Mark Gluzman
Mg3627
HW5
ELEN 4903
Due 5/2

Problem 1 (Markov chains) – 50 points

In this problem, you will rank 759 college football teams based only on the scores of every game in the 2015 season. The data provided contains the result of one game on each line. For the ith line, the information contained in "scores" is

```
scores(i,1) = Team1 index,   scores(i,2) = Team1 points,
scores(i,3) = Team2 index,   scores(i,4) = Team2 points.
```

If `scores(i,2) > scores(i,4)` then Team1 wins and Team2 loses, and vice versa. The index of a team refers to the row of "legend" where that team's name can be found.

Construct a 759×759 random walk matrix M on the college football teams. First construct the unnormalized matrix $\widehat{M}$, initially set to all zeros. For a particular game i, let j_1 be the index of Team1 and j_2 the index of Team2. Then update

$$\widehat{M}_{j_1 j_1} \leftarrow \widehat{M}_{j_1 j_1} + \mathbb{1}\{\texttt{Team1 wins}\} + \frac{\texttt{points}_{j_1}}{\texttt{points}_{j_1}+\texttt{points}_{j_2}},$$

$$\widehat{M}_{j_2 j_2} \leftarrow \widehat{M}_{j_2 j_2} + \mathbb{1}\{\texttt{Team2 wins}\} + \frac{\texttt{points}_{j_2}}{\texttt{points}_{j_1}+\texttt{points}_{j_2}},$$

$$\widehat{M}_{j_1 j_2} \leftarrow \widehat{M}_{j_1 j_2} + \mathbb{1}\{\texttt{Team2 wins}\} + \frac{\texttt{points}_{j_2}}{\texttt{points}_{j_1}+\texttt{points}_{j_2}},$$

$$\widehat{M}_{j_2 j_1} \leftarrow \widehat{M}_{j_2 j_1} + \mathbb{1}\{\texttt{Team1 wins}\} + \frac{\texttt{points}_{j_1}}{\texttt{points}_{j_1}+\texttt{points}_{j_2}}.$$

After processing all games, let M be the matrix formed by normalizing the rows of $\widehat{M}$ so they sum to 1.

- Let w_t be the 1×759 state vector at step t. Set w_0 to the uniform distribution. Therefore, w_t is the distribution on the state after t steps given that the starting state at time 0 is uniformly distributed.
- Use w_t to rank the teams by sorting in decreasing value according to this vector. List the top 25 teams and their corresponding values in w_t for $t = 10, 100, 1000, 2500$.

Solution
Teams:

10	100	1000	2500
435	435	6	6
440	6	133	133
608	440	409	409
6	608	489	489
363	133	492	492
673	409	611	611
343	489	390	390

133	363	274	274
598	492	389	389
719	611	624	292
465	390	292	624
696	274	215	215
668	598	321	321
20	343	29	29
736	389	482	482
432	292	440	219
489	624	219	629
370	202	629	53
571	215	435	679
564	715	53	442
715	321	679	494
144	29	442	476
643	482	494	458
492	20	476	751
302	219	458	498

Values:

10	100	1000	2500
0.017087	0.057013	0.060819	0.066456
0.011808	0.032894	0.031824	0.034762
0.011683	0.028864	0.02534	0.027684
0.011224	0.023794	0.0242	0.026442
0.009443	0.017333	0.020832	0.022725
0.008525	0.013734	0.019563	0.021354
0.008059	0.013147	0.019307	0.021086
0.008032	0.011531	0.018964	0.020723
0.007494	0.010981	0.015605	0.017045
0.00728	0.010776	0.014585	0.015899
0.006902	0.010555	0.014581	0.015899
0.006711	0.010294	0.014409	0.015744
0.0067	0.008544	0.014228	0.015542
0.006612	0.008514	0.013698	0.01496
0.006542	0.008432	0.013498	0.014743
0.006342	0.008155	0.012785	0.013959
0.006247	0.008147	0.012677	0.013761
0.006187	0.007795	0.012601	0.012163
0.005946	0.007727	0.012203	0.011985
0.005782	0.007674	0.011162	0.011717
0.005744	0.007559	0.010982	0.011676

