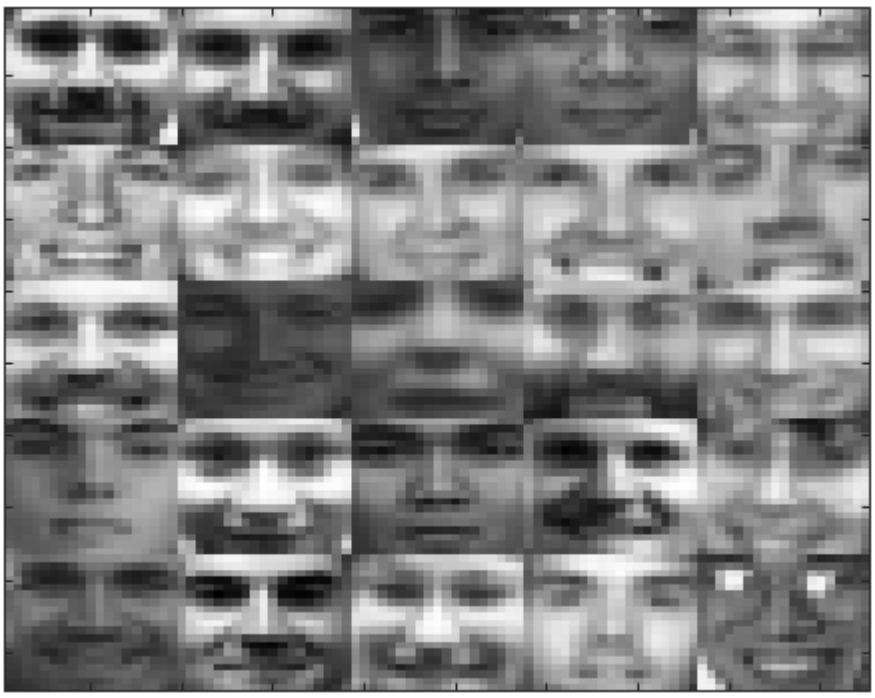
```
% PRINcipal COMPonent calculator
%   Calculates the principal components of a collection of points.
% Input:
%   X - D-by-N data matrix of N points in D dimensions.
% Output:
%   W - A D-by-M matrix containing the M principal components of the data.
%   Z - A M-by-N matrix containing the latent variables of the data.
%   mu - A D-by-1 vector containing the mean of the data.
%   lambda - A vector containing the eigenvalues associated with the above principal components.

function [W, Z, mu, lambda] = princomp(X,M)
    [~,N] = size(X);

    %remove mean from X to make projection computation simple
    mu = mean(X,2);
    X = X - repmat(mu,[1 N]);

    %SVD of X = U*S*V^T. Here we only need U,S as cov(X) = 1/N(X*X^T),
    %so S^2 has eigenvalues and U has normalized eigenvectors of X*X^T
    [U,S,~] = svd(X);

    %extract the M principle components from U into W. Since singular values
    %in matlab are stored in descending order in S, indexing is easy.
    W = U(:,1:M);

    %similarly extract leading M eigenvalues from S into lambda
    %note we divide by N since cov(X) = (1/N)*(X*X^T)
    lambda = zeros(M);
    for i = 1:M
        lambda(i) = S(i,i)^2;
    end
    lambda = lambda/N;

    %Zij stores inner product <ui,xj>, which are the latent variables
    %can think of Z as "the coordinates of X w.r.t W basis", although
    %the basis is strictly far from spanning.
    Z = W'*X;
end
```

Published with MATLAB® R2020b

Contents

1.2 load CBCL and visualize faces

```
load cbcl.mat

%Here X is a 361x2900 data set, for 2900 image of size 361 pixels

%randomly pick 25 data pts and store in idx
idx = randperm(size(X,2), 25);

%display the images of the 25 random data pts in 5x5 grid
figure;
imgrid(X(:,idx), dims, [5 5]);
```

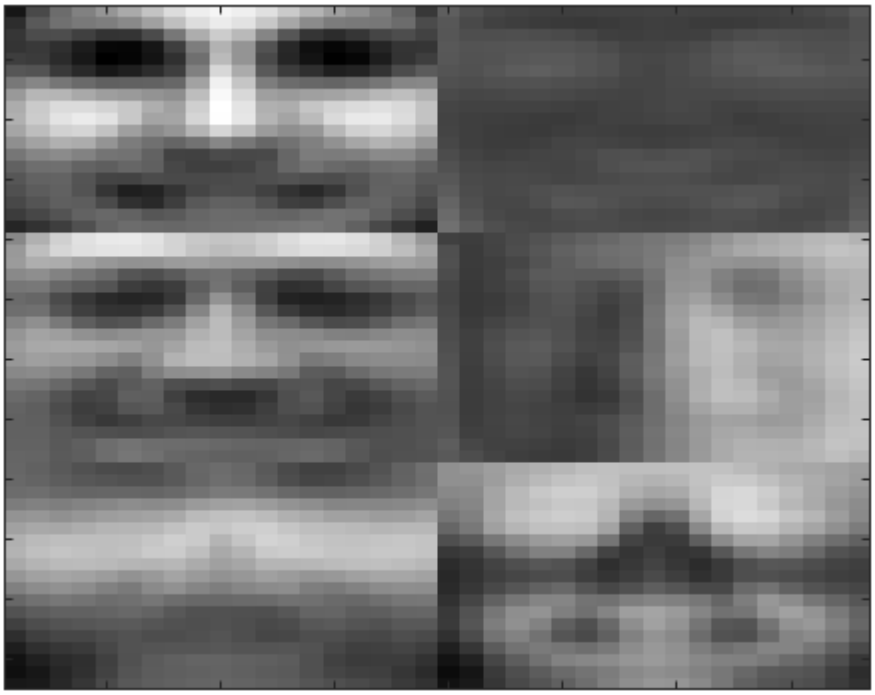


1.3.1 apply PCA on data

```
%do PCA dimension reduction onto M = 25 components
[W, Z, mu, lambda] = princomp(X, 25);
```

