446 Section 10/10

Plans for today!

1. This
2. Reminders
3. PCA
4. Final Review

Reminders

- HW4 now due Thursday (June 4th)!
- Final is Wednesday June 10th!
  - In ~6 days…
- **Final Exam Review Session - Monday, June 8th**
  - **Time: 5:30pm - 7pm**
  - **Location: ECE 105**
- SVD will **NOT** be tested on the final

SVD

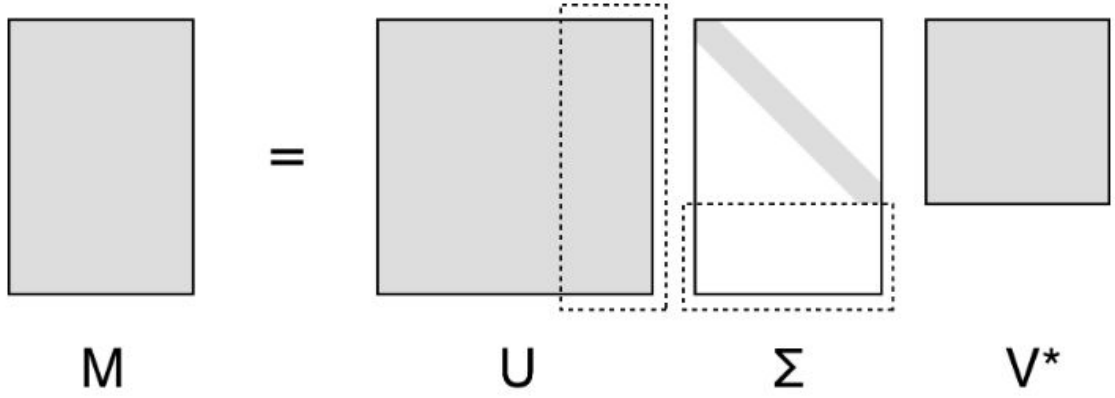
Note that this is not on the final

Singular **V**alue **D**ecomposition

Fact: Any matrix **M** can be decomposed into three separate matrices.

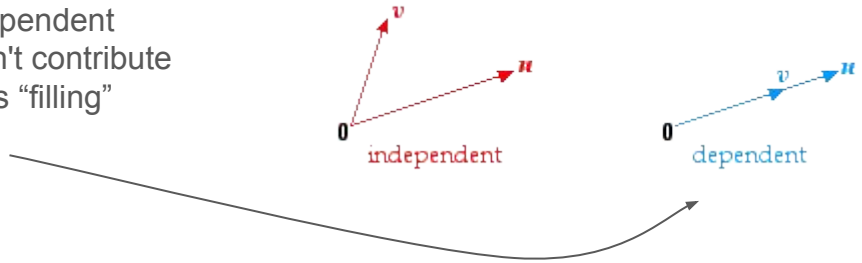
(BTW: This is equivalent to thinking of **M** as a rotation, scaling, and another rotation).

Say **M** is rank **r**



Alternative explanation:
Rank is how much space the matrix “fills”.

Linearly dependent vectors don't contribute more to this “filling”



How I like to think of rank: How much space is **actually** taken up by the vectors in the matrix?

Rank-r approximations

Say that **M** is “bloated”. It has many linearly dependent vectors which increase its size by a lot for no reason.

The “information” of **M** is confined to the subspace spanned by its linearly independent vectors

Basically minimizing MSE for the matrix reconstruction

$$\text{minimize}_{L \in \mathbb{R}^{m \times n}} \sum_{i=1}^{m} \sum_{j=1}^{n} (A_{i,j} - L_{i,j})^2$$