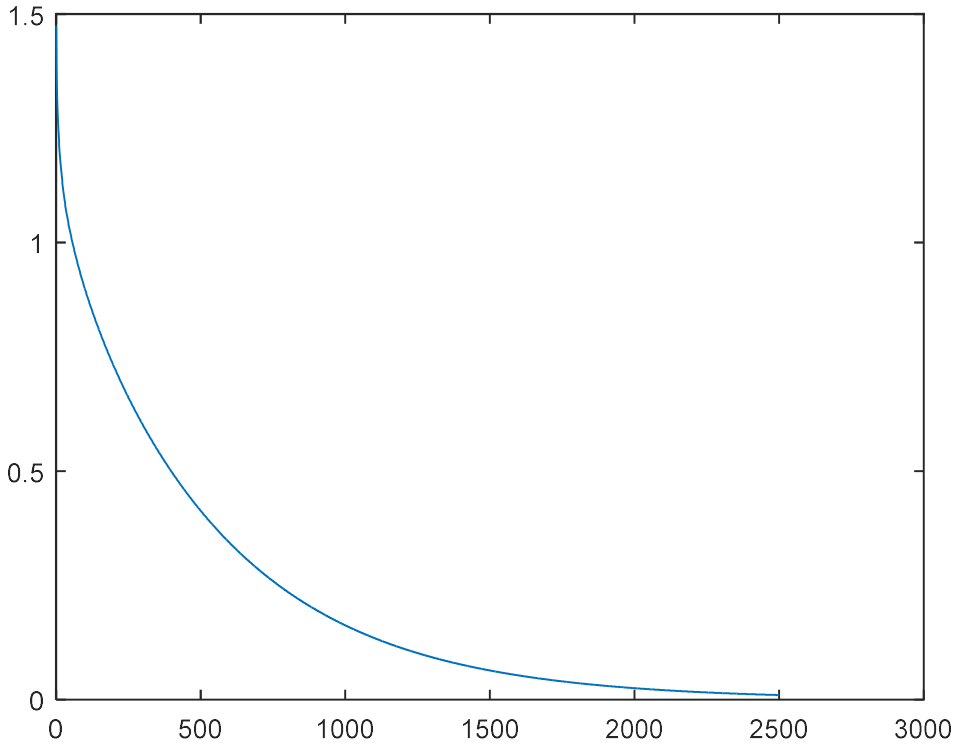
0.005633	0.007456	0.010733	0.011632
0.005607	0.007381	0.010715	0.011009
0.005287	0.00708	0.010657	0.010697
0.005173	0.007013	0.01009	0.010614

- We saw that w_∞ is related to the first eigenvector of M^T. That is, we can find w_∞ by getting the first eigenvector and eigenvalue of M^T and post-processing:

$$M^T u_1 = \lambda_1 u_1, \quad w_\infty = u_1^T / \left[\sum_j u_1(j)\right]$$

This is because $u_1^T u_1 = 1$ by convention. Also, we observe that $\lambda_1 = 1$ for this specific matrix. Plot $\|w_t - w_\infty\|_1$ as a function of t for $t = 1, \ldots, 2500$. What is the value of $\|w_{2500} - w_\infty\|_1$?

Solution



$||w_{2500}-w_\infty|| = 0.0101$

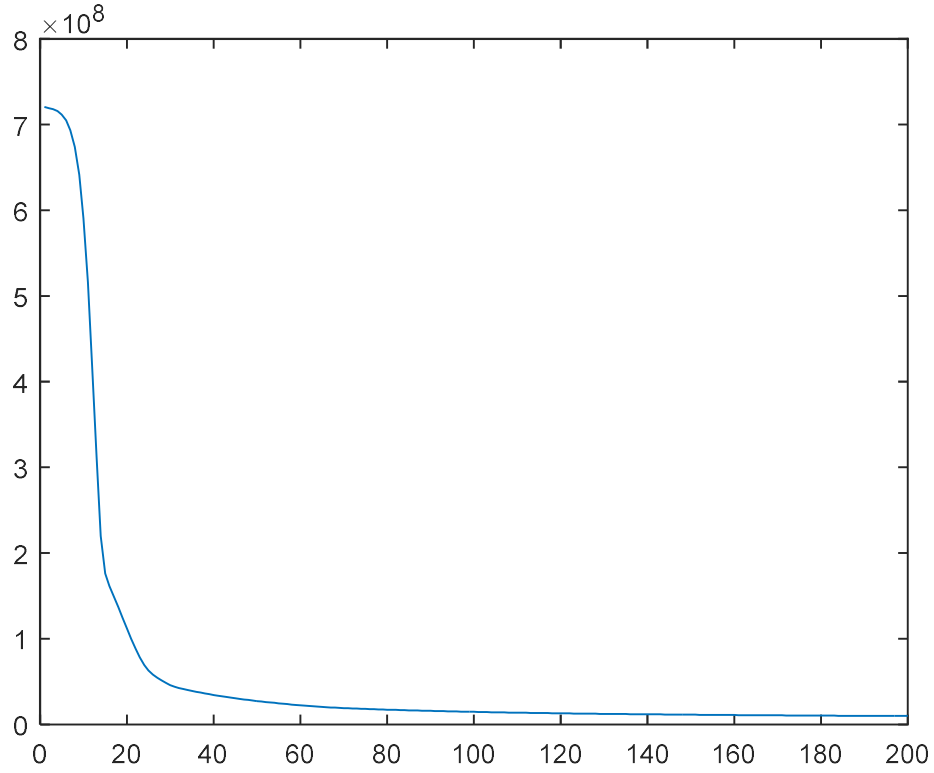
Problem 2 (Nonnegative matrix factorization) – 70 points

In this problem you will factorize a $n \times m$ matrix X into a rank-K approximation WH, where W is $n \times K$, H is $K \times m$ and all values in the matrices are nonnegative. Each value in W and H can be initialized randomly, e.g., from a Uniform(0,1) distribution. (See a hint about the implementation below.)

