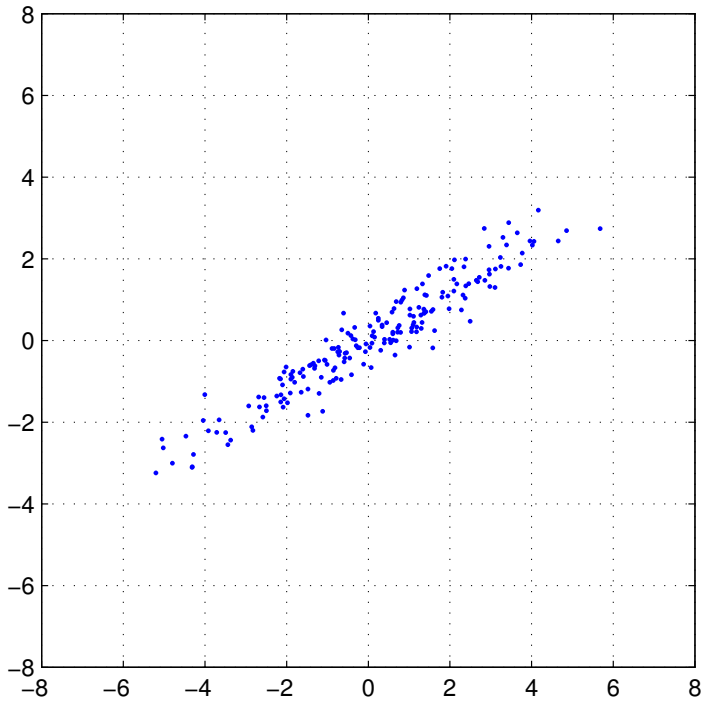
Illustration of 2D PCA

This code illustrates PCA for a 'toy' example. PCA gives a way to approximate an N-point vector by an M-point vector with M $<$ N.

Contents

Make data for example

```
L = 200;  % number of data points

% generate data for example
C = [1 2; 1 1];
X = C * randn(2,L);

% display data
plot(X(1,:),X(2,:),'b.')
grid
axis([-1 1 -1 1]*8)
axis square

% data is zero mean - no need to subtract mean first.
```



Compute PCA matrix

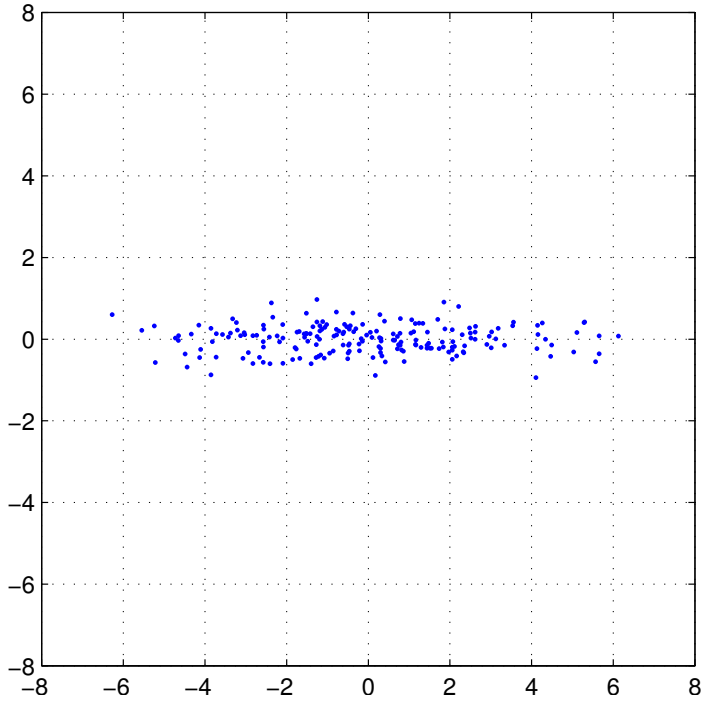
```
A = X * X';      % data covariance matrix

% compute eigenvectors and eigenvalues
[E,D] = eig(A);

% check that A = E D E'
err = A - E * D * E';
max(abs(err(:)));

% eigevalues in descending order
d = diag(D);
[tmp, k] = sort(-d);

% resort
d = d(k)
D = diag(d);
E = E(:,k);

% check that A = E D E'
err = A - E * D * E';
max(abs(err(:)));

% Set P matrix
P = E';
```

```
d =

   1.0e+03 *

    1.2962
    0.0233
```

Transform data to new coordinates

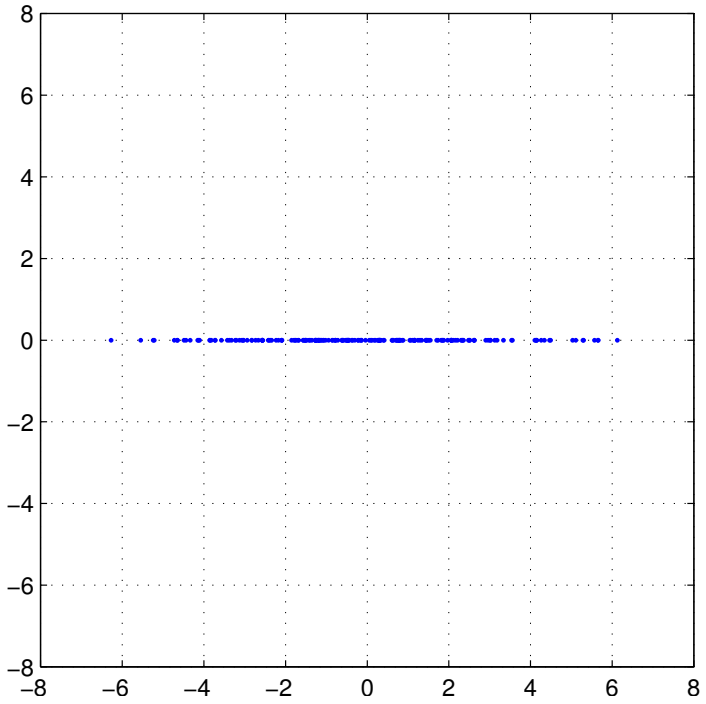
Decorrelate data

```
% create new data
Y = P * X;

% display new data
plot(Y(1,:),Y(2,:),'b.')
grid
axis([-1 1 -1 1]*8)
axis square
```



Approximate each data vector using one value

```
Y(2,:) = 0;

% display approximate data
plot(Y(1,:), Y(2,:),'b.')
grid
axis([-1 1 -1 1]*8)
axis square
```



Convert back to original coordinates

```
X_approx = P'*Y;

% display approximate data
plot(X_approx(1,:),X_approx(2,:),'b.')
grid
axis([-1 1 -1 1]*8)
axis square
```