Principal Component Analysis (PCA)

Presented By: Suresh Pokharel
The University of Queensland

Dimensionality Reduction

Fereshteh Sadeghi

CSEP 546

PCA : Outlines

- What
- When
- How
- Why

What is PCA?

Task : Predict Gross Domestic Product (GDP) of Australia for 2018.

Variables:

- GDP for the first quarter of 2018
- The Australia GDP for the entirety of 2017, 2016, and so on
- unemployment rate
- inflation rate
- stock price data,
- the number of IPOs
- ....

- You might ask the question, "How do I take all of the variables I've collected and focus on only a few of them?"
- In technical terms, you want to "reduce the dimension of your feature space."

Somewhat unsurprisingly, ***reducing*** the ***dimension*** of the feature space is called "***dimensionality reduction***."

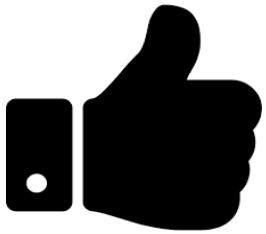
What is PCA?

Approaches for dimensionality reduction:

- **Feature Elimination:**
  - we reduce the feature space by eliminating features. In the GDP example above, instead of considering every single variable, we might drop all variables except the three we think will best predict what the Australia's gross domestic product will look like.
  - **Advantages:** Simplicity and maintaining the interpretability of variables.
  - **Disadvantages:** By eliminating features, we've also entirely eliminated any benefits those dropped variables would bring.

- **Feature Extraction:**
  - Transformation of raw data into features suitable for modelling.
  - For dimensionality reduction, we keep as many of the new independent variables as we want, but we drop the "least important ones."

Where is PCA?



What is PCA?

- PCA was invented by Karl Pearson in 1901.
- PCA is a technique for feature extraction.
- It combines our input variables in a specific way, then we can drop the "least important" variables while still retaining the most valuable parts of all of the variables!
- As an added benefit, each of the "new" variables after PCA are all independent of one another.
- This is a benefit because the assumptions of a linear model require our independent variables to be independent of one another.

Karl Pearson
FRS

Pearson in 1912

Born	Carl Pearson 27 March 1857 Islington, London, England
Died	27 April 1936 (aged 79) Coldharbour, Surrey, England
Residence	England
Nationality	British
Alma mater	King's College, Cambridge University of Heidelberg
Known for	Principal Component Analysis Pearson distribution Pearson's r Pearson's chi-squared test Phi coefficient
Awards	Darwin Medal (1898) Weldon Memorial Prize (1912)

Karl Pearson, father of mathematical statistics (1857-1936)

When should I use PCA?

1. Do you want to reduce the number of variables, but aren't able to identify variables to completely remove from consideration?

2. Do you want to ensure your variables are independent of one another?

3. Are you comfortable making your independent variables less interpretable?

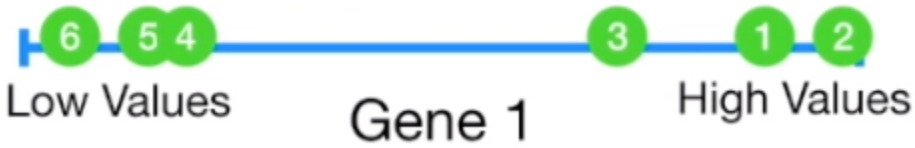
If you answered "yes" to all three questions, then PCA is a good method to use. If you answered "no" to question 3, you **should not** use PCA.

How does PCA work?

Transcriptions of Genes						
	Mouse 1	**Mouse 2**	**Mouse 3**	**Mouse 4**	**Mouse 5**	**Mouse 6**
Gene 1	10	11	8	3	2	1

Consider Mice as samples and Genes as variables

If we only measure 1 gene, we can plot the data on a number line…



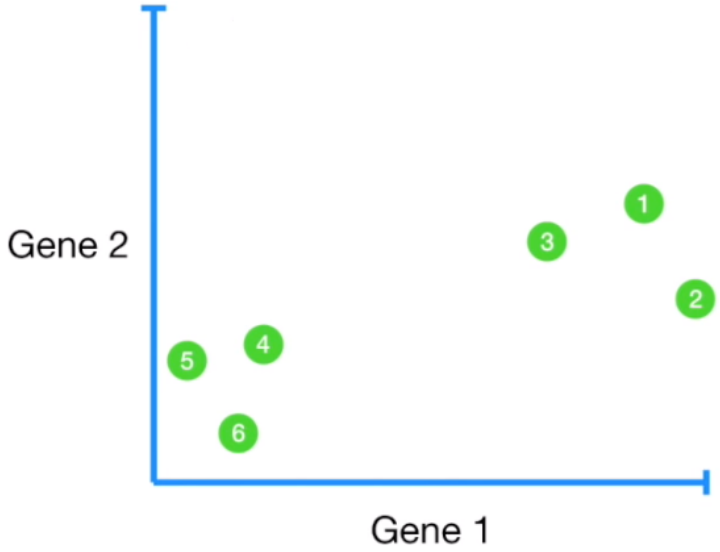
Even though it's a simple graph, it shows us that mice 1, 2 and 3 are similar to each other than they are to mice 4, 5, 6.

How does PCA work?

Transcriptions of Genes						
	Mouse 1	Mouse 2	Mouse 3	Mouse 4	Mouse 5	Mouse 6
Gene 1	10	11	8	3	2	1
Gene 2	6	4	5	3	2.8	1

If we measured 2 genes ….

Gene 2 is the y-axis and spans one of the other dimensions.

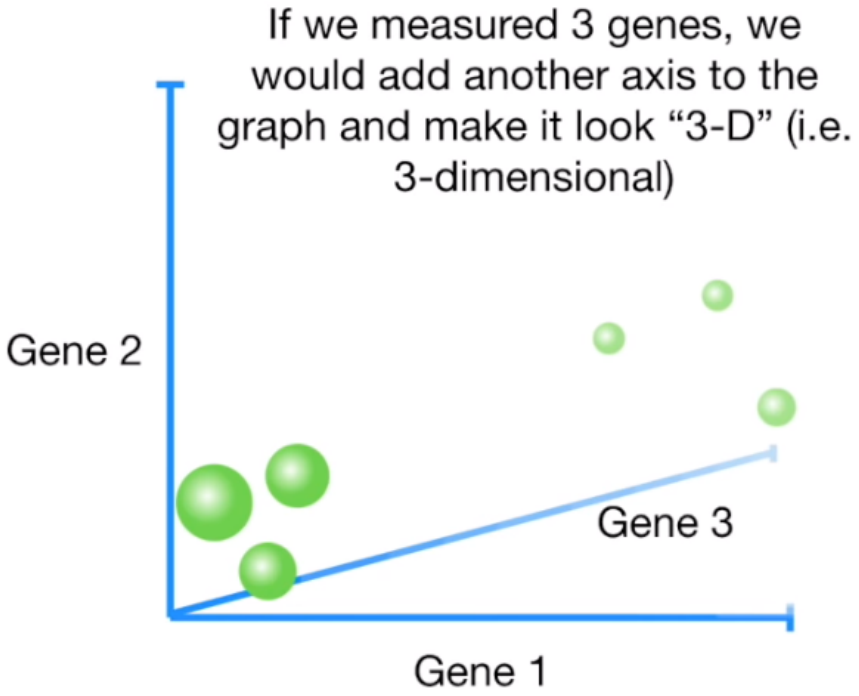


Gene 1 is the x-axis and spans one of the 2 dimensions in this graph

- We can see that mice 1, 2, and 3 cluster on the right side
- Mice 4, 5, and 6 cluster on the lower left side.

How does PCA work?

	Mouse 1	Mouse 2	Mouse 3	Mouse 4	Mouse 5	Mouse 6
Gene 1	10	11	8	3	2	1
Gene 2	6	4	5	3	2.8	1
Gene 3	12	9	10	2.5	1.3	2

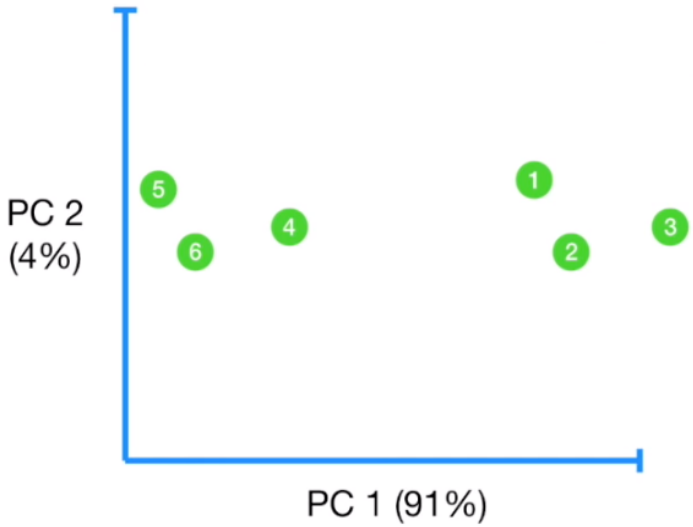


How does PCA work?

