Lecture 4: Principal Component Analysis (PCA)

- To make explicit the concept of the SVD, a simple model example will be formulated that will illustrate all the key concepts associated with the SVD. The model to be considered will be a simple **spring–mass system** (弹簧质量系统) as illustrated in Fig. 15.6. Of course, this is a fairly easy problem to solve from basic concepts of $F = ma$. But for the moment, let's suppose we didn't know the **governing equations (控制方程)**.
- In fact, our aim in this section is to use a number of cameras (probes探头) to extract out data concerning the behavior of the system and then to extract empirically (以经验为主地) the governing equations of motion (运动).

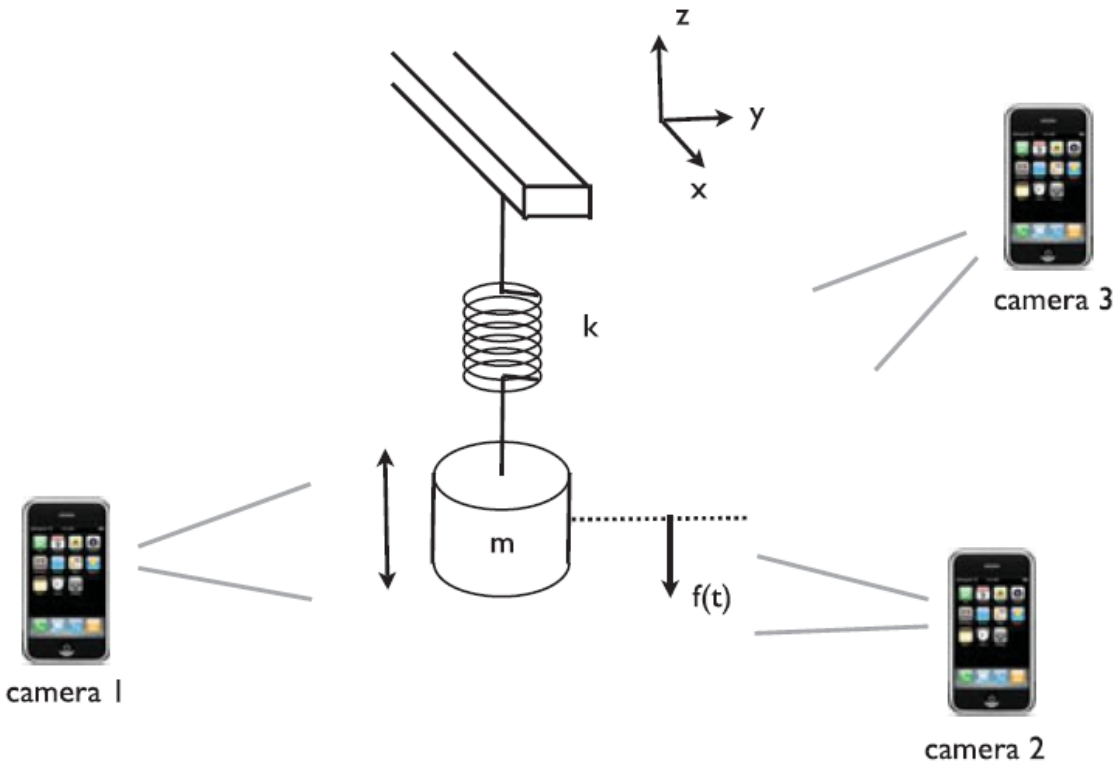


Figure 15.6: A prototypical example of how we might apply a principal component analysis, or SVD, is the simple mass–spring system exhibited here. The mass m is suspended with a spring with Hooke's constant k. Three video cameras collect data about its motion in order to ascertain its governing equations.

