05   2.15388422e-05
    5.69037602e-05   6.03697510e-05   3.43174664e-05   5.63811408e-05
    1.03198162e-04]]
400 iteration, weights = [[  2.72130960e-04   4.71557100e-03   1.43473238e-
04   3.12012628e-04
    2.17923849e-04   7.46589833e-05   9.72981348e-05   3.86502751e-05
    6.15235668e-05   5.18107897e-05   3.86533949e-05   4.04888480e-05
    2.81019879e-05   3.92542878e-05   4.57216155e-05   2.81000363e-05
    7.48446412e-05   7.88006810e-05   4.46740789e-05   7.38115589e-05
    1.34508013e-04]]
500 iteration, weights = [[  3.33857364e-04   5.79303169e-03   1.75913283e-
04   3.82748426e-04
    2.67399129e-04   9.19960095e-05   1.19754129e-04   4.69816942e-05
    7.51255311e-05   6.38934818e-05   4.69868205e-05   5.00105418e-05
    3.43732251e-05   4.79767187e-05   5.62465808e-05   3.43699956e-05
    9.23018550e-05   9.64346505e-05   5.45209011e-05   9.05999577e-05
    1.64362915e-04]]
600 iteration, weights = [[  3.93236703e-04   6.83274152e-03   2.07077409e-
04   4.50779501e-04
    3.15011423e-04   1.08838888e-04   1.41514685e-04   5.48177076e-05
    8.80654223e-05   7.56523693e-05   5.48252889e-05   5.93087445e-05
    4.03637497e-05   5.62930540e-05   6.64345568e-05   4.03589395e-05
    1.09292374e-04   1.13299631e-04   6.38758310e-05   1.06768867e-04
    1.92813935e-04]]
700 iteration, weights = [[  4.50351351e-04   7.83602525e-03   2.37010398e-
04   5.16200786e-04
    3.60826118e-04   1.25204961e-04   1.62604201e-04   6.21757107e-05
    1.00366478e-04   8.70988120e-05   6.21861758e-05   6.83913033e-05
```

```
    4.60834164e-05   6.42175470e-05   7.62973673e-05   4.60767290e-05
    1.25832573e-04   1.29422613e-04   7.27561383e-05   1.22340026e-04
    2.19910345e-04]]
800 iteration, weights = [[  5.05280793e-04   8.80416120e-03   2.65755459e-
04   5.79103884e-04
    4.04906311e-04   1.41110962e-04   1.83046223e-04   6.90724886e-05
    1.12051119e-04   9.82437709e-05   6.90862470e-05   7.72657905e-05
    5.15417338e-05   7.17639508e-05   8.58464214e-05   5.15328787e-05
    1.41938256e-04   1.44829638e-04   8.11784869e-05   1.37334412e-04
    2.45699689e-04]]
900 iteration, weights = [[  5.58101725e-04   9.73838281e-03   2.93354285e-
04   6.39577183e-04
    4.47312881e-04   1.56573037e-04   2.02863469e-04   7.55242371e-05
    1.23140982e-04   1.09097823e-04   7.55416802e-05   8.59395123e-05
    5.67478771e-05   7.89455356e-05   9.50927275e-05   5.67365695e-05
    1.57624668e-04   1.59545836e-04   8.91589559e-05   1.51772265e-04
    2.70227843e-04]]
1000 iteration, weights = [[  6.08888150e-04   1.06398802e-02   3.19847105e
-04   6.97705970e-04
    4.88104571e-04   1.71606766e-04   2.22077859e-04   8.15465838e-05
    1.33656941e-04   1.19671173e-04   8.15680852e-05   9.44195188e-05
    6.17106993e-05   8.57751063e-05   1.04046908e-04   6.16966602e-05
    1.72906522e-04   1.73595454e-04   9.67130600e-05   1.65673114e-04
    2.93539072e-04]]
1100 iteration, weights = [[  6.57711476e-04   1.15098018e-02   3.45272739e
-04   7.53572537e-04
    5.27338066e-04   1.86227182e-04   2.40710546e-04   8.71546079e-05
    1.43619141e-04   1.29973668e-04   8.71805240e-05   1.02712613e-04
    6.64387428e-05   9.22650182e-05   1.12719212e-04   6.64216985e-05
    1.87798015e-04   1.87001887e-04   1.03855770e-04   1.79055804e-04
    3.15676089e-04]]
1200 iteration, weights = [[  7.04640603e-04   1.23492557e-02   3.69668641e
-04   8.07256290e-04
    5.65068056e-04   2.00448789e-04   2.58781938e-04   9.23628595e-05
    1.53047016e-04   1.40014810e-04   9.23935300e-05   1.10825357e-04
    7.09402504e-05   9.84271930e-05   1.21119530e-04   7.09199323e-05
    2.02312842e-04   1.99787712e-04   1.10601529e-04   1.91938520e-04
    3.36680108e-04]]
1300 iteration, weights = [[  7.49742014e-04   1.31593111e-02   3.93070953e
-04   8.58833847e-04
    6.01347318e-04   2.14285584e-04   2.76311730e-04   9.71853778e-05
    1.61959323e-04   1.49803767e-04   9.72211268e-05   1.18764087e-04
    7.52231753e-05   1.04273134e-04   1.29257404e-04   7.51993200e-05
    2.16464222e-04   2.11974709e-04   1.16964278e-04   2.04338805e-04
    3.56590895e-04]]
1400 iteration, weights = [[  7.93079859e-04   1.39409997e-02   4.15514549e
-04   9.08379139e-04
```

```
    6.36226771e-04   2.27751072e-04   2.93318922e-04   1.01635710e-04
    1.70374155e-04   1.59349385e-04   1.01676845e-04   1.26534912e-04
    7.92951918e-05   1.09813941e-04   1.37142043e-04   7.92675407e-05
