Principal Component Analysis (PCA)

DSA5103 Lecture 1

Yangjing Zhang

12-Jan-2023

NUS

Today's content

1. Recap linear algebra (eigenvalues/eigenvectors) and statistics (mean/covariance)
2. PCA

Preliminaries

Notation

- $\mathbb{S}^n$: the space of $n \times n$ symmetric matrices.
- $\mathbb{S}^n_+$: the space of $n \times n$ symmetric and positive semidefinite matrices (all eigenvalues ≥ 0).
- Let A be a matrix. $A_{i\cdot}$ denotes the i-th row of A, and $A_{\cdot j}$ denotes the j-th column of A.
  - ▷ For example, given $A = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \end{bmatrix}$. Then $A_{\cdot 1} = \begin{bmatrix} 1 \\ 6 \\ 11 \end{bmatrix}$, and $A_{1\cdot} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \end{bmatrix}$.

Eigenvalues/eigenvectors

- The eigenvalue decomposition of $A \in \mathbb{S}^n$ is given by

$$A = Q\Lambda Q^T = \begin{bmatrix} | & & | \\ Q_{\cdot 1} & \cdots & Q_{\cdot n} \\ | & & | \end{bmatrix} \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} \begin{bmatrix} | & & | \\ Q_{\cdot 1} & \cdots & Q_{\cdot n} \\ | & & | \end{bmatrix}^T$$

where Q is an orthogonal matrix whose columns are eigenvectors of A, Λ is a diagonal matrix with eigenvalues of A on the diagonal.

Eigenvalues/eigenvectors: An example

Consider $A = \begin{bmatrix} 1.5 & 0.5 \\ 0.5 & 1.5 \end{bmatrix}$.

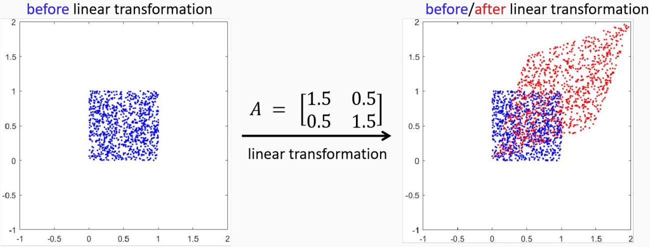
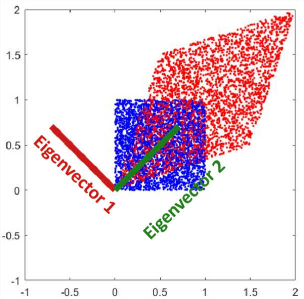


Figure 1: Blue points are $x_i \in \mathbb{R}^2$, $i = 1, \dots, 1000$ randomly drawn from $[0, 1] \times [0, 1]$. Red points are $y_i = Ax_i$ after linear transformation.

Eigenvalues/eigenvectors: An example

- Eigenvalue decomposition:

$$A = \underbrace{\begin{bmatrix} -0.7071 & 0.7071 \\ 0.7071 & 0.7071 \end{bmatrix}}_{\text{eigenvector 1} \quad \text{eigenvector 2}} \begin{bmatrix} 1 & \\ & 2 \end{bmatrix} \begin{bmatrix} -0.7071 & 0.7071 \\ 0.7071 & 0.7071 \end{bmatrix}^T.$$



- ▷ The linear transformation results in a scaling of 2 along eigenvetor 2.
- ▷ The linear transformation results in a scaling of 1 along eigenvetor 1.

- The linear transformation results in a scaling of λ along the eigenvector associated with λ.

Eigenvalues/eigenvectors: An example

Listing 1: Matlab codes implementing the linear transformation

```
n = 1000; p = 2;
X = rand(p,n);
plot(X(1,:),X(2,:),'b.');
hold on; axis square; xlim([-1,2]); ylim([-1,2]);
A = [1.5 0.5; 0.5 1.5]; % A = [1.25 0.75; 0.75 1.25];
Y = A*X; % linear transformation
plot(Y(1,:),Y(2,:),'r.');
[Q,D] = eig(A); % eigenvalue decomposition
plot([0; Q(1,1)],[0; Q(2,1)],'-','Color',[0.8,0.1,0.1],...
    'LineWidth',6);
plot([0; Q(1,2)],[0; Q(2,2)],'-','Color',[0.1,0.5,0.1],...
    'LineWidth',6);
```

Exercise

Use another matrix A, e.g., $\begin{bmatrix} 1.25 & 0.75 \\ 0.75 & 1.25 \end{bmatrix}$. What is your observation?

