Submitted by: Ashish Jain

HomeWork 2A

Question 1.

Part 1.1)

POS	E	T	A	I	N	O	S	H	R	D
1	-7.6443	18.468	-6.3285	10.4224	-4.9671	-1.934	-0.9451	-5671	5.395	-6.8098
2	-4.0744	5.7448	1.1763	-1.7931	-1.2122	-1.7848	-8.2998	3.0951	6.8065	0.34161
3	-10.208	0.8973	17.1910	-12.017	5.5793	-0.594	-21.426	9.1489	9.4824	1.9471
4	6.4648	24.531	-13.342	5.8712	-10.954	-11.496	-5.4946	-7.1956	8.0456	3.5714

Part 1.2)

Words	-Ew
THAT	63.9793
HIRE	89.6109
RISES	96.9406

Part 1.3)

Words	log Z (log partition function)
THAT	67.6018
HIRE	89.6144
RISES	103.5275

Part 1.4)

Test Image	Likely Labeling	Probability
THAT	TRAT	0.7958
HIRE	HIRE	0.9965
RISES	RISER	0.937

Part 1.5) First test word: TEST

	Position 1	**Position 2**	**Position 3**	**Position 4**
E	7.2226e-12	1.2658e-05	1.1321e-12	8.8682e-09
T	0.9995	0.1724	2.2945e-08	0.9999
A	2.6261e-11	0.0027	0.9994	2.1356e-17
I	0.0004	0.0001	1.6118e-13	7.4054e-09
N	7.1555e-11	0.0002	3.6975e-06	3.29e-16
O	2.1138e-09	0.0001	1.761e-08	1.441e-16
S	3.2959e-09	1.0646e-07	5.1721e-18	5.371e-14
H	4.3492e-11	0.0267	0.0002	1.3178e-14
R	2.628e-06	0.7965	0.0002	6.3939e-08
D	1.0693e-11	0.0009	9.4637e-08	6.3736e-10

Question 2)

Part 2.1)

Clique 1:

	T	A	H
T	17.814	18.749	18.8338
A	-6.0478	-6.5592	-6.2812
H	-5.2916	-5.6097	-5.7932

Clique 2:

	T	A	H
T	5.091	6.0255	6.1103
A	1.457	0.9456	1.2237
H	3.4606	3.1425	2.959

Clique 3:

	T	A	H
T	24.7748	-12.1648	-5.9327
A	42.0029	3.6173	10.0427
H	34.0456	-4.1466	1.8171

Part 2.2)

Messages	E	T	A	I	N	O	S	H	R	D
$\delta_{1\to2}(\mathrm{Y2})$	18.589	17.815	18.749	18.522	18.180	18.677	18.091	18.834	18.363	18.216
$\delta_{2\to3}(\mathrm{Y2})$	25.651	25.236	25.598	25.577	25.271	25.601	25.071	25.388	25.414	25.202
$\delta_{2\to1}(Y3)$	37.735	48.029	42.949	40.43	40.907	40.051	33.455	45.146	49.011	42.411
$\delta_{3\to2}(Y3)$	14.443	24.774	42.002	12.567	29.822	24.145	2.727	34.045	33.908	26.226

Part 2.3)

$\boldsymbol{\beta}(\boldsymbol{Y1}, \boldsymbol{Y2})$

	T	**A**
T	65.843	61.698
A	41.981	36.39

$\boldsymbol{\beta}(\boldsymbol{Y2}, \boldsymbol{Y3})$

	T	**A**
T	47.681	65.843
A	44.981	61.697

$\boldsymbol{\beta}(\boldsymbol{Y3}, \boldsymbol{Y4})$

	T	**A**
T	50.011	13.071
A	67.601	29.215

Part 2.4)

Pairwise Marginals for labels "t,a,h"

Clique 1:

	T	A	H
T	0.1723	0.0027	0.0267
A	7.4658e-12	2.7859e-14	3.3086e-13
H	1.5904e-11	7.20009e-14	5.3896e-13

Clique 2:

	T	A	H
T	2.2314e-09	0.1723	6.56861e-05
A	1.49969e-10	0.00272	1.2615e-06
H	1.2104e-09	0.0267	7.7862

Clique 3:

	T	A	H
T	2.2945e-08	2.0795e-24	1.0580e-21
A	0.9994	2.1336e-17	1.3170e-14
H	0.00028	7.3431e-21	2.857e-18

Marginal distribution:

	Position 1	**Position 2**	**Position 3**	**Position 4**
E	7.222e-12	1.2658	1.1321e-12	8.8682e-09
T	0.9995	0.1724	2.2945e-08	0.9999
A	2.6261e-11	0.0027	0.9994	2.1356e-17
I	0.00047	0.00017	1.6118e-13	7.4054e-09
N	7.15554e-11	0.0002	3.6975e-06	3.29e-16
O	2.1138e-09	0.00014	1.761e-08	1.441e-16
S	3.2959e-09	1.0646e-07	5.1721e-18	5.371e-14
H	4.3492e-11	0.02673	0.00028	1.3178e-14
R	2.628e-06	0.7965	0.00025	6.3939e-08
D	1.0693e-11	0.000936	9.4637e-08	6.3736e-10

Part 2.5)

Predictions for first five test sequences:

Test Word	**Predicted Word**
THAT	TRAT
HIRE	HIRE
RISES	RISER
EDISON	EDISON
SHORE	SHORE

Accuracy: Total number of correct characters/total number of characters **= 89.916**

Question 4)

4.1 Since we need to maximize f(x,y), we multiply it by negative sign and take its derivative. Then we can use numerical optimizer from python library to minimize -f(x,y).
Minimizing –f(x,y) will give us maximum for f(x,y). Therefore, we find derivatives for –f(x,y) as shown below.

$$f_w(x, y) = -(1-x)^2 - 100(y-x^2)^2$$
$$-f_w(x, y) = (1-x)^2 + 100(y-x^2)^2$$
$$\frac{\partial(-f(x, y))}{\partial(x)} = 2(x-1) + 400 * x * (x^2 - y)$$
$$\frac{\partial(-f(x, y))}{\partial(y)} = 200 * (y - x^2)$$

4.2 I used bfgs_min numerical optimizer from python scipy package. This optimizer minimizes the input function and returns co-ordinates at which minima is achieved.

Code:

```
from numpy import *
from scipy.optimize import fmin_bfgs

def objective_function(x):

    val = (1-x[0])**2 + 100*(x[1]-x[0]**2)**2
    return val

def gradient_function(x):

   derivative = zeros_like(x)
   derivative[0] = 2*(x[0]-1)+400*(x[0])*(x[0]**2 - x[1])
   derivative[1] = 200*(x[1]-x[0]**2)
   return derivative

def main():

    x0 = [1.3, 0.7]
    xopt = fmin_bfgs(objective_function, x0, fprime=gradient_function, disp=True,
retall=True)
    print xopt
```

```
if __name__ == "__main__":
    main()
```

Output:

Current function value: 0.000000
    Iterations: 40
    Function evaluations: 57
    Gradient evaluations: 57

Location of maximum value = (x, y) = (1.0, 1.0)
Value of objective function at the maximum = 0.0