# Principal Component Analysis

Abhijith Asok

Thursday, 9 November 2017

# Based on

1. Principal Component Analysis with linear algebra, by Jeff Jauregui
2. PCA Numerical Example, Engineering Statistics Handbook, NIST
3. Custom R code based on the numerical example

# What is PCA?

- Powerful analytical tool formulated using linear algebra.
- To detail its uses, consider a simple example:

We measure $m$ characteristics of $n$ individuals, i.e. we have $n$ samples of $m$-dimensional data. Hence, the $i$ th individual has her/his/their $m$ measurements recorded as $\vec{x_i}$ in the real number space of $m$ dimensions, denoted as $\mathbb{R}^m$.

For instance, let $n = 30$ and $m = 3$, comprising of height(in metres), weight(in kilograms) and IQ. A sample $\vec{x_i}$ would be

$$\vec{x_i} = [1.8, 70.3, 105]$$

# Why PCA?

PCA has the potential to answer the following questions:

## Can we *visualise* the data in a simpler way?

- Helpful to use lower number of dimensions in an effective way when $m$ is large.
- In the above example, $\mathbb{R}^3$ might be clustered around a plane and hence, the data could be effectively represented using just 2 dimensions.

## Which variables are *correlated*?

- Height and IQ may not be very correlated
- Height and weight might be very correlated

## Which variables are the most *significant* in the dataset?

# Linear Algebra Review

Let $A$ be an $m$ x $n$ matrix of real numbers and $A^T$ be its transpose.

## Theorem (Spectral Theorem)

**If $A$ is symmetric ($A^T = A$)**, then $A$ is orthogonally diagonalizable and has only real eigenvalues. In other words, there exist real numbers $\lambda_1, ..., \lambda_n$ (the eigenvalues) and orthogonal, non-zero real vectors $\vec{v_1}, ..., \vec{v_n}$ (the eigenvectors) such that for each $i = 1, 2, ..., n$:

$$A\vec{v_i} = \lambda_i \vec{v_i}$$

- **Orthogonal Matrix** - An $nxn$ matrix is orthogonal if $AA^T = I$.
- **Eigenvalue** - An eigenvalue of a matrix is any number $\lambda$ such that $det(A - \lambda I) = 0$
- **Eigenvector** - Vector corresponding to an eigenvalue that makes the above identity true.

# Linear Algebra Review

**Theorem**

If $A$ is any $m \times n$ matrix of real numbers, then the $m \times m$ matrix $AA^T$ and the $n \times n$ matrix $A^TA$ are both symmetric.

# Linear Algebra Review

## Theorem

The matrices $AA^T$ and $A^TA$ share the same **non-zero** eigenvalues.

## Proof

Let $\vec{v}$ be a **non-zero** eigenvector of $A^TA$ with eigenvalue $\lambda \neq 0$. Hence,

$$(A^TA)\vec{v} = \lambda\vec{v}$$

Multiplying by $A$ on both sides,

$$A(A^TA)\vec{v} = A\lambda\vec{v}$$

$$(AA^T)(A\vec{v}) = \lambda(A\vec{v})$$

Hence Proved, provided that $A\vec{v}$ is not a zero vector.

# Linear Algebra Review

**Alternatively**,

Let $\vec{v}$ be a **non-zero** eigenvector of $AA^T$ with eigenvalue $\lambda \neq 0$. Hence,

$$(AA^T)\vec{v} = \lambda\vec{v}$$

Multiplying by $A^T$ on both sides,

$$A^T(AA^T)\vec{v} = A^T\lambda\vec{v}$$

$$(A^TA)(A^T\vec{v}) = \lambda(A^T\vec{v})$$

Hence Proved, provided that $A\vec{v}$ is not a zero vector.

# Linear Algebra Review

The real power of this theorem arises when $m$ and $n$ are drastically different. eg: An $m$x$n$ matrix $A$ has $m = 500$ and $n = 2$.

Here, the dimensions of $AA^T$ are 500 x 500, which makes the calculation of 500 eigenvalues very difficult and time-consuming. Therefore, we calculate the 2 eigenvalues of $A^TA$ which is just a 2 x 2 matrix, which is much easier to do. By the previous theorem, the eigenvalues of the latter should be the same as the former. **Hence, the remaining 498 eigenvalues of $AA^T$ are zero**.

