Programming Assignment 1

Results

Question 1

k = 1:

Training Error: 0.0

Validation Error: 0.082

Test Error: 0.094

k = 5:

Training Error: 0.0565

Validation Error: 0.099

Test Error: 0.099

k = 9:

Training Error: 0.0705

Validation Error: 0.101

Test Error: 0.097

k = 15:

Training Error: 0.092

Validation Error: 0.107

Test Error: 0.116

The classifier of k = 1 performs the best on validation data.

The test error of this classifier is 0.094.

Question 2

k = 1:

Training Error: 0.0

Validation Error: 0.32

Test Error: 0.314

k = 5:

Training Error: 0.1975

Validation Error: 0.3

Test Error: 0.301

k = 9:

Training Error: 0.234

Validation Error: 0.295

Test Error: 0.286

k = 15:

Training Error: 0.2585

Validation Error: 0.288

Test Error: 0.306

The classifier of k = 15 performs the best on validation data.

The test error of this classifier is 0.306.

The classification accuracy decreases as affected by projection.

The program runs faster after projection as dimensions of matrices are reduced.

Code

In [343]:

```
train_data = open("./pa1train.txt", "r")
test_data = open("./pa1test.txt", "r")
validate_data = open("./pa1validate.txt", "r")
projection_data = open("./projection.txt", "r")
```

In [344]:

```
#train_data = [each_line.strip() for each_line in train_data]
train_data = [[int(s) for s in each_line.strip().split()] for each_line in train_data]

train = [i[:784] for i in train_data]
train_label = [i[-1] for i in train_data]
```

In [345]:

```
#test_data = [each_line.strip() for each_line in test_data]
test_data = [[int(s) for s in each_line.strip().split()] for each_line in test_data]

test = [i[:784] for i in test_data]
test_label = [i[-1] for i in test_data]
```

In [346]:

```
#validate_data = [each_line.strip() for each_line in validate_data]
validate_data = [[int(s) for s in each_line.strip().split()] for each_line in validate_data]

validate = [i[:784] for i in validate_data]
validate_label = [i[-1] for i in validate_data]
```

In [336]:

```
import numpy as np

projection_data = [each_line.strip() for each_line in projection_data]
projection = []

for each_line in projection_data:
    block = []
    split = each_line.split()
    for i in range(20):
        block.append(float(split[i]))
    projection.append(block)
projection = np.array(projection)
projection = projection.transpose().tolist()
```

Question 1

In [182]:

```
import random
def select(data):
    record = dict()
    for x in data:
        if x not in record:
            record[x] = 0
        record[x] += 1
    most = max([record[x] for x in record])
    output = [x for x in record if record[x] == most]
    return random.choice(output)
```

In [167]:

```
def get_dist(x, y):
    return sum([(x[i]-y[i])**2 for i in range(len(x))])
```

In [176]:

```
def data_dist(data, train_data):
    result = []
    for i in range(len(data)):
        each = []
        for j in range(len(train_data)):
            each.append((get_dist(data[i], train_data[j]), j))
        block = []
        for x in sorted(each):
            block.append(x[1])
        result.append(block)
    return result
```

In [315]:

```
def KNN(k):
    labels = []
    for i in train_dist:
        block = []
        for j in i[:k]:
            block.append(train_label[j])
        labels.append(block)
    temp = labels
    labels = []
    for x in temp:
        labels.append(select(x))
    train_error = sum(train_label[i] != labels[i] for i in range(len(train_label)))/len(train_label)
    print("Training Error: ", train_error)

    labels = []
    for i in validate_dist:
        block = []
        for j in i[:k]:
            block.append(train_label[j])
        labels.append(block)
    temp = labels
    labels = []
    for x in temp:
        labels.append(select(x))
    validate_error = sum(validate_label[i] != labels[i] for i in range(len(validate_label)))/len(validate_label)
    print("Validation Error: ", validate_error)

    labels = []
    for i in test_dist:
        block = []
        for j in i[:k]:
            block.append(train_label[j])
        labels.append(block)
    temp = labels
    labels = []
    for x in temp:
        labels.append(select(x))
    test_error = sum(test_label[i] != labels[i] for i in range(len(test_label)))/len(test_label)
    print("Test Error: ", test_error)
```

