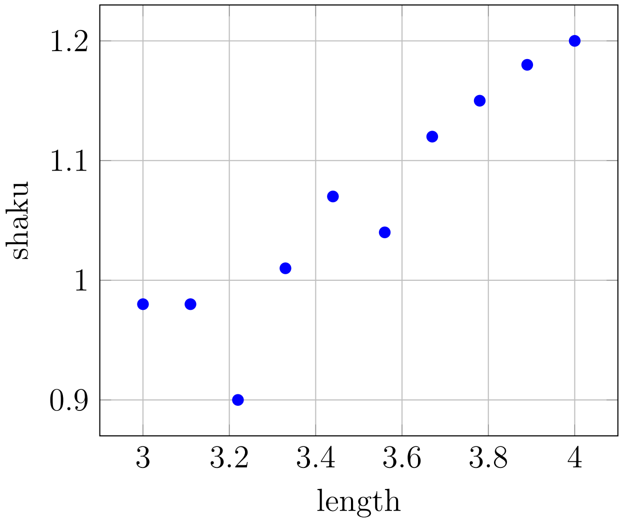
Dimension Reduction

Principal Component Analysis(PCA)

One of a common problem we see in data analysis is the problem is too much information. For example, let us consider the problem of handwritten digit recognition. Suppose the image is 50×50, you will have 2500 features. But If we consider a random 50×50 image, most of them are going to be just random picture and not a number. So the underlying dimension is most likely less than 2500.

Recall very first example we had with Bayes classifier. The more feature we have the exponentially more data we need to do a good fit. What we are gonna learn here is how to reduce the dimensinoality of the data.

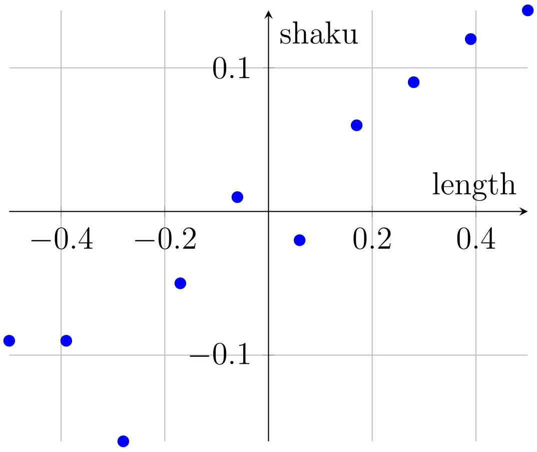
Let us consider the example of data with two features: length and shaku. The plot of the data is shown below.



Normalizing Data

To make the math simpler later on, let us normalize the data such that the mean of each variable is zero. This can be done by subtract each column by its mean. The normalized data and the original data contains exactly the same information. There is no loss of information here. The plot of the data above after zero-out the mean is shown below.

Some people also normalize it so that the standard deviation is one to make all feature lies on the same scale. This can be done by divide the number by the standard deviation. But we are not going to do it here.



As expected the shape looks the same. So far the data pipe line looks like the following.

$$\text{Raw Data} \longrightarrow \boxed{\text{Normalize}} \longrightarrow \text{Normalized Data}$$

Correlation

As the plot shown, whenever the length is above the mean the shaku is also above the mean. Whenever the length is below the mean, the shaku is also below the mean. This kind of behavior is call correlated. We can define a measure for how correlated the two variables are using covariance.

Let X and Y be two random variable(think about it as features), the covariance of variable X and Y is defined by

$$\mathrm{Cov}(X, Y) = \mathrm{E}[(X - \mathrm{E}[X])\,(Y - \mathrm{E}[Y])] \tag{1}$$

where E is the expected value.

To make sense out of this quantity. Let us consider the sign of $(X - \mathrm{E}[X])\,(Y - \mathrm{E}[Y])$. The sign of the term $(X - \mathrm{E}[X])$ is positive for each data point which X is above the mean and negative if it is below the mean. Similarily, the sign of the term $(Y - \mathrm{E}[Y])$ measure is positive if Y of that data point is above the mean and negative if it is below the mean.

If for each data point, the feature X and Y always go above and below the mean together. Each data point will contribute positively to the $\mathrm{Cov}(X, Y)$. Making the value of covariance really big.

On the other hands, in the situation whrere for each data point when X goes above the mean Y always go below the mean and when X goes below the mean Y goes above the mean. Each data point will contribute negatively to the $\mathrm{Cov}(X, Y)$. Making the value of covariance very negative. Such beahvior is called **anticorrelated**. Both correlated situation and anticorrelated situation indicate that there must be an underlying commonality between the two variable.

If the two data goes above and below the mean, irregardless of each other. Some of data points will contribute postive and some will contribute negatively. The two will cancel out each other and make the $\mathrm{Cov}(X, Y)$ small value. In the case where the $\mathrm{Cov}(X, Y)$ is 0 we say that X and Y are independent.

One useful fact of the covaraiace is the covariance of the variable against itself is the variance of that variable.

$$\mathrm{Cov}(X, X) = \mathrm{E}[(X - \mathrm{E}[X])\,(X - \mathrm{E}[X])] = \mathrm{E}[(X - \mathrm{E}[X])^2] = \mathrm{Var}[X] \tag{2}$$

Another useful fact is what covariance looks like when $\mathrm{E}[X] = \mathrm{E}[Y] = 0$.

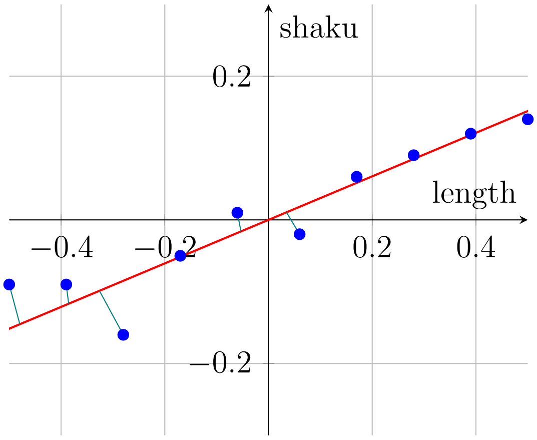
$$\mathrm{Cov}(X, Y) = \mathrm{E}[(X - 0)(Y - 0)] = \mathrm{E}[XY] \tag{3}$$

Rotated Feature

In fact the "shaku" is a japanese length unit. The two varaible are essentially the same. There are correlated. There are some noise since may be the rulers used to measure comes from two different factories. There is really 1 underlying variable, length, not 2.

So our goal is to capture that underlying variable. One way to do it is just pick one variable but doing so will make you lose a bit too much information since we don't really know which ruler is the more accurate one.

We can do a bit better by just consider a "rotated variable". This concept is illustrated with the new red axis shown in the figure below[1].



Recall that each data point is a vector. Each component is just the projection of the vector onto each basis direction. So, to write it in a terms of new red axis all we need to find is the projection of each vector onto red axis direction. This calls for dot product. Let $\hat{e}_1$ be the unit vector along the new rotated red axis. Then the new rotated variable is given by

$$x_1' = \hat{e}_1 \cdot (\text{length}, \text{shaku}) \tag{4}$$

For example, consider a datation whose length and shakku is given[2] (-0.50, -0.09). The unit vector along red axis is[3] given by (0.96, 0.29)[4]. Then, the rotated variable for this data point is

$$x_1' = (0.96, 0.29) \cdot (-0.50, -0.09) = -0.505 \tag{5}$$

This number represent the length of the projection of this data point along the red axis.

This rotated is special in the sense that it captures the underlying variable. The reason is that this the rotated variable on this axis has the **highest** variance/variation. Projection of variable along other axis will not give as high variance. Almost all the variation in the data can be explained by this new rotated variable.

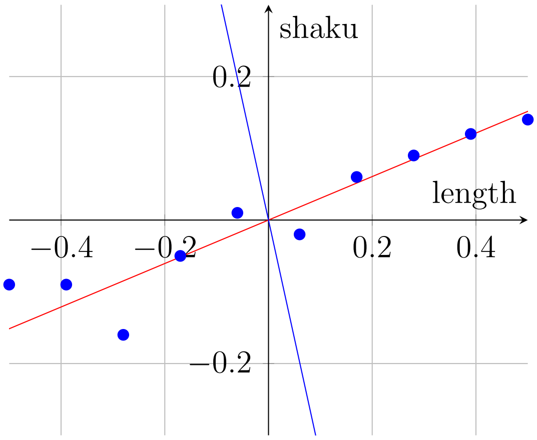
[1] All the teal lines are supposed to be perpendicular to the blue line. Will fix this later.

[2] After normalization

[3] in $\hat{x}$, $\hat{y}$

[4] We will learn how to find this magical axis later.

To complete this section, if we only use one rotated variable, of course we are going to lose information. After all we went from 2 dimension to just 1 dimension. We need another axis shown in blue. The unit vector along that axis $\hat{e}_2 = [0.29, -0.96]$. This axis will have less variance than the blue axis but it is necessary to completely preserve the information. One can go back and forth between the standard axis and the red-blue axis.



The component of the data point $(-0.50, -0.09)$ along the new blue axis is given by

$$x'_2 = \hat{e}_2 \cdot (\text{length}, \text{shaku}) \tag{6}$$
$$x'_2 = (0.29, -0.96) \cdot (-0.50, -0.09) \cdot = -0.059 \tag{7}$$

We can combine Equation 4 and 6 and write it in matrix form[5]

$$\begin{bmatrix} x'_1 \\ x'_2 \end{bmatrix} = \begin{bmatrix} \cdots \hat{e}_1^T \cdots \\ \cdots \hat{e}_2^T \cdots \end{bmatrix} \times \begin{bmatrix} \text{length} \\ \text{shaku} \end{bmatrix} = \begin{bmatrix} 0.96 & 0.29 \\ 0.29 & -0.96 \end{bmatrix} \times \begin{bmatrix} \text{length} \\ \text{shaku} \end{bmatrix} \tag{8}$$

In summary, we have learn that data is nothing but vector and we can convert one to another. Our goal is to convert it to the basis which we can rank it by the amount of variance each axis capture.

Variance Along Axis

As we can see from the last section, there is some axis which capture the most variation in the data. Our goal now is to figure out how to find such axis from the data point. This axis is special in the sense that the projection of the data points on this axis has the most variance.

So, let us write variance along an arbitary axis defined by unit vector $\hat{e}$ for datapoints $\vec{x}^{(i)}$

$$\text{Variance along } \hat{e} = V(\hat{e}) = \text{E}[(\vec{x} \cdot \hat{e} - \text{E}[\vec{x} \cdot \hat{e}])^2] \tag{9}$$

Remember that since we **normalize** the data making the mean of each feature 0 in the beginning. This means

$$\text{E}[\vec{x} \cdot \hat{e}] = \text{E}[\vec{x}] \cdot \hat{e} = \vec{0} \cdot \hat{e} = 0$$

So, Equation 9 becomes

[5] You may notice that the matrix has nice structure. This is actually not a coincidence. It has a bunch of nice properties. It is an orthorgonal matrix.

$$V(\hat{e}) = \mathrm{E}[(\vec{x} \cdot \hat{e})^2] \tag{10}$$

with constraint that the length of $\hat{e}$ is 1. That is

$$\hat{e} \cdot \hat{e} - 1 = 0 \tag{11}$$

Using vector notation gives us a nice looking concise equation but it is kind of hard to see what is going on. So let us bring back the index sum. Let $x_j^{(i)}$ be the the j-th feature of the i-th data point. Also, let e_j be the component of $\hat{e}$ along j-th feature axis. The two equations above becomes

$$V(\hat{e}) = \frac{1}{M} \sum_i \left(\sum_j x_j^{(i)} e_j \right)^2 \tag{12}$$

where M is the number of data points and the unit vector constraint becomes

$$\sum_j e_j^2 - 1 = 0 \tag{13}$$

Our goal is then to find all the e_j that maximize Equation 12 subject to constraint indicated in 13

Constrained Minimization and Lagrange Multiplier

The problem we had in Equation 12 and 13 is called constrained minimization problem. That is we want to

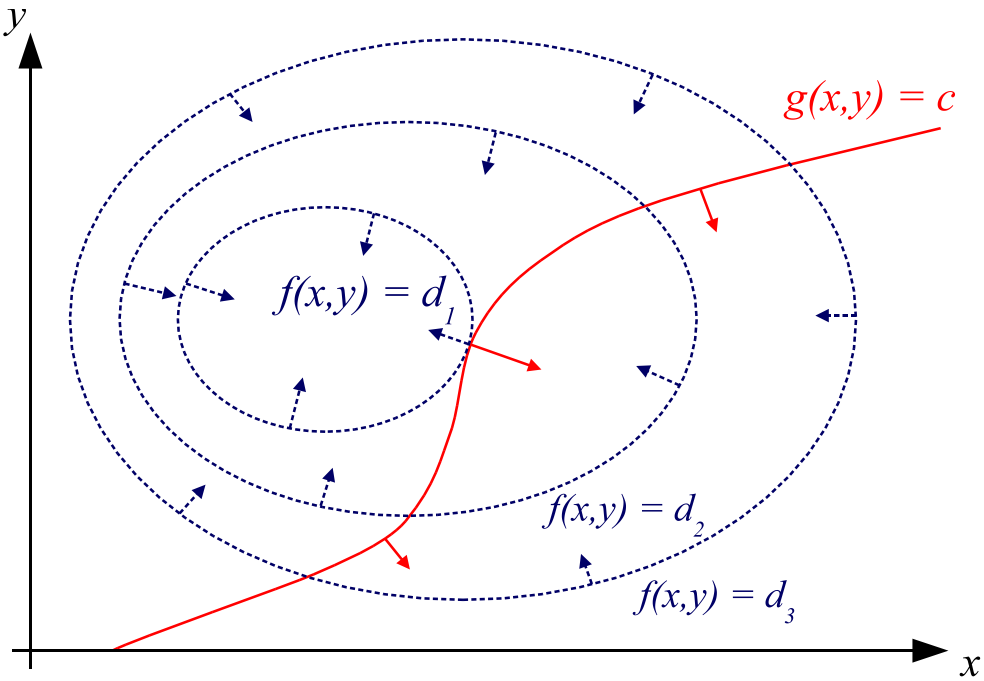
- minimize $f(\vec{x})$
- subject to $g(\vec{x}) = c$

Without the constrained all we need to do is to find $\vec{\nabla} f = 0$. This will give us n equation and n unknown where n is the number of dimension. We can solve for this.

But with constrain we can't really do that since the minimum with the constrained may not happen at the true minimum of $f(\vec{x})$.

To solve this kind of problem we need a tool called Lagrange multiplier. The idea comes from a simple yet powerful observation shown in the figure below[6].

[6] From wikipedia.https://upload.wikimedia.org/wikipedia/commons/b/bf/LagrangeMultipliers2D.svg



The important observation is that at the point of minimum $\vec{\nabla} f$ and $\vec{\nabla} g$ are parallel. The argument goes like that if they are not parallel then some perturbation of x on that point will make f less in one direction and more in the other direction. Meaning that, at the point of minimum,

$$\vec{\nabla} f(\vec{x}) = \lambda \vec{\nabla} g(\vec{x}) \tag{14}$$

The constant λ is called Lagrange multiplier.

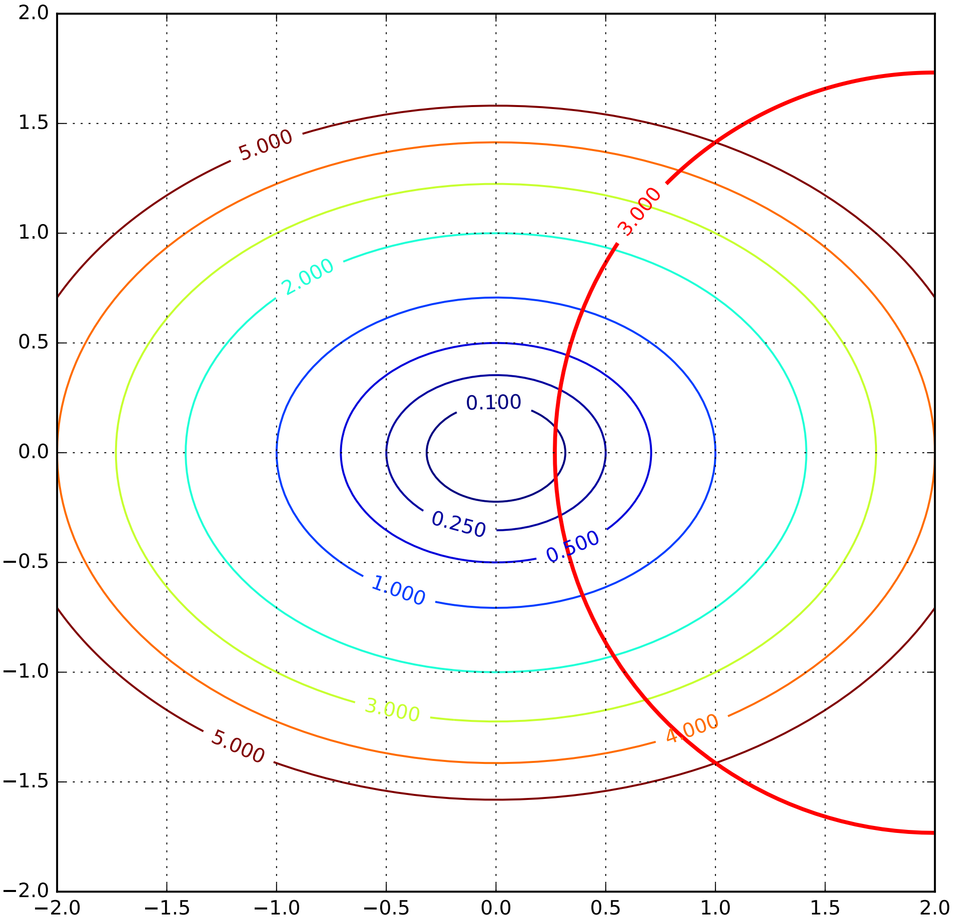
Let us count the number of equation and unknowns. First the number of unknown is n+1. n is from $\vec{x}$ and extra one from λ. Equation 14 contains n equations; one for each dimension. The constrain then add another one. So the number of equation and unknown match which means we can solve for $\vec{x}$ and λ

Example

Let us do one example. Let us try to minimize

- $f(\vec{x}) = x^2 + 2y^2$
- subject to $g(\vec{x}) = (x-2)^2 + y^2 = 3$

The plot of $f(\vec{x})$ (rainbow color) and the line $g(\vec{x}) = 3$(red line) is shown below.



So using Equation 14 we have

$$\vec{\nabla} f(\vec{x}) = \lambda \vec{\nabla} g(\vec{x}) \tag{15}$$
$$2x\hat{x} + 4y\hat{y} = 2\lambda(x-2)\hat{x} + 2\lambda y\hat{y} \tag{16}$$

The equation for $\hat{x}$ and $\hat{y}$ component and the constraint becomes

$$x = \lambda(x-2) \tag{17}$$
$$y = \lambda y \tag{18}$$
$$(x-2)^2 + y^2 = 3 \tag{19}$$

Equation 18 implies

$$\lambda = 1 \text{ or } y = 0$$

Pluggin $\lambda = 1$ into Equation 17, we can see that $\lambda = 1$ can't satisfy it. So, our answer would be $y = 0$. Pluggin $y = 0$ into Equation 19 gives

$$(x-2)^2 = 3 \rightarrow x = 2 \pm \sqrt{3}$$

You could continue to find λ but we already got what we want x and y. So the extremum are at

$$x = 2 + \sqrt{3}, y = 0 \text{ and } x = 2 - \sqrt{3}, y = 0$$

If you look at the picture we do expect 2 answers. One of them is maximum and the other is minimum.

$$
\begin{align}
x = 2 - \sqrt{3}, y = 0 \rightarrow f(\vec{x}) = 0.07 \tag{20}\\
x = 2 + \sqrt{3}, y = 0 \rightarrow f(\vec{x}) = 13.9 \tag{21}\\
\tag{22}
\end{align}
$$

Special Axis

Let us use lagrange multiplier to solve the problem at hand: Equation 12 and 13.

- minimize

$$
V(\hat{e}) = \frac{1}{M}\sum_i \left(\sum_j x_j^{(i)} e_j\right)^2 \tag{23}
$$

- subject to

$$
g(\hat{e}) = \left(\sum_j e_j^2\right) - 1 = 0 \tag{24}
$$

So all we need to do is to use lagrange multiplier. But first let us find ∇g. Since this is the first time we are taking derivative with respect to vector component. Let's write it down explicitly. Let n be number of dimension(number of features).

$$
g(\hat{e}) = g(e_1, e_2, e_3, \ldots) = e_1^2 + e_2^2 + \ldots + e_n^2
$$

Writing it down this ways makes it easy to find the gradient

$$
\begin{align}
\frac{\partial g}{\partial e_1} = 2e_1 \tag{25}\\
\vdots \tag{26}\\
\frac{\partial g}{\partial e_n} = 2e_n \tag{27}
\end{align}
$$

So we write

$$
\begin{bmatrix} \frac{\partial g}{\partial e_1} \\ \frac{\partial g}{\partial e_2} \\ \frac{\partial g}{\partial e_3} \\ \vert \end{bmatrix} = 2 \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ \vert \end{bmatrix} \tag{28}
$$

Always, when you have to take index of some crazy matrix expression and get confused just expand it and you will see what you need to do.

Now let's take partial derivative of $V(\hat{e})$

$$\begin{aligned}
\frac{\partial V(\hat{e})}{\partial e_k} &= \frac{1}{M}\frac{\partial}{\partial e_k}\sum_i\left(\sum_j x_j^{(i)}e_j\right)^2 && (29)\\
&= \frac{1}{M}\frac{\partial}{\partial e_k}\sum_i\left(x_1^{(i)}e_1+\ldots x_n^{(i)}e_n\right)^2 && (30)\\
&= \frac{1}{M}\sum_i\frac{\partial}{\partial e_k}\left(x_1^{(i)}e_1+\ldots x_n^{(i)}e_n\right)^2 && (31)\\
&= \frac{1}{M}\sum_i 2\left(x_1^{(i)}e_1+\ldots x_n^{(i)}e_n\right)x_k^{(i)} && (32)\\
&= 2\frac{1}{M}\sum_i\sum_j x_j^{(i)}e_jx_k^{(i)} && (33)\\
&= 2\frac{1}{M}\sum_j\sum_i x_j^{(i)}e_jx_k^{(i)} \quad \text{note the index of the sum} && (34)\\
&= 2\sum_j\frac{\left(\sum_i x_j^{(i)}x_k^{(i)}\right)}{M}e_j && (35)
\end{aligned}$$

Let us pause here for a bit the last line here is much nicer if we write it in term of matrix equation

$$\begin{bmatrix}\frac{\partial V}{\partial e_1}\\ \frac{\partial V}{\partial e_2}\\ \frac{\partial V}{\partial e_3}\\ \vert\end{bmatrix} = \begin{bmatrix}\frac{1}{M}\sum_i x_1^{(i)}x_1^{(i)} & \frac{1}{M}\sum_i x_2^{(i)}x_1^{(i)} & \frac{1}{M}\sum_i x_3^{(i)}x_1^{(i)} & \text{—}\\ \frac{1}{M}\sum_i x_1^{(i)}x_2^{(i)} & \frac{1}{M}\sum_i x_2^{(i)}x_2^{(i)} & \frac{1}{M}\sum_i x_3^{(i)}x_2^{(i)} & \text{—}\\ \frac{1}{M}\sum_i x_1^{(i)}x_2^{(i)} & \frac{1}{M}\sum_i x_2^{(i)}x_2^{(i)} & \frac{1}{M}\sum_i x_3^{(i)}x_2^{(i)} & \text{—}\\ \vert & \vert & \vert & \end{bmatrix}\begin{bmatrix}e_1\\ e_2\\ e_3\\ \vert\end{bmatrix} \tag{36}$$

Let's simplify the scarying looking matrix in the middle interms of expected value

$$\begin{bmatrix}\frac{\partial V}{\partial e_1}\\ \frac{\partial V}{\partial e_2}\\ \frac{\partial V}{\partial e_3}\\ \vert\end{bmatrix} = \begin{bmatrix}\mathrm{E}[x_1x_1] & \mathrm{E}[x_2x_1] & \mathrm{E}[x_3x_1] & \text{—}\\ \mathrm{E}[x_1x_2] & \mathrm{E}[x_2x_2] & \mathrm{E}[x_3x_2] & \text{—}\\ \mathrm{E}[x_1x_3] & \mathrm{E}[x_1x_3] & \mathrm{E}[x_3x_3] & \text{—}\\ \vert & \vert & \vert & \end{bmatrix}\begin{bmatrix}e_1\\ e_2\\ e_3\\ \vert\end{bmatrix} \tag{37}$$

The scary looking huge matrix in the middle is nothing but the covariance between feature x_j and x_j as in Equation 3. This matrix is called **covariance matrix**. This is also why we subtract off the mean in the first place.

Let us use lagrange multiplier and combine this with Equation 28. We got

$$\begin{bmatrix}\frac{\partial V}{\partial e_1}\\ \frac{\partial V}{\partial e_2}\\ \frac{\partial V}{\partial e_3}\\ \vert\end{bmatrix} = \begin{bmatrix}\mathrm{E}[x_1x_1] & \mathrm{E}[x_2x_1] & \mathrm{E}[x_3x_1] & \text{—}\\ \mathrm{E}[x_1x_2] & \mathrm{E}[x_2x_2] & \mathrm{E}[x_3x_2] & \text{—}\\ \mathrm{E}[x_1x_3] & \mathrm{E}[x_1x_3] & \mathrm{E}[x_3x_3] & \text{—}\\ \vert & \vert & \vert & \end{bmatrix}\begin{bmatrix}e_1\\ e_2\\ e_3\\ \vert\end{bmatrix} = \lambda\begin{bmatrix}\frac{\partial g}{\partial e_1}\\ \frac{\partial g}{\partial e_2}\\ \frac{\partial g}{\partial e_3}\\ \vert\end{bmatrix} \tag{38}$$

or

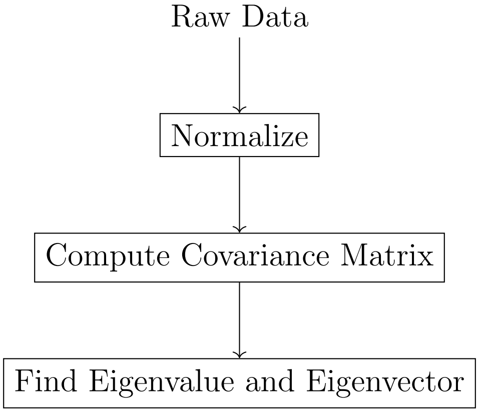
$$\begin{bmatrix} \mathrm{E}[x_1x_1] & \mathrm{E}[x_2x_1] & \mathrm{E}[x_3x_1] & — \\ \mathrm{E}[x_1x_2] & \mathrm{E}[x_2x_2] & \mathrm{E}[x_3x_2] & — \\ \mathrm{E}[x_1x_3] & \mathrm{E}[x_1x_3] & \mathrm{E}[x_3x_3] & — \\ | & | & | & \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ | \end{bmatrix} = 2\lambda \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ | \end{bmatrix} \tag{39}$$

And this is just **eigenvalue problem** we have seen before. So, we can conclude that **eigenvector of covariance matrix** is the direction which gives the maximum variance. We can write this more succinctly using vector notation

$$\mathbf{C}\hat{e} = 2\lambda\hat{e} \tag{40}$$

where **C** is the covariance matrix.

To recap, so far we have this.



Eigenvalue is the Variance along Eigenvector Direction

But as we have seen before there are many eigenvectors which one do we pick? In this section we are going to rank eigenvectors by its variance.

Now that we know that $\hat{e}$ is an eigenvector of covariance matrix. Let us find the variance starting with equation 12.

$$\begin{aligned} V(\hat{e}) &= \frac{1}{M}\sum_i \left(\sum_j x_j^{(i)} e_j\right)^2 && (41) \\ &= \frac{1}{M}\sum_i \left(\sum_j x_j^{(i)} e_j\right)\left(\sum_k x_k^{(i)} e_k\right) && (42) \\ &= \frac{1}{M}\sum_i\sum_j\sum_k x_j^{(i)} e_j x_k^{(i)} e_k && (43) \\ &= \sum_k\sum_j \left(\frac{1}{M}\sum_i x_j^{(i)} x_k^{(i)}\right) e_j e_k && (44) \end{aligned}$$

