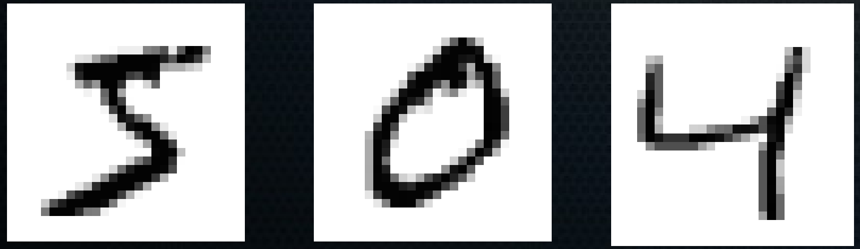
Dimensionality reduction

- Manipulate dimensions in a smart way to capture the data as good as possible in lower dimensions.
- Two main flavours:
  - Linear
  - Non-linear

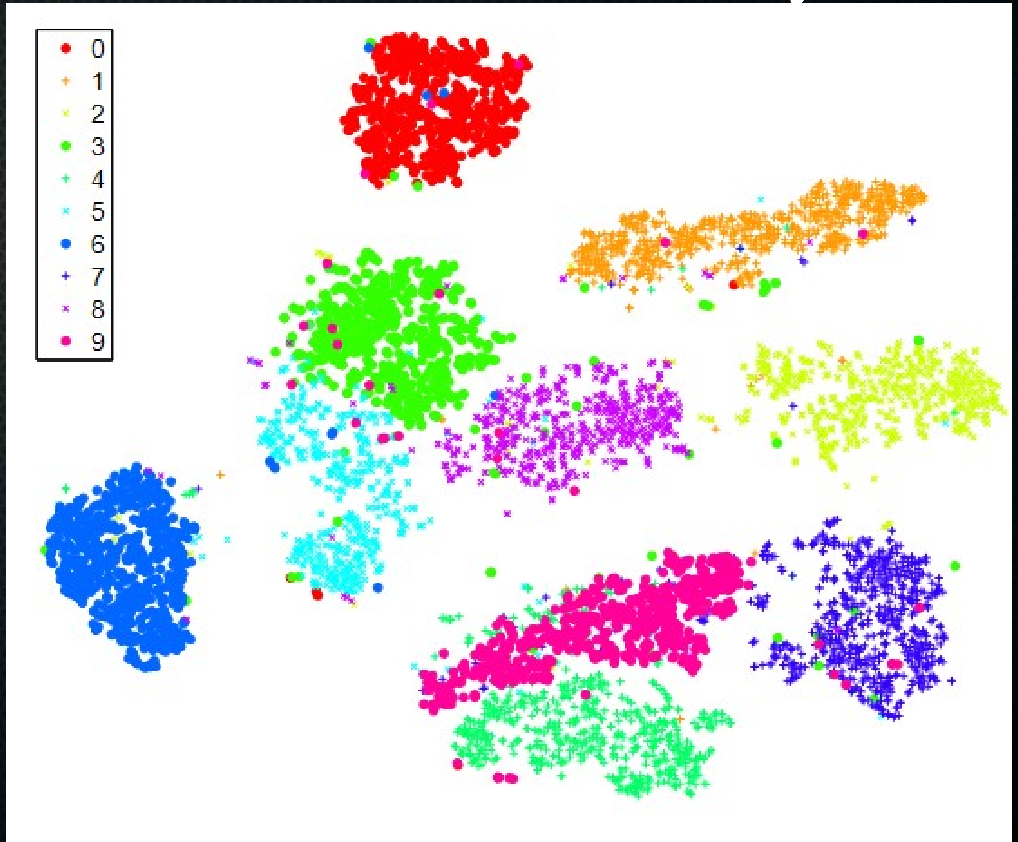
Dimensionality reduction: non-linear

- Extrinsic dimensionality may be high, but the data we care about might lie in a specific subspace of lower dimensionality.

- MNIST:
  28*28 = 784-dimensional data



Source: http://colah.github.io/posts/2014-10-Visualizing-MNIST/

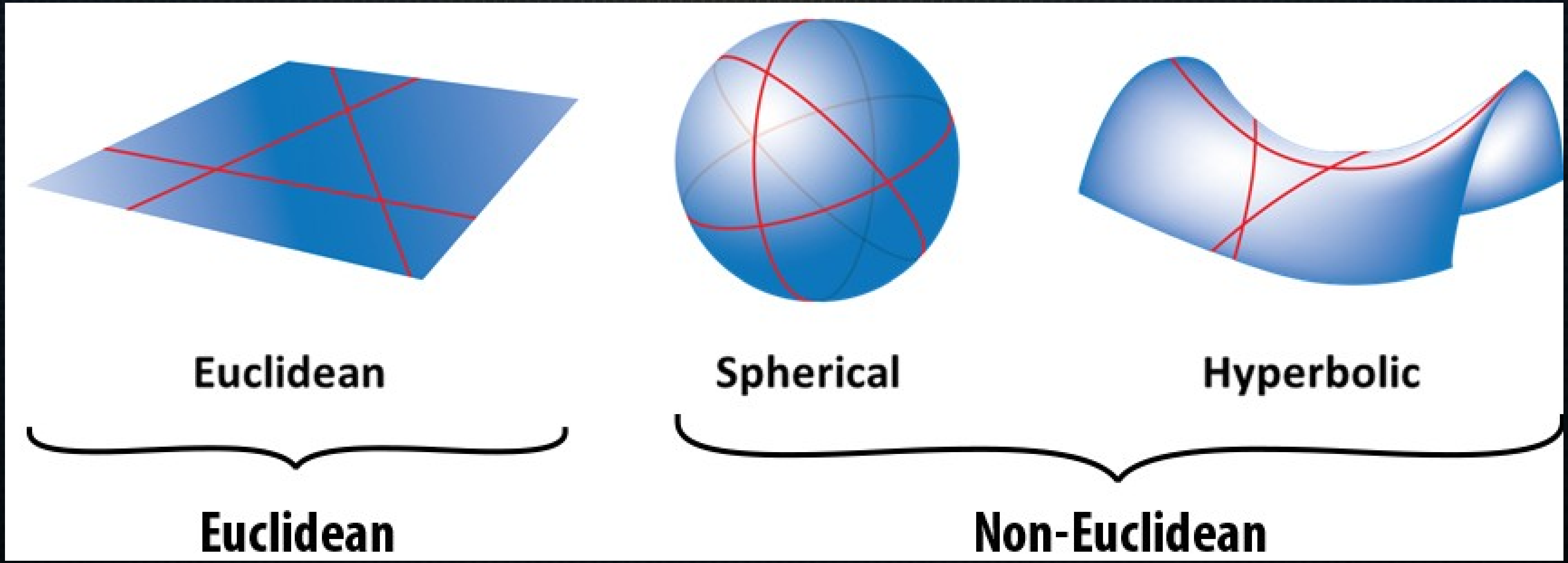


(a) Visualization by t-SNE.

Source: Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. Journal of machine learning research, 9(11).

Dimensionality reduction: non-linear

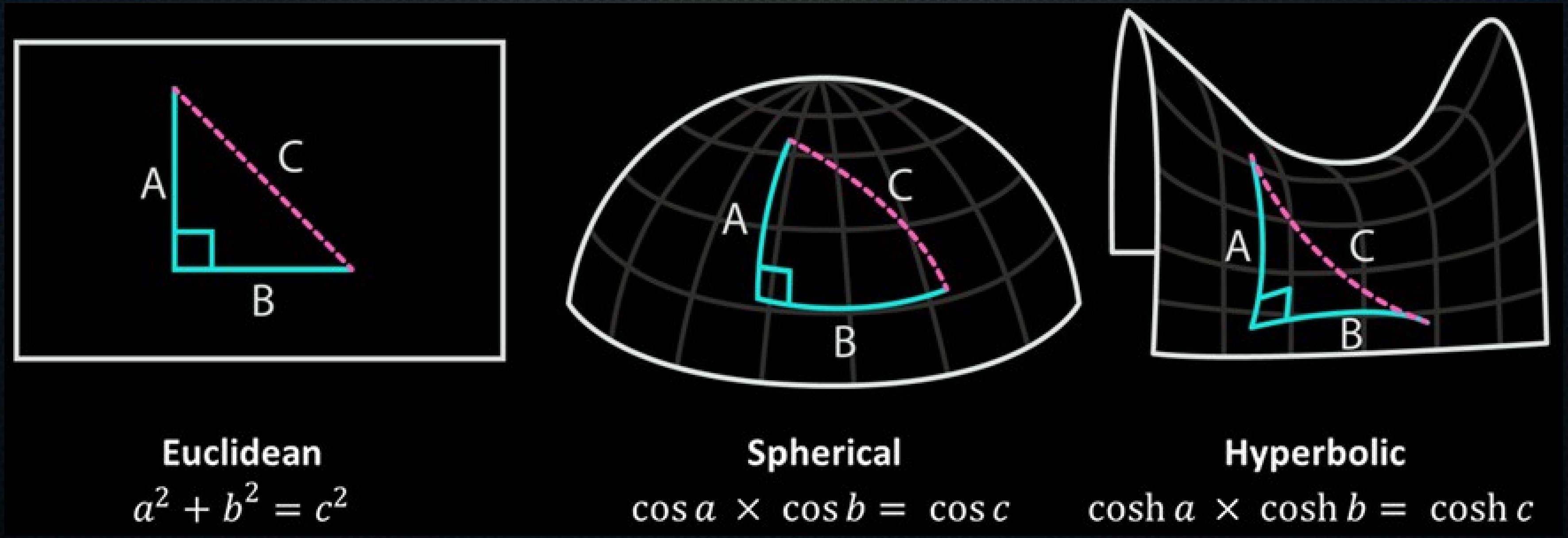
- Won't be covering this in detail.
- Based on the idea of a manifold. Non-Euclidean space can be locally approximated with Euclidean space.



Source: https://static1.squarespace.com/static/56ee72d9c2ea51bd675641da/t/57fdaf651b631b13d85fe0ac/1476243320274/

Dimensionality reduction: non-linear

- Won't be covering this in detail.
- Based on the idea of a manifold. Non-Euclidean space can be locally approximated with Euclidean space.



Source: https://images.squarespace-cdn.com/content/v1/56ee72d9c2ea51bd675641da/1476242359445-8N16VQ6VAQT38XXNLD8Q/ke17ZwdGBToddI8pDm48kPQNnkEdFtON3V0mnbpTD-lZw-zPPgdn4jUwVcJE1ZvWQUxwkmyExglNqGp0IvTJZamWLI2zvYWH8K3-s_4yszcp2ryTI0HqTOaaUohrI8PIPKdhB7HsyVNaCLrIgM8dFwbZx21t-vKmnfsWocTJKSAKMshLAGzx4R3EDFOm1kBS/image-asset.jpeg

Dimensionality reduction: non-linear

- Won't be covering this in detail.
- Based on the idea of a manifold. Non-Euclidean space can be locally approximated with Euclidean space.

Source: https://www.mespetitespuces.com/wp-content/uploads/2017/09/1-1.jpg

Source: http://www.globalsecurity.org/military/world/war/images/map-south-america-2.jpg

Dimensionality reduction: non-linear

- Won’t be covering this in detail.
- Based on the idea of a manifold. Non-Euclidean space can be locally approximated with Euclidean space.

Source: https://www.mespetitespuces.com/wp-content/uploads/2017/09/1-1.jpg

Source: http://www.globalsecurity.org/military/world/war/images/map-south-america-2.jpg

The globe is non-Euclidian, and any place on the 3D globe can only be described fully with 3 coördinates

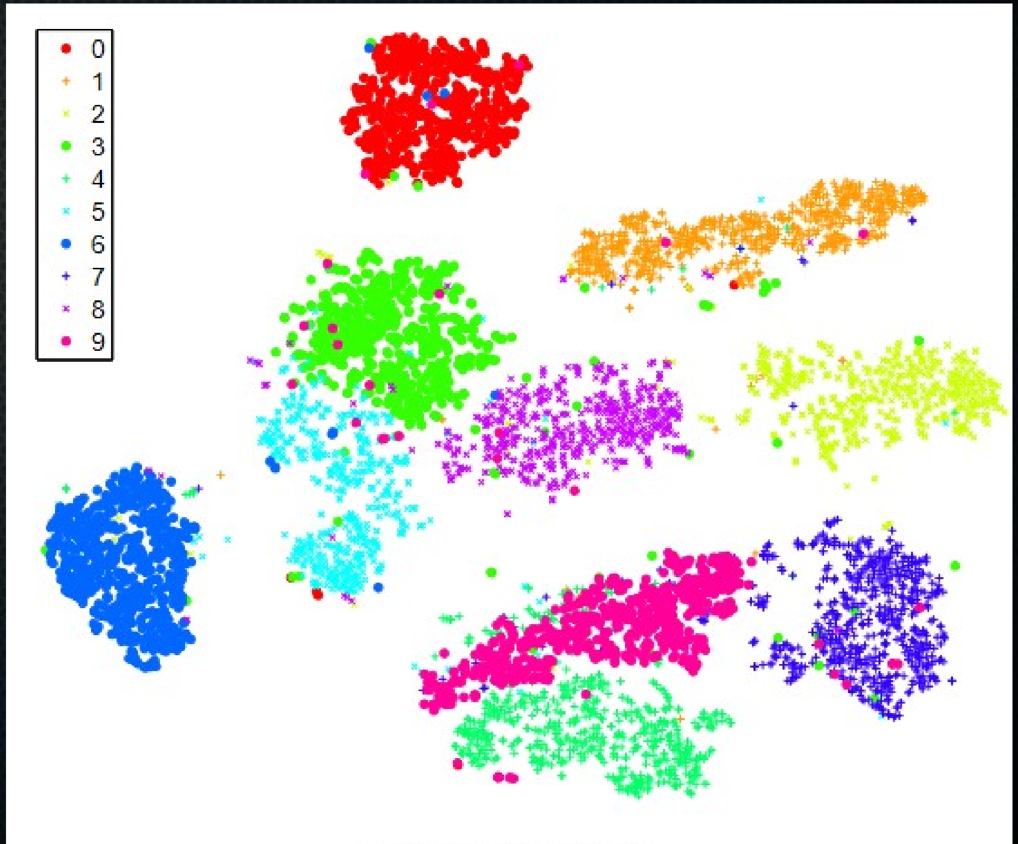
But if we want to discern separate parts of South America, we can do that just fine on an Euclidean projection!

Dimensionality reduction: non-linear

- Won’t be covering this in detail.
- Based on the idea of a manifold. Non-Euclidean space can be locally approximated with Euclidean space.
- Mostly try to preserve local structure (don’t care about global characteristics of space, just that low-dimensional distances are similar to high-dimensional distances)

Dimensionality reduction: non-linear

- t-SNE
- Colours correspond to digit groups that are distinct in high-D space
- Distances between these groups in low-D are meaningless (global distances not preserved)



(a) Visualization by t-SNE.

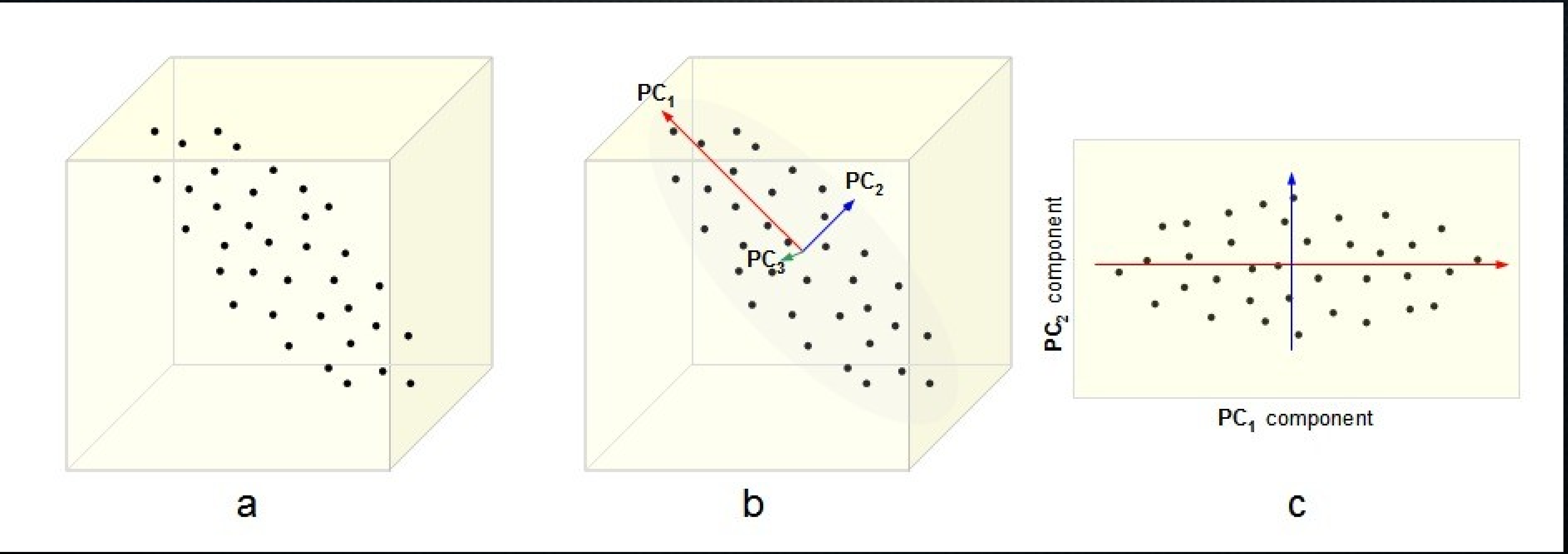
Source: Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. Journal of machine learning research, 9(11).

Dimensionality reduction: linear

- Concepts to understand PCA:
  - Covariance matrix
  - Eigenvectors and determinant
  - Projection and selection #of PCs

Dimensionality reduction: linear

- Idea: rather than selecting a subset of features, we make linear combinations of all existing features (and then select a subset from those to reduce dimensionality)
- Most-used: PCA



Source: https://cnx.org/resources/269e78e2506cdbdc7f45ca17bb7d83491ce0c644/pca.jpg

Dimensionality reduction: PCA

- Formally: want to make some mapping from original data X to projected data Y.

Dimensionality reduction: PCA

- Formally: want to make some mapping from original data X to projected data Y.

One data point: 1*n row vector

$$\begin{bmatrix} \text{feat}_1 & \text{feat}_2 & \ldots & \text{feat}_n \end{bmatrix}$$

Dimensionality reduction: PCA

- Formally: want to make some mapping from original data X to projected data Y.

One data point: 1*n row vector

$$\begin{bmatrix} \text{feat}_1 & \text{feat}_2 & \ldots & \text{feat}_n \end{bmatrix}$$

Full data: m*n matrix

$$\mathrm{X} = \begin{bmatrix} \text{feat}_{1,1} & \text{feat}_{2,1} & \ldots & \text{feat}_{n,1} \\ \text{feat}_{1,2} & \text{feat}_{2,2} & \ldots & \text{feat}_{n,2} \\ \ldots & \ldots & \ldots & \ldots \\ \text{feat}_{1,m} & \text{feat}_{2,m} & \ldots & \text{feat}_{n,m} \end{bmatrix}$$

Dimensionality reduction: PCA

- Formally: want to make some mapping from original data X to projected data Y.

One data point: 1*n row vector

$$\begin{bmatrix} feat_1 & feat_2 & \ldots & feat_n \end{bmatrix}$$

$$l < n$$

$$\begin{bmatrix} feat_1 & feat_2 & \ldots & feat_l \end{bmatrix}$$

Full data: m*n matrix

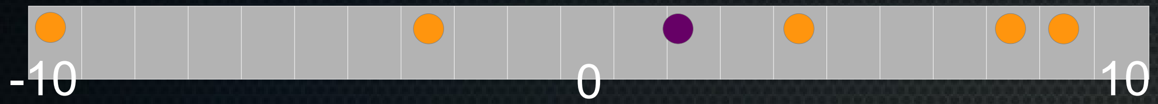
$$X = \begin{bmatrix} feat_{1,1} & feat_{2,1} & \ldots & feat_{n,1} \\ feat_{1,2} & feat_{2,2} & \ldots & feat_{n,2} \\ \ldots & \ldots & \ldots & \ldots \\ feat_{1,m} & feat_{2,m} & \ldots & feat_{n,m} \end{bmatrix} \longrightarrow Y = \begin{bmatrix} feat_{1,1} & feat_{2,1} & \ldots & feat_{l,1} \\ feat_{1,2} & feat_{2,2} & \ldots & feat_{l,2} \\ \ldots & \ldots & \ldots & \ldots \\ feat_{1,m} & feat_{2,m} & \ldots & feat_{l,m} \end{bmatrix}$$

Dimensionality reduction: PCA

- So, how do we do it?
- To understand that, first need to look at the determinant of a matrix and the covariance matrix

Covariance matrix

- Variance= average sum of square differences between a feature and its mean, spread of the data.

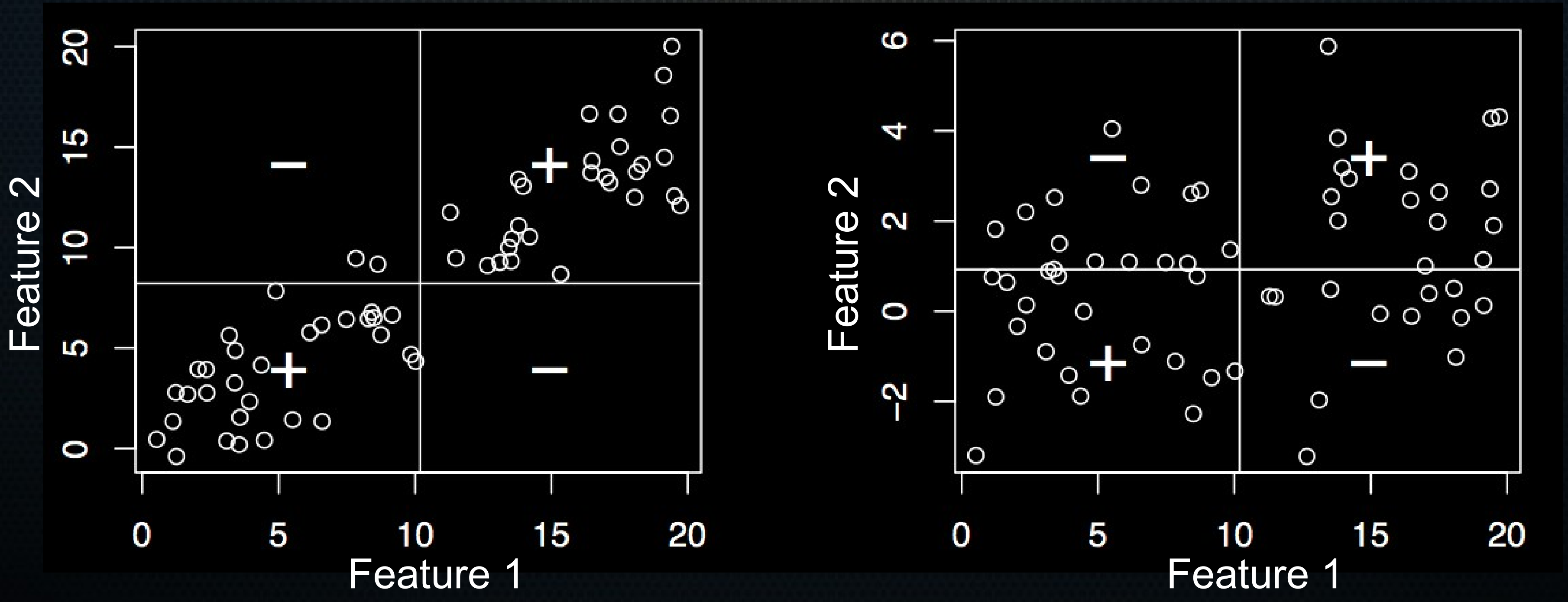


- Mean = $\mu = \frac{1}{5} \cdot (-10 + -3 + 4 + 8 + 9) = 1.6$
- Variance = $\sigma^2 = \frac{\sum (x_i - \mu)^2}{m-1} = 64.3$
- Standard deviation = $\sigma = \sqrt{\sigma^2} \approx 8.02$

Covariance matrix

- Multiple features can be correlated, and hence vary with each other (co-vary):

$$Cov(feat_1, feat_2) = \frac{1}{n}\sum_{n=1}^{n}(\mathrm{feat1} - \mu_{feat1})\cdot(\mathrm{feat2} - \mu_{feat2})$$



Covariance matrix

- Multiple features can be correlated, and hence vary with each other (co-vary):

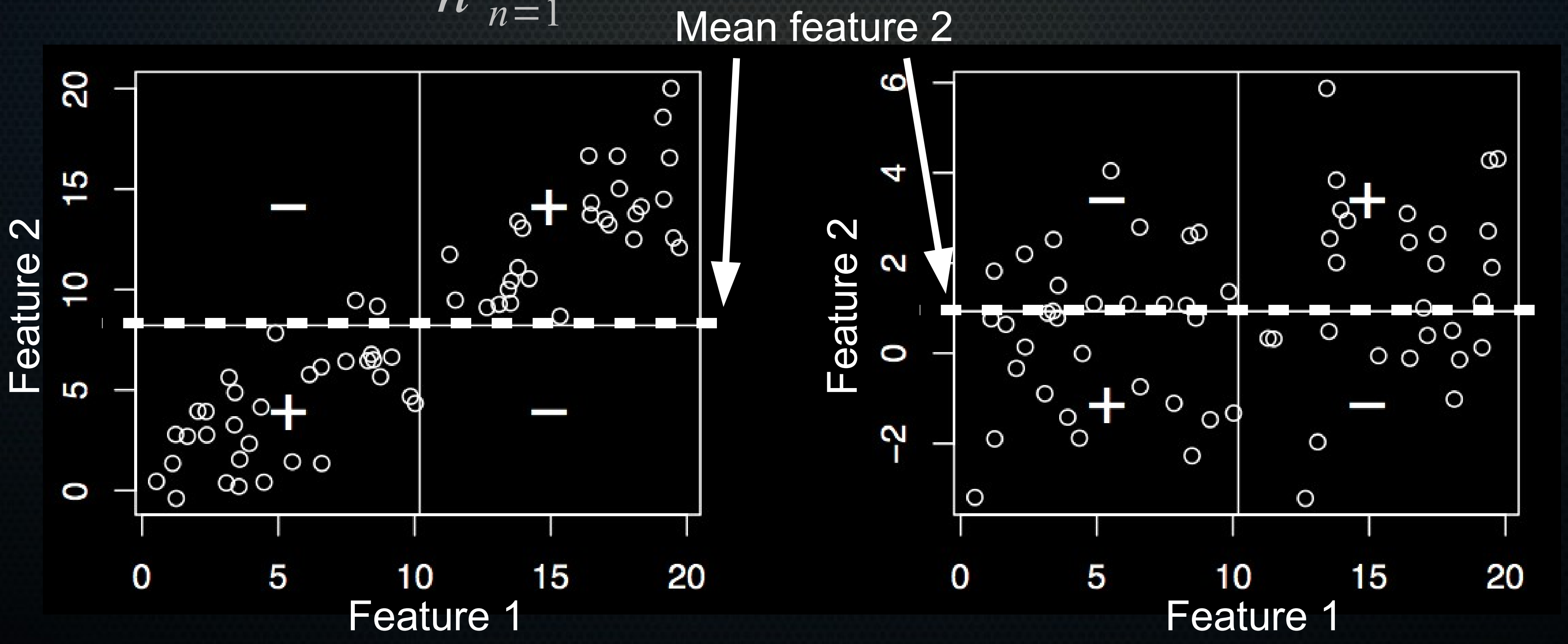
$$Cov(feat_1, feat_2) = \frac{1}{n}\sum_{n=1}^{n}(\mathrm{feat1} - \mu_{feat\,1})\cdot(\mathrm{feat2} - \mu_{feat\,2})$$

Mean feature 1

Covariance matrix

- Multiple features can be correlated, and hence vary with each other (co-vary):

$$Cov(feat_1, feat_2) = \frac{1}{n}\sum_{n=1}^{n}(\mathrm{feat1} - \mu_{feat\,1}) \cdot (\mathrm{feat2} - \mu_{feat\,2})$$



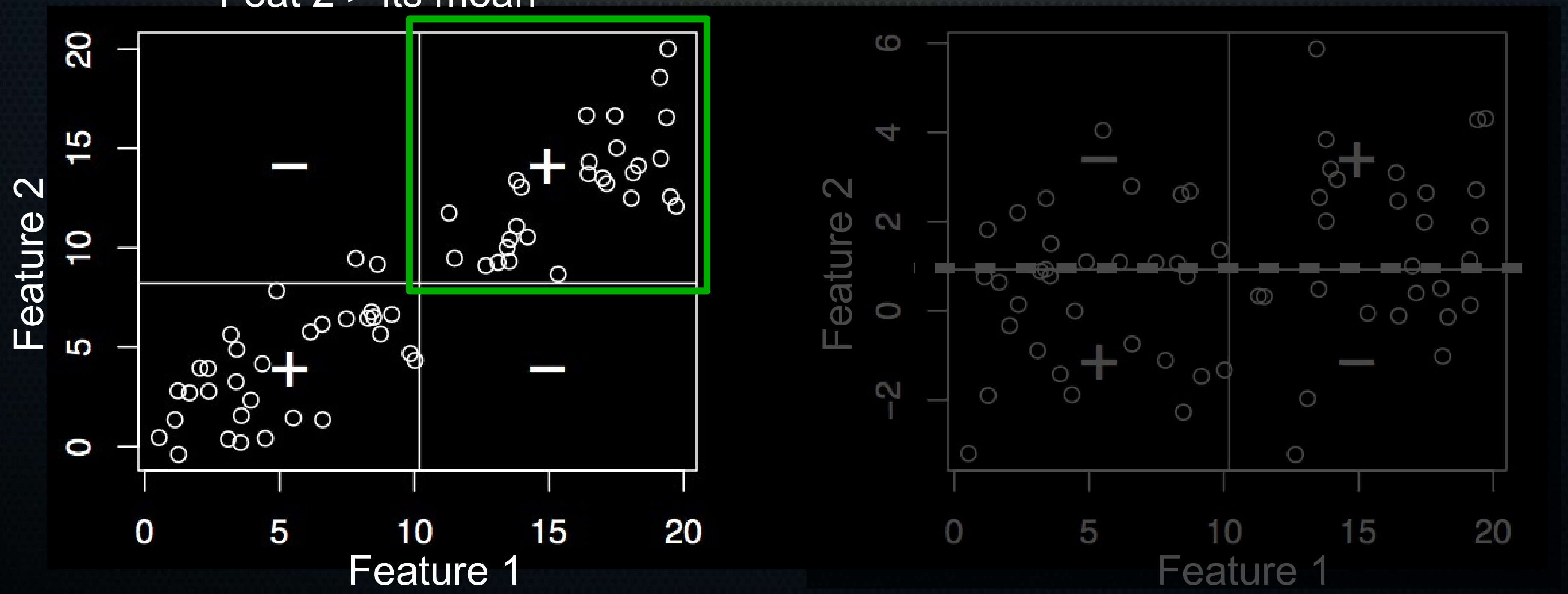
Covariance matrix

- Multiple features can be correlated, and hence vary with each other (co-vary):

$$Cov(feat_1, feat_2) = \frac{1}{n}\sum_{n=1}^{n}(\mathrm{feat1} - \mu_{feat1})\cdot(\mathrm{feat2} - \mu_{feat2})$$

$+ \quad * \quad + \quad = +$

When Feat 1 > its mean,
Feat 2 > its mean



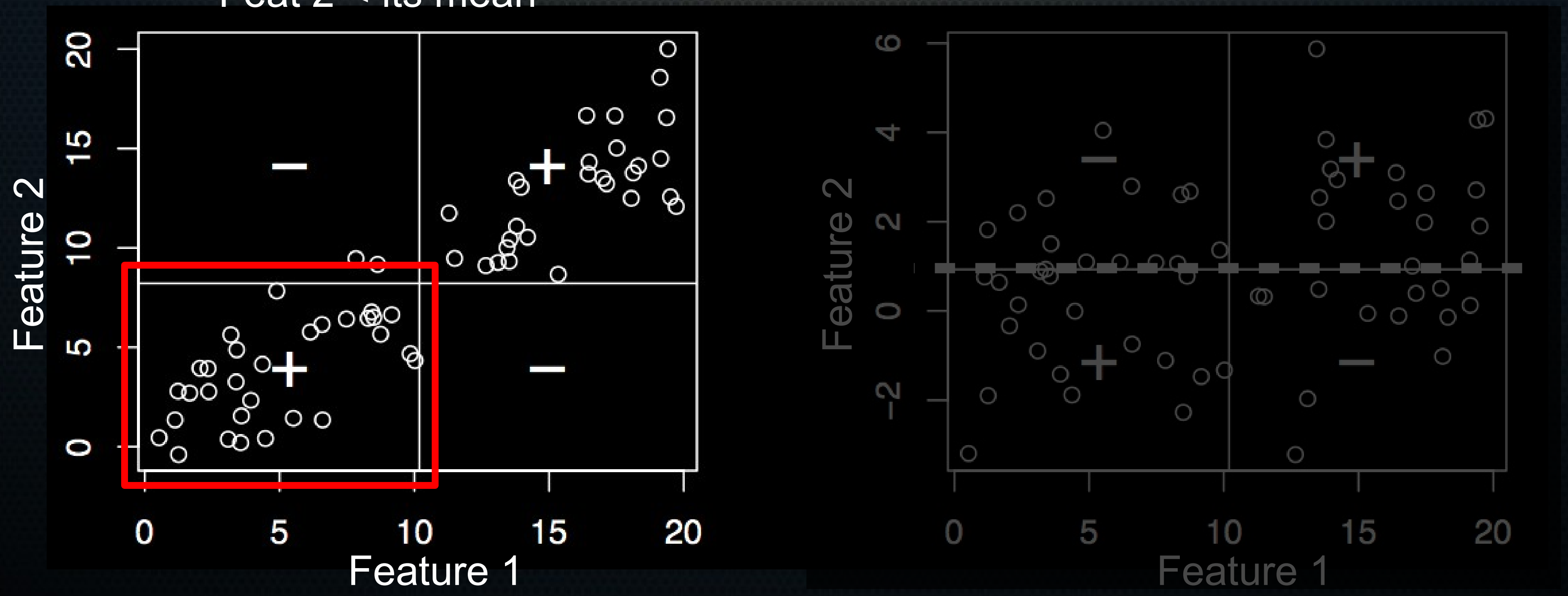
Covariance matrix

- Multiple features can be correlated, and hence vary with each other (co-vary):

$$Cov(feat_1, feat_2) = \frac{1}{n}\sum_{n=1}^{n}(\text{feat1} - \mu_{feat1})\cdot(\text{feat2} - \mu_{feat2})$$

- * - = +

When Feat 1 < its mean,
Feat 2 < its mean

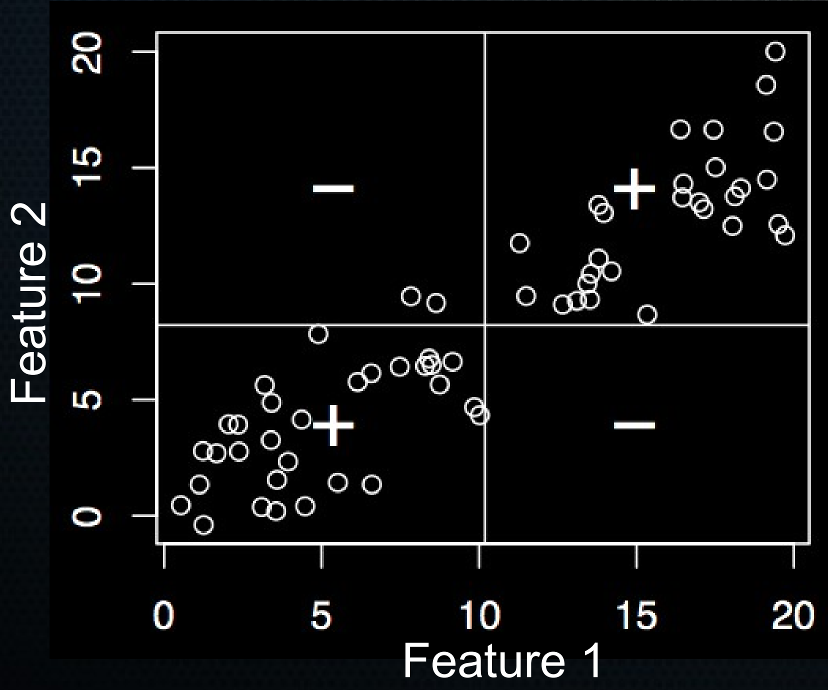


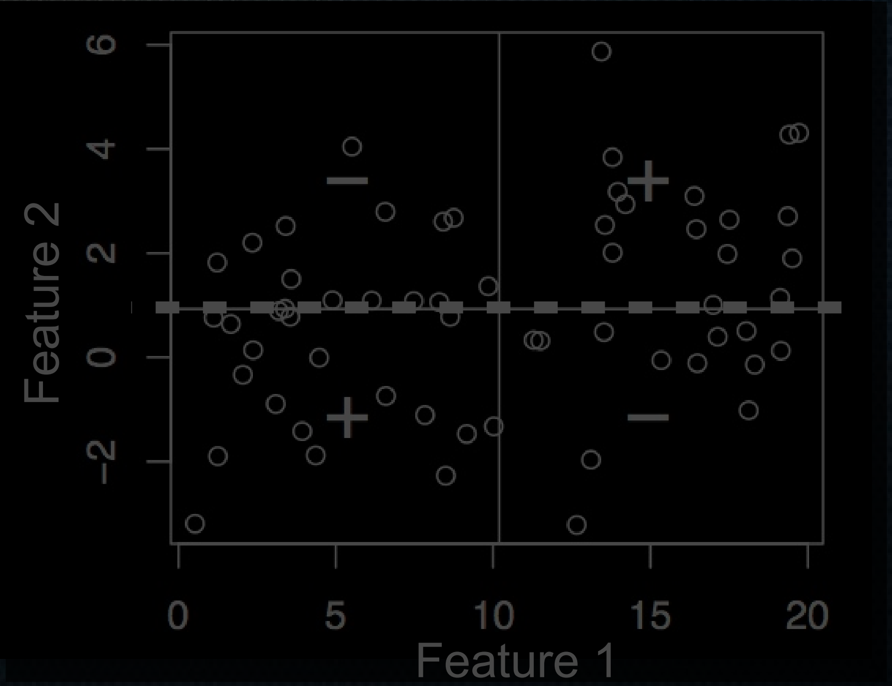
Covariance matrix

- Multiple features can be correlated, and hence vary with each other (co-vary):

$$Cov(feat_1, feat_2) = \frac{1}{n}\sum_{n=1}^{n}(\mathrm{feat1} - \mu_{feat1}) \cdot (\mathrm{feat2} - \mu_{feat2})$$

Here: covariance = 30.11

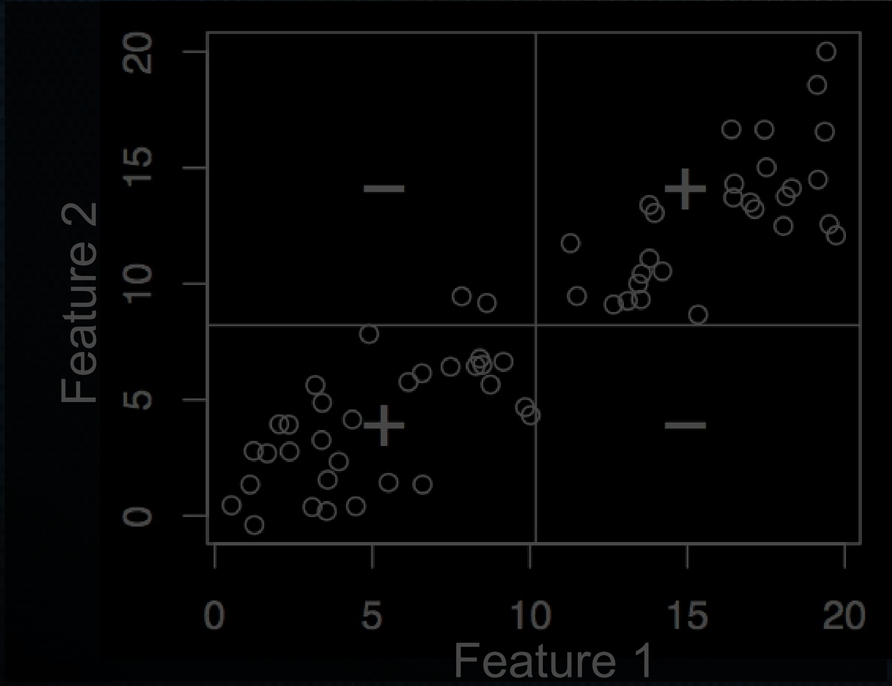




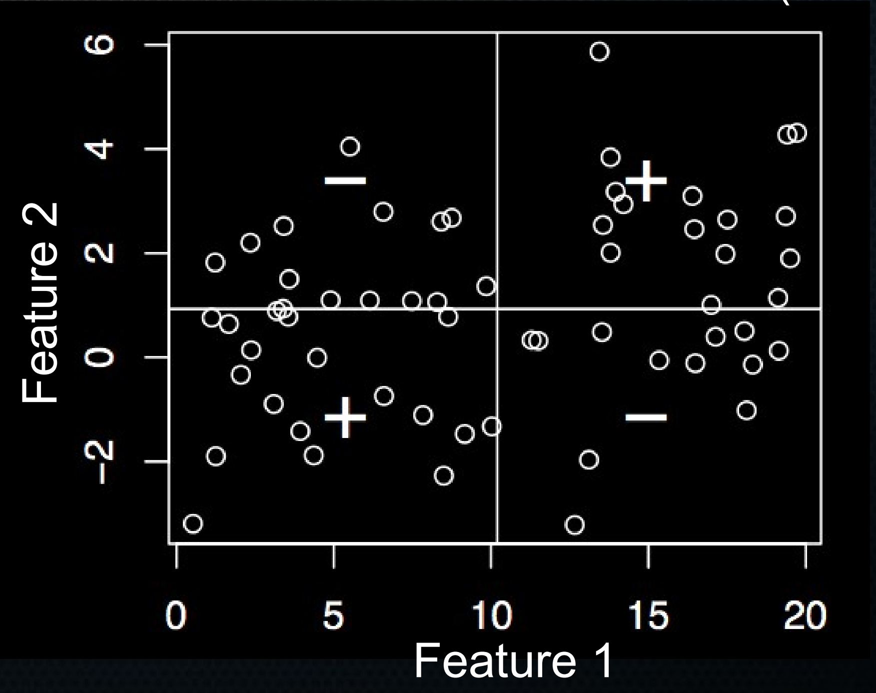
Covariance matrix

- Multiple features can be correlated, and hence vary with each other (co-vary):

$$Cov(feat_1, feat_2) = \frac{1}{n}\sum_{n=1}^{n} (\mathrm{feat1} - \mu_{feat1}) \cdot (\mathrm{feat2} - \mu_{feat2})$$



Here: almost no co-variance (3.02)

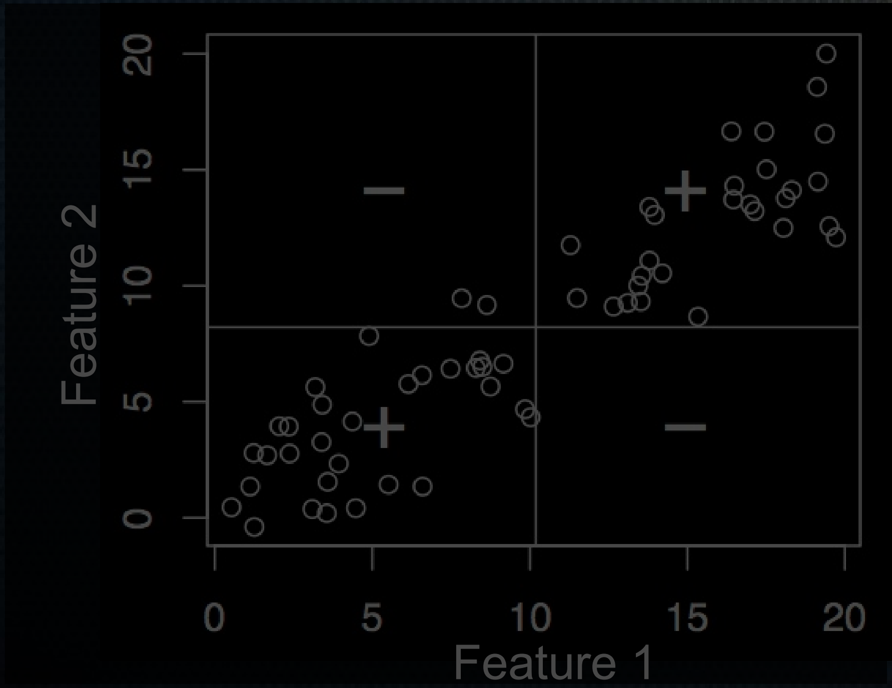


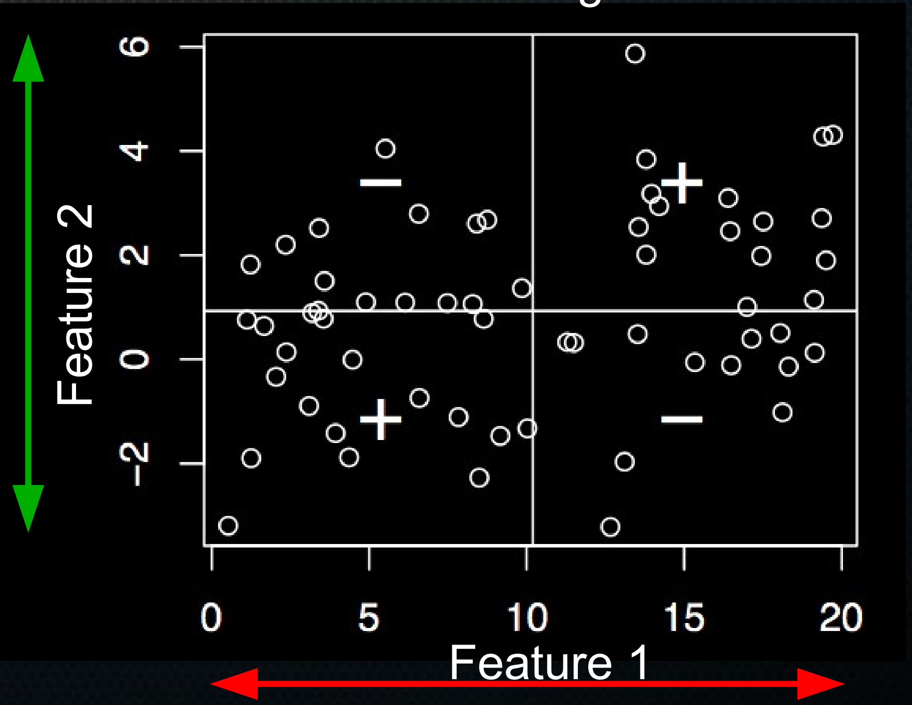
Covariance matrix

- Multiple features can be correlated, and hence vary with each other (co-vary):

$$Cov(feat_1, feat_2) = \frac{1}{n}\sum_{n=1}^{n}(\text{feat1} - \mu_{feat1}) \cdot (\text{feat2} - \mu_{feat2})$$

Note: different feature ranges influence co-variance!





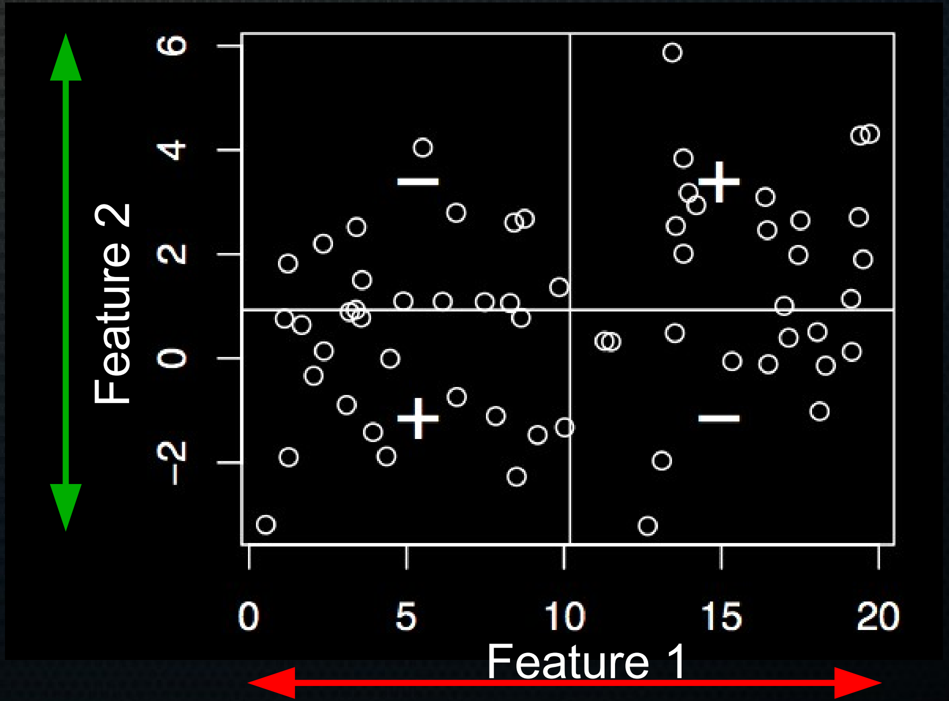
Covariance matrix

- Multiple features can be correlated, and hence vary with each other (co-vary):

$$Cov(feat_1, feat_2) = \frac{1}{n}\sum_{n=1}^{n}(\text{feat1} - \mu_{feat1})\cdot(\text{feat2} - \mu_{feat2})$$

$$Correlation(feat_1, feat_2) = \frac{Cov(feat_1, feat_2)}{\sqrt{Var(feat_1)\cdot Var(feat_2)}}$$

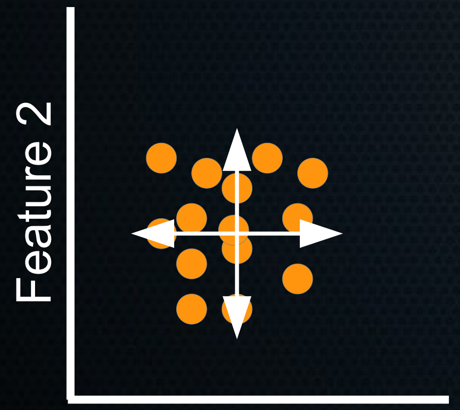
Standardise: well-known correlation coëfficient



Covariance matrix

- What about the matrix part?

$$Cov(feat_1, feat_2) = \frac{1}{n}\sum_{n=1}^{n} (\mathrm{feat1} - \mu_{feat\,1}) \cdot (\mathrm{feat2} - \mu_{feat\,2})$$

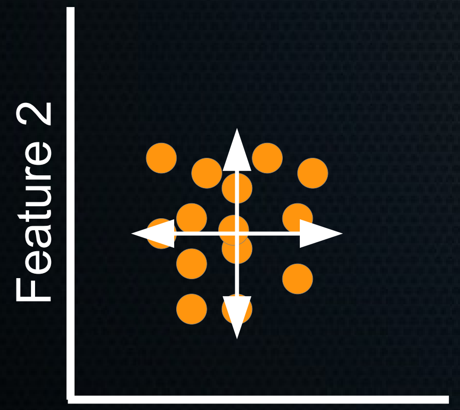


$$\begin{array}{ccc} & \text{Feat 1} & \text{Feat 2} \\ \begin{array}{c}\text{Feat 1} \\ \text{Feat 2}\end{array} & \multicolumn{2}{c}{\begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}} \end{array}$$

Covariance matrix

- What about the matrix part?

$$Cov(feat_1, feat_2) = \frac{1}{n}\sum_{n=1}^{n}(\text{feat1} - \mu_{feat\,1})\cdot(\text{feat2} - \mu_{feat\,2})$$



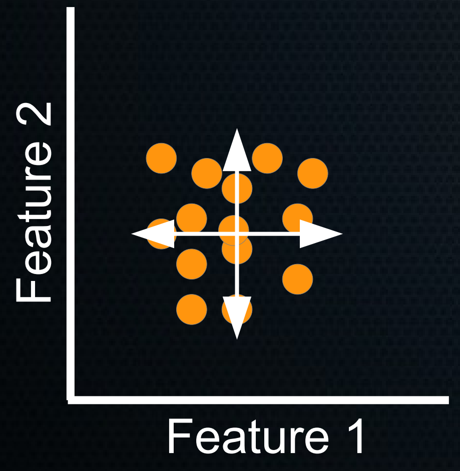
$$\begin{array}{ccc} & \text{Feat 1} & \text{Feat 2} \\ \begin{array}{c}\text{Feat 1}\\ \text{Feat 2}\end{array} & \multicolumn{2}{c}{\begin{bmatrix} \boxed{\sigma^2} & 0 \\ 0 & \boxed{\sigma^2} \end{bmatrix}} \end{array}$$

Both features have the same variance
(co-variance with yourself = variance)

Covariance matrix

- What about the matrix part?

$$Cov(feat_1, feat_2) = \frac{1}{n}\sum_{n=1}^{n}(\text{feat1} - \mu_{feat\,1})\cdot(\text{feat2} - \mu_{feat\,2})$$



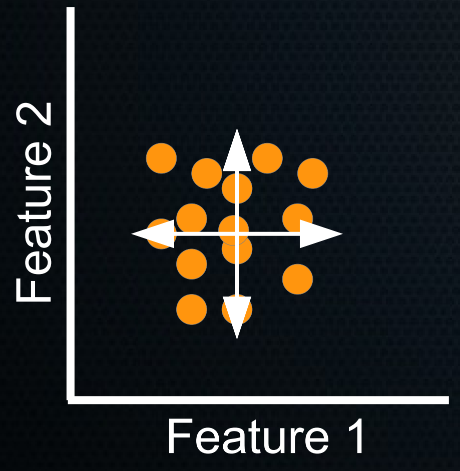
$$\begin{array}{ccc} & \text{Feat 1} & \text{Feat 2} \\ \text{Feat 1} & \multicolumn{2}{c}{\multirow{2}{*}{$\begin{bmatrix} \sigma^2 & \boxed{0} \\ \boxed{0} & \sigma^2 \end{bmatrix}$}} \\ \text{Feat 2} & & \end{array}$$

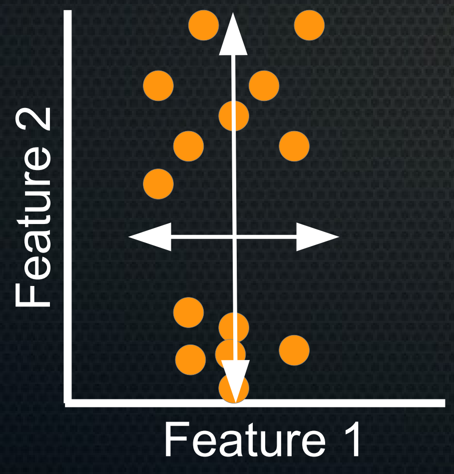
Neither feature tells you something about the other (no co-variance)

Covariance matrix

- What about the matrix part?

$$Cov(feat_1, feat_2) = \frac{1}{n}\sum_{n=1}^{n}(\mathrm{feat1} - \mu_{feat1}) \cdot (\mathrm{feat2} - \mu_{feat2})$$



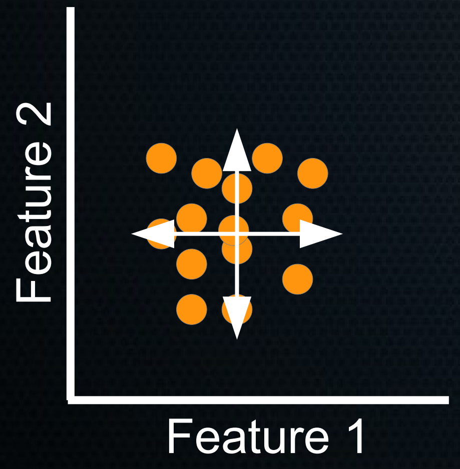


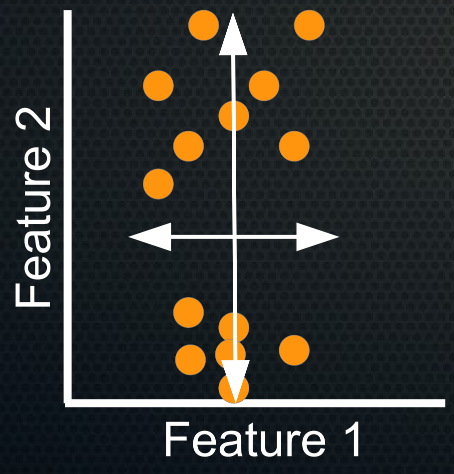
$$\begin{array}{c|cc} & \text{Feat 1} & \text{Feat 2} \\ \hline \text{Feat 1} & \sigma^2 & 0 \\ \text{Feat 2} & 0 & \sigma^2 \end{array} \qquad \begin{array}{c|cc} & \text{Feat 1} & \text{Feat 2} \\ \hline \text{Feat 1} & \sigma_1^2 & 0 \\ \text{Feat 2} & 0 & \sigma_2^2 \end{array}$$

Covariance matrix

- What about the matrix part?

$$Cov(feat_1, feat_2) = \frac{1}{n}\sum_{n=1}^{n}(\mathrm{feat1} - \mu_{feat\,1}) \cdot (\mathrm{feat2} - \mu_{feat\,2})$$





$$\begin{array}{cc} & \begin{array}{cc}\text{Feat 1} & \text{Feat 2}\end{array} \\ \begin{array}{c}\text{Feat 1} \\ \text{Feat 2}\end{array} & \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix} \end{array}$$

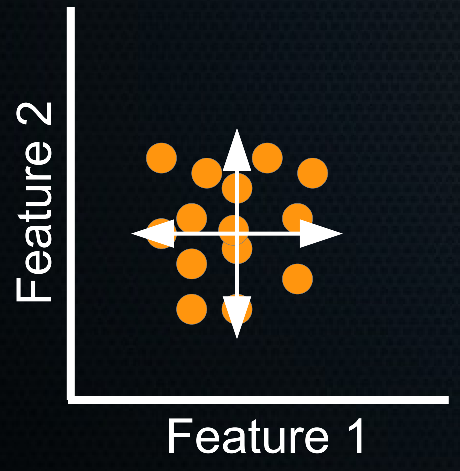
$$\begin{array}{cc} & \begin{array}{cc}\text{Feat 1} & \text{Feat 2}\end{array} \\ \begin{array}{c}\text{Feat 1} \\ \text{Feat 2}\end{array} & \begin{bmatrix} \boxed{\sigma_1^2} & 0 \\ 0 & \boxed{\sigma_2^2} \end{bmatrix} \end{array}$$

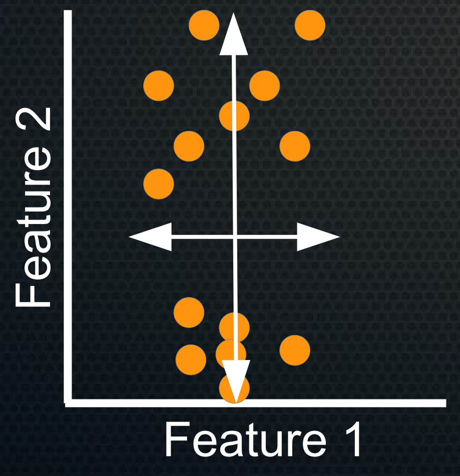
Different variances (Feature 2 more spread out)

Covariance matrix

- What about the matrix part?

$$Cov(feat_1, feat_2) = \frac{1}{n}\sum_{n=1}^{n}(\mathrm{feat1} - \mu_{feat\,1}) \cdot (\mathrm{feat2} - \mu_{feat\,2})$$





$$\begin{array}{cc} & \begin{array}{cc} \text{Feat 1} & \text{Feat 2} \end{array} \\ \begin{array}{c} \text{Feat 1} \\ \text{Feat 2} \end{array} & \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix} \end{array}$$

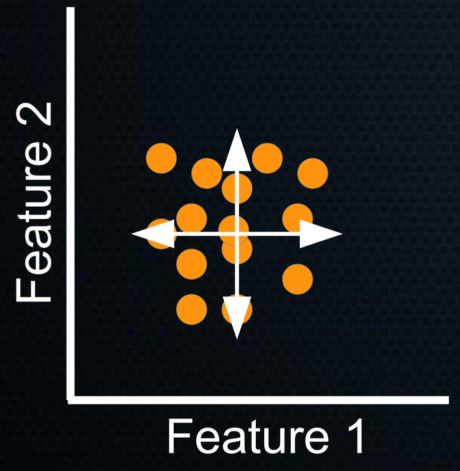
$$\begin{array}{cc} & \begin{array}{cc} \text{Feat 1} & \text{Feat 2} \end{array} \\ \begin{array}{c} \text{Feat 1} \\ \text{Feat 2} \end{array} & \begin{bmatrix} \sigma_1^2 & \boxed{0} \\ \boxed{0} & \sigma_2^2 \end{bmatrix} \end{array}$$

Still no co-variance (Feature 1 tells nothing about 2 and vice-versa)

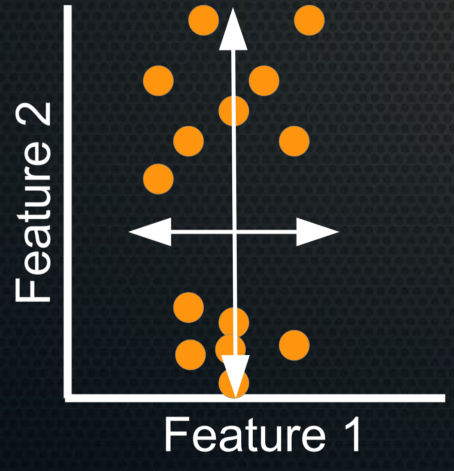
Covariance matrix

- What about the matrix part?

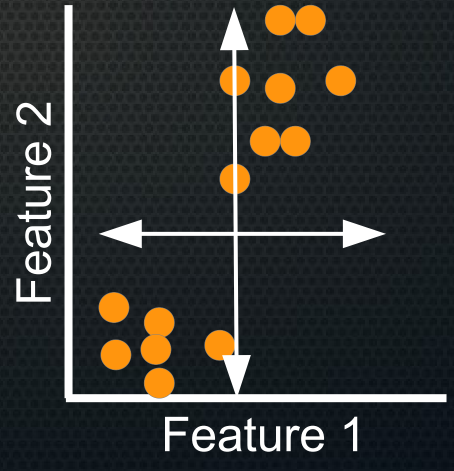
$$Cov(feat_1, feat_2) = \frac{1}{n}\sum_{n=1}^{n}(\mathrm{feat1} - \mu_{feat\,1})\cdot(\mathrm{feat2} - \mu_{feat\,2})$$



$$\begin{array}{c c} & \begin{array}{cc} \text{Feat 1} & \text{Feat 2} \end{array} \\ \begin{array}{c} \text{Feat 1} \\ \text{Feat 2} \end{array} & \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix} \end{array}$$



$$\begin{array}{c c} & \begin{array}{cc} \text{Feat 1} & \text{Feat 2} \end{array} \\ \begin{array}{c} \text{Feat 1} \\ \text{Feat 2} \end{array} & \begin{bmatrix} \sigma_1^2 & \boxed{0} \\ \boxed{0} & \sigma_2^2 \end{bmatrix} \end{array}$$



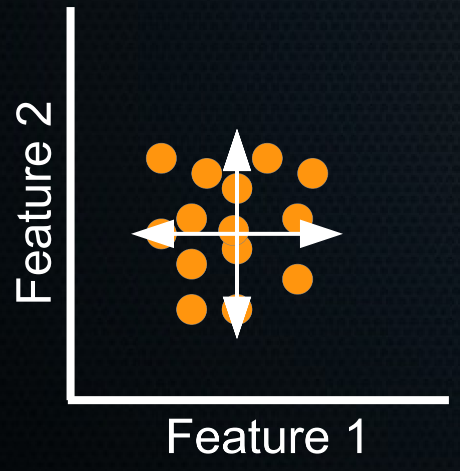
Full covariance matrix (positive correlation)

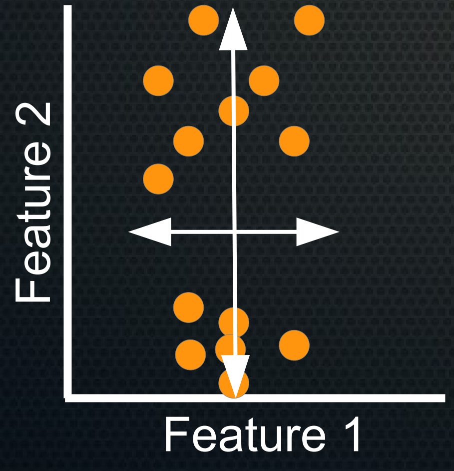
$$\begin{array}{c c} & \begin{array}{cc} \text{Feat 1} & \text{Feat 2} \end{array} \\ \begin{array}{c} \text{Feat 1} \\ \text{Feat 2} \end{array} & \begin{bmatrix} \sigma_1^2 & Cov(1,2) \\ Cov(2,1) & \sigma_2^2 \end{bmatrix} \end{array}$$

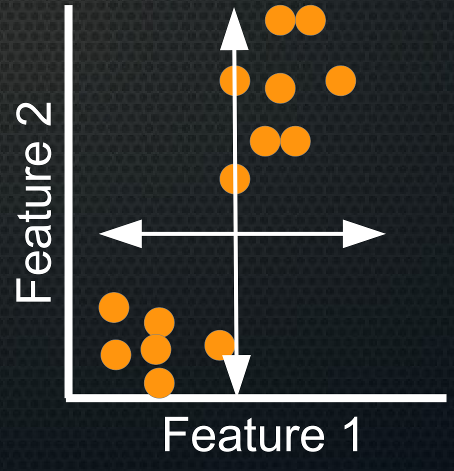
Covariance matrix

- What about the matrix part?

$$Cov(feat_1, feat_2) = \frac{1}{n}\sum_{n=1}^{n}(\text{feat1} - \mu_{feat\,1})\cdot(\text{feat2} - \mu_{feat\,2})$$







$$\begin{array}{c c} & \begin{array}{cc}\text{Feat 1} & \text{Feat 2}\end{array} \\ \begin{array}{c}\text{Feat 1} \\ \text{Feat 2}\end{array} & \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix} \end{array}$$

$$\begin{array}{c c} & \begin{array}{cc}\text{Feat 1} & \text{Feat 2}\end{array} \\ \begin{array}{c}\text{Feat 1} \\ \text{Feat 2}\end{array} & \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix} \end{array}$$

Full covariance matrix (positive correlation)

$$\begin{array}{c c} & \begin{array}{cc}\text{Feat 1} & \text{Feat 2}\end{array} \\ \begin{array}{c}\text{Feat 1} \\ \text{Feat 2}\end{array} & \begin{bmatrix} \sigma_1^2 & \boxed{Cov(1,2)} \\ \boxed{Cov(2,1)} & \sigma_2^2 \end{bmatrix} \end{array}$$

These are the same thing (symmetric matrix)

Covariance matrix

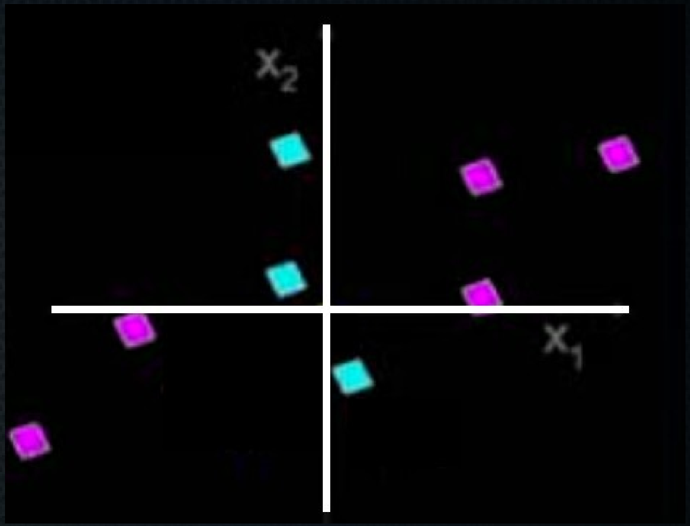
- Symmetric matrix that contains information on linear relationships between features (including on the spread of each individual feature (diagonal elements))

Covariance matrix

- Symmetric matrix that contains information on linear relationships between features (including on the spread of each individual feature (diagonal elements))
- So what can we do with it?

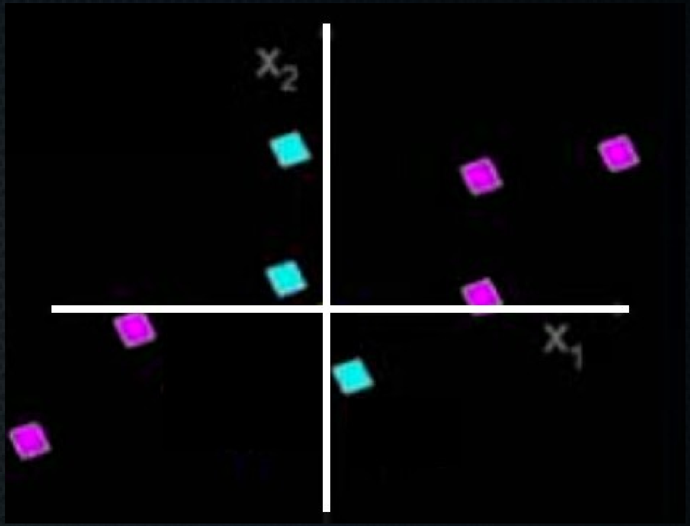
Start of PCA: apply the covariance matrix

- Take your n-dimensional (here 2D) data set and center each feature (set its mean to 0)



Start of PCA: apply the covariance matrix

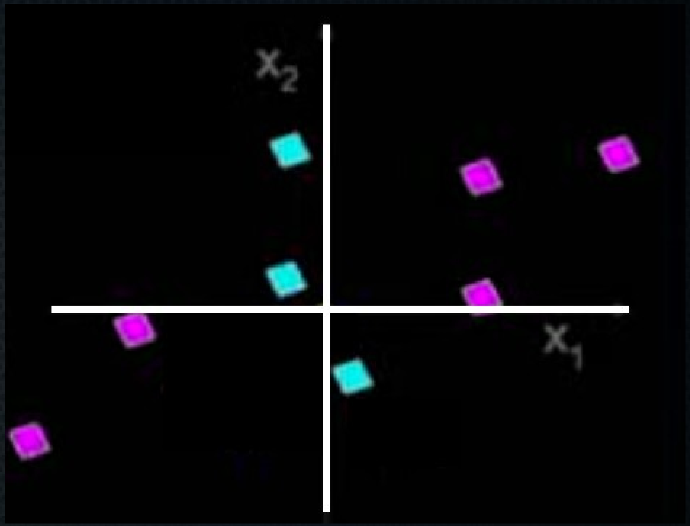
- Take your n-dimensional (here 2D) data set and center each feature (set its mean to 0)
- Covariance matrix of this sample data:



$$\begin{array}{cc} & \begin{array}{cc} \text{x1} & \text{x2} \end{array} \\ \begin{array}{c} \text{x1} \\ \text{x2} \end{array} & \begin{bmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{bmatrix} \end{array} \qquad Cov(x_1, x_2) = \frac{1}{n} \sum_{n=1}^{n} \text{feat1} \cdot \text{feat2}$$

Start of PCA: apply the covariance matrix

- Take your n-dimensional (here 2D) data set and center each feature (set its mean to 0)
- Covariance matrix of this sample data:



$$\begin{array}{c|cc} & x1 & x2 \\ \hline x1 & \multicolumn{2}{c}{\multirow{2}{*}{$\begin{bmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{bmatrix}$}} \\ x2 & & \end{array}$$

$$Cov(x_1, x_2) = \frac{1}{n} \sum_{n=1}^{n} \boxed{\mathrm{feat1} \cdot \mathrm{feat2}}$$

Mean already zero: can leave it out of the formula

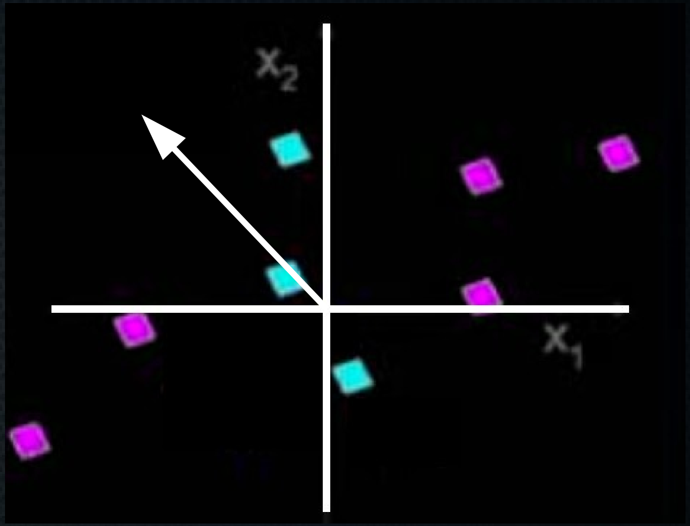
$$Cov(feat_1, feat_2) = \frac{1}{n} \sum_{n=1}^{n} (\mathrm{feat1} - \mu_{feat1}) \cdot (\mathrm{feat2} - \mu_{feat2})$$

Start of PCA: apply the covariance matrix

- Take your n-dimensional (here 2D) data set and center each feature (set its mean to 0)
- Covariance matrix of this sample data:

$$\begin{array}{ccc} & x1 & x2 \\ \begin{array}{c} x1 \\ x2 \end{array} & \multicolumn{2}{c}{\begin{bmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{bmatrix}} \end{array}$$

$$Cov(x_1, x_2) = \frac{1}{n} \sum_{n=1}^{n} \text{feat1} \cdot \text{feat2}$$

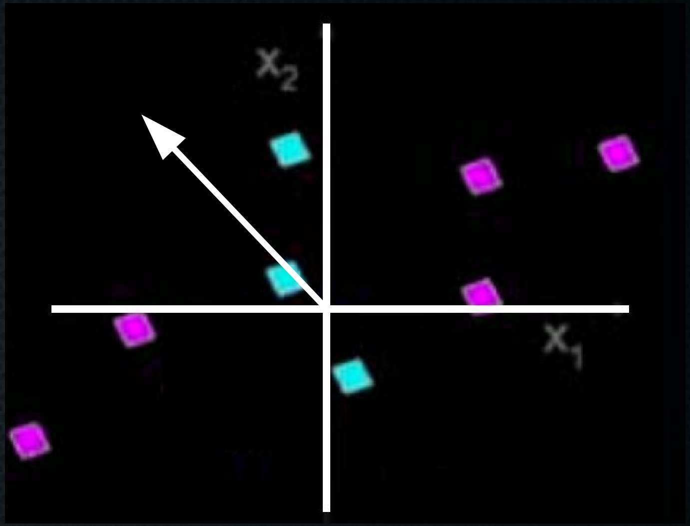


- Pick any vector (data point) in the image and multiply it with the covariance matrix. Here pick $\begin{bmatrix} -1 \\ 1 \end{bmatrix}$

Start of PCA: apply the covariance matrix

- Take your n-dimensional (here 2D) data set and center each feature (set its mean to 0)

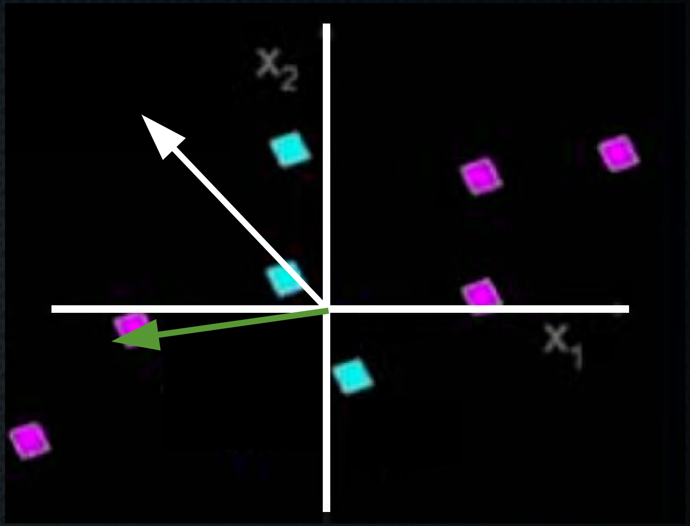
$$\begin{array}{cc} & \begin{array}{cc} x1 & x2 \end{array} \\ \begin{array}{c} x1 \\ x2 \end{array} & \begin{bmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{bmatrix} \end{array} . \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$



Start of PCA: apply the covariance matrix

- Take your n-dimensional (here 2D) data set and center each feature (set its mean to 0)

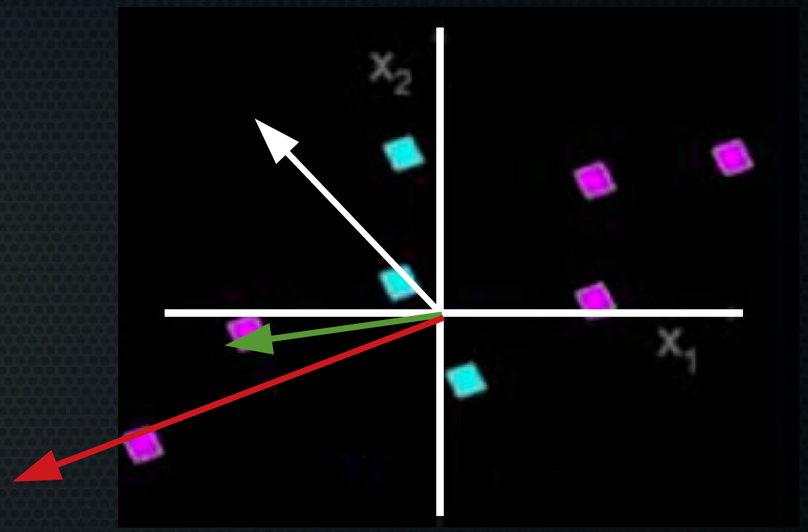
$$\begin{array}{cc} & \begin{array}{cc} x1 & x2 \end{array} \\ \begin{array}{c} x1 \\ x2 \end{array} & \begin{bmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{bmatrix} \end{array} \cdot \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2.0 \cdot 1 + 0.8 \cdot -1 \\ 0.8 \cdot -1 + 0.6 \cdot 1 \end{bmatrix} = \begin{bmatrix} -1.2 \\ -0.2 \end{bmatrix}$$



Start of PCA: apply the covariance matrix

- Take your n-dimensional (here 2D) data set and center each feature (set its mean to 0)

$$\begin{array}{c c} & \begin{array}{cc} x1 & x2 \end{array} \\ \begin{array}{c} x1 \\ x2 \end{array} & \begin{bmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{bmatrix} \end{array} \cdot \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2.0 \cdot 1 + 0.8 \cdot -1 \\ 0.8 \cdot -1 + 0.6 \cdot 1 \end{bmatrix} = \begin{bmatrix} -1.2 \\ -0.2 \end{bmatrix}$$



- Do that again

$$\begin{array}{c c} & \begin{array}{cc} x1 & x2 \end{array} \\ \begin{array}{c} x1 \\ x2 \end{array} & \begin{bmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{bmatrix} \end{array} \cdot \begin{bmatrix} -1.2 \\ -0.2 \end{bmatrix} = \begin{bmatrix} -2.5 \\ -1.0 \end{bmatrix}$$

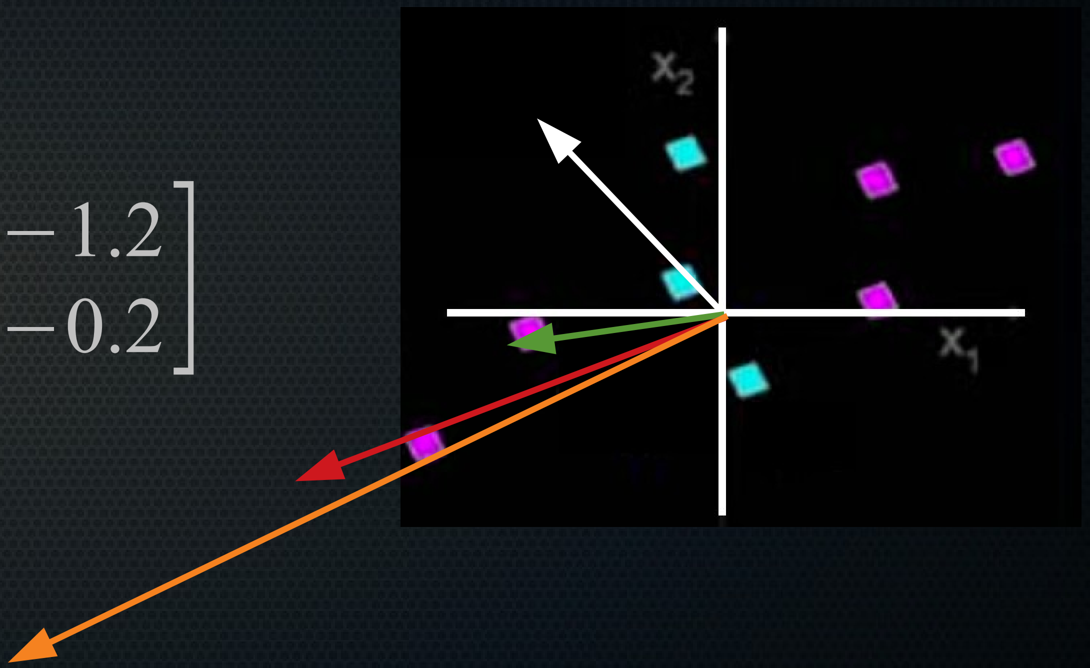
Start of PCA: apply the covariance matrix

- Take your n-dimensional (here 2D) data set and center each feature (set its mean to 0)

$$\begin{array}{cc} & \begin{array}{cc} x1 & x2 \end{array} \\ \begin{array}{c} x1 \\ x2 \end{array} & \begin{bmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{bmatrix} \end{array} \cdot \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2.0 \cdot 1 + 0.8 \cdot -1 \\ 0.8 \cdot -1 + 0.6 \cdot 1 \end{bmatrix} = \begin{bmatrix} -1.2 \\ -0.2 \end{bmatrix}$$

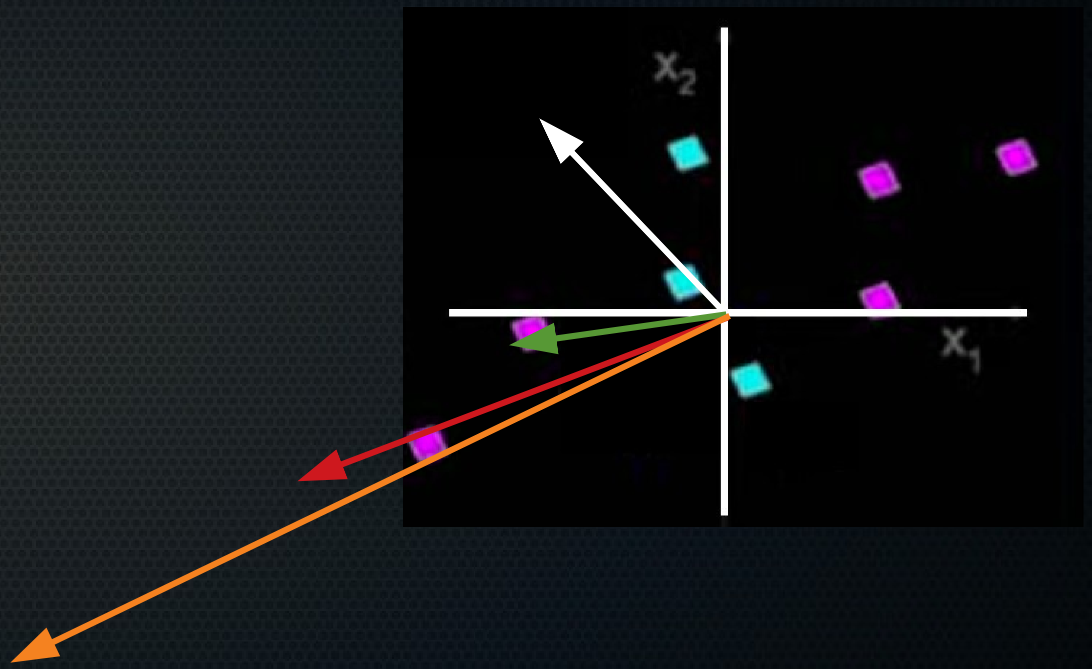
- Do that again, and again:

$$\begin{array}{cc} & \begin{array}{cc} x1 & x2 \end{array} \\ \begin{array}{c} x1 \\ x2 \end{array} & \begin{bmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{bmatrix} \end{array} \cdot \begin{bmatrix} -2.5 \\ -1.0 \end{bmatrix} = \begin{bmatrix} -6.0 \\ -2.7 \end{bmatrix}$$



Start of PCA: apply the covariance matrix

- Take your n-dimensional (here 2D) data set and center each feature (set its mean to 0)
- You tell me: what‘s happening?

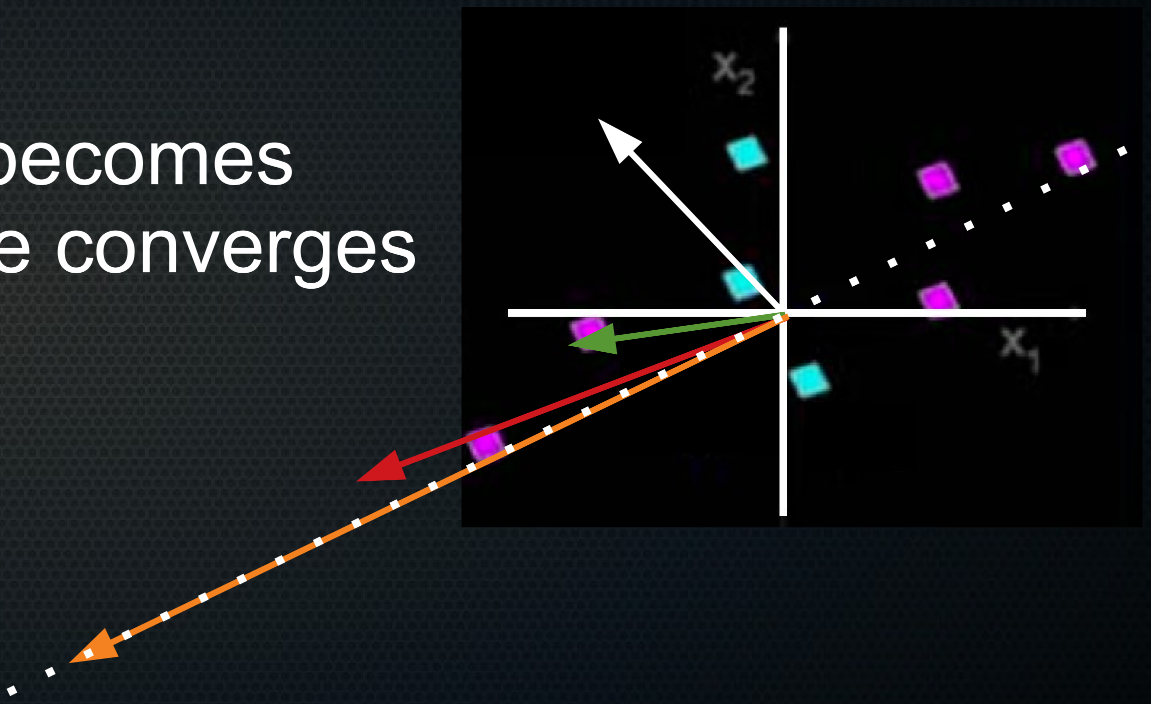


Start of PCA: apply the covariance matrix

- Take your n-dimensional (here 2D) data set and center each feature (set its mean to 0)
- You tell me: what‘s happening?
- This random vector is turning to the axis that holds the largest variance!

Start of PCA: apply the covariance matrix

- Take your n-dimensional (here 2D) data set and center each feature (set its mean to 0)
- If you do this even more: vector becomes too large to draw, but slope of line converges

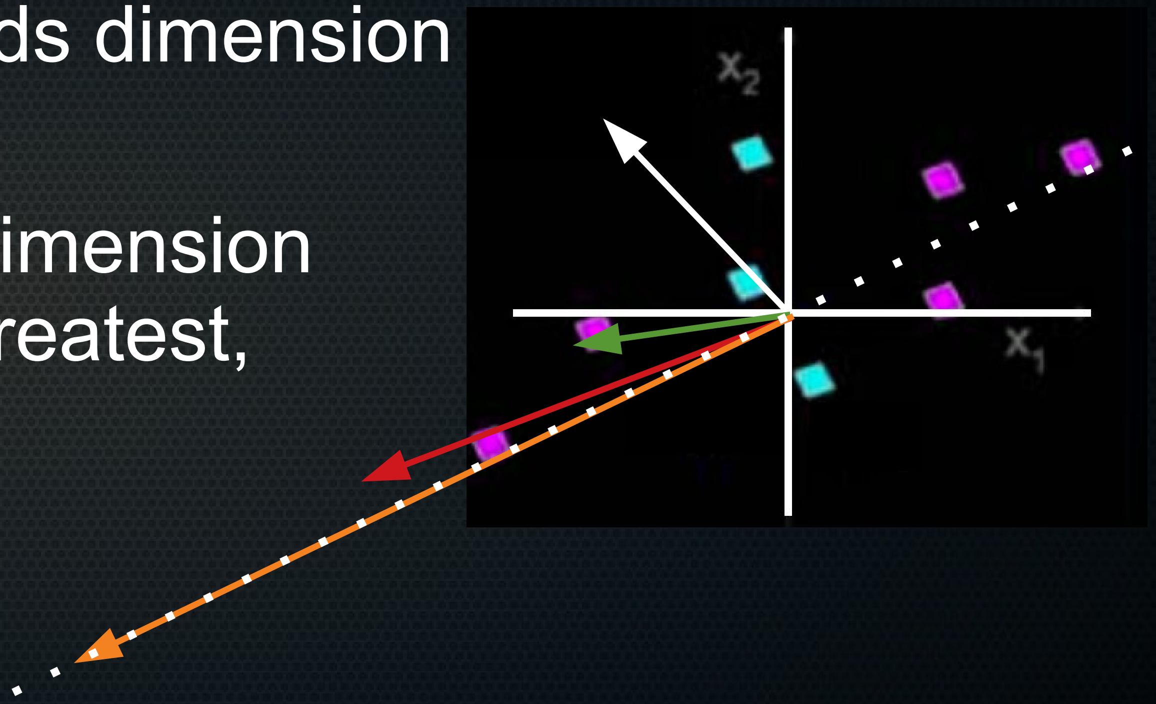


$$\begin{bmatrix} -6.0 \\ -2.7 \end{bmatrix} \rightarrow \begin{bmatrix} -14.1 \\ -6.4 \end{bmatrix} \rightarrow \begin{bmatrix} -33.3 \\ -15.1 \end{bmatrix}$$

$$\frac{-2.7}{-6} = 0.45 \rightarrow \frac{-6.4}{-14.1} = 0.454 \rightarrow \frac{-15.1}{-33.3} = 0.454$$

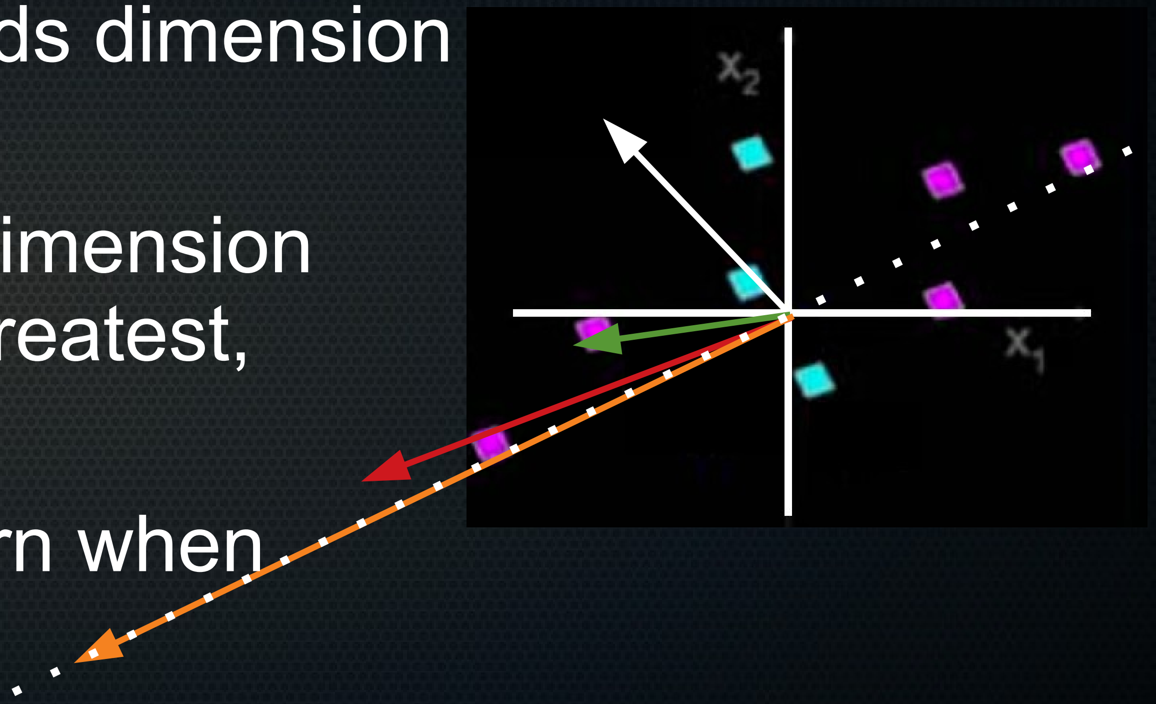
Start of PCA: apply the covariance matrix

- So: taking a random vector, multiplying with covariance matrix → turns towards dimension of greatest variance
- Exactly what we want for PCA: dimension of greatest variance (then next greatest, next greatest, etc.).



Start of PCA: apply the covariance matrix

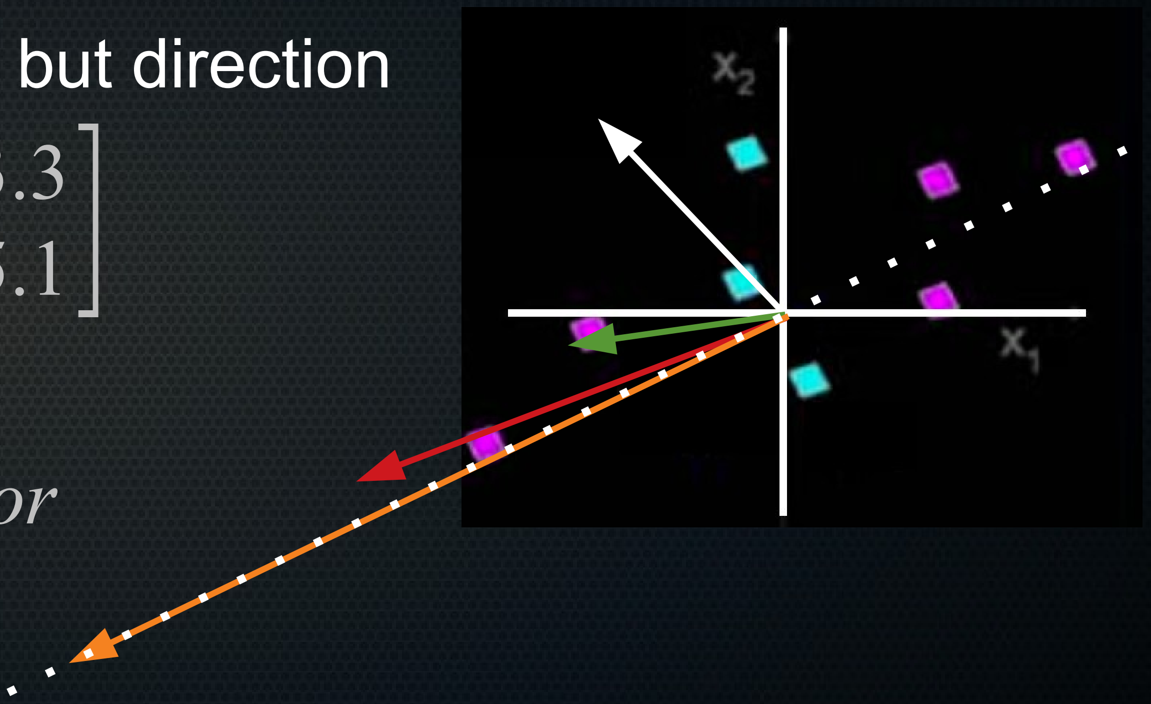
- So: taking a random vector, multiplying with covariance matrix → turns towards dimension of greatest variance
- Exactly what we want for PCA: dimension of greatest variance (then next greatest, next greatest, etc.).
- Can we find vectors that don‘t turn when multiplied with the covariance matrix?



Start of PCA: apply the covariance matrix

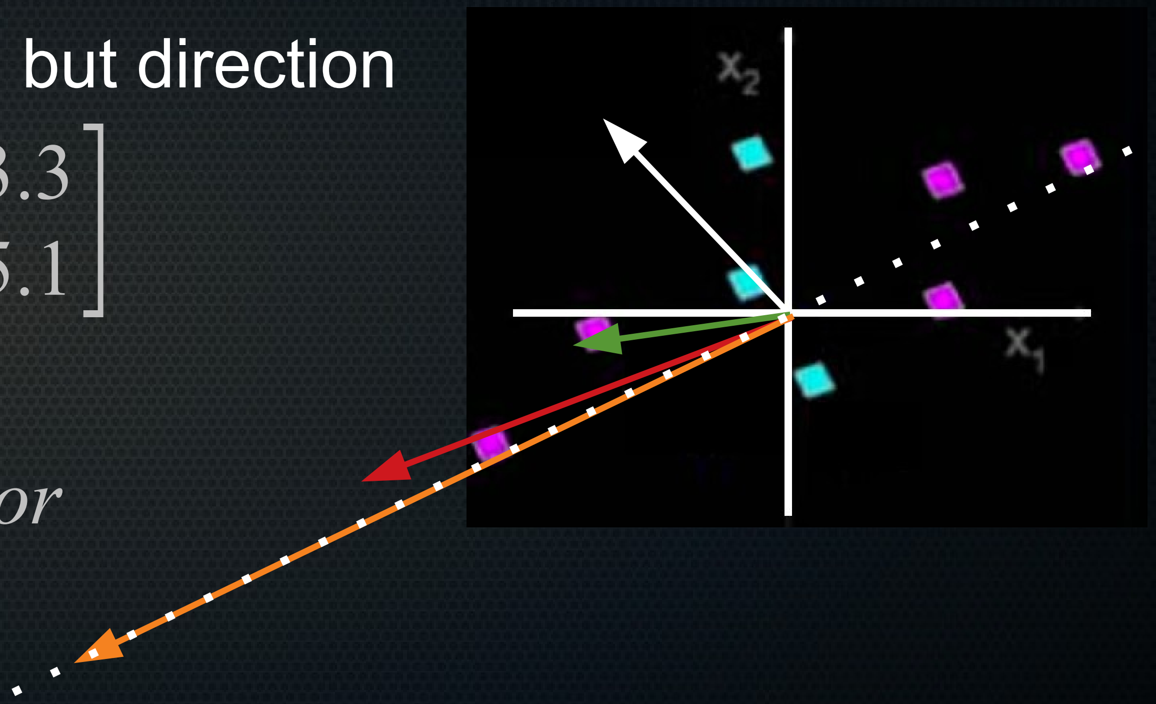
- Vectors that don‘t change direction with covariance matrix:
  - Size of the vector keeps changing, but direction doesn‘t $\begin{bmatrix} -6.0 \\ -2.7 \end{bmatrix} \rightarrow \begin{bmatrix} -14.1 \\ -6.4 \end{bmatrix} \rightarrow \begin{bmatrix} -33.3 \\ -15.1 \end{bmatrix}$
  - Looking for vectors that satisfy:

$$CovMatrix * vector = scalar * vector$$



Start of PCA: apply the covariance matrix

- Vectors that don't change direction with covariance matrix:
  - Size of the vector keeps changing, but direction doesn't $\begin{bmatrix}-6.0\\-2.7\end{bmatrix} \rightarrow \begin{bmatrix}-14.1\\-6.4\end{bmatrix} \rightarrow \begin{bmatrix}-33.3\\-15.1\end{bmatrix}$
  - Looking for vectors that satisfy:



$$CovMatrix * vector = scalar * vector$$

$$\sum * e = \lambda * e$$

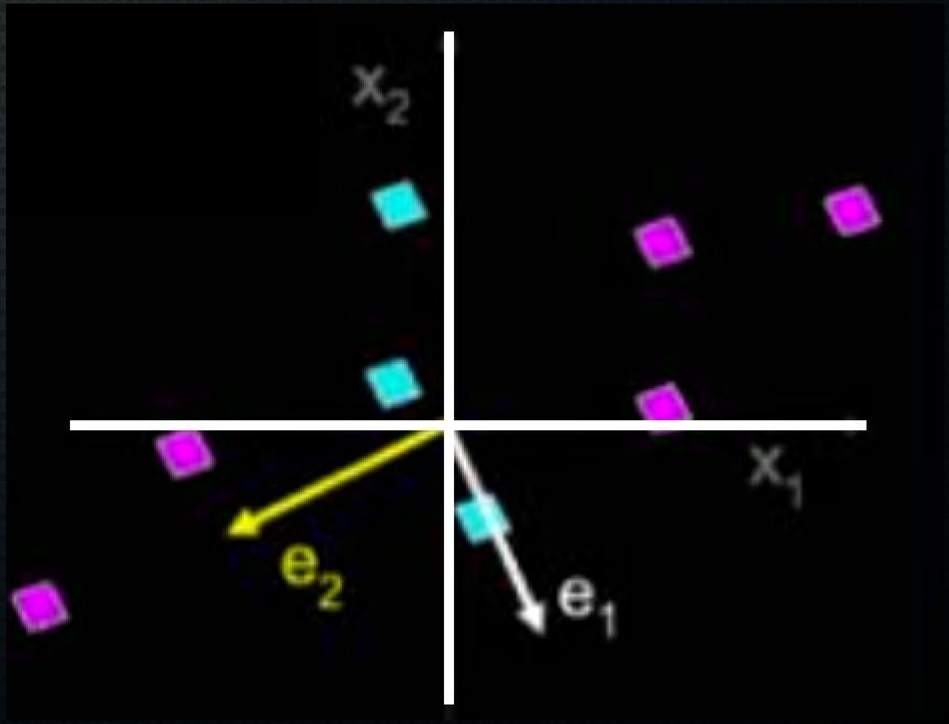
$$\begin{array}{cc} & \begin{array}{cc} x1 & x2 \end{array} \\ \begin{array}{c} x1 \\ x2 \end{array} & \begin{bmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{bmatrix} \end{array} \cdot \begin{bmatrix} a \\ b \end{bmatrix} = \lambda \cdot \begin{bmatrix} a \\ b \end{bmatrix} \longrightarrow$$

No rotation, only elongation (or shortening)

Start of PCA: eigenvectors and eigenvalues

- This concept has a name:
  - A vector that doesn‘t rotate when multiplied by a matrix is an *eigenvector* of that matrix. The factor by which it is elongated or shrunk is called the corresponding *eigenvalue*.

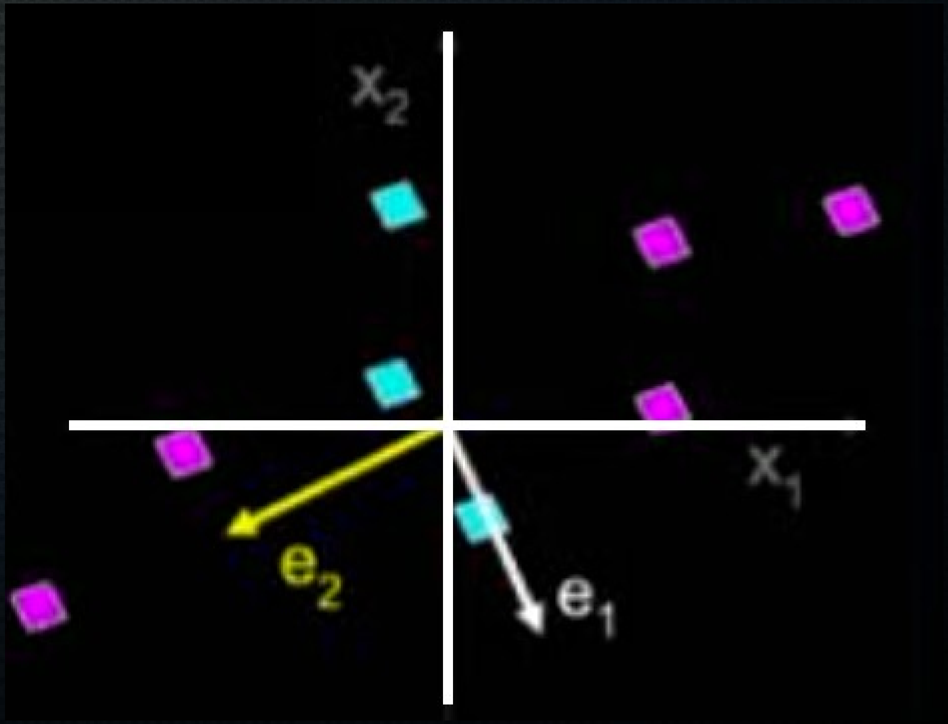
$A * e = \lambda * e$



Start of PCA: eigenvectors and eigenvalues

- This concept has a name:
  - A vector that doesn‘t rotate when multiplied by a matrix is an *eigenvector* of that matrix. The factor by which it is elongated or shrunk is called the corresponding *eigenvalue*.

    $A * e = \lambda * e$
  - Principal components of PCA:
    n eigenvectors with highest eigenvalues
    =
    n orthogonal axes on which to
    project the data and retain most structure
    in the data!

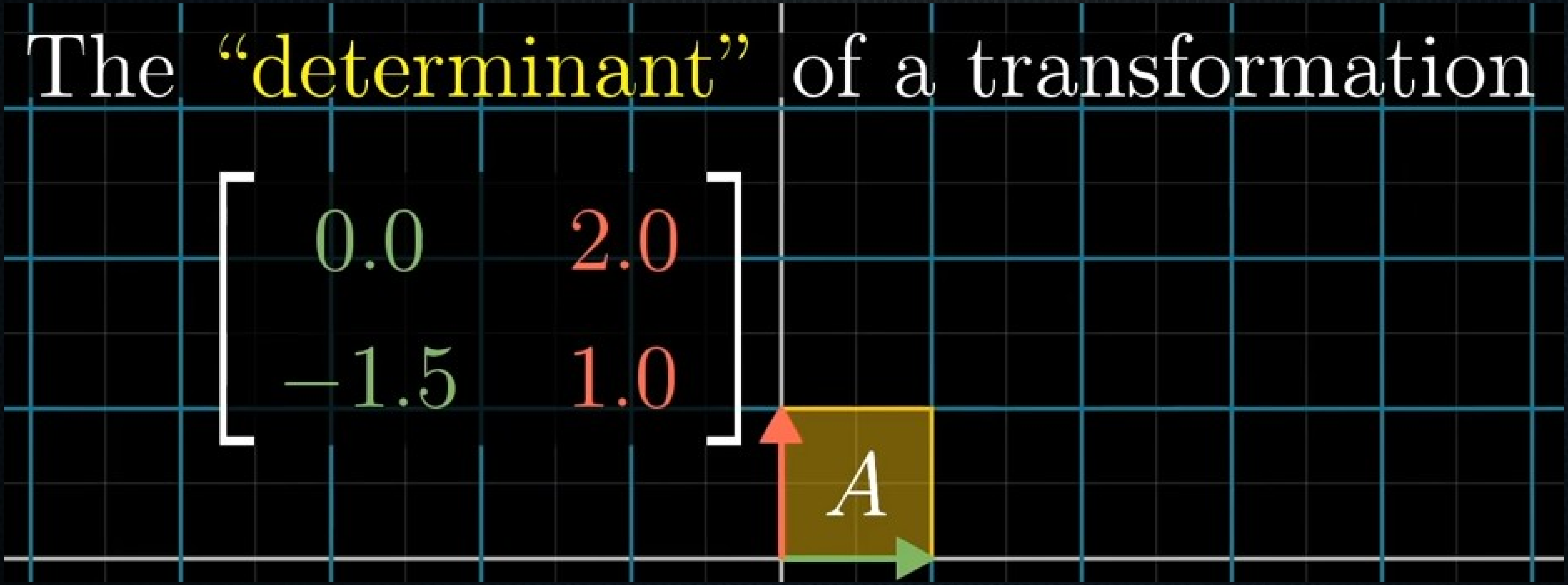


Start of PCA: determinant

- How do we get eigenvectors? Need to talk about determinant of a matrix.

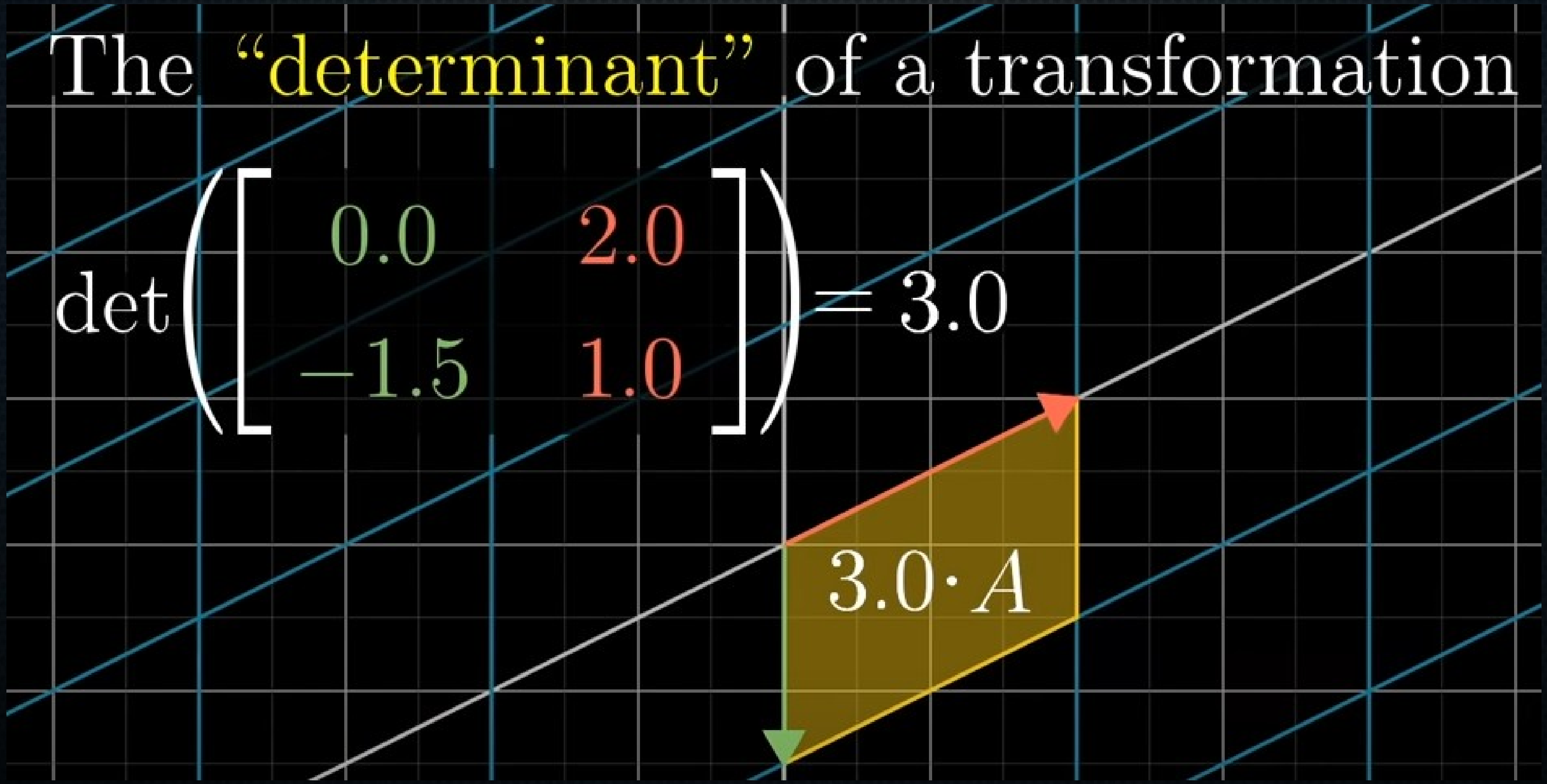
Start of PCA: determinant

- How do we get eigenvectors? Need to talk about determinant of a matrix.



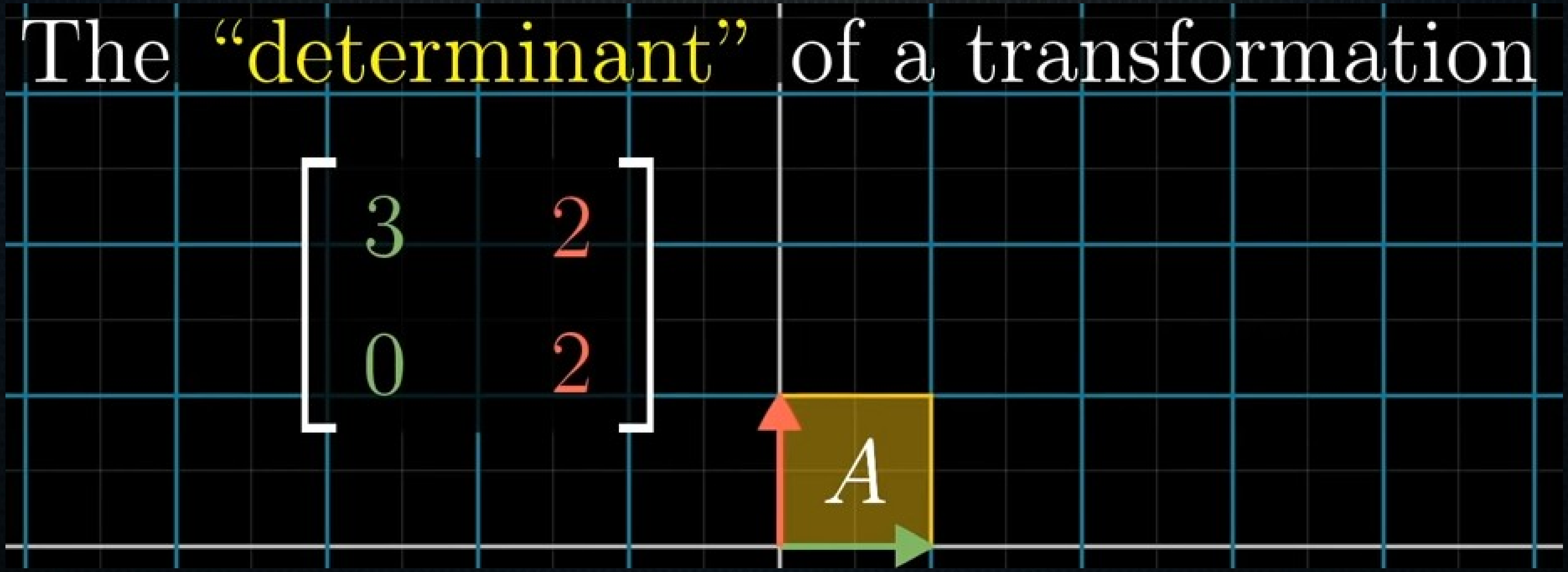
Start of PCA: determinant

- How do we get eigenvectors? Need to talk about determinant of a matrix.



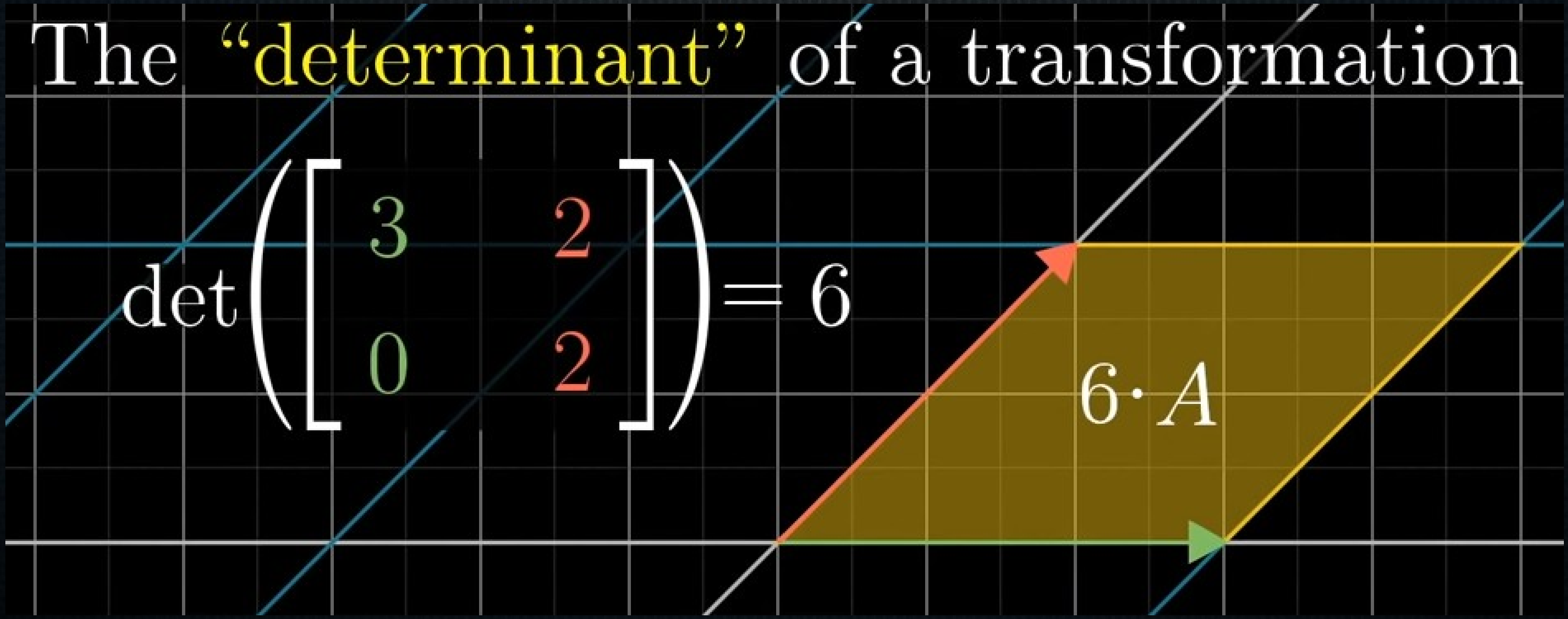
Start of PCA: determinant

- How do we get eigenvectors? Need to talk about determinant of a matrix.



Start of PCA: determinant

- How do we get eigenvectors? Need to talk about determinant of a matrix.



Start of PCA: determinant

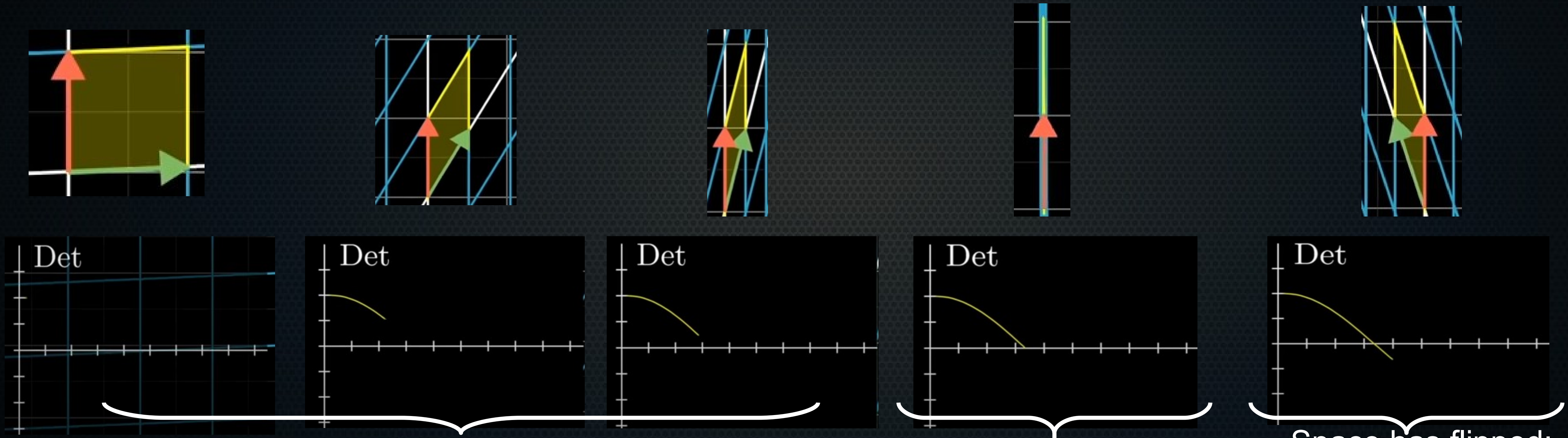
- How do we get eigenvectors? Need to talk about determinant of a matrix.
- So: matrices transform (rotate/squish) linear space. The determinant tells us how much the area enclosed by the basis vectors [0 ; 1] and [1; 0] changes.

Start of PCA: determinant

- How do we get eigenvectors? Need to talk about determinant of a matrix.
- So: matrices transform (rotate/squish) linear space. The determinant tells us how much the area enclosed by the basis vectors [0 ; 1] and [1; 0] changes.
- Caveat: determinant can be negative. How can area change negatively?

Start of PCA: determinant

- Negative determinant



Multiply by matrices that squish space more and more

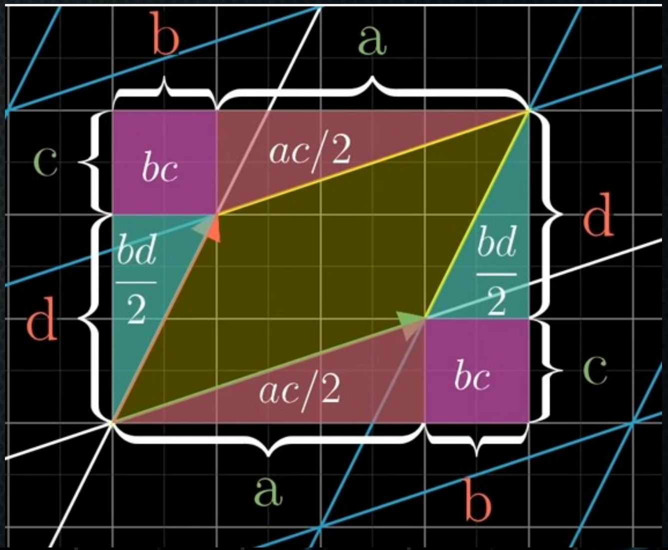
Space has flipped: negative determinant (see orientation of green and red vector)

Multiply with a matrix whose determinant is 0: squishes all of 2D space onto a 1D line

Start of PCA: determinant

- Now, with a feel for the determinant, its calculation:

$$\det\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right) = ad - bc$$



- As for *why* that's the calculation:
  watch 3Blue1Brown's video and see:

$$\det\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right) = (a+b)(c+d) - ac - bd - 2bc = ad - bc$$

Start of PCA: Identity matrix

- Now for the final piece of the linear algebra puzzle to get PCA to work, the identity matrix:

$$\begin{bmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{bmatrix} = \begin{bmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{bmatrix}$$

Identity matrix

Start of PCA: Identity matrix

- Now for the final piece of the linear algebra puzzle to get PCA to work, the identity matrix:

$$\begin{bmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 2.0 \cdot 1 + 0.8 \cdot 0 & 2.0 \cdot 0 + 0.8 \cdot 1 \\ 0.8 \cdot 1 + 0.6 \cdot 0 & 0.8 \cdot 0 + 0.6 \cdot 1 \end{bmatrix}$$

Start of PCA: Identity matrix

- Now for the final piece of the linear algebra puzzle to get PCA to work, the identity matrix:

$$\begin{bmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 2.0 \cdot 1 + \cancel{0.8 \cdot 0} & \cancel{2.0 \cdot 0} + 0.8 \cdot 1 \\ 0.8 \cdot 1 + \cancel{0.6 \cdot 0} & \cancel{0.8 \cdot 0} + 0.6 \cdot 1 \end{bmatrix} = \begin{bmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{bmatrix}$$

Start of PCA: calculating eigenvectors and eigenvalues

- Wanted to use:

  $A \cdot v = \lambda \cdot v \longrightarrow$ find v and lambda that make this true

Start of PCA: calculating eigenvectors and eigenvalues

- Wanted to use:

$A \cdot v = \lambda \cdot v \longrightarrow$ find v and lambda that make this true

- Left-hand side matrix-vector multiplication. Right-hand side scalar-vector multiplication. Use identity matrix to make it the same:

$A \cdot v = (\lambda \cdot I) \cdot v$

$$\begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} \lambda \cdot 1 \\ \lambda \cdot 2 \\ \lambda \cdot 3 \end{bmatrix} = \lambda \cdot v$$

Start of PCA: calculating eigenvectors and eigenvalues

- Wanted to use:

  $A \cdot v = \lambda \cdot v \longrightarrow$ find v and lambda that make this true

- Left-hand side matrix-vector multiplication. Right-hand side scalar-vector multiplication. Use identity matrix to make it the same:

  $A \cdot v = (\lambda \cdot I) \cdot v$

- Rewrite:

  $A \cdot v - \lambda \cdot I \cdot v = 0 \rightarrow \underbrace{(A - \lambda I)} \cdot v = 0$

$$\begin{bmatrix} 3-\lambda & 1 & 4 \\ 1 & 5-\lambda & 9 \\ 2 & 6 & 5-\lambda \end{bmatrix}$$

Start of PCA: calculating eigenvectors and eigenvalues

- Wanted to use:

  $A \cdot v = \lambda \cdot v \longrightarrow$ find v and lambda that make this true

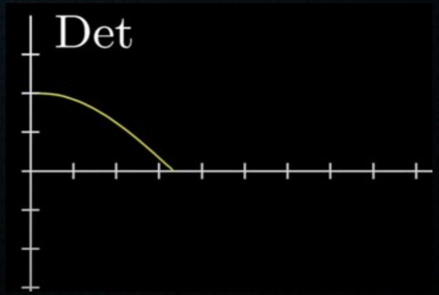
- Left-hand side matrix-vector multiplication. Right-hand side scalar-vector multiplication. Use identity matrix to make it the same:

  $A \cdot v = (\lambda \cdot I) \cdot v$

- Rewrite:

  $A \cdot v - \lambda \cdot I \cdot v = 0 \rightarrow (A - \lambda I) \cdot v = 0$

- Only way this can be true (if v != 0): $det(A - \lambda I) = 0$



Start of PCA: calculating eigenvectors and eigenvalues

- Finding eigenvalues (lambda‘s):

$$det(A-\lambda I)=0$$

$$A=\begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix} \qquad det\left(\begin{bmatrix} 3-\lambda & 1 \\ 0 & 2-\lambda \end{bmatrix}\right)=0$$

Start of PCA: calculating eigenvectors and eigenvalues

- Finding eigenvalues (lambda‘s):

$$det(A-\lambda I)=0$$

$$A=\begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix} \qquad det\left(\begin{bmatrix} 3-\lambda & 1 \\ 0 & 2-\lambda \end{bmatrix}\right)=0$$

$$\det\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right)=ad-bc$$

$$(3-\lambda)\cdot(2-\lambda)-1\cdot 0=0$$

$$\lambda=3 \vee \lambda=2$$

Start of PCA: calculating eigenvectors and eigenvalues

- Finding eigenvalues (lambda‘s):

$$det(A-\lambda I)=0$$

$$A=\begin{bmatrix}3 & 1\\0 & 2\end{bmatrix} \qquad det\left(\begin{bmatrix}3-\lambda & 1\\0 & 2-\lambda\end{bmatrix}\right)=0$$

$$\det\left(\begin{bmatrix}a & b\\c & d\end{bmatrix}\right)=ad-bc$$

$$(3-\lambda)\cdot(2-\lambda)-1\cdot 0=0$$

$$\lambda=3\vee\lambda=2$$

- Which eigenvectors correspond to these eigenvalues?

Start of PCA: calculating eigenvectors and eigenvalues

- Finding eigenvectors (v‘s):

$$(A-\lambda I)*v=0 \qquad (3-\lambda)\cdot(2-\lambda)-1\cdot 0=0$$

$$\lambda=3 \vee \lambda=2$$

Start of PCA: calculating eigenvectors and eigenvalues

- Finding eigenvectors (v‘s):

$$(A-\boxed{\lambda}I)*v=0$$

$$\boxed{\begin{gathered}(3-\lambda)\cdot(2-\lambda)-1\cdot 0=0\\ \lambda=3\vee\lambda=2\end{gathered}}$$

Know lambda‘s now

Start of PCA: calculating eigenvectors and eigenvalues

- Finding eigenvectors (v‘s):

$$(A-\lambda I)*v=0 \qquad (3-\lambda)\cdot(2-\lambda)-1\cdot 0=0$$

$$A=\begin{bmatrix}3 & 1\\ 0 & 2\end{bmatrix} \qquad \boxed{\lambda=3}\vee\lambda=2$$

$$\begin{bmatrix}3-\boxed{3} & 1\\ 0 & 2-\boxed{3}\end{bmatrix}\cdot\begin{bmatrix}x\\ y\end{bmatrix}=\begin{bmatrix}0\\ 0\end{bmatrix} \longrightarrow$$

Which vector [x; y] shows this behaviour?

$$\begin{bmatrix}0 & 1\\ 0 & -1\end{bmatrix}\cdot\begin{bmatrix}x\\ y\end{bmatrix}=\begin{bmatrix}0\\ 0\end{bmatrix}\rightarrow \left.\begin{matrix}0\cdot x+1\cdot y=0\\ 0\cdot x+-1\cdot y=0\end{matrix}\right\}$$

Start of PCA: calculating eigenvectors and eigenvalues

- Finding eigenvectors (v's):

$$(A-\lambda I)*v=0$$

$$(3-\lambda)\cdot(2-\lambda)-1\cdot 0=0$$

$$\boxed{\lambda=3}\vee\lambda=2$$

$$A=\begin{bmatrix}3 & 1\\ 0 & 2\end{bmatrix}$$

$$\begin{bmatrix}3-\boxed{3} & 1\\ 0 & 2-\boxed{3}\end{bmatrix}\cdot\begin{bmatrix}x\\ y\end{bmatrix}=\begin{bmatrix}0\\ 0\end{bmatrix} \longrightarrow$$

Which vector [x; y] shows this behaviour?

$$\begin{bmatrix}0 & 1\\ 0 & -1\end{bmatrix}\cdot\begin{bmatrix}x\\ y\end{bmatrix}=\begin{bmatrix}0\\ 0\end{bmatrix}\rightarrow \left.\begin{array}{c}0\cdot x+1\cdot y=0\\ 0\cdot x+-1\cdot y=0\end{array}\right\}$$

x → any number
y → 0

$$\boxed{\begin{bmatrix}1\\ 0\end{bmatrix}}\quad\begin{bmatrix}55\\ 0\end{bmatrix}\quad\begin{bmatrix}3.14\\ 0\end{bmatrix}$$

Start of PCA: calculating eigenvectors and eigenvalues

- Finding eigenvectors (v‘s):

$$(A-\lambda I)*v=0 \qquad (3-\lambda)\cdot(2-\lambda)-1\cdot 0=0$$

$$\lambda=3 \vee \boxed{\lambda=2}$$

$$A=\begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix}$$

$$\begin{bmatrix} 3-\boxed{2} & 1 \\ 0 & 2-\boxed{2} \end{bmatrix}\cdot\begin{bmatrix} x \\ y \end{bmatrix}=\begin{bmatrix} 0 \\ 0 \end{bmatrix} \longrightarrow$$

Which vector [x; y] shows this behaviour?

$$\begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}\cdot\begin{bmatrix} x \\ y \end{bmatrix}=\begin{bmatrix} 0 \\ 0 \end{bmatrix} \rightarrow \left.\begin{matrix} 1\cdot x+1\cdot y=0 \\ 0\cdot x+0\cdot y=0 \end{matrix}\right\} \text{x = -y}$$

$$\boxed{\begin{bmatrix} -1 \\ 1 \end{bmatrix}}$$

Start of PCA: calculating eigenvectors and eigenvalues

- Finding eigenvectors (v‘s):

$$(A-\lambda I)*v=0 \qquad (3-\lambda)\cdot(2-\lambda)-1\cdot 0=0$$

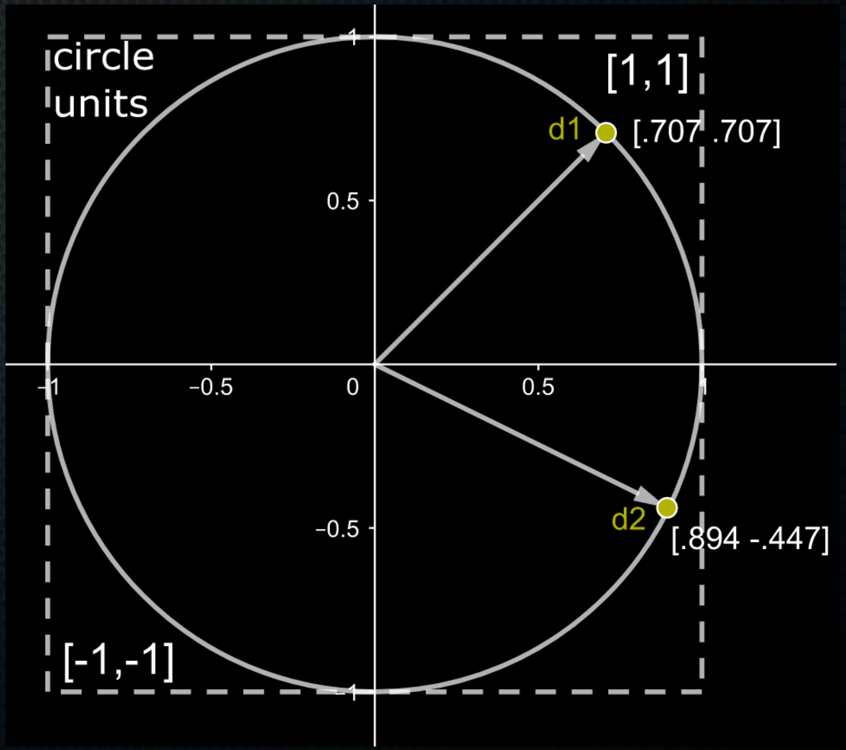
$$A=\begin{bmatrix}3 & 1\\ 0 & 2\end{bmatrix} \qquad \lambda=3 \vee \lambda=2$$

$$\begin{bmatrix}1\\0\end{bmatrix} \quad \begin{bmatrix}-1\\1\end{bmatrix}$$

Detail: want vectors of unit length. You can otherwise choose any scalar multiplication of ‚the‘ eigenvector as the eigenvector.

$$A*v=\lambda*v$$

$$\begin{bmatrix}-1\\1\end{bmatrix} \quad \begin{bmatrix}-5\\5\end{bmatrix} \quad \begin{bmatrix}-33.33\\33.33\end{bmatrix}$$



Start of PCA: calculating eigenvectors and eigenvalues

- Finding eigenvectors (v's):

$$(A-\lambda I)*v=0 \qquad (3-\lambda)\cdot(2-\lambda)-1\cdot 0=0$$

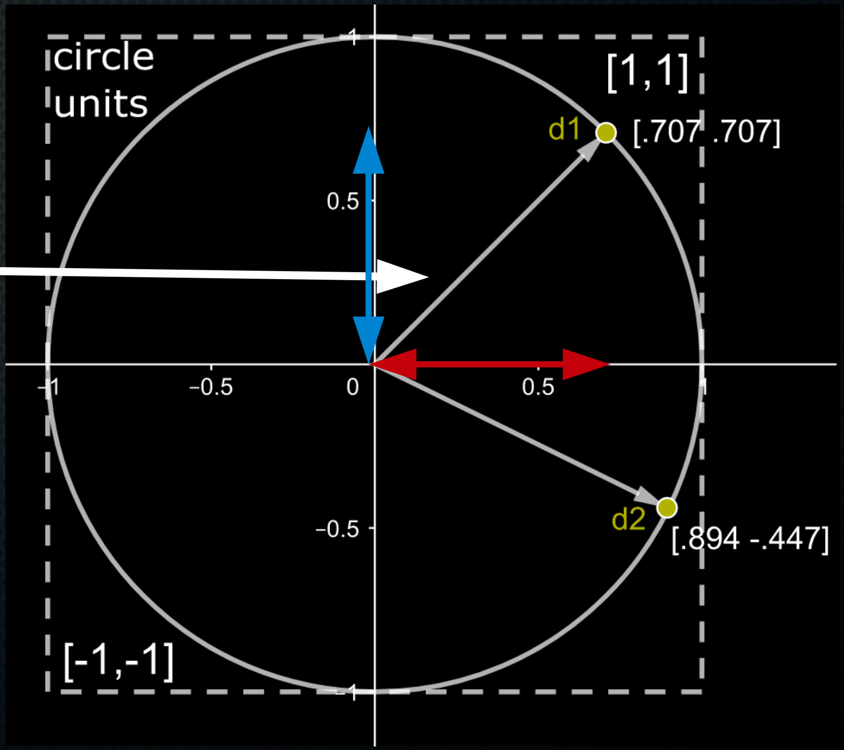
$$A=\begin{bmatrix}3 & 1\\ 0 & 2\end{bmatrix} \qquad \lambda=3 \vee \lambda=2$$

$$\begin{bmatrix}1\\ 0\end{bmatrix} \quad \begin{bmatrix}-1\\ 1\end{bmatrix}$$

Detail: want unit length

Length of this vector =
total distance from origin

$$\sqrt{(0.707^2+0.707^2)}=1$$



Start of PCA: calculating eigenvectors and eigenvalues

- Finding eigenvectors (v's):