Transcriptions of Genes						
	Mouse 1	**Mouse 2**	**Mouse 3**	**Mouse 4**	**Mouse 5**	**Mouse 6**
Gene 1	10	11	8	3	2	1
Gene 2	6	4	5	3	2.8	1
Gene 3	12	9	10	2.5	1.3	2
Gene 4	5	7	6	2	4	7

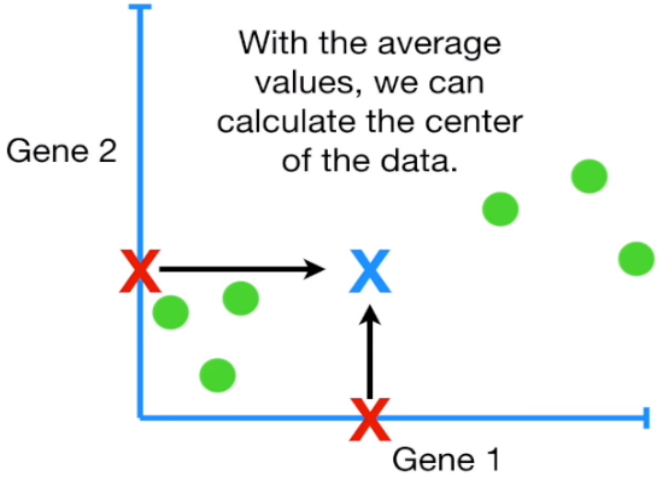
If we measured 4 genes, however, we can no longer plot the data – 4 genes require 4 dimensions

- **So, we're going to talk about how PCA can take 4 or more gene measurements (and thus, 4 or more dimensions of data), and make 2-D PCA plot.**
- **How accurately the 2-D graph is.**



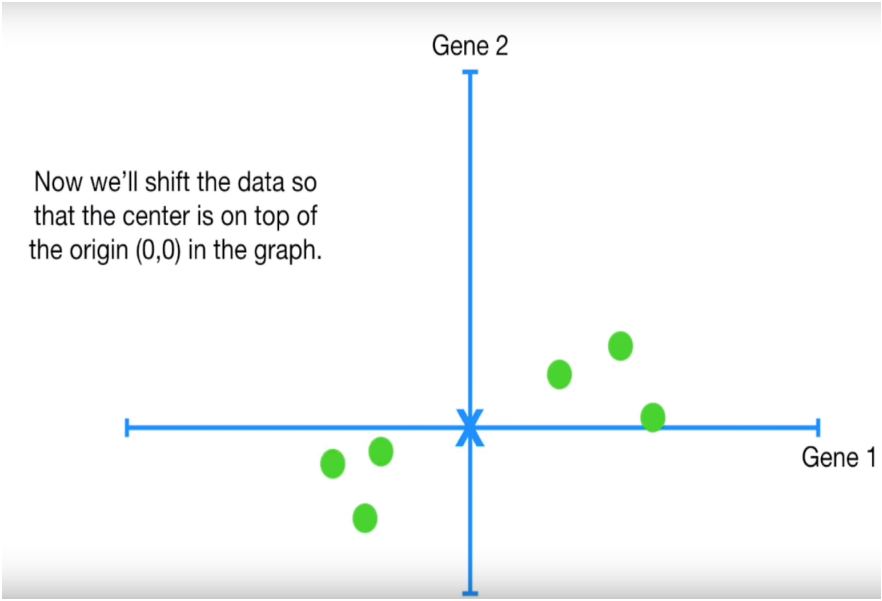
How does PCA work?

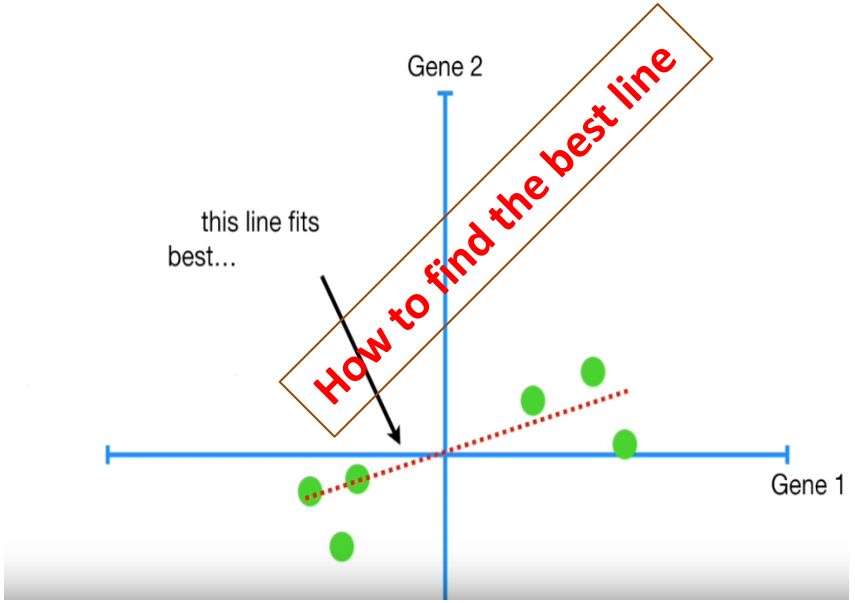
Transcriptions of Genes						
	Mouse 1	**Mouse 2**	**Mouse 3**	**Mouse 4**	**Mouse 5**	**Mouse 6**
Gene 1	10	11	8	3	2	1
Gene 2	6	4	5	3	2.8	1



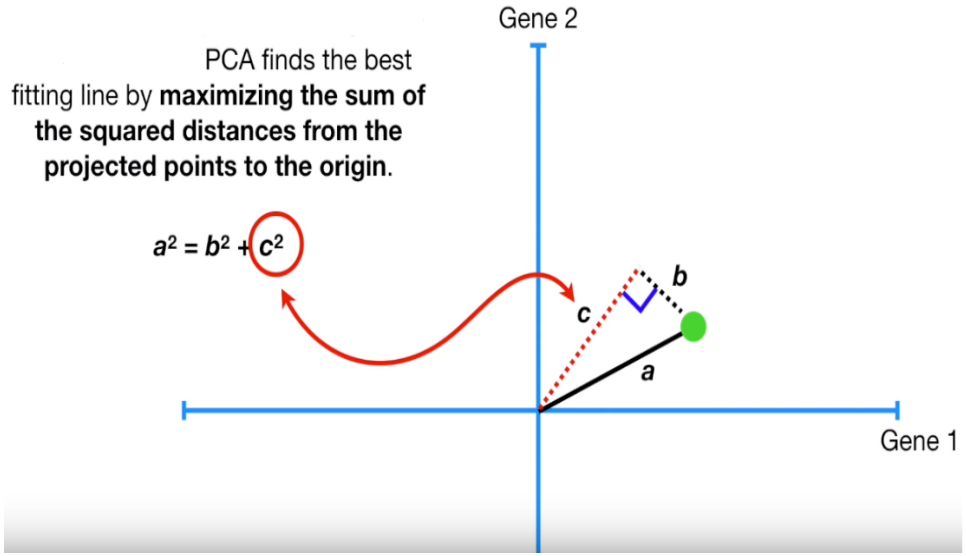
Find the best fitting line.

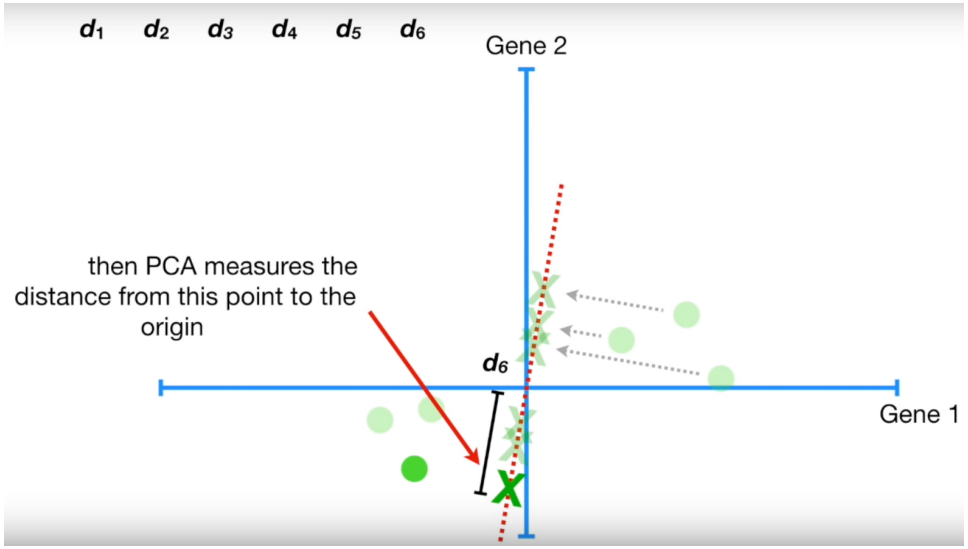
Note: Shifting the data did not change how the data points are positioned relative to each other.





How does PCA work?





Next, Square them so that negative values don't cancel

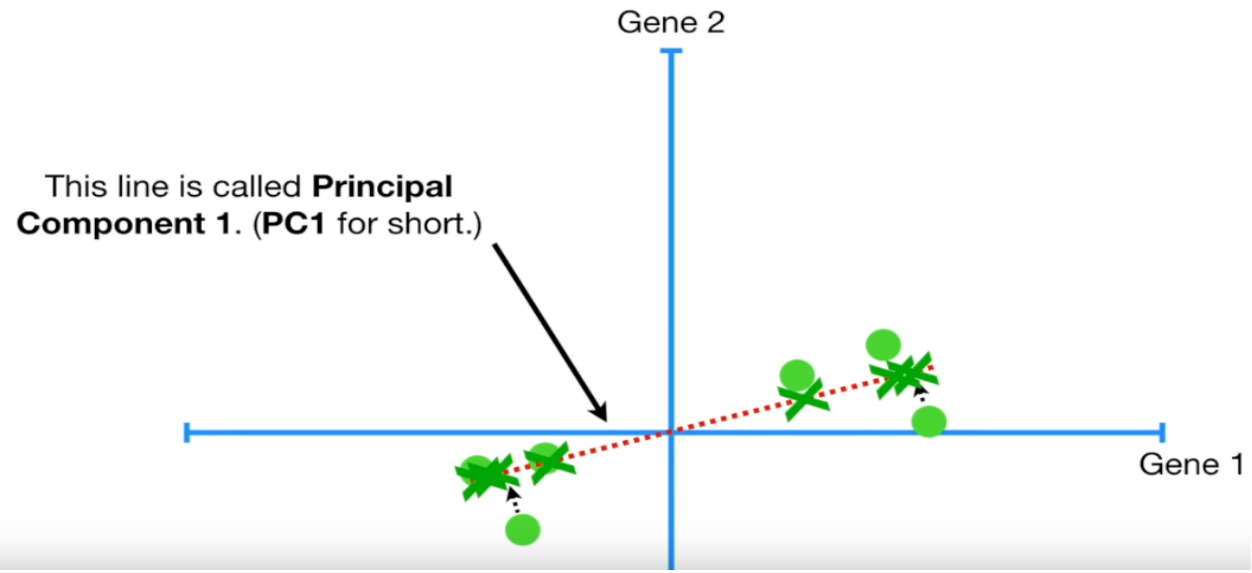
$$d_1^2 \quad d_2^2 \quad d_3^2 \quad d_4^2 \quad d_5^2 \quad d_6^2$$

Next, add all of them

$$d_1^2 + d_2^2 + d_3^2 + d_4^2 + d_5^2 + d_6^2$$

= Sum of Squared distances (SS)

How does PCA work?

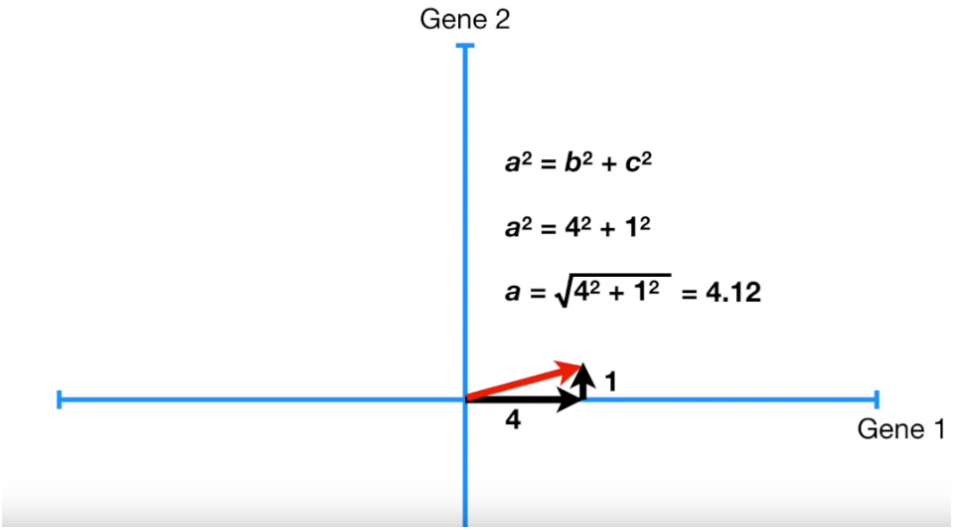


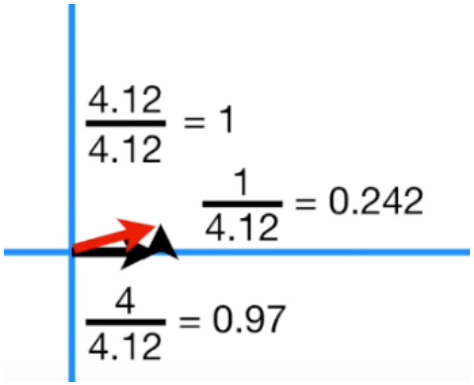
- Let, PC1 has a slope of 0.25.
- In other words, for every 4 units that we go out along the Gene 1 axis, we go up 1 unit along the Gene 2 axis.
- That means that the data are mostly spread out along the Gene 1 axis, and only a little bit spread out along the Gene 2 axis.

Terminology : Mathematicians call this as **linear combination** of Genes 1 and 2

How does PCA work?

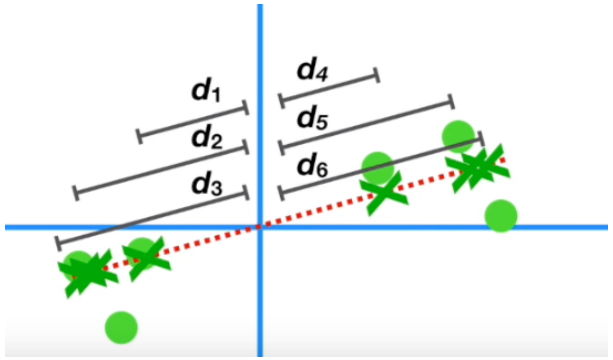
Wait…. We need to calculate more.. Let's calculate the unit vector