胡克定律，曾译为**虎克定律**，是力学弹性理论中的一条基本定律，表述为：固体材料受力之后，材料中的应力与应变（单位变形量）之间成线性关系。满足胡克定律的材料称为**线弹性**或**胡克型**（英文Hookean）材料。

从物理的角度看，胡克定律源于多数固体（或孤立分子）内部的原子在无外载作用下处于稳定平衡的状态。

许多实际材料，如一根长度为L、横截面积A的棱柱形棒，在力学上都可以用胡克定律来模拟——其单位伸长（或缩减）量（应变）在常系数E（称为弹性模量）下，与拉（或压）应力 σ 成正比例，即：弹簧给予物体的力F与长度变化量x成线性关系(F=-k·x或-F=-k·Δx)

This prologue(开场) highlights one of **the key applications of the SVD**, or alternatively a variant of **principal component analysis (PCA) (主成分分析)**.

Namely, from seemingly complex, perhaps random data, can low dimensional reductions of the dynamics and behavior be produced (动态降维捕捉运动特征) when the governing equations are not known?

Such methods can be used to **quantify(确定) low dimensional dynamics(低维动力学)** arising in such areas as structural vibrations (结构振动), damage detection (损伤检测), and neural decision making strategies (神经网络决策策略), image processing and signal analysis, to name just a few areas of application.

Thus the perspective to be taken here is clearly one in which the data analysis of an unknown, but potentially low dimensional system is to be analyzed.

Again we turn our attention to the simple experiment at hand: amass (质量) suspended (悬挂) by a spring (弹簧) as depicted in Fig. 15.6. **If the mass is perturbed (扰动) or taken from equilibrium (平衡) in the z-direction only**, we know that the governing equations are simply

$$\frac{d^2 f(t)}{dt^2} = -\omega^2 f(t)$$

where the **function *f (t)*** measures the displacement (位移) of the mass in the z-direction as a function of time. This has the well-known solution (in amplitude–phase form)

$$f(t) = A\cos(\omega t + \omega_0)$$

where the values of A and ω_0 are determined from the initial state of the system. This essentially states that the state of the system can be described by a one degree of freedom system.

In the above analysis, there are many things we have ignored. Included in the list of things we have ignored is the possibility that **the initial excitation (激励) of the system actually produces movement in the x−y plane** (平面). Further, there is potentially **noise in the data** from, for instance, shaking of the cameras during the video capture.

Moreover, from what we know of the solution, only a single camera is necessary to capture the underlying (潜在的) motion. In particular, a single camera in the x−y plane at z = 0 would be ideal. Thus we have oversampled (过采样的) the data with three cameras and have produced redundant data sets (冗余数据集).

From all of these potential perturbations (扰动) and problems, it is our goal to extract out the simple solution given by the simple harmonic motion (简谐振动).

This problem is a good example of what kind of processing is required to analyze a realistic data set. Indeed, one can imagine that most data will **be quite noisy, perhaps redundant, and certainly not produced from an optimal viewpoint**.

But through the process of PCA, these can be circumvented (徊避的) in order to extract out the ideal or simplified behavior. Moreover, we may even learn how to transform the data into the optimal viewpoint for analyzing the data.

Data collection and ordering

Assume now that we have started the mass in motion by applying a small perturbation (微扰) in the z-direction only. Thus the mass will begin to oscillate (摆动). **Three cameras are then used to record the motion (运动).** Each camera produces a two-dimensional representation of the data. If we denote the data from the three cameras with subscripts *a*, *b* and *c*, then the data collected are represented by the following:

$$\begin{aligned} \text{camera1}: &\ (\mathbf{x}_a, \mathbf{y}_a) \\ \text{camera2}: &\ (\mathbf{x}_b, \mathbf{y}_b) \\ \text{camera3}: &\ (\mathbf{x}_c, \mathbf{y}_c) \end{aligned}$$

where each set (x_j, y_j) is data collected over time of the position in the $x-y$ plane of the camera. Note that this is not the same $x-y$ plane of the oscillating mass system as shown in Fig. 15.6. Indeed, we should pretend (假装) we don't know the correct $x-y-z$ coordinates (坐标系) of the system. Thus the camera positions and their relative $x-y$ planes are arbitrary. **The length of each vector $\mathbf{x}_i$ and $\mathbf{y}_i$ depends on the data collection rate and the length of time the dynamics is observed.** We denote the length of these vectors as *n*.