Mean/covariance

Let $x_1, \ldots, x_n \in \mathbb{R}^p$ be n observations of a random variable x.

- Mean vector:

$$\mu = \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i.$$

- (Sample/Empirical) Covariance matrix:

$$\Sigma = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)(x_i - \mu)^T.$$

  - ▷ Recall: Covariance matrices are symmetric and positive semidefinite.

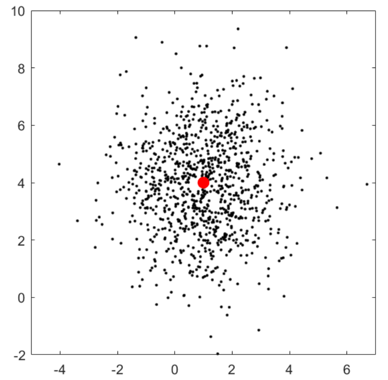
- Standard deviation (for $p = 1$):

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2}.$$

Mean/covariance: An example

Mean $\mu = \begin{bmatrix} 1 \\ 4 \end{bmatrix}$, covariance $\Sigma = \begin{bmatrix} 2 & \\ & 3 \end{bmatrix}$. We generate 1000 observations of a Gaussian random variable

$$x_i \sim \mathcal{N}(\mu, \Sigma), \quad i = 1, \dots, 1000.$$



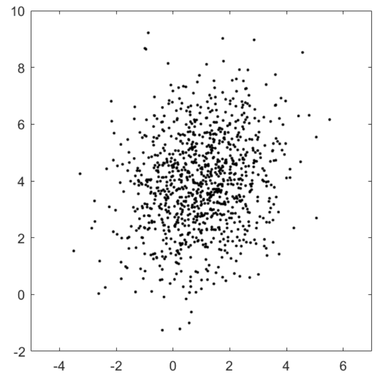

```
mu = [1;4];
Sigma = [2 0; 0 3];
for i = 1:1000
    xi = mvnrnd(mu,Sigma);
    plot(xi(1),xi(2),'k.');
    hold on;
end
```

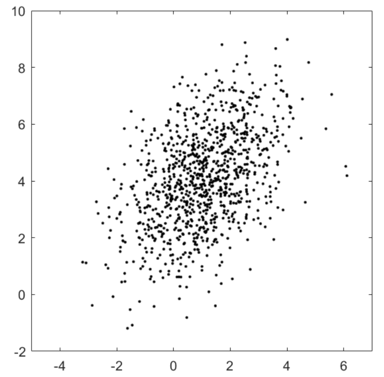
- ▷ We can see that the data is centered at mean $\mu = \begin{bmatrix} 1 \\ 4 \end{bmatrix}$

Mean/covariance: An example

$$\mu = \begin{bmatrix} 1 \\ 4 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 2 & 0.5 \\ 0.5 & 3 \end{bmatrix}$$

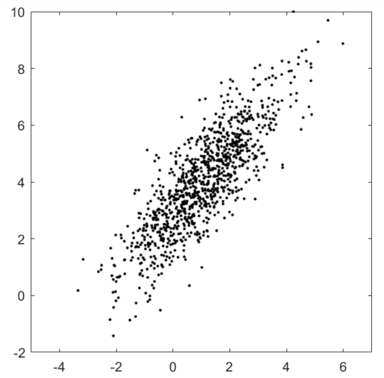
$$\mu = \begin{bmatrix} 1 \\ 4 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix}$$



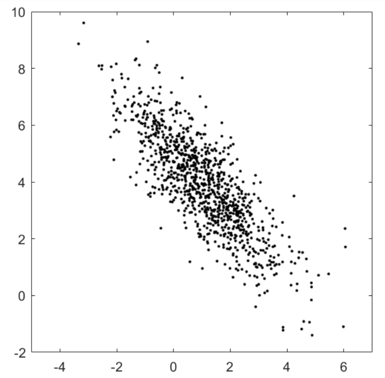


Mean/covariance: An example

$$\mu = \begin{bmatrix} 1 \\ 4 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 2 & 2 \\ 2 & 3 \end{bmatrix}$$



