University of Washington
AMATH 301 Spring 2017
Instructor: Dr. King-Fai Li

Lecture Notes

Week 9

Singular value decomposition (SVD)

Any given 2D (real) matrix $\mathbf{A}$ can be factorized into a product of three 2D matrices:

$$\mathbf{A}_{m\times n} = \mathbf{U}_{m\times n}\mathbf{\Sigma}_{n\times n}\mathbf{V}^T_{n\times n} = \begin{bmatrix} | & & | \\ \mathbf{u}_1 & \cdots & \mathbf{u}_n \\ | & & | \end{bmatrix}\begin{bmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_n \end{bmatrix}\begin{bmatrix} | & & | \\ \mathbf{v}_1 & \cdots & \mathbf{v}_n \\ | & & | \end{bmatrix}^T$$

$\mathbf{\Sigma}$ is a non-negative, real diagonal matrix; σ are called singular values. The columns of $\mathbf{U}$ and $\mathbf{V}$ are the left and right singular vectors, respectively. $\mathbf{U}$ and $\mathbf{V}$ are unitary such that $\mathbf{U}\mathbf{U}^T = \mathbf{U}^T\mathbf{U} = \mathbf{I}$ and $\mathbf{V}\mathbf{V}^T = \mathbf{V}^T\mathbf{V} = \mathbf{I}$. $\mathbf{U}$ and $\mathbf{V}$ are the eigenvectors of $\mathbf{A}\mathbf{A}^T$ and $\mathbf{A}^T\mathbf{A}$:

$$\begin{aligned} \mathbf{A}\mathbf{A}^T &= \left(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\right)\left(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\right)^T \\ &= \mathbf{U}\mathbf{\Sigma}\left(\mathbf{V}^T\mathbf{V}\right)\mathbf{\Sigma}^T\mathbf{U}^T & \because\ \left(\mathbf{A}\mathbf{B}\right)^T = \mathbf{B}^T\mathbf{A}^T \\ &= \mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^T & \because\ \mathbf{V}^T\mathbf{V} = \mathbf{I} \text{ and } \mathbf{\Sigma}^T = \mathbf{\Sigma} \end{aligned}$$

$$\begin{aligned} \mathbf{A}^T\mathbf{A} &= \left(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\right)^T\left(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\right) \\ &= \mathbf{V}\mathbf{\Sigma}^T\left(\mathbf{U}^T\mathbf{U}\right)\mathbf{\Sigma}\mathbf{V}^T \\ &= \mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^T & \because\ \mathbf{U}^T\mathbf{U} = \mathbf{I} \end{aligned}$$

Given an $m \times n$ matrix $\mathbf{A}$, the SVD of $\mathbf{A}$ can be obtained by

```
[U,S,V]=svd(A,'econ');
```

Note:

1. If $\mathbf{A}$ is complex, then replace the transpose by conjugate transpose, e.g. $\mathbf{A}^T \to \mathbf{A}^*$.
2. If the option `'econ'` is not turned on, then the decomposition will take the form $\mathbf{A}_{m\times n} = \mathbf{U}_{m\times m}\mathbf{\Sigma}_{m\times n}\mathbf{V}^T_{n\times n}$. This may be slower if $m > n$.

Principal components analysis (PCA)

Any given 2D matrix $\mathbf{A}$ can be factorized into a product of two 2D matrices:

$$\mathbf{A}_{m\times n} = \mathbf{Y}_{m\times n}\mathbf{V}_{n\times n}^T$$

where the columns of $\mathbf{Y}$ are the principal components and the columns of $\mathbf{V}$ are the eigenvectors of the covariance matrix of $\mathbf{A}$. By such definition, $\mathbf{V}$ satisfies

$$\mathbf{A}^T\mathbf{A} = \mathbf{V}\mathbf{S}\mathbf{V}^{-1}$$

Therefore, $\mathbf{V}$ are just the right singular vectors of $\mathbf{A}$ and $\mathbf{S} = \mathbf{\Sigma}^2$. Then $\mathbf{Y} = \mathbf{A}\mathbf{V} = \mathbf{U}\mathbf{\Sigma}$.