$$\text{subject to } \text{rank}(L) = r$$

This is SVD

- The optimal rank-r approximation is $L = U_{1:r} S_{1:r,1:r} V_{1:r}^T$

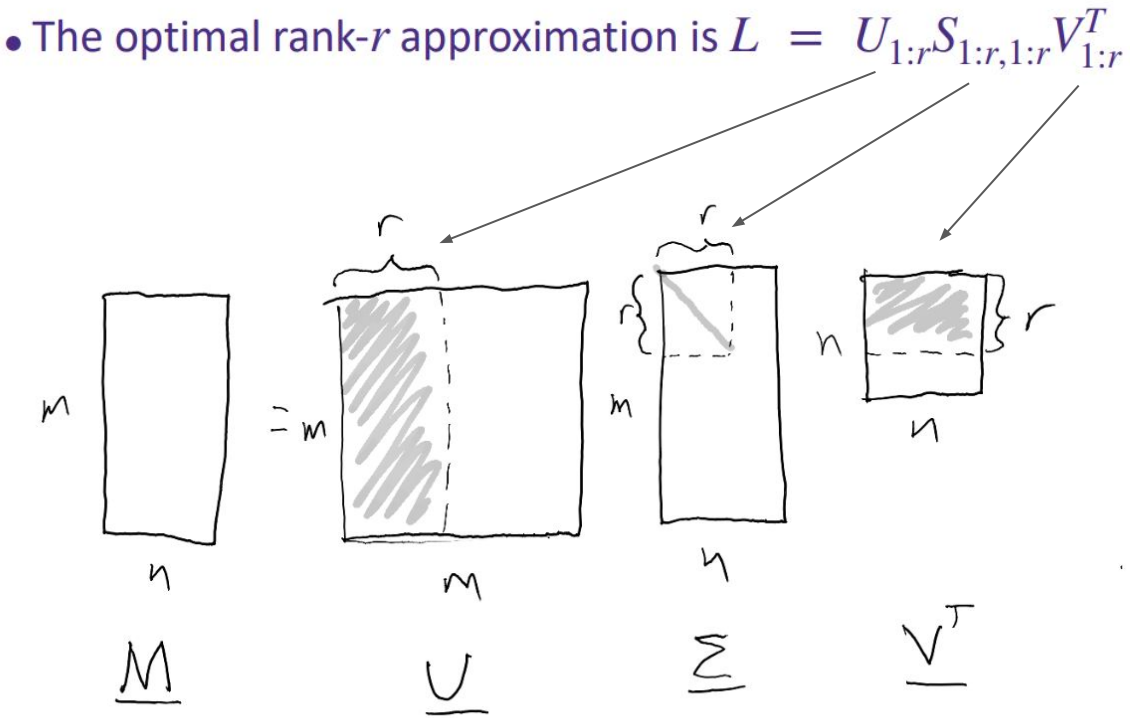
TL;DR: SVD is useful because it lets us approximately describe **M** with much less data (especially if **M** is “bloated”)

What does this look like in terms of **U, S, and V?**

Rank-r approximations

Shaded areas are what is “enough” to reconstruct **M**. Blank areas are 0

For compression using SVD, we pick **r** to be small to save us lots of space!

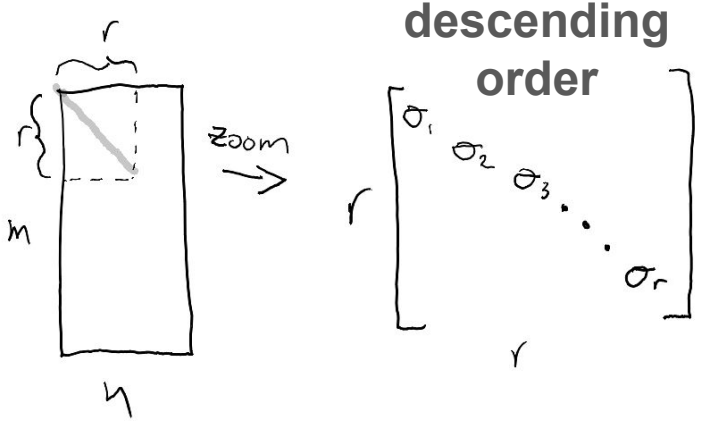


Understanding the decomposition

Consider what actually happens during the matrix multiplication.

Give names to the diagonal values of **Σ**.

Write the multiplication as a summation.



Sorted in descending order

Σ

Indexes refer to columns

$$\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top = \sum_{i=1}^{r} u_i \sigma_i v_i^\top$$

Understanding the decomposition

Make sure the shapes check out in your head:

$$\mathbf{M} \in \mathbb{R}^{m \times n}$$

$$u_i \in \mathbb{R}^m;\ \sigma_i \in \mathbb{R};\ v_i \in \mathbb{R}^n \therefore$$

$$u_i \sigma_i v_i^\top \in \mathbb{R}^{m \times n}$$

$$\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top = \sum_{i=1}^{r} u_i \sigma_i v_i^\top$$

Rank-r approximation

Rank-2 approximation

Rank-1 approximation

$$\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top = u_1 \sigma_1 v_1^\top + u_2 \sigma_2 v_2^\top + \ldots + u_r \sigma_r v_r^\top$$

Understanding the decomposition

Make sure the shapes check out in your head:

$$\mathbf{M} \in \mathbb{R}^{m \times n}$$

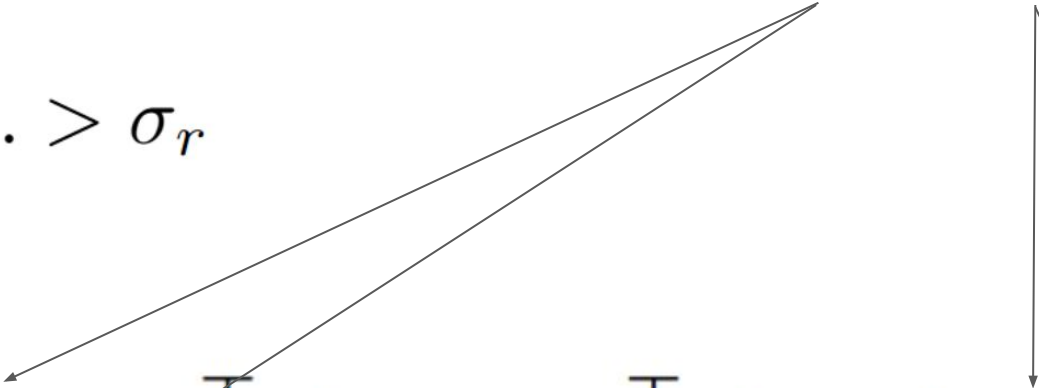
$$u_i \in \mathbb{R}^m;\ \sigma_i \in \mathbb{R};\ v_i \in \mathbb{R}^n \therefore$$

$$u_i \sigma_i v_i^\top \in \mathbb{R}^{m \times n}$$

Note: The u_i and v_i vectors are called left and right singular vectors respectively.

Q: Which singular vectors affect the reconstruction of **M** the most and least. Why?

A: $\sigma_1 > \sigma_2 > \ldots > \sigma_r$



$$\mathbf{U\Sigma V}^\top = u_1\sigma_1 v_1^\top + u_2\sigma_2 v_2^\top + \ldots + u_r\sigma_r v_r^\top$$

Section Participation

How are the singular values ordered in terms of Magnitude?