All the data collected can then be gathered into a single matrix:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_a \\ \mathbf{y}_a \\ \mathbf{x}_b \\ \mathbf{y}_b \\ \mathbf{x}_c \\ \mathbf{y}_c \end{bmatrix}$$

Thus the matrix $X \in \mathbb{R}^{m \times n}$ where *m* represents the number of measurement types and *n* is the number of data points taken from the camera over time.

Now that the data have been arranged, two issues must be addressed: noise and redundancy. Everyone has an intuitive (直觉的) concept that noise in your data can only deteriorate (恶化), or corrupt (破坏), your ability to extract the true dynamics.

Just as in image processing, noise can alter (改变) an image beyond restoration (不能修复). Thus there is also some idea that if the measured data are too noisy, fidelity (保真度) of the underlying (底层) dynamics is compromised (破坏的) from a data analysis point of view.

The key measure of this is the so-called signal-to-noise ratio (信噪比): $\mathrm{SNR} = \sigma^2_{\mathrm{signal}}/\sigma^2_{\mathrm{noise}}$ where the ratio is given as the ratio of the variances(方差) of the signal and noise fields. A high SNR (much greater than unity) gives almost noiseless (high precision) data whereas a low SNR suggests the underlying signal is corrupted (毁坏的) by the noise.

The second issue to consider is redundancy. In the example of Fig. 15.6, the single degree of freedom is sampled by three cameras, each of which is really recording the same single degree of freedom. Thus **the measurements should be rife (普遍的) with redundancy**, suggesting that the different measurements are statistically dependent (统计相关). Removing this redundancy is critical for data analysis.

怎么消除冗余数据？？？

The covariance matrix

An easy way **to identify redundant data** is by considering the covariance (协方差) between data sets. Specifically, consider two sets of measurements with **zero means (0均值)** expressed in row vector form:

$$\mathbf{a} = [a_1 \ a_2 \ \cdots \ a_n] \quad \text{and} \quad \mathbf{b} = [b_1 \ b_2 \ \cdots \ b_n]$$

where the subscript denotes the sample number.

The variances (方差) of ***a*** and ***b*** are given by

$$\sigma_{\mathbf{a}}^2 = \frac{1}{n-1}\mathbf{a}\mathbf{a}^T$$
$$\sigma_{\mathbf{b}}^2 = \frac{1}{n-1}\mathbf{b}\mathbf{b}^T$$

while the **covariance between these two data sets** is given by

$$\sigma_{\mathbf{ab}}^2 = \frac{1}{n-1}\mathbf{a}\mathbf{b}^T$$

where the normalization constant (归一化常数) of $1/(n-1)$ is for an unbiased estimator (无偏估计).

由上面的公式，我们可以得到以下结论：

(1) 方差的计算公式是针对一维特征，即针对同一特征不同样本的取值来进行计算得到；而协方差则必须要求至少满足二维特征；方差是协方差的特殊情况。

(2) 方差和协方差的除数是n-1,这是为了得到方差和协方差的无偏估计。

协方差为正时，说明X和Y是正相关关系；协方差为负时，说明X和Y是负相关关系；协方差为0时，说明X和Y是相互独立。Cov(X,X)就是X的方差。当样本是n维数据时，它们的协方差实际上是协方差矩阵(对称方阵)。例如，对于3维数据(x,y,z)，计算它的协方差就是：

$$Cov(X,Y,Z) = \begin{bmatrix} Cov(x,x) & Cov(x,y) & Cov(x,z) \\ Cov(y,x) & Cov(y,y) & Cov(y,z) \\ Cov(z,x) & Cov(z,y) & Cov(z,z) \end{bmatrix}$$

样本均值：

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

样本方差：

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

样本X和样本Y的协方差：

$$\begin{aligned} Cov(X,Y) &= E[(X - E(X))(Y - E(Y))] \\ &= \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) \end{aligned}$$