Given an $m \times n$ matrix $\mathbf{A}$, the PCA of $\mathbf{A}$ can be obtained by

```
[V,Y,S2]=pca(A,'Centered','off');
```

where `S2=S.^2` (up to some multiplicative constants).

Note:

1. Prof. Kutz defined $\mathbf{A}_{m\times n} = \mathbf{U}_{m\times n}\mathbf{Y}_{n\times n}$. To follow his definition, then

   ```
   [U,Yt,S2]=pca(A','Centered','off'); Y=Yt';
   ```

   This form is used in Homework 5.

2. If the option `'Centered'` is not turned off, then Matlab will subtract the row mean by default, which sometimes will lead to confusions and incorrect conclusions. A good practice is to always remove the mean (whether row-wise or column-wise) yourself, and run `pca` with `'Centered'` turned off.

Multidimensional datasets

If $\mathbf{A}$ has dimensions $m \times n_1 \times \cdots \times n_p$, then reshape $\mathbf{A}$ such that it has dimensions $m \times n$, where $n = n_1 n_2 \cdots n_p$.

e.g. For 3-D data sets

```
[m,n1,n2]=size(A); A=reshape(A,[m,n1*n2]);
```

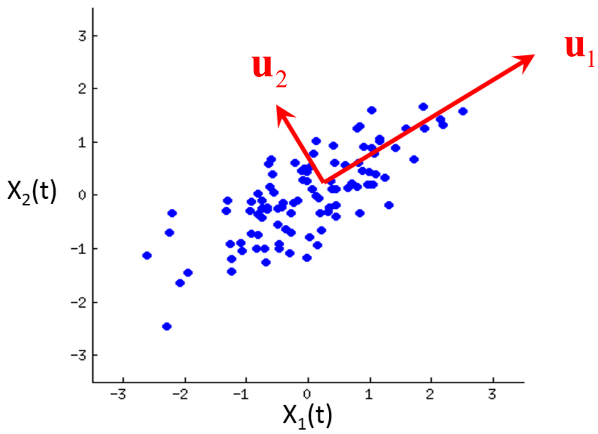
Then SVD/PCA can be applied to the reshaped $\mathbf{A}$. After the calculation, reshape $\mathbf{A}$, $\mathbf{U}$, $\mathbf{V}$, and/or $\mathbf{Y}$ back to the original dimensions.

Note: To avoid too many zero singular values, $\mathbf{A}$ should be reshaped such that $m \approx n$.

Physical Meaning
For the sake of illustration, consider two time series at two different locations

$$\mathbf{x}_1 = \begin{bmatrix} x_1(t_1) & x_1(t_2) & \cdots & x_1(t_n) \end{bmatrix}$$
$$\mathbf{x}_2 = \begin{bmatrix} x_2(t_1) & x_2(t_2) & \cdots & x_2(t_n) \end{bmatrix}$$

These two time series, however, may be correlated. To show this, make the scattered plot:



This graph suggests that $\mathbf{x}_1$ and $\mathbf{x}_2$ are NOT independent. Most of the variations can be adequately described in the principal direction $\mathbf{u}_1$. There may be small variations in the other direction $\mathbf{u}_2$ but the $\mathbf{u}_1$ essentially captures everything. Thus

1. If the new coordinate system is defined using the principal axes $\mathbf{u}_1$ and $\mathbf{u}_2$, then the above dataset is essentially 1-D.

2. If the new coordinate system, then the data points are uncorrelated in the new coordinate system, i.e. the covariance matrix becomes *diagonalized*.

It is thus more convenient to describe the data using $\mathbf{u}_1$ and $\mathbf{u}_2$.

More generally, if there are *m* time series at different locations, form the data matrix

$$\mathbf{A} = \begin{bmatrix} \longleftarrow & \mathbf{x}_1 & \longrightarrow \\ \longleftarrow & \mathbf{x}_2 & \longrightarrow \\ & \vdots & \\ \longleftarrow & \mathbf{x}_m & \longrightarrow \end{bmatrix}$$

where each $\mathbf{x}_j$ is a column vector $\mathbf{x}_j = \begin{bmatrix} x_j(t_1) & x_j(t_2) & \cdots & x_j(t_n) \end{bmatrix}$, then the principal axes $\mathbf{u}$ are the left singular vectors of $\mathbf{A}$.