The term in the middle parenthesis is just a covariance matrix

$$V(\hat{e}) = \sum_k \sum_j C_{jk} e_j e_k \tag{45}$$

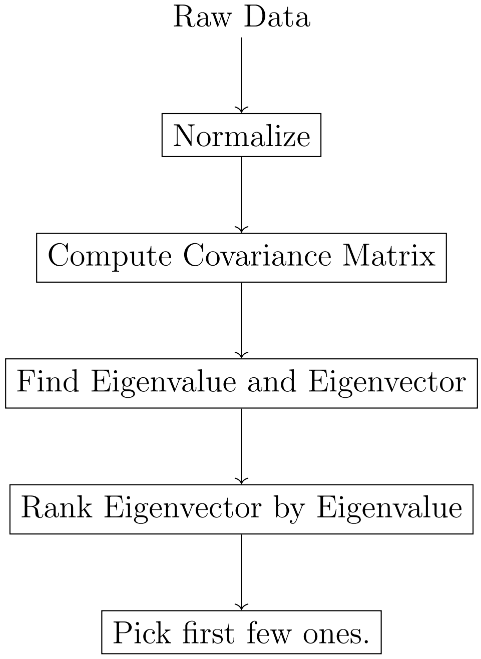
The sum then be comes

$$V(\hat{e}) = \hat{e}^T \mathbf{C} \hat{e} \tag{46}$$

Since $\hat{e}$ is an eigenvector of C $C\hat{e} = \lambda\hat{e}$[7]

$$V(\hat{e}) = \lambda \hat{e}^T \hat{e} = \lambda \tag{47}$$

All these manipulation just shows that the **variance along each eigenvector axis is just the corresponding eigenvalue**. This tells us something very important. Then, after we find the eigenvector for the covariance matrix. All we need to do is rank them by corresponding eigenvalue. First few one will most likely capture almost all the variance.


Raw Data
Normalize
Compute Covariance Matrix
Find Eigenvalue and Eigenvector
Rank Eigenvector by Eigenvalue
Pick first few ones.

New Variable

After we get first few eigenvectors $\{\hat{e}_1, \hat{e}_2, \ldots, \hat{e}_m\}$ we need to use these to turn the raw data points into new data points which are the component/projection along each eigenvector direction. So that means we need to do the following transformation to the raw data.

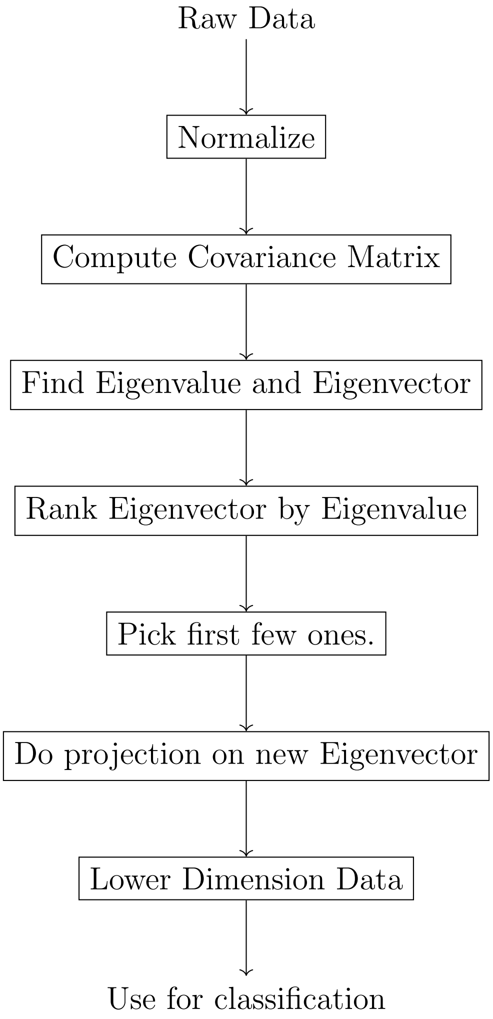
$$\vec{x}^{(i)} \longrightarrow (\vec{x}^{(i)} \cdot \hat{e}_1, \vec{x}^{(i)} \cdot \hat{e}_2, \ldots \vec{x}^{(i)} \cdot \hat{e}_m)$$

Then we can pass this new variable to train classifier. Note that since we do a transformation on the training data. When we want to use the classifier trained with this transformed data, we will need to do the same transformation before giving it to classifier.

[7] This lambda is a different from λ we use in 39 by a factor of two. I use λ here for an eigen value since this is what we typically got from numpy.

Summary

We have shown that the eigenvector of the covriance matrix are those axis are the axis in which we want to describe our data in. We can also rank each axis by its eigenvalue(which is variance). The summary of the process is shown below.


Raw Data
Normalize
Compute Covariance Matrix
Find Eigenvalue and Eigenvector
Rank Eigenvector by Eigenvalue
Pick first few ones.
Do projection on new Eigenvector
Lower Dimension Data
Use for classification

Remember that since we do a data transformation before giving it to classifier. This means that for new data, we need to do the same transformation as we did before giving it to classifier.