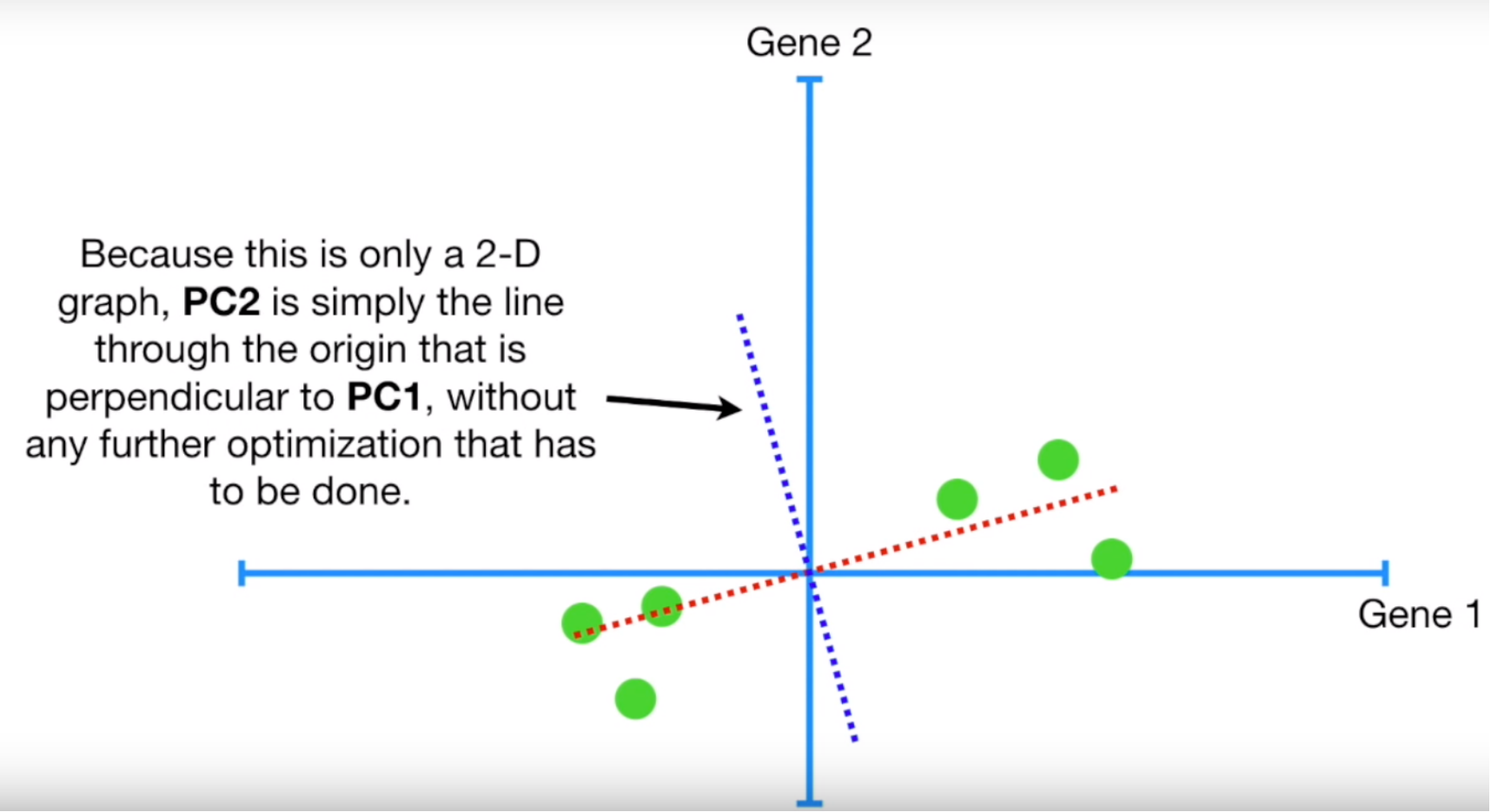


Terminology alert:

- This 1 unit long vector, consisting of 0.97 parts Gene 1 and 0.242 parts Gene2 is called the Singular Vector or the Eigenvector for PC1.
- The proportions of each gene are called Loading Scores.
- SS(distances for PC1) = Eigenvalue for PC1
- The square root of the Eigenvalue for PC1 is called the Singular Value for PC1.



How does PCA work?

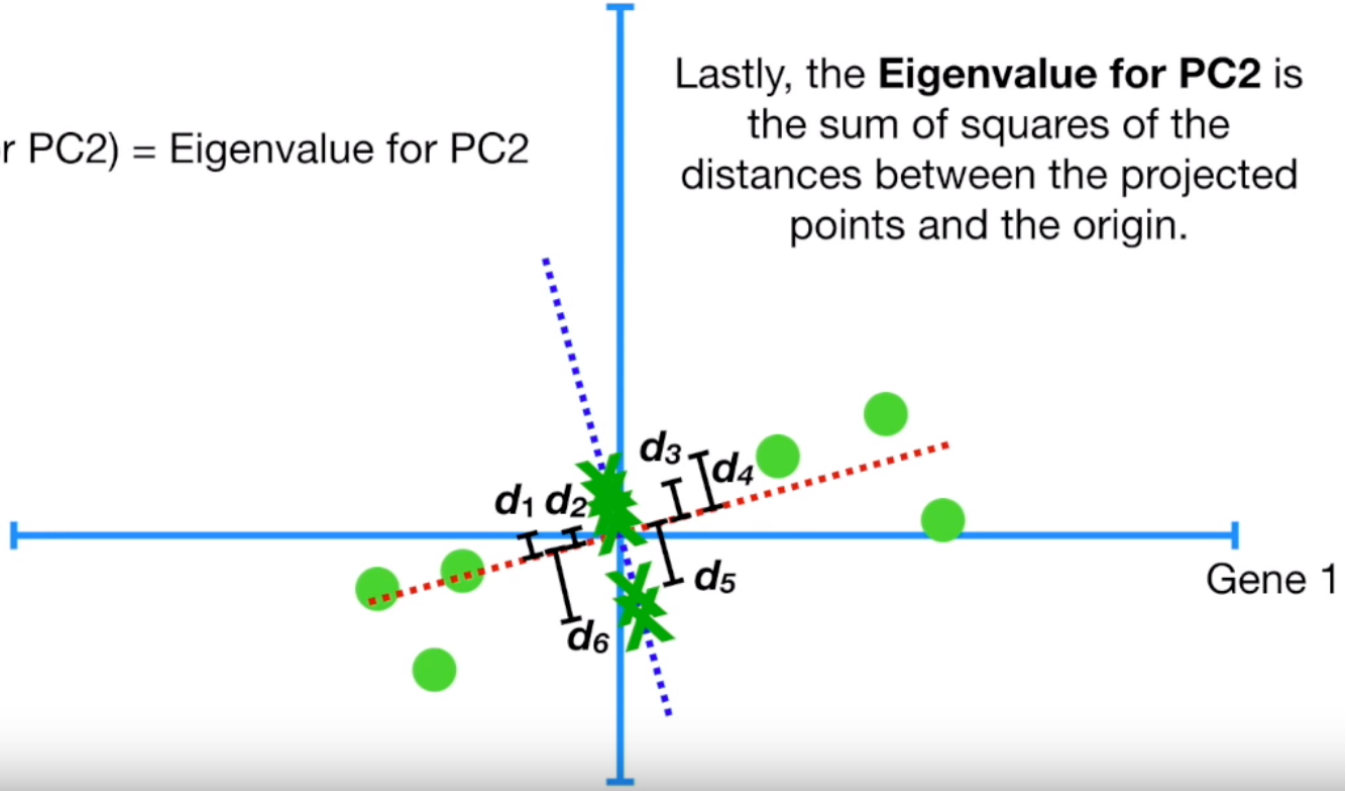


How does PCA work?

$d_1^2 + d_2^2 + d_3^2 + d_4^2 + d_5^2 + d_6^2$ = sum of squared distances = SS(distances)

SS(distances for PC2) = Eigenvalue for PC2

Lastly, the **Eigenvalue for PC2** is the sum of squares of the distances between the projected points and the origin.



How does PCA work?