    2.30264907e-04   2.23583899e-04   1.22957463e-04   2.16273589e-04
    3.75446821e-04]]
1500 iteration, weights = [[  8.34716037e-04   1.46953169e-02   4.37033077e
-04   9.55963501e-04
    6.69755552e-04   2.40858285e-04   3.09821850e-04   1.05726926e-04
    1.78308975e-04   1.68660201e-04   1.05773743e-04   1.34143729e-04
    8.31637048e-05   1.15060325e-04   1.44782330e-04   8.31320041e-05
    2.43727203e-04   2.34635562e-04   1.28594063e-04   2.27759208e-04
    3.93284913e-04]]
1600 iteration, weights = [[  8.74710274e-04   1.54232233e-02   4.57659005e
-04   1.00165576e-03
    7.01981071e-04   2.53619795e-04   3.25838207e-04   1.09471640e-04
    1.85780631e-04   1.77744451e-04   1.09524417e-04   1.41596229e-04
    8.68358595e-05   1.20022619e-04   1.52186839e-04   8.67998597e-05
    2.56862984e-04   2.45149267e-04   1.33886599e-04   2.38811424e-04
    4.10140895e-04]]
1700 iteration, weights = [[  9.13120200e-04   1.61256458e-02   4.77423663e
-04   1.04552235e-03
    7.32949074e-04   2.66047737e-04   3.41385062e-04   1.12882022e-04
    1.92805380e-04   1.86610082e-04   1.12941023e-04   1.48897901e-04
    9.03185502e-05   1.24710795e-04   1.59363842e-04   9.02780062e-05
    2.69683706e-04   2.55143896e-04   1.38847152e-04   2.49445446e-04
    4.26049239e-04]]
1800 iteration, weights = [[  9.50001425e-04   1.68034785e-02   4.96357281e
-04   1.08762733e-03
    7.62703702e-04   2.78153816e-04   3.56478888e-04   1.15969813e-04
    1.99398908e-04   1.95264763e-04   1.16035293e-04   1.56054044e-04
    9.36184292e-05   1.29134475e-04   1.66321321e-04   9.35731002e-05
    2.82200425e-04   2.64637669e-04   1.43487382e-04   2.59675949e-04
    4.41043208e-04]]
1900 iteration, weights = [[  9.85407603e-04   1.74575846e-02   5.14489027e
-04   1.12803256e-03
    7.91287549e-04   2.89949328e-04   3.71135578e-04   1.18746346e-04
    2.05576351e-04   2.03715893e-04   1.18818542e-04   1.63069768e-04
    9.67419156e-05   1.33302943e-04   1.73066977e-04   9.66915649e-05
    2.94423807e-04   2.73648166e-04   1.47818537e-04   2.69517094e-04
    4.55154896e-04]]
2000 iteration, weights = [[  1.01939051e-03   1.80887965e-02   5.31847045e
-04   1.16679770e-03
    8.18741712e-04   3.01445174e-04   3.85370469e-04   1.21222553e-04
    2.11352314e-04   2.11970611e-04   1.21301693e-04   1.69950006e-04
    9.96952030e-05   1.37225158e-04   1.79608241e-04   9.96395976e-05
    3.06364146e-04   2.82192351e-04   1.51851471e-04   2.78982543e-04
    4.68415272e-04]]
```

```
2100 iteration, weights = [[ 0.001052    0.01869792  0.00054846  0.00120398
0.00084511  0.00031265
   0.0003992   0.00012341  0.00021674  0.00022004  0.0001235   0.0001767
   0.00010248  0.00014091  0.00018595  0.00010242  0.00031803  0.00029029
   0.0001556   0.00028809  0.00048085]]
2200 iteration, weights = [[ 0.00108328  0.01928572  0.00056435  0.00123964
0.00087042  0.00032358
   0.00041263  0.00012532  0.00022176  0.00022792  0.00012541  0.00018332
   0.00010511  0.00014437  0.00019211  0.00010505  0.00032944  0.00029795
   0.00015906  0.00029684  0.0004925 ]]
2300 iteration, weights = [[ 0.00111329  0.01985296  0.00057955  0.00127382
0.00089472  0.00033424
   0.00042569  0.00012695  0.00022641  0.00023562  0.00012705  0.00018982
   0.00010759  0.0001476   0.00019808  0.00010752  0.00034058  0.00030519
   0.00016226  0.00030525  0.00050338]]
2400 iteration, weights = [[ 0.00114206  0.02040035  0.00059407  0.00130658
0.00091803  0.00034464
   0.00043838  0.00012833  0.00023072  0.00024316  0.00012844  0.00019621
   0.00010992  0.00015062  0.00020387  0.00010984  0.00035149  0.00031202
   0.00016521  0.00031334  0.00051353]]
2500 iteration, weights = [[ 0.00116964  0.0209286   0.00060795  0.00133797
0.00094041  0.00035478
   0.00045072  0.00012946  0.00023469  0.00025053  0.00012957  0.00020248
   0.00011211  0.00015343  0.00020949  0.00011203  0.00036216  0.00031847
   0.0001679   0.00032112  0.00052296]]
2600 iteration, weights = [[ 0.00119608  0.02143836  0.0006212   0.00136803
0.00096187  0.00036469
   0.00046272  0.00013034  0.00023833  0.00025774  0.00013046  0.00020864
   0.00011416  0.00015605  0.00021495  0.00011407  0.00037259  0.00032454
   0.00017035  0.00032859  0.0005317 ]]
2700 iteration, weights = [[ 0.0012214   0.0219303   0.00063385  0.00139682
0.00098245  0.00037436
   0.00047438  0.00013099  0.00024167  0.0002648   0.00013112  0.0002147