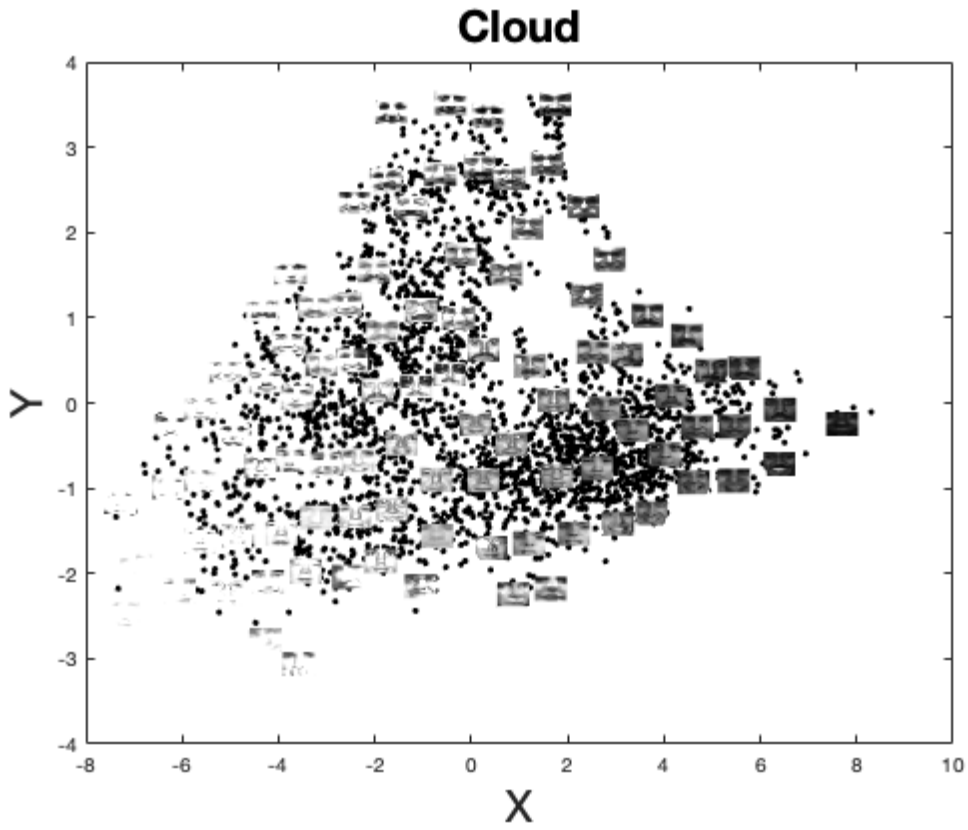
1.3.2 display first 5 principle components/"eigenfaces"

```
%display mean face in (1,1) & first five eigenfaces for the remaining
%2*W+0.5 accentuates the eigenfaces for clarity

%observations: The leading principle components are the basis vectors that
%are used to reconstruct most features in the face, so as seen, the first 5
%leading principle component pick up important components that when added to
%the mean face at different quantities, will recover most important features
%in the faces. also worth noting that features from different data pts will
%vary most along the first principle component, due to it contributing to
%most variance (as max eigenvalue) b/w data pts. The importance of subsequent
%princomps decrease since eigenvalues are in descending order.

%for example, the first princomp in row 1 col 2 seems to be adding to the mean
%face its specific shade of color since the pattern is similar, then the second
%to fifth princomp contributes some outline of face, e.g. where eyes, mouth,
%dents/protruding area e.g. nose, etc. are located
figure;
imgrid([mu/max(mu), 2*W(:,1:5)+0.7], dims, [3 2]);
```



1.4 Choosing leading 2 principle components

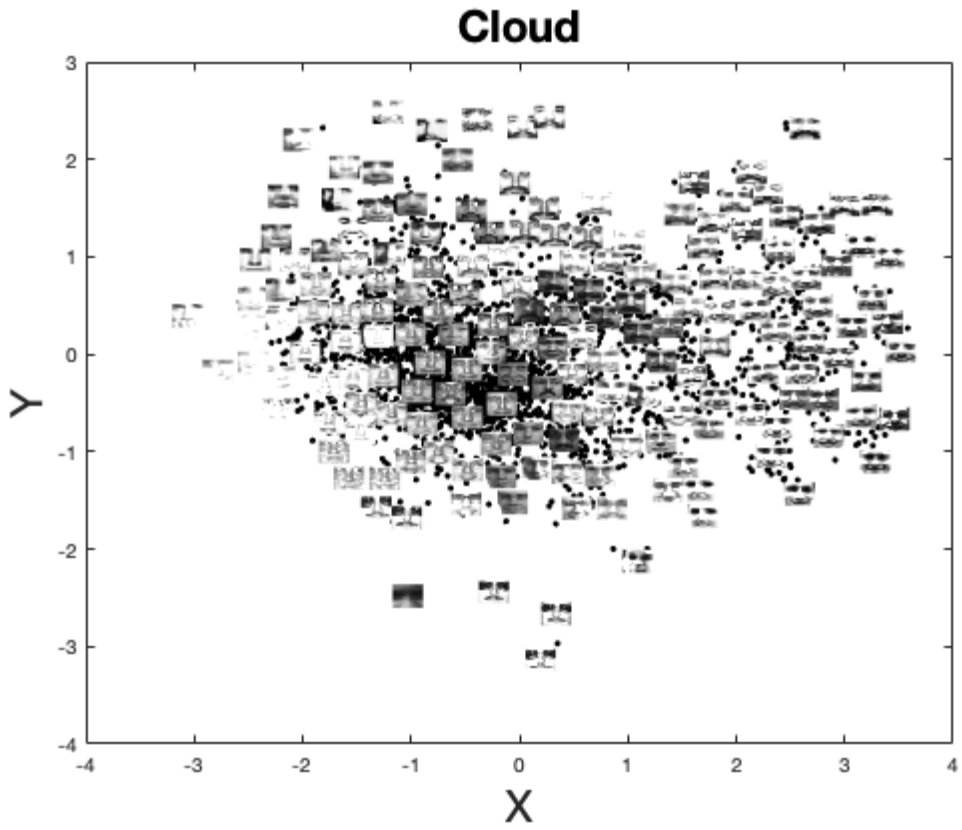
```
%plot the latent variables (first 2 for each data pt) on the XY plane to
%visualize spread of data in a subspace of the leading 2 PCA components

%notice the spread is pretty wide, which is expected since the projected flat
%is the 2-dimensional subspace s.t. data achieves maximum variance
figure;
imcloud(Z(1:2,:)',X,dims);
```

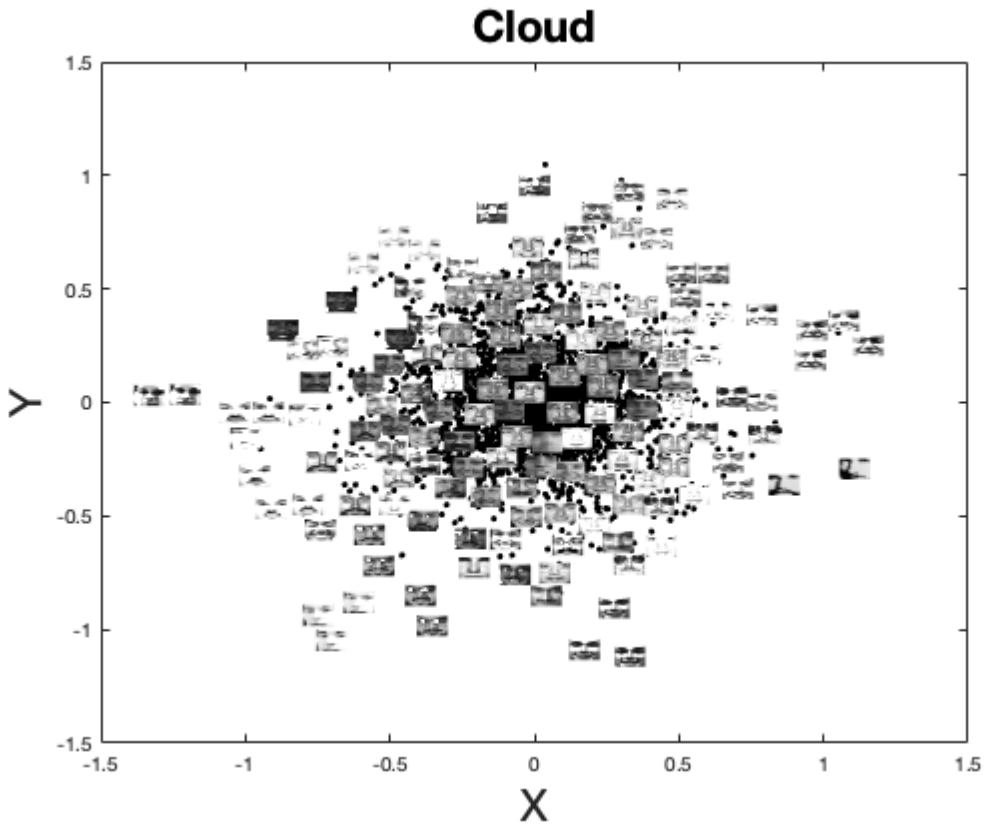


1.5 Choosing other principle components

```
%Here we project on 2nd and 4th principle components, which suggest variance
%between data points is smaller by choosing smaller eigenvalues.

%as expected, there is more cluttering in the scatterplot
figure;
imcloud(Z([2 4],:)',X,dims);


%project on 11th and 20th principle components, so points even more cluttered
figure;
imcloud(Z([19 20],:)',X,dims);

%to improve description of info captured by principle components always use
%the first few principle components.
```

But interestingly, the outliers in the cluttered plot of these latter princomp plots suggest that these less significan princomp are still picking up features from certain members which could be useful details e.g. particular types of faces, slanted views of faces, moustauche, etc.

