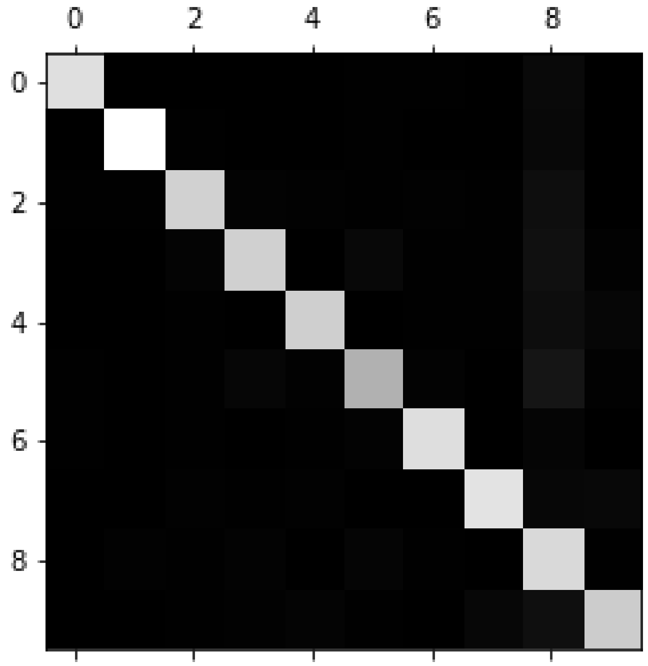
“Determining the Optimal Training Data Size for Recognizing Handwriting”

Elizabeth Chason

January 19, 2020

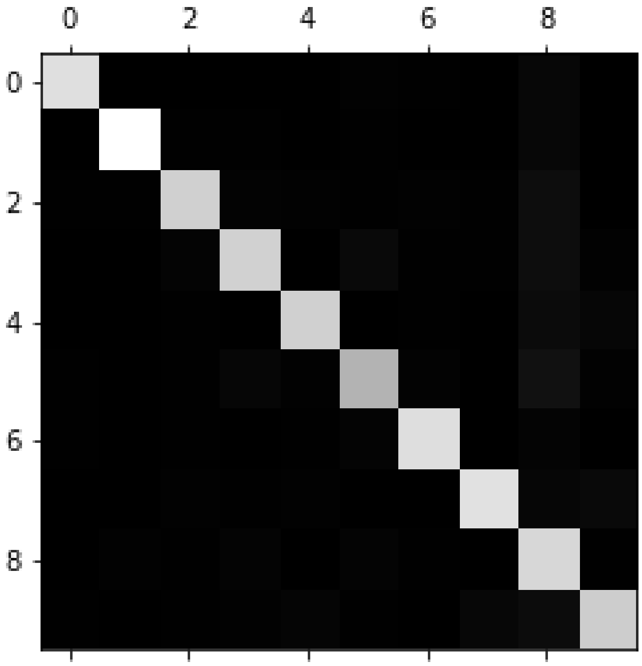
```
In [8]: from sklearn.datasets import fetch_openml

In [9]: mnist = fetch_openml('mnist_784', version=1)

In [10]: mnist.keys()

Out[10]: dict_keys(['data', 'target', 'feature_names', 'DESCR', 'details', 'categories', 'url'

In [11]: X, y = mnist["data"], mnist["target"]
         X.shape

Out[11]: (70000, 784)

In [12]:

['5' '0' '4' ... '4' '5' '6']


In [13]: import matplotlib as mpl
         import matplotlib.pyplot as plt

In [17]: import numpy as np
         y = y.astype(np.uint8)
         X_train, X_test, y_train, y_test = X[:60000], X[60000:], y[:60000], y[60000:]
         y_train_5 = (y_train == 5)
         y_test_5 = (y_test == 5)
         from sklearn.linear_model import SGDClassifier
         sgd_clf = SGDClassifier(max_iter=5000, random_state=42)
         from sklearn.preprocessing import StandardScaler
         from sklearn.model_selection import cross_val_predict
         from sklearn.metrics import confusion_matrix
         scaler = StandardScaler()
         X_train_scaled = scaler.fit_transform(X_train.astype(np.float64))
         y_train_pred = cross_val_predict(sgd_clf, X_train_scaled, y_train, cv=3)
         conf_mx = confusion_matrix(y_train, y_train_pred)
         plt.matshow(conf_mx, cmap=plt.cm.gray)
         conf_mx
         plt.show()
```



```
In [18]: conf_mx
```

```
Out[18]: array([[5576,    0,   21,    6,    9,   42,   37,    6,  225,    1],
                [   0, 6398,   38,   23,    4,   44,    4,    8,  213,   10],
                [  26,   27, 5242,   90,   71,   26,   62,   36,  371,    7],
                [  24,   17,  117, 5219,    2,  208,   28,   40,  406,   70],
                [  12,   14,   48,   10, 5192,   10,   36,   26,  330,  164],
                [  28,   15,   33,  166,   55, 4436,   76,   14,  539,   59],
                [  30,   14,   41,    2,   43,   95, 5558,    4,  130,    1],
                [  21,    9,   51,   26,   51,   12,    3, 5693,  190,  209],
                [  17,   63,   46,   90,    3,  125,   25,   10, 5429,   43],
                [  23,   18,   31,   65,  116,   32,    1,  179,  378, 5106]])
```