   0.00011608  0.00015847  0.00022025  0.00011598  0.00038281  0.00033025
   0.00017257  0.00033577  0.00053978]]
2800 iteration, weights = [[ 0.00124566  0.02240504  0.00064592  0.00142438
0.00100218  0.00038381
   0.00048574  0.00013141  0.0002447   0.00027172  0.00013155  0.00022065
   0.00011787  0.00016071  0.00022539  0.00011776  0.00039282  0.00033561
   0.00017456  0.00034267  0.00054723]]
2900 iteration, weights = [[ 0.00126888  0.02286317  0.00065743  0.00145074
0.0010211   0.00039304
   0.00049678  0.00013162  0.00024744  0.00027849  0.00013177  0.00022651
   0.00011954  0.00016277  0.00023039  0.00011942  0.00040262  0.00034063
   0.00017634  0.0003493   0.00055405]]
3000 iteration, weights = [[ 0.00129111  0.02330529  0.0006684   0.00147596
0.00103922  0.00040206
```

```
   0.00050753  0.00013161  0.0002499   0.00028512  0.00013177  0.00023227
   0.00012108  0.00016466  0.00023524  0.00012096  0.00041222  0.00034532
   0.00017791  0.00035566  0.00056028]]
3100 iteration, weights = [[ 0.00131238  0.02373195  0.00067885  0.00150008
0.00105658  0.00041088
   0.000518    0.00013141  0.00025209  0.00029162  0.00013158  0.00023794
   0.00012252  0.00016639  0.00023995  0.00012239  0.00042163  0.0003497
   0.00017928  0.00036177  0.00056593]]
3200 iteration, weights = [[ 0.00133272  0.0241437   0.00068879  0.00152312
0.00107321  0.0004195
   0.00052819  0.00013101  0.00025402  0.00029799  0.00013118  0.00024352
   0.00012384  0.00016795  0.00024454  0.0001237   0.00043086  0.00035377
   0.00018045  0.00036763  0.00057103]]
3300 iteration, weights = [[ 0.00135216  0.02454106  0.00069824  0.00154513
0.00108913  0.00042794
   0.00053812  0.00013042  0.00025569  0.00030424  0.0001306   0.00024902
   0.00012505  0.00016935  0.00024899  0.00012491  0.00043991  0.00035755
   0.00018144  0.00037326  0.0005756 ]]
3400 iteration, weights = [[ 0.00137074  0.02492454  0.00070723  0.00156615
0.00110436  0.00043619
   0.00054779  0.00012964  0.00025712  0.00031037  0.00012984  0.00025443
   0.00012617  0.00017061  0.00025332  0.00012601  0.00044879  0.00036105
   0.00018225  0.00037866  0.00057965]]
3500 iteration, weights = [[ 0.00138849  0.02529462  0.00071576  0.00158621
0.00111893  0.00044427
   0.00055721  0.00012869  0.00025832  0.00031638  0.0001289   0.00025977
   0.00012718  0.00017173  0.00025753  0.00012702  0.0004575   0.00036428
   0.00018288  0.00038384  0.0005832 ]]
3600 iteration, weights = [[ 0.00140544  0.02565178  0.00072386  0.00160535
0.00113286  0.00045218
   0.00056639  0.00012758  0.00025929  0.00032228  0.00012779  0.00026503
   0.00012809  0.0001727   0.00026162  0.00012793  0.00046605  0.00036724
   0.00018334  0.00038881  0.00058627]]
3700 iteration, weights = [[ 0.00142161  0.02599646  0.00073153  0.00162359
0.00114618  0.00045993
   0.00057535  0.0001263   0.00026004  0.00032808  0.00012652  0.00027021
   0.00012892  0.00017354  0.00026561  0.00012874  0.00047445  0.00036995
   0.00018365  0.00039357  0.00058888]]
3800 iteration, weights = [[ 0.00143704  0.0263291   0.0007388   0.00164097
0.0011589   0.00046751
   0.00058408  0.00012486  0.00026058  0.00033377  0.00012509  0.00027533
   0.00012965  0.00017425  0.00026948  0.00012947  0.0004827   0.00037241
   0.00018379  0.00039814  0.00059104]]
3900 iteration, weights = [[ 0.00145174  0.02665013  0.00074568  0.00165752
0.00117105  0.00047495
   0.0005926   0.00012327  0.00026092  0.00033937  0.00012351  0.00028037
   0.0001303   0.00017484  0.00027325  0.0001301   0.0004908   0.00037464
```

```
   0.00018378  0.00040251  0.00059276]]
4000 iteration, weights = [[ 0.00146574  0.02695996  0.00075218  0.00167326
0.00118265  0.00048224
   0.00060092  0.00012153  0.00026106  0.00034486  0.00012178  0.00028535
   0.00013086  0.0001753   0.00027692  0.00013066  0.00049877  0.00037664
   0.00018363  0.0004067   0.00059407]]
4100 iteration, weights = [[ 0.00147907  0.02725897  0.00075831  0.00168823
0.00119371  0.00048939
   0.00060903  0.00011965  0.00026101  0.00035027  0.00011991  0.00029027
   0.00013135  0.00017565  0.0002805   0.00013113  0.0005066   0.00037842
   0.00018334  0.00041072  0.00059498]]
4200 iteration, weights = [[ 0.00149176  0.02754755  0.00076409  0.00170245
0.00120426  0.0004964
   0.00061695  0.00011763  0.00026078  0.00035558  0.0001179   0.00029512
   0.00013176  0.00017589  0.00028398  0.00013153  0.00051431  0.00037998
