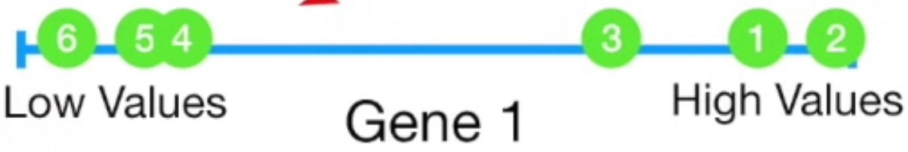
PCA – PRINCIPLE COMPONENT ANALYSIS

	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

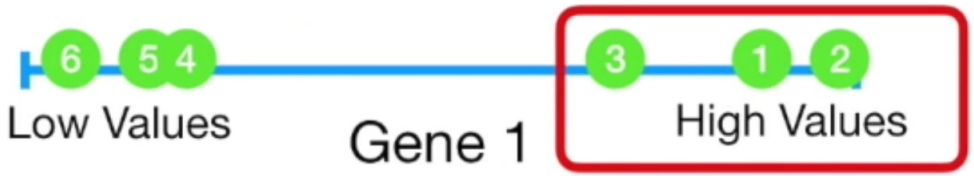
	Mouse 1	Mouse 2	Mouse 3	Mouse 4	Mouse 5	Mouse 6
Gene 1	10	11	8	3	2	1

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



	Mouse 1	Mouse 2	Mouse 3	Mouse 4	Mouse 5	Mouse 6
Gene 1	10	11	8	3	2	1

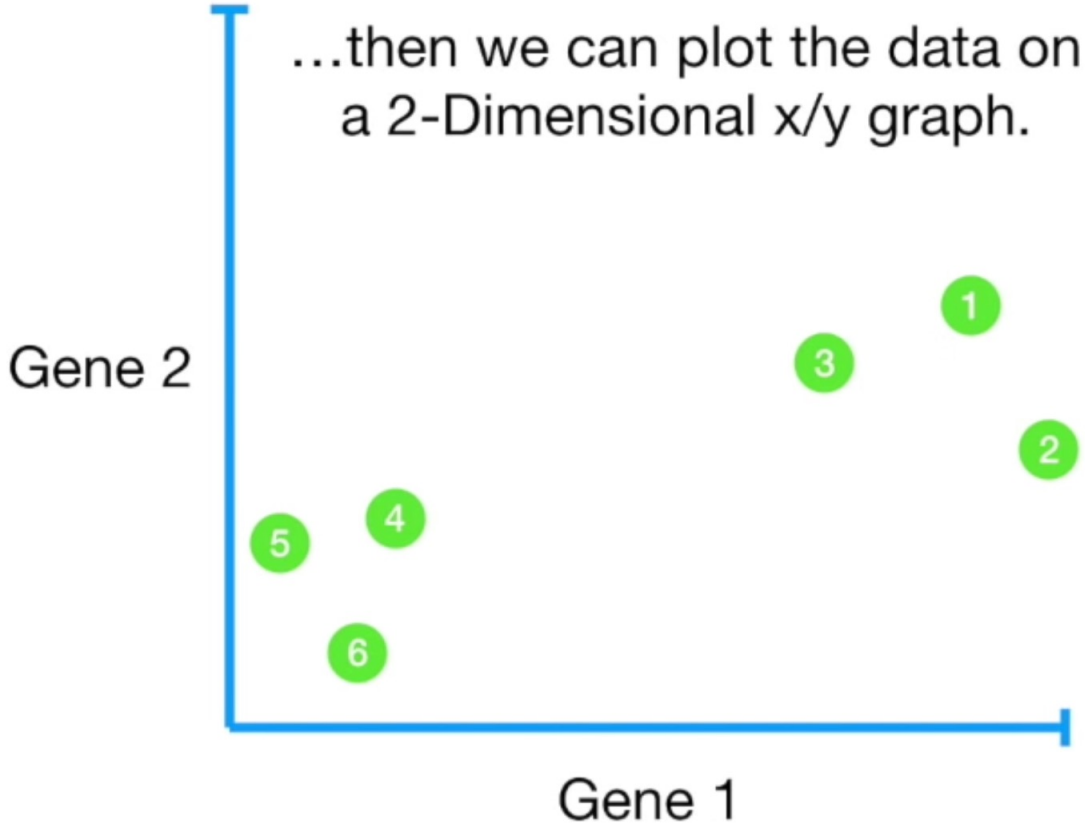
Mice 1, 2 and 3 have relatively high values…



	Mouse 1	Mouse 2	Mouse 3	Mouse 4	Mouse 5	Mouse 6
Gene 1	10	11	8	3	2	1

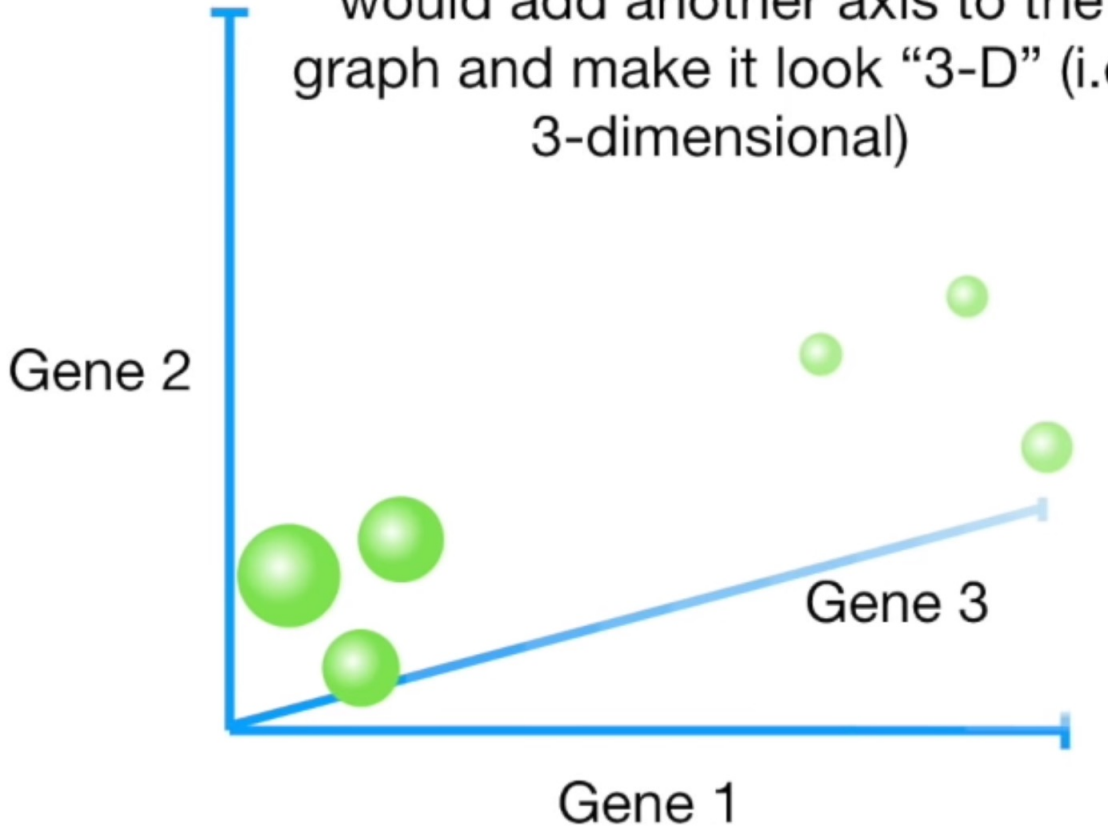
…and mice 4, 5 and 6 have relatively low values.

	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



	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

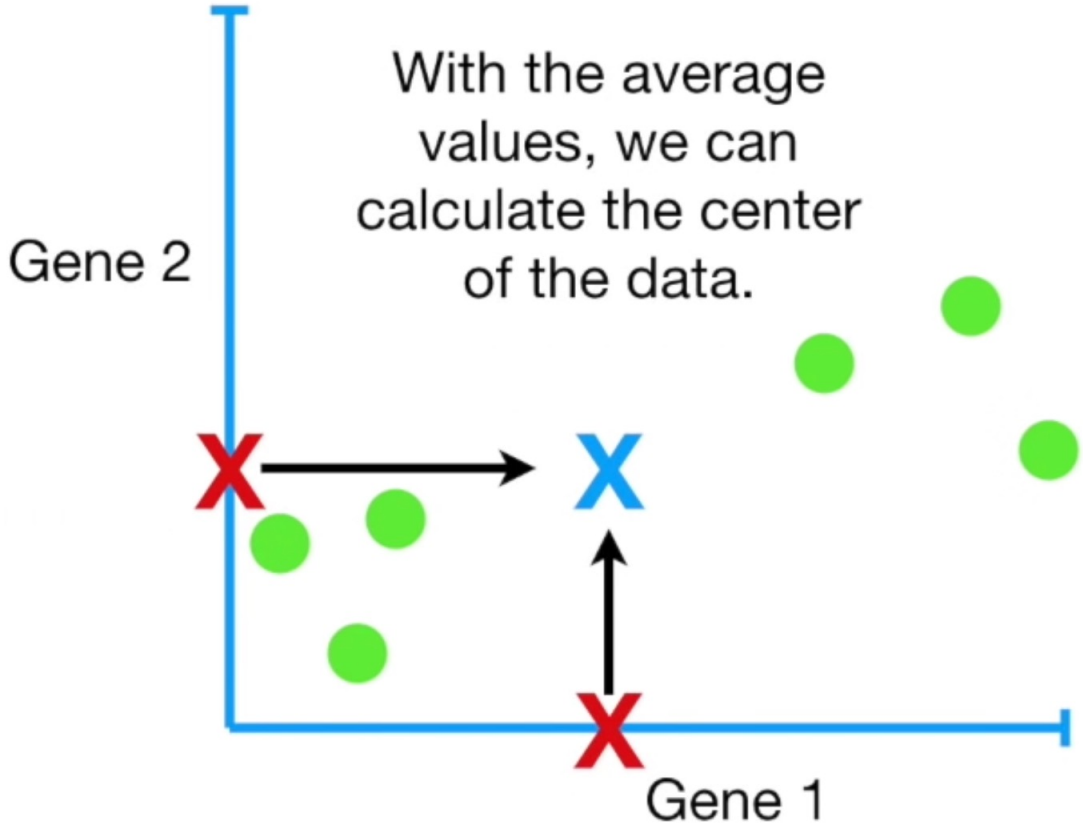
If we measured 3 genes, we would add another axis to the graph and make it look “3-D” (i.e. 3-dimensional)



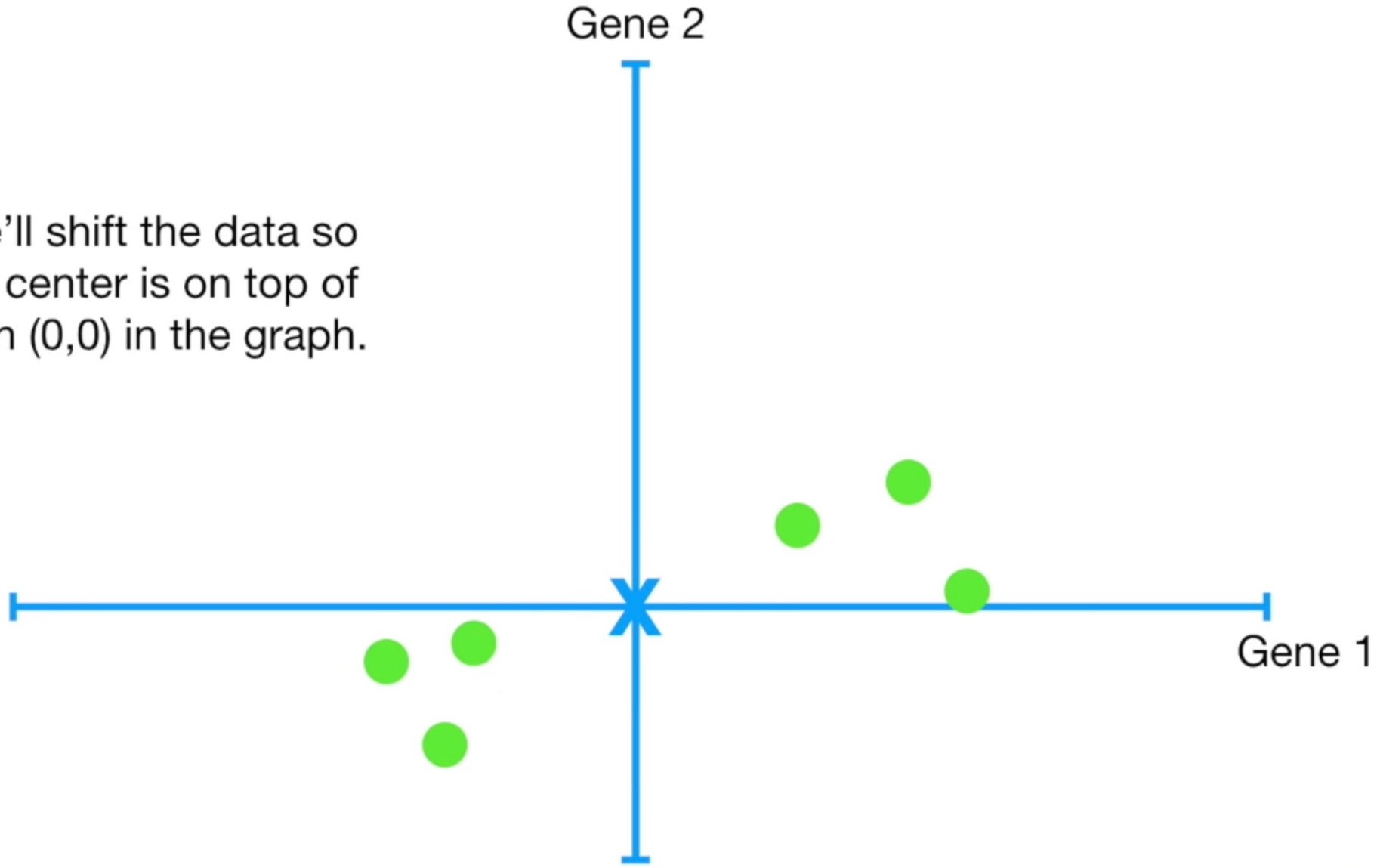
	Mouse 1	Mouse 2	Mouse 3	Mouse 4	Mouse 5	Mouse 6
Gene 1	10	11	8	3	2	1
Gene 2	6	4	5	3	2.8	1

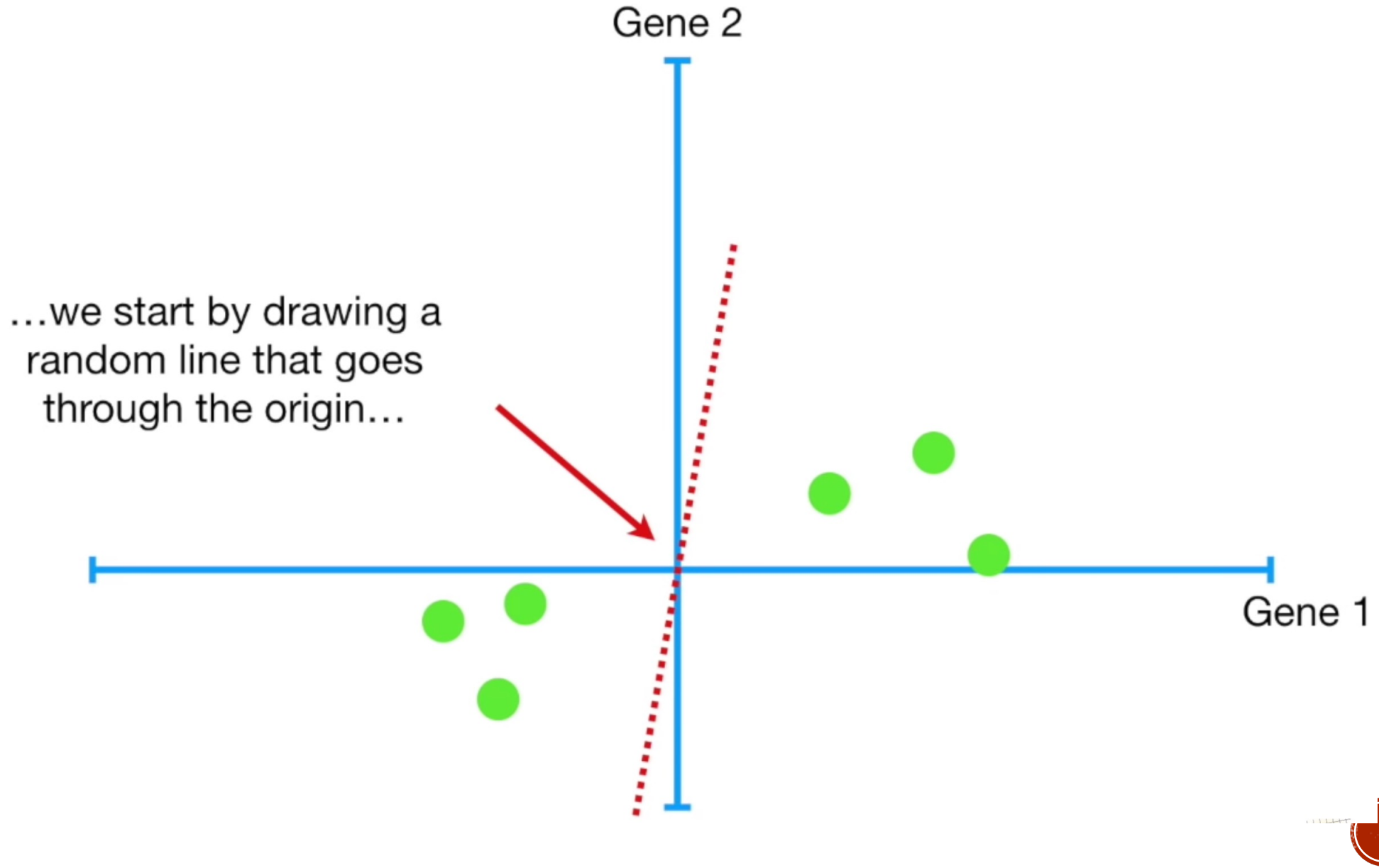
To understand what PCA does and how it works, let's go back to the dataset that only had 2 genes…