Part 1: The data to be used for Part 1 consists of 1000 images of faces, each originally 32×32, but vectorized to length 1024. The data matrix is therefore 1024×1000.

- Implement and run the NMF algorithm on this data using the *Euclidean penalty*. Set the rank of the factorization to 25 and run for 200 iterations.
- Plot the objective as a function of iteration.

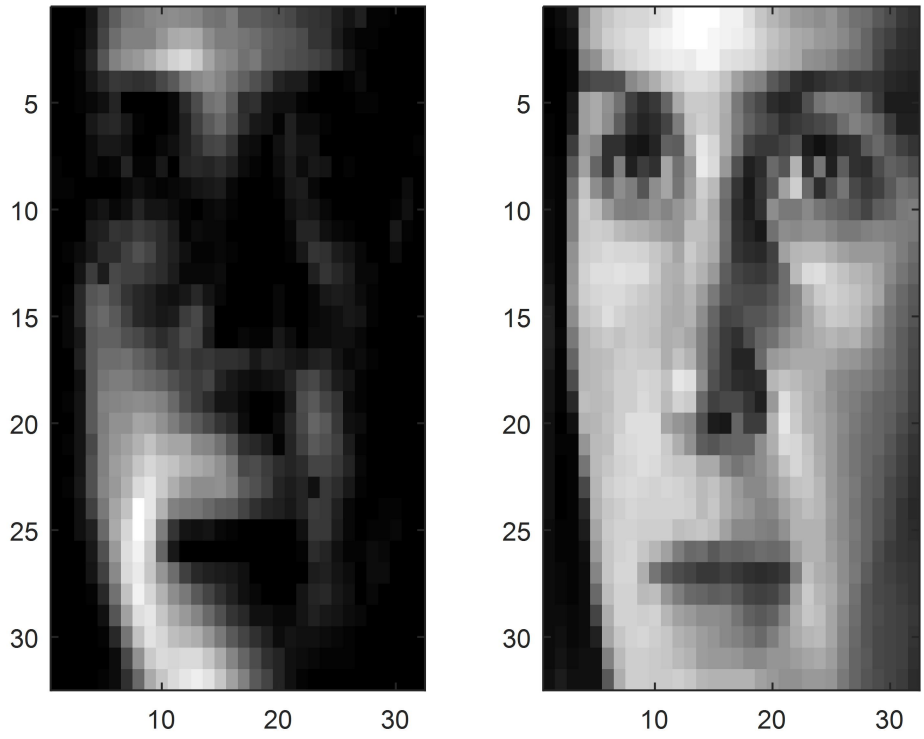
Solution

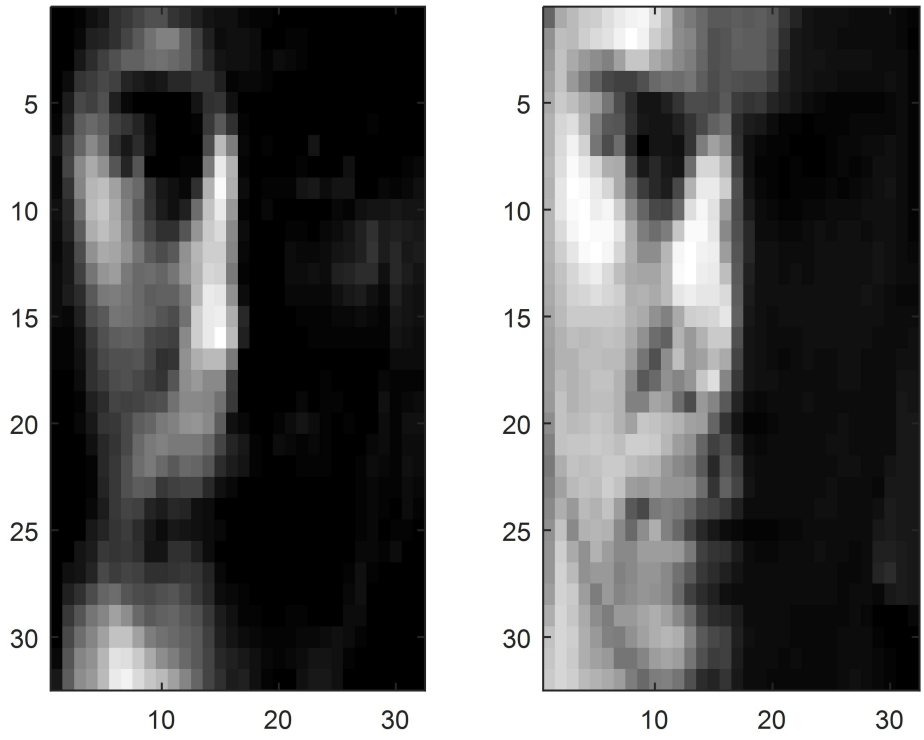


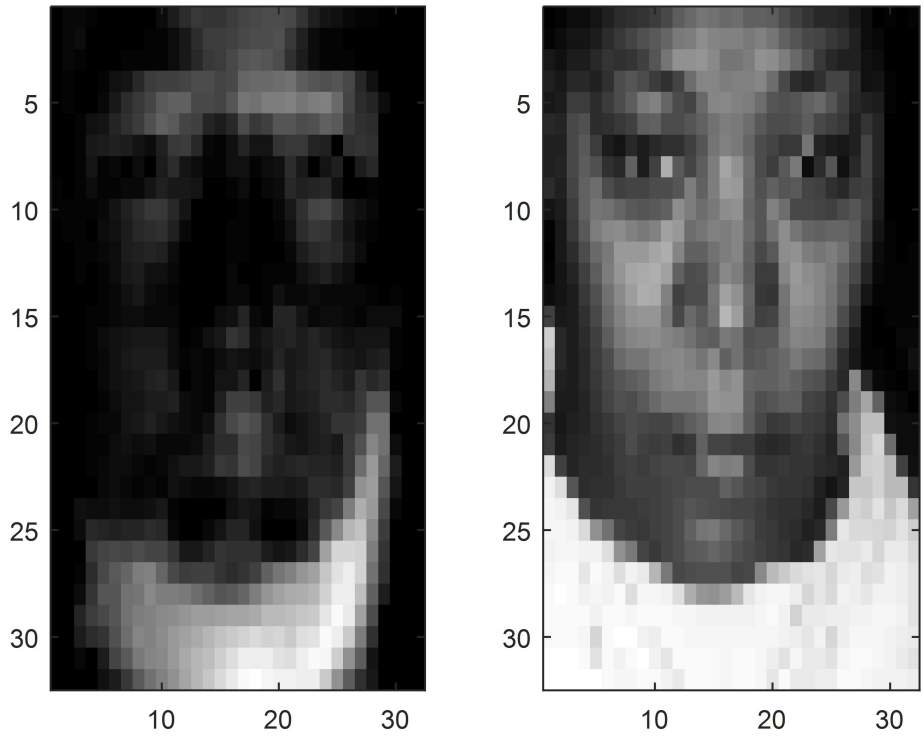
- Pick 10 columns from W and show them as 32×32 images. For each vector you select from W, find the column of H that places the highest weight on this vector and show the corresponding column of X as a 32×32 image.

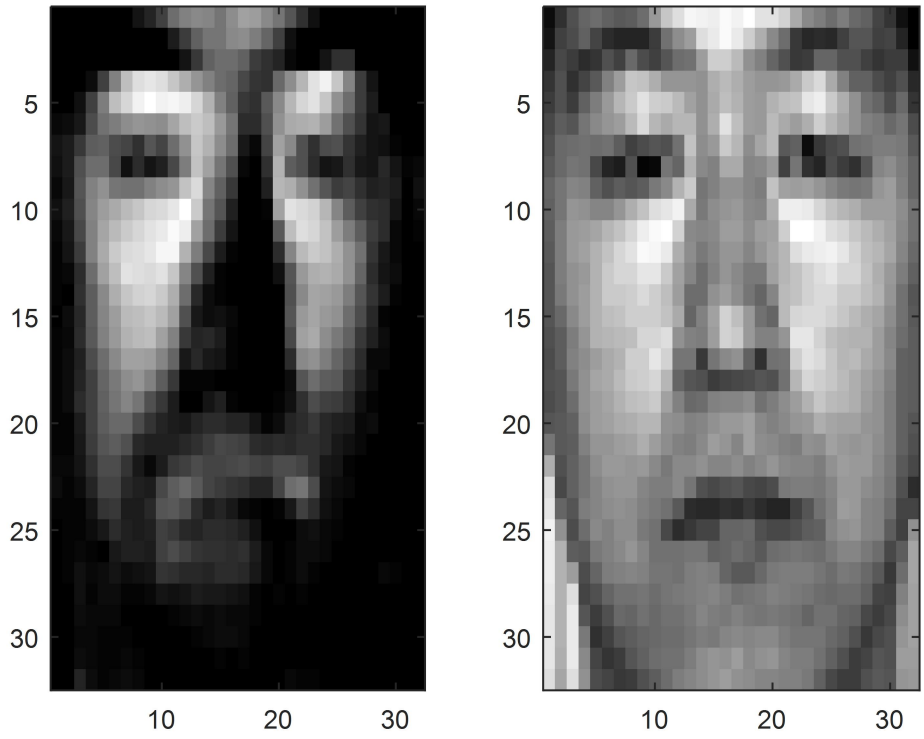
Solution

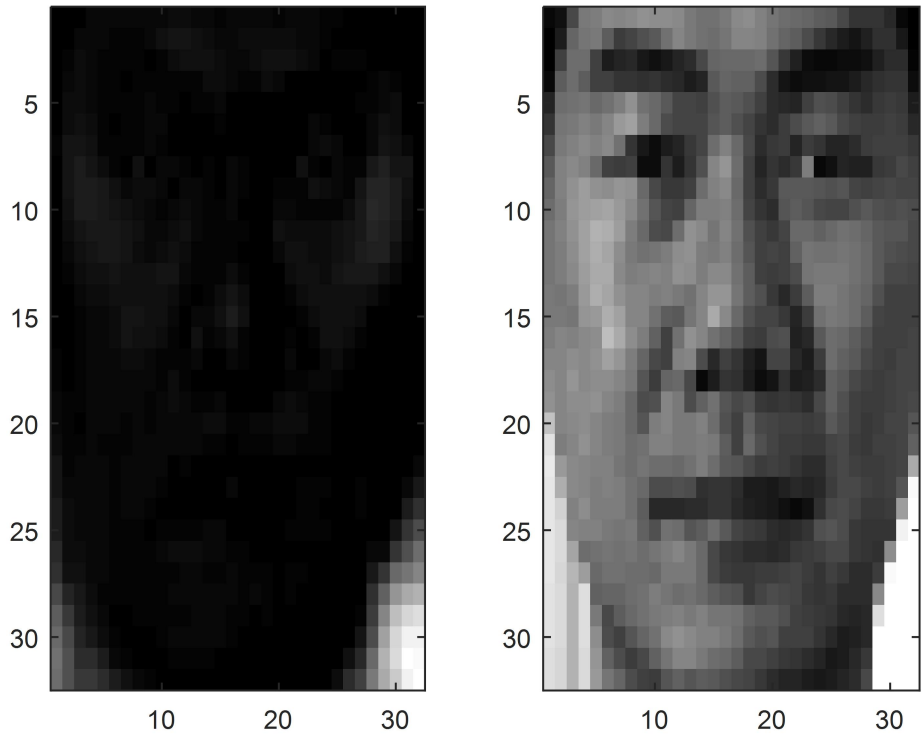
The left side is an image formed from W, the right side – from X