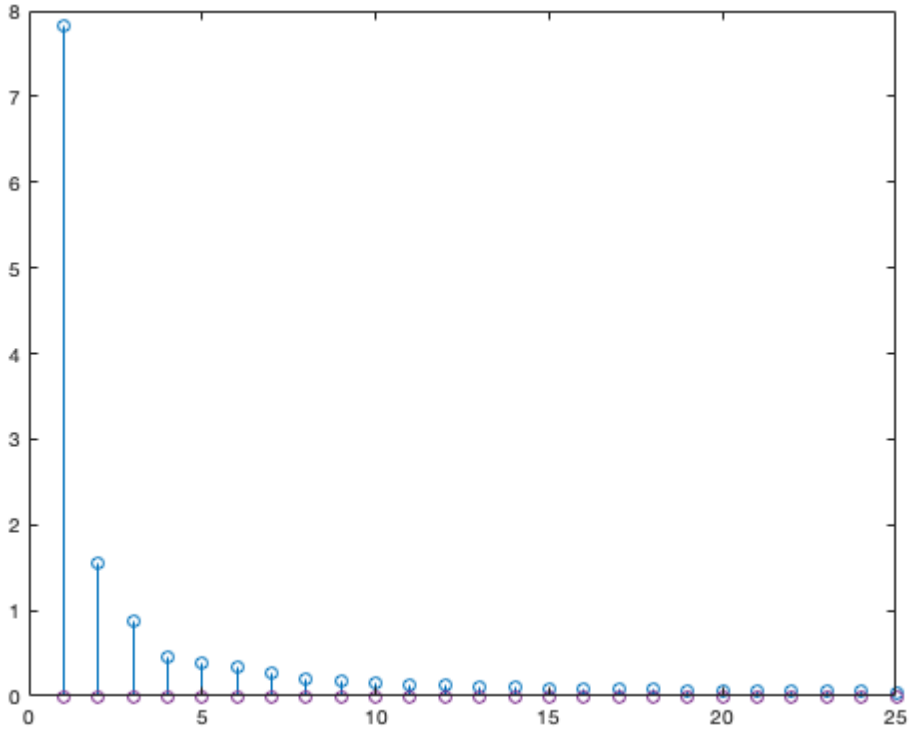




1.6 Eigenvalue plot and implication

```
%perhaps if we're interested in the best compromise between speed and accuracy
%of a ML method (e.g. nearest neighbour classification), this stem-plot is
%useful as sum of eigenvalues is the variance of the data pts in our flat, so
```

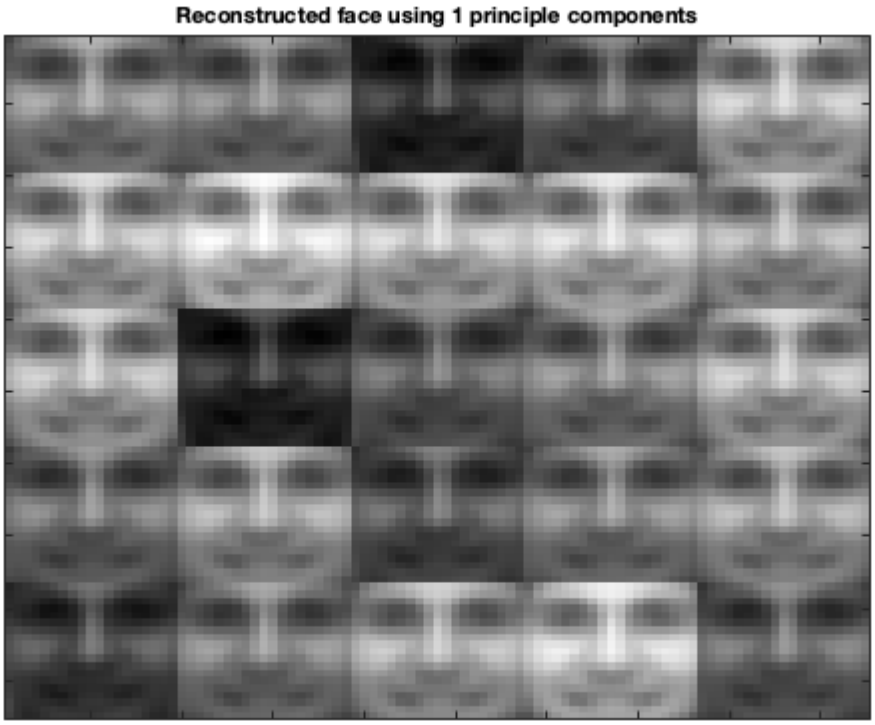
```
%this provides info on how much variance each eigenvalue contributes, i.e. the
%accuracy gained by introducing an additional principle component, which helps
%us decide whether we want to add the component.

%for example, clearly the leading eigenvalue is very important, and the
%second and third is also pretty important, but next few are starting to
%contribute finer and smaller details/variations
figure;
stem(lambda);
```