We don’t just have two vectors, but potentially quite a number of experiments and data that would need to be correlated and checked for redundancy. In fact, the matrix in Eq. (15.3.4) is exactly what needs to be checked for covariance. The appropriate covariance matrix for this case is then

$$\mathbf{C_X} = \frac{1}{n-1}\mathbf{X}\mathbf{X}^T$$

This is easily computed with MATLAB from the command line:

```
cov(X)
```

The covariance matrix C_x is a square, symmetric (对称的) $m \times m$ matrix whose **diagonal (对角线的) represents the variance of particular measurements**. The **off-diagonal (非对角的) terms are the covariances between measurement types**. Thus C_x **captures the correlations between all possible pairs of measurements**.

Redundancy is thus easily captured since if two data sets are identical (identically redundant), the **off-diagonal term and diagonal term would be equal** since:

$$\sigma^2_{\mathbf{ab}} = \sigma^2_{\mathbf{a}} = \sigma^2_{\mathbf{b}} \text{ if } \mathbf{a} = \mathbf{b}$$

Thus **large off-diagonal terms correspond to redundancy** while **small off-diagonal terms suggest that the two measured quantities are close to being statistically independent and have low redundancy**. It should also be noted that **large diagonal terms**, or those with large variances, typically represent what we might consider **the dynamics of interest since the large variance suggests strong fluctuations in that variable**.

Thus the covariance matrix is the key component to understanding the entire data analysis.

方差是用来度量单个随机变量的离散程度，
而协方差则一般用来刻画两个随机变量的相似程度

如果结果为正值，则说明两者是正相关的（从协方差可以引出“相关系数”的定义），如果结果为负值，就说明两者是负相关，如果为0，则两者之间没有关系，就是统计上说的“相互独立”。

一. 数据中的冗余和噪声

我们先举一个例子，假设现在我们拿到这样一组数据，里面有两个属性，既有以“千米/每小时”度量的最大速度特征，也有“英里/小时”的最大速度特征，显然我们一眼就看出这两个特征有一个多余。

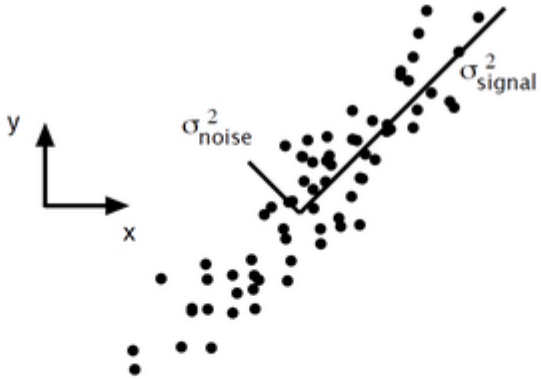
low redundancy　　high redundancy

怎么直观的判断数据是否冗余？上图从左往右，我们可以发现数据之间的关联性越来越强，也就是说两组数据越来越“相似”，我们用其中一组数据就能预测出另一组数据，这就是数据需要降维的其中一个原因：**冗余**。

另一方面，除了冗余之外，数据中可能还存在噪声，数据记录过程中存在某些不可抗因素的干扰。我们常常用信噪比signal-to-noise ratio (SNR) 来评价一组数据的好坏。

$$SNR = \frac{\delta^2_{signal}}{\delta^2_{noise}}$$

$SNR \gg 1$ 意味着数据比较纯净，可信度比较高，SNR 较小时，我们常说信号被噪声淹没了。



> 换句话说，一组数据中" 信号" 部分的**方差**较大，方差较小的我们可以认定为噪声。

下面我们来说一说协方差矩阵，

$$X = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_m \end{pmatrix}$$

协方差矩阵可表示为:

$$S_X = \frac{1}{n-1} X X^T$$

我们要消除冗余，也就是要使 S_x 非对角线上的元素为0（**相互独立**），也就是使得矩阵 X 的任意两行线性无关。

三. 数学描述

我们得到了降维问题可以表达为这样一个过程：

> 将一组 N 维向量降为K 维（K 大于0，小于N），其目标是选择K 个单位（模为1）**正交基**，使得原始数据变换到这组基上后，各字段两两间协方差为0（去冗余），而字段的方差则尽可能大（在正交的约束下，取**最大的K 个方差：降噪**）。

Achieving the goal

The insight given by the **covariance matrix** leads to our ultimate aim of

(i) removing redundancy

(ii) identifying those signals with maximal variance.