A: Descending

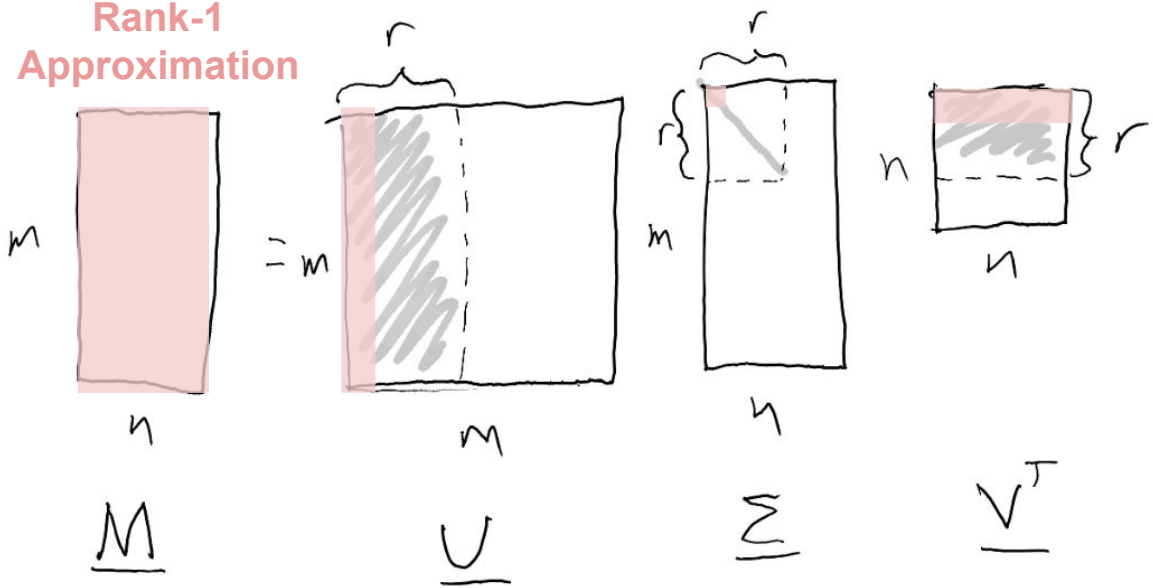
Understanding the decomposition

$$\mathbf{M} \in \mathbb{R}^{m \times n}$$

$$u_i \in \mathbb{R}^m;\ \sigma_i \in \mathbb{R};\ v_i \in \mathbb{R}^n \therefore$$

$$u_i \sigma_i v_i^\top \in \mathbb{R}^{m \times n}$$

Rank-1 Approximation



Analogy:
Coats of paint

$$\mathbf{U\Sigma V}^\top = u_1\sigma_1 v_1^\top + u_2\sigma_2 v_2^\top + \ldots + u_r\sigma_r v_r^\top$$

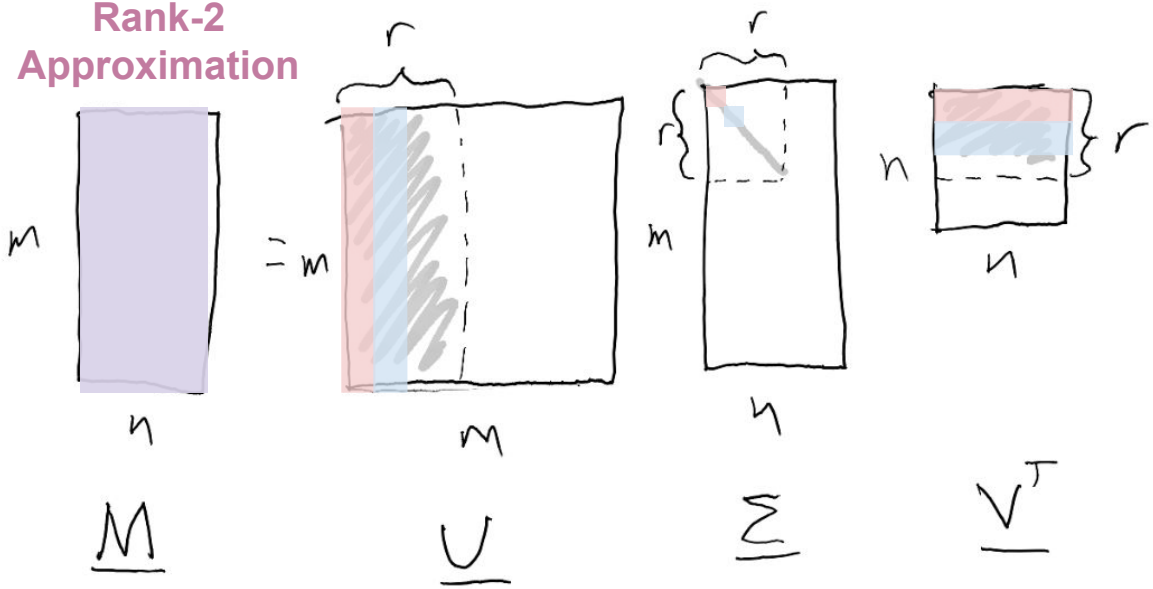
Understanding the decomposition

$$\mathbf{M} \in \mathbb{R}^{m \times n}$$

$$u_i \in \mathbb{R}^m;\ \sigma_i \in \mathbb{R};\ v_i \in \mathbb{R}^n \therefore$$

$$u_i \sigma_i v_i^\top \in \mathbb{R}^{m \times n}$$

Rank-2 Approximation



Analogy:
Coats of paint

$$\mathbf{U\Sigma V}^\top = u_1\sigma_1 v_1^\top + u_2\sigma_2 v_2^\top + \ldots + u_r\sigma_r v_r^\top$$