   0.00018291  0.00041456  0.0005955 ]]
4300 iteration, weights = [[ 0.00150382  0.02782607  0.00076953  0.00171596
0.00121432  0.00050328
   0.00062469  0.00011548  0.00026037  0.00036081  0.00011576  0.00029991
   0.00013209  0.00017602  0.00028737  0.00013185  0.00052189  0.00038134
   0.00018235  0.00041824  0.00059565]]
4400 iteration, weights = [[ 0.00151527  0.02809487  0.00077464  0.00172876
0.00122389  0.00051003
   0.00063224  0.00011321  0.00025979  0.00036595  0.0001135   0.00030465
   0.00013235  0.00017604  0.00029068  0.0001321   0.00052935  0.00038251
   0.00018167  0.00042175  0.00059544]]
4500 iteration, weights = [[ 0.00152615  0.0283543   0.00077943  0.0017409
0.001233    0.00051666
   0.00063963  0.00011081  0.00025905  0.00037102  0.00011111  0.00030933
   0.00013254  0.00017597  0.0002939   0.00013228  0.00053669  0.00038348
   0.00018086  0.00042512  0.00059487]]
4600 iteration, weights = [[ 0.00153646  0.02860468  0.00078392  0.00175239
0.00124167  0.00052317
   0.00064685  0.00010829  0.00025815  0.000376    0.0001086   0.00031396
   0.00013266  0.00017579  0.00029704  0.0001324   0.00054392  0.00038427
   0.00017993  0.00042834  0.00059398]]
4700 iteration, weights = [[ 0.00154622  0.02884634  0.00078812  0.00176325
0.0012499   0.00052957
   0.0006539   0.00010566  0.00025709  0.00038092  0.00010598  0.00031853
   0.00013272  0.00017553  0.0003001   0.00013244  0.00055104  0.00038488
   0.00017889  0.00043141  0.00059275]]
4800 iteration, weights = [[ 0.00155547  0.02907958  0.00079202  0.00177351
0.00125772  0.00053585
   0.0006608   0.00010292  0.00025589  0.00038576  0.00010325  0.00032306
   0.00013271  0.00017517  0.00030309  0.00013243  0.00055806  0.00038532
   0.00017774  0.00043434  0.00059122]]
```

```
4900 iteration, weights = [[ 0.00156421  0.02930469  0.00079566  0.00178319
0.00126514  0.00054203
   0.00066755  0.00010007  0.00025455  0.00039053  0.00010041  0.00032754
   0.00013265  0.00017472  0.00030601  0.00013235  0.00056498  0.00038559
   0.00017649  0.00043714  0.00058938]]
5000 iteration, weights = [[  1.57246093e-03   2.95219643e-02   7.99030208e
-04   1.79230455e-03
    1.27217470e-03   5.48112793e-04   6.74161959e-04   9.71213751e-05
    2.53064805e-04   3.95227788e-04   9.74674712e-05   3.31971213e-04
    1.32527587e-04   1.74194030e-04   3.08851972e-04   1.32220870e-04
    5.71805682e-04   3.85706110e-04   1.75133070e-04   4.39816070e-04
    5.87248676e-04]]
5100 iteration, weights = [[  1.58024357e-03   2.97316736e-02   8.02144970e
-04   1.80088021e-03
    1.27883439e-03   5.54091798e-04   6.80629559e-04   9.40719402e-05
    2.51450277e-04   3.99865136e-04   9.44277161e-05   3.36358632e-04
    1.32348618e-04   1.73583237e-04   3.11629451e-04   1.32030785e-04
    5.78534417e-04   3.85666399e-04   1.73678299e-04   4.42364458e-04
    5.84838006e-04]]
5200 iteration, weights = [[  1.58757265e-03   2.99340840e-02   8.05013155e
-04   1.80893331e-03
    1.28513413e-03   5.59975282e-04   6.86962599e-04   9.09269544e-05
    2.49707814e-04   4.04439797e-04   9.12924047e-05   3.40702943e-04
    1.32115455e-04   1.72893975e-04   3.14341727e-04   1.31786348e-04
    5.85172887e-04   3.85478096e-04   1.72128604e-04   4.44793062e-04
    5.82156058e-04]]
5300 iteration, weights = [[  1.59446408e-03   3.01294515e-02   8.07643421e
-04   1.81648222e-03
    1.29108658e-03   5.65766599e-04   6.93165797e-04   8.76897811e-05
    2.47841908e-04   4.08953968e-04   8.80648963e-05   3.45005664e-04
    1.31830004e-04   1.72129002e-04   3.16991085e-04   1.31489465e-04
    5.91724258e-04   3.85146418e-04   1.70487324e-04   4.47106084e-04
    5.79212358e-04]]
5400 iteration, weights = [[  1.60093325e-03   3.03180234e-02   8.10044122e
-04   1.82354465e-03
    1.29670394e-03   5.71468982e-04   6.99243705e-04   8.43636656e-05
    2.45856894e-04   4.13409768e-04   8.47484326e-05   3.49268258e-04
    1.31494104e-04   1.71290975e-04   3.19579733e-04   1.31141976e-04
    5.98191586e-04   3.84676400e-04   1.68757681e-04   4.49307580e-04
    5.76016099e-04]]
5500 iteration, weights = [[  1.60699497e-03   3.05000381e-02   8.12223319e
-04   1.83013769e-03
    1.30199797e-03   5.77085555e-04   7.05200716e-04   8.09517391e-05
    2.43756956e-04   4.17809243e-04   8.13461413e-05   3.53492139e-04
    1.31109529e-04   1.70382461e-04   3.22109798e-04   1.30745655e-04
    6.04577818e-04   3.84072902e-04   1.66942783e-04   4.51401464e-04
