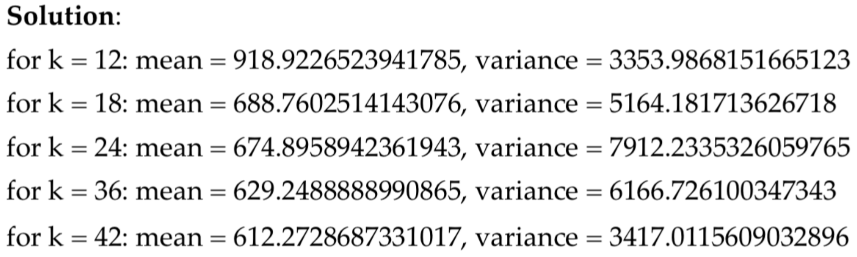
Homework 5 Solution

Problem 1:

1. (the results could be different)



Solution:

for k = 12: mean = 918.9226523941785, variance = 3353.9868151665123

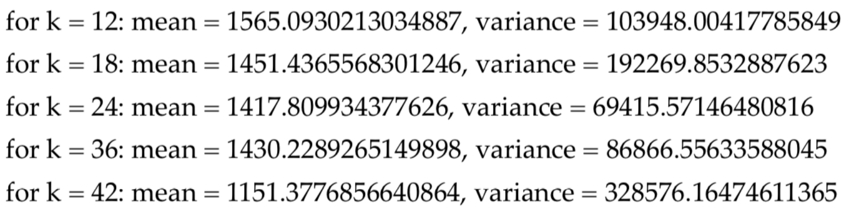
for k = 18: mean = 688.7602514143076, variance = 5164.181713626718

for k = 24: mean = 674.8958942361943, variance = 7912.2335326059765

for k = 36: mean = 629.2488888990865, variance = 6166.726100347343

for k = 42: mean = 612.2728687331017, variance = 3417.0115609032896

2. (the results could be different)



for k = 12: mean = 1565.0930213034887, variance = 103948.00417785849

for k = 18: mean = 1451.4365568301246, variance = 192269.8532887623

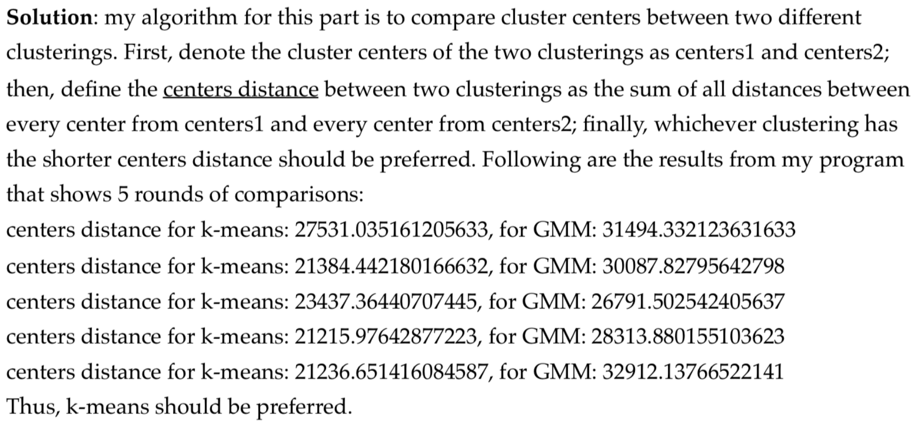
for k = 24: mean = 1417.809934377626, variance = 69415.57146480816

for k = 36: mean = 1430.2289265149898, variance = 86866.55633588045

for k = 42: mean = 1151.3776856640864, variance = 328576.16474611365

3. the answer of this question is based on your measurement.

Solution 1:



Solution: my algorithm for this part is to compare cluster centers between two different clusterings. First, denote the cluster centers of the two clusterings as centers1 and centers2; then, define the centers distance between two clusterings as the sum of all distances between every center from centers1 and every center from centers2; finally, whichever clustering has the shorter centers distance should be preferred. Following are the results from my program that shows 5 rounds of comparisons:

centers distance for k-means: 27531.035161205633, for GMM: 31494.332123631633

centers distance for k-means: 21384.442180166632, for GMM: 30087.82795642798

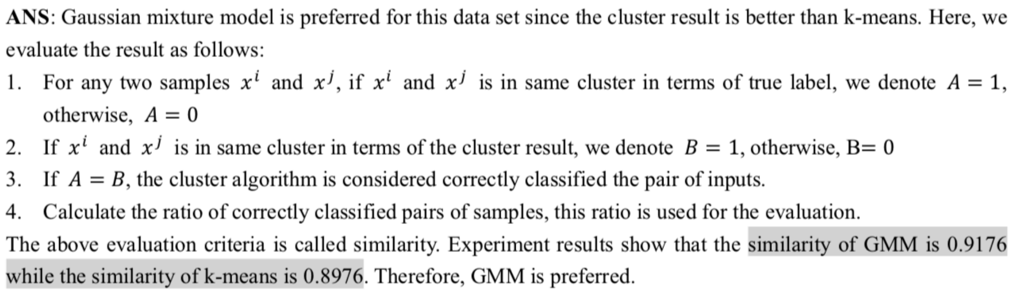
centers distance for k-means: 23437.36440707445, for GMM: 26791.502542405637