# Linear Algebra Review

## Theorem

The eigenvalues of $AA^T$ and $A^TA$ are nonnegative numbers

## Proof

The squared length of a vector $\vec{v}$ is effectively the dot product($\vec{v}.\vec{v}$), which is $\vec{v}^T\vec{v}$.
Here, let $\vec{v}$ be an eigenvector of $A^TA$ with an eigenvalue of $\lambda$. Then,

$$||A\vec{v}||^2 = (A\vec{v})^T(A\vec{v})$$

Since $(AB)^T = B^TA^T$,

$$||A\vec{v}||^2 = \vec{v}^T(A^TA)\vec{v} = \vec{v}^T\lambda\vec{v} = \lambda\vec{v}^T\vec{v} = \lambda||\vec{v}||^2$$

Since lengths are non-negative, the LHS is non-negative, which means the RHS has to be non-negative. Since$||\vec{v}||^2$ is non-negative, $\lambda$ is non-negative as well.

# Linear Algebra Review

**Theorem**

The trace of a matrix(sum of its diagonal elements) is the sum of its eigenvalues.

# Statistics Review

## Sample Average

$$\mu_A = \frac{1}{n}(a_1 + ..... + a_n)$$

## Sample Variance

$$\mu_A = \frac{1}{n-1}((a_1 - \mu_A)^2 + ..... + (a_n - \mu_A)^2)$$

## Covariance

$$Cov(A, B) = \frac{1}{n-1}((a_1 - \mu_A)(b_1 - \mu_B) + ..... + (a_n - \mu_A)(b_n - \mu_B))$$

# PCA Core

Remember that our data contains $n$ observations and $m$ variables.

First, we store the means of all $m$ variables of our data as a single vector in $\mathbb{R}^m$, i.e., having $m$ elements in it, each representing the mean of each variable.

$$\vec{\mu} = \frac{1}{n}(\vec{x_1} + \vec{x_2} + ..... + \vec{x_n})$$

Now that we have the mean of each variable in this single vector, we use vector operations and location-scale every sample vector $\vec{x_i}$, representing each of the $n$ observations, to center them around the mean. Let's call the transpose of this resultant data matrix, $B$. Since the original matrix $A$ had dimensions $n$ x $m$, this transposed matrix has dimensions $m$ x $n$. Every column of B is a particular $(\vec{x_i} - \vec{\mu})$

$$B = [(\vec{x_i} - \vec{\mu}).....(\vec{x_n} - \vec{\mu})]$$

# PCA Core

Remember that

$$Cov(A, B) = \frac{1}{n-1}((a_1 - \mu_A)(b_1 - \mu_B) + ..... + (a_n - \mu_A)(b_n - \mu_B))$$

Based on this formula, let's define a covariance matrix, $S$ as

$$S = \frac{1}{n-1} BB^T$$

Since $B$ has dimensions $m \times n$ and $B^T$ has dimensions $n \times m$, $S$ has dimensions $m \times m$.

# PCA Core

Remember our theorem that:

If $A$ is any $m \times n$ matrix of real numbers, then the $m \times m$ matrix $AA^T$ and the $n \times n$ matrix $A^TA$ are both symmetric.

Using this theorem, $BB^T$ is symmetric. Since $S$, the covariance matrix is just dividing $BB^T$ by a scalar, $S$ is also symmetric, meaning $S = S^T$.

Let's take an example to view what $S$ would look like. Let

$$\vec{x_1} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix}, \vec{x_2} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}, \vec{x_3} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix}, \vec{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \end{bmatrix}$$

# PCA Core

Therefore, $B$ is now

$$B = \begin{bmatrix} a_1 - \mu_1 & b_1 - \mu_1 & c_1 - \mu_1 \\ a_2 - \mu_2 & b_2 - \mu_2 & c_2 - \mu_2 \\ a_3 - \mu_3 & b_3 - \mu_3 & c_3 - \mu_3 \\ a_4 - \mu_4 & b_4 - \mu_4 & c_4 - \mu_4 \end{bmatrix}$$

In this case the entry in cell at the intersection of the first row and the first column of $S$ will be:

$$S_{11} = \frac{1}{3-1}((a_1 - \mu_1)^2 + (b_1 - \mu_1)^2 + (c_1 - \mu_1)^2)$$

This is effectively the variance of the first variable.

