1 PCA vs naive Bayes for digit recognition

Download code.zip. Load the file usps_digits_479.mat (or usps_digits_479V6.mat if you have matlab 6) which contains data.Xtrain of size 300 × 256, data.ytrain of size 300 × 1, data.Xtest of size 300 × 256, data.Xytest of size 300 × 1. Each data case is a 16 × 16 image of a handwritten digit, either a 4, 7 or 9.

1. Train a classifier in which the class conditional density is based on PCA and is given by

\[ p(x|y = c) = \mathcal{N}(x|\mu_c, W_c W_c^T + \sigma_c^2 I) \]  

where \( W_c, \mu_c, \) and \( \sigma_c^2 \) are estimated using PPCAFit applied to the training data for class \( c \).

2. Using a uniform class prior \( p(y = c) \propto 1 \), compute the class posterior \( p(y = c|x_i) \). You can use the function ppcaLoglik and logsumexp. Plot the posterior as an an image using imagesc(posterior). You should get something like Figure 1(left).

3. Compute the MAP estimate \( \hat{y}_i = \arg \max_c p(y = c|x_i) \) for each test case. Compute the number of errors, \( \sum_i I(\hat{y}_i \neq y_i) \) for each test case. (I get 16 errors). Which test cases did you get wrong? Plot the first 9 erroneously labeled images using something like

```matlab
img = reshape(data.Xtest(i, :), [16 16]);
imagesc(img); colormap(gray); axis off
```

You should get something like Figure 1(right).

![Figure 1](image.png)

Figure 1: Left: posterior over 3 classes and 300 test cases using PPCA. Right: first 9 erroneously labeled images in test set.
Figure 2: Left: posterior over 3 classes and 300 test cases using Naive Bayes. Right: first 9 erroneously labeled images in test set.

4. Now repeat all of the above using a naive Bayes classifier

\[
p(x|y = c) = \prod_{j=1}^{d} N(x_j|\mu_{cj}, \sigma_{cj}^2)
\]

where \(d = 256\) represent the number of dimensions. I get 70 errors and Figure 2.