Problem Set4 : CS 6375

Ananya Banerjee

oblem 1: PCA & Feature Selection

SVM & PCAs (25 points)

- The top 6 eigen values of the data covariance

∴ matrix = [1073.72 ~~63.56349684~~ 462.28 ~~36.76148871~~ 61.33 ~~16.41583154~~
38.25 ~~10.71620451~~ 28.97 ~~9.68540952~~ 23.20 ~~8.94757052~~]

Validation Set

K	C	Accuracy	Err
1	1	57.6	42.4 %
1	10	57.9	42.1 %
1	100	59.6	42.4 %
1	1000	55.7	44.3 %
2	1	57.6 %	42.4 %
2	10	42.3 %	57.7 %
2	100	53.8 %	46.2 %
2	1000	48.07 %	51.9 %
3	1	59.6 %	40.4 %
3	10	50 %	50 %
3	100	57.6 %	42.4 %
3	1000	55.7 %	44.3 %

K	C	Accuracy	Err
4	1	57.6 %	42.4 %
4	10	50 %	50 %
4	100	48.07 %	51.93 %
4	1000	53.8 %	46.2
5	1	75 %	25 %
5	10	76.9 % ~~67.3~~ %	23.1 %
5	100	67.3 %	32.7 %
5	1000	59.6 %	40.4 %
6	1	76.9 %	23.1 %
6	10	82.6 %	17.4 %
6	100	73.07 %	26.93
6	1000	75 %	25 %

BEST PAIR: 6 10 82.6 %

4

- Test Set

k	c	Accuracy	Err
1	1	44.23%	55.77%
1	10	44.23%	65.77%
1	100	44.23%	55.77%
1	1000	38.46%	61.54%
2	1	50%	50%
2	10	57.6%	42.4%
2	100	51.9%	48.1%
2	1000	54.23%	45.77%
3	1	46.15%	53.85%
3	10	55.76%	42.4%
3	100	53.84%	46.16%
3	1000	59.61%	40.84%
4	1	46.15%	53.88%
4	10	59.6%	40.4%
4	100	63.94%	36.1%
4	1000	65.38%	34.62%

k	c	Accuracy	Err
5	1	71.11%	29%
5	10	78.94%	22%
5	100	78.89%	22%
5	1000	78.88%	22
6	1	69.23%	30
Best Pair 6	10	82.69%	17
6	100	78.84%	21.
6	1000	76.92%	23

For PC

Best Pair: k = 6, C=10

Accuracy = 82.69%.

Err = 17.31

* Without feature selection:

Accuracy for c = 10
for validation set

is 76.92%.

* Without feature selection:

Accuracy for c = 10
on test set

is 65.38%.

2)Problem: 1 SVM & PCA (continued)

mth) Without feature selection, accuracy is slightly lower than as compared with feature selection. The reason for this observation is that using PCA, we don't just reduce computation time by reducing demensions but we also choose the best features that effect our target variable ~~which~~ & take thoose to make our predict
Hence, getting better accuracy.

If I had to pick a value of k before ~~the~~ evaluating the perfomance, I would first look at the problem itself to develop a intuition regarding how many & what ~~cluster~~ columns or attributes or features make more sense & corelate better with the goal of the problem.

For example, if I was given a dataset & my goal was to predict whether or not someone of a certain age shall take the class or not, then intuitively columns such as 'Name', 'ID No' etc do not make much sense in terms of contributing to the goal i.e., predecting whether they will repeat a class or not

Also, trying to find relation (eg: corelation (very basic) tells linear relation between diff features) might help us develop better intuition.

(3)

Problem: 1 PCA & Feature Selection

- **PCA for feature selection (25 points)**

  Accuracy on test set :- 69.23%. [Gaussian Naive Bayes without PCA : HW-4_2.py]

- The top k=6 eigenvector & values are :- (Chosen Best K by PCA) [HW-4_naive_2.py for first 3 parts of Ques]

Top 6 eigen val = [~~63.56~~ ~~36.76~~ ~~16.41~~ ~~10.71~~ ~~9.68~~ ~~8.94~~
[1073.72 462.28 61.33 38.25 28.97 23.20]]

Top 6 eigen vec = [[-4.3866e-01, ...]] eigen vectors are in eigen_vec.txt

- For ~~Gaussian~~ naive bayes: (with π probability distribution)

  Best K = 8
  Best S = 12
  Accuracy = 78.84 %

The Avg Test Error for 100 iterations is attached with the homework

in avg-test_error.docx

I prob distribution is a function that describes the likelihood
f obtaining the possible values that a random variable can assume.
n the same way, π is a function that describes the probability
~~to~~ choosing a column given the possible values that the
ndom number can take. π defines the underlying probability
stribution of eigen vectors corresponding to features of the dataset

so, $\prod_{j=1}^{k} \pi_j = 1$

- This approach might not be reasonable alternative to Naive Bayes without featue selection since it's not reliable. For eg: for s=18, we might sample 18 columns differently for any of the 100 iterations & thus might get a range of accuracies or errors. This disuades us from deciding which values of k & s will work best & give us highest accuracy.

Also, how do we know if 100 iterations are good enough? We might do it a thousand times & still might get differnt results every time we run it.

The only pros that I can see so far is that we can encounter a certain set of features that might give better accuracy than Naive Bayes without feature selection.

[Basically PCA but a bit differently i.e, using T1, etc]

Problem 2 : Spectral Clustering (50 points)

The Basic Algorithm :-

1, 2, 3, 4 & 5 are done & stored in spectral_clustering-2.m.

In this case, L=D-A is positive semi definite coz for any scalar z of n length, $z^T L z$ is either positive or zero.
no of samples

A. Simple Comparison :-

- 1&2 are done & stored in Assignment_4_ques2_A_f.m

The 2 scatter plots are attached with the code as

scatter_simple1 & scatter-simple2
(Spectral Clustering) (K-Means Clustering)

- Choice of σ for which the Spectral Method outperforms is 0.1.

The Spectral clustering method uses P.C.A (choosing ~~least~~ top k-eigen values (vec) i.e, the least possible columns or features) to perform feature selection & then it runs k-means on those eigen vectors which creates much better clusters.

However, using only K-means directly works across all features & also takes into account the features which might effect the clusters the least.

Thus, we know that in this case spectral clustering works better than k-means always.

Partitioning Images

1 & 2 are done in Assignment-4-2-partition.m

The Img obtained after applying K-means is in KM fig.
The Img obtained after applying Spectral Clustering is in Spec.fig.
In this case as well, spectral works slightly better than

k-means due to the same reason as before.