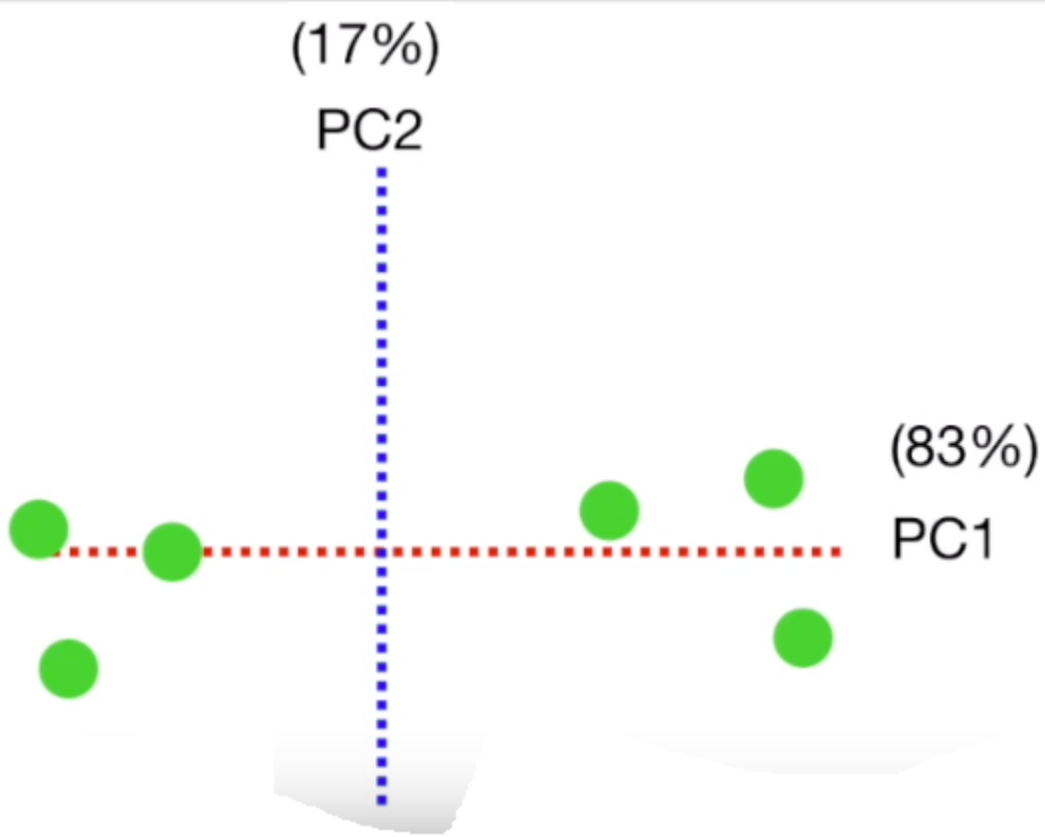
SS(distances for PC1) = Eigenvalue for PC1

SS(distances for PC2) = Eigenvalue for PC2

$$\frac{\text{SS(distances for PC1)}}{n - 1} = \text{Variation for PC1}$$

$$\frac{\text{SS(distances for PC2)}}{n - 1} = \text{Variation for PC2}$$

For the sake of the example, imagine that the Variation for **PC1 = 15**, and the variation for **PC2 = 3**.

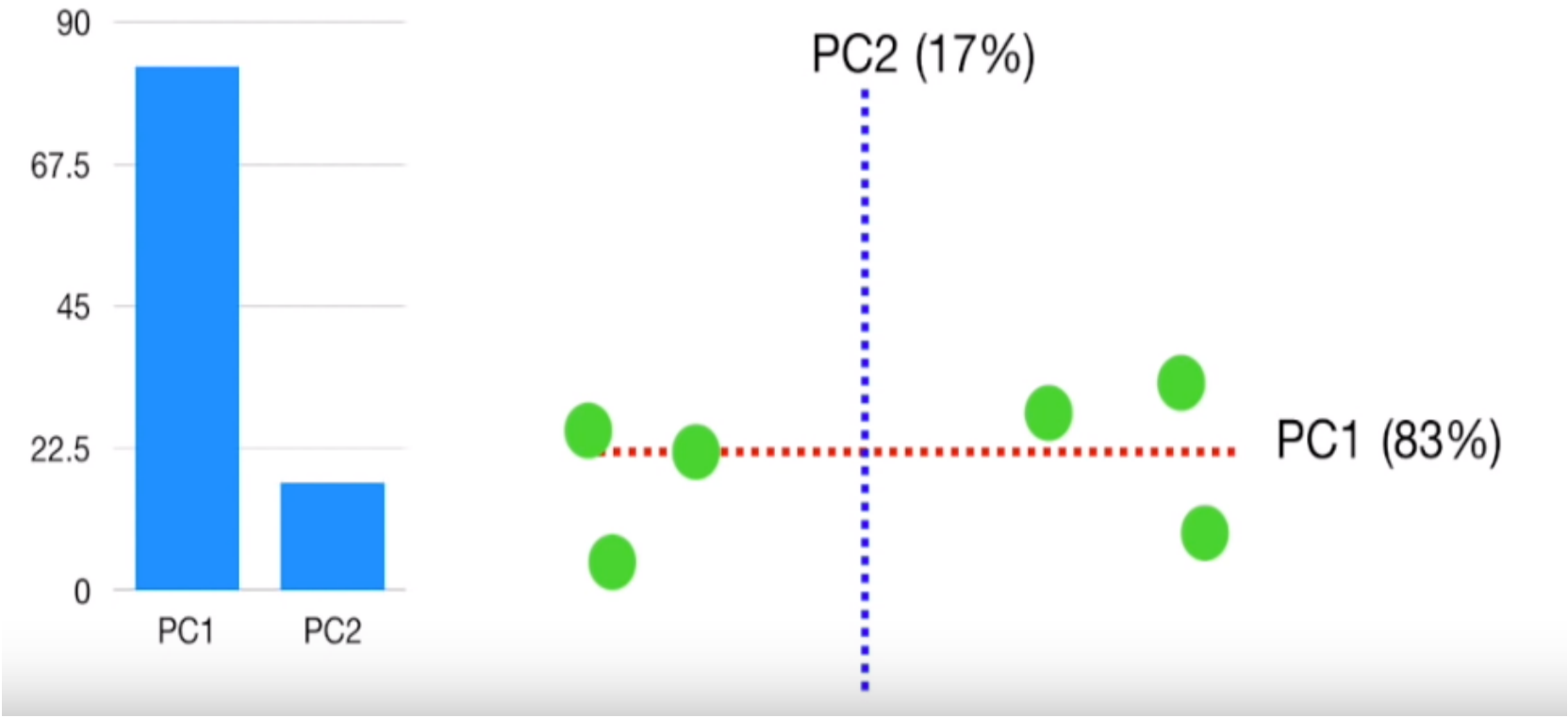


That means that the total variation around both PCs is **15 + 3 = 18**...

...and that means PC1 accounts for **15 / 18 = 0.83 = 83%** of the total variation around the PCs.

PC2 accounts for **3 / 18 = 0.17 = 17%** of the total variation around the PCs.

How does PCA work?



Terminology Alert: A Scree Plot is a graphical representation of the percentages of variation that each PC accounts for.

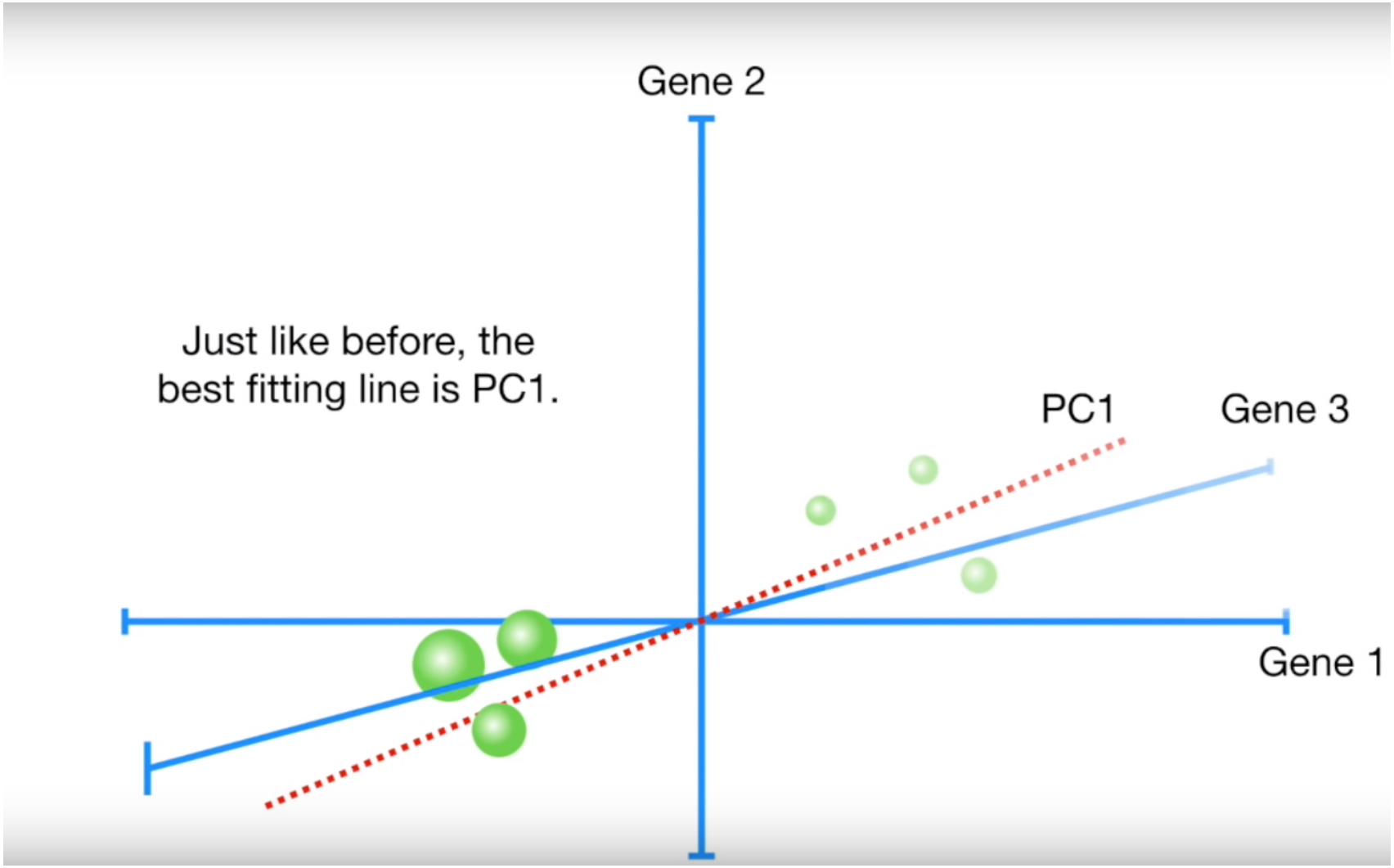
How does PCA work?