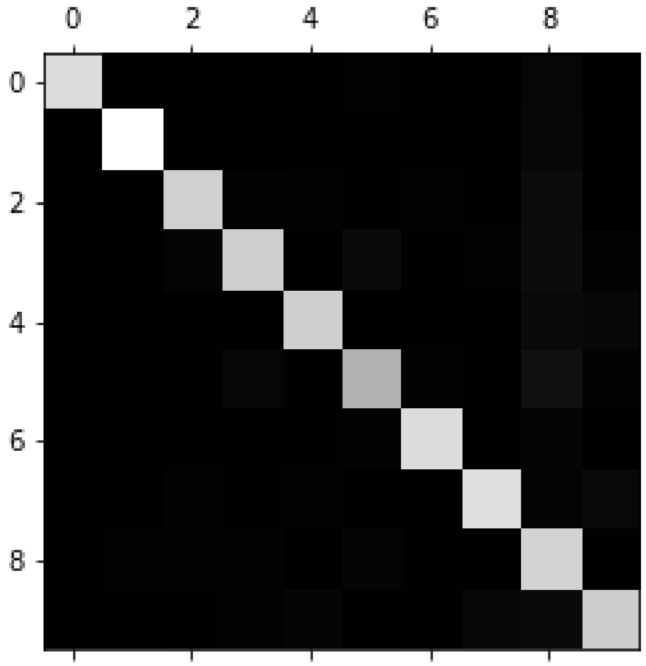
```
In [38]: X_train2, X_test2, y_train2, y_test2 = X[:56000], X[56000:], y[:56000], y[56000:]
         X_train_scaled2 = scaler.fit_transform(X_train2.astype(np.float64))
         y_train_pred2 = cross_val_predict(sgd_clf, X_train_scaled2, y_train2, cv=3)
         conf_mx2 = confusion_matrix(y_train2, y_train_pred2)
         plt.matshow(conf_mx2, cmap=plt.cm.gray)
         plt.show()
```



```
In [39]: conf_mx2
```

```
Out[39]: array([[5225,    0,   21,    6,    7,   51,   36,    4,  179,    1],
                [   1, 5992,   40,   27,    4,   45,    3,    7,  179,   10],
                [  24,   27, 4881,   90,   65,   31,   65,   41,  325,   11],
                [  23,   18,  109, 4903,    1,  221,   28,   42,  313,   70],
                [  13,   13,   37,   10, 4877,   14,   38,   19,  264,  163],
                [  26,   15,   31,  160,   51, 4207,   74,   16,  420,   60],
                [  26,   15,   41,    2,   41,   94, 5201,    5,  110,    2],
                [  17,   11,   51,   26,   50,   11,    2, 5281,  141,  220],
                [  19,   58,   46,   94,    3,  113,   27,    9, 5047,   44],
                [  25,   19,   28,   66,  118,   34,    1,  179,  288, 4801]])
```

```
In [40]: X_train3, X_test3, y_train3, y_test3 = X[:49000], X[49000:], y[:49000], y[49000:]
         X_train_scaled3 = scaler.fit_transform(X_train3.astype(np.float64))
         y_train_pred3 = cross_val_predict(sgd_clf, X_train_scaled3, y_train3, cv=3)
         conf_mx3 = confusion_matrix(y_train3, y_train_pred3)
         plt.matshow(conf_mx3, cmap=plt.cm.gray)
         plt.show()
```



```
In [41]: conf_mx3

Out[41]: array([[4566,    0,   19,    6,    8,   46,   34,    4,  141,    1],
                [   0, 5297,   40,   20,    3,   40,    3,    7,  150,   10],
                [  26,   26, 4311,   75,   59,   23,   62,   34,  242,    9],
                [  20,   18,  112, 4272,    2,  189,   20,   44,  252,   57],
                [  12,   12,   33,   10, 4298,    9,   33,   15,  201,  150],
                [  23,   12,   28,  137,   40, 3683,   64,   12,  340,   67],
                [  25,   15,   38,    2,   37,   79, 4565,    4,   89,    1],
                [  14,   10,   48,   27,   45,   10,    4, 4622,   93,  203],
                [  17,   59,   43,   78,    3,  104,   26,    7, 4377,   37],
                [  20,   16,   27,   59,  110,   34,    1,  150,  216, 4258]])

In [48]: flist = []
         for num in range(10):
             recall = conf_mx[num][num] / sum(conf_mx[num])
             lisst = []
             for i in range(10):
                 prec_denom = conf_mx[i][num]
                 lisst.append(prec_denom)
             precision = conf_mx[num][num] / sum(lisst)
             fone = 2 / ((1/precision) + (1/recall))
             flist.append(fone)
         cumulfone1 = sum(flist)
         meanfone1 = cumulfone / 10
         meanfone1
```

```
Out[48]: 0.8985060679733488
```

```
In [53]: flist2 = []
         for num in range(10):
             recall2 = conf_mx2[num][num] / sum(conf_mx2[num])
             lisst2 = []
             for i in range(10):
                 prec_denom2 = conf_mx2[i][num]
                 lisst2.append(prec_denom2)
             precision2 = conf_mx2[num][num] / sum(lisst2)
             fone2 = 2 / ((1/precision2) + (1/recall2))
             flist2.append(fone2)
         cumulfone2 = sum(flist2)
         meanfone2 = cumulfone2 / 10
         meanfone2
```

```
Out[53]: 0.9006294715151826
```

```
In [54]: flist3 = []
         for num in range(10):
             recall3 = conf_mx3[num][num] / sum(conf_mx3[num])
             lisst3 = []
             for i in range(10):
                 prec_denom3 = conf_mx3[i][num]
                 lisst3.append(prec_denom3)
             precision3 = conf_mx3[num][num] / sum(lisst3)
             fone3 = 2 / ((1/precision3) + (1/recall3))
             flist3.append(fone3)
         cumulfone3 = sum(flist3)
         meanfone3 = cumulfone3 / 10
         meanfone3
```

```
Out[54]: 0.9029226433725686
```