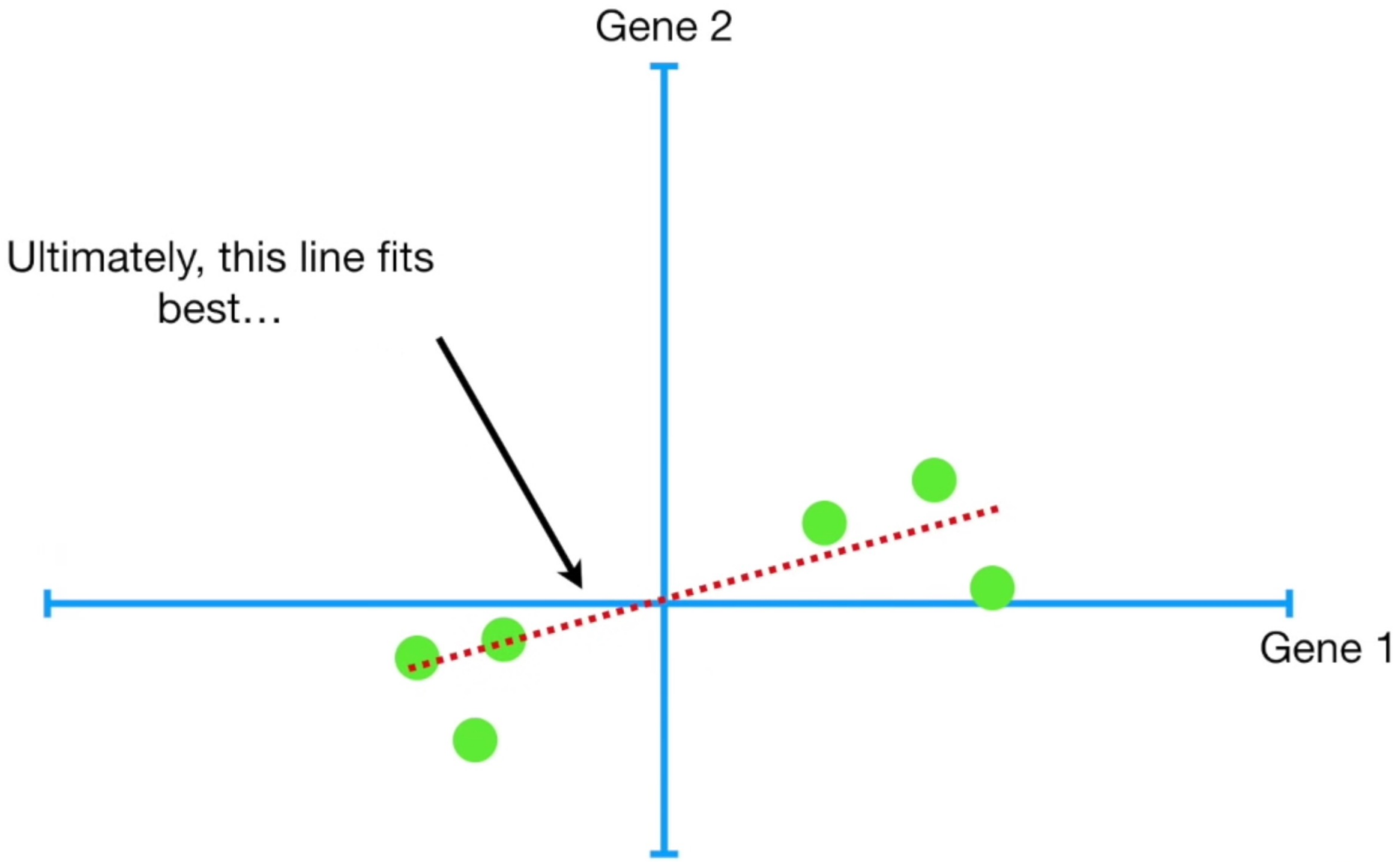
	Mouse 1	Mouse 2	Mouse 3	Mouse 4	Mouse 5	Mouse 6
Gene 1	10	11	8	3	2	1
Gene 2	6	4	5	3	2.8	1



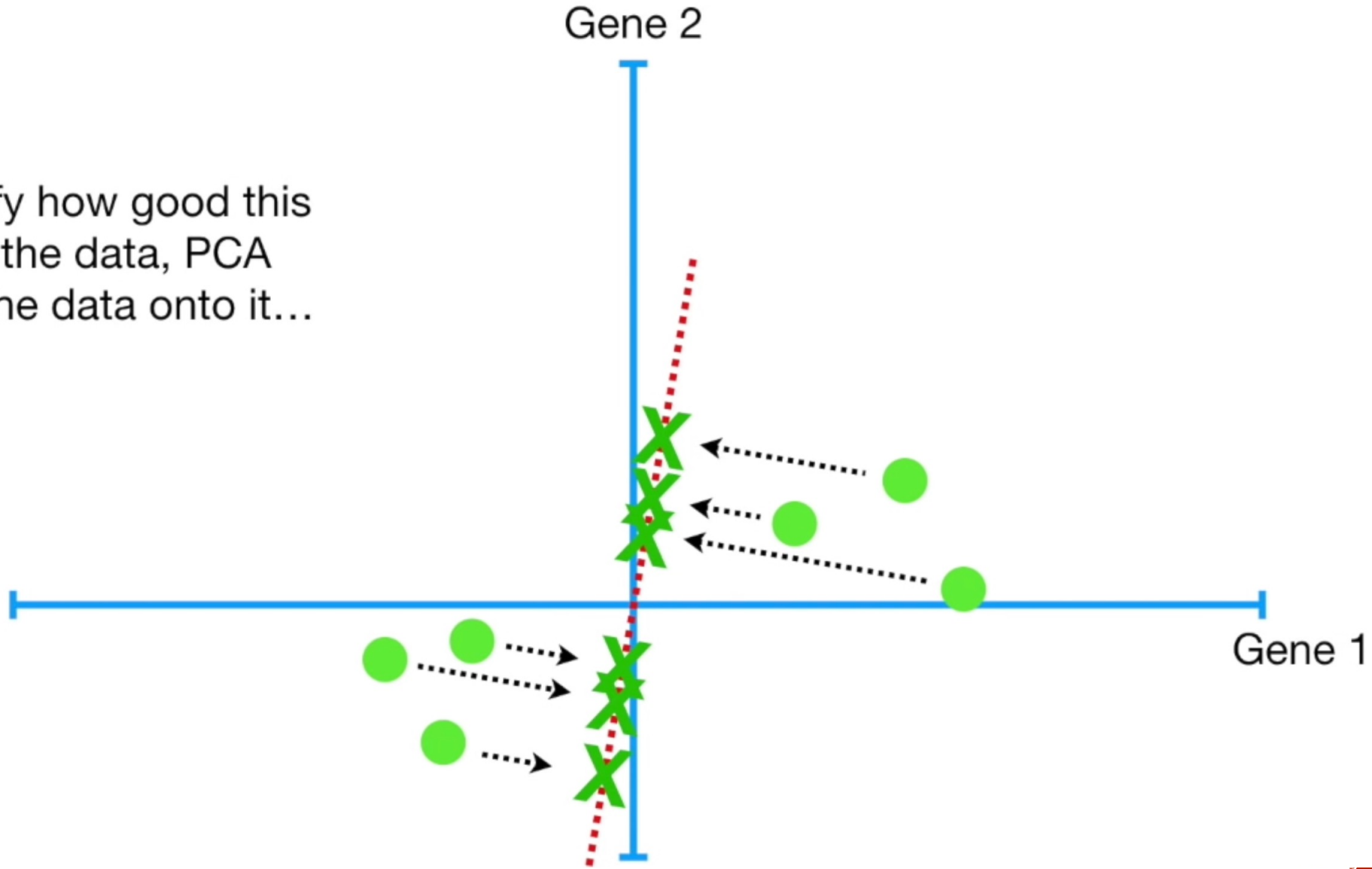
Now we'll shift the data so that the center is on top of the origin (0,0) in the graph.

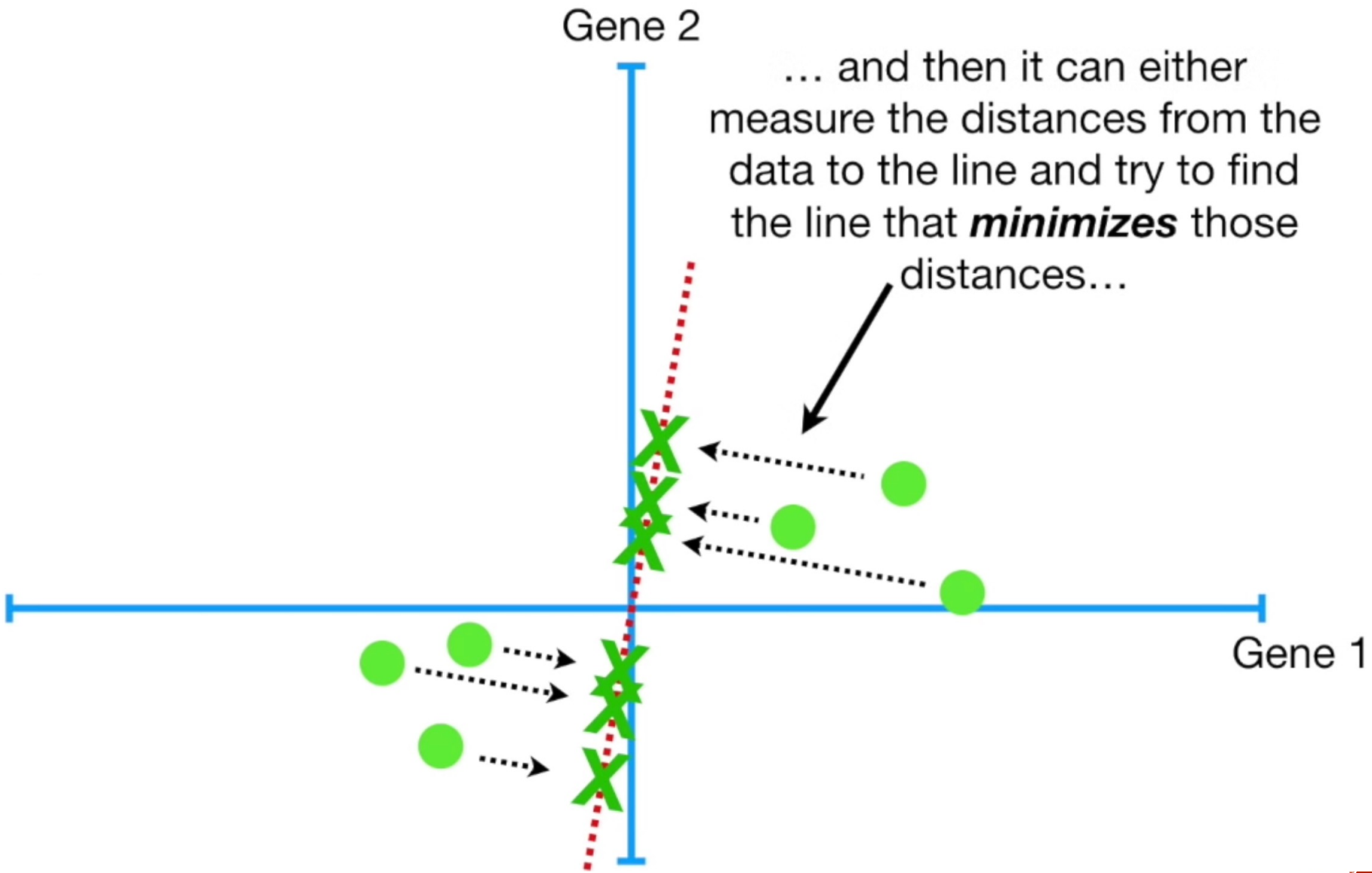



Gene 2
…we start by drawing a random line that goes through the origin…
Gene 1

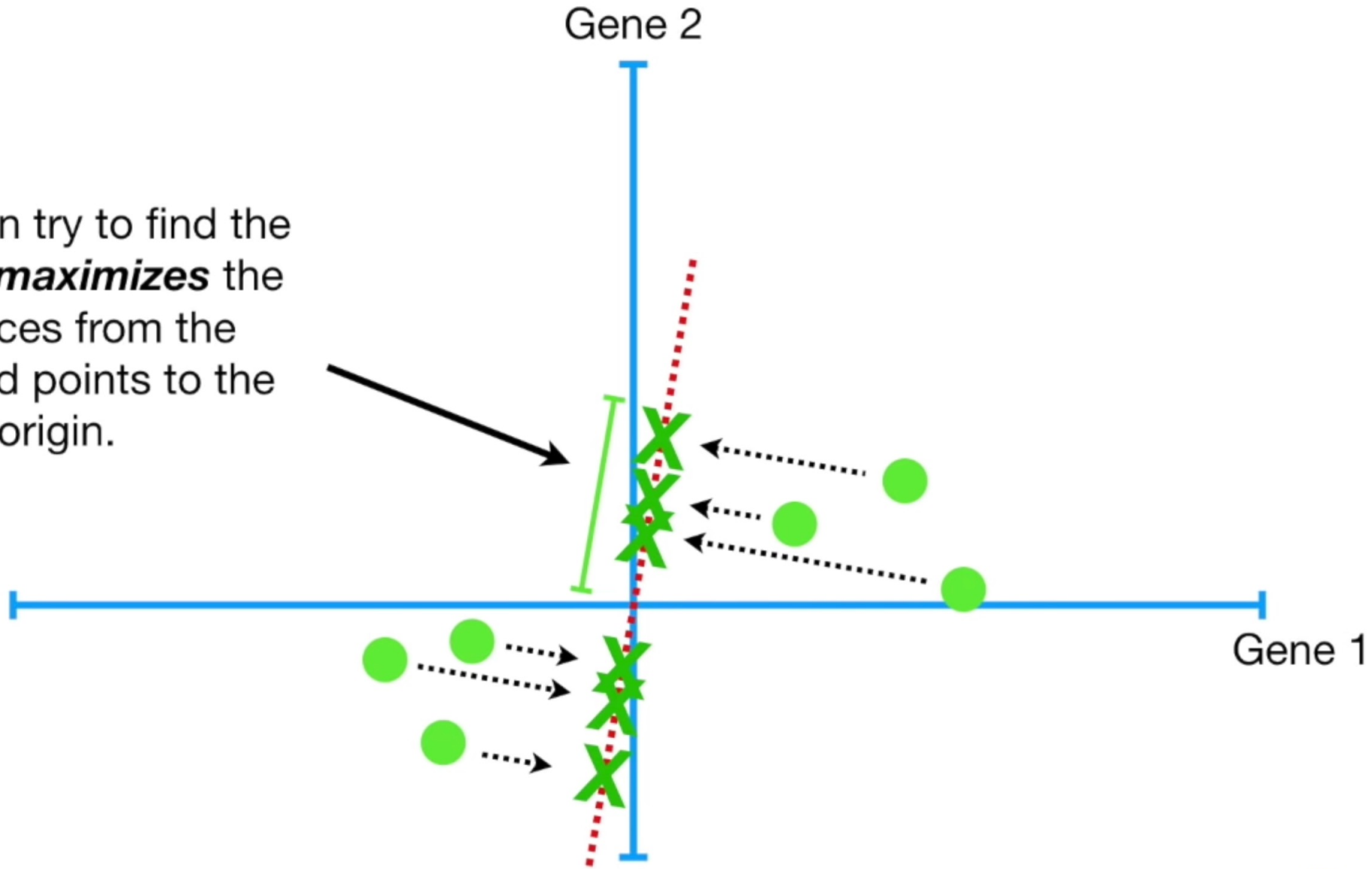

Gene 2
Ultimately, this line fits best…
Gene 1

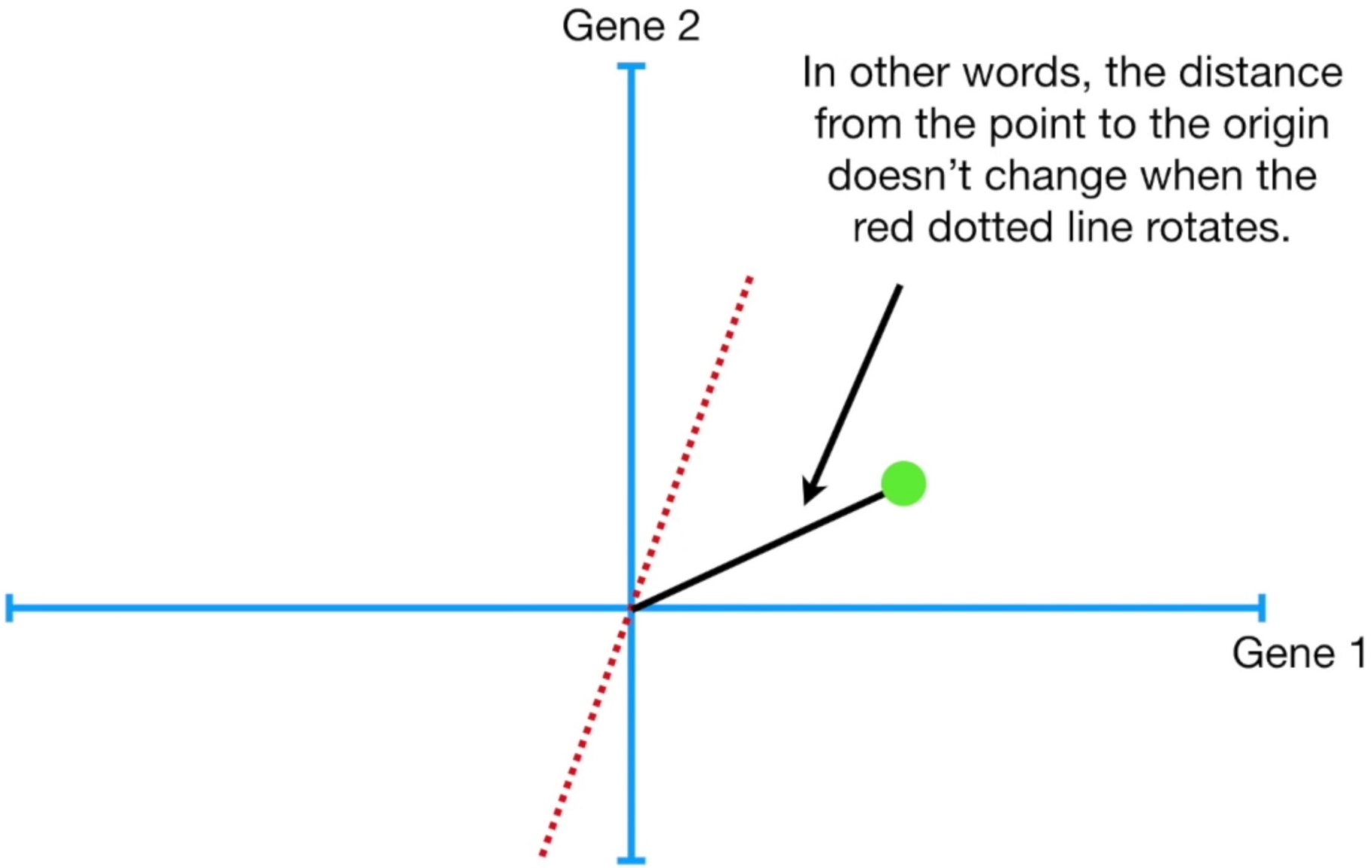
To quantify how good this line fits the data, PCA projects the data onto it…

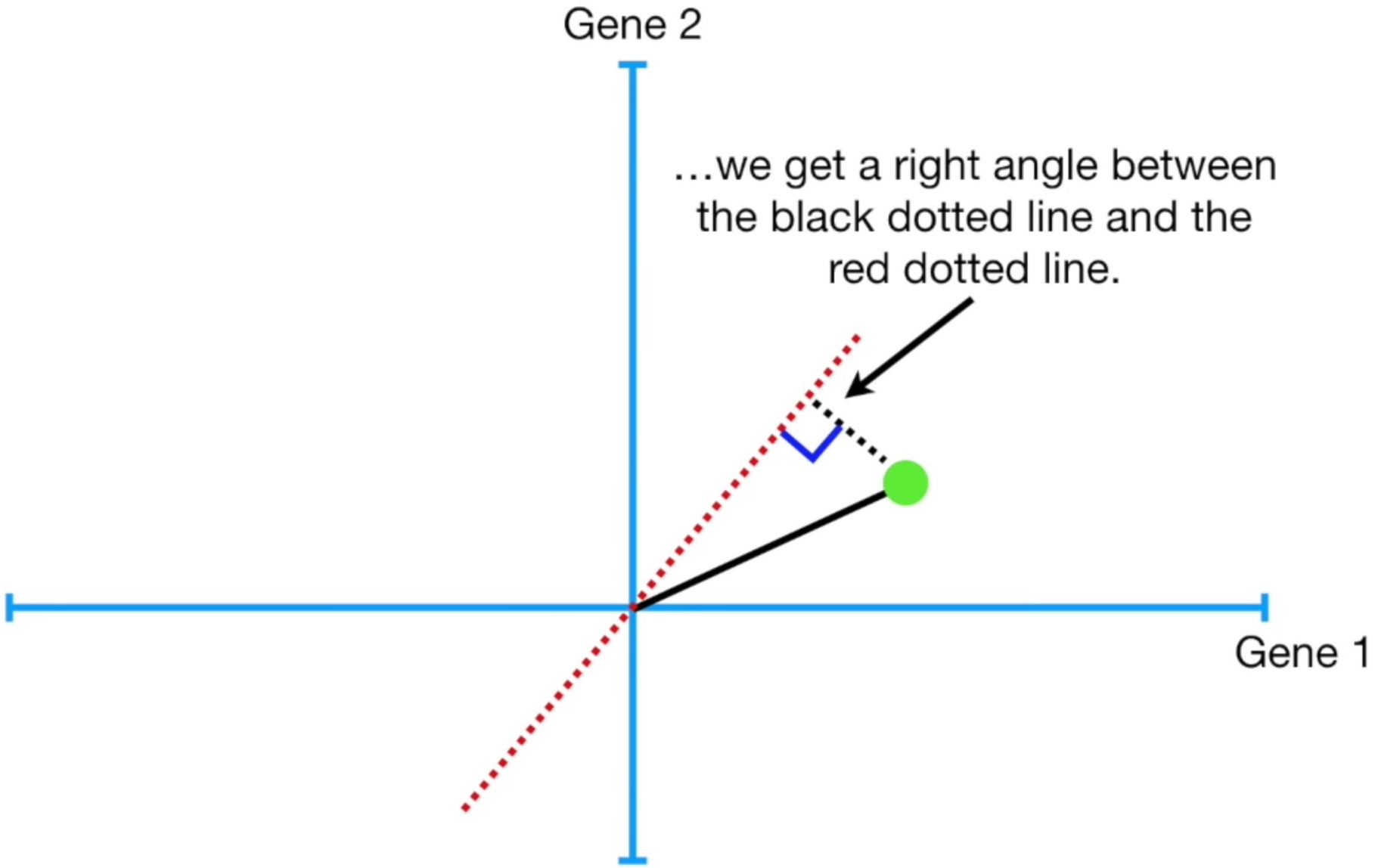



Gene 2
… and then it can either measure the distances from the data to the line and try to find the line that ***minimizes*** those distances…
Gene 1

…or it can try to find the line that ***maximizes*** the distances from the projected points to the origin.



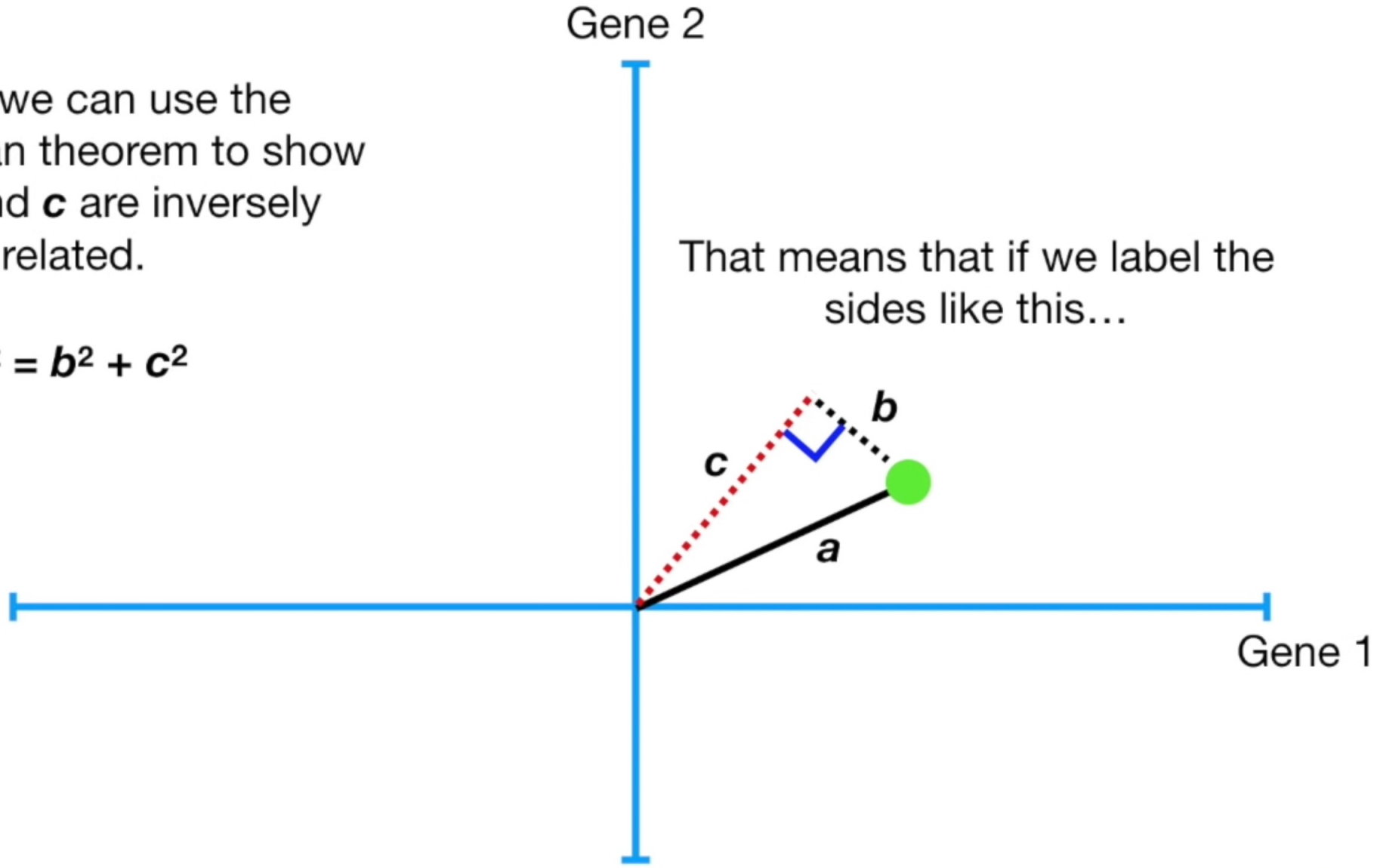

Gene 2
In other words, the distance from the point to the origin doesn't change when the red dotted line rotates.
Gene 1

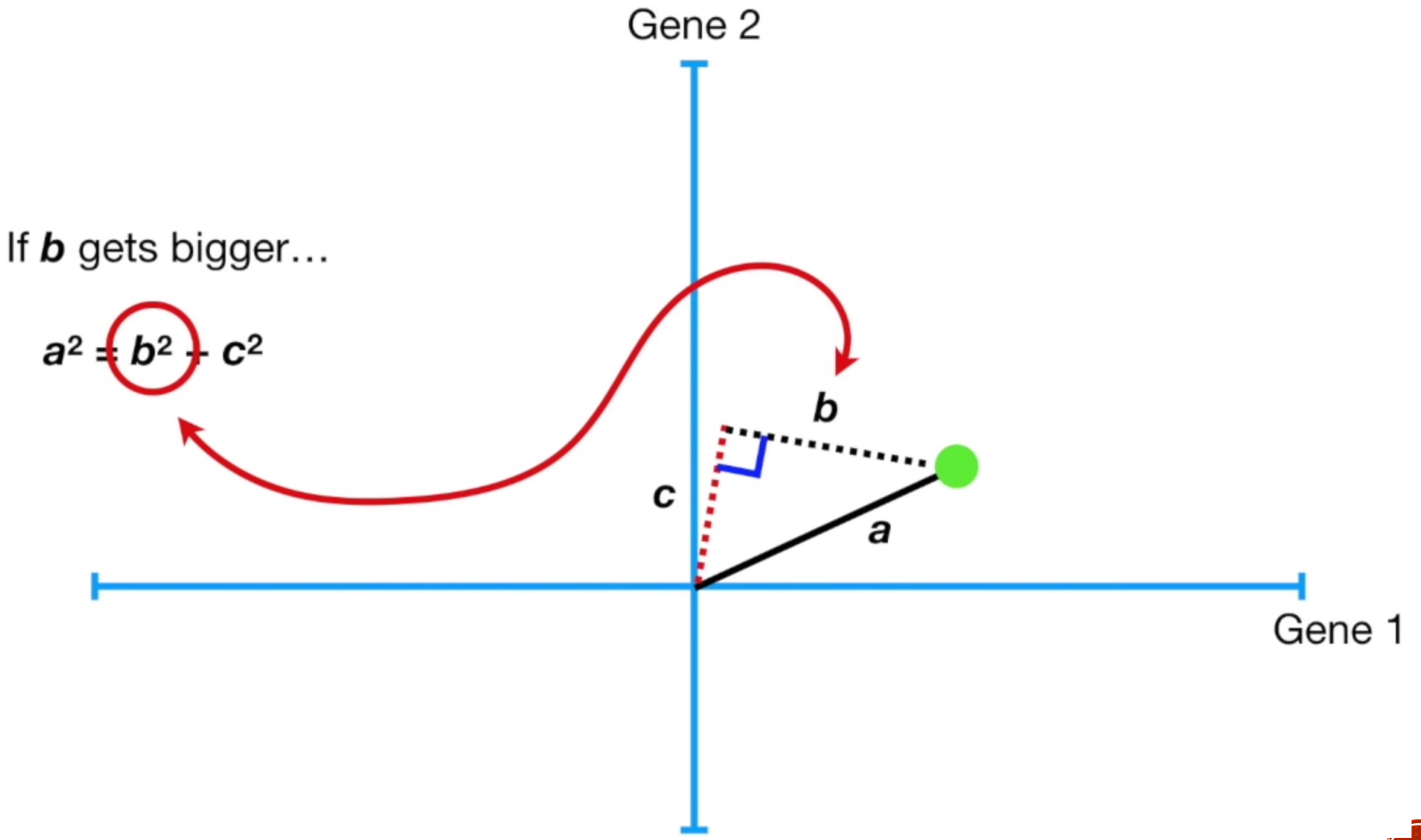

Gene 2
…we get a right angle between the black dotted line and the red dotted line.
Gene 1

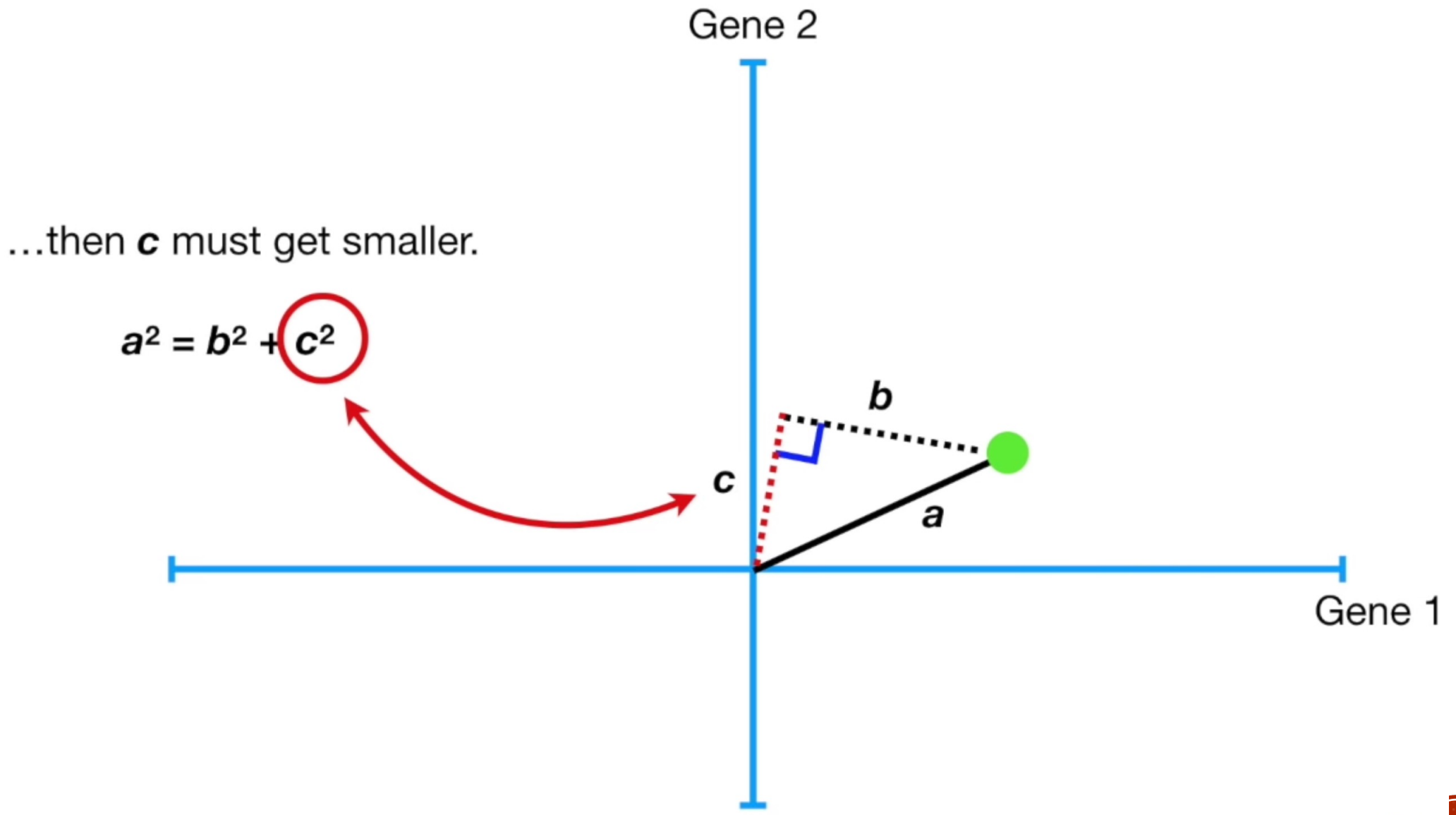
…then we can use the Pythagorean theorem to show how ***b*** and ***c*** are inversely related.

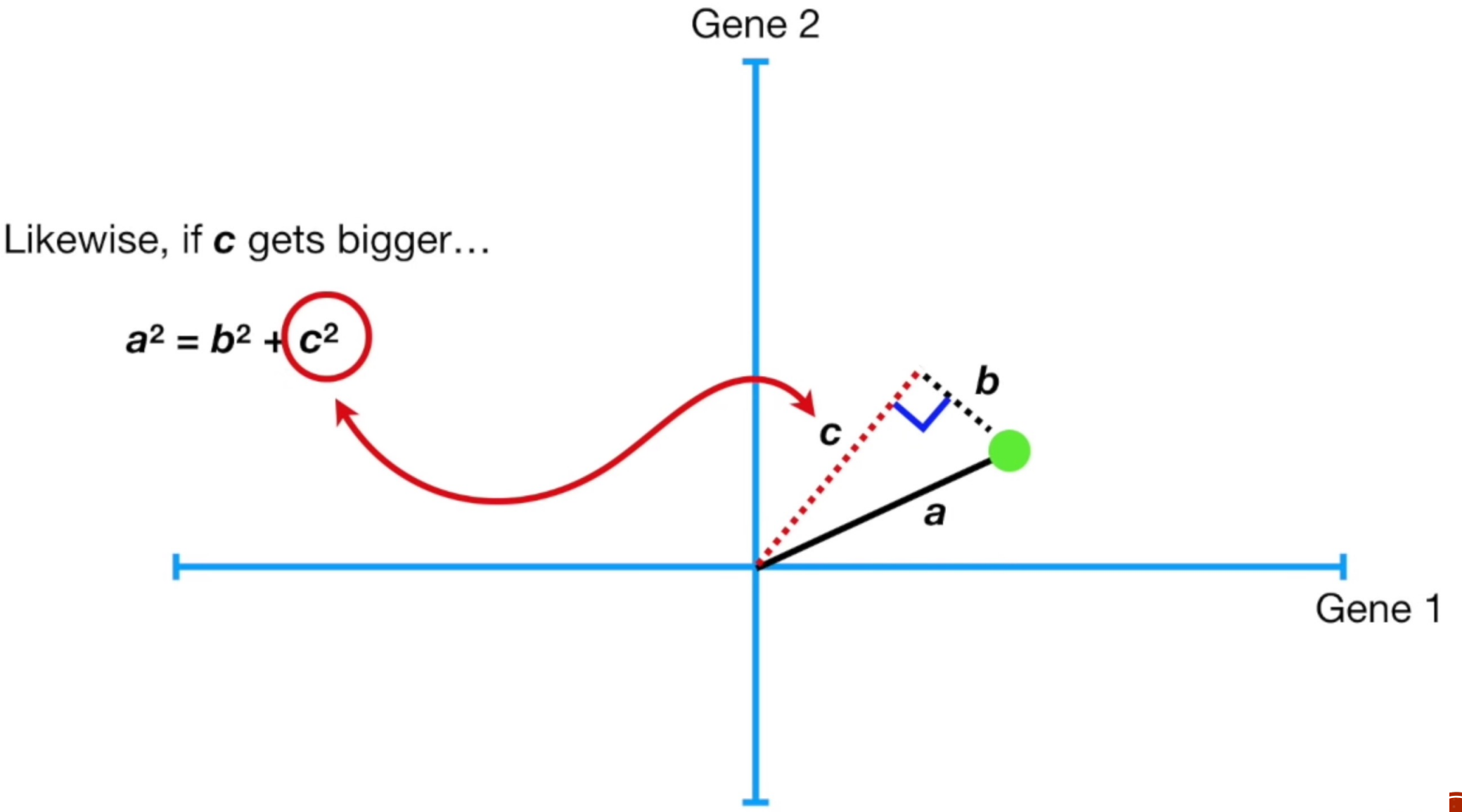
$$a^2 = b^2 + c^2$$

That means that if we label the sides like this…



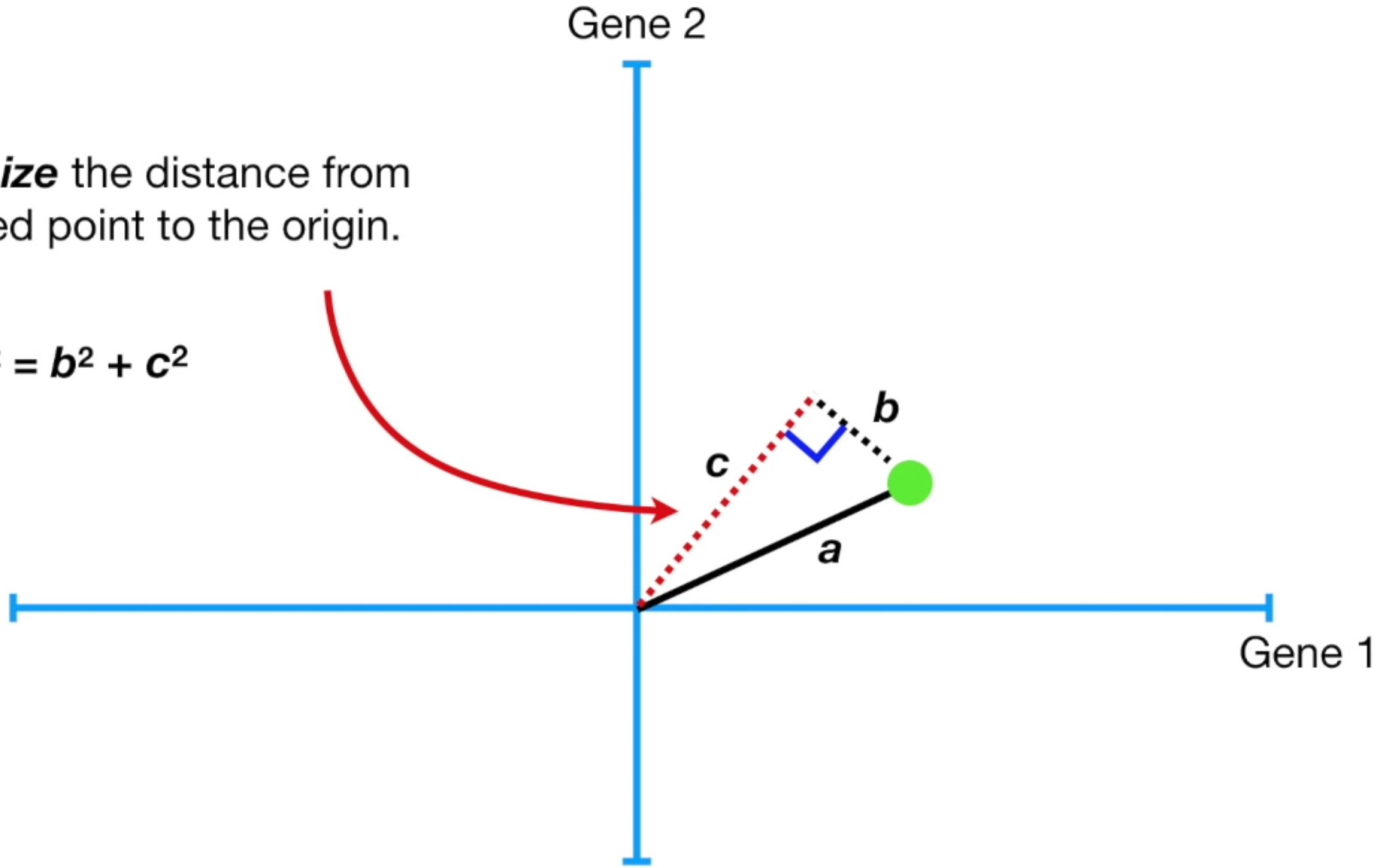

Gene 2
If b gets bigger…
$a^2 = b^2 + c^2$
b
c
a
Gene 1

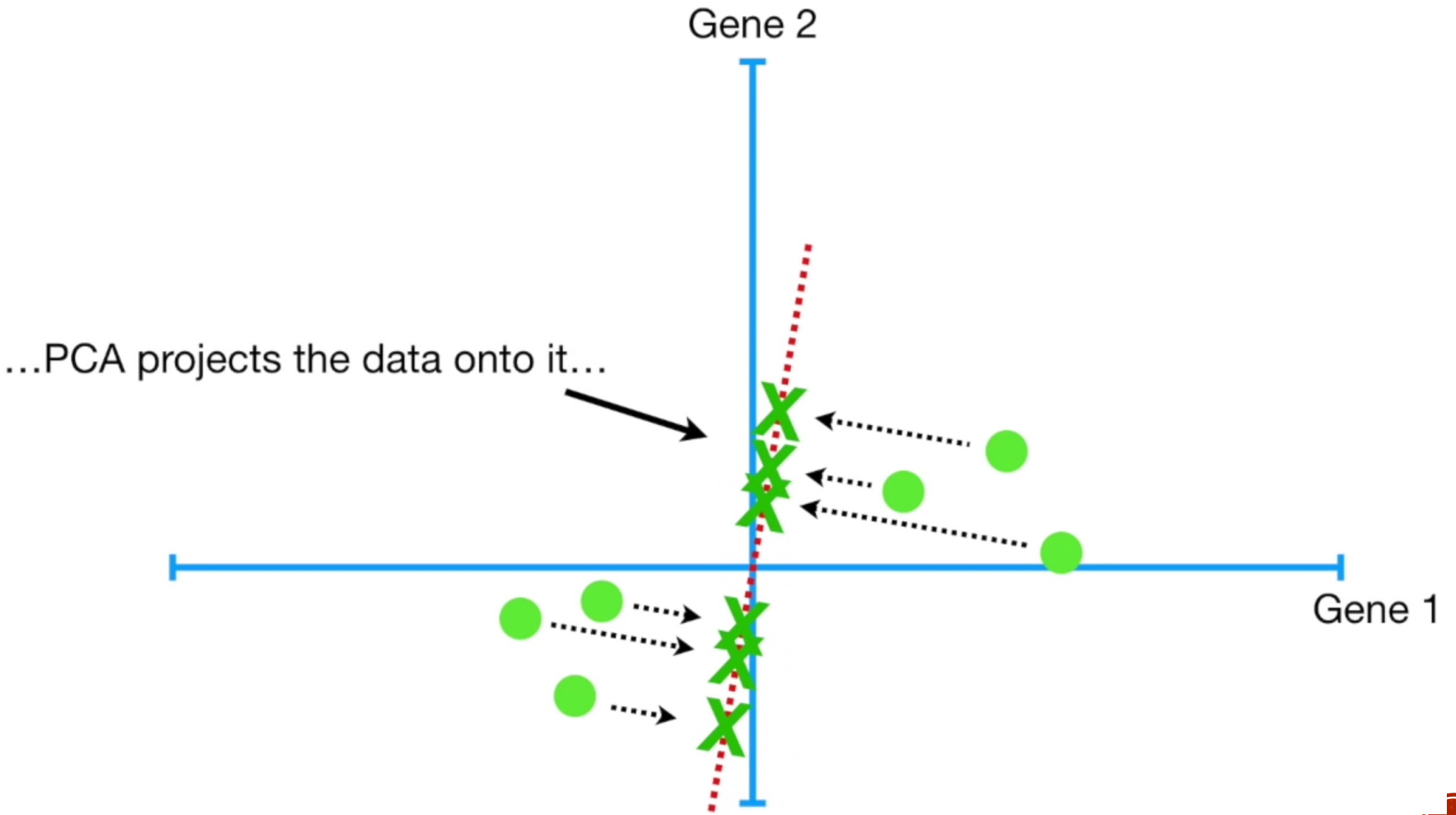

Gene 2
…then c must get smaller.
$a^2 = b^2 + c^2$
b
c
a
Gene 1

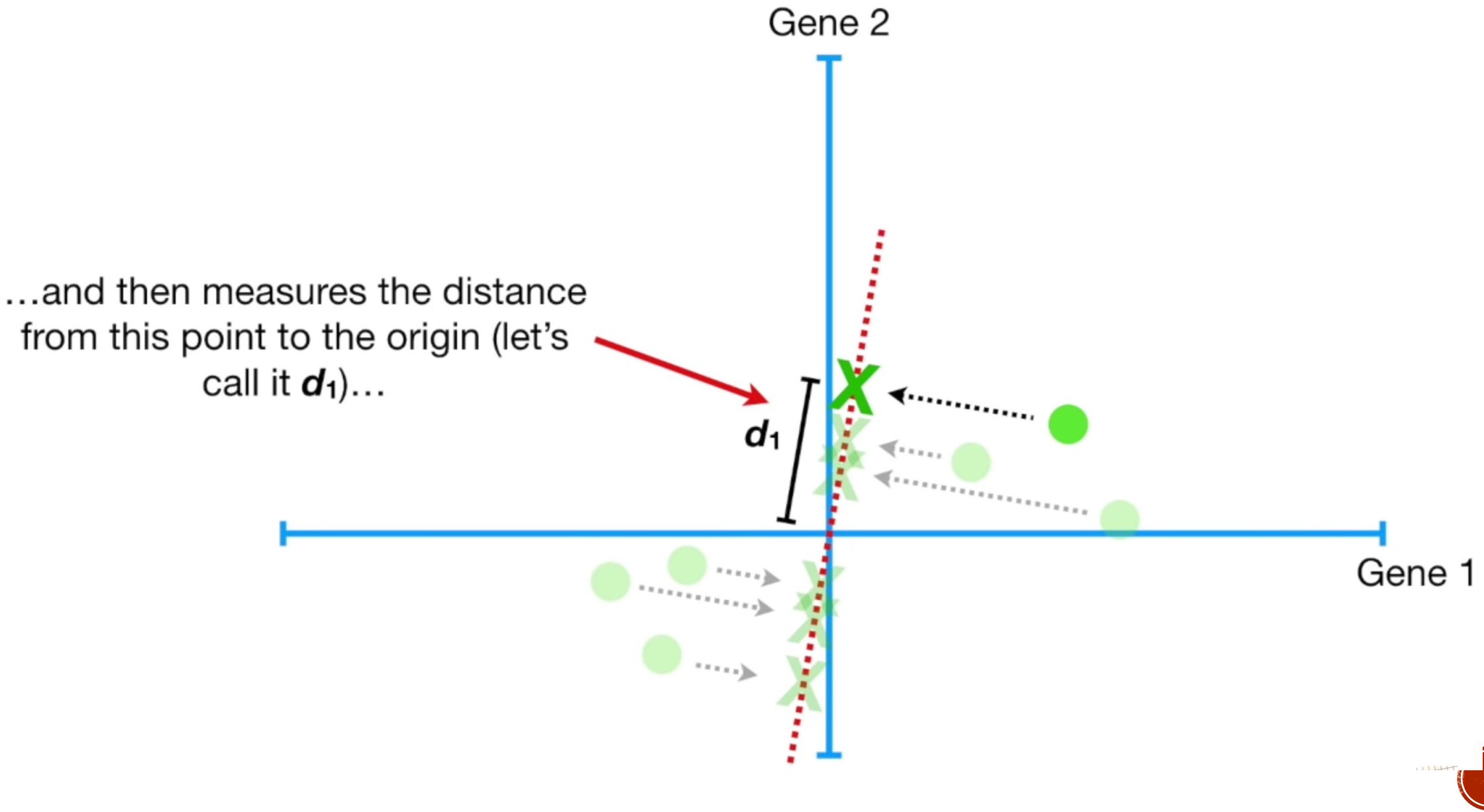

Gene 2
Likewise, if c gets bigger…
a² = b² + c²
c
b
a
Gene 1

…or ***maximize*** the distance from the projected point to the origin.

$a^2 = b^2 + c^2$

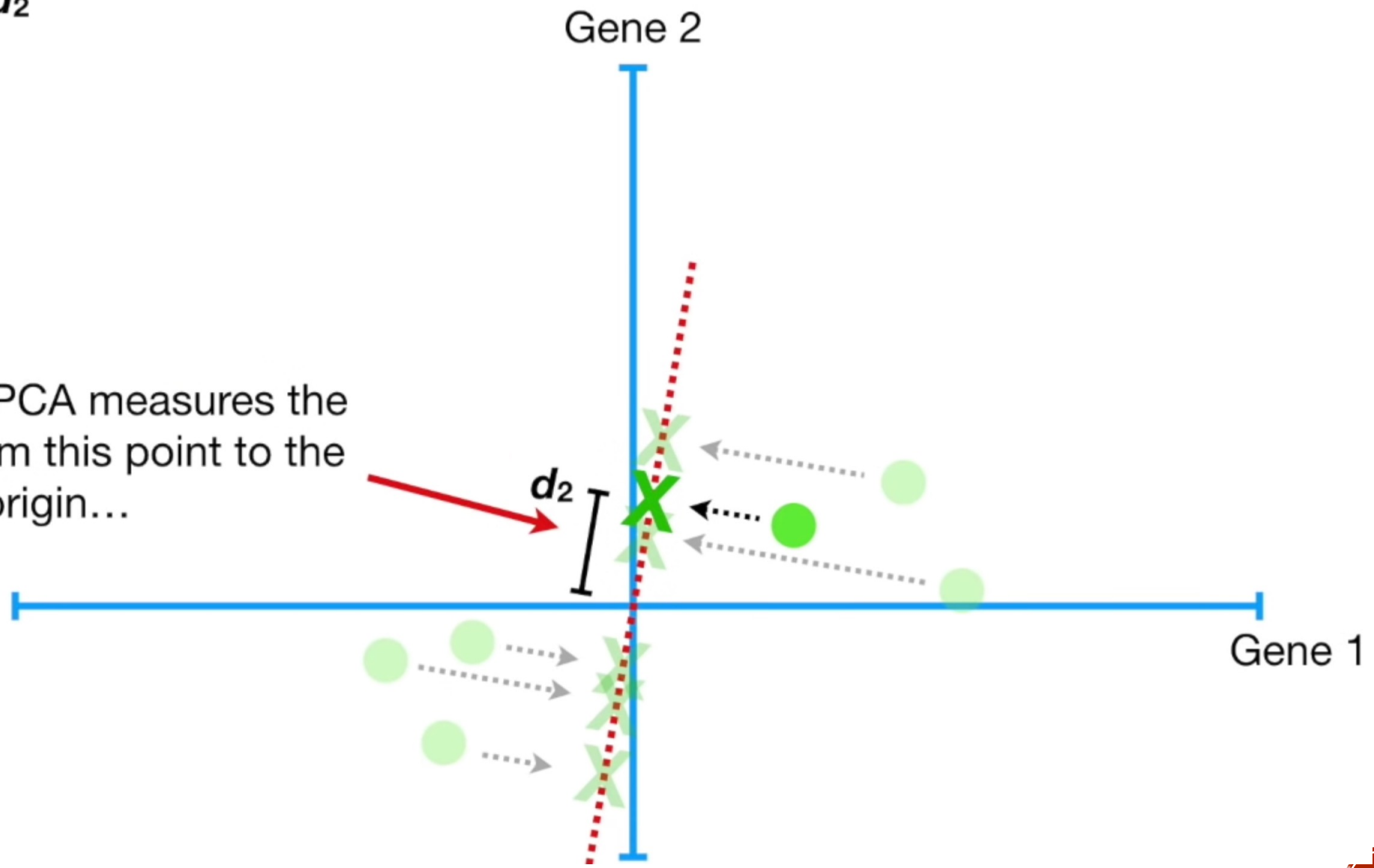



Gene 2
…PCA projects the data onto it…
Gene 1

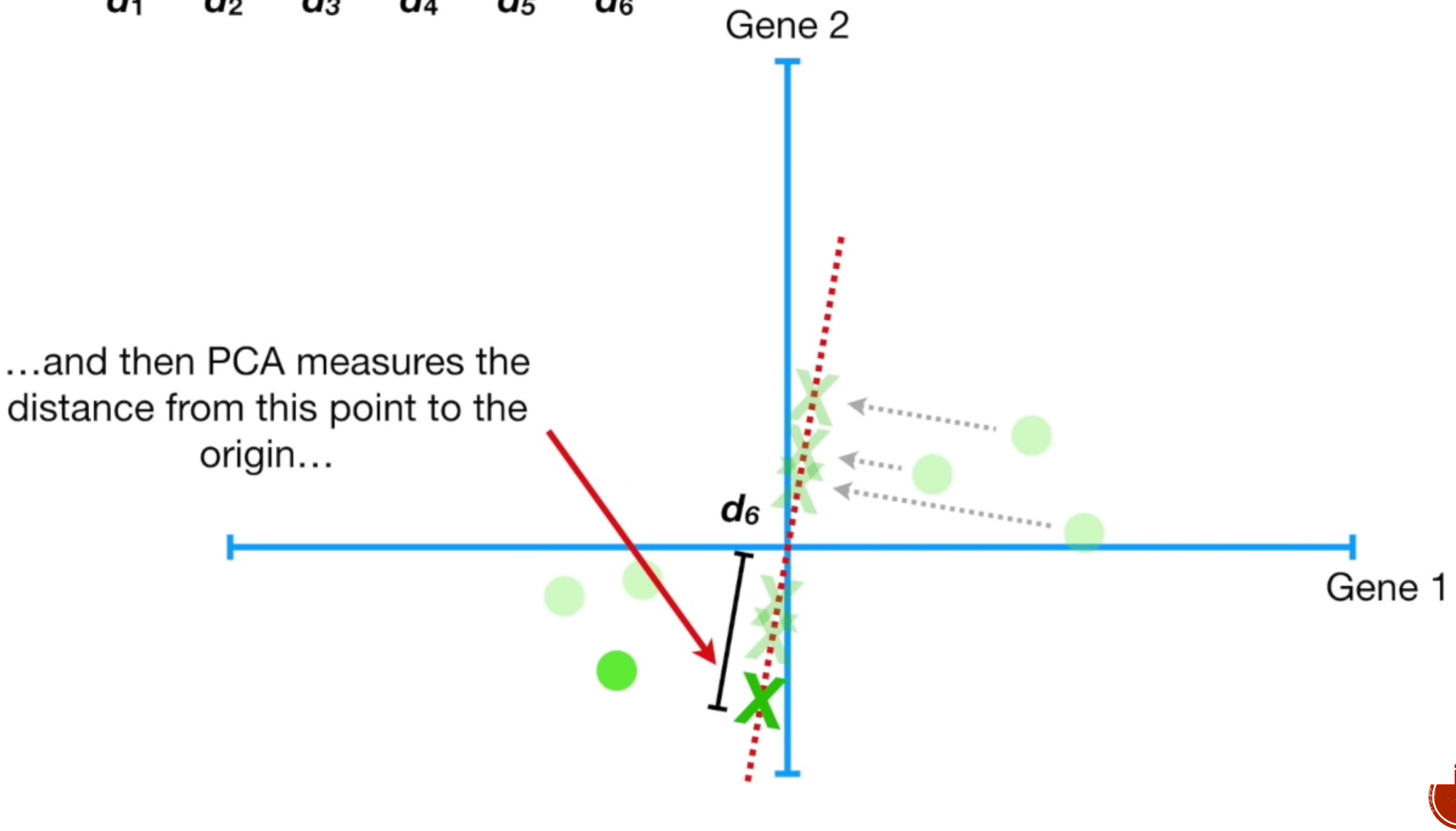

Gene 2
…and then measures the distance from this point to the origin (let's call it d_1)…
d_1
Gene 1

d_1 d_2

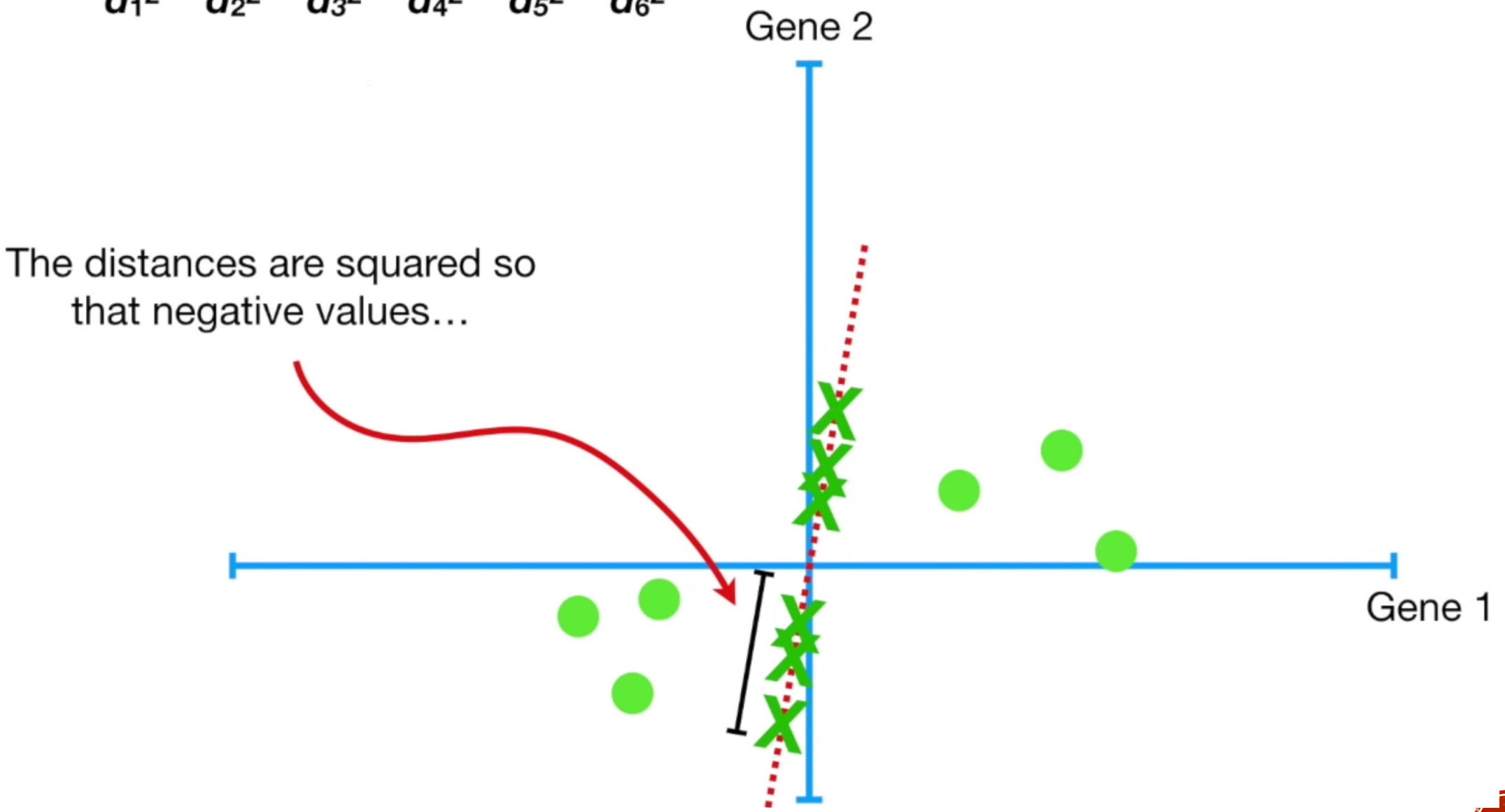
…and then PCA measures the distance from this point to the origin…



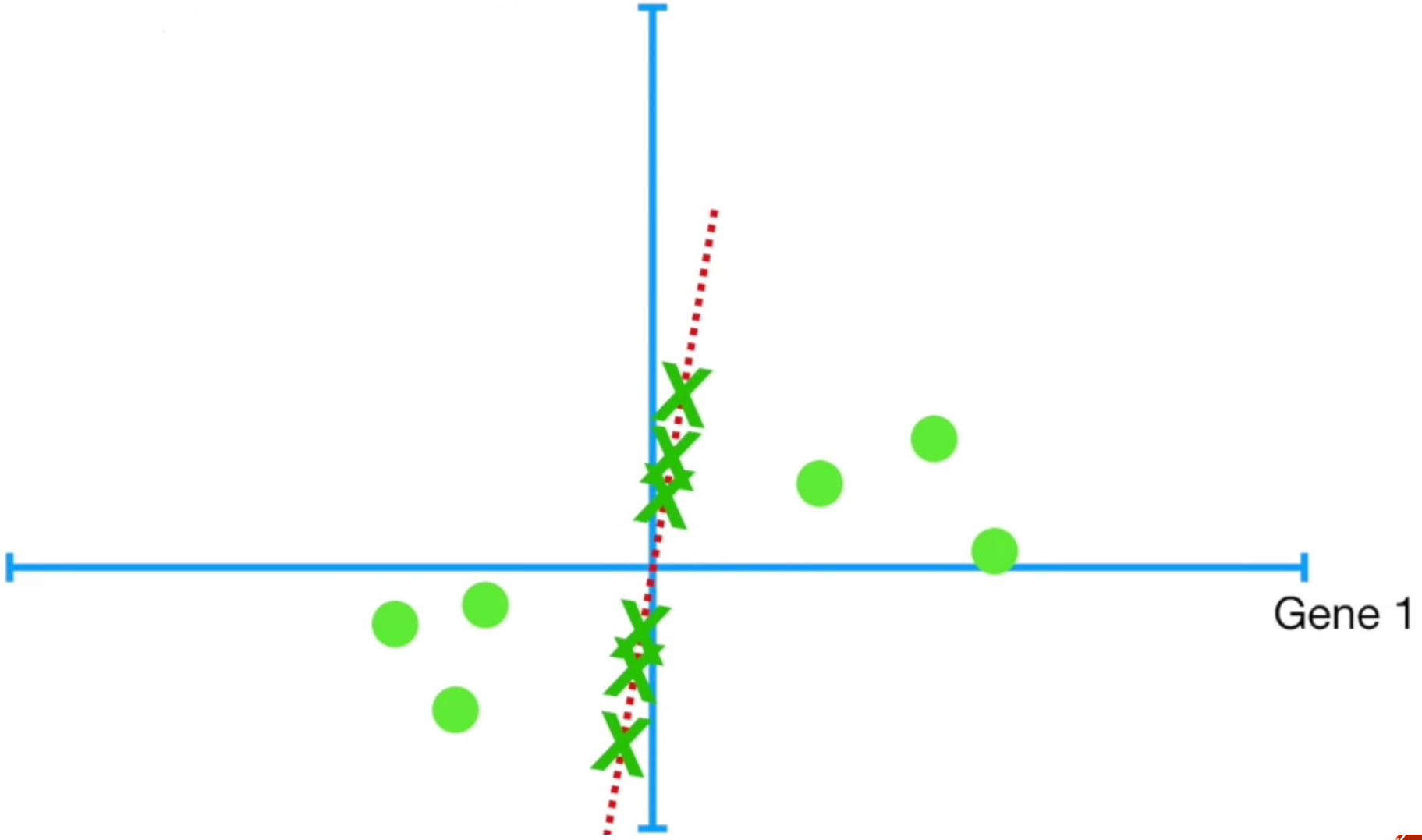
d_1 d_2 d_3 d_4 d_5 d_6



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

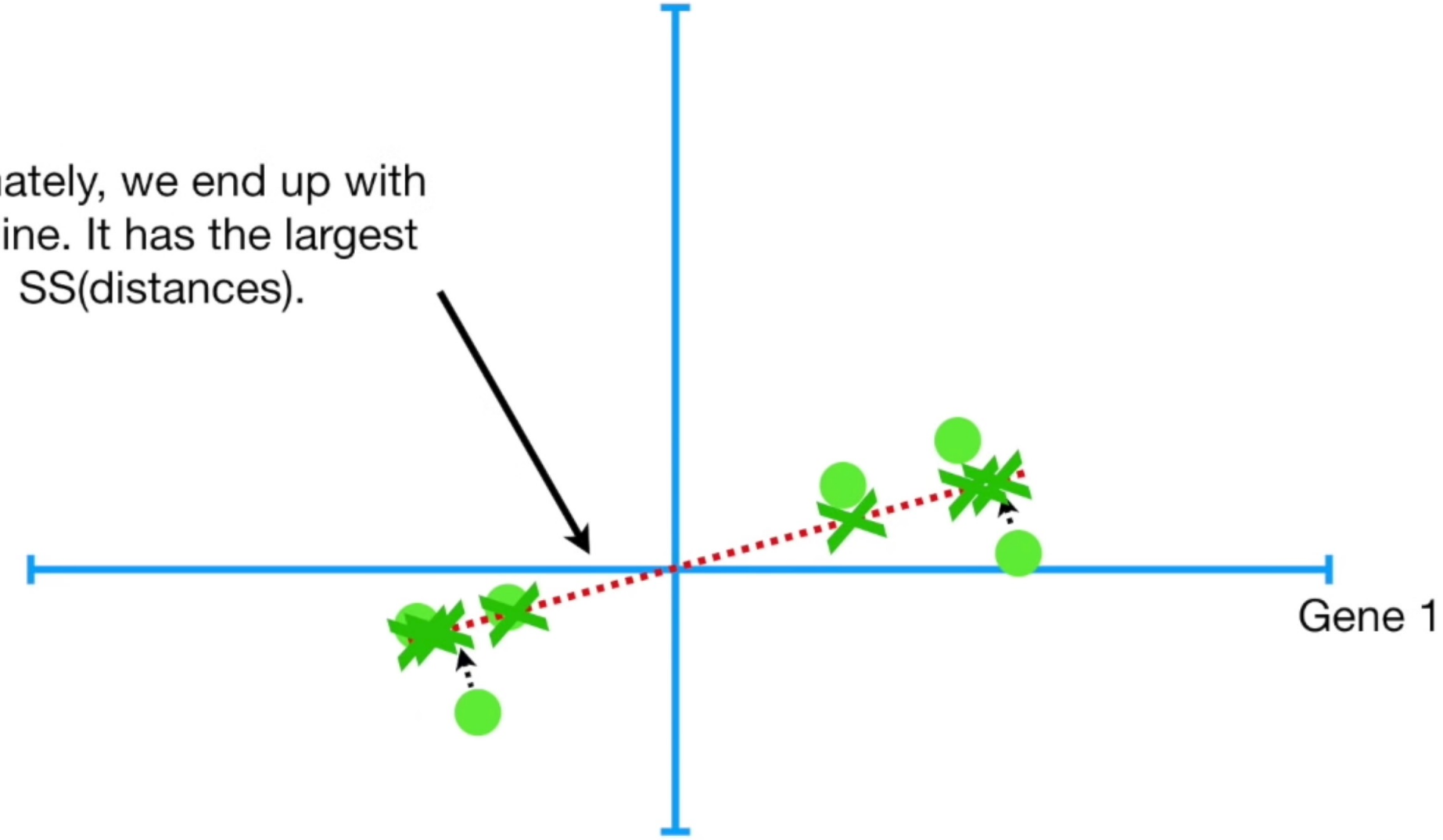


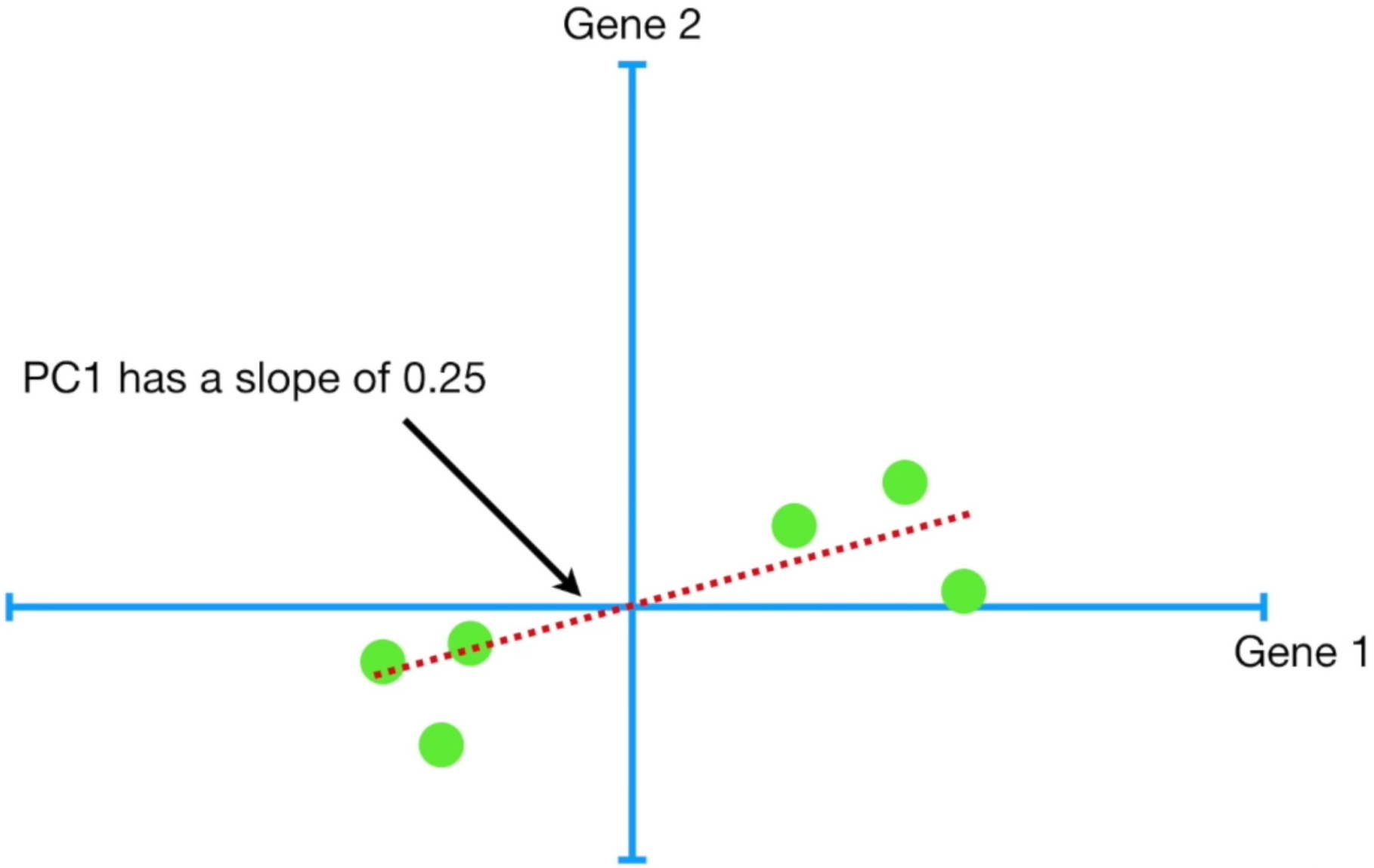
$d_1^2 + d_2^2 + d_3^2 + d_4^2 + d_5^2 + d_6^2$ = sum of squared distances



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

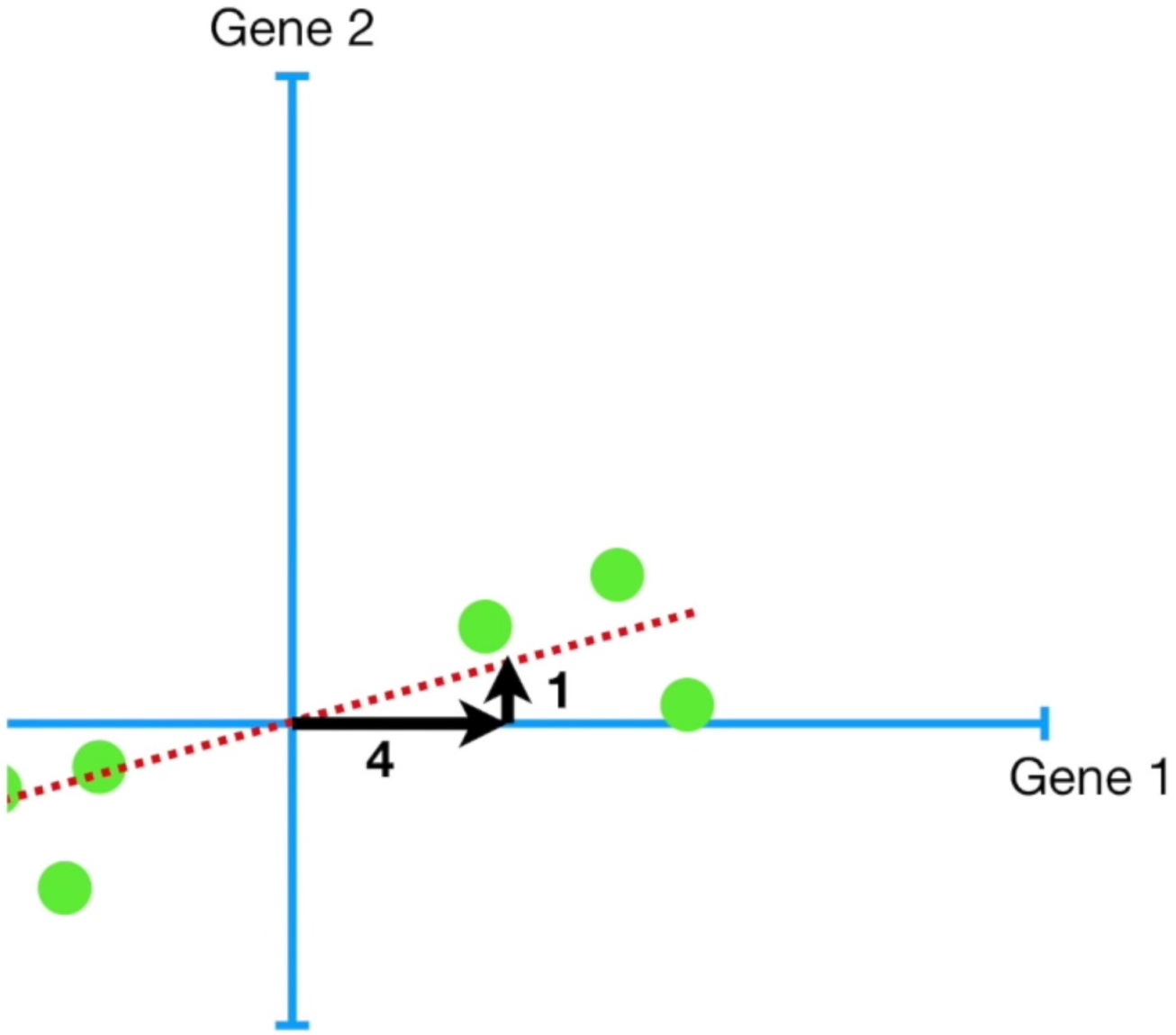
Ultimately, we end up with this line. It has the largest SS(distances).




Gene 2
PC1 has a slope of 0.25
Gene 1

To make PC1
Mix **4** parts Gene 1
with **1** part Gene 2

Terminology Alert!!!!
Mathematicians call this cocktail recipe a "*linear combination*" of Genes 1 and 2.

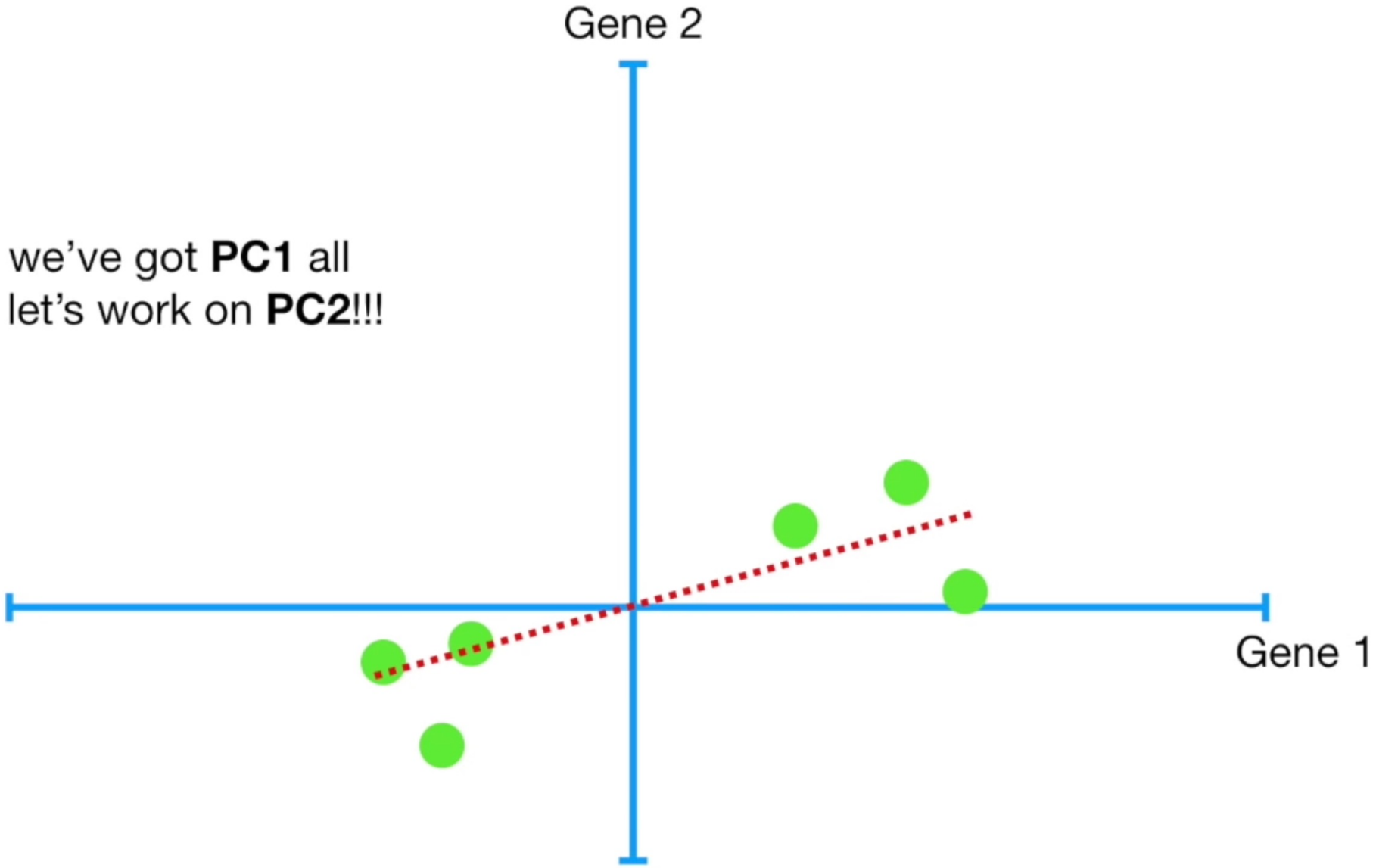


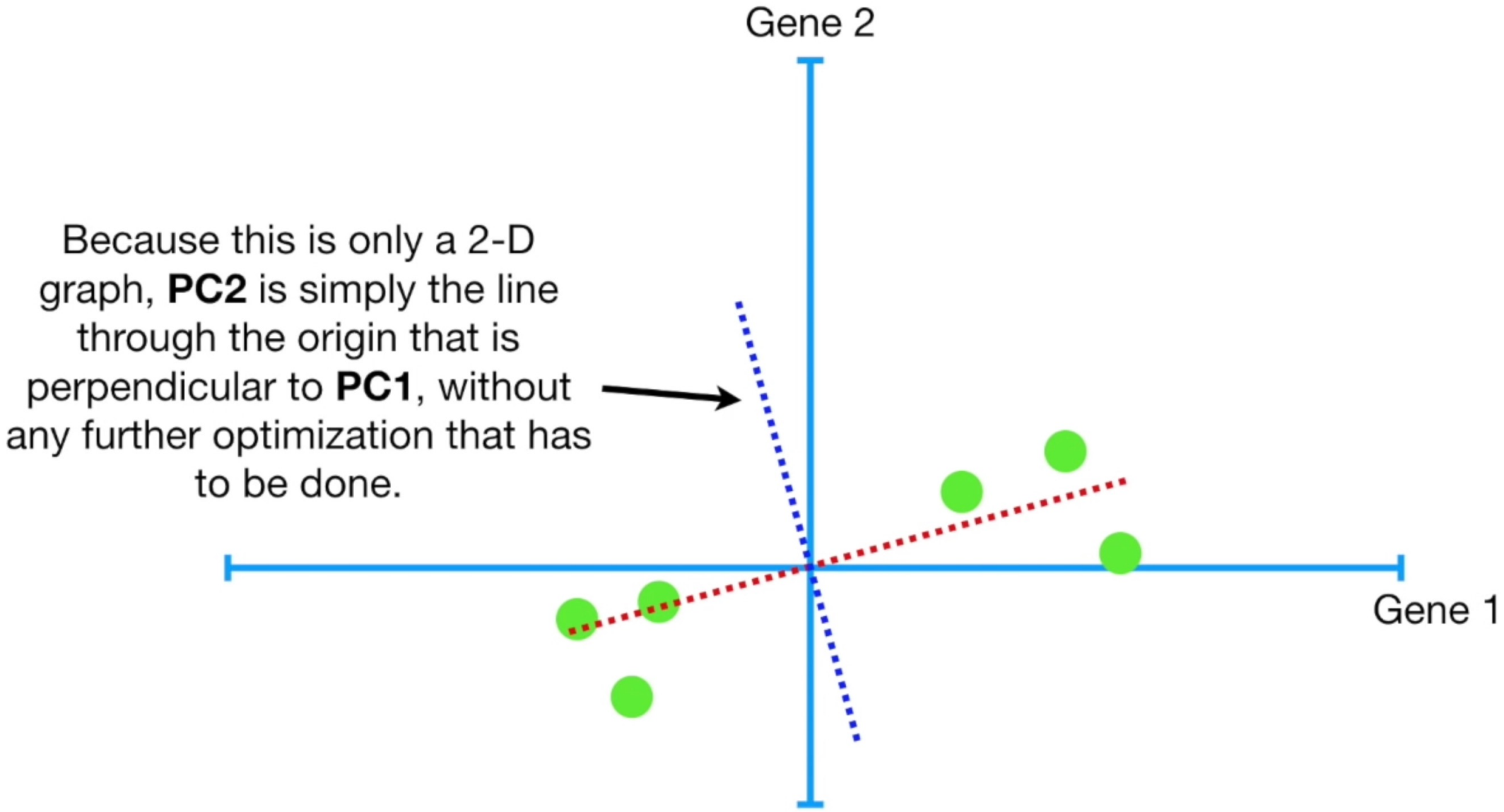
The new values change our recipe…

To make PC1
Mix **0.97** parts Gene 1
with **0.242** parts Gene 2

…but the ratio is the same: we still use 4 times as much Gene 1 as Gene 2.

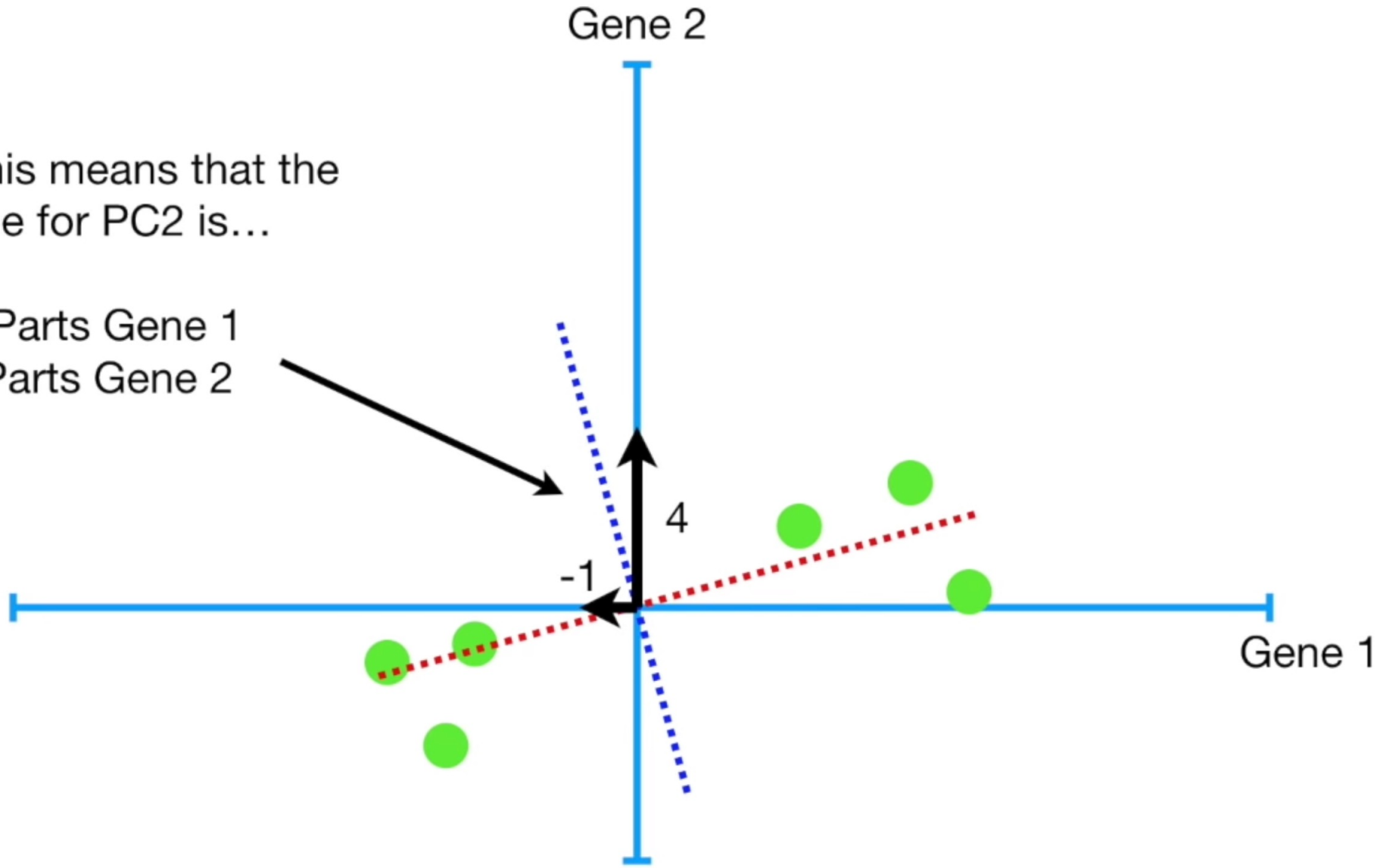
Now that we've got **PC1** all figured out let's work on **PC2**!!!




Gene 2
Because this is only a 2-D graph, **PC2** is simply the line through the origin that is perpendicular to **PC1**, without any further optimization that has to be done.
Gene 1

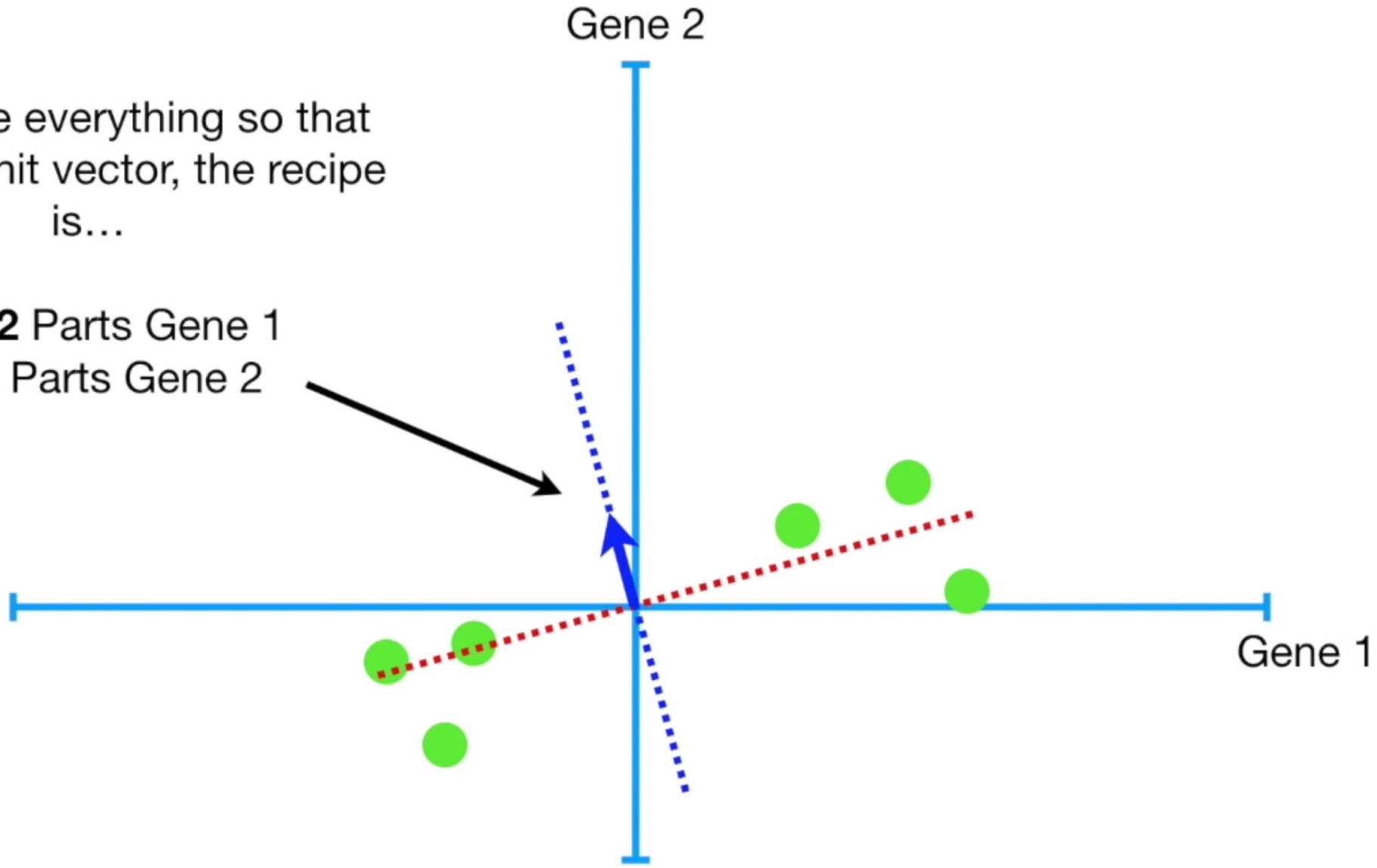
…and this means that the recipe for PC2 is…