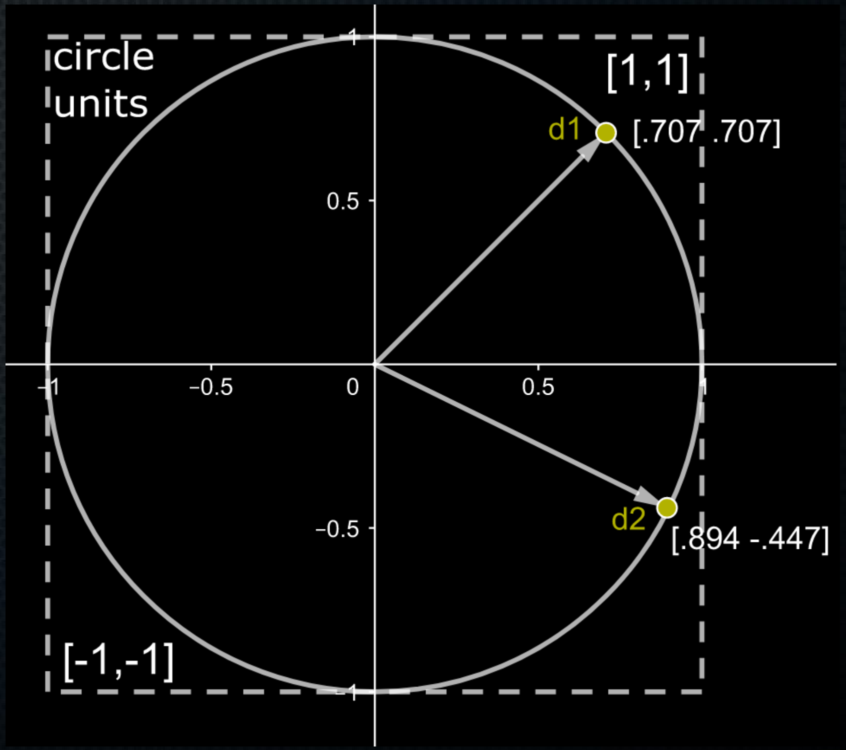
$$(A-\lambda I)*v=0 \qquad (3-\lambda)\cdot(2-\lambda)-1\cdot 0=0$$

$$A=\begin{bmatrix}3 & 1\\ 0 & 2\end{bmatrix} \qquad \lambda=3 \vee \lambda=2$$

$$\begin{bmatrix}1\\0\end{bmatrix} \quad \begin{bmatrix}-1\\1\end{bmatrix}$$

Detail: want unit length

$$\left\|\begin{bmatrix}-c\\c\end{bmatrix}\right\|_2=\sqrt{(c^2+(-c)^2)}=1$$



Start of PCA: calculating eigenvectors and eigenvalues

- Finding eigenvectors (v's):

$$(A-\lambda I)*v=0 \qquad (3-\lambda)\cdot(2-\lambda)-1\cdot 0=0$$

$$A=\begin{bmatrix}3 & 1\\ 0 & 2\end{bmatrix} \qquad \lambda=3 \vee \lambda=2$$

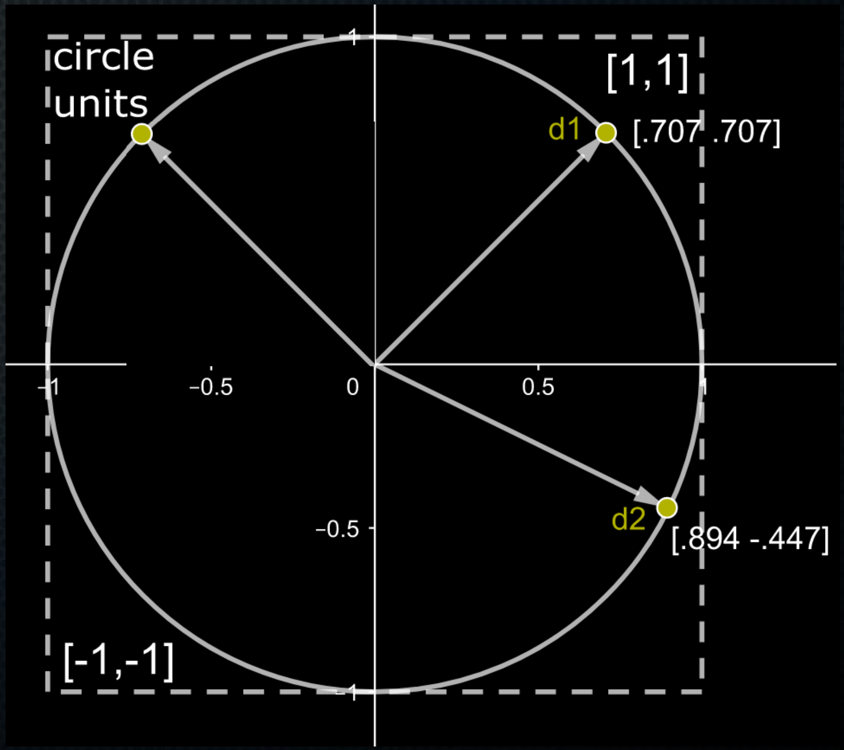
$$\begin{bmatrix}1\\ 0\end{bmatrix} \quad \begin{bmatrix}-1\\ 1\end{bmatrix}$$

Detail: want unit length

$$\left\|\begin{bmatrix}-c\\ c\end{bmatrix}\right\|_2=\sqrt{(2c^2)}=1$$

$$\sqrt{(2)}\cdot\sqrt{(c^2)}=1$$

$$c=\frac{1}{\sqrt{(2)}}$$



Start of PCA: calculating eigenvectors and eigenvalues

- Finding eigenvectors (v‘s):

$$(A-\lambda I)*v=0$$

$$A=\begin{bmatrix}3 & 1\\ 0 & 2\end{bmatrix}$$

$$(3-\lambda)\cdot(2-\lambda)-1\cdot 0=0$$

$$\lambda=3 \vee \lambda=2$$

$$\begin{bmatrix}1\\ 0\end{bmatrix} \quad \begin{bmatrix}-1\\ 1\end{bmatrix}$$

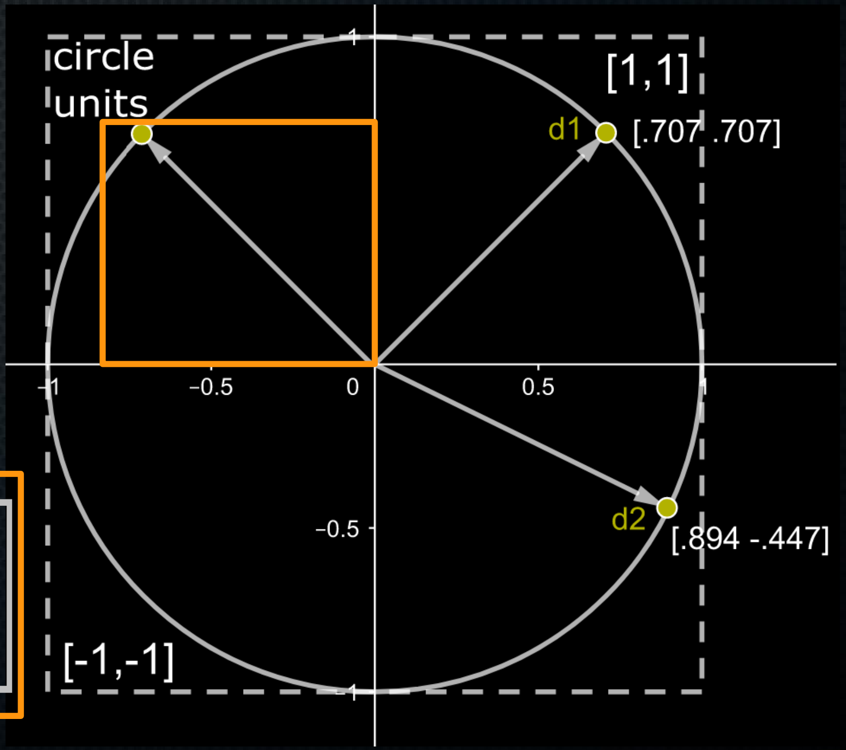
Detail: want unit length

$$\left\|\begin{bmatrix}-c\\ c\end{bmatrix}\right\|_2=\sqrt{(2c^2)}=1$$

$$\sqrt{(2)}\cdot\sqrt{(c^2)}=1$$

$$c=\frac{1}{\sqrt{(2)}}$$

$$\begin{bmatrix}\frac{-1}{\sqrt{(2)}}\\ \frac{1}{\sqrt{(2)}}\end{bmatrix}\approx\boxed{\begin{bmatrix}-.707\\ .707\end{bmatrix}}$$

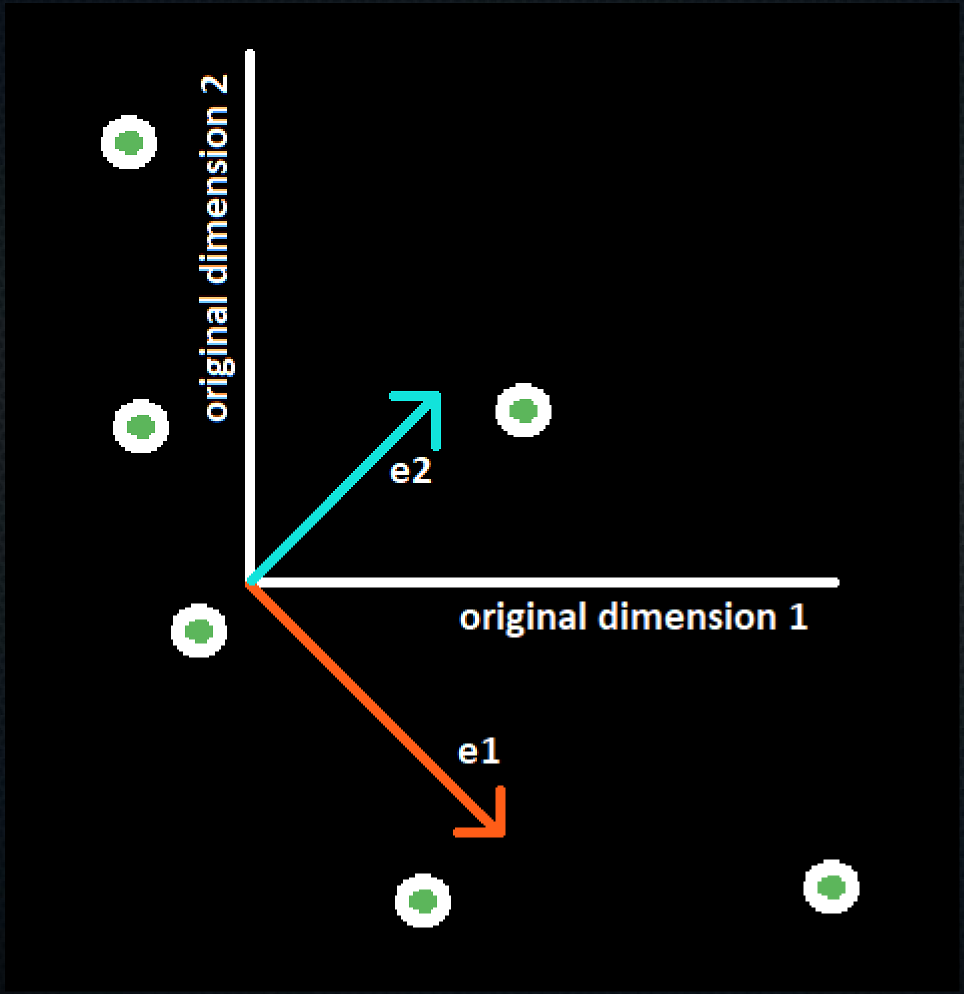


So far

- We know that eigenvectors are the linear axes among which data varies most
- We know how to get eigenvectors
- Now, we want to take our original data and project it onto a *subset* of the eigenvectors → keeping only some of the dimensions!

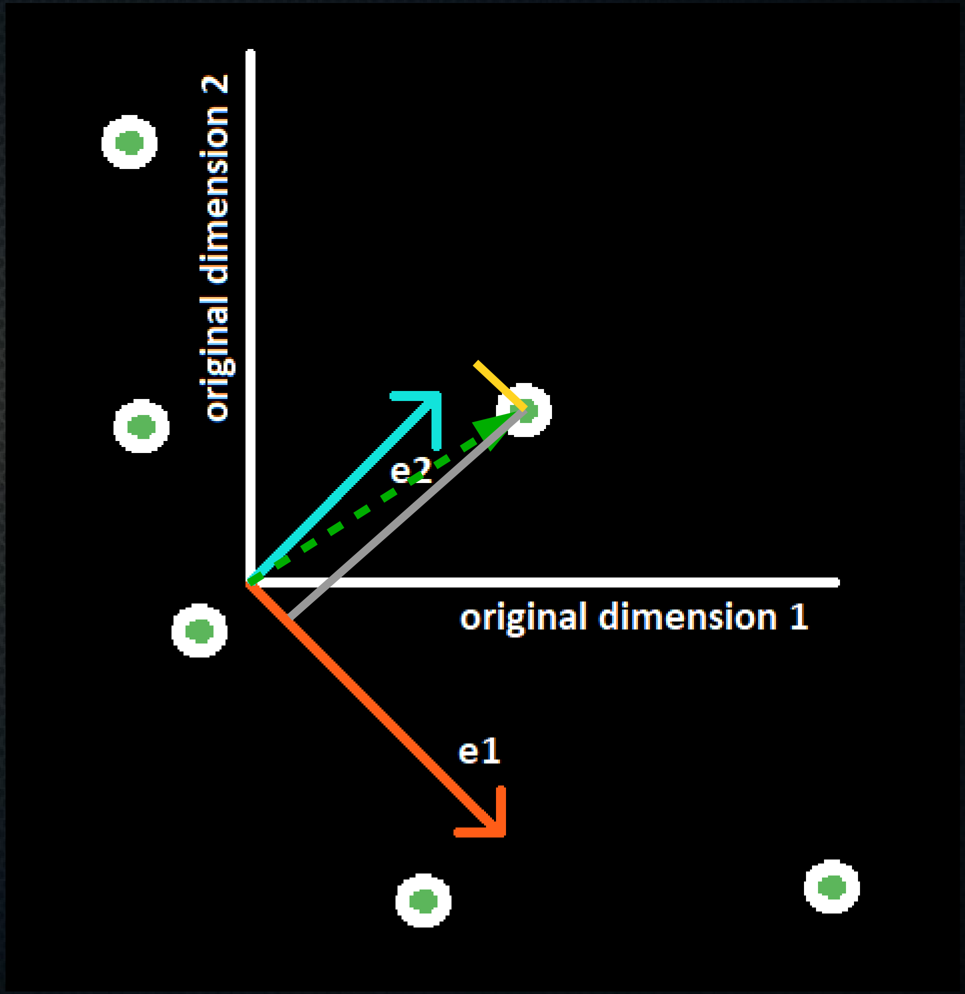
Projecting data

- Visually



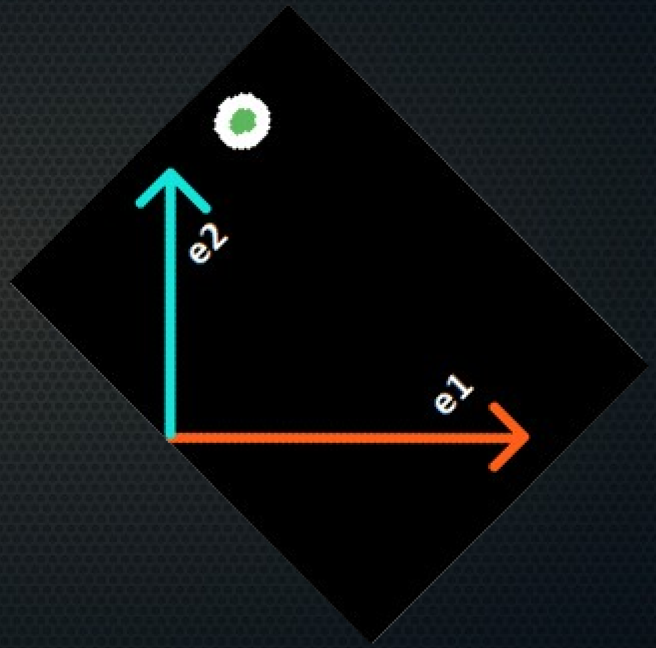
Projecting data

- Visually:
- Subtract the mean (let's assume 0-centered already)
- Project onto each eigenvector



Projecting data

- Visually:
- Subtract the mean (let’s assume 0-centered already)
- Project onto each eigenvector



Projecting data

- 3D:

$$\begin{array}{cc} & \begin{array}{ccc} \text{Feat 1} & \text{Feat 2} & \text{Feat 3} \end{array} \\ \begin{array}{c} \text{Sample 1} \\ \text{Sample 2} \\ \text{Sample 3} \end{array} & \begin{bmatrix} 33 & 8 & 13 \\ 5 & 90 & 54 \\ 12 & 8 & 7 \end{bmatrix} \end{array}$$

$\mu_1 = 16.67$ $\mu_2 = 35.33$ $\mu_3 = 24.67$

Projecting data

- 3D:

$$
\begin{array}{c|ccc}
 & \text{Feat 1} & \text{Feat 2} & \text{Feat 3} \\
\hline
\text{Sample 1} & & & \\
\text{Sample 2} & & & \\
\text{Sample 3} & & &
\end{array}
\quad
\begin{array}{r}
\text{Sample 1} \\ \text{Sample 2} \\ \text{Sample 3}
\end{array}
\begin{bmatrix} 33 & 8 & 13 \\ 5 & 90 & 54 \\ 12 & 8 & 7 \end{bmatrix}
\xrightarrow[\text{Mean-center}]{}
\begin{bmatrix} 16.33 & -27.33 & -11.67 \\ -11.67 & 54.67 & 29.33 \\ -4.67 & -27.33 & -17.67 \end{bmatrix}
$$

$\mu_1 = 16.67$ $\mu_2 = 35.33$ $\mu_3 = 24.67$

Projecting data

- 3D:

$$
\begin{array}{r c}
 & \begin{array}{ccc}\text{Feat 1} & \text{Feat 2} & \text{Feat 3}\end{array} \\
\begin{array}{r}\text{Sample 1}\\ \text{Sample 2}\\ \text{Sample 3}\end{array} & \begin{bmatrix} 33 & 8 & 13 \\ 5 & 90 & 54 \\ 12 & 8 & 7 \end{bmatrix}
\end{array}
\xrightarrow{\text{Mean-center}}
\begin{bmatrix} 16.33 & -27.33 & -11.67 \\ -11.67 & 54.67 & 29.33 \\ -4.67 & -27.33 & -17.67 \end{bmatrix}
$$

$\mu_1 = 16.67$ $\mu_2 = 35.33$ $\mu_3 = 24.67$

Calculate covariance matrix:

$$Cov(feat_1, feat_2) = \frac{1}{n}\sum_{n=1}^{n} (\text{feat1} - \cancel{\mu_{feat1}}) \cdot (\text{feat2} - \cancel{\mu_{feat2}})$$

$$
\begin{bmatrix} 212.33 & -478.33 & -225.17 \\ -478.33 & 2241.33 & 1202.67 \\ -225.17 & 1202.67 & 654.33 \end{bmatrix}
$$

Projecting data

- 3D:

$$\begin{array}{r c}
 & \begin{array}{ccc}\text{Feat 1} & \text{Feat 2} & \text{Feat 3}\end{array} \\
\begin{array}{r}\text{Sample 1} \\ \text{Sample 2} \\ \text{Sample 3}\end{array} & \begin{bmatrix} 33 & 8 & 13 \\ 5 & 90 & 54 \\ 12 & 8 & 7 \end{bmatrix}
\end{array}
\xrightarrow[\text{Mean-center}]{}
\begin{bmatrix} 16.33 & -27.33 & -11.67 \\ -11.67 & 54.67 & 29.33 \\ -4.67 & -27.33 & -17.67 \end{bmatrix}$$

$\mu_1 = 16.67$ $\mu_2 = 35.33$ $\mu_3 = 24.67$

$$\begin{bmatrix} 212.33 & -478.33 & -225.17 \\ -478.33 & 2241.33 & 1202.67 \\ -225.17 & 1202.67 & 654.33 \end{bmatrix}$$

$$A \cdot v = (\lambda \cdot I) \cdot v$$

Find eigenvalues and eigenvectors as explained

Projecting data

- 3D:

$$\begin{array}{r c} & \begin{array}{ccc} \text{Feat 1} & \text{Feat 2} & \text{Feat 3} \end{array} \\ \begin{array}{r} \text{Sample 1} \\ \text{Sample 2} \\ \text{Sample 3} \end{array} & \begin{bmatrix} 16.33 & -27.33 & -11.67 \\ -11.67 & 54.67 & 29.33 \\ -4.67 & -27.33 & -17.67 \end{bmatrix} \end{array}$$

Cov matrix
$$\begin{bmatrix} 212.33 & -478.33 & -225.17 \\ -478.33 & 2241.33 & 1202.67 \\ -225.17 & 1202.67 & 654.33 \end{bmatrix}$$

Calculate eigenvalues $\lambda_1 \approx 2989\,;\,\lambda_2 \approx 119\,;\,\lambda_3 \approx 0$

Eigenvectors belonging to them
$$e_1 = \begin{bmatrix} 0.187 \\ -0.866 \\ -0.46 \end{bmatrix};\, e_2 = \begin{bmatrix} 0.953 \\ 0.045 \\ 0.299 \end{bmatrix};\, e_3 = \begin{bmatrix} -0.238 \\ -0.498 \\ 0.834 \end{bmatrix}$$

Projecting data

- 3D:

To make PC1, combine:
0.187 part feature 1
-0.866 part feature 2
-0.46 part feature 3

In other words: feature 2 has the largest influence on the coordinate on PC1. This is logical: it varies the most!

