Matlab for the Best Basis

The Best Basis in Matlab

A matrix X whose size is $n \times p$ may be thought of as an array of p points in $\mathbb{R}^n$, so we might ask how we might find a basis for this space, the column space of X.
We already know one method: Gram-Schmidt (or QR).
The best basis is given by the first k eigenvectors of U or V (depends on whether your data is in columns or rows). Assuming the SVD has been computed, then here are some common tasks, where we assume that X is $n \times p$, with p points in $\mathbb{R}^n$.

- The low dimensional representation of the data using the first k columns of V is the set of the k coordinates (one set of coordinates per vector):

  ```
  Coords=U(:,1:k)'*X;
  ```

  Note that the dimensions match up appropriately. To get these data points represented back in $\mathbb{R}^n$,

  ```
  Recon=U(:,1:k)*Coords;
  ```

  This is the same as: `U(:,1:k)*U(:,1:k)'*X`, which shows that it is the orthogonal projection of X into the space spanned by the first k columns of U.

Example: A Movie

The data was taken from a short webcam movie. It consists of 109 frames, each frame is $120 \times 160 = 19,200$ pixels. Therefore, the data is given to you as a matrix that is 19200×109 matrix. The "best basis" for movie space should be some set of vectors in $\mathbb{R}^{19200}$ (so that each basis vector can be visualized as a movie frame).
Go to the class website and download `author.mat`, open up your editor and type the following:

```
load author.mat
Y1=double(Y1);  %The original data is unsigned integer format
imagesc( reshape(Y1(:,1),120,160) );
colormap(gray);
```

If you're curious, you can watch the movie in Matlab:

```
for j=1:109
imagesc(reshape(Y1(:,j),120,160));
colormap(gray);
pause(0.1);
end
```

Updated Copy: Here are the class notes I was referring to in the original text Here is the Matlab code:

```
%% Mean Image

Ym=mean(Y1,2);                %The mean is 19200 x 1 (so it is a photo)
Y=Y1-repmat(Ym,1,109);  %Subtract the mean from each of the 109 frames

figure
imagesc(reshape(Ym,120,160));  %Let's look at the mean as a photo
title('Mean Image from the Movie');
colormap(gray);

% Watch the mean subtracted movie!  It is kind of funky!
% (Copy and paste the appropriate code and change Y1 to Y)

%% Bases for the Fundamental Subspaces:

[U,S,V]=svd(Y,'econ');  %The SVD!

% Best two dimensional representation of the 109 frames:
Coeffs=U(:,1:2)'*Y;           %These are the coordinates of the data with respect
                        % to the basis vectors in U-  So Coeffs is 2 x 109
                        % for 109 two dimensional coordinates.

Recon=U(:,1:2)*Coeffs;    %This is the result of projecting the data into the first
                        % two columns of U.
                        % Recon is 19200 x 109

% Plot in two dimensions:
plot(Coeffs(1,:),Coeffs(2,:),'r*-');
title('Two-Dim representation of the data- Points are movie frames')


% Watch the 2 dimensional version of the movie (Be sure to add
% the mean back in!)
```

Appendices: Matlab Commands Used

Miscellaneous Image Commands

We will be treating an image that has $m \times n$ pixels as a vector in $\mathbb{R}^{mn}$. Matlab has some easy commands to do conversions and visualizations. Let A be an image in $\mathbb{R}^{m \times n}$:

- `v=A(:);` Converts matrix A to vector $\mathbf{v}$.
- `B=reshape(v,m,n);` Converts $\mathbf{v}$ back to an $m \times n$ array B.
- `imagesc(A)` Short for "image scale", this results in an $m \times n$ array of colors, where each color is from the value in A. It is a good idea to use `imagesc` rather than the non-scaled version, `image`, unless you know what you're doing.

  You might try the following example, which also shows you how to use some built-in colorings. The colorbar command is useful to give you a color legend.

  ```
  load clown
  imagesc(X)
  colormap(summer)
  colorbar
  ```

  Type `doc colormap` to see all of the built-in options.

Some data plots

If you have an array of data, sometimes it is useful to plot certain coordinates as one type, and other coordinates as another type. For example, try typing the following. The first two lines just build a data set in a matrix X:

```
X1=rand(100,2); X2=rand(100,2)+4;
X=[X1;X2];
index01=1:100;
index02=101:200;
plot(X(index01,1),X(index01,2),'r.',X(index02,1),X(index02,2),'b^');
```