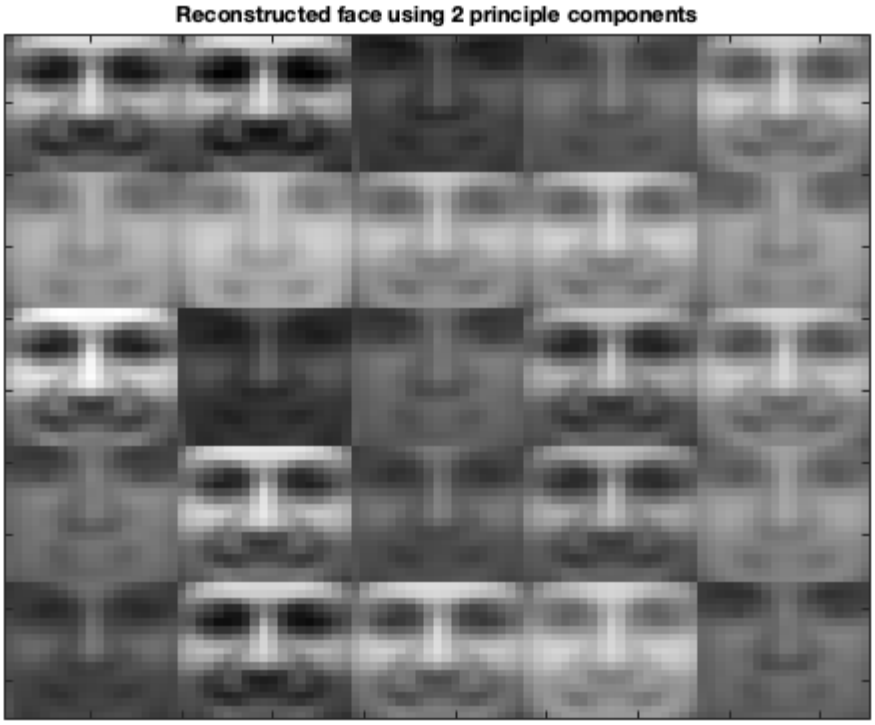


1.7 Projection of X onto W via Z

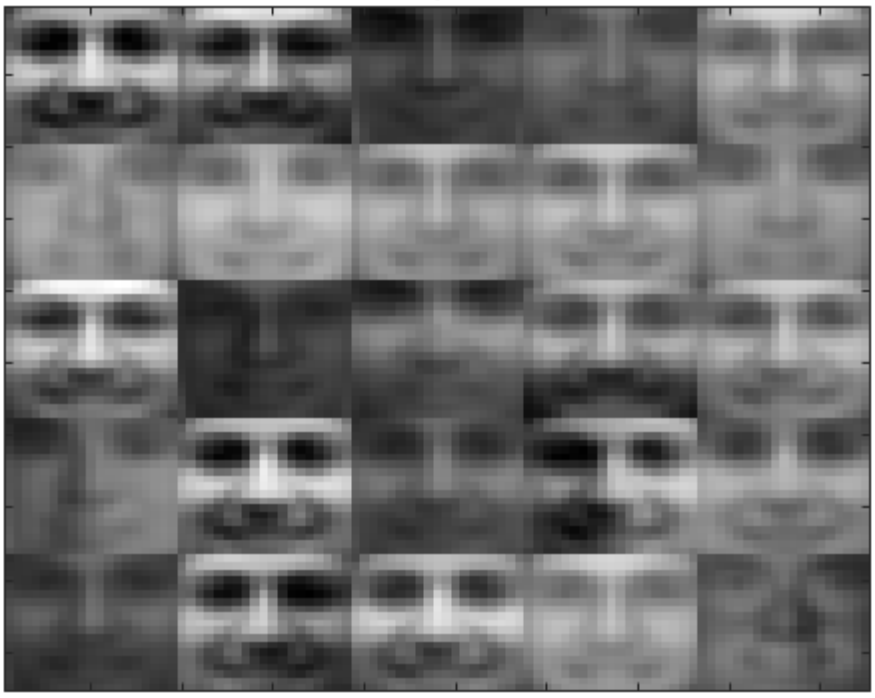
```
%using various M's which represent number of dimension in PCA subspace,
%visualize how well the data reconstructs
M = [1 2 5 10 25];

%observation: As seen below, using 1 or 2 principle components recovers the
%general attributes of faces that gives them clear distinction, but the more
%detailed features need 10, 25 or perhaps more principle components to extract

%here we verify that indeed the 1st princomp gives the face its shade, then
%the 2nd princomp gives to faces various degree of a specific outline that
%may vary rather greatly between faces, and so on... The face reconstructed
%from 25 dimension subspace seem to approximate actual face well.though it
%does appear 25 princomps is not enough to capture glasses yet. This is
%kind of expected as the glasses may be thin differences that doesnt generate
%alot of variance to be captured in leading princomps.
for i = 1:length(M)
    figure;
    imgrid(W(:,1:M(i))*Z(1:M(i),idx)+repmat(mu, [1 25]),dims,[5 5]);
    title(['Reconstructed face using ', num2str(M(i)), ' principle components']);
end
```

```
figure;
imgrid(X(:,idx), dims, [5 5]);
title('Original faces (for comparison)');
```