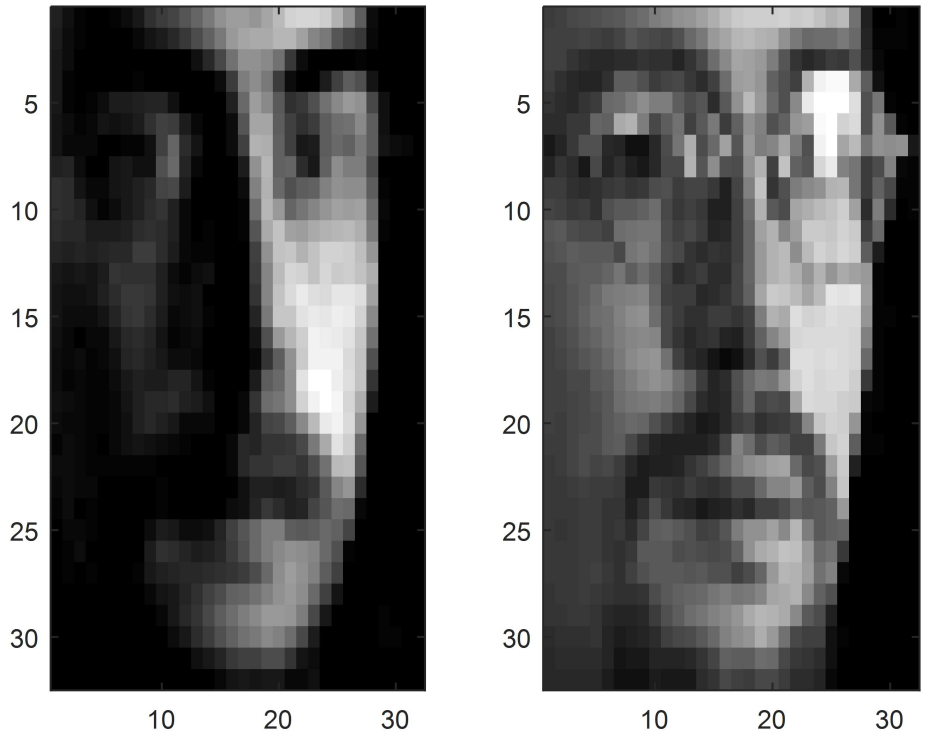
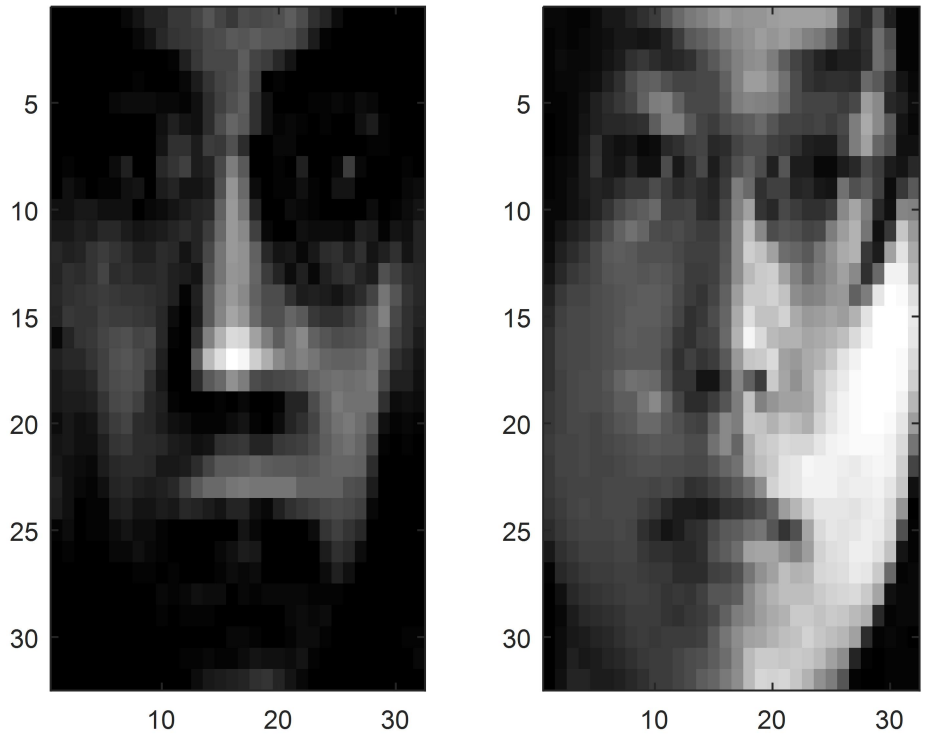