$$\begin{array}{cc} & \begin{array}{ccc} \text{Feat 1} & \text{Feat 2} & \text{Feat 3} \end{array} \\ \begin{array}{c} \text{Sample 1} \\ \text{Sample 2} \\ \text{Sample 3} \end{array} & \begin{bmatrix} 16.33 & -27.33 & -11.67 \\ -11.67 & 54.67 & 29.33 \\ -4.67 & -27.33 & -17.67 \end{bmatrix} \end{array}$$

Calculate eigenvalues

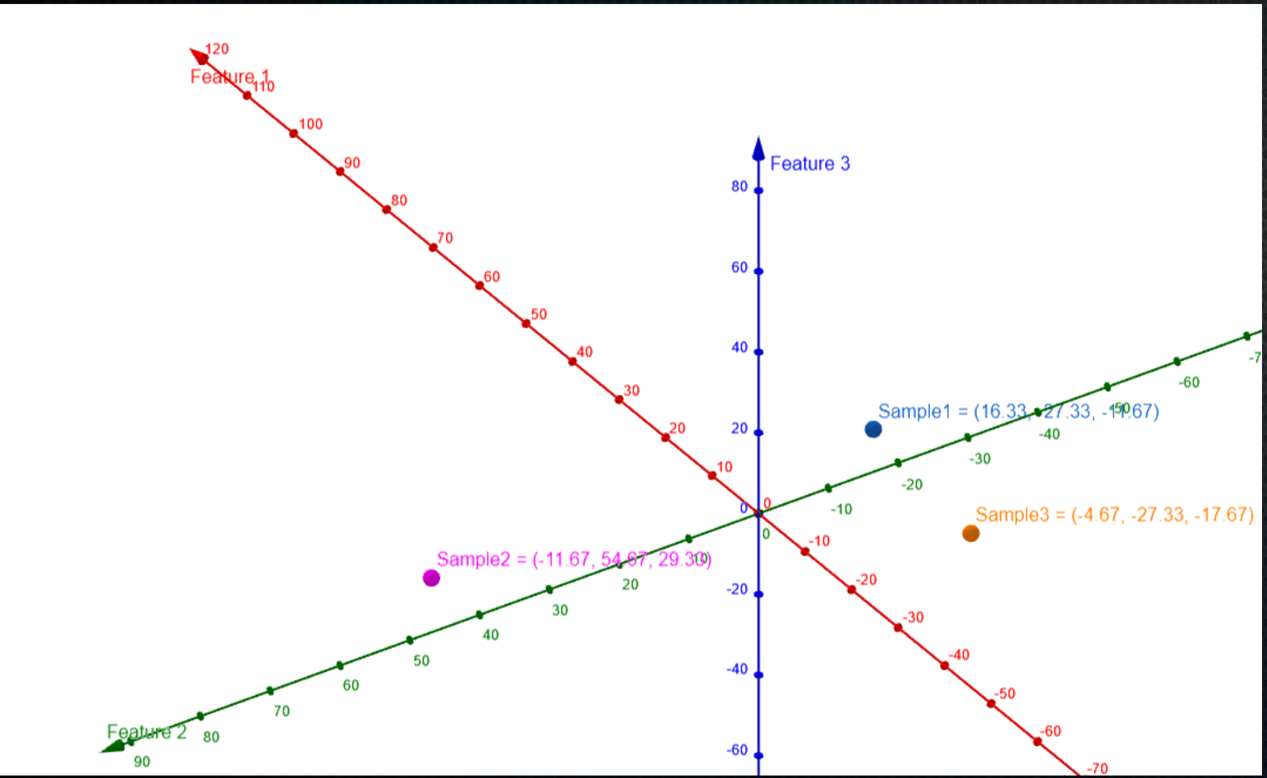
$$\lambda_1 \approx 2989 \,;\, \lambda_2 \approx 119 \,;\, \lambda_3 \approx 0$$

Eigenvectors belonging to them

$$\boxed{e_1 = \begin{bmatrix} 0.187 \\ -0.866 \\ -0.46 \end{bmatrix}} ; e_2 = \begin{bmatrix} 0.953 \\ 0.045 \\ 0.299 \end{bmatrix} ; e_3 = \begin{bmatrix} -0.238 \\ -0.498 \\ 0.834 \end{bmatrix}$$

Projecting data

- How does this look?



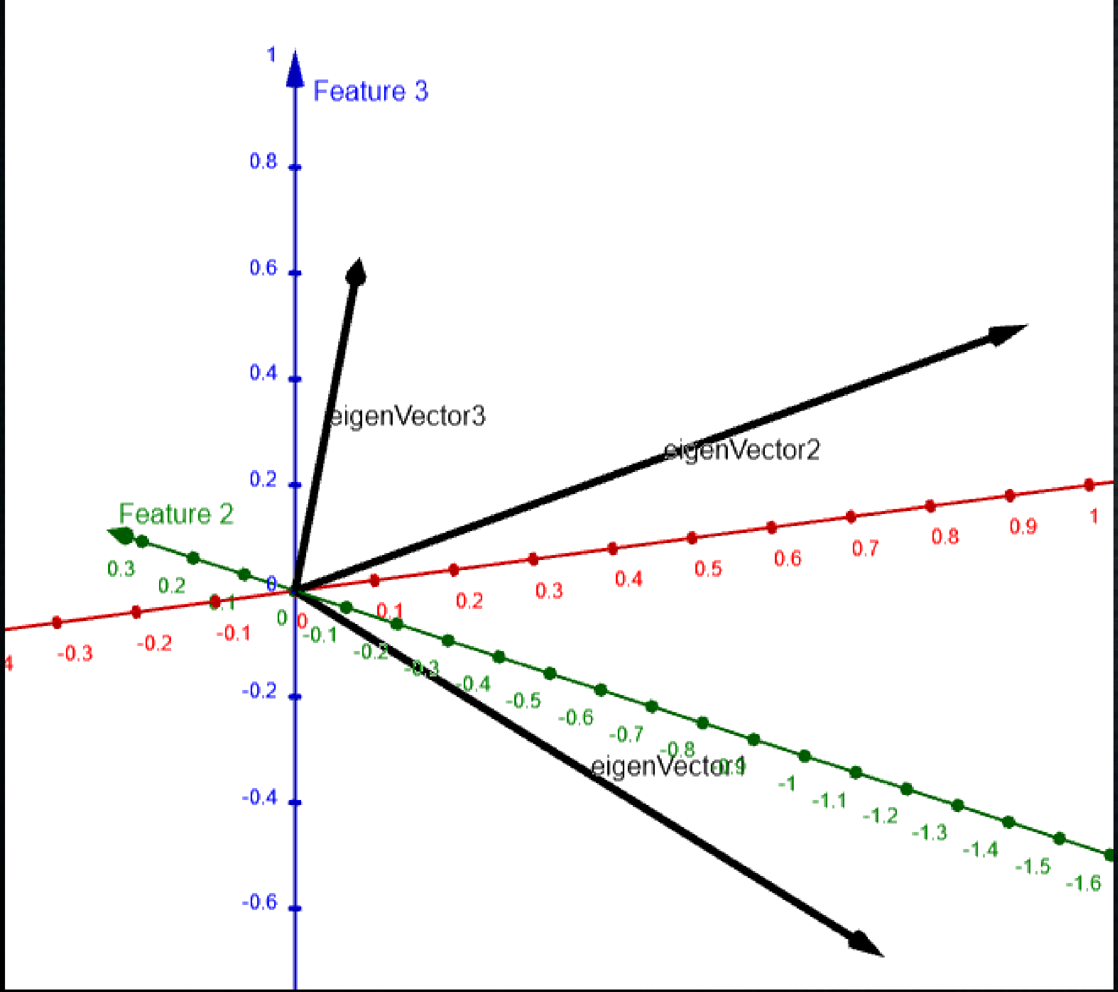
$$\begin{array}{c} \\ \text{Sample 1} \\ \text{Sample 2} \\ \text{Sample 3} \end{array} \begin{array}{c} \begin{array}{ccc} \text{Feat 1} & \text{Feat 2} & \text{Feat 3} \end{array} \\ \begin{bmatrix} 16.33 & -27.33 & -11.67 \\ -11.67 & 54.67 & 29.33 \\ -4.67 & -27.33 & -17.67 \end{bmatrix} \end{array}$$

$$\lambda_1 \approx 2989 \,;\, \lambda_2 \approx 119 \,;\, \lambda_3 \approx 0$$

$$e_1 = \begin{bmatrix} 0.187 \\ -0.866 \\ -0.46 \end{bmatrix} ; e_2 = \begin{bmatrix} 0.953 \\ 0.045 \\ 0.299 \end{bmatrix} ; e_3 = \begin{bmatrix} -0.238 \\ -0.498 \\ 0.834 \end{bmatrix}$$

Projecting data

- How does this look?



$$\begin{array}{c} \\ \text{Sample 1} \\ \text{Sample 2} \\ \text{Sample 3} \end{array} \begin{array}{ccc} \text{Feat 1} & \text{Feat 2} & \text{Feat 3} \\ \left[\begin{array}{ccc} 16.33 & -27.33 & -11.67 \\ -11.67 & 54.67 & 29.33 \\ -4.67 & -27.33 & -17.67 \end{array}\right] \end{array}$$

$$\lambda_1 \approx 2989\,;\, \lambda_2 \approx 119\,;\, \lambda_3 \approx 0$$

$$\boxed{e_1 = \begin{bmatrix} 0.187 \\ -0.866 \\ -0.46 \end{bmatrix}; e_2 = \begin{bmatrix} 0.953 \\ 0.045 \\ 0.299 \end{bmatrix}; e_3 = \begin{bmatrix} -0.238 \\ -0.498 \\ 0.834 \end{bmatrix}}$$

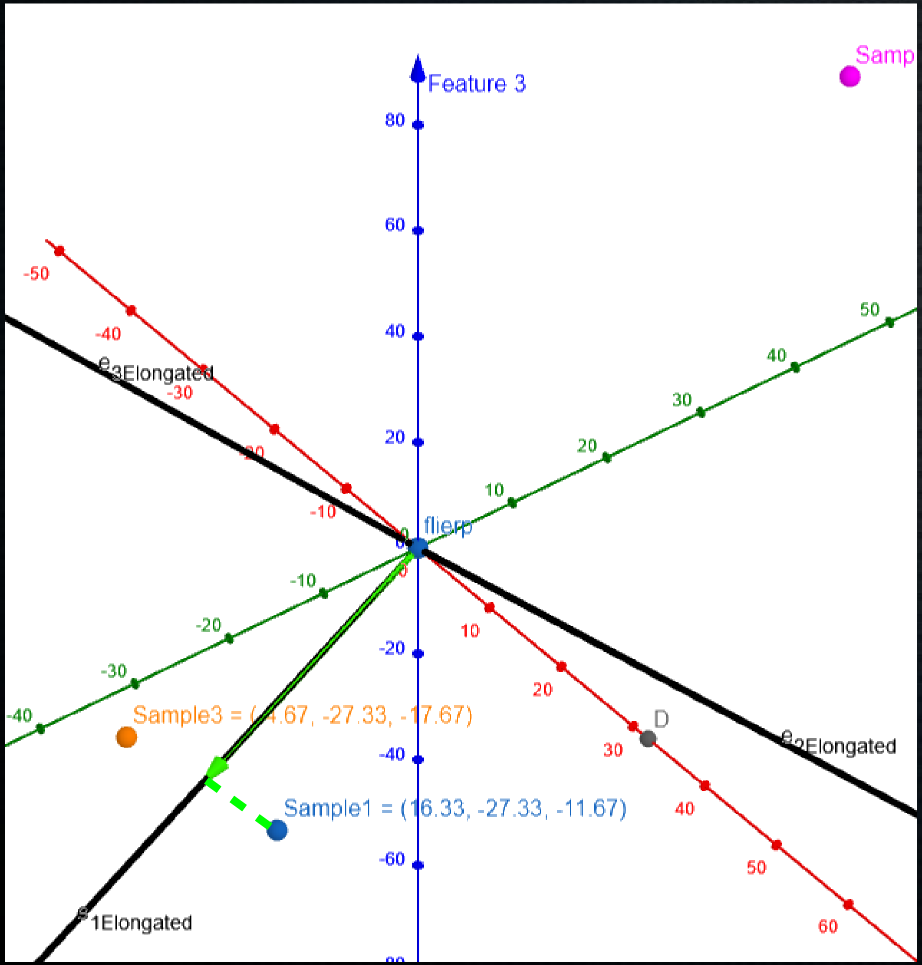
Projecting data

- Now: take the data points we have, and project them onto a *subset* of original eigenvectors. In this case on the first 2.
- Multiply its value on each of the original axes with the eigenvector (recipe for how much each of the original axes contributes to the value on that PC).
- Sample 1 PC1:

$$\begin{bmatrix} 16.33 & -27.33 & -11.67 \end{bmatrix} \cdot \overbrace{\begin{bmatrix} 0.187 \\ -0.866 \\ -0.46 \end{bmatrix}}^{e_1} \approx 32.09$$

Projecting data

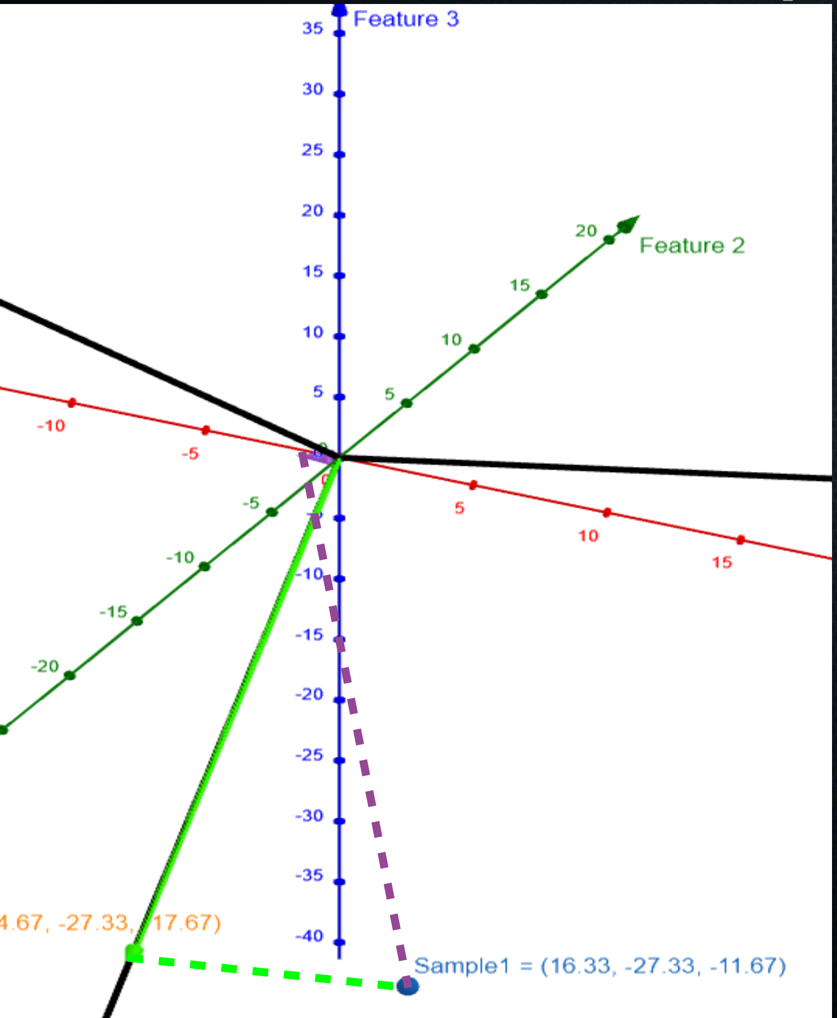
- Sample 1 PC1: $\begin{bmatrix} 16.33 & -27.33 & -11.67 \end{bmatrix} \cdot \begin{bmatrix} 0.953 \\ 0.045 \\ -0.46 \end{bmatrix} \approx 32.09$



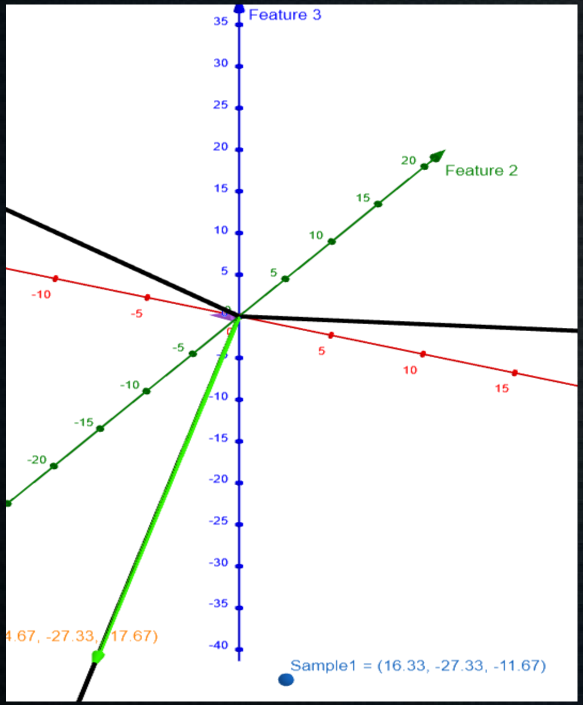
Projecting data

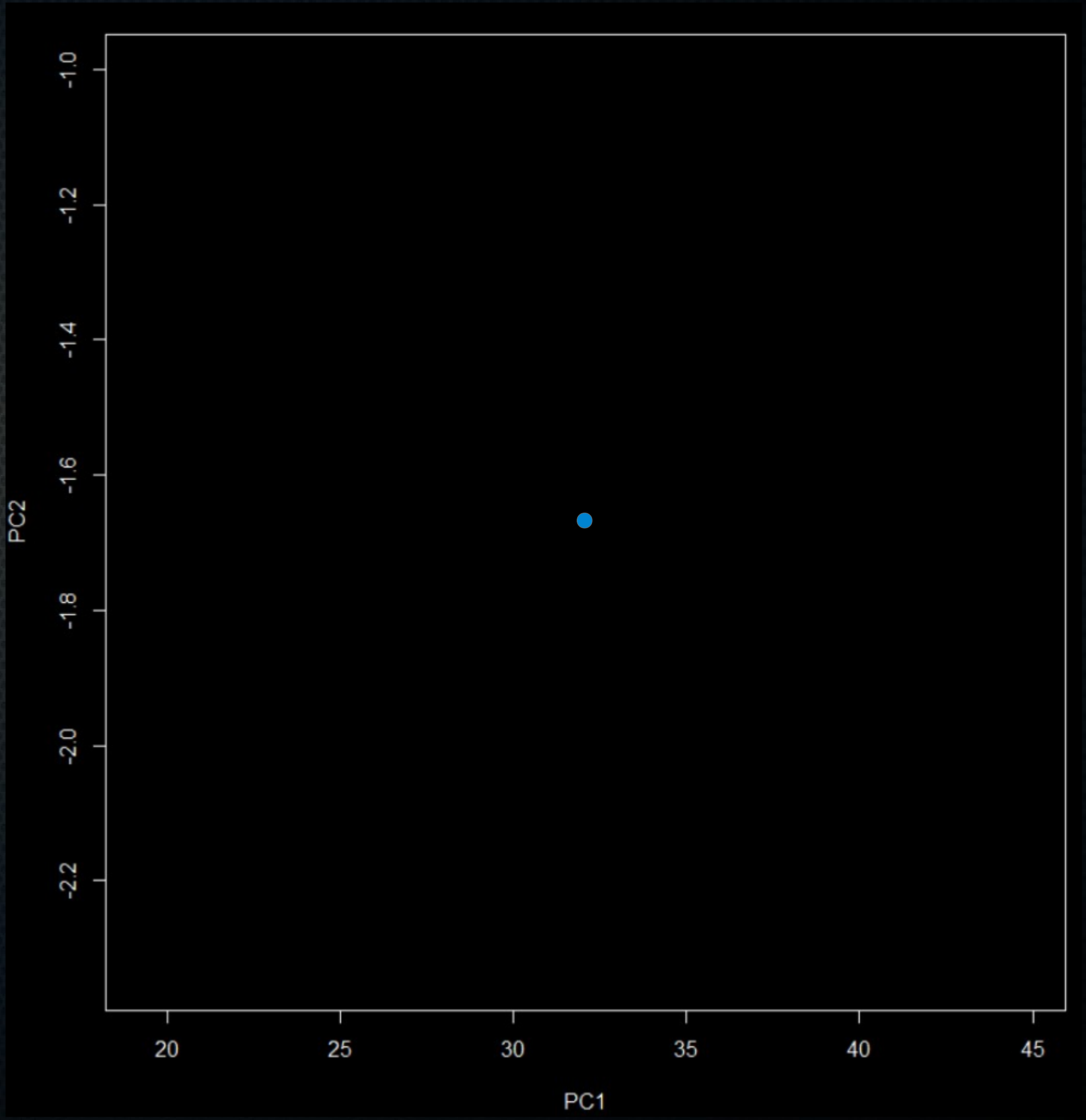
- Sample 1 PC1: $\begin{bmatrix} 16.33 & -27.33 & -11.67 \end{bmatrix} \cdot \begin{bmatrix} 0.187 \\ -0.866 \\ -0.46 \end{bmatrix} \approx 32.09$

$$\begin{bmatrix} 16.33 & -27.33 & -11.67 \end{bmatrix} \cdot \overbrace{\begin{bmatrix} 0.187 \\ -0.866 \\ 0.299 \end{bmatrix}}^{e_2} \approx -1.67$$



Projecting data



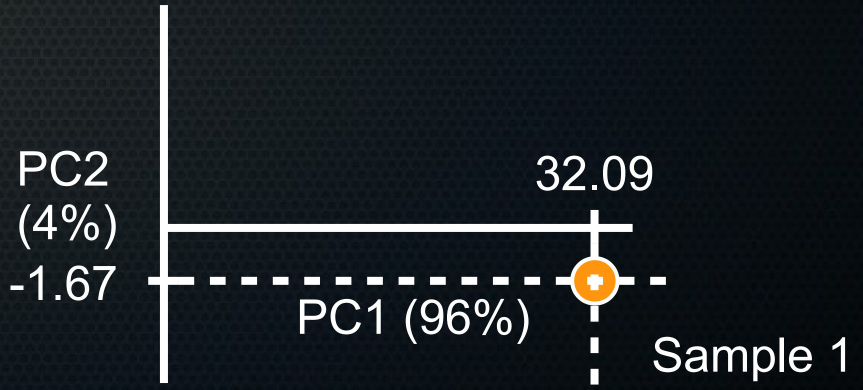


Variance explained

- The eigenvalue (λ) belonging to each (unit) eigenvector is the amount of variance on that eigenvector → to calculate %variance per PC just do eigenvalue/sum(eigenvalues)*100%
- PC1: 2989/(2989+119)*100% ~ 96%

$$\lambda_1 \approx 2989\,;\,\lambda_2 \approx 119\,;\,\lambda_3 \approx 0$$

- PC2: ~4%



Practical Break 2

- Let them watch PCA video StatQuest, read through Lindsey manuscript?
- Program their own PCA algorithm (though using SVD?) → or not do it yourself?
- Do it on eigenfaces dataset
- Do it on some biological data
- Quickly show how to do it with scikit-learn