    5.72576149e-04]]
```

```
5600 iteration, weights = [[  1.61266355e-03   3.06757257e-02   8.14188790e
-04   1.83627782e-03
    1.30698003e-03   5.82619330e-04   7.11041068e-04   7.74570232e-05
    2.41546130e-04   4.22154365e-04   7.78610406e-05   3.57678669e-04
    1.30677990e-04   1.69405936e-04   3.24583334e-04   1.30302216e-04
    6.10885798e-04   3.83340611e-04   1.65045631e-04   4.53391510e-04
    5.68901069e-04]]
5700 iteration, weights = [[  1.61795280e-03   3.08453083e-02   8.15948041e
-04   1.84198095e-03
    1.31166108e-03   5.88073212e-04   7.16768850e-04   7.38824334e-05
    2.39228309e-04   4.26447038e-04   7.42960424e-05   3.61829163e-04
    1.30201139e-04   1.68363788e-04   3.27002322e-04   1.29813312e-04
    6.17118272e-04   3.82484050e-04   1.63069118e-04   4.55281364e-04
    5.64999115e-04]]
5800 iteration, weights = [[  1.62287605e-03   3.10090001e-02   8.17508313e
-04   1.84726245e-03
    1.31605171e-03   5.93450006e-04   7.22388009e-04   7.02307827e-05
    2.36807253e-04   4.30689100e-04   7.06539569e-05   3.65944891e-04
    1.29680570e-04   1.67258321e-04   3.29368676e-04   1.29280536e-04
    6.23277888e-04   3.81507584e-04   1.61016038e-04   4.57074540e-04
    5.60878257e-04]]
5900 iteration, weights = [[  1.62744614e-03   3.11670078e-02   8.18876594e
-04   1.85213711e-03
    1.32016210e-03   5.98752419e-04   7.27902352e-04   6.65047855e-05
    2.34286586e-04   4.34882322e-04   6.69374953e-05   3.70027077e-04
    1.29117819e-04   1.66091762e-04   3.31684239e-04   1.28705428e-04
    6.29367201e-04   3.80415425e-04   1.58889085e-04   4.58774431e-04
    5.56546182e-04]]
6000 iteration, weights = [[  1.63167548e-03   3.13195309e-02   8.20059627e
-04   1.85661925e-03
    1.32400210e-03   6.03983061e-04   7.33315552e-04   6.27070611e-05
    2.31669809e-04   4.39028416e-04   6.31492738e-05   3.74076902e-04
    1.28514372e-04   1.64866254e-04   3.33950792e-04   1.28089473e-04
    6.35388677e-04   3.79211637e-04   1.56690860e-04   4.60384309e-04
    5.52010310e-04]]
6100 iteration, weights = [[  1.63557604e-03   3.14667618e-02   8.21063917e
-04   1.86072264e-03
    1.32758122e-03   6.09144452e-04   7.38631156e-04   5.88401366e-05
    2.28960296e-04   4.43129031e-04   5.92918168e-05   3.78095508e-04
    1.27871660e-04   1.63583870e-04   3.36170054e-04   1.27434102e-04
    6.41344695e-04   3.77900141e-04   1.54423871e-04   4.61907334e-04
    5.47277800e-04]]
6200 iteration, weights = [[  1.63915936e-03   3.16088863e-02   8.21895742e
-04   1.86446060e-03
    1.33090862e-03   6.14239024e-04   7.43852584e-04   5.49064506e-05
    2.26161305e-04   4.47185760e-04   5.53675600e-05   3.82083993e-04
    1.27191064e-04   1.62246606e-04   3.38343680e-04   1.26740699e-04
```

```
    6.47237551e-04   3.76484720e-04   1.52090539e-04   4.63346553e-04
    5.42355558e-04]]
6300 iteration, weights = [[  1.64243659e-03   3.17460836e-02   8.22561158e
-04   1.86784596e-03
    1.33399315e-03   6.19269121e-04   7.48983135e-04   5.09083560e-05
    2.23275980e-04   4.51200140e-04   5.13788538e-05   3.86043419e-04
    1.26473917e-04   1.60856392e-04   3.40473272e-04   1.26010596e-04
    6.53069460e-04   3.74969025e-04   1.49693202e-04   4.64704905e-04
    5.37250250e-04]]
6400 iteration, weights = [[  1.64541848e-03   3.18785265e-02   8.23066012e
-04   1.87089112e-03
    1.33684334e-03   6.24237007e-04   7.54025995e-04   4.68481234e-05
    2.20307353e-04   4.55173654e-04   4.73279660e-05   3.89974811e-04
    1.25721506e-04   1.59415087e-04   3.42560372e-04   1.25245082e-04
    6.58842559e-04   3.73356578e-04   1.47234111e-04   4.65985230e-04
    5.31968306e-04]]
6500 iteration, weights = [[  1.64811540e-03   3.20063820e-02   8.23415942e
-04   1.87360804e-03
    1.33946743e-03   6.29144867e-04   7.58984237e-04   4.27279433e-05
    2.17258350e-04   4.59107731e-04   4.32170849e-05   3.93879157e-04
    1.24935071e-04   1.57924487e-04   3.44606469e-04   1.24445398e-04
    6.64558910e-04   3.71650780e-04   1.44715443e-04   4.67190264e-04
    5.26515932e-04]]
6600 iteration, weights = [[  1.65053736e-03   3.21298109e-02   8.23616390e
-04   1.87600825e-03
    1.34187335e-03   6.33994807e-04   7.63860826e-04   3.85499296e-05
    2.14131797e-04   4.63003753e-04   3.90483218e-05   3.97757410e-04
    1.24115809e-04   1.56386323e-04   3.46612999e-04   1.23612742e-04
    6.70220505e-04   3.69854909e-04   1.42139296e-04   4.68322649e-04