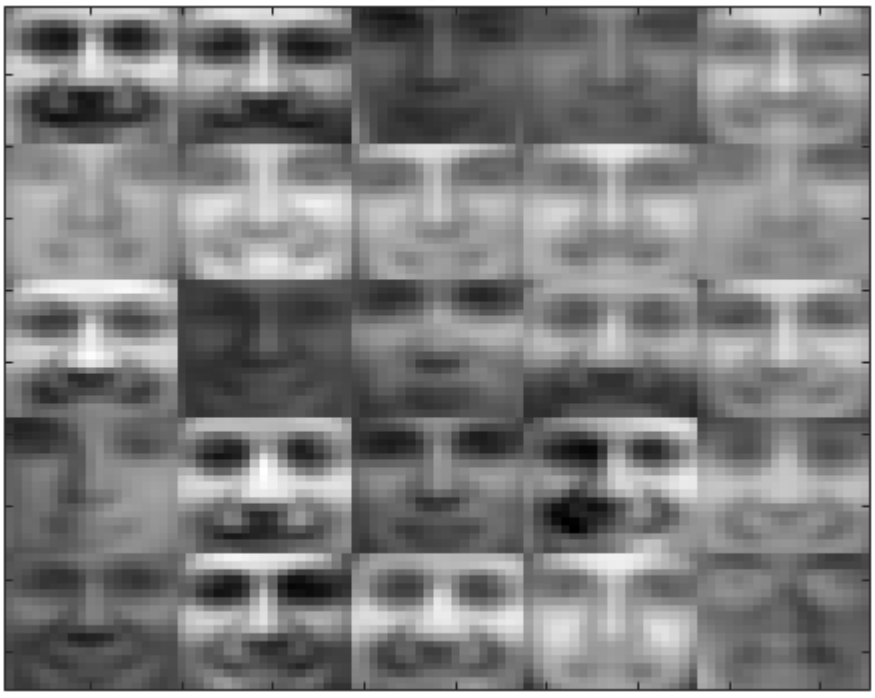


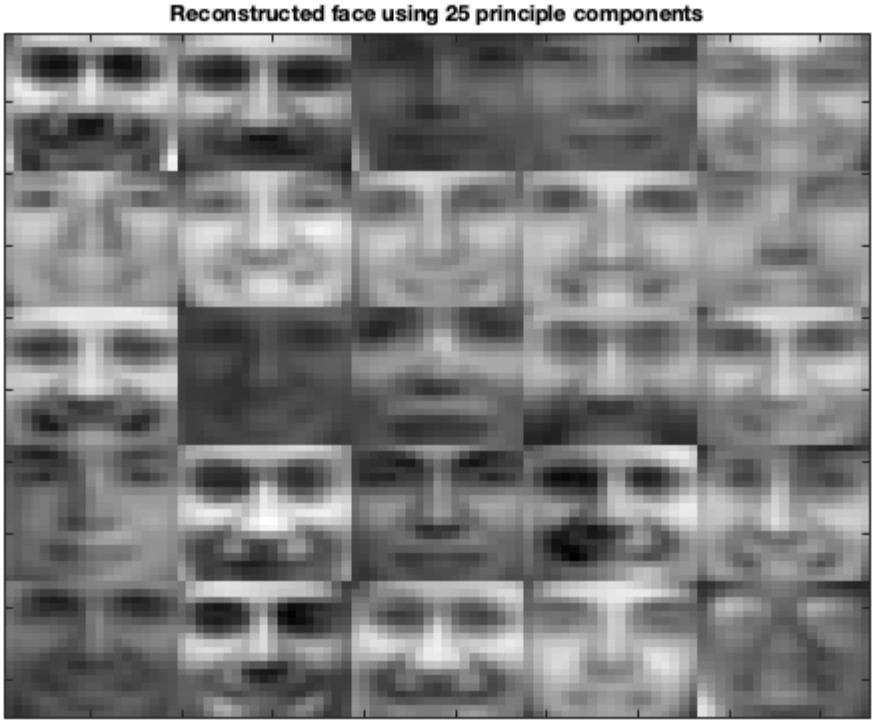


Reconstructed face using 5 principle components



Reconstructed face using 10 principle components



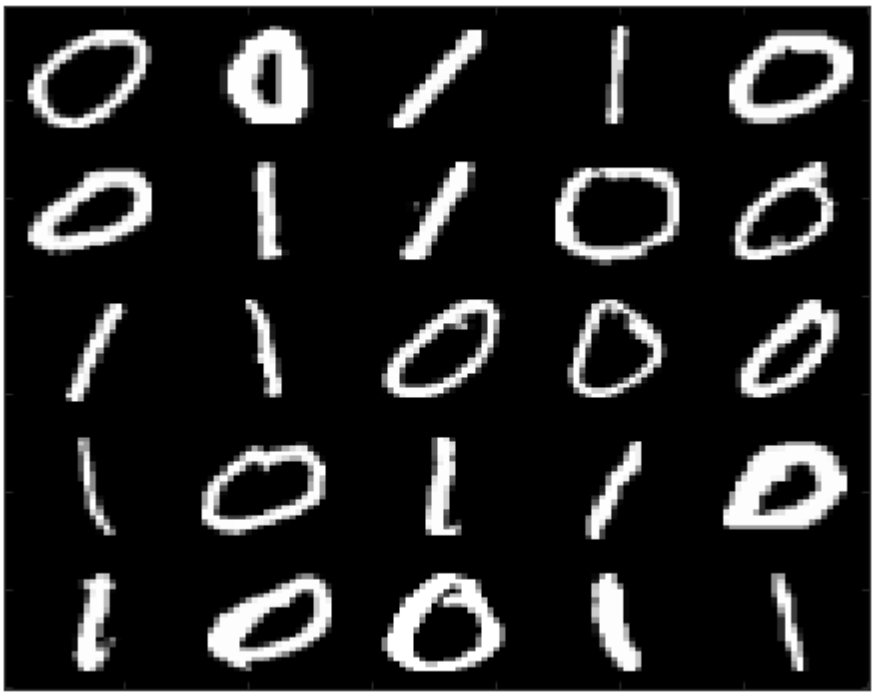


Published with MATLAB® R2020b

Contents

2.2 Same process for the MNIST handwritten digits

```
%Here X is a 784x12665 data set, for 12665 image of size 784 pixels

%(same explanation as 1.2 in cbcl.mat data)
load mnist.mat

idx = randperm(size(X,2), 25);

figure;
imgrid(X(:,idx), dims, [5 5]);
```

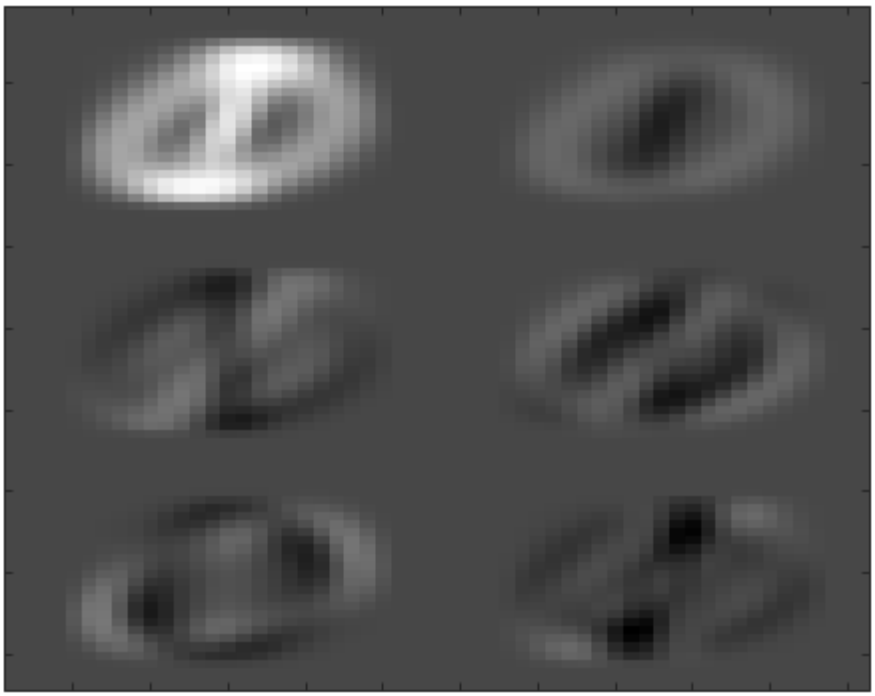


2.3.1 do PCA on data

```
%(same explanation as 1.3.1 in cbcl.mat data)
[W, Z, mu, lambda] = princomp2(X, 25);
```

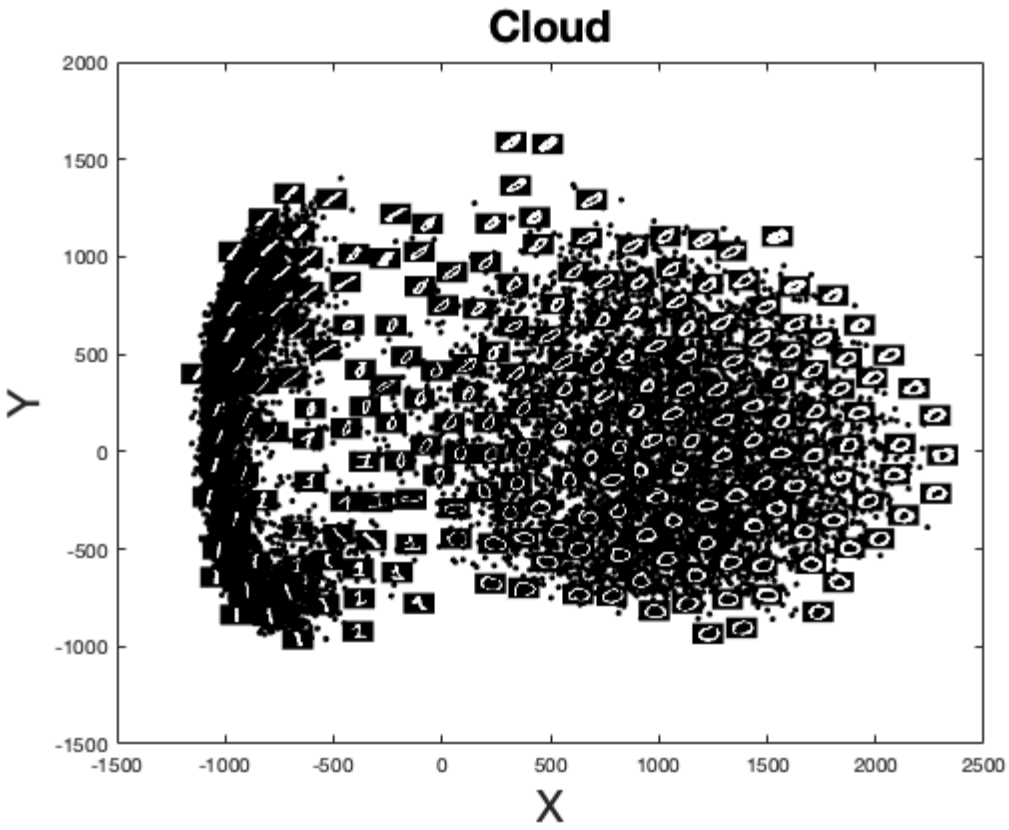
2.3.2 display first 5 principle components/"eigenfaces"

```
%(same description as 1.3 in cbcl.mat data)

%observation: the first princomp (1,2) has a clear white circle and black line
%suggesting its only making distinction between generic shape of 1 and 0.
%the second, third princomp (2,1) seem to incorporate a slanted variety of '1'
%and the fourth including a deformed version of 0. 5th princomp seem like a
%complex mixture of contribution, so difficult to guess utility.
figure;
imgrid([mu/max(mu),2*W(:,1:5)], dims, [3,2]);
```