# PCA Core

Now,

$$S_{21} = \frac{1}{3-1}((a_1 - \mu_1)(a_2 - \mu_2) + (b_1 - \mu_1)(b_2 - \mu_2) + (c_1 - \mu_1)(c_2 - \mu_2))$$

This is the covariance of the first and second variables.

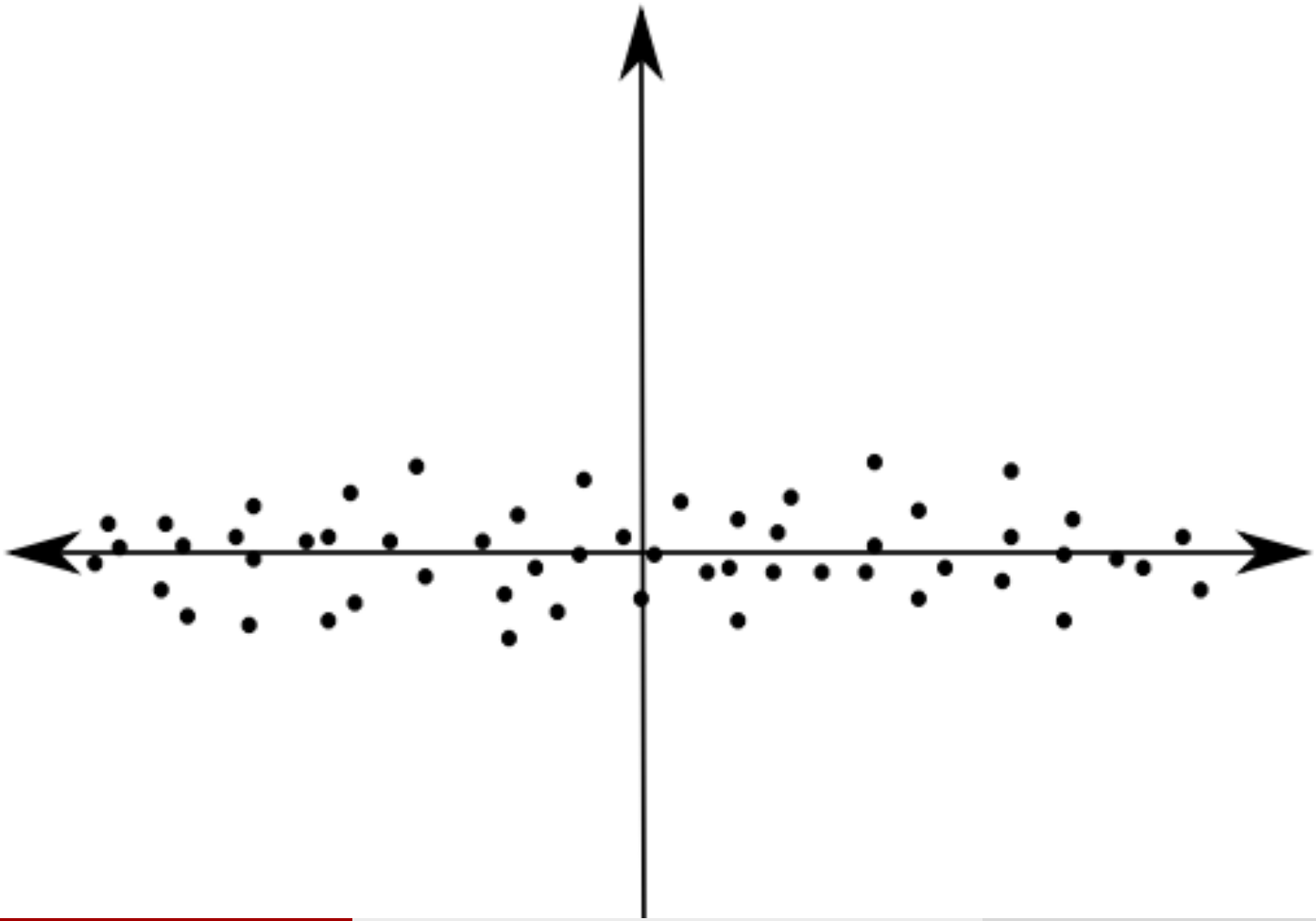
Generally

- The $i$th entry on the diagonal of $S$, namely $S_{ii}$, is the variance of the $i$th variable.
- The $ij$th entry of $S$, $S_{ij}$, with $i \neq j$, is the covariance between the *ith* and *jth* variables.

Let's see two examples to understand this better(m = 2).

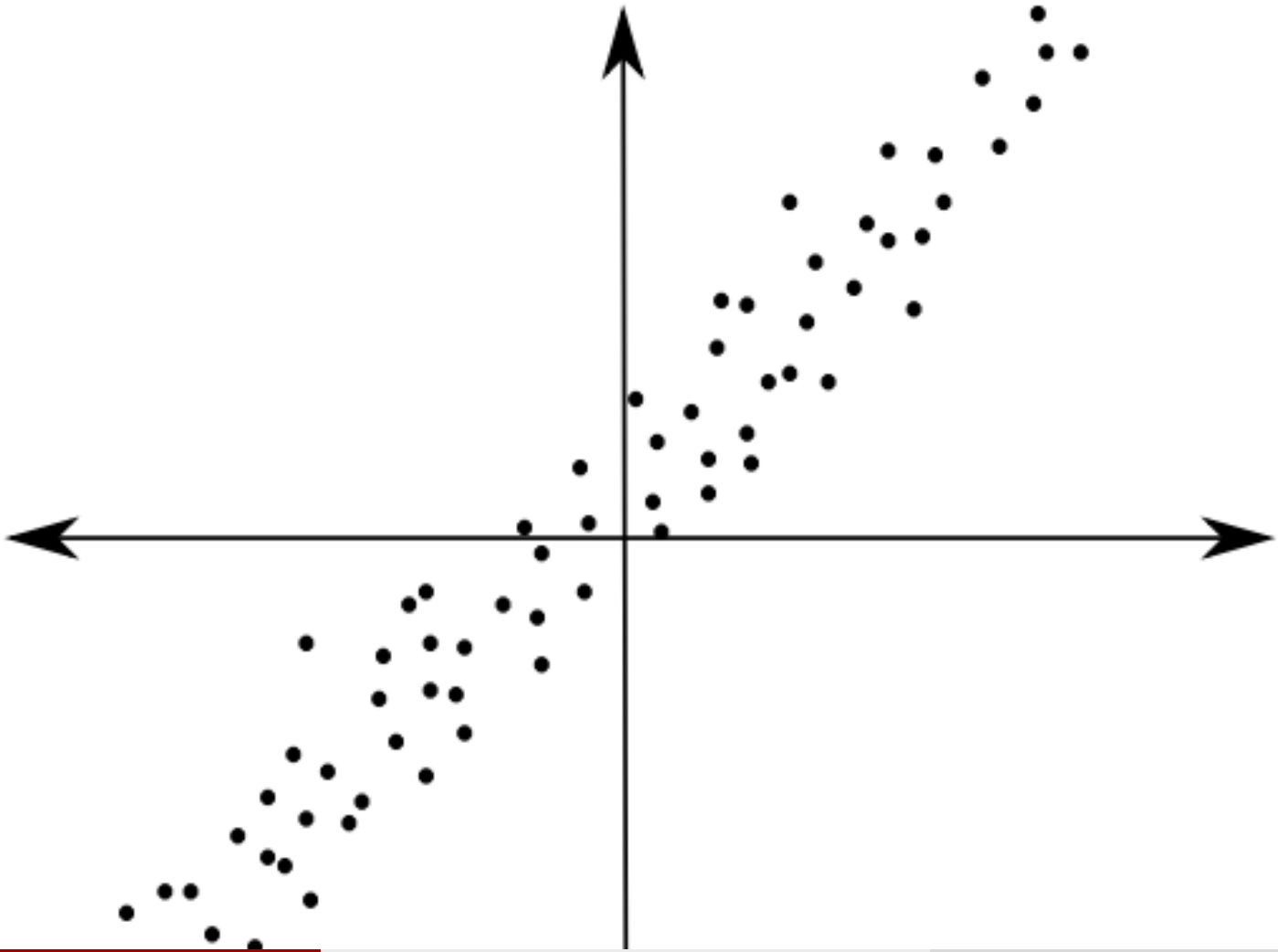
# PCA Core

We can expect $S_{11}$ to be high and $S_{12}, S_{21}, S_{22}$ to be low.



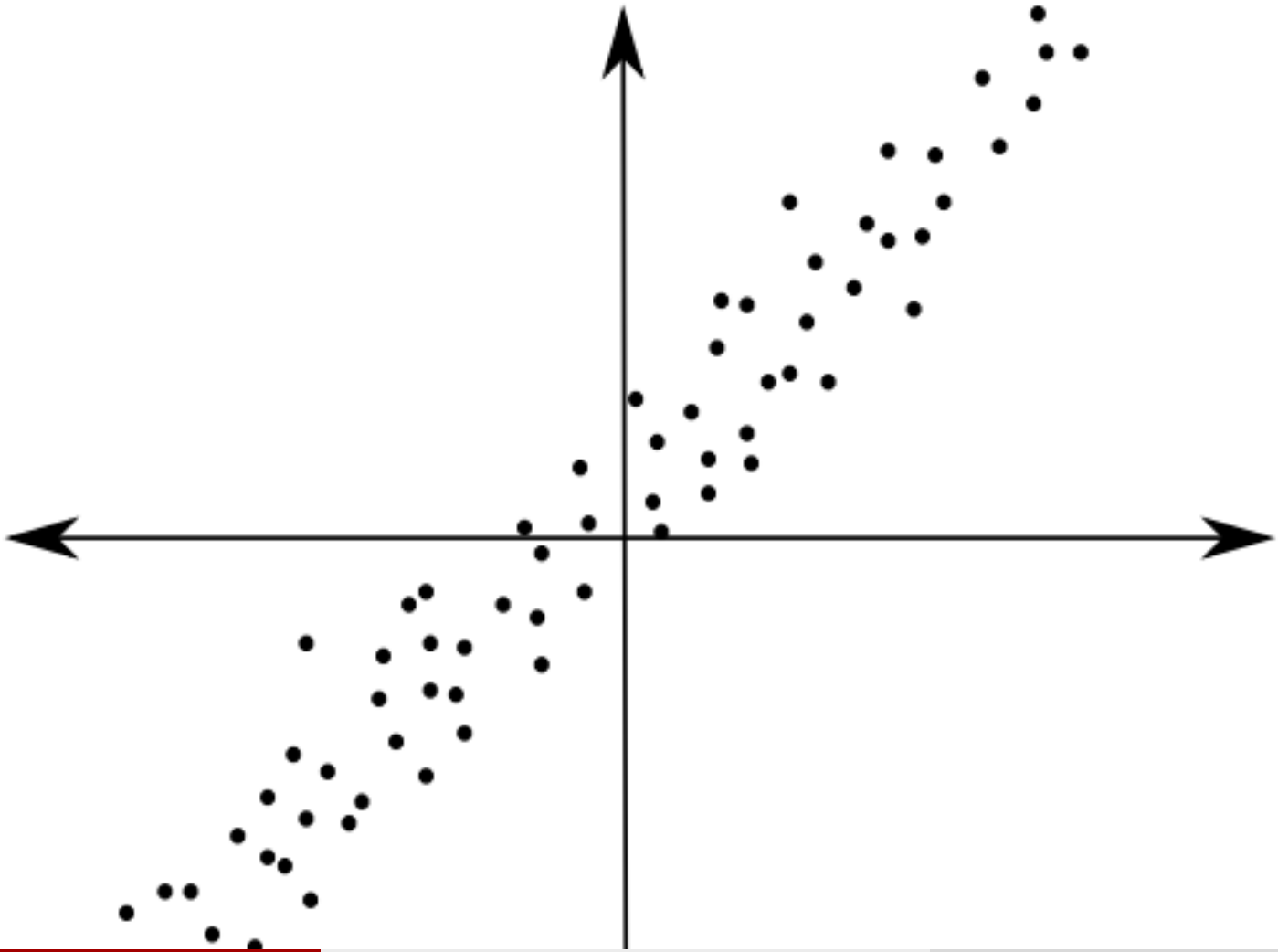
# PCA Core

How about this one?



# PCA Core

All of them are at a considerable level.



# Principal Components

Both the examples have the same shape - a set of points clustered along a line. However, their covariance matrix is completely different. PCA is able to recognize this algebraically.

Since the covariance matrix $S$ is symmetric, it is orthogonally diagonalizable by our previous theorem $A\vec{v_i} = \lambda_i \vec{v_i}$.

- Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m \geq 0$. They are all non-negative because the eigenvalues of $BB^T$ are nonnegative(as proved earlier) and $S$ is just $\frac{1}{n-1} BB^T$.
- The correspinding orthonormal eigenvectors are $\vec{u_1}, \vec{u_2}, \ldots, \vec{u_m}$. These are what are called **Principal Components of the dataset**. You could even replace all the eigenvectors by their negatives and it would make no difference(Since the negative sign cancels on either side of $A\vec{v} = \lambda\vec{v}$)

# Observations

The trace of $S$ is the sum of its diagonal elements, which is the sum of the $m$ variable variances. Let this be $T$, the total variance. From before, we know that $T$ is also the sum of the eigenvalues of the dataset, which is $T = \lambda_1 + \lambda_2 + \cdots + \lambda_m$.

- The first principal direction, $\vec{u_1}$ in $\mathbb{R}^m$ explains $\lambda_1$ amount of the total variance $T$, i.e. a fraction $\frac{\lambda_1}{T}$ of the total variance of the dataset. Similarly, $\vec{u_2}$ accounts for $\frac{\lambda_2}{T}$ and so on.
- Since $\lambda_1$ is the largest eigenvalue of the dataset, $\frac{\lambda_1}{T}$ is the largest fraction of the total variance among all the fractions represented by the eigenvalues and hence, the corresponding eigenvector $\vec{u_1} \in \mathbb{R}^m$ is the most significant direction of the dataset, since it stands for the most variance in the dataset.$\vec{u_2}$ follows it, since $\lambda_2$ is the next highest and so on...

# PCA's best use - Dimension reduction

- Often, the largest few eigenvaluesof S are much greater than all the others. For example,

$m = 10$, total variance, $T = 100$.Then, $\lambda_1$ might be 90.5, $\lambda_2$ might be 8.9 and all others might be less than 0.1. This means that the first and second principal components explain 99.4% of the variance.

- Hence, even though our original data is in $\mathbb{R}^{10}$, which is practically impossible to visualise, PCA tells us that all these points cluster around a 2-dimensional plane, where the dimensions are defined by the directions of $\vec{u_1}$ and $\vec{u_2}$. Since $\lambda_1$ is very much larger than $\lambda_2$, we can also say that the points will look like a rectangular strip on that 2D plane, since there is much more variance in the direction of $\vec{u_1}$ than in that of $\vec{u_2}$, which stand for $\lambda_1$ and $\lambda_2$ respectively.

**Therefore, we have reduced our 10-dimensional problem to just a 2-dimensional problem.**

# Summary of the PCA Algorithm

- Use the n samples of data, $\vec{x_1}, \vec{x_2}, \ldots, \vec{x_n}$ in $m$ dimensions(variables) to calculate the mean vector $\mu$ that contains the means of each variable.
- Build the matrix $B$, which is $B = [(\vec{x_i} - \vec{\mu}).....(\vec{x_n} - \vec{\mu})]$
- Compute the covariance matrix, S, which is $S = \frac{1}{n-1} BB^T$.
- Find the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_m$ of S(arranged in decreasing order for easy analysis of the top variance explainers), as well as the corresponding orthogonal eigenvectors $\vec{u_1}, \vec{u_2}, \ldots, \vec{u_m}$, using $A\vec{v_i} = \lambda_i \vec{v_i}$.
- **Interpret the results** : If a small number of $\lambda_i$ are much bigger than the others, a reduction of dimensions is possible. By examining what variables from the original dataset constitute each of those small number of $\lambda_i$s the most, we can find out the most important variables in the dataset. We could also examine which variables appear with opposite signs in constituting these $\lambda_i$s to gauge the relationship between them.

# Full-fledged example

Let's take a sample dataset of 3 variables and 10 observations regarding the measurements of a wafer. The 3 variables are *thickness*, *horizontal displacement* and *vertical displacement*. The dataset is:

$$X = \begin{bmatrix} 7 & 4 & 3 \\ 4 & 1 & 8 \\ 6 & 3 & 5 \\ 8 & 6 & 1 \\ 8 & 5 & 7 \\ 7 & 2 & 9 \\ 5 & 3 & 3 \\ 9 & 5 & 8 \\ 7 & 4 & 5 \\ 8 & 2 & 2 \end{bmatrix}$$

# Full-fledged example

Let us feed this into R, to assist with computation:

```
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version
## 3.4.1
```

```
library(formatR)
X <- matrix(c(7, 4, 6, 8, 8, 7, 5, 9, 7,
    8, 4, 1, 3, 6, 5, 2, 3, 5, 4, 2, 3, 8,
    5, 1, 7, 9, 3, 8, 5, 2), nrow = 10, ncol = 3)
```

# Full-fledged example

```
X
```

```
##       [,1] [,2] [,3]
##  [1,]    7    4    3
##  [2,]    4    1    8
##  [3,]    6    3    5
##  [4,]    8    6    1
##  [5,]    8    5    7
##  [6,]    7    2    9
##  [7,]    5    3    3
##  [8,]    9    5    8
##  [9,]    7    4    5
## [10,]    8    2    2
```

# Full-fledged example

In a practical setting, it usually done that B is scaled as well by the standard deviation, so that $S$ is the correlation matrix instead of covariance matrix. This is for better interpretation, which we will see towards the end.

Now, we create $B$ which, in this case is a 3 x 10 matrix, where each column is a particular $(\vec{x} - \vec{\mu})/\vec{\sigma}$

```
B <- apply(X, 1, function(x) {
    (x - apply(X, 2, mean))/apply(X, 2, sd)
})
```

# Full-fledged example

```
B
```

```
##             [,1]      [,2]        [,3]       [,4]
## [1,]  0.0656218 -1.903032 -0.59059616  0.7218398
## [2,]  0.3162278 -1.581139 -0.31622777  1.5811388
## [3,] -0.7481995  1.033228 -0.03562855 -1.4607705
##             [,5]       [,6]       [,7]      [,8]
## [1,] 0.7218398  0.0656218 -1.2468141 1.3780577
## [2,] 0.9486833 -0.9486833 -0.3162278 0.9486833
## [3,] 0.6769424  1.3895134 -0.7481995 1.0332279
##              [,9]       [,10]
## [1,]  0.06562180  0.7218398
## [2,]  0.31622777 -0.9486833
## [3,] -0.03562855 -1.1044850
```

# Full-fledged example

Now we compute $S$, which is $\frac{BB^T}{10-1}$.

```
S <- round((B %*% t(B))/(10 - 1), digits = 2)
S
```

```
##       [,1]  [,2]  [,3]
## [1,]  1.00  0.67 -0.10
## [2,]  0.67  1.00 -0.29
## [3,] -0.10 -0.29  1.00
```

# Full-fledged example

We can see that $S$ is symmetric

```
t(S) == S
```

```
##       [,1] [,2] [,3]
## [1,] TRUE TRUE TRUE
## [2,] TRUE TRUE TRUE
## [3,] TRUE TRUE TRUE
```

Therefore, by our theorem $A\vec{v} = \lambda\vec{v}$, we can say that $S\vec{v} = \lambda\vec{v}$. This implies that $S = \lambda$, if you cancel out the eigenvectors. But, in matrix form, we know that $\lambda = \lambda I$. Therefore, $S = \lambda I$ and then $|S - \lambda I|$ is zero, where $||$ stands for determinant of the matrix.

# Full-fledged example

$$\left| \begin{bmatrix} 1 & 0.67 & -0.1 \\ 0.67 & 1 & -0.29 \\ -0.1 & -0.29 & 1 \end{bmatrix} - \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix} \right| = 0$$

$$\left| \begin{bmatrix} 1-\lambda & 0.67 & -0.1 \\ 0.67 & 1-\lambda & -0.29 \\ -0.1 & -0.29 & 1-\lambda \end{bmatrix} \right| = 0$$

# Full-fledged example

$$(1-\lambda)[(1-\lambda)(1-\lambda)-0.29^2]-0.67[0.67(1-\lambda)-(0.1 \times 0.29)]-$$

$$0.1[(-0.67 \times 0.29)+0.1(1-\lambda)]=0$$

This is a third degree polynomial in $\lambda$, whose 3 roots are the eigenvalues of S that we need.

After getting $\lambda_1, \lambda_2, \lambda_3$, we find the corresponding eigenvectors by substituting each in $S\vec{v} = \lambda\vec{v}$. For example, we call $\vec{v_1}$ as

$$\begin{bmatrix} v_{11} \\ v_{21} \\ v_{31} \end{bmatrix}$$

# Full-fledged example

We solve for $S\vec{v} = \lambda\vec{v}$, which means

$$\begin{bmatrix} 1 & 0.67 & -0.1 \\ 0.67 & 1 & -0.29 \\ -0.1 & -0.29 & 1 \end{bmatrix} \begin{bmatrix} v_{11} \\ v_{21} \\ v_{31} \end{bmatrix} = \lambda_1 \begin{bmatrix} v_{11} \\ v_{21} \\ v_{31} \end{bmatrix}$$

We substitute $\lambda_1$ and find $v_{11}, v_{21}, v_{31}$ individually by comparing the LHS and RHS, thereby finding $\vec{v_1}$. Similarly, we find $\vec{v_2}$ and $\vec{v_3}$

# Full-fledged example

This entire process can be done in R using the *eigen*() function.

```
eigen(S)
```

```
## eigen() decomposition
## $values
## [1] 1.7703534 0.9277398 0.3019068
##
## $vectors
##             [,1]       [,2]       [,3]
## [1,]  0.6415972 0.38675484  0.6624000
## [2,]  0.6866832 0.09519199 -0.7206974
## [3,] -0.3417884 0.91725633 -0.2045031
```

# Full-fledged example

Therefore, our eigenvalues are:

```
eigen(S)$values
```

```
## [1] 1.7703534 0.9277398 0.3019068
```

Our corresponding eigenvectors are:

```
eigen(S)$vectors
```

```
##             [,1]       [,2]       [,3]
## [1,]  0.6415972 0.38675484  0.6624000
## [2,]  0.6866832 0.09519199 -0.7206974
## [3,] -0.3417884 0.91725633 -0.2045031
```

# Full-fledged example

Our eigenvalues are 1.77, 0.93, 0.3. Our total variance is the trace of $S$, which is the sum of diagonal elements of our covariance matrix $S$, where each diagonal element stands for the variance of each variable.

```
T <- sum(diag(S))
T
```

```
## [1] 3
```

Our first eigenvalue, $\lambda_1 = 1.77$, explains $(\frac{\lambda_1}{T}) * 100 = (\frac{1.77}{3}) * 100 = 59\%$ of the variance in the dataset. The first and second eigenvalues together represent $(\frac{\lambda_1+\lambda_2}{T}) * 100 = (\frac{1.77+0.93}{12.7}) * 100 = 90\%$ of the variance in the dataset. Hence, by ignoring the remaining 10% of variance, we have brought down our 3-variable model to a 2-variable model, thereby saving a lot of computation. This saving is amplified exponentially as the number of variables in the model increase.

# Full-fledged example

Now, we would also like to know how much of each variable in the original dataset is explained by the principal components, to know their composition for interpretation of the model. We don't go deep into the math of why we do what we do, but this is found out by first finding the **factor structure**, which is given by $F = \vec{V}\sqrt{(}\lambda)$, where $\vec{V}$ is the matrix consisting of all our eigenvectors and $\lambda$ is the diagonal matrix of our eigenvalues.

$$F = \begin{bmatrix} 0.64 & 0.38 & -0.66 \\ 0.69 & 0.1 & 0.72 \\ -0.34 & 0.91 & 0.2 \end{bmatrix} \begin{bmatrix} 1.33 & 0 & 0 \\ 0 & 0.96 & 0 \\ 0 & 0 & 0.55 \end{bmatrix}$$

# Full-fledged example

In R,

```
F <- eigen(S)$vectors %*% sqrt(matrix(c(1.77,
    0, 0, 0, 0.93, 0, 0, 0, 0.3), nrow = 3,
    ncol = 3))
F
```

```
##              [,1]       [,2]       [,3]
## [1,]  0.8535895 0.37297286  0.3628114
## [2,]  0.9135725 0.09179984 -0.3947422
## [3,] -0.4547199 0.88456997 -0.1120110
```

Since we computed the correlation matrix instead of covariance matrix, this matrix can be used for interpretation, i.e. for example, 0.91 is the correlation between the second variable and the first principal component.

# Full-fledged example

Now, computing communalities(which is the extent of correlation of an item with all other times), we multiply $F$ by its transpose, but using the first 2 principal components only(first 2 columns of F).

```
C <- F[,c(1,2)] %*% t(F[,c(1,2)])
C
```

```
##                [,1]       [,2]        [,3]
## [1,]  0.86772381  0.8140548 -0.05822351
## [2,]  0.81405479  0.8430420 -0.33421620
## [3,] -0.05822351 -0.3342162  0.98923418
```

The diagonal elements of this matrix stand for **communality**. This means that the first 2 principal components explain 87% of the first variable, 84% of the second variable and 99% of the third variable.