centers distance for k-means: 21215.97642877223, for GMM: 28313.880155103623

centers distance for k-means: 21236.651416084587, for GMM: 32912.13766522141

Thus, k-means should be preferred.

Solution 2:



ANS: Gaussian mixture model is preferred for this data set since the cluster result is better than k-means. Here, we evaluate the result as follows:

1. For any two samples x^i and x^j, if x^i and x^j is in same cluster in terms of true label, we denote $A = 1$, otherwise, $A = 0$
2. If x^i and x^j is in same cluster in terms of the cluster result, we denote $B = 1$, otherwise, B= 0
3. If $A = B$, the cluster algorithm is considered correctly classified the pair of inputs.
4. Calculate the ratio of correctly classified pairs of samples, this ratio is used for the evaluation.

The above evaluation criteria is called similarity. Experiment results show that the similarity of GMM is 0.9176 while the similarity of k-means is 0.8976. Therefore, GMM is preferred.

4.

Solution: This procedure does result in an improvement in both cases.

```
k_means:
for k = 12: mean = 898.6818845984996, variance = 1930.6080803894467
for k = 18: mean = 662.0575218978771, variance = 973.810242678748
for k = 24: mean = 540.6432792274977, variance = 444.3179851565401
for k = 36: mean = 394.7639344016793, variance = 287.0700344384582
for k = 42: mean = 340.3261453007869, variance = 136.6808634376481
GMM:
for k = 12: mean = 1589.3983573462497, variance = 51846.767089664914
for k = 18: mean = 1064.435604639249, variance = 313765.85596019216
for k = 24: mean = 426.0761077113583, variance = 746739.4268239825
for k = 36: mean = -590.8704535645077, variance = 256789.45577883153
for k = 42: mean = -703.6828964573909, variance = 3191.585397063181
```

5.

Solution: let x be a data point as a column vector, and μ_y as the mean of all data points w.r.t cluster y, then the covariance matrix for this cluster is defined as:

$$\Sigma_y = (x - \mu_y)(x - \mu_y)^T$$

to require this matrix to be diagonal means that:

$$\Sigma_y = \text{diag}\left[(x_1 - \mu_{y_1})^2 \quad (x_2 - \mu_{y_2})^2 \quad \dots \quad (x_m - \mu_{y_m})^2\right] = \text{diag}\left[\sigma_{y_1}^2 \quad \sigma_{y_2}^2 \quad \cdots \quad \sigma_{y_m}^2\right]$$

where $\sigma_{y_j}^2$ is the variance of the j-th component of all data points in cluster y.

Thus the only change we need to make in the EM algorithm is the iteration formula of Σ_y in the M-step. Instead of updating the whole covariance matrix, we only need to update the variances on individual component as:

$$(\sigma_{y_j}^2)^{t+1} = \frac{\sum_{i=1}^{N} q_i^t(y)(\sigma_{y_j}^2)^t}{\sum_{i=1}^{N} q_i^t(y)}$$

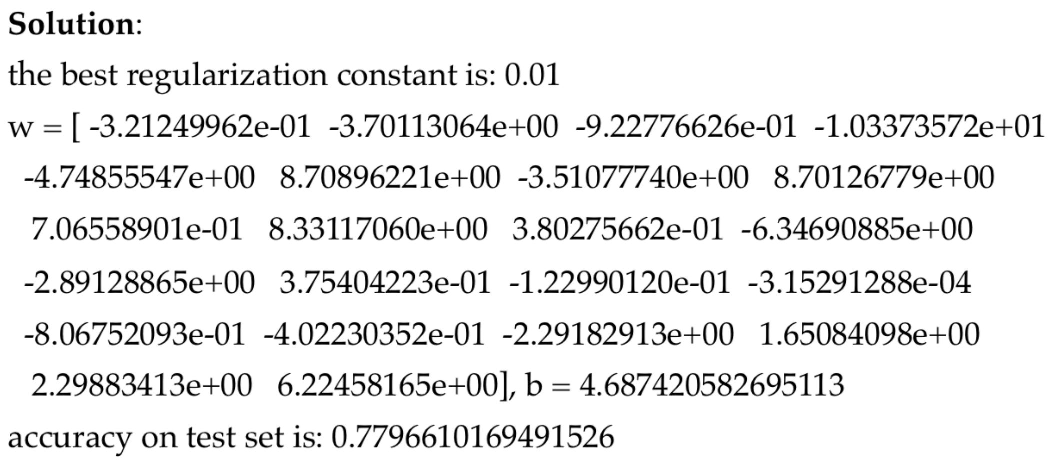
Problem 2:

1.