In [177]:

```
train_dist = data_dist(train, train)
```

In [179]:

```
test_dist = data_dist(test, train)
```

In [180]:

```
validate_dist = data_dist(validate, train)
```

In [316]:

```
KNN(1)
```

```
Training Error:  0.0
Validation Error:  0.082
Test Error:  0.094
```

In [317]:

```
KNN(5)
```

```
Training Error:  0.0565
Validation Error:  0.099
Test Error:  0.099
```

In [318]:

```
KNN(9)
```

```
Training Error:  0.0705
Validation Error:  0.101
Test Error:  0.097
```

In [319]:

```
KNN(15)
```

```
Training Error:  0.092
Validation Error:  0.107
Test Error:  0.116
```

In [377]:

```
KNN(3)
```

```
Training Error:  0.042
Validation Error:  0.093
Test Error:  0.085
```

Question 2

In [313]:

```
def add(x, y):
    result = []
    for i in range(len(x)):
        result.append(x[i] + y[i])
    return result
```

In [312]:

```
def dot(x, y):
    result = 0
    for i in range(len(x)):
        result += (x[i] * y[i])
    return result
```

In [314]:

```
def mul(m, x):
    return [m * x[i] for i in range(len(x))]
```

In [351]:

```
def proj(mat):
    i =1
    result = []
    for x in mat:
        block = [0] * len(x)
        for y in projection:
            block = add(block, mul(dot(x, y)/sum(np.array(y)**2), y))
        result.append(block)
    return result
```

In [369]:
```
def proj_KNN(k):
    labels = []
    for i in proj_train_dist:
        block = []
        for j in i[:k]:
            block.append(train_label[j])
        labels.append(block)
    temp = labels
    labels = []
    for x in temp:
        labels.append(select(x))
    train_error = sum(train_label[i] != labels[i] for i in range(len(t
rain_label)))/len(train_label)
    print("Training Error: ", train_error)

    labels = []
    for i in proj_validate_dist:
        block = []
        for j in i[:k]:
            block.append(train_label[j])
        labels.append(block)
    temp = labels
    labels = []
    for x in temp:
        labels.append(select(x))
    validate_error = sum(validate_label[i] != labels[i] for i in range
(len(validate_label)))/len(validate_label)
    print("Validation Error: ", validate_error)

    labels = []
    for i in proj_test_dist:
        block = []
        for j in i[:k]:
            block.append(train_label[j])
        labels.append(block)
    temp = labels
    labels = []
    for x in temp:
        labels.append(select(x))
    test_error = sum(test_label[i] != labels[i] for i in range(len(tes
t_label)))/len(test_label)
    print("Test Error: ", test_error)
```

In [353]:
```
proj_train = proj(train)
```

In [354]:
```
proj_test = proj(test)
```

```
In [355]: proj_validate = proj(validate)
```

```
In [359]: proj_train_dist = data_dist(proj_train, proj_train)
```

```
In [366]: proj_test_dist = data_dist(proj_test, proj_train)
```

```
In [367]: proj_validate_dist = data_dist(proj_validate, proj_train)
```

```
In [370]: proj_KNN(1)
```

```
Training Error:  0.0
Validation Error:  0.32
Test Error:  0.314
```

```
In [371]: proj_KNN(5)
```

```
Training Error:  0.1975
Validation Error:  0.3
Test Error:  0.301
```

```
In [380]: proj_KNN(9)
```

```
Training Error:  0.234
Validation Error:  0.295
Test Error:  0.286
```

```
In [382]: proj_KNN(15)
```

```
Training Error:  0.2585
Validation Error:  0.288
Test Error:  0.306
```

```
In [374]: proj_KNN(3)
```

```
Training Error:  0.157
Validation Error:  0.32
Test Error:  0.303
```