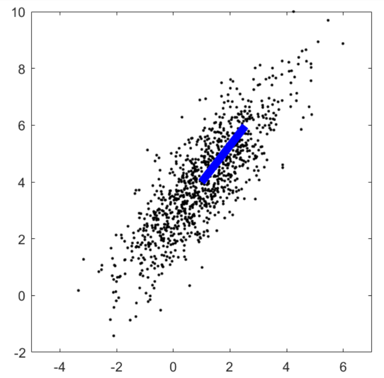
$$\mu = \begin{bmatrix} 1 \\ 4 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 2 & -2 \\ -2 & 3 \end{bmatrix}$$



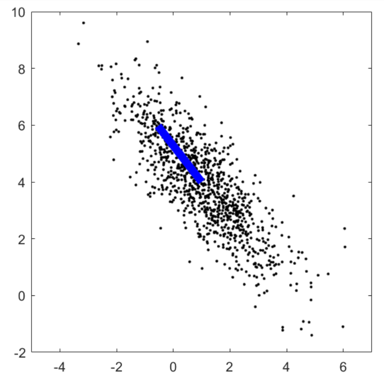
Can you find a vector that would approximate the data?

Mean/covariance: An example

$$\mu = \begin{bmatrix} 1 \\ 4 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 2 & 2 \\ 2 & 3 \end{bmatrix}$$



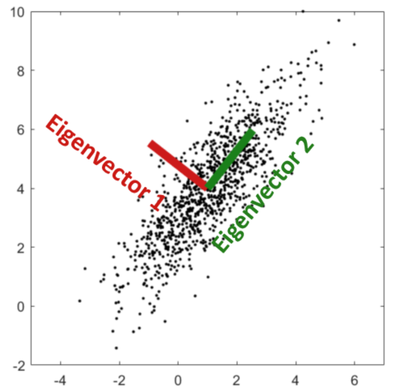
$$\mu = \begin{bmatrix} 1 \\ 4 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 2 & -2 \\ -2 & 3 \end{bmatrix}$$



Can you find a vector that would approximate the data? Blue one!

PCA: Intuition

$\mu = \begin{bmatrix} 1 \\ 4 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 2 & 2 \\ 2 & 3 \end{bmatrix}$ Consider eigenvectors of covariance matrix Σ



- $\lambda = 0.4382$

  eigenvector 1 $= \begin{bmatrix} -0.7882 \\ 0.6154 \end{bmatrix}$
- $\lambda = 4.5616$

  eigenvector 2 $= \begin{bmatrix} 0.6154 \\ 0.7882 \end{bmatrix}$
- eigenvector 2 accounts for most of the data

Principal component analysis

PCA

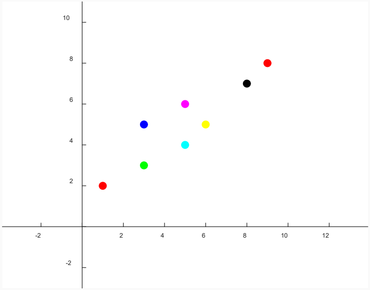
- PCA is often used to reduce the dimensionality of large data sets while preserving as much information as possible
- PCA allows us to identify the principal directions in which the data varies

Next, we use an example to demonstrate PCA step by step.

Example 1

Consider a data set that consists of $p = 2$ variables on $n = 8$ samples

$(1, 2), (3, 3), (3, 5), (5, 4), (5, 6),$
$(6, 5), (8, 7), (9, 8).$



PCA: step-be-step explanation

- **Input:** an $n \times p$ matrix X, each row is an observation or sample, each column is a predictor variable
- **Step 1: standardization** of each column of X — transform all variables to the same scale

$$X_{\cdot j} = \frac{X_{\cdot j} - \text{mean}(X_{\cdot j})}{\text{standard deviation}(X_{\cdot j})}$$

```
X = [1,2; 3,3; 3,5; 5,4; 5,6; 6,5; 8,7; 9,8]; % input
[n,p] = size(X);
% standardize each column
for j = 1:p
    X(:,j) = (X(:,j) - mean(X(:,j)))/std(X(:,j));
end
```

PCA: step-be-step explanation

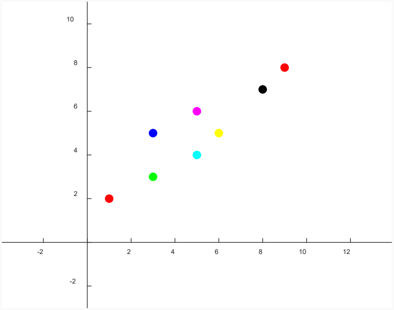


Figure 2: Input data

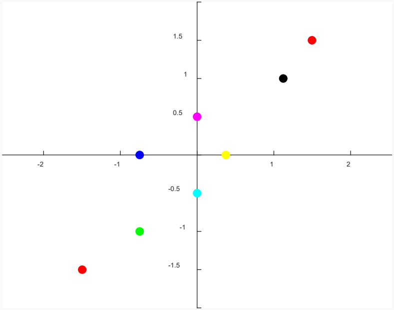


Figure 3: Standardized data

PCA: step-be-step explanation

- **Step 2: $p \times p$ covariance matrix computation** — understand how the variables of the data are varying from the mean with respect to each other

```
Sigma = cov(X); % compute covariance matrix
```

which yields

```
Sigma =

    1.0000    0.9087
    0.9087    1.0000
```

PCA: step-be-step explanation

- **Step 3: eigenvalue decomposition** — by ranking eigenvectors in order of their eigenvalues, **highest to lowest**, one obtains the principal components in order of significance

```
[Q,D] = eig(Sigma); % eigenvalue decomposition
[d,ind] = sort(diag(D),'descend'); % sort eigenvalues
D = D(ind,ind); % reorder diagonal of D (eigenvalues)
Q = Q(:,ind); % reorder columns of Q (eigenvectors)
```

which yields

```
Q =

   -0.7071    0.7071
   -0.7071   -0.7071

D =

    1.9087         0
         0    0.0913
```

- ▷ The first eigenvalue of 1.9087 is larger than the second eigenvalue of 0.0913
- ▷ The first eigenvector captures 95% $\left(\frac{1.9087}{1.9087+0.0913}\right)$ of the information
- ▷ The second eigenvector captures 5% $\left(\frac{0.0913}{1.9087+0.0913}\right)$ of the information

PCA: step-be-step explanation

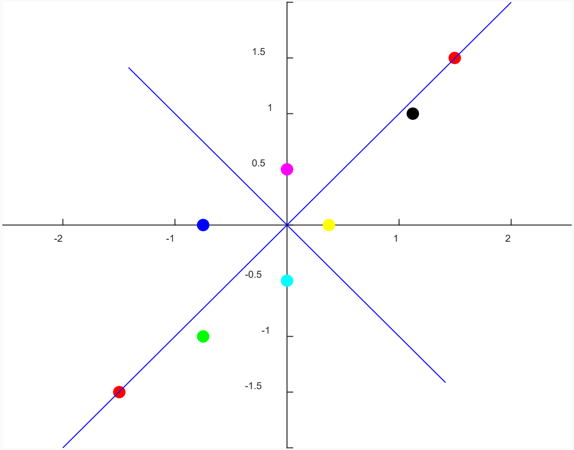


Figure 4: Principal directions. (-0.7071, -0.7071) carries 95% of the information, and (0.7071, -0.7071) carries 5% of the information

PCA: step-be-step explanation

- **Step 4: recast the data** along the principal components axes — the jth column of X_{new} is the jth principal component

$$X_{\text{new}} = XQ$$

```
Xnew = X*Q; % new data
```

It is up to you to choose whether

- ▷ to keep all the components
- ▷ or discard the ones of lesser significance

There are some practical rules in real applications (introduced later).

PCA: step-be-step explanation

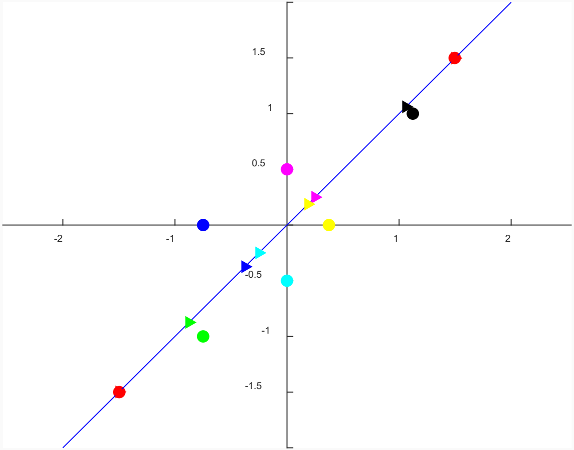


Figure 5: Recast the data along the first principal direction. Discard the second component and the dimension is reduced by 1.