Thus in a mathematical sense we are simply asking to represent C_x (计算) so that the **diagonals are ordered from largest to smallest(去噪声) and the off-diagonals are zero(去冗余)**, i.e. our task is to **diagonalize(对角化) the covariance matrix**.

This is exactly what the SVD does, thus allowing it to becomes the tool of choice for data analysis and dimensional reduction (数据降维). In fact, the SVD diagonalizes and each singular direction (奇异方向) captures as much energy as possible as measured by the singular values σ_j.

Principal Components, Diagonalization and SVD

The example presented in the previous section shows that the key to analyzing a given experiment is to consider the **covariance matrix**

$$\mathbf{C_X} = \frac{1}{n-1}\mathbf{X}\mathbf{X}^T$$

where the matrix *X* contains the experimental data of the system. In particular, $\mathrm{X} \in \mathbb{C}^{m\times n}$ where *m* are the number of probes (探头) or measuring positions, and *n* is the number of experimental data points taken at each location.

In this setup, the following facts are highlighted:

- $C_x$is a square, symmetric (对称的) $m \times m$ matrix.
- The **diagonal terms of C_x are the variances for particular measurements**. By assumption, large variances correspond to dynamics of interest, whereas low variances are assumed to correspond to uninteresting dynamics.
- The **off-diagonal terms of C_x are the covariances between measurements**. Indeed, the off-diagonals capture the correlations between all possible pairs of measurements. A large off-diagonal term represents two events that have a high degree of redundancy, whereas a small off-diagonal coefficient means there is little redundancy in the data, i.e. they are statistically independent (统计独立).

Diagonalization(对角化)

The concept of diagonalization is critical for understanding the underpinnings (基础) of many physical systems. In this process of diagonalization, the correct coordinates (坐标系), or basis functions (基函数), are revealed (揭露) that reduce the given system to its low dimensional essence (本质). There is more than one way to diagonalize (对角化) a matrix, and this is certainly true here as well since the constructed covariance matrix C_x is square and symmetric, both properties that are especially beneficial for standard eigenvalue/eigenvector expansion techniques.

The key idea behind the diagonalization is simply this: there exists an ideal basis (理想基) in which the $C_x$can **be written (diagonalized) so that in this basis, all redundancies have been removed, and the largest variances of particular measurements are ordered**. In the language being developed here, this means that the system has been written in terms of its principal components (主成分), or in a proper orthogonal decomposition (正交分解).

Eigenvectors and eigenvalues

The most straightforward way to diagonalize the covariance matrix is by making the observation that $\mathbf{x}\mathbf{x}^T$ is a square, symmetric $m \times m$ matrix, i.e. it is self-adjoint so that the m eigenvalues are real and distinct. Linear algebra provides theorems which state that such a matrix can be rewritten as

$$\mathbf{X}\mathbf{X}^T = \mathbf{S}\Lambda\mathbf{S}^{-1}$$

where the **matrix *S* is a matrix of the eigenvectors** of $\mathbf{x}\mathbf{x}^T$ arranged in columns.