    5.20899116e-04]]
6700 iteration, weights = [[  1.65269402e-03   3.22489687e-02   8.23672607e
-04   1.87810288e-03
    1.34406879e-03   6.38788861e-04   7.68658623e-04   3.43161220e-05
    2.10930416e-04   4.66863052e-04   3.48237141e-05   4.01610491e-04
    1.23264878e-04   1.54802267e-04   3.48581350e-04   1.22748271e-04
    6.75829262e-04   3.67972130e-04   1.39507695e-04   4.69384935e-04
    5.15123635e-04]]
6800 iteration, weights = [[  1.65459471e-03   3.23640053e-02   8.23589658e
-04   1.87990267e-03
    1.34606113e-03   6.43528991e-04   7.73380388e-04   3.00284885e-05
    2.07656838e-04   4.70686912e-04   3.05452278e-05   4.05439287e-04
    1.22383390e-04   1.53173931e-04   3.50512860e-04   1.21853099e-04
    6.81387035e-04   3.66005497e-04   1.36822593e-04   4.70379580e-04
    5.09195063e-04]]
6900 iteration, weights = [[  1.65624840e-03   3.24750652e-02   8.23372434e
-04   1.88141799e-03
    1.34785752e-03   6.48217091e-04   7.78028786e-04   2.56889283e-05
```

```
    2.04313598e-04   4.74476573e-04   2.62147595e-05   4.09244655e-04
    1.21472423e-04   1.51502871e-04   3.52408818e-04   1.20928304e-04
    6.86895612e-04   3.63957956e-04   1.34085877e-04   4.71308958e-04
    5.03118780e-04]]
7000 iteration, weights = [[  1.65766379e-03   3.25822879e-02   8.23025650e
-04   1.88265883e-03
    1.34946484e-03   6.52854986e-04   7.82606387e-04   2.12992736e-05
    2.00903145e-04   4.78233233e-04   2.18341397e-05   4.13027421e-04
    1.20533014e-04   1.49790589e-04   3.54270470e-04   1.19974925e-04
    6.92356717e-04   3.61832351e-04   1.31299365e-04   4.72175359e-04
    4.96899976e-04]]
7100 iteration, weights = [[  1.65884924e-03   3.26858083e-02   8.22553859e
-04   1.88363485e-03
    1.35088974e-03   6.57444440e-04   7.87115671e-04   1.68612925e-05
    1.97427841e-04   4.81958046e-04   1.74051345e-05   4.16788384e-04
    1.19566165e-04   1.48038533e-04   3.56099018e-04   1.18993963e-04
    6.97772015e-04   3.59631424e-04   1.28464812e-04   4.72980993e-04
    4.90543659e-04]]
7200 iteration, weights = [[  1.65981284e-03   3.27857561e-02   8.21961452e
-04   1.88435535e-03
    1.35213863e-03   6.61987153e-04   7.91559030e-04   1.23766910e-05
    1.93889965e-04   4.85652126e-04   1.29294478e-05   4.20528313e-04
    1.18572842e-04   1.46248101e-04   3.57895623e-04   1.17986386e-04
    7.03143112e-04   3.57357823e-04   1.25583913e-04   4.73727991e-04
    4.84054662e-04]]
7300 iteration, weights = [[  1.66056237e-03   3.28822567e-02   8.21252667e
-04   1.88482932e-03
    1.35321770e-03   6.66484767e-04   7.95938775e-04   7.84711532e-06
    1.90291714e-04   4.89316548e-04   8.40872399e-06   4.24247951e-04
    1.17553977e-04   1.44420643e-04   3.59661402e-04   1.16953126e-04
    7.08471558e-04   3.55014101e-04   1.22658301e-04   4.74418411e-04
    4.77437647e-04]]
7400 iteration, weights = [[  1.66110536e-03   3.29754312e-02   8.20431592e
-04   1.88506544e-03
    1.35413293e-03   6.70938864e-04   8.00257132e-04   3.27415381e-06
    1.86635213e-04   4.92952350e-04   3.84454967e-06   4.27948013e-04
    1.16510472e-04   1.42557459e-04   3.61397435e-04   1.15895085e-04
    7.13758848e-04   3.52602723e-04   1.19689554e-04   4.75054238e-04
    4.70697114e-04]]
7500 iteration, weights = [[  1.66144907e-03   3.30653962e-02   8.19502174e
-04   1.88507205e-03
    1.35489006e-03   6.75350974e-04   8.04516252e-04  -1.34066090e-06
    1.82922507e-04   4.96560532e-04  -7.61544269e-07   4.31629192e-04
    1.15443193e-04   1.40659805e-04   3.63104764e-04   1.14813131e-04
    7.19006425e-04   3.50126067e-04   1.16679193e-04   4.75637387e-04
    4.63837403e-04]]
```

```
7600 iteration, weights = [[  1.66160050e-03   3.31522644e-02   8.18468220e
-04   1.88485723e-03
    1.35549467e-03   6.79722570e-04   8.08718210e-04  -5.99584996e-06
    1.79155572e-04   5.00142059e-04  -5.40808065e-06   4.35292154e-04
    1.14352979e-04   1.38728893e-04   3.64784395e-04   1.13708102e-04
    7.24215681e-04   3.47586429e-04   1.13628685e-04   4.76169707e-04
    4.56862705e-04]]
7700 iteration, weights = [[  1.66156641e-03   3.32361444e-02   8.17333402e
-04   1.88442878e-03
    1.35595211e-03   6.84055075e-04   8.12865007e-04  -1.06899864e-05
    1.75336314e-04   5.03697865e-04  -1.00936341e-05   4.38937544e-04
    1.13240639e-04   1.36765893e-04   3.66437297e-04   1.12580807e-04
    7.29387960e-04   3.44986020e-04   1.10539449e-04   4.76652980e-04
