ECEN 649 Homework #1: Naive Bayes and Perceptron

Colton Riedel

All code along with sample output are available online at my github:
https://github.com/coltonriedel/pattern_recognition/

Problem 1

I did not use quantization in my approach. I did use Laplace (Lidstone) smoothing with a smoothing parameter equal to 1 (*i.e.* I added 1 to the numerator and 256 to the denominator of each conditional probability). When performing predictions I added the log of each probability to minimize errors due to loss of precision.

Output including priors, randomly selected conditional probabilities, and accuracy:

```
Parsed 60000 training set records in 1.55652 seconds
Parsed 10000 test set records in 0.249572 seconds
Trained model in 16.0305 seconds

Prior probabilities:
        0: 0.0987167
        1: 0.112367
        2: 0.0993
        3: 0.102183
        4: 0.0973667
        5: 0.09035
        6: 0.0986333
        7: 0.104417
        8: 0.0975167
        9: 0.09915

Randomly selected conditional probabilities:
        P(x_517 = 218 | y = 6) = 0.000485909
        P(x_41  = 197 | y = 7) = 0.000153351
        P(x_6   = 128 | y = 7) = 0.000153351
        P(x_627 = 82  | y = 2) = 0.0024139
        P(x_289 = 122 | y = 7) = 0.000613403
        P(x_442 = 70  | y = 9) = 0.000483481
        P(x_468 = 250 | y = 9) = 0.000805802
        P(x_745 = 70  | y = 9) = 0.000322321
        P(x_73  = 198 | y = 8) = 0.000163747
        P(x_702 = 107 | y = 5) = 0.000176149

Evaluated 10000 test records in 0.279734 seconds with 83.63% accuracy

        Digit     Precision       Recall
        0         0.889796        0.91023
        1         0.970925        0.860938
        2         0.790698        0.893757
        3         0.826733        0.766055
        4         0.822811        0.82449
        5         0.674888        0.775773
        6         0.889353        0.884735
        7         0.838521        0.912169
        8         0.779261        0.791449
        9         0.847374        0.752641
```

Problem 2

Output including weights, bias, and accuracy:

```
Parsed 75 training set records in 0.000129694 seconds
Parsed 25 test set records in 3.8692e-05 seconds

Trained perceptron with 1 iterations in 4.958e-06 seconds

Evaluated 25 test records in 1.746e-06 seconds with 100% accuracy
Weights: y = x_1 * (0.41) + x_2 * (1.31) + -1.99
```

Plots of training set and test set data, with perceptron boundary ($z = 0$) shown: