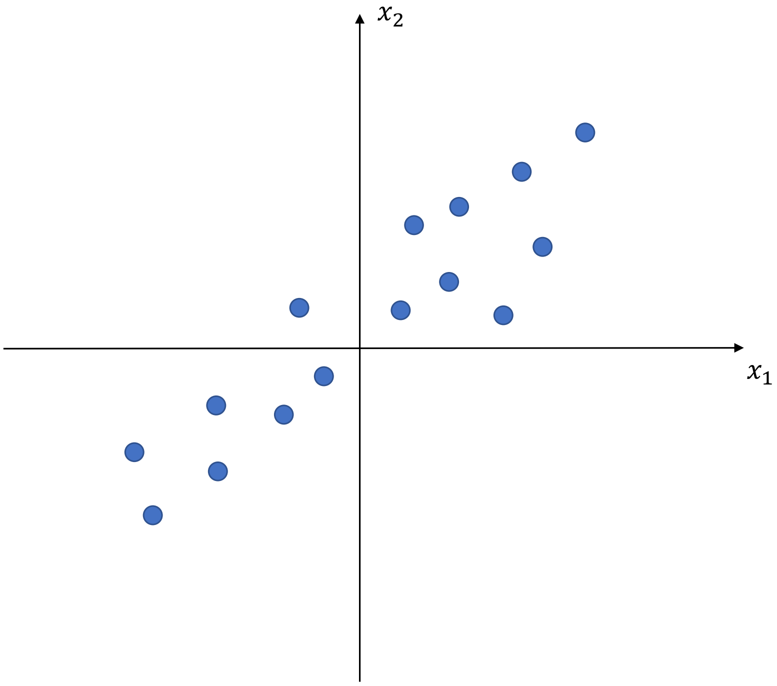
Intelligent Data Analysis
Lecture Notes on
Principal Component Analysis

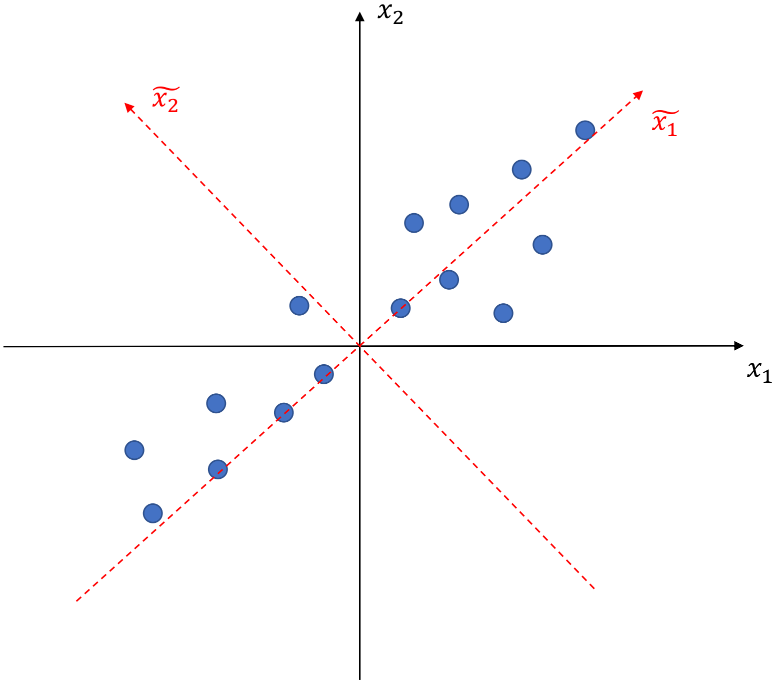
Peter Tiňo

1 Motivation

When we are given a large multi-dimensional data set (dimensionality is $d > 1$), it may be the case that the data is in fact embedded in a low-dimensional linear subspace. As a very simple example, let $\mathcal{D} = \{\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \ldots, \mathbf{x}^N\}$ be a two-dimensional data set of $N \in \mathbb{N}$ points $\mathbf{x}^i \in \mathbb{R}^2$, $i = 1, 2, \ldots, N$. Each point has $d = 2$ coordinates, $\mathbf{x}^i = [x_1^i, x_2^i]^T$, $x_1^i, x_2^i \in \mathbb{R}$. One can plot all the points in a two-dimensional Cartesian graph. We may obtain the following figure:



It appears that the data has some sort of "structure". But what exactly do we mean by saying that? The data is really 1-dimensional (modulo some "noise"). This would become more apparent if we expressed the same data in a new set of co-ordinate axes $\widetilde{x_1}$ and $\widetilde{x_2}$, obtained by rotating our original axes about the origin:



Most of the data structure is captured by axis $\widetilde{x_1}$. Projections of our points onto the second axis $\widetilde{x_2}$ represent "noise". In this sense we can say that our data set is inherently one-dimensional, aligned along $\widetilde{x_1}$ with a certain amount of noise aligned along $\widetilde{x_2}$. In other words, the axis $\widetilde{x_1}$ is special - it preserves most of the variability in $\mathcal{D}$. So **when projecting our data onto a lower dimensional subspace, we need to do it "intelligently" - i.e. by picking a subspace that contains most of the variability (and structure) of the original data.**

To talk about variability of the data in a quantitative manner, we will borrow notions from statistics and probability theory. We can do so because we will assume that our data points were generated by some (unknown) probability distribution ("the nature"). In other words, the data co-ordinates are realisations of some vector random variable. But let us start simple by concentrating on a single coordinate (1-dimensional case).

2 Mean and Variance of a Univariate Random Variable

We will review some fundamental basic concepts from statistics that will be needed for development of our dimensionality reduction technique. Suppose that we have a random variable X and we would like to measure its ”variability”. We have realisations of the random variable X - the data: $\mathcal{D}_X = \{x^1, x^2, \ldots, x^N\} \subset \mathbb{R}$. To characterise X in simple terms we can ask:

1) *What is the ‘center of gravity’ of X?* and

2) *How much does X fluctuate around this center of gravity?*

For question 1), we would like to know what is the average of the data. This is given by the following definition:

Definition 2.1. Let X be a random variable with probability distribution P and event set A. The *expected value* or *mean* of X, denoted as $\mathbb{E}[X]$, is defined by

$$\mathbb{E}[X] = \sum_{x \in A} P(X = x) \cdot x. \tag{1}$$

Equation (1) gives the theoretical quantity of the mean of X, because we assume we know its distribution P. Also, this definition works for discrete random variables. For the continuous case, one would need probability density of X and integral over support of X.

In practice we only can estimate what $\mathbb{E}[X]$ might be using our data. Our estimation of the expected value is given by

$$\widehat{\mathbb{E}[X]} = \frac{1}{N} \sum_{i=1}^{N} x^i. \tag{2}$$

Now we look at question 2), asking ourselves if there is a way to quantify deviations away from the mean. To answer this requires some simple intuition: consider the *square fluctuation* of a value x of X from the mean:

$$y = (x - \mathbb{E}[X])^2.$$

The values y can be considered realisations of a random variable $Y = (X - \mathbb{E}[X])^2$ representing the square fluctuation of X away from its mean. Our task is then to calculate the expected value of Y.

Definition 2.2. For a random variable X, let A be its event set. Furthermore, let Y be the random variable for the square fluctuations about $\mathbb{E}[X]$. The *variance* of X, denoted as $Var[X]$, is defined by

$$Var[X] = \mathbb{E}[Y] = \mathbb{E}[(X - \mathbb{E}[X])^2] = \sum_{x \in A} P(X = x) \cdot (x - \mathbb{E}[X])^2. \tag{3}$$

Figure 1: Examples of a) low variance and b) high variance.

As $Var(X)$ is the mean of $Y = (X - \mathbb{E}[X])^2$, we can estimate the variance using the data set:

$$\widehat{Var[X]} = \widehat{\mathbb{E}[Y]} = \frac{1}{N}\sum_{i=1}^{N}(x^i - \widehat{\mathbb{E}[X]})^2 \tag{4}$$

Remark 2.1. Strictly speaking, for our estimator $\widehat{Var[X]}$, we should divide the sum by $N-1$ instead of N (to obtain an unbiased estimate). An intuitive reason is that we use data $\mathcal{D}$ to estimate $\mathbb{E}[X]$ and the same data is then used again to estimate $\mathbb{E}[Y]$ that employs $\mathbb{E}[X]$. However, in practice, as $N >> 1$, we don't need to worry about this.

3 Quantifying variability of Multivariate Random Variables

We have established how to quantify the amount of variability/fluctuations of a scalar (univariate) random variable X around its mean. We would now like to generalise these notions to the case of vector random variables. After all, our data points will be higher dimensional vectors and, as mentioned before, they will be considered realisations of a vector random variable $\mathbf{X}$. Again, let us start simple by considering 2-dimensional case $d = 2$ and two random variables X_1 and X_2.

3.1 2-D Example

Let $\mathcal{D} = \{\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \ldots, \mathbf{x}^N\} \subset \mathbb{R}^2$ be our 2-D data set, $\mathbf{x}^i = [x_1^i, x_2^i]^T$, $i = 1, 2, ..., N$. We could compute the variance for each data dimension in isolation, but we might be missing an important information! What we should be asking ourselves is if the data coordinates are in some sense 'statistically linked/coupled' while their values fluctuate around their means.

Definition 3.1. Let X_1 and X_2 be random variables. Introduce a new random variable $Z = (X - \mathbb{E}[X_1]) \cdot (Y - \mathbb{E}[X_2])$. The *Co-variance* of X_1 and X_2 is defined by

$$Cov[X_1, X_2] = \mathbb{E}[Z] = \mathbb{E}[(X - \mathbb{E}[X_1]) \cdot (Y - \mathbb{E}[X_2])].$$

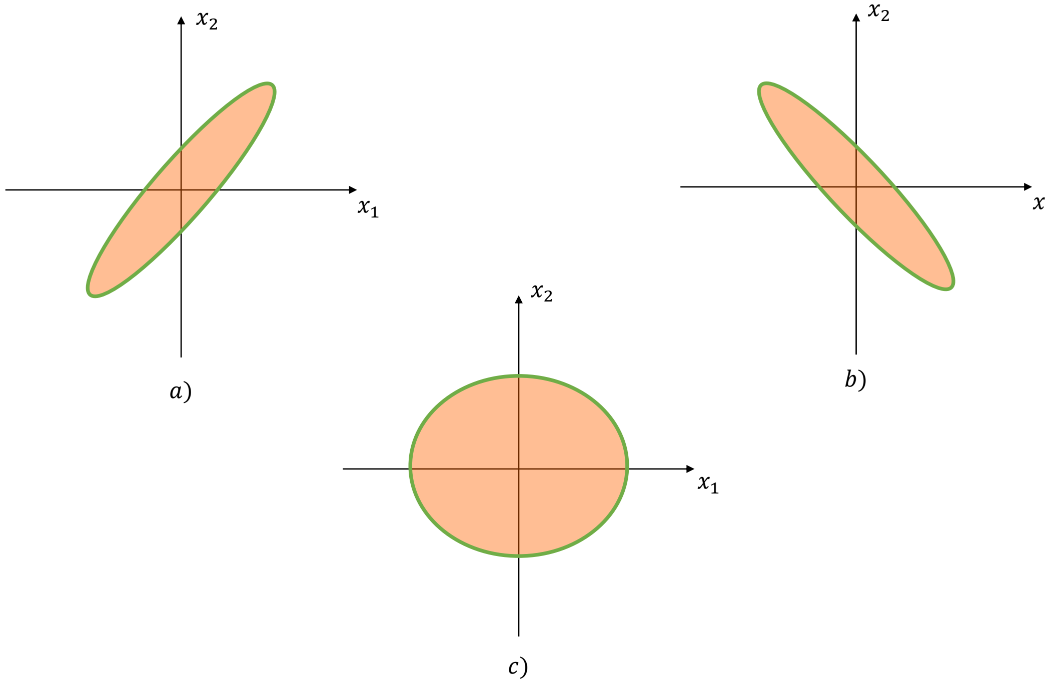
To estimate the co-variance of X_1 and X_2 in practice, if $\mathcal{D}$ is taken to be N realisations of (X_1, X_2),

$$Cov[X_1, X_2] \approx \widehat{Cov[X_1, X_2]} = \frac{1}{N} \sum_{i=1}^{N} \left(x_1^i - \widehat{\mathbb{E}[X_1]} \right) \cdot \left(x_2^i - \widehat{\mathbb{E}[X_2]} \right).$$

If the means of the random variables are centered around zero, then the estimate is simply

$$\widehat{Cov[X_1, X_2]} = \frac{1}{N} \sum_{i=1}^{N} x_1^i x_2^i.$$

To demonstrate how the co-variance of X_1 and X_2 behaves, we consider 3 cases below where the data set produces a) a positive covariance b) a negative covariance and c) no (zero) covariance.



a)

b)

c)

The following table provides the parity of $(X_1 - \mathbb{E}[X_1])$ and $(X_2 - \mathbb{E}[X_2])$ and the result of $Cov[X_1, X_2]$ in each case. In case a) if $(X_1 - \mathbb{E}[X_1])$ is a positive value then $(X_2 - \mathbb{E}[X_2])$ is *on average* also positive, resulting in $Cov[X_1, X_2]$ being greater than zero. In case b) if $(X_1 - \mathbb{E}[X_1])$ is a positive value then $(X_2 - \mathbb{E}[X_2])$ is *on average* negative, resulting in negative $Cov[X_1, X_2]$. Finally, in case c) if $(X_1 - \mathbb{E}[X_1])$ is a positive value then $(X_2 - \mathbb{E}[X_2])$ can be positive or negative ”with equal measure”, resulting in $Cov[X_1, X_2] \approx 0$.

cases	$(X_1 - \mathbb{E}[X_1])$	$(X_2 - \mathbb{E}[X_2])$	$Cov[X_1, X_2]$
a)	$+/-$	$+/-$	> 0
b)	$+/-$	$-/+$	< 0
c)	$+/-$	$+/-$ in both cases	≈ 0

3.2 General Case, $d \geq 2$

It is unreasonable to expect that our data sets will always be two-dimensional, so how we generalise our idea of the co-variance when data dimensionality gets larger than 2? We need to measure covariances of all pairs of coordinates.

Definition 3.2. Consider the vector random variable $\mathbf{X} = (X_1, X_2, \ldots, X_d)^T$, where $d \geq 2$. The *co-variance matrix* of $\mathbf{X}$ is a $d \times d$ square and symmetric matrix, defined by

$$Cov[\mathbf{X}] = \begin{bmatrix} Var[X_1] & Cov[X_1, X_2] & Cov[X_1, X_3] & ... & Cov[X_1, X_d] \\ Cov[X_2, X_1] & Var[X_2] & Cov[X_2, X_3] & ... & Cov[X_2, X_d] \\ Cov[X_3, X_1] & Cov[X_3, X_2] & Var[X_3] & ... & Cov[X_3, X_d] \\ . & . & . & ... & . \\ . & . & . & ... & . \\ Cov[X_d, X_1] & Cov[X_d, X_2] & Cov[X_d, X_3] & ... & Var[X_d] \end{bmatrix}$$

Note that, for each $i = 1, 2, \ldots, d$, $Cov[X_i, X_i] = Var[X_i]$. So along the diagonal we have variances.

Suppose now that we have N realisations of the vector random variable $\mathbf{X}$:

$$\begin{aligned} &\mathbf{x}^1 = (x_1^1, x_2^1, ..., x_d^1)^T \\ &\mathbf{x}^2 = (x_1^2, x_2^2, ..., x_d^2)^T \\ &\vdots \\ &\mathbf{x}^N = (x_1^N, x_2^N, ..., x_d^N)^T. \end{aligned}$$

Let $\mathcal{X} = (\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^N)$ be the design matrix storing the N realisations $\mathbf{x}^i$ of $\mathbf{X}$ as columns. Suppose also that each random variable is centered: $\mathbb{E}[X_i] = 0$, $i = 1, 2, \ldots, d$. Then we can estimate the co-variance of $\mathbf{X}$ by

$$Cov[\mathbf{X}] \approx \widehat{Cov[\mathbf{X}]} = \frac{1}{N} \mathcal{X} \mathcal{X}^T.$$

4 Aligning Co-ordinate Axes with the Data

Our objective is to rotate our axes so that the data is“more aligned” with the new axes than with the original ones. By this we mean that in the new axes system the co-variances vanish - there is no (linear) interplay between the co-ordinates. All we need to consider are the variances (average amount of squared fluctuations) along the new axes. It will then be easy to identify the “good” directions where most of data variation happens and the “redundant ones” (those where the fluctuations are small/negligible).

Let $\widetilde{\mathbf{X}} = (\tilde{X}_1, \tilde{X}_2, \ldots, \tilde{X}_d)^T$ be the original vector random variable $\mathbf{X}$ expressed in the new axes. The co-variance matrix of $\widetilde{\mathbf{X}}$ is

$$Cov[\widetilde{\mathbf{X}}] = \begin{bmatrix} Var[\tilde{X_1}] & 0 & 0 & ... & 0 \\ 0 & Var[\tilde{X_2}] & 0 & ... & 0 \\ 0 & 0 & Var[\tilde{X_3}] & ... & 0 \\ . & . & . & ... & . \\ . & . & . & ... & . \\ 0 & 0 & 0 & ... & Var[\tilde{X_d}] \end{bmatrix}$$

How to compute just from the data what our new axes should in order to achieve the desired result (diagonal covariance matrix)? We will first look at how we project a given data set onto the new axes. This (as will be clear soon) can be achieved through a (canonical) dot product of vectors: Let $\mathbf{a} = [a_1, a_2, \ldots, a_d]^T$ and $\mathbf{b} = [b_1, b_2, \ldots, b_d]^T$ be vectors in $\mathbb{R}^d$. Let α be the angle between $\mathbf{a}$ and $\mathbf{b}$. The *dot product* between $\mathbf{a}$ and $\mathbf{b}$ is

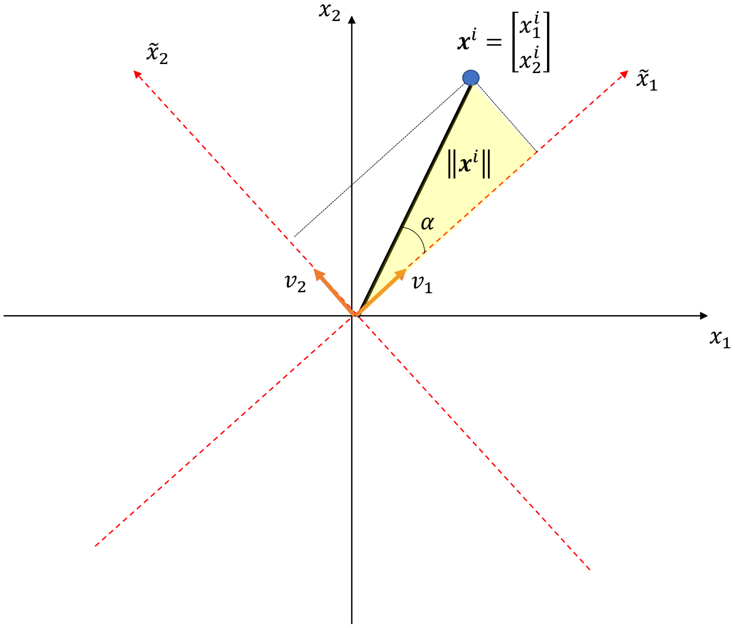
$$\mathbf{a} \cdot \mathbf{b} = \mathbf{a}^T\mathbf{b} = \sum_{i=1}^{d} a_i \cdot b_i = ||\mathbf{a}|| \cdot ||\mathbf{b}|| \cos \alpha,$$

where $||\mathbf{a}||$ is the (L_2) *norm* (or length) of $\mathbf{a}$, defined as $||\mathbf{a}|| = \sqrt{a_1^2 + a_2^2 + \ldots + a_d^2}$. Note that $||\mathbf{a}||^2 = \mathbf{a}^T\mathbf{a}$.

4.1 Changing the Axes - 2-D Case

Let $\mathbf{x}^i = [x_1^i, x_2^i]^T \in \mathbb{R}^2$ be a data point in 2-D space. Our mew axes can be fully specified by *unit vectors* (length one) $\mathbf{v}_1$ and $\mathbf{v}_2$ pointing in the axes directions. Obviously, $\|\mathbf{v}_1\| = \|\mathbf{v}_2\| = 1$ and $\mathbf{v}_1$, $\mathbf{v}_2$ are orthogonal, i.e. $\mathbf{v}_1^T\mathbf{v}_2 = 0$.

When changing our axes (x_1, x_2) to $(\tilde{x}_1, \tilde{x}_2)$, we want to project $\mathbf{x}^i$ onto $\mathbf{v}_1$ and $\mathbf{v}_2$ to obtain $\widetilde{\mathbf{x}}^i = [\tilde{x}_1^i, \tilde{x}_2^i]^T$. Let α be the angle between $\mathbf{x}^i$ and $\mathbf{v}_1$.



To obtain our first co-ordinate $\tilde{x}_1^i$, we notice that (recall $||\mathbf{v}_1|| = 1$):

$$\tilde{x}_1^i = ||\mathbf{x}^i|| \cos\alpha = ||\mathbf{v}_1|| \cdot ||\mathbf{x}^i|| \cos\alpha = \mathbf{v}_1^T \mathbf{x}^i$$

Similarly, we also find that $\tilde{x}_2^i = \mathbf{v}_2^T \mathbf{x}^i$. Using these facts, let $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2]$ be the matrix containing the direction vectors for each axes as columns. We see that

$$\mathbf{V}^T \cdot \mathbf{x}^i = \tilde{\mathbf{x}}^i$$

Suppose now that we are in our new axes system $(\tilde{x}_1, \tilde{x}_2)$; how do we express our point $\tilde{\mathbf{x}}^i$ back in the original system (x_1, x_2)? We would of course multiply $\tilde{\mathbf{x}}^i$ by the matrix $(\mathbf{V}^T)^{-1}$, the inverse matrix to $\mathbf{V}^T$. Because of the properties of the direction vectors (they are orthogonal to each other and have length 1 - we also say that they form an orthonormal basis), finding this matrix is very simple.

Claim: $(\mathbf{V}^T)^{-1} = \mathbf{V}$

Proof.

$$\begin{aligned}
\mathbf{V}^T \cdot \mathbf{V} &= \left[\begin{array}{c} \mathbf{v}_1^T \\ \mathbf{v}_2^T \end{array} \right] \cdot [\mathbf{v}_1 \mathbf{v}_2] \\
&= \left[\begin{array}{cc} \mathbf{v}_1^T \mathbf{v}_1 & \mathbf{v}_1^T \mathbf{v}_2 \\ \mathbf{v}_2^T \mathbf{v}_1 & \mathbf{v}_2^T \mathbf{v}_2 \end{array} \right] \\
&= \left[\begin{array}{cc} ||\mathbf{v}_1||^2 & 0 \\ 0 & ||\mathbf{v}_2||^2 \end{array} \right] \\
&= \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array} \right] \\
&= I_2
\end{aligned}$$

Similarly, $\mathbf{V} \cdot \mathbf{V}^T = I_2$ □

4.2 Changing the Axes - multi-dimensional case, $d \geq 2$

Note that there was nothing in the arguments above that was special to 2 dimensions. Exactly the same trick can be applied to change co-ordinate systems in $d > 2$-dimensional case. If we let $(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_d)$ be our new axes and $\mathbf{x}^i = [x_1^i, x_2^i, \ldots, x_d^i]^T$ a data point expressed in the original axes, we can project our point onto our new axis using:

$$\mathbf{V}^T \cdot \mathbf{x}^i = \tilde{\mathbf{x}}^i.$$

As before, $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_d]$, each column $\mathbf{v}_i$ being the unit direction vector of axis $\tilde{x}_i$ orthogonal to the other direction vectors.

If we wanted to express $\tilde{\mathbf{x}}^i$ in the original axes, we would simply use:

$$\mathbf{V} \cdot \tilde{\mathbf{x}}^i = \mathbf{x}^i$$

As mentioned before, we can collect all data points (as columns) in the so-called *design matrix* $\mathcal{X} = [\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^N]$ (a $d \times N$ matrix). Then it is easy to show that if the data is centered (zero mean), the covariance matrix can be estimated as (show it!)

$$\mathbf{C} = \frac{1}{N} \mathcal{X} \mathcal{X}^T.$$

Of course, the data can always be easily centered - just subtract from each data point the mean of the whole data set $\mathcal{D}$.

It is straightforward to show that the data expressed in the new coordinate system stays centered. Since the data in the original system is centered, we have

$$\frac{1}{N} \sum_{i=1}^{N} \mathbf{x}^i = \mathbf{0},$$

where $\mathbf{0}$ is a d-dimensional vector of 0's. Hence,

$$\begin{aligned}
\frac{1}{N}\sum_{i=1}^{N}\tilde{\mathbf{x}}^i &= \frac{1}{N}\sum_{i=1}^{N}\mathbf{V}^T\mathbf{x}^i \\
&= \mathbf{V}^T\left[\frac{1}{N}\sum_{i=1}^{N}\mathbf{x}^i\right] \\
&= \mathbf{V}^T\mathbf{0} = \mathbf{0}.
\end{aligned}$$

Note that the design matrix of the data in the new system can be obtained as:

$$\tilde{\mathcal{X}} = \mathbf{V}^T\mathcal{X}$$

and, of course, we can transform back as

$$\mathcal{X} = \mathbf{V}\tilde{\mathcal{X}}.$$

Now, since the data $\tilde{\mathbf{x}}^i$ is also centered, the co-variance matrix calculated in the new system can be obtained as

$$\tilde{\mathbf{C}} = \frac{1}{N}\tilde{\mathcal{X}}\tilde{\mathcal{X}}^T.$$

We ask: What is the relationship of the two covariance matrices $\mathbf{C}$ and $\tilde{\mathbf{C}}$? After all, they are *covariance matrices of the same data, expressed in two different co-ordinate systems.* We have

$$\begin{aligned}
\mathbf{C} &= \frac{1}{N}\sum_{i=1}^{N}\mathcal{X}\mathcal{X}^T \\
&= \frac{1}{N}\sum_{i=1}^{N}\mathbf{V}\tilde{\mathcal{X}}\left(\mathbf{V}\tilde{\mathcal{X}}\right)^T \\
&= \frac{1}{N}\sum_{i=1}^{N}\mathbf{V}\tilde{\mathcal{X}}\tilde{\mathcal{X}}^T\mathbf{V}^T \\
&= \mathbf{V}\left[\frac{1}{N}\sum_{i=1}^{N}\tilde{\mathcal{X}}\tilde{\mathcal{X}}^T\right]\mathbf{V}^T \\
&= \mathbf{V}\tilde{\mathbf{C}}\mathbf{V}^T
\end{aligned}$$

We have found that if we rotate our co-ordinate system as prescribed by directional vectors stored as columns of matrix $\mathbf{V}$, and these directional vectors from an orthonormal system, then the relationship between the covariance matrix $\mathbf{C}$ calculated in the original system and the covariance matrix $\tilde{\mathbf{C}}$ calculated in the new system reads:

$$\mathbf{C} = \mathbf{V}\tilde{\mathbf{C}}\mathbf{V}^T.$$

4.3 Finding the right axes

We must now finally must face the question of what exactly the new axis should be? In other words, we must, based on our data $\mathcal{D}$ expressed in the original co-ordinate system, suggest a new co-ordinate system through supplying appropriate $\mathbf{V}$ that will result in diagonal covariance matrix $\tilde{\mathbf{C}}$. To do this, we need to learn about an important mathematical concept - eigen decomposition of symmetric matrices into a set of *eigenvectors* and their associated real *eigenvalues*.

Definition 4.1. Let $\mathbf{A} \in \mathbb{R}^{d \times d}$ be a $d \times d$ square symmetric matrix. A vector $\mathbf{v} \in \mathbb{R}^d$ is called an *eigenvector* of $\mathbf{A}$ if there exist a non-zero constant λ, such that

$$\mathbf{A}\mathbf{v} = \lambda \mathbf{v}.$$

The scalar $\lambda \in \mathbb{R}$ is called *eigenvalue* of $\mathbf{A}$ corresponding to the eigenvector $\mathbf{v}$.

This concept of vectors that only 'stretch' or 'contract' under the transformation by a matrix is important because it can be shown that

a) A $d \times d$ symmetric matrix will have exactly d eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_d$, with real $\lambda_1, \lambda_2, \ldots, \lambda_d$ as the corresponding eigenvalues[1].

b) For *any* square matrix $\mathbf{A}$, we can decompose it into a special canonical form, known as *matrix diagonalisation*:

$$\mathbf{A} = \mathbf{V}\mathbf{D}\mathbf{V}^T,$$

were $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_d]$ is a matrix formed by the eigenvectors of $\mathbf{A}$ (as columns), and $\mathbf{D} = diag(\lambda_1, \lambda_2, \ldots, \lambda_d)$ is a diagonal matrix with non-zero diagonal entries equal to the eigenvalues of $\mathbf{A}$.

Amazingly, this is exactly what we need! :-)
Using the facts above, we can equate the original covariance matrix $\mathbf{C}$ with the symmetric matrix $\mathbf{A}$, directional matrix $\mathbf{V}$ with the matrix containing eigenvectors of $\mathbf{C}$ and the transformed covariance matrix $\tilde{\mathbf{C}}$ with the diagonal matrix $\mathbf{D} = diag(\lambda_1, \lambda_2, \ldots, \lambda_d)$ containing eigenvalues of $\mathbf{C}$. This means that the eigendecomposition of the original covariance matrix $\mathbf{C}$ can give us *both* the directions of the new co-ordinate system (eigenvectors of $\mathbf{C}$) *and* variances $Var[\tilde{X}_1], Var[\tilde{X}_2], \ldots, Var[\tilde{X}_d]$ that our data has along the new co-ordinate axes (eigenvalues of $\mathbf{C}$ appearing as diagonal elements of $\tilde{\mathbf{C}}$).

To further demonstrate the arguments above, suppose that we are in two-dimensional space. We want to show that $\mathbf{C}\mathbf{a} = \mathbf{V}\tilde{\mathbf{C}}\mathbf{V}^T\mathbf{a}$ for every $\mathbf{a} \in \mathbb{R}^2$. It suffices to show that $\mathbf{C}\mathbf{v}_i = \mathbf{V}\tilde{\mathbf{C}}\mathbf{V}^T\mathbf{v}_i$ for $i = 1, 2$, with $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2]$ and $\tilde{\mathbf{C}} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$.

We know that $\mathbf{C}\mathbf{v}_i = \lambda_i \mathbf{v}_i$. Now,

[1] Strictly speaking this is true only if all eigenvalues are non-zero and different.

$$
\begin{aligned}
\mathbf{V}\tilde{\mathbf{C}}\mathbf{V}^T\mathbf{v}_1 &= \mathbf{V}\tilde{\mathbf{C}}\begin{bmatrix} \mathbf{v}_1^T\mathbf{v}_1 \\ \mathbf{v}_2^T\mathbf{v}_1 \end{bmatrix} \\
&= \mathbf{V}\tilde{\mathbf{C}}\begin{bmatrix} 1 \\ 0 \end{bmatrix} \\
&= \mathbf{V}\begin{bmatrix} \lambda_1 \\ 0 \end{bmatrix} \\
&= \lambda_1\mathbf{v}_1
\end{aligned}
$$

Using similar reasoning, $\mathbf{C}\mathbf{v}_2 = \mathbf{V}\tilde{\mathbf{C}}\mathbf{V}^T\mathbf{v}_2$.

To summarize, we have shown that the covariance matrix of the data in the original coordinate system, in which the data was originally expressed, will most likely be non-diagonal. However, by aligning the new axis with eigenvectors of the original covariance matrix, the data will be expressed in such a way that the new covariance matrix will be diagonal. Not only that, as a bonus point we get that the diagonal entries of the new covariance matrix, which of course correspond to variances along the new axes, are exactly the eigenvectors of the original covariance matrix. We are now ready to formulate the Principal Component Analysis algorithm.

5 Principal Component Analysis (PCA)

We finally have the tools we need in order to perform PCA on a given data set $\mathcal{D} = \{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^N\}$ in d-dimensional space:

1) Using the data, calculate the co-variance matrix $\mathbf{C}$.

2) Calculate the eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_d$ of $\mathbf{C}$ (vectors for our new axes) and their corresponding eigenvalues $\lambda_1, \ldots, \lambda_d$ (the variability along each new axis), such that $||\mathbf{v}_i|| = 1$ for every i and $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d \geq 0$.[2]

3) Pick the first k of those eigenvectors such that a proportion of variability is ‘sufficiently’ preserved. We can find the fraction of data variability preserved in the first k dominant axes, ρ, using the following expression:

$$\rho = \frac{\lambda_1 + \lambda_2 + \ldots + \lambda_k}{\lambda_1 + \lambda_2 + \ldots + \lambda_d}$$

Suppose that we want to preserve a proportion ρ' of variability. Then choose k eigenvectors such that $\rho \geq \rho'$.

4) Project our data onto the new axes by using

$$(\mathbf{V}^{(k)})^T\mathbf{x}^i = \tilde{\mathbf{x}}^i,$$

where $\mathbf{V}^{(k)} = [\mathbf{v}_1, \ldots, \mathbf{v}_k]$.

[2] This ordering can be achieved in packages like Octave or MATLAB which saves us the task of doing it manually.