2.4 Choosing leading 2 principle components

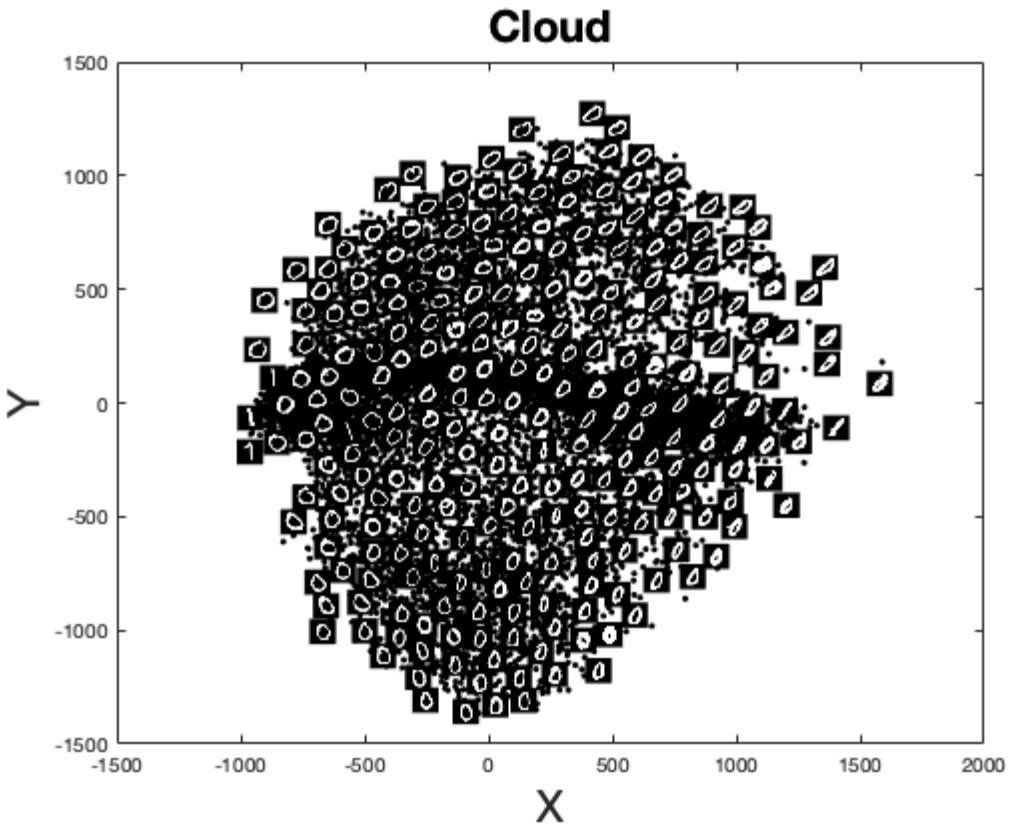
```
%(same explanation as 1.4 in cbcl.mat data)

%additional observation: note the general shape is 2 separate clouds, where
%left region is where almost all the '1' data points are, well separated with
%the right region of 0. This suggests good info retained by subspace w/ only
%2 PCA components.
figure;
imcloud(Z(1:2,:)',X,dims);
```

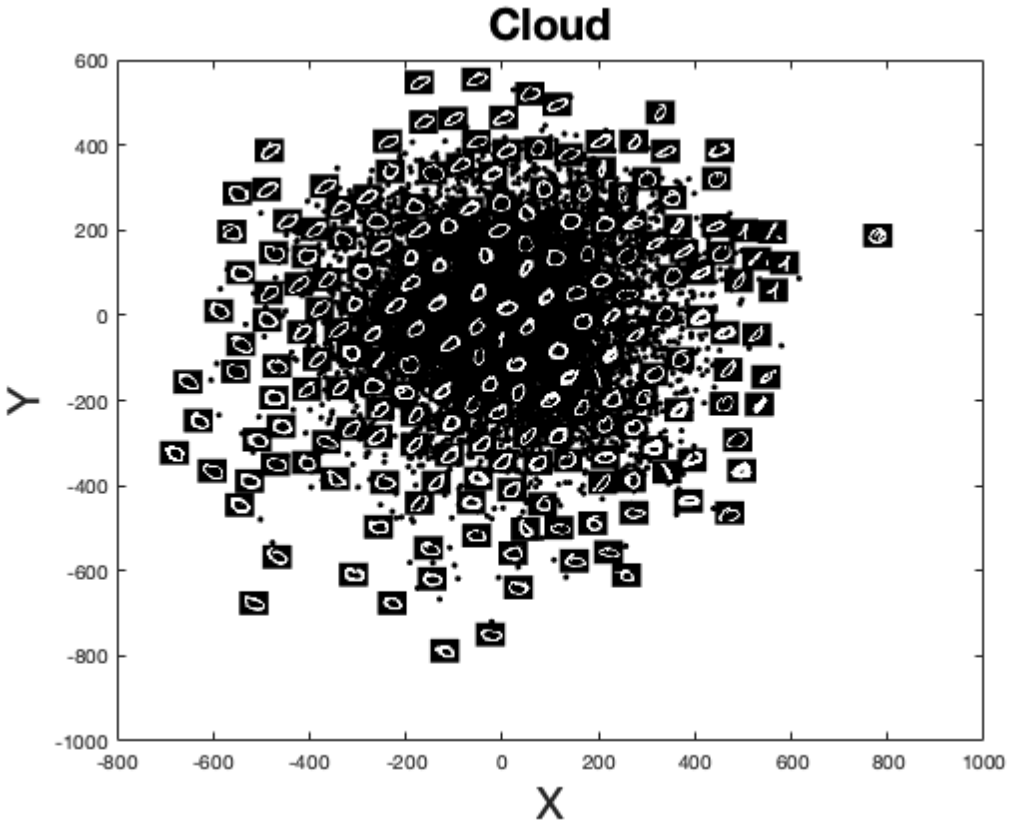


2.5 Choosing other principle components

```
%(same explanation as 1.5 in cbcl.mat data)

%observe the 1 labels is mixed with the 0 labels here, so we can't distinguish
%them at all, since we aren't using the best principle components
figure;
imcloud(Z([2 4],:)',X,dims);

figure;
imcloud(Z([19 20],:)',X,dims);
```

The latter plots seem to not be contributing much, even the outliers doesn't look informative except showing more deformed versions of 0. unlike the face one where outliers are picking potential special feature