-1 Parts Gene 1
4 Parts Gene 2

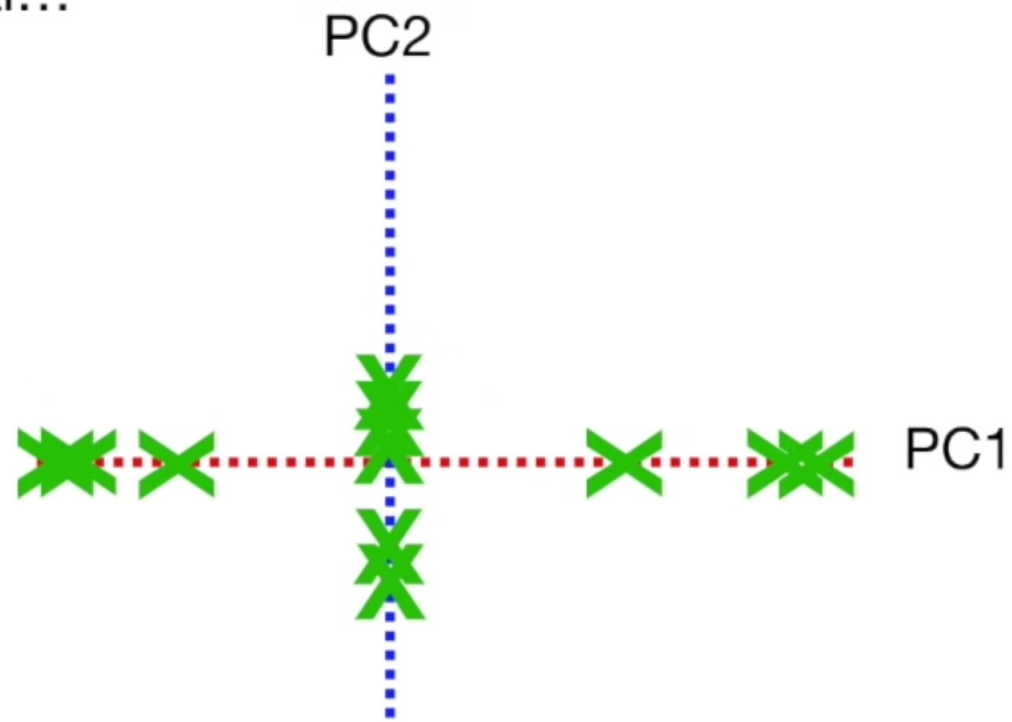


If we scale everything so that we get a unit vector, the recipe is…

-0.242 Parts Gene 1
0.97 Parts Gene 2



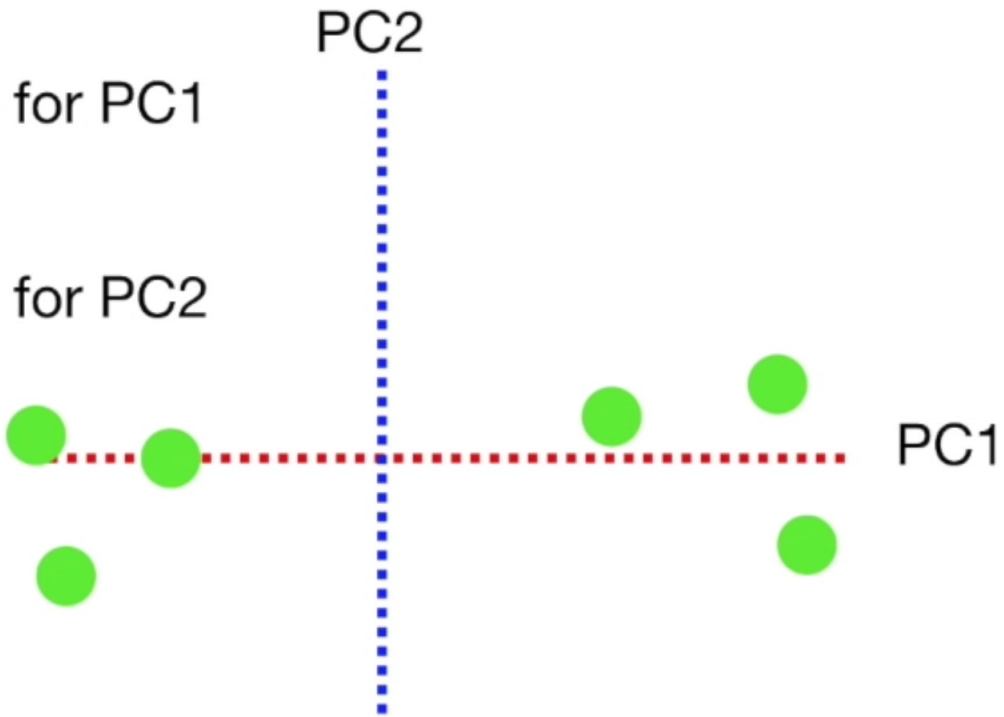
We simply rotate everything so that PC1 is horizontal…



We can convert them into variation around the origin (0, 0) by dividing by the sample size minus 1 (i.e. n - 1).

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

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

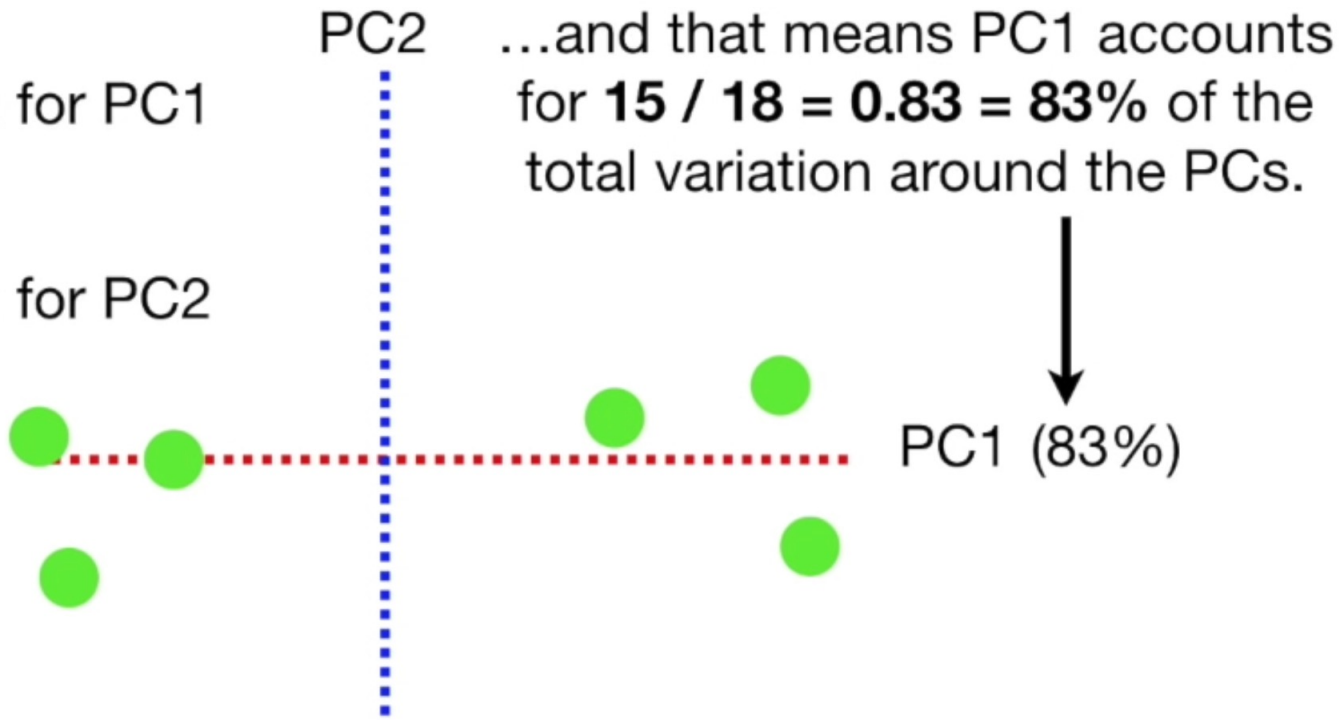


For the sake of the example, imagine that the Variation for **PC1 = 15**, and the variation for **PC2 = 3**.

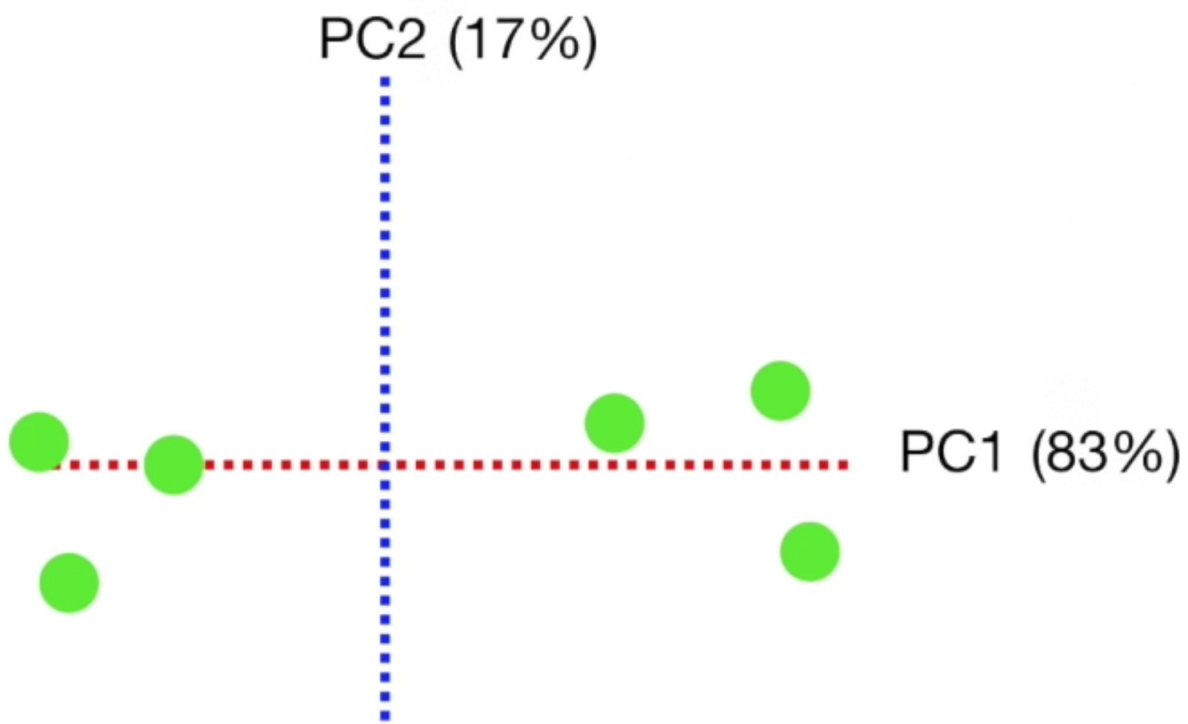
That means that the total variation around both PCs is **15 + 3 = 18**…

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

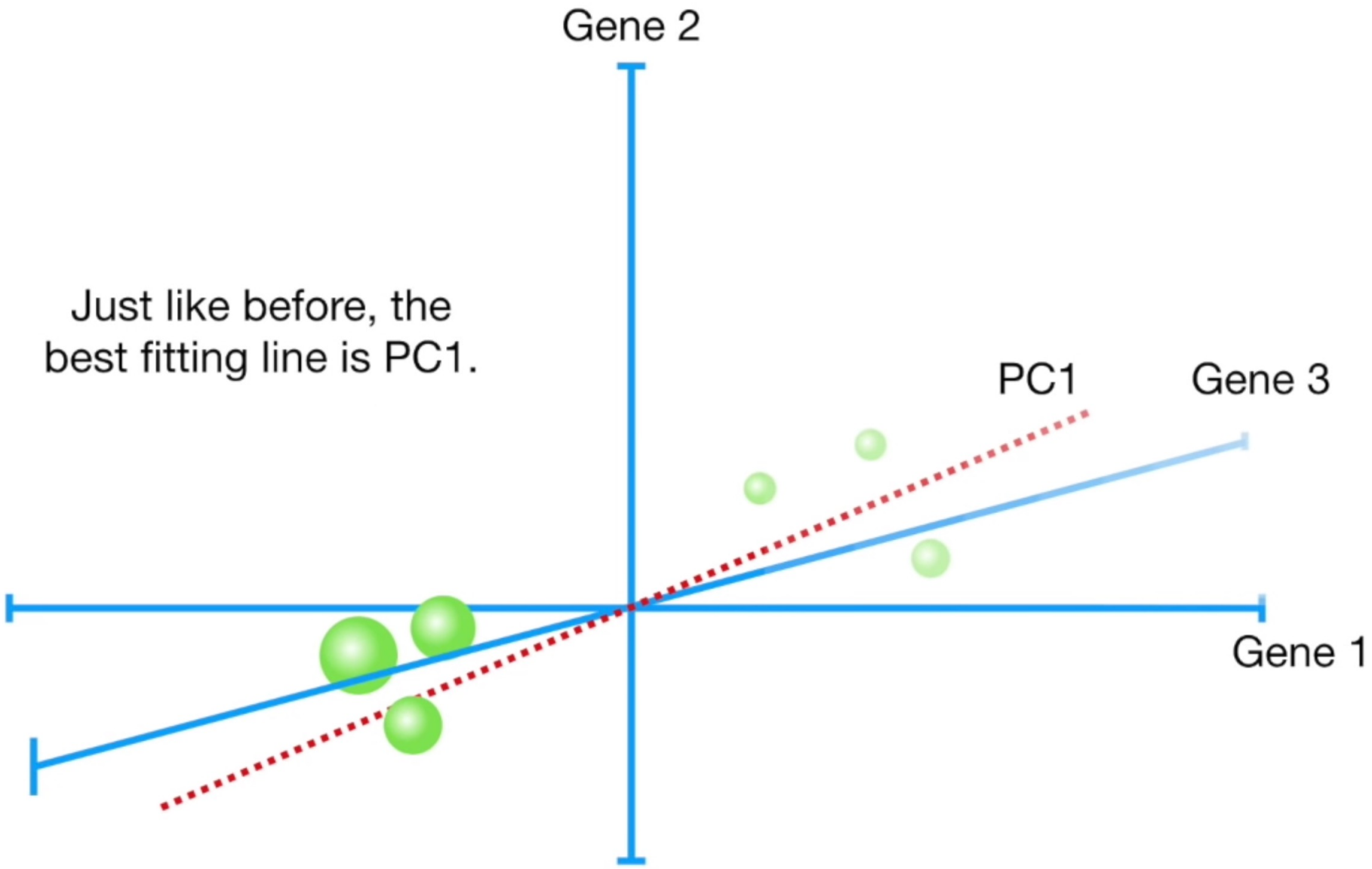
$$\frac{SS(\text{distances for PC2})}{n - 1} = \text{Variation for PC2}$$