Example 1: Approximation of a Black-White Photo

```
clear all; close all;
pic = imread('taylor03.jpg');  % pic consists of 8-bit integers
pic = squeeze(pic(:,:,1)); % black-white photo
imshow(pic)

% Do an SVD; SVD only accepts double numbers
[U,S,V]=svd(double(pic),'econ');

% Add back the singular components one by one
% to approximate the original photo
for j=1:length(S)
  pic1 = U(:,1:j)*S(1:j,1:j)*V(:,1:j)';
  imshow(uint8(pic1))  % imshow only accepts intergers
  title(['j=' num2str(j)]);
  pause
end
```

$$\tilde{P}_j(x,y)=\sum_{k=1}^{j}\sigma_k U_k(x) V_k^T(y)$$

```
% SVD and PCA are equivalent
[V,Y,S2]=pca(double(pic),'Centered','off');
for j=1:length(S2)
  pic1 = Y(:,1:j)*V(:,1:j)';
  imshow(uint8(pic1))
  title(['j=' num2str(j)]);
  pause
end
```

$$\tilde{P}_j(x,y)=\sum_{k=1}^{j}Y_k(x) V_k^T(y)$$

Example 2: Approximation of a Color Photo

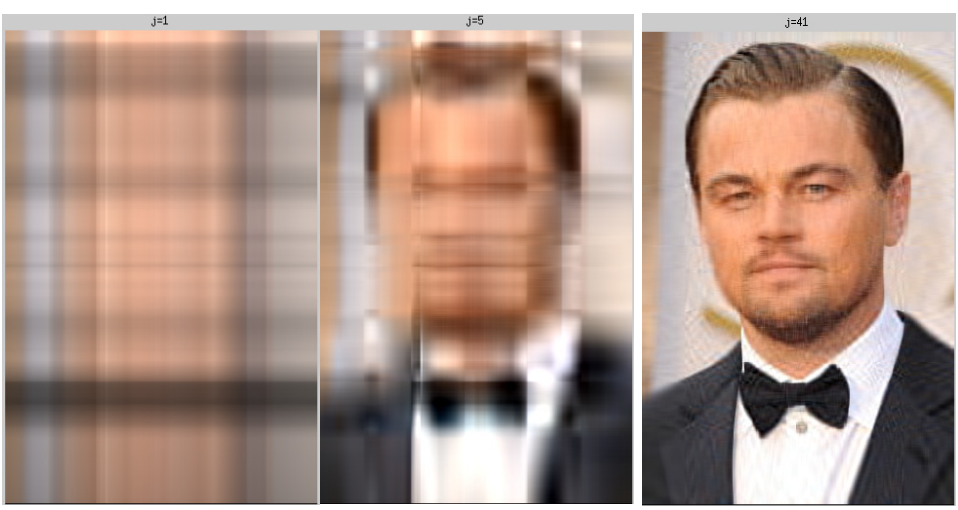
```
clear all; close all;
pic = imread('leonardo.jpg');
imshow(pic)
%pic(:,:,2:3)=0;imshow(pic);     % See the red image
%pic(:,:,[1 3])=0;imshow(pic);   % See the green image
%pic(:,:,1:2)=0;imshow(pic);     % See the blue image

% The color photo has the third dimension in RGB
% Reshape the 3D picture into a 2D one by merging
% the third dimension into the second.
pic = reshape(pic,486,320*3);

% PCA analysis in Prof. Kutz's way
[U,Yt,S2]=pca(double(pic'),'Centered','off'); Y=Yt';

% Add back the singular components one by one
% to approximate the original photo
for j=1:length(S2)
  pic1 = U(:,1:j)*Y(1:j,:);
  % Get the RGB dimension back
  pic1 = reshape(pic1,486,320,3);
  imshow(uint8(pic1))
  title(['j=' num2str(j)]);
  pause
end
```

$$\tilde{P}_j(x,y)=\sum_{k=1}^{j}U_k(x)Y_k(y)$$



Example 3: Face recognition — Decomposition of Multiple Photos

Homework 5, Q1.