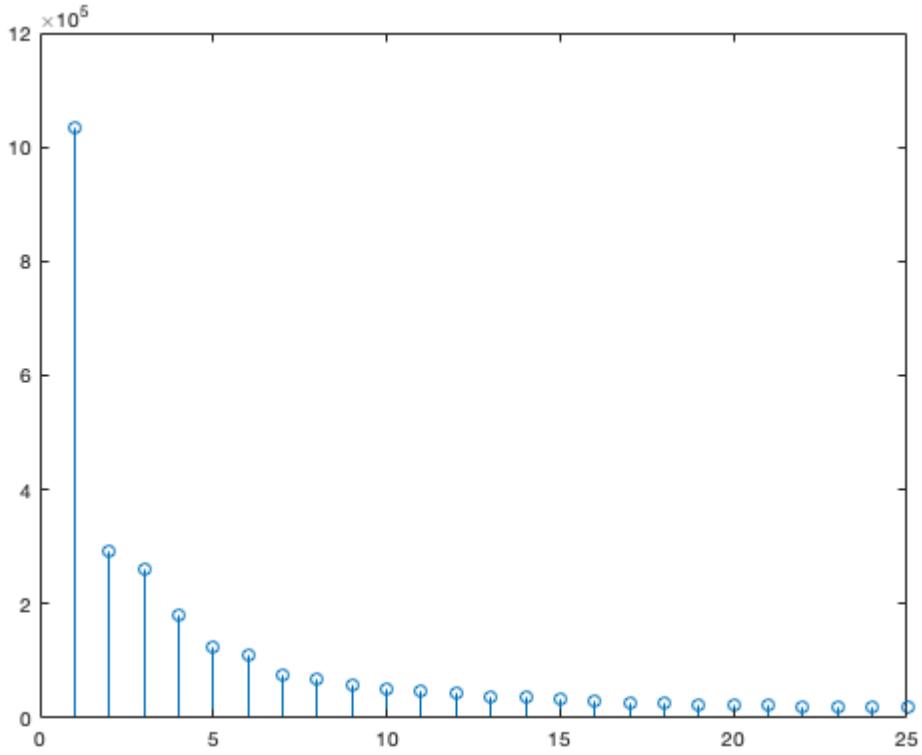




2.6 Eigenvalue plot and implication

```
%(same explanation as 1.6 in cbcl.mat data)

%for example, clearly the leading eigenvalue is very important, and the
```

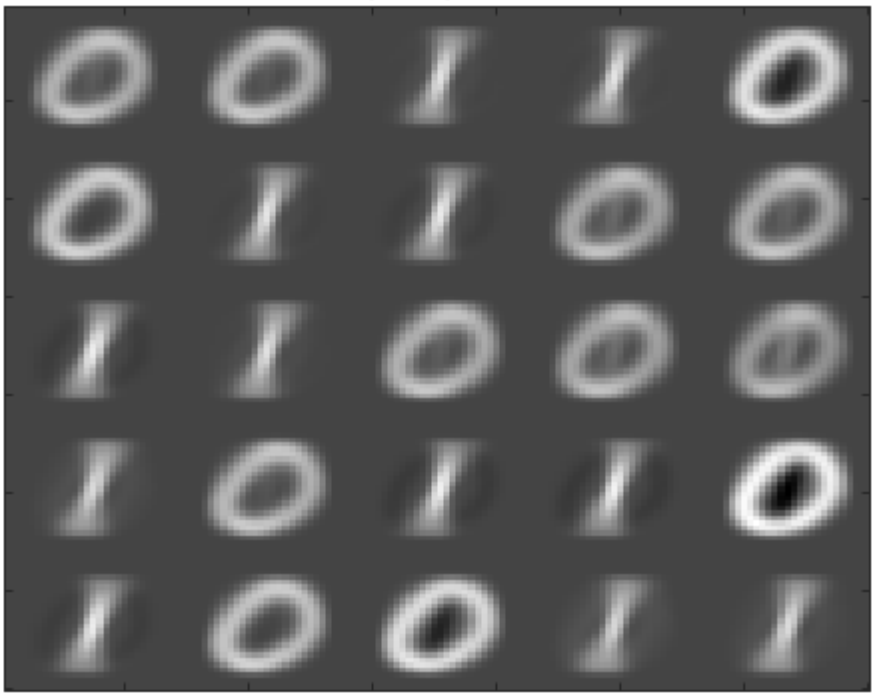
```
%next 5 are also quite important, but the rest seem to be not contributing
%significant amounts to separating the 0 and 1.
figure;
stem(lambda);
```



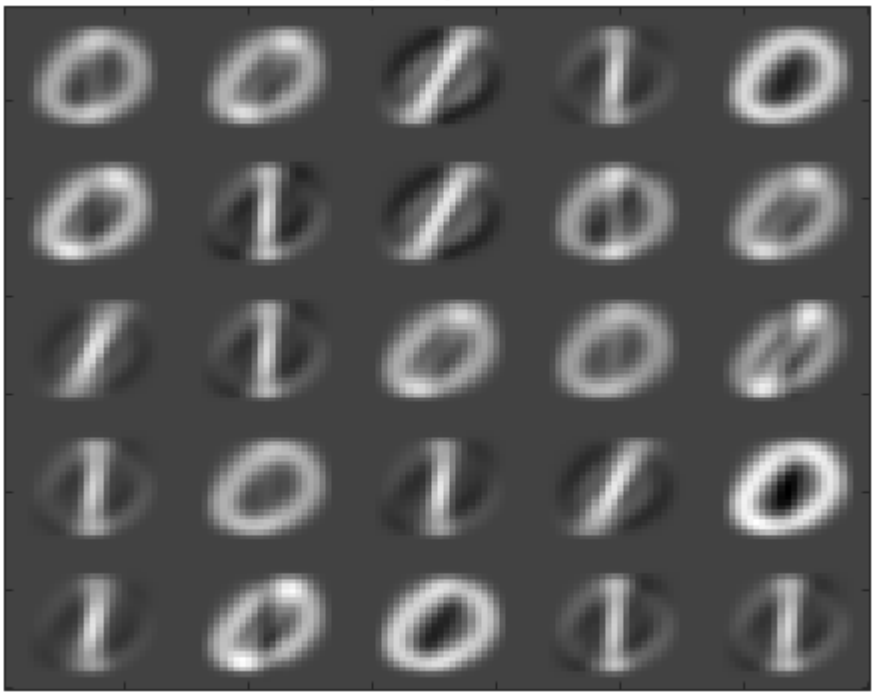
2.7 Projection of X onto W via Z

```
%(same explanation as 1.7 in cbcl.mat data)
M = [1 2 5 10 25];

%observation: Here the first princomp appears to contribute the most generic
%shape of 0's and 1's and so it can reconstruct almost all of the time the
%actual handwritten number as data with handwritten 1 and 0 should be projected
%to different signs by the princomp thanks to adhering to the generic shape,
%which when reconstructed will make clear if its 0 or 1.

%up to the 5th princomp appears to contribute to slight clockwise rotations
%and linear deformation of the 0 and 1, and up to 10th princomp will also
%record squishifications of the digits. up to 25th princomp seem to include
%also the curvature and finer, less variable details.
for i = 1:length(M)
    figure;
    imgrid(W(:,1:M(i))*Z(1:M(i),idx)+repmat(mu, [1 25]),dims,[5 5]);
    title(['Reconstructed image using ', num2str(M(i)), ' principle components']);
end

figure;
imgrid(X(:,idx), dims, [5 5]);
title('Original image (for comparison)');
```