5
10
15
20
25
30
10
20
30
5
10
15
20
25
30
10
20
30

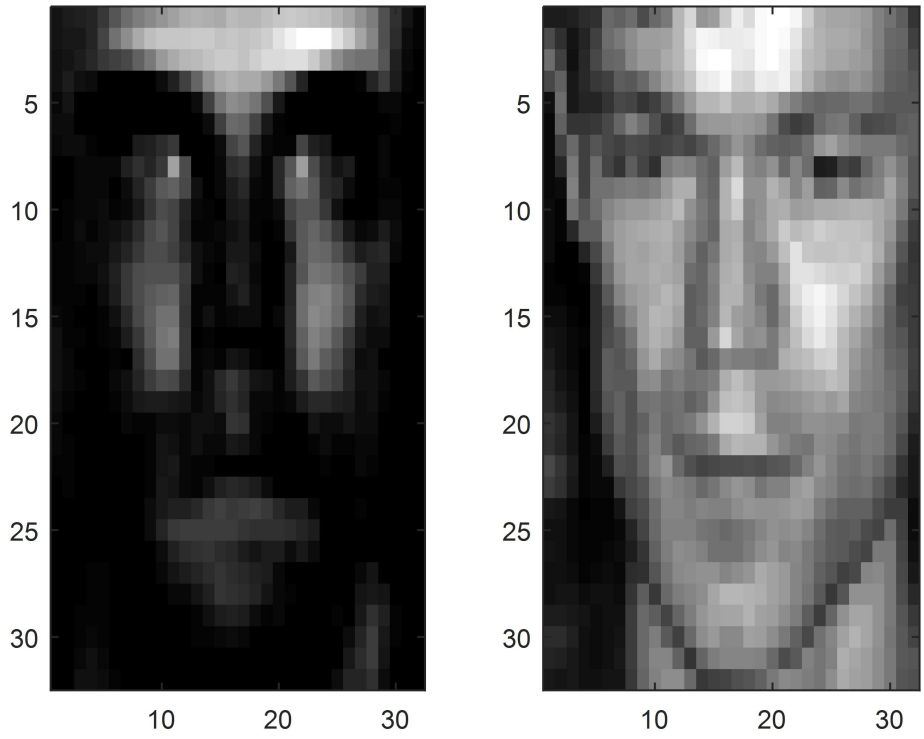
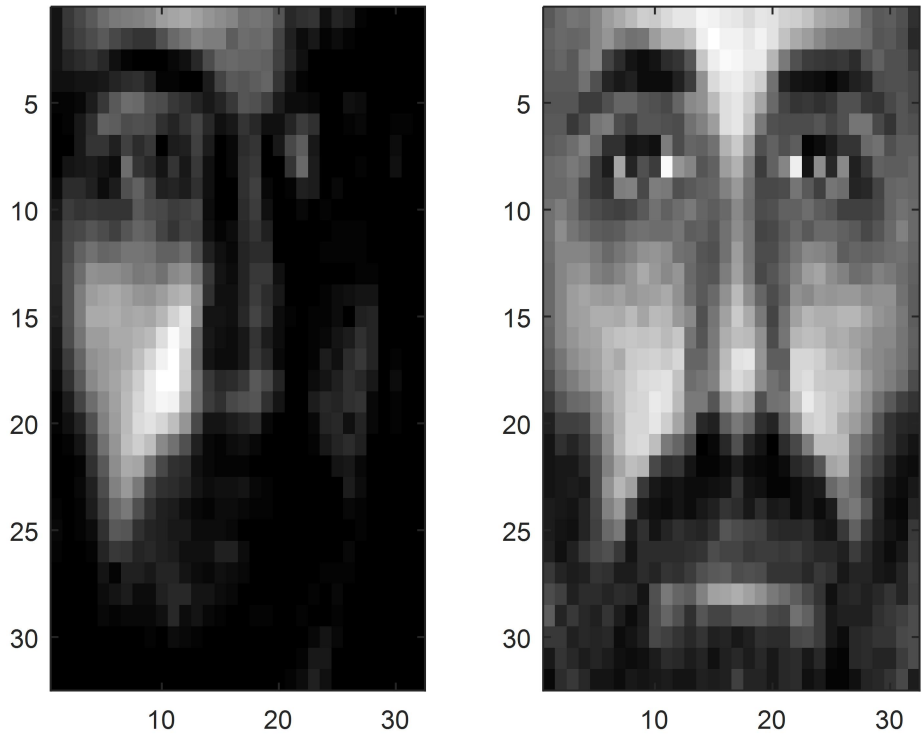