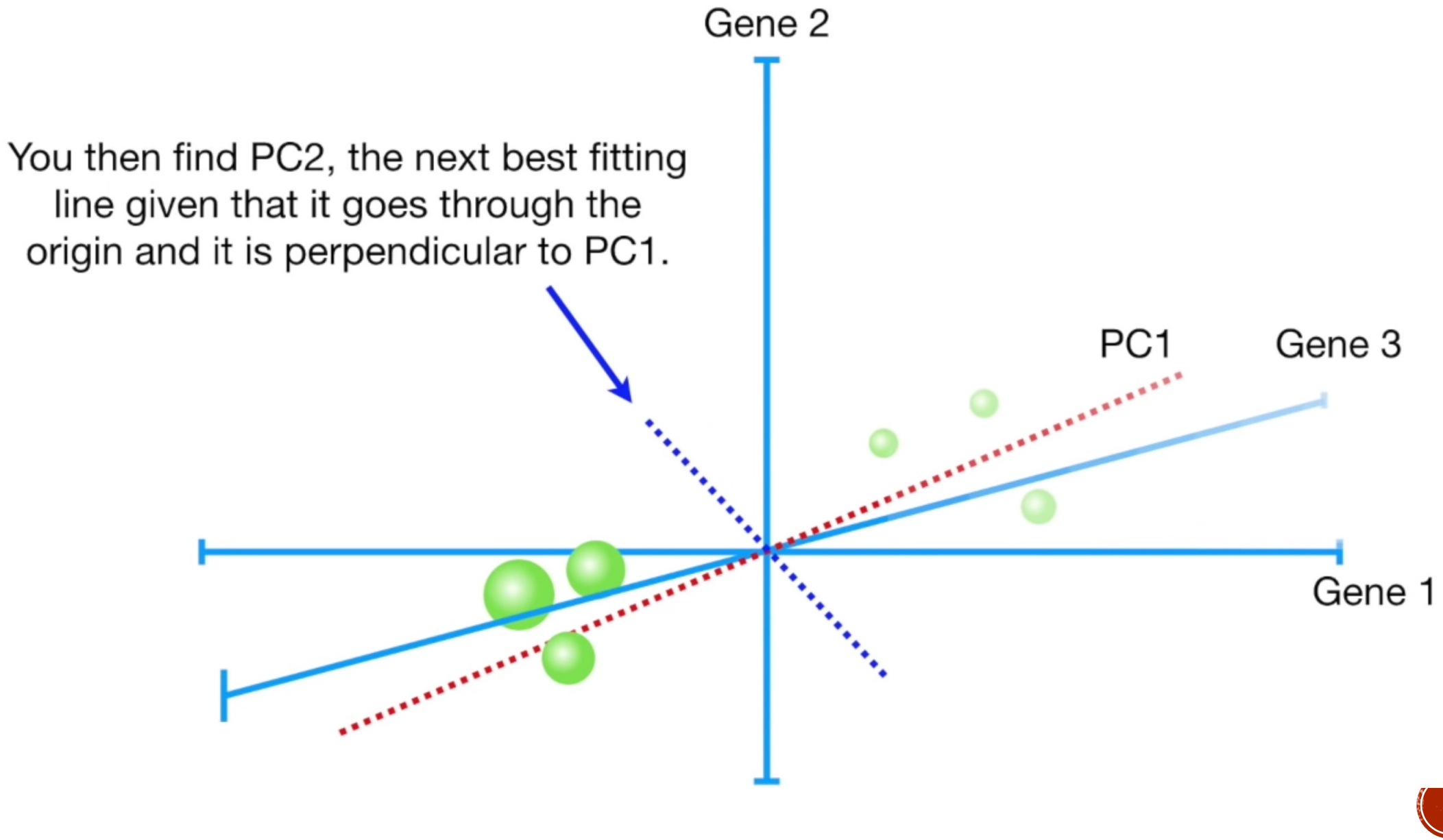


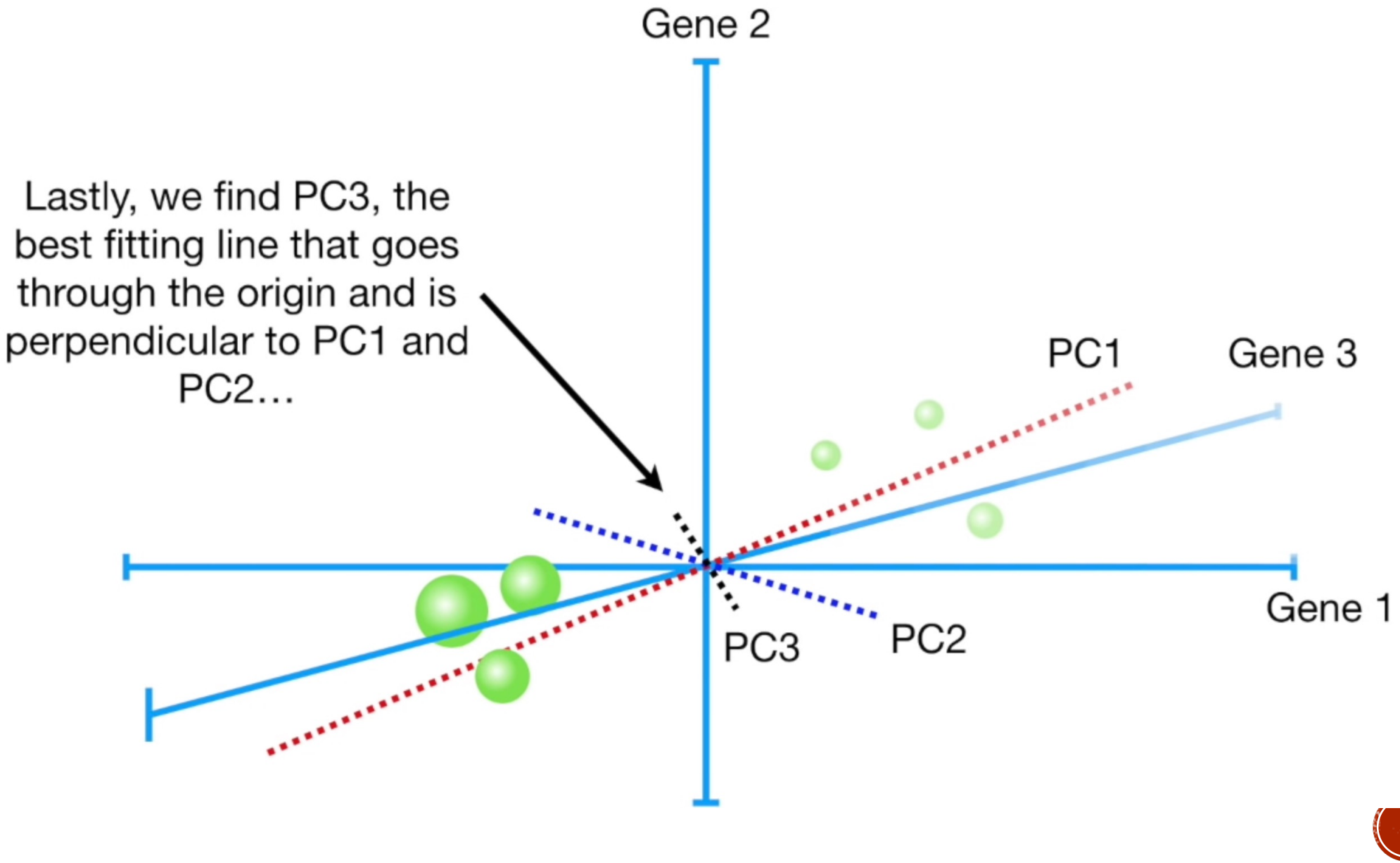
PC2 accounts for **3 / 18 = 0.17 = 17%** of the total variation around the PCs.



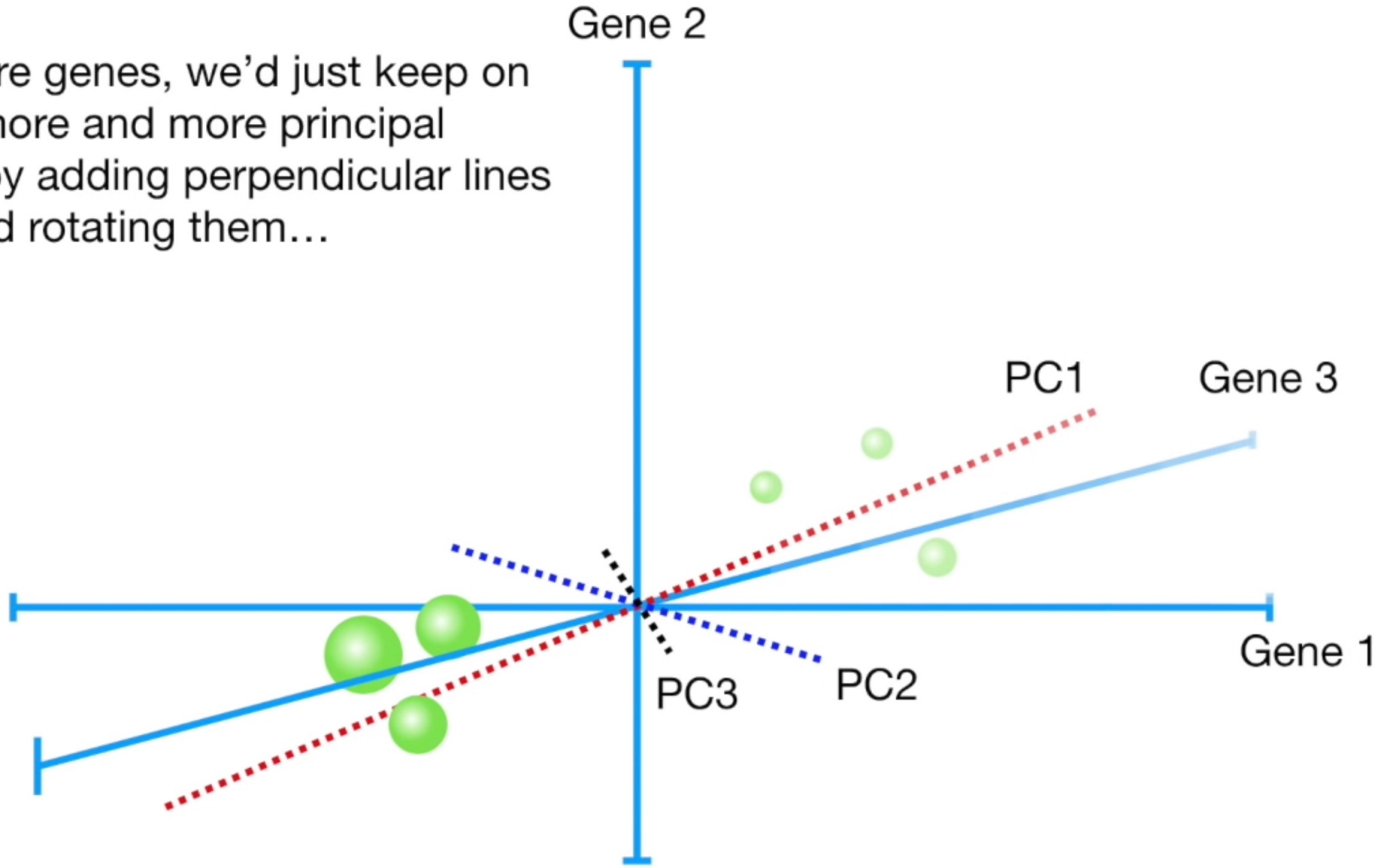
For 3 Features..

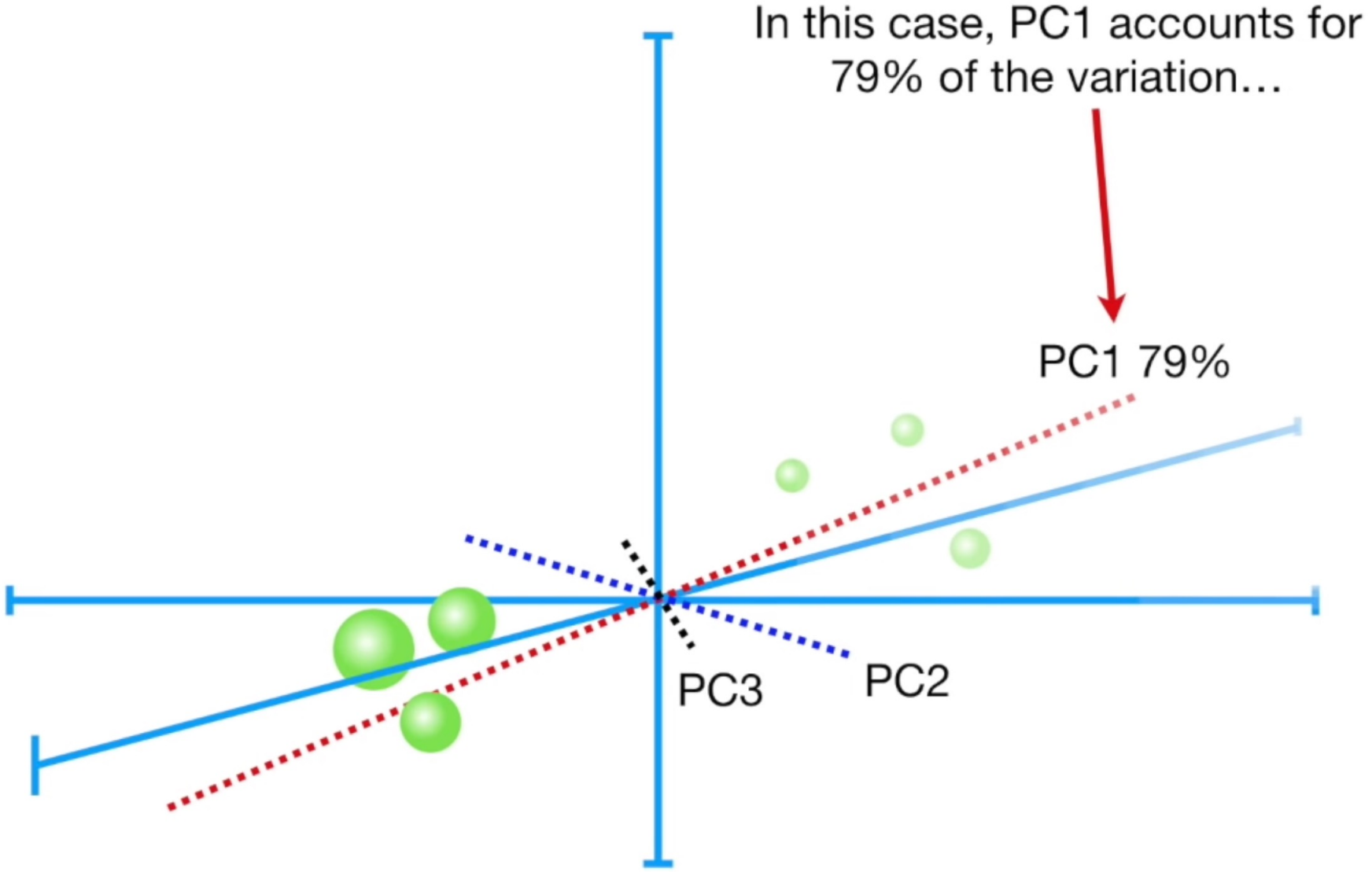


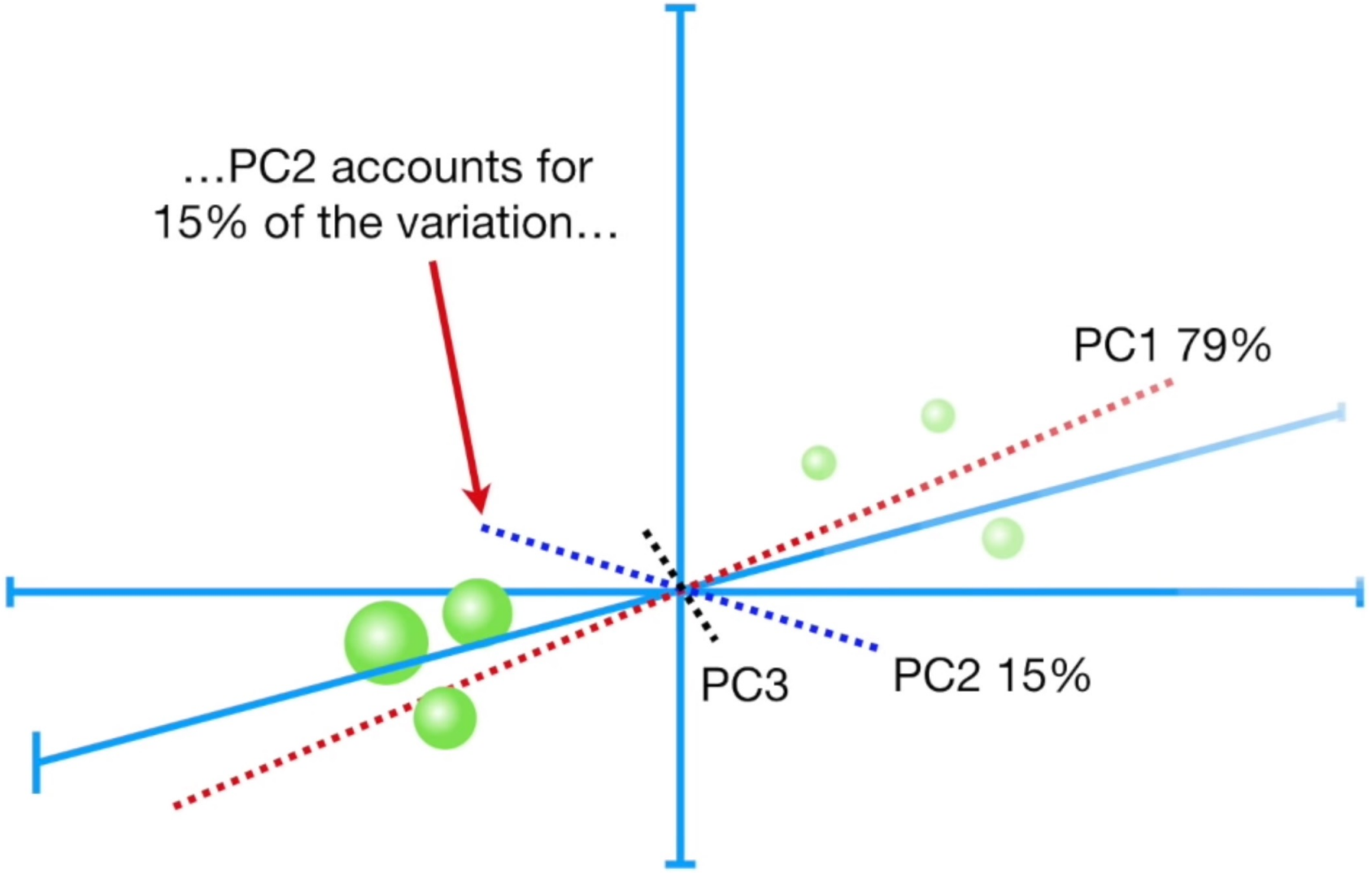

You then find PC2, the next best fitting line given that it goes through the origin and it is perpendicular to PC1.
Gene 2
PC1
Gene 3
Gene 1

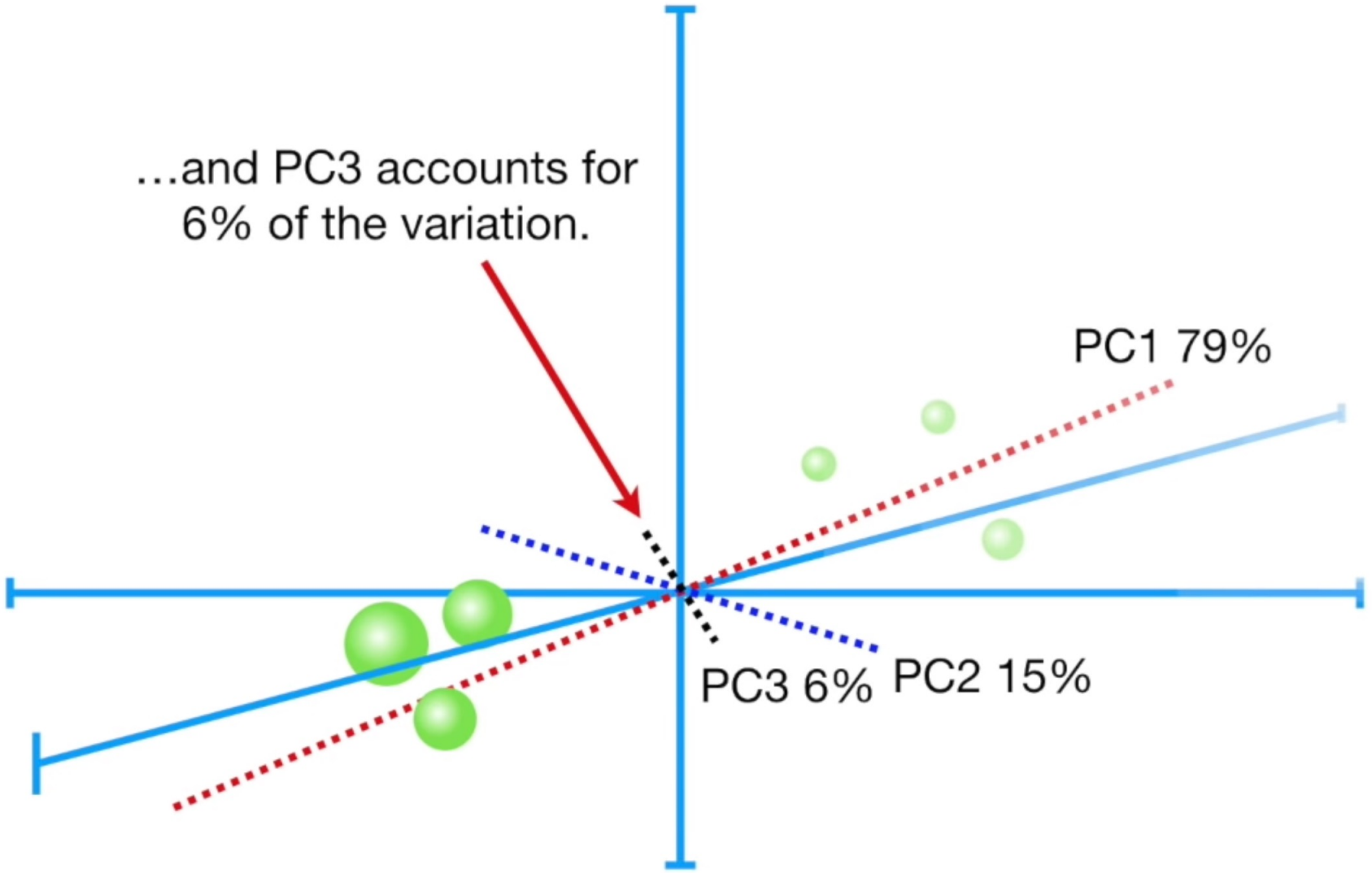

Gene 2
Lastly, we find PC3, the best fitting line that goes through the origin and is perpendicular to PC1 and PC2…
PC1
Gene 3
Gene 1
PC3
PC2

If we had more genes, we'd just keep on finding more and more principal components by adding perpendicular lines and rotating them…




In this case, PC1 accounts for
79% of the variation…
PC1 79%
PC3
PC2


…PC2 accounts for
15% of the variation…
PC1 79%
PC3
PC2 15%


…and PC3 accounts for
6% of the variation.
PC1 79%
PC3 6%
PC2 15%