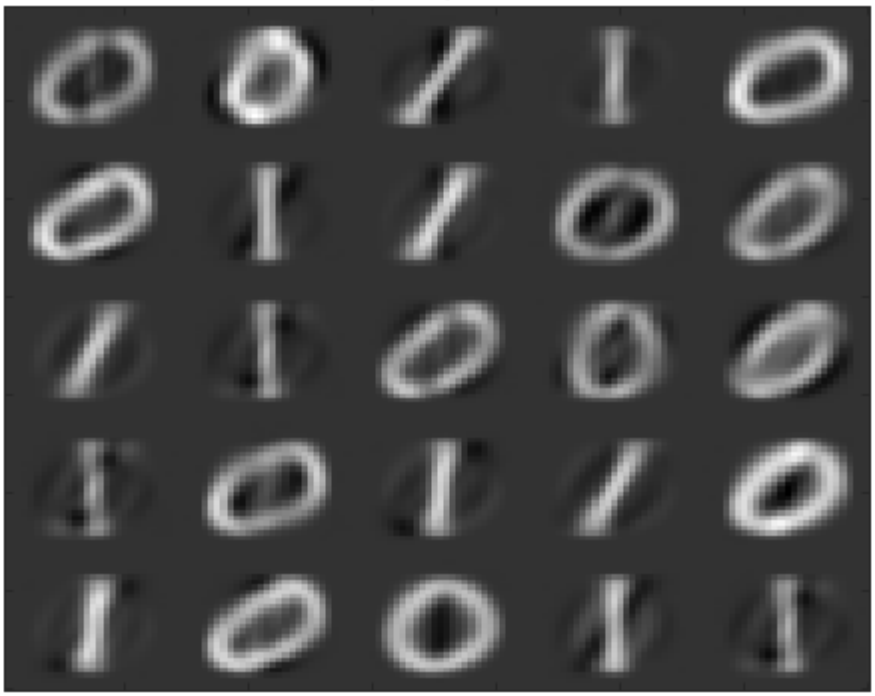
Reconstructed image using 1 principle components



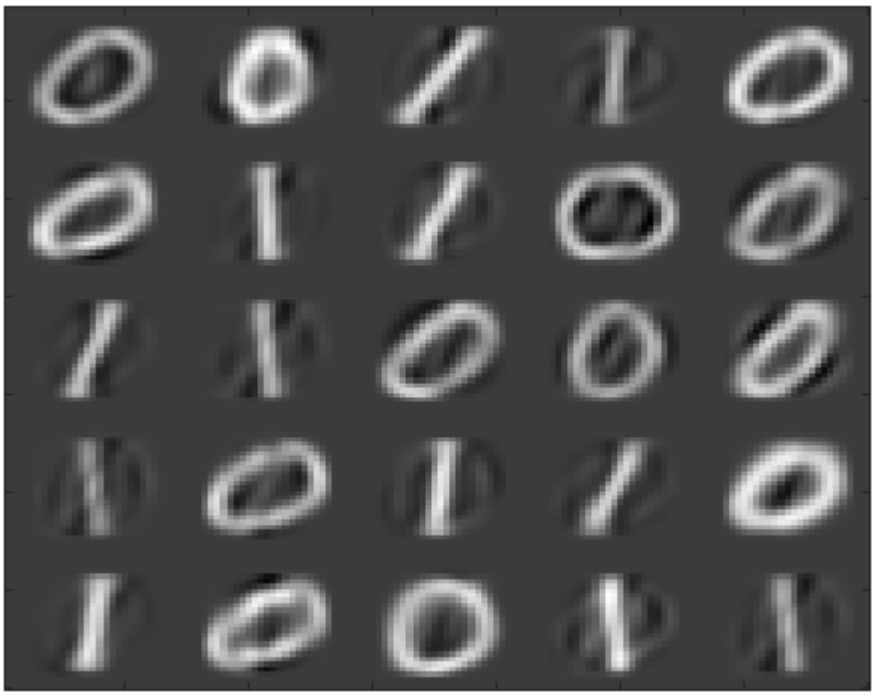
Reconstructed image using 2 principle components

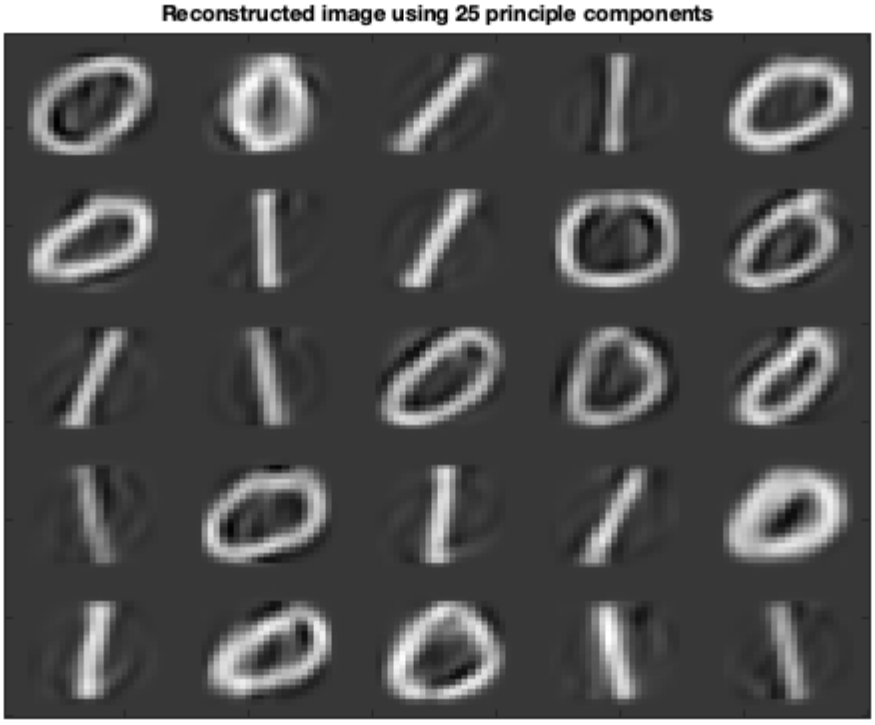


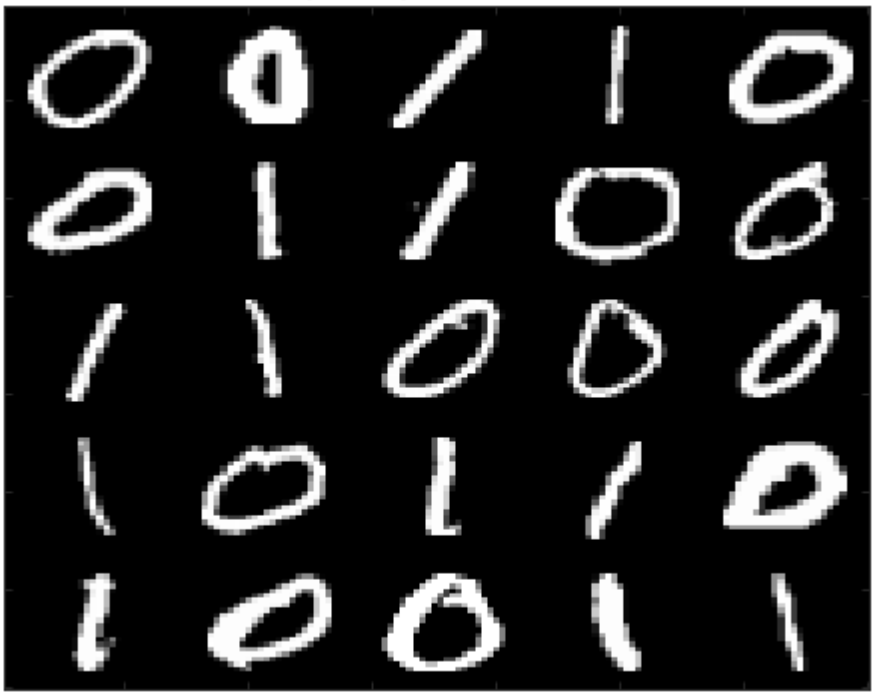
Reconstructed image using 5 principle components



Reconstructed image using 10 principle components







Published with MATLAB® R2020b