5
10
15
20
25
30
10
20
30
5
10
15
20
25
30
10
20
30


5
10
15
20
25
30
10
20
30


5
10
15
20
25
30
10
20
30
5
10
15
20
25
30
10
20
30


5
10
15
20
25
30
10
20
30
5
10
15
20
25
30
10
20
30

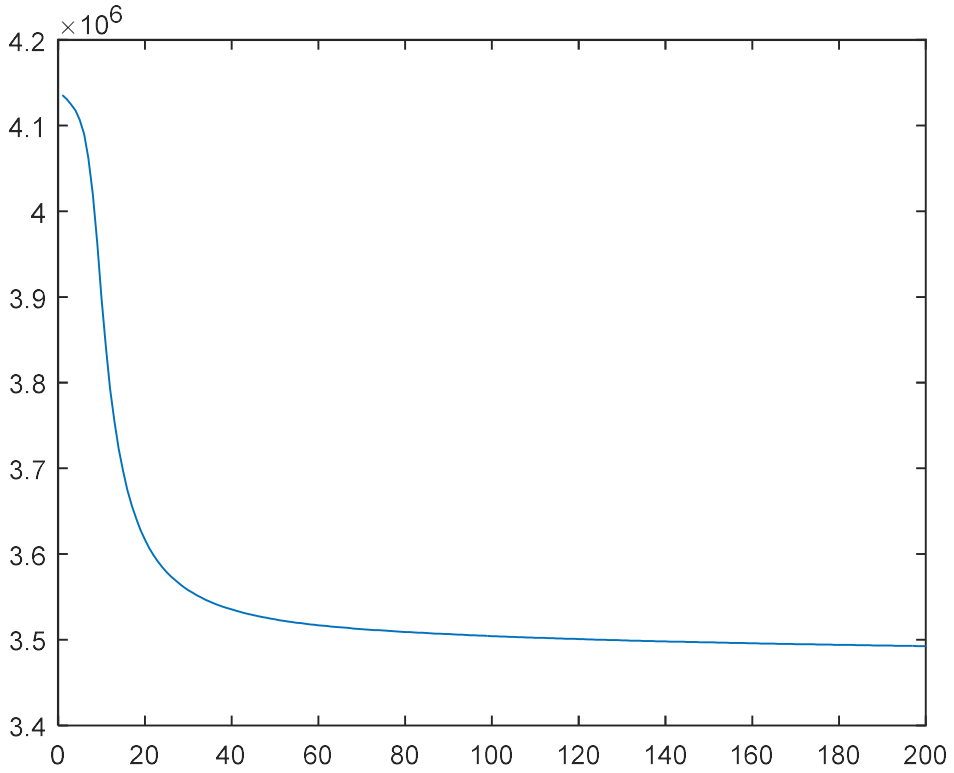

5
10
15
20
25
30
10
20
30
5
10
15
20
25
30
10
20
30

5
10
15
20
25
30
10
20
30
5
10
15
20
25
30
10
20
30


5
10
15
20
25
30
10
20
30
5
10
15
20
25
30
10
20
30

5
10
15
20
25
30
10
20
30
5
10
15
20
25
30
10
20
30

Part 2: The data to be used for Part 2 consists of 8447 documents from *The New York Times*. (See below for how to process the data.) The vocabulary size is 3012 words. You will need to use this data to constitute the matrix X, where X_{ij} is the number of times word i appears in document j. Therefore, X is 3012×8447 and most values in X will equal zero.

- Implement and run the NMF algorithm on this data using the *divergence penalty*. Set the rank to 25 and run for 200 iterations. This corresponds to learning 25 topics.
- Plot the objective as a function of iteration.

Solution



- After running the algorithm, normalize the columns of W so they sum to one. Pick 10 columns of W. For each column you select show the 10 words having the largest weight according to that vector and show the weight. The ith row of W corresponds to the ith word in the "dictionary" provided with the data.

Solution

Topic 1

index	weight	word
219	0.054991	'father'
253	0.048717	'mother'
325	0.045857	'son'
172	0.043181	'mrs'
460	0.035707	'daughter'
434	0.026491	'graduate'

45	0.024571	'family'
755	0.024543	'marry'
381	0.022446	'wife'
119	0.019181	'receive'

Topic 2

index	weight	word
10	0.046894	'school'
59	0.034726	'student'
283	0.016343	'class'
359	0.016163	'college'
21	0.014688	'child'
410	0.014547	'education'
14	0.01298	'program'
508	0.012884	'teacher'
295	0.01097	'parent'
11	0.010335	'group'

Topic 3

index	weight	word
19	0.00743	'tell'
17	0.006705	'home'
32	0.006435	'thing'
73	0.006372	'live'
372	0.006113	'sit'
477	0.006064	'wear'
178	0.006013	'friend'
316	0.005985	'walk'
20	0.005936	'woman'
88	0.005886	'feel'

Topic 4

index	weight	word
87	0.022951	'cost'
40	0.015404	'pay'
50	0.014156	'money'
2	0.013012	'percent'
65	0.012077	'number'
309	0.009399	'benefit'
14	0.009137	'program'
370	0.008222	'reduce'
594	0.008172	'insurance'
119	0.008146	'receive'

Topic 5

index	weight	word

6	0.020734	'game'
54	0.01826	'season'
104	0.01679	'player'
29	0.01663	'team'
642	0.01239	'baseball'
4	0.011816	'play'
196	0.011032	'hit'
635	0.010988	'fan'
644	0.009561	'league'
27	0.009389	'start'

Topic 6

index	weight	word
122	0.022512	'building'
150	0.013852	'build'
83	0.013661	'house'
220	0.012556	'site'
319	0.012516	'space'
103	0.01054	'area'
8	0.010086	'city'
53	0.009165	'open'
201	0.008807	'project'
415	0.008299	'floor'

Topic 7

index	weight	word
18	0.024381	'case'
127	0.022769	'court'
163	0.022653	'lawyer'
49	0.017133	'law'
314	0.016344	'judge'
89	0.016259	'charge'
478	0.011929	'legal'
405	0.011123	'trial'
444	0.010593	'file'
3	0.009585	'state'

Topic 8

index	weight	word
8	0.024707	'city'
3	0.024375	'state'
5	0.015015	'official'
180	0.013033	'local'
50	0.012482	'money'
135	0.011722	'tax'

9	0.011675	'plan'
258	0.011564	'budget'
221	0.01146	'community'
331	0.011368	'resident'

Topic 9

index	weight	word
32	0.010715	'thing'
51	0.008143	'put'
81	0.007925	'question'
62	0.00766	'keep'
204	0.007114	'fact'
15	0.007041	'point'
223	0.006845	'happen'
69	0.006832	'little'
60	0.006631	'problem'
147	0.006412	'mean'

Topic 10

index	weight	word
2	0.025581	'percent'
48	0.016369	'price'
35	0.016134	'market'
153	0.012944	'rise'
98	0.012934	'rate'
141	0.012693	'stock'
236	0.011244	'bank'
389	0.010582	'economy'
99	0.010304	'fall'
501	0.009102	'investor'