Since it is a symmetric (对称的) matrix, these eigenvector columns are orthogonal (正交的) so that ultimately the ***S* can be written as a unitary matrix** with $\mathbf{S}^{-1} = \mathbf{S}^T$. Recall that the matrix $\mathbf{\Lambda}$ is a diagonal matrix whose entries correspond to the m distinct eigenvalue of $\mathbf{x}\mathbf{x}^T$.

$$\mathbf{\Lambda} = \begin{pmatrix} \lambda_1 & 0 & \cdots & & 0 \\ 0 & \lambda_2 & 0 & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & & \cdots & 0 & \lambda_n \end{pmatrix} \qquad \mathbf{S} = \begin{pmatrix} \mathrm{x}_1 & \mathrm{x}_2 & \cdots & \mathrm{x}_n \end{pmatrix}$$

$$\mathbf{A} = \mathbf{S}\Lambda\mathbf{S}^{-1}$$

This suggests that instead of working directly with the matrix X, we consider working with the transformed variable, or in the **principal component basis(主成分基)**

$$\mathbf{Y} = \mathbf{S}^T\mathbf{X}$$

For this new basis, we can then consider its covariance

$$\begin{aligned} C_y &= \frac{1}{n-1} YY^T \\ &= \frac{1}{n-1}(S^T X)(S^T X)^T \\ &= \frac{1}{n-1} S^T XX^T S \\ &= \frac{1}{n-1} S^T S\Lambda S^{-} S \\ &= \frac{1}{n-1} S^T S\Lambda \\ &= \frac{1}{n-1}\Lambda \end{aligned}$$

In this basis (基), the principal components are the eigenvectors of $\mathbf{X}\mathbf{X}^T$ with the interpretation (说明) that the j_{th} diagonal value of C_Y is the variance of X along (沿) x_j, the j_{th} column of S.

The following lines of code produce the principal components of interest.

```
1  X = [1 2 3;4 5 6;7 8 9];
2  [m,n]=size(X); % compute data size
3  mn=mean(X,2); % compute mean for each row
4  X=X-repmat(mn,1,n); % subtract mean
5  Cx=(1/(n-1))*X*X'; % covariance
6  [V,D]=eig(Cx); % eigenvectors(V)/eigenvalues(D)
7  lambda=diag(D); % get eigenvalues
8  [dummy,m_arrange]=sort(-1*lambda); % sort in decreasing order
9  lambda=lambda(m_arrange);
10 V=V(:,m_arrange);
11 Y=V'*X; % produce the principal components projection
```

如果 A 为矩阵，则 mean(A,2) 是包含每一行均值的列向量。

B = sort(A) 按升序对 A 的元素进行排序。

当 A 具有 N 维时，B 的大小为 size(A).*[r1...rN]。例如：repmat([1 2; 3 4],2,3) 返回一个 4×6 的矩阵。

This simple code thus produces the eigenvalue decomposition and the projection (投影) of the original data onto the principal component basis (主成分基).

(1) 基于特征值分解协方差矩阵实现PCA算法

输入：数据集 $X=\{x_1,x_2,x_3,\ldots,x_n\}$ ，需要降到k维。

1) 去平均值(即去中心化)，即每一位特征减去各自的平均值。

2) 计算协方差矩阵 $\frac{1}{n}XX^T$,注：这里除或不除样本数量n或n-1,其实对求出的特征向量没有影响。

3) 用特征值分解方法求协方差矩阵 $\frac{1}{n}XX^T$ 的特征值与特征向量。

4) 对特征值从大到小排序，选择其中最大的k个。然后将其对应的k个特征向量分别作为行向量组成特征向量矩阵P。

5) 将数据转换到k个特征向量构建的新空间中，即Y=PX。

Singular value decomposition

A **second method for diagonalizing the covariance matrix** is the SVD method. In this case, **the SVD can diagonalize any matrix** by working in the appropriate pair of bases *U* and *V*. Thus by defining the transformed variable

$$\mathbf{Y} = \mathbf{U}^* \mathbf{X}$$

where *U* is the unitary transformation (酉变换) associated with the SVD: $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$.