So now we are back to looking at…

- The data

Revisited

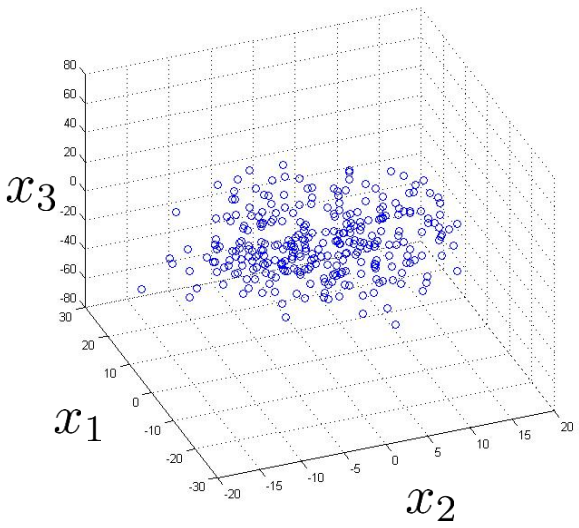
How does PCA work?

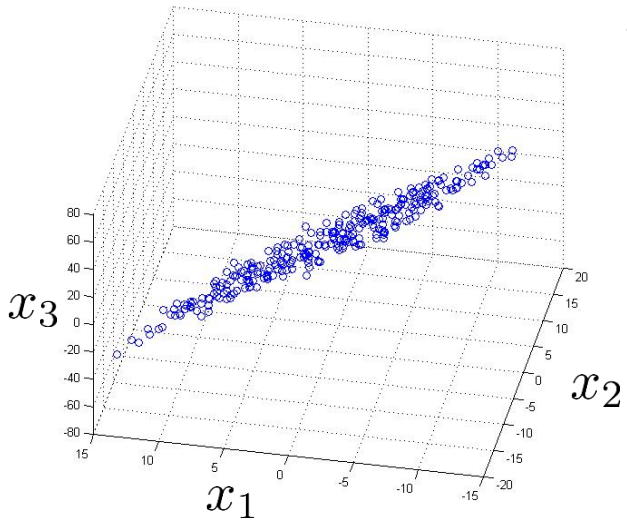


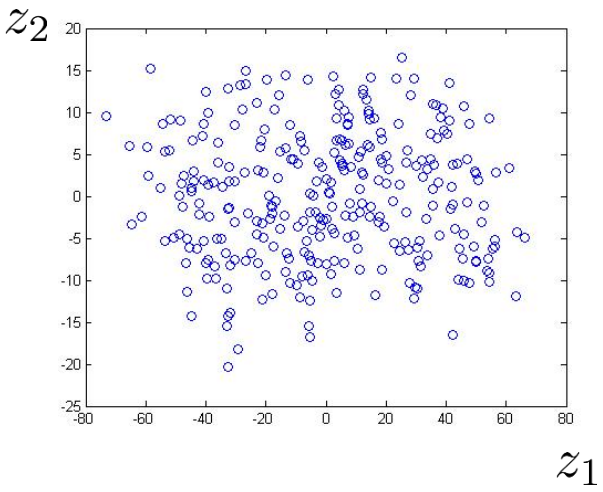
What about 3 Gene

Data Compression

Reduce data from 3D to 2D

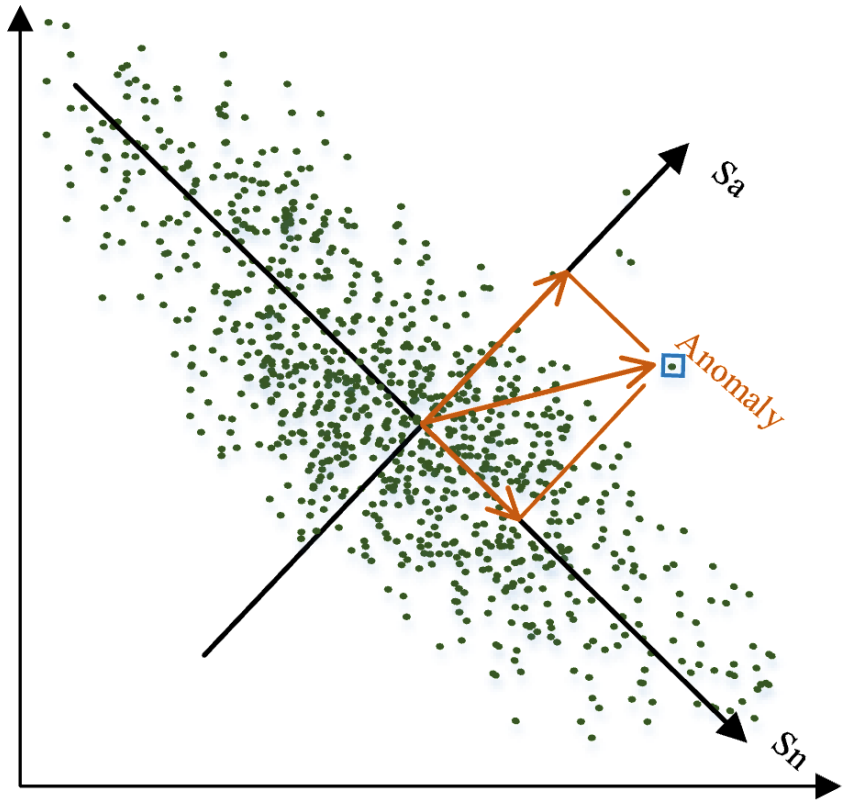






Andrew Ng

PCA for Anomaly Detection



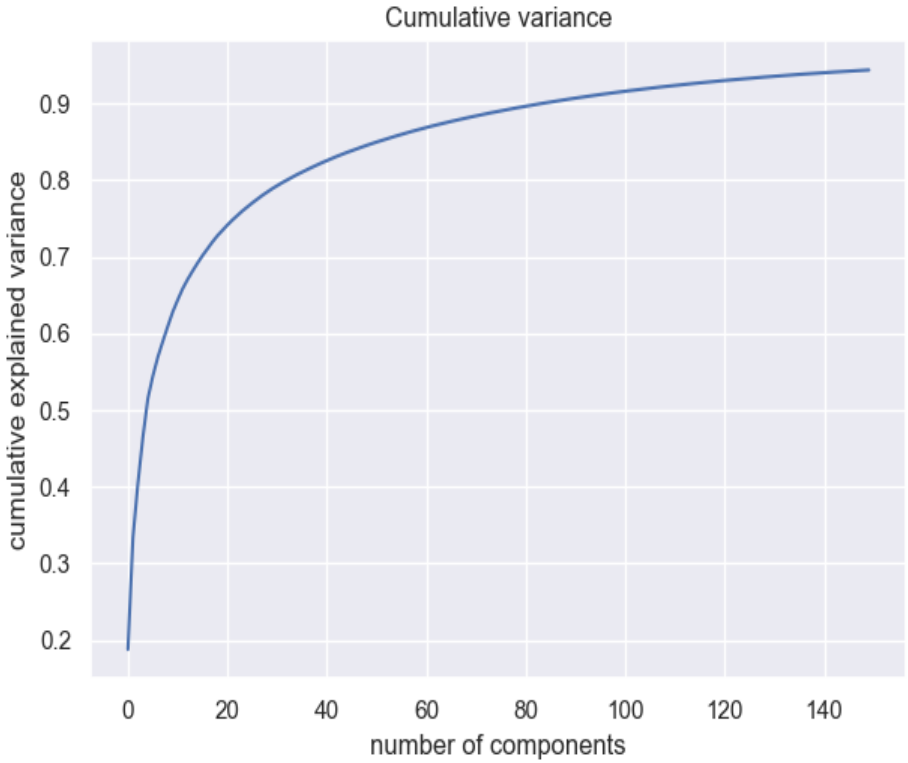
Two subspaces are generated by PCA:

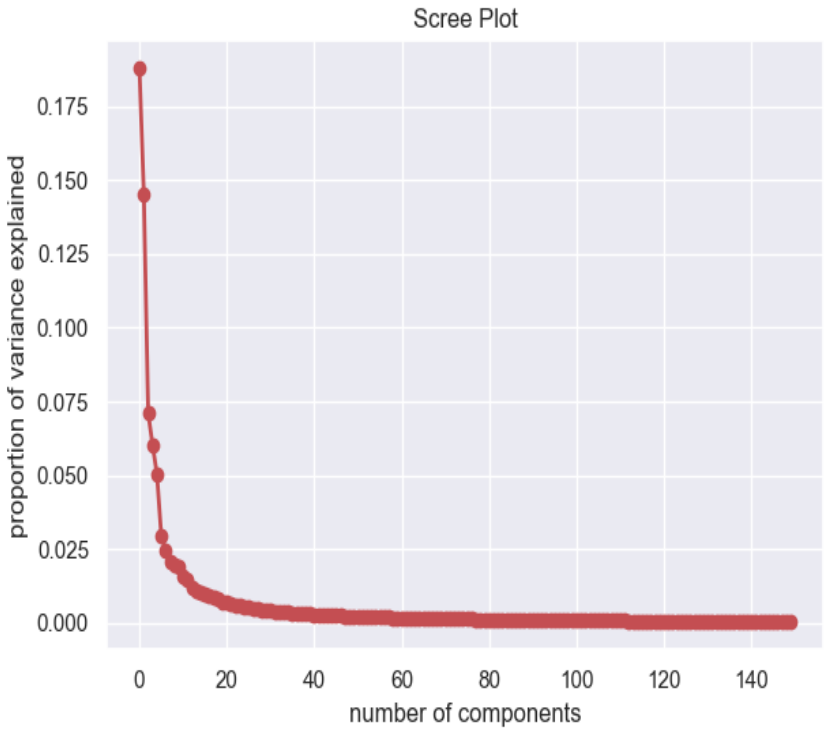
1. **S_n: Normal Space**, constructed by first *k* principal components.
2. **S_a: Anomaly Space**, constructed by remaining *(n-k)* components.

Project *y* into anomaly space

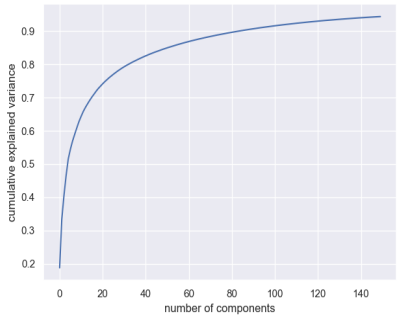
A data point is regarded as anomaly if the project value to anomaly space is higher than a threshold.

How many components?





How many components?



The space of all face images

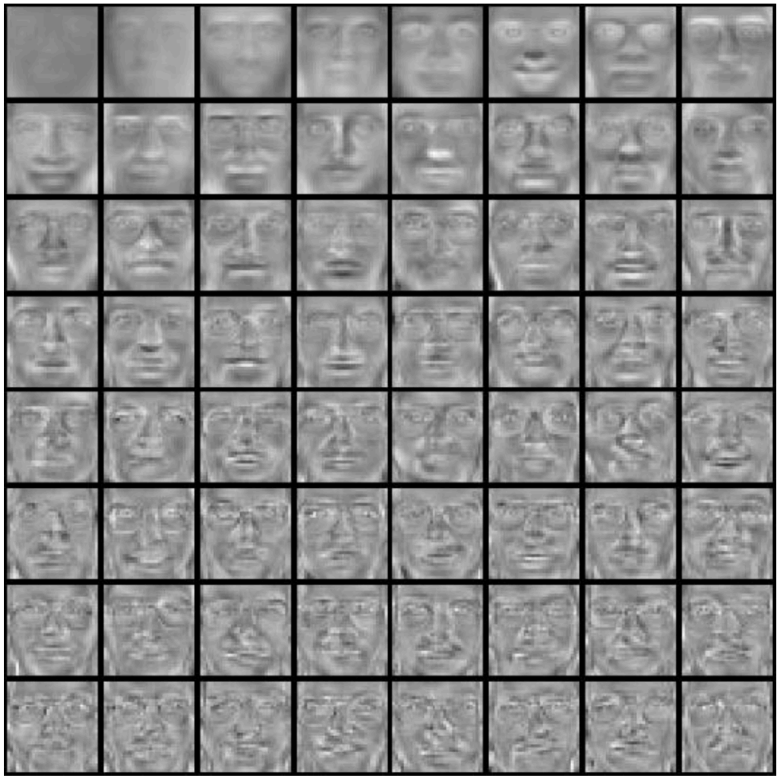
- When viewed as vectors of pixel values, face images are extremely high-dimensional
  - 100x100 image = 10,000 dimensions
  - Slow and lots of storage
- But very few 10,000-dimensional vectors are valid face images
- We want to effectively model the subspace of face images

slide by Derek Hoiem

Eigenfaces example

Top eigenvectors: $u_1,\ldots u_k$

Mean: μ



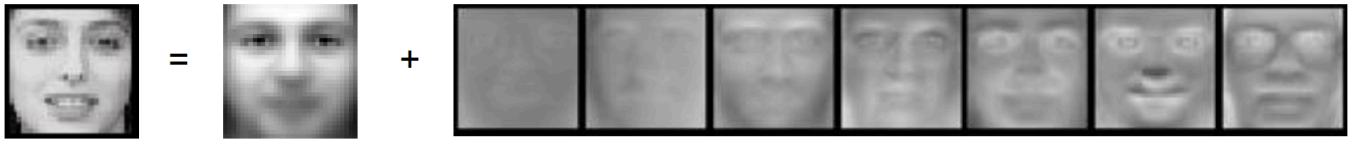
slide by Derek Hoiem

Representation and reconstruction

- Face **x** in "face space" coordinates:

$$\mathbf{x} \rightarrow \left[\mathbf{u}_1^{\mathrm{T}}(\mathbf{x}-\mu), \ldots, \mathbf{u}_k^{\mathrm{T}}(\mathbf{x}-\mu)\right]$$
$$= \quad w_1, \ldots, w_k$$