    4.49777060e-04]]
7800 iteration, weights = [[  1.66135332e-03   3.33171411e-02   8.16101265e
-04   1.88379420e-03
    1.35626756e-03   6.88349861e-04   8.16958574e-04  -1.54216934e-05
    1.71466574e-04   5.07228847e-04  -1.48168294e-05   4.42565982e-04
    1.12106951e-04   1.34771932e-04   3.68064406e-04   1.11432027e-04
    7.34524558e-04   3.42326979e-04   1.07412850e-04   4.77088928e-04
    4.42584369e-04]]
7900 iteration, weights = [[  1.66096751e-03   3.33953557e-02   8.14775230e
-04   1.88296075e-03
    1.35644602e-03   6.92608253e-04   8.21000775e-04  -2.01896423e-05
    1.67548125e-04   5.10735874e-04  -1.95763394e-05   4.46178068e-04
    1.10952669e-04   1.32748098e-04   3.69666626e-04   1.10262515e-04
    7.39626725e-04   3.39611366e-04   1.04250208e-04   4.77479211e-04
    4.35288396e-04]]
8000 iteration, weights = [[  1.66041507e-03   3.34708858e-02   8.13358596e
-04   1.88193541e-03
    1.35649230e-03   6.96831529e-04   8.24993408e-04  -2.49925512e-05
    1.63582679e-04   5.14219783e-04  -2.43708836e-05   4.49774380e-04
    1.09778519e-04   1.30695444e-04   3.71244826e-04   1.09072997e-04
    7.44695669e-04   3.36841170e-04   1.01052796e-04   4.77825430e-04
    4.27892770e-04]]
8100 iteration, weights = [[  1.65970184e-03   3.35438256e-02   8.11854548e
-04   1.88072495e-03
    1.35641105e-03   7.01020922e-04   8.28938209e-04  -2.98291832e-05
    1.59571891e-04   5.17681382e-04  -2.91992265e-05   4.53355476e-04
    1.08585201e-04   1.28614981e-04   3.72799850e-04   1.07864175e-04
    7.49732554e-04   3.34018311e-04   9.78218409e-05   4.78129133e-04
    4.20400997e-04]]
8200 iteration, weights = [[  1.65883348e-03   3.36142660e-02   8.10266159e
-04   1.87933588e-03
    1.35620676e-03   7.05177622e-04   8.32836851e-04  -3.46983447e-05
    1.55517352e-04   5.21121449e-04  -3.40601759e-05   4.56921895e-04
    1.07373393e-04   1.26507688e-04   3.74332506e-04   1.06636726e-04
```

```
    7.54738503e-04   3.31144640e-04   9.45585282e-05   4.78391809e-04
    4.12816456e-04]]
8300 iteration, weights = [[  1.65781545e-03   3.36822947e-02   8.08596392e
-04   1.87777447e-03
    1.35588376e-03   7.09302778e-04   8.36690950e-04  -3.95988841e-05
    1.51420603e-04   5.24540737e-04  -3.89525814e-05   4.60474155e-04
    1.06143745e-04   1.24374509e-04   3.75843580e-04   1.05391302e-04
    7.59714601e-04   3.28221945e-04   9.12640013e-05   4.78614899e-04
    4.05142411e-04]]
8400 iteration, weights = [[  1.65665300e-03   3.37479964e-02   8.06848109e
-04   1.87604680e-03
    1.35544623e-03   7.13397496e-04   8.40502064e-04  -4.45296902e-05
    1.47283128e-04   5.27939972e-04  -4.38753331e-05   4.64012759e-04
    1.04896889e-04   1.22216353e-04   3.77333825e-04   1.04128533e-04
    7.64661894e-04   3.25251949e-04   8.79393633e-05   4.78799792e-04
    3.97382008e-04]]
8500 iteration, weights = [[  1.65535121e-03   3.38114527e-02   8.05024070e
-04   1.87415872e-03
    1.35489820e-03   7.17462846e-04   8.44271699e-04  -4.94896906e-05
    1.43106358e-04   5.31319855e-04  -4.88273599e-05   4.67538191e-04
    1.03633430e-04   1.20034100e-04   3.78803970e-04   1.02849026e-04
    7.69581391e-04   3.22236316e-04   8.45856787e-05   4.78947826e-04
    3.89538284e-04]]
8600 iteration, weights = [[  1.65391499e-03   3.38727424e-02   8.03126939e
-04   1.87211587e-03
    1.35424355e-03   7.21499859e-04   8.48001304e-04  -5.44778509e-05
    1.38891676e-04   5.34681059e-04  -5.38076284e-05   4.71050915e-04
    1.02353956e-04   1.17828598e-04   3.80254720e-04   1.01553369e-04
    7.74474066e-04   3.19176651e-04   8.12039746e-05   4.79060297e-04
    3.81614172e-04]]
8700 iteration, weights = [[  1.65234906e-03   3.39319416e-02   8.01159285e
-04   1.86992370e-03
    1.35348603e-03   7.25509531e-04   8.51692281e-04  -5.94931726e-05
    1.34640416e-04   5.38024238e-04  -5.88151414e-05   4.74551384e-04
    1.01059032e-04   1.15600663e-04   3.81686751e-04   1.00242128e-04
    7.79340860e-04   3.16074504e-04   7.77952422e-05   4.79138450e-04
    3.73612497e-04]]
8800 iteration, weights = [[  1.65065797e-03   3.39891235e-02   7.99123587e
-04   1.86758747e-03
    1.35262928e-03   7.29492822e-04   8.55345980e-04  -6.45346926e-05
    1.30353865e-04   5.41350021e-04  -6.38489367e-05   4.78040032e-04
    9.97492031e-05   1.13351087e-04   3.83100720e-04   9.89158477e-05
    7.84182679e-04   3.12931369e-04   7.43604379e-05   4.79183489e-04
    3.65535990e-04]]