Standard PCA workflow

1. Make sure the data X are rows=observations and columns=variables
2. Standardize the columns of X
3. Run [Q, X_{new}, d, tsquared, explained] = pca(X)
4. Using the variance% in "explained", choose k (usually 1, 2, or 3) components for visual analysis
   - For example, if $d = (1.9087, 0.0913)$, explained= $(95.4, 4.6)$, one may choose $k = 1$ as the first principal component carries 95.4% of the information
   - For example, if $d = (2.9108, 0.9212, 0.1474, 0.0206)$, explained= $(72.8, 23.0, 3.7, 0.5)$, one may choose $k = 2$ as the first two principal components carry 95.8% of the information
5. Plot $X_{\text{new}}(:, 1), \ldots, X_{\text{new}}(:, k)$ on a k-dimensional plot

Application: iris flower data set

- Iris flower data set contains 3 species (Setosa/Versicolor/Virginica) of 50 instances each. $n = 150$
- $p = 4$ attributes: sepal length in cm, sepal width in cm, petal length in cm, petal width in cm
- See Wikepedia for more details

sepal length (cm)	sepal width (cm)	pedal length (cm)	petal width (cm)	species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa

Table 1: The first 3 rows of the 150-instance Iris data set

Application: Iris flower data set

- Choose 2 principal components (reduce the size of our data by 50%)
- First 2 principal components explain $72.8\% + 23.0\% = 95.8\%$ of total variance of the data
- Based on 2 principal components, classification would still be an easy task

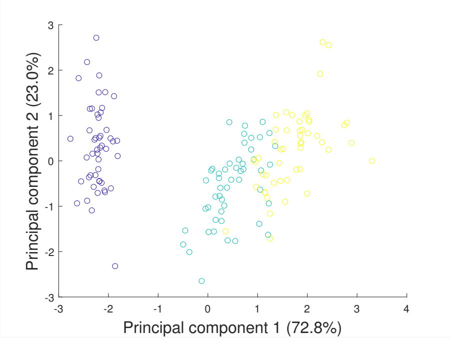


Figure 6: Setosa (blue), Versicolor (green), Virginica (yellow)

Application: iris flower data set

- Choose 1 principal components (throwing out a massive 75% of our data)
- Still end up with a remarkably effective single principle component for the purposes of classification

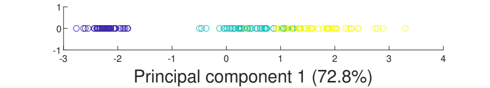


Figure 7: Setosa (blue), Versicolor (green), Virginica (yellow)

Programming Exercise

Do PCA on the Iris flower data set[1] and plot Figure 6 and Figure 7.

[1] `https://gist.githubusercontent.com/curran/a08a1080b88344b0c8a7/raw/0e7a9b0a5d22642a06d3d5b9bcbad9890c8ee534/iris.csv`

Application: iris flower data set

Listing 2: Matlab codes applying PCA to Iris flower data set

```
[X,y] = iris_dataset; % load the data
X = X'; % make sure rows=observations and columns=variables
y = vec2ind(y)'; % species label
for j = 1:size(X,2) % standardize columns of X
    X(:,j) = (X(:,j) - mean(X(:,j)))/std(X(:,j));
end
[Q,Xnew,d,tsquared,explained] = pca(X); % PCA
% plot 2 principal components
figure; scatter(Xnew(:,1),Xnew(:,2),25,y);
pc1 = sprintf('Principal component 1 (%2.1f%%)',...
    explained(1));
pc2 = sprintf('Principal component 2 (%2.1f%%)',...
    explained(2));
xlabel(pc1,'fontSize',15); ylabel(pc2,'fontSize',15);
% plot 1 principal component
figure; scatter(Xnew(:,1),zeros(size(X,1),1),25,y);
xlabel(pc1,'fontSize',15);
```