- Reconstruction:



$$\hat{x} = \mu + w_1 u_1 + w_2 u_2 + w_3 u_3 + w_4 u_4 + \ldots$$

slide by Derek Hoiem

Reconstruction

P = 4

P = 200

P = 400

After computing eigenfaces using 400 face images from ORL face database

slide by Derek Hoiem

Application: Image compression

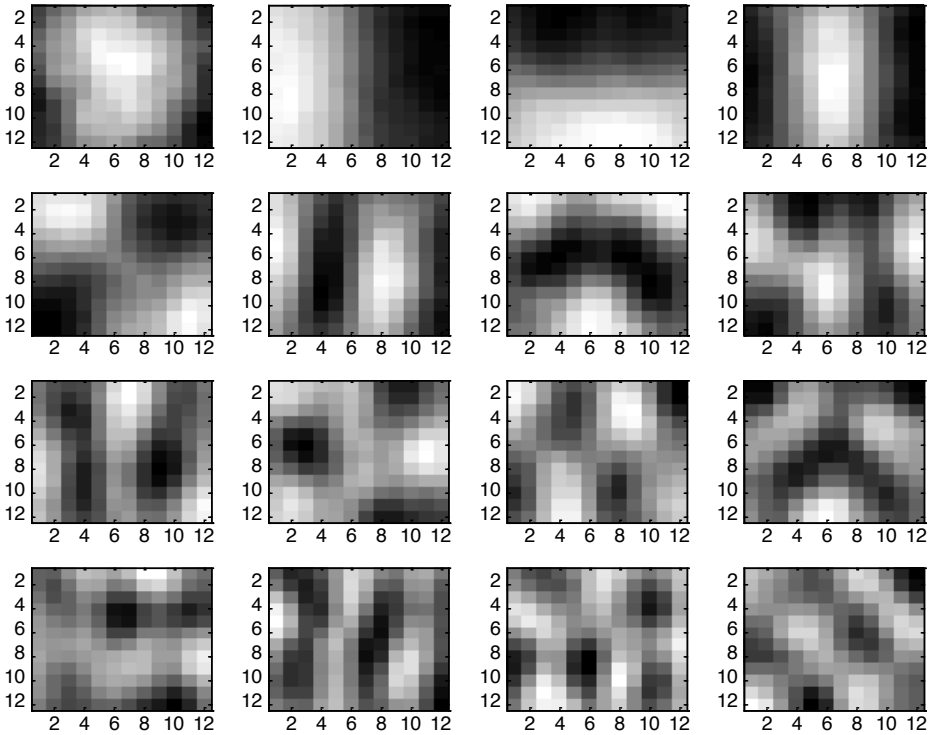
Original Image

- Divide the original 372x492 image into patches:
  - Each patch is an instance that contains 12x12 pixels on a grid
- View each as a 144-D vector

PCA compression: 144D → 60D

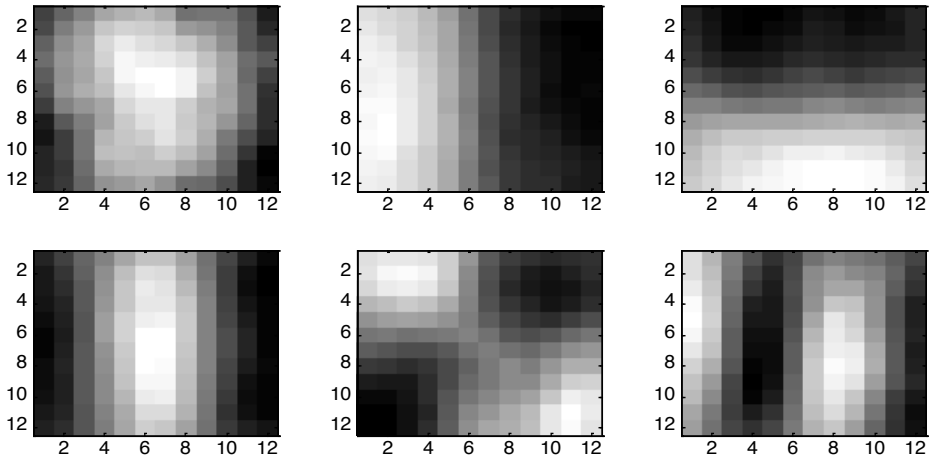
PCA compression: 144D → 16D

16 most important eigenvectors



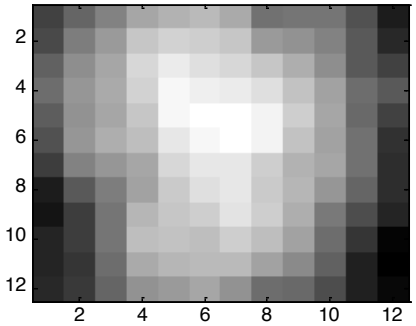
PCA compression: 144D -> 6D

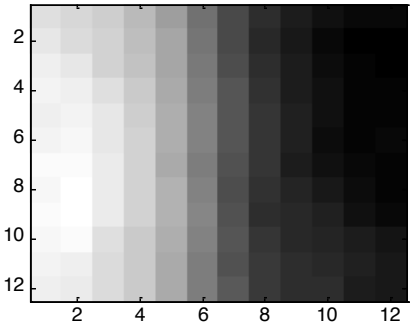
6 most important eigenvectors

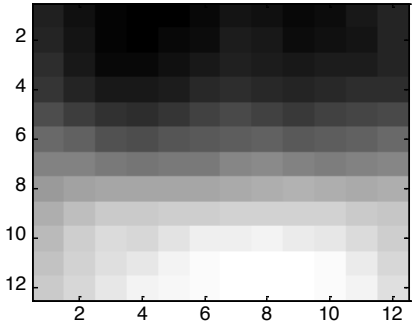


PCA compression: 144D → 3D

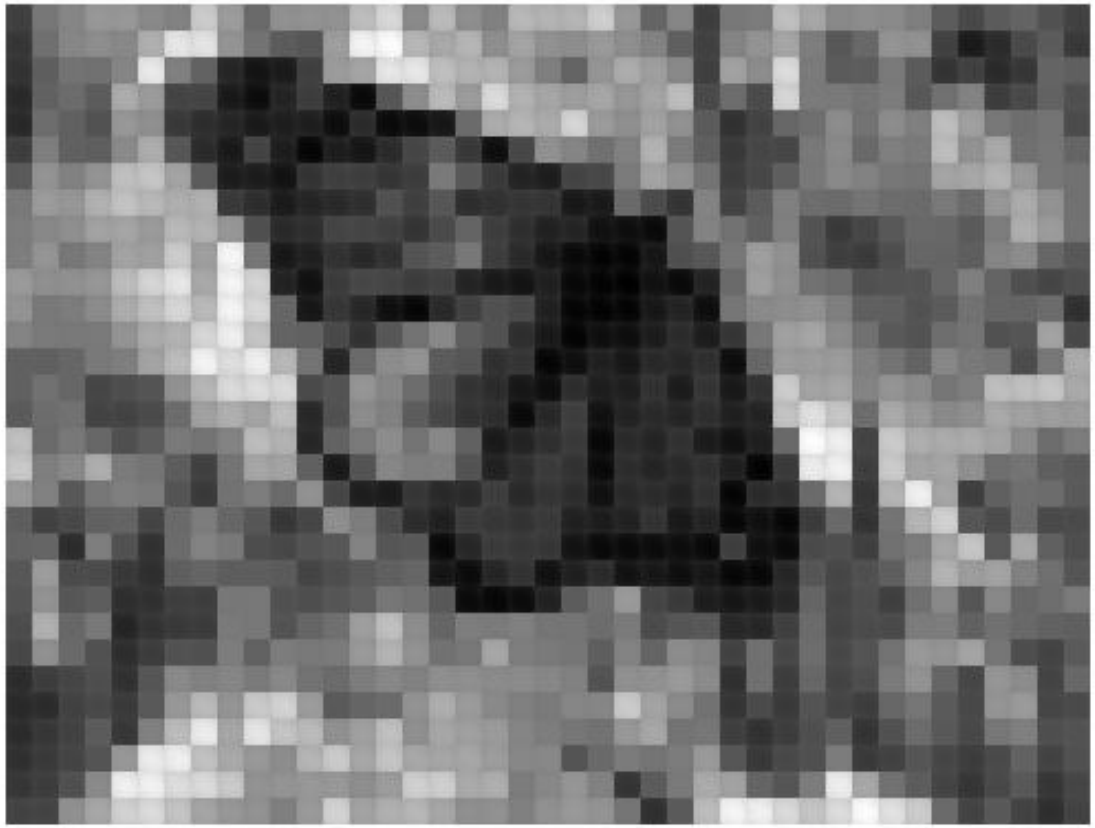
3 most important eigenvectors







PCA compression: 144D → 1D



Why PCA?

- Dimensionality reduction (feature extraction and engineering)

Other:
- a tool for visualization
- for noise filtering
- anomaly detection

Limitation of PCA

- highly affected by outliers in the data

References

1. https://www.youtube.com/watch?v=FgakZw6K1QQ

2. https://jakevdp.github.io/PythonDataScienceHandbook/05.09-principal-component-analysis.html#Introducing-Principal-Component-Analysis

3. https://towardsdatascience.com/a-one-stop-shop-for-principal-component-analysis-5582fb7e0a9c

Thank You