8900 iteration, weights = [[  1.64884615e-03   3.40443590e-02   7.97022240e
-04   1.86511226e-03
    1.35167678e-03   7.33450659e-04   8.58963706e-04  -6.96014813e-05
```

```
    1.26033263e-04   5.44659015e-04  -6.89080860e-05   4.81517278e-04
    9.84249964e-05   1.11080629e-04   3.84497258e-04   9.75750559e-05
    7.89000397e-04   3.09748687e-04   7.09004842e-05   4.79196577e-04
    3.57387281e-04]]
9000 iteration, weights = [[  1.64691782e-03   3.40977164e-02   7.94857552e
-04   1.86250294e-03
    1.35063191e-03   7.37383935e-04   8.62546716e-04  -7.46926419e-05
    1.21679809e-04   5.47951804e-04  -7.39916933e-05   4.84983526e-04
    9.70869200e-05   1.08790026e-04   3.85876974e-04   9.62202607e-05
    7.93794860e-04   3.06527851e-04   6.74162716e-05   4.79178835e-04
    3.49168913e-04]]
9100 iteration, weights = [[  1.64487709e-03   3.41492616e-02   7.92631749e
-04   1.85976426e-03
    1.34949791e-03   7.41293513e-04   8.66096225e-04  -7.98073091e-05
    1.17294660e-04   5.51228955e-04  -7.90988943e-05   4.88439167e-04
    9.57354639e-05   1.06479986e-04   3.87240458e-04   9.48519527e-05
    7.98566882e-04   3.03270204e-04   6.39086593e-05   4.79131343e-04
    3.40883336e-04]]
9200 iteration, weights = [[  1.64272791e-03   3.41990582e-02   7.90346983e
-04   1.85690076e-03
    1.34827793e-03   7.45180226e-04   8.69613403e-04  -8.49446478e-05
    1.12878929e-04   5.54491013e-04  -8.42288549e-05   4.91884578e-04
    9.43711013e-05   1.04151192e-04   3.88588275e-04   9.34706053e-05
    8.03317249e-04   2.99977040e-04   6.03784762e-05   4.79055145e-04
    3.32532916e-04]]
9300 iteration, weights = [[  1.64047410e-03   3.42471675e-02   7.88005327e
-04   1.85391684e-03
    1.34697499e-03   7.49044877e-04   8.73099381e-04  -9.01038523e-05
    1.08433695e-04   5.57738504e-04  -8.93807703e-05   4.95320122e-04
    9.29942886e-05   1.01804307e-04   3.89920975e-04   9.20766752e-05
    8.08046720e-04   2.96649610e-04   5.68265222e-05   4.78951248e-04
    3.24119934e-04]]
9400 iteration, weights = [[  1.63811934e-03   3.42936488e-02   7.85608782e
-04   1.85081675e-03
    1.34559202e-03   7.52888241e-04   8.76555249e-04  -9.52841453e-05
    1.03959995e-04   5.60971935e-04  -9.45538640e-05   4.98746150e-04
    9.16054662e-05   9.94399656e-05   3.91239085e-04   9.06706032e-05
    8.12756027e-04   2.93289120e-04   5.32535691e-05   4.78820624e-04
    3.15646592e-04]]
9500 iteration, weights = [[  1.63566718e-03   3.43385592e-02   7.83159279e
-04   1.84760458e-03
    1.34413184e-03   7.56711066e-04   8.79982060e-04  -1.00484777e-04
    9.94588316e-05   5.64191796e-04  -9.97473867e-05   5.02163000e-04
    9.02050590e-05   9.70587835e-05   3.92543115e-04   8.92528143e-05
    8.17445875e-04   2.89896735e-04   4.96603616e-05   4.78664210e-04
    3.07115017e-04]]
```

```
9600 iteration, weights = [[  1.63312105e-03   3.43819537e-02   7.80658683e
-04   1.84428427e-03
    1.34259717e-03   7.60514073e-04   8.83380829e-04  -1.05705022e-04
    9.49311713e-05   5.67398561e-04  -1.04960615e-04   5.05570998e-04
    8.87934771e-05   9.46613536e-05   3.93833558e-04   8.78237191e-05
    8.22116946e-04   2.86473576e-04   4.60476184e-05   4.78482911e-04
    2.98527257e-04]]
9700 iteration, weights = [[  1.63048426e-03   3.44238857e-02   7.78108790e
-04   1.84085965e-03
    1.34099062e-03   7.64297960e-04   8.86752535e-04  -1.10944184e-04
    9.03779474e-05   5.70592685e-04  -1.10192852e-04   5.08970461e-04
    8.73711162e-05   9.22482482e-05   3.95110887e-04   8.63837133e-05
    8.26769898e-04   2.83020728e-04   4.24160328e-05   4.78277600e-04
    2.89885291e-04]]
9800 iteration, weights = [[  1.62775999e-03   3.44644064e-02   7.75511336e
-04   1.83733438e-03
    1.33931475e-03   7.68063396e-04   8.90098125e-04  -1.16201588e-04
    8.58000600e-05   5.73774608e-04  -1.15443425e-04   5.12361691e-04
    8.59383580e-05   8.98200194e-05   3.96375562e-04   8.49331790e-05
    8.31405365e-04   2.79539237e-04   3.87662738e-05   4.78049118e-04
    2.81191029e-04]]
9900 iteration, weights = [[  1.62495134e-03   3.45035653e-02   7.72867994e
-04   1.83371203e-03
    1.33757199e-03   7.71811032e-04   8.93418509e-04  -1.21476584e-04
    8.11983779e-05   5.76944755e-04  -1.20711682e-04   5.15744982e-04
    8.44955709e-05   8.73772000e-05   3.97628024e-04   8.34724849e-05
    8.36023959e-04   2.76030111e-04   3.50989870e-05   4.77798279e-04
    2.72446312e-04]]
```