
```

% Create new features with the PCA algorithm for MNIST

[W,M_data_train_PCA] = pca(M_data_train);
M_data_test_PCA = M_data_test*W;

% Choose the number of PCA component to add to the original data
nb_comp = 10*24;
M_data_train_PCA = [repmat(M_data_train_PCA(:,1:nb_comp), [1 2])
    M_data_train];
M_data_test_PCA = [repmat(M_data_test_PCA(:,1:nb_comp), [1 2])
    M_data_test];

% Train the classifier
[M_new_means, M_new_variances] =
    f1_train_naive_bayes_classifier( M_data_train_PCA, M_labels_train);

% Test the predictions on the test data for the MNIST dataset
[M_labels_prediction, M_confusion_matrix, M_accuracy] =
    f2_predict_naive_bayes_classifier( M_new_means, M_new_variances,
    M_data_test_PCA, M_labels_test, 0.084);

% Display the confusion matrix and the accuracy
M_confusion_matrix
M_accuracy

% Display the confusion matrix through an image
close all
colormap hot;
image(M_confusion_matrix*2.5);
title('confusion matrix - naive bayes classifier - MNIST dataset with
    PCA')

% plot the PCA data
figure()
colormap hsv
x = M_data_train_PCA(1:10000,1);
y = M_data_train_PCA(1:10000,2);
sz = 5;
c = 3+M_labels_train(1:10000);
scatter(x,y,sz,c,'filled')
title('two principal components of the PCA-data-MNIST')

clearvars W x y sz c

```

```
M_confusion_matrix =
```

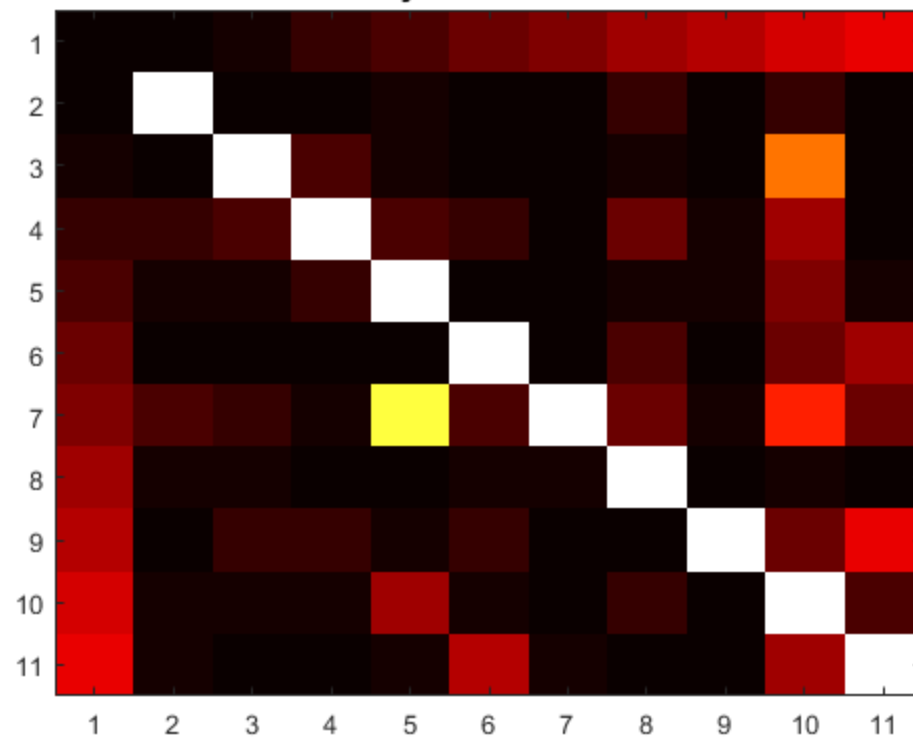
NaN	0	1	2	3	4	5	6	7	8	9
0	95	0	0	1	0	0	2	0	2	0
1	0	81	3	1	0	0	1	0	14	0
2	2	3	80	3	2	0	4	1	6	0
3	1	1	2	88	0	0	1	1	5	1

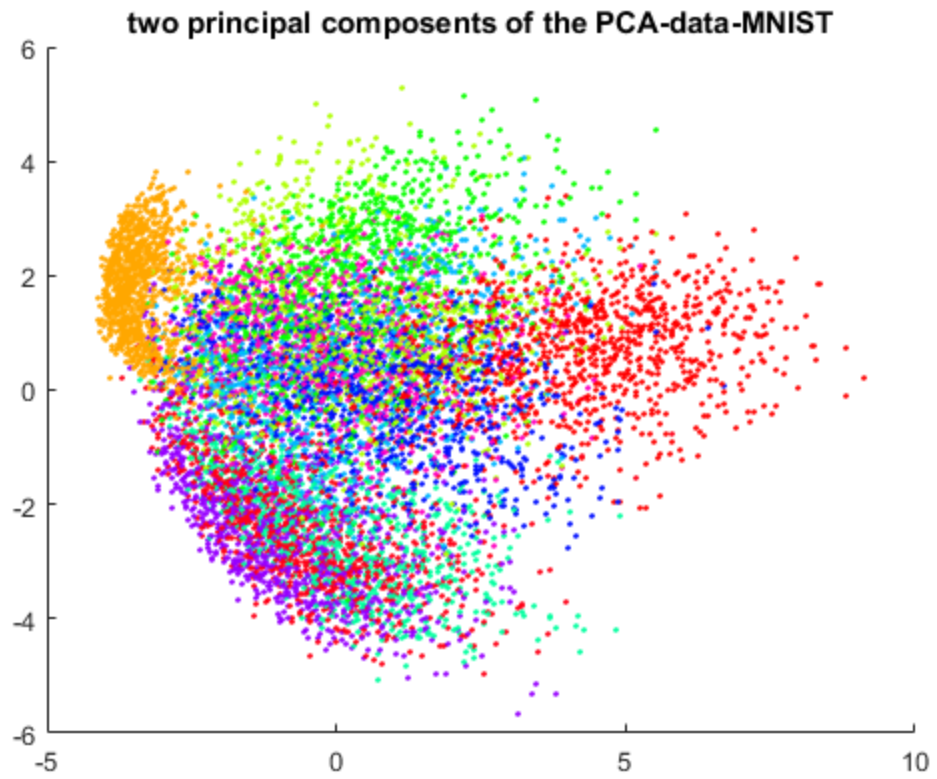
4	0	0	0	0	85	0	3	0	4	6
5	3	2	1	21	3	52	4	1	11	4
6	1	1	0	0	1	1	94	0	1	0
7	0	2	2	1	2	0	0	80	4	9
8	1	1	1	6	1	0	2	0	85	3
9	1	0	0	1	7	1	0	0	6	83

$M_{accuracy} =$

0.8254

confusion matrix - naive bayes classifier - MNIST dataset with PCA





Published with MATLAB® R2015b