Accuracy : 79.66%

Because here we just want to maximize the likelihood, so the bigger the weights, the bigger the likelihood. A very large weights mean that the model is super confident about the label of the data points, but large weights may just have the same performance with the small weights.

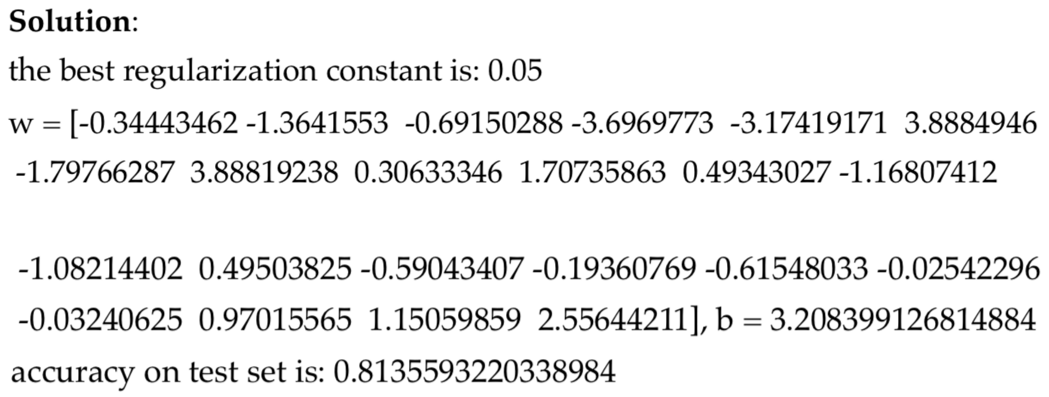
2.



Solution:

```
the best regularization constant is: 0.01
w = [ -3.21249962e-01  -3.70113064e+00  -9.22776626e-01  -1.03373572e+01
  -4.74855547e+00   8.70896221e+00  -3.51077740e+00   8.70126779e+00
   7.06558901e-01   8.33117060e+00   3.80275662e-01  -6.34690885e+00
  -2.89128865e+00   3.75404223e-01  -1.22990120e-01  -3.15291288e-04
  -8.06752093e-01  -4.02230352e-01  -2.29182913e+00   1.65084098e+00
   2.29883413e+00   6.22458165e+00], b = 4.687420582695113
accuracy on test set is: 0.7796610169491526
```

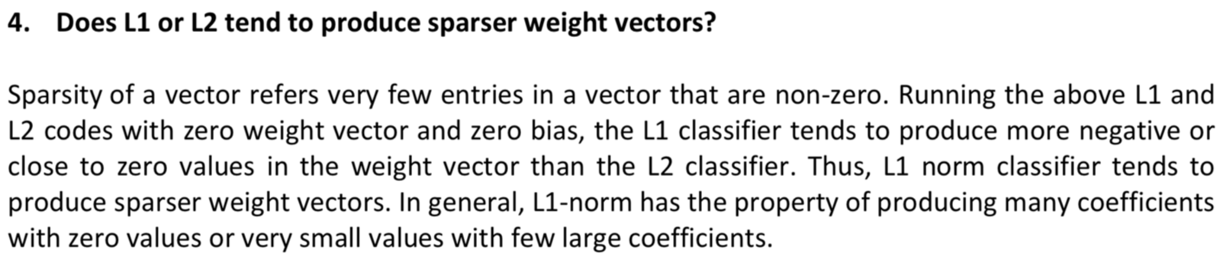
3.



Solution:

```
the best regularization constant is: 0.05
w = [-0.34443462 -1.3641553  -0.69150288 -3.6969773  -3.17419171  3.8884946
 -1.79766287  3.88819238  0.30633346  1.70735863  0.49343027 -1.16807412

 -1.08214402  0.49503825 -0.59043407 -0.19360769 -0.61548033 -0.02542296
 -0.03240625  0.97015565  1.15059859  2.55644211], b = 3.208399126814884
accuracy on test set is: 0.8135593220338984
```



4. Does L1 or L2 tend to produce sparser weight vectors?

Sparsity of a vector refers very few entries in a vector that are non-zero. Running the above L1 and L2 codes with zero weight vector and zero bias, the L1 classifier tends to produce more negative or close to zero values in the weight vector than the L2 classifier. Thus, L1 norm classifier tends to produce sparser weight vectors. In general, L1-norm has the property of producing many coefficients with zero values or very small values with few large coefficients.