Just as in the eigenvalue/eigenvector formulation, we then compute the variance in *Y*:

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$$

$\mathbf{U} \in \mathbb{C}^{m\times m}$ is unitary
$\mathbf{V} \in \mathbb{C}^{n\times n}$ is unitary
$\mathbf{\Sigma} \in \mathbb{R}^{m\times n}$ is diagonal.

$$\begin{aligned}
\mathbb{C}_Y &= \frac{1}{n-1}\mathbf{Y}\mathbf{Y}^T \\
&= \frac{1}{n-1}(\mathbf{U}^*\mathbf{X})(\mathbf{U}^*\mathbf{X})^T \\
&= \frac{1}{n-1}\mathbf{U}^*(\mathbf{X}\mathbf{X}^T)\mathbf{U} \\
&= \frac{1}{n-1}\mathbf{U}^*\mathbf{U}\mathbf{\Sigma}^2\mathbf{U}\mathbf{U}^* \\
\mathbf{C}_X &= \frac{1}{n-1}\Sigma^2 .
\end{aligned}$$

$$\begin{aligned}
\mathbf{A}\mathbf{A}^T &= (\mathbf{U}\mathbf{\Sigma}\mathbf{V}^*)(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^*)^T \\
&= \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*\mathbf{V}\mathbf{\Sigma}\mathbf{U}^* \\
&= \mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^*
\end{aligned}$$

This makes explicit the connection between the SVD and the eigenvalue method, namely that $\mathbf{\Sigma}^2 = \mathbf{\Lambda}$

The following lines of code produce the principal components of interest using the SVD (assume that you have the first three lines from the previous MATLAB code)

```
X=[1 2 3; 4 5 6; 7 8 9]
[m,n]=size(X)    %compute data size
mn=mean(X, 2)    %compute mean for each row
[u,s,v]=svd(X'/sqrt(n-1))    %perform the SVD
lambda=diag(s).^2    %produce diagonal variances
Y=u'*X    %produce the principal components projection
```

This gives the SVD method for producing the principal components. Overall, the SVD method is the more robust method and should be used. However, the connection between the two methods becomes apparent in these calculations.

(2) 基于SVD分解协方差矩阵实现PCA算法

输入：数据集 $X = \{x_1, x_2, x_3, \ldots, x_n\}$ ，需要降到k维。

1) 去平均值，即每一位特征减去各自的平均值。

2) 计算协方差矩阵。

3) 通过SVD计算协方差矩阵的特征值与特征向量。

4) 对特征值从大到小排序，选择其中最大的k个。然后将其对应的k个特征向量分别作为列向量组成特征向量矩阵。

5) 将数据转换到k个特征向量构建的新空间中。

在PCA降维中，我们需要找到样本协方差矩阵 XX^T 的最大k个特征向量，然后用这最大的k个特征向量组成的矩阵来做低维投影降维。可以看出，在这个过程中需要先求出协方差矩阵 XX^T ,当样本数多、样本特征数也多的时候，这个计算还是很大的。当我们用到SVD分解协方差矩阵的时候，SVD有两个好处：

1) 有一些SVD的实现算法可以先不求出协方差矩阵 XX^T 也能求出我们的右奇异矩阵V。也就是说，我们的PCA算法可以不用做特征分解而是通过SVD来完成，这个方法在样本量很大的时候很有效。实际上，scikit-learn的PCA算法的背后真正的实现就是用的SVD，而不是特征值分解。

2)注意到PCA仅仅使用了我们SVD的左奇异矩阵，没有使用到右奇异值矩阵，那么右奇异值矩阵有什么用呢？

假设我们的样本是m*n的矩阵X，如果我们通过SVD找到了矩阵 X^TX 最大的k个特征向量组成的k*n的矩阵 V^T ,则我们可以做如下处理：

$$X'_{m*k} = X_{m*n}V^T_{n*k}$$

可以得到一个m*k的矩阵X',这个矩阵和我们原来m*n的矩阵X相比，列数从n减到了k，可见对列数进行了压缩。也就是说，左奇异矩阵可以用于对行数的压缩；右奇异矩阵可以用于对列(即特征维度)的压缩。这就是我们用SVD分解协方差矩阵实现PCA可以得到两个方向的PCA降维(即行和列两个方向)。