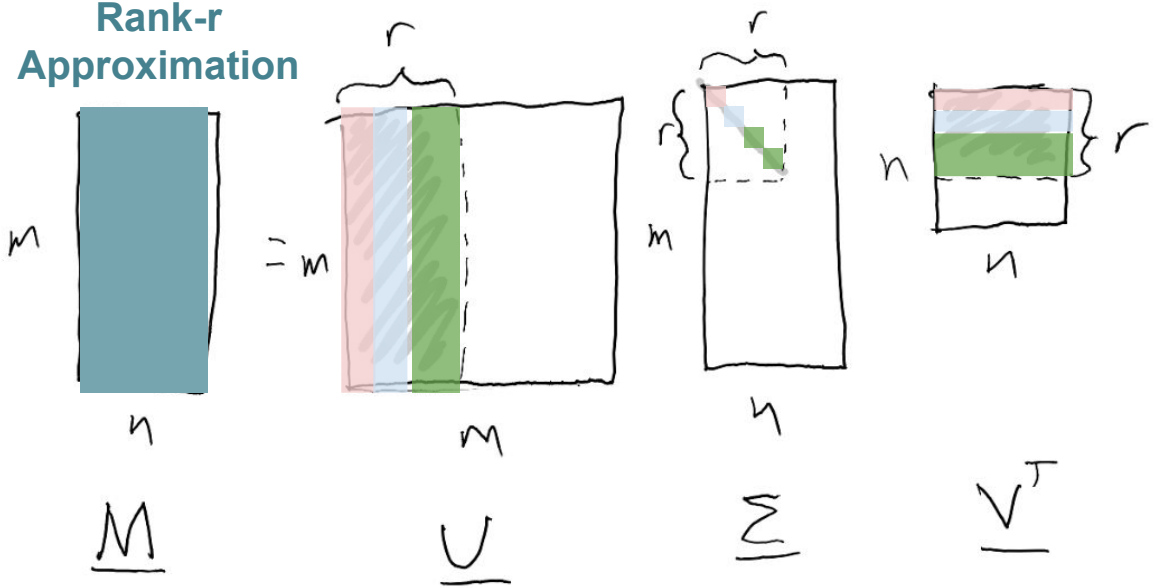
Understanding the decomposition

$$\mathbf{M} \in \mathbb{R}^{m \times n}$$

$$u_i \in \mathbb{R}^m;\ \sigma_i \in \mathbb{R};\ v_i \in \mathbb{R}^n \therefore$$

$$u_i \sigma_i v_i^\top \in \mathbb{R}^{m \times n}$$

Rank-r Approximation



Analogy:
Coats of paint

$$\mathbf{U\Sigma V}^\top = u_1\sigma_1 v_1^\top + u_2\sigma_2 v_2^\top + \ldots + u_r\sigma_r v_r^\top$$

23. 6 points | Select All That Apply

Consider a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ with singular value decomposition $\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^\top$, where $\mathbf{S}$ is an $r \times r$ diagonal matrix and $r = \text{rank}(\mathbf{A}) \leq \min(m, n)$. Which of the following statements are correct?

- [ ] a The columns of $\mathbf{U}$ are the eigenvectors of $\mathbf{A}^\top\mathbf{A}$.
- [x] b The columns of $\mathbf{U}$ are the eigenvectors of $\mathbf{A}\mathbf{A}^\top$.
- [x] c The columns of $\mathbf{V}$ are the eigenvectors of $\mathbf{A}^\top\mathbf{A}$.
- [ ] d The columns of $\mathbf{V}$ are the eigenvectors of $\mathbf{A}\mathbf{A}^\top$.
- [x] e The singular values in $\mathbf{S}$ are the square roots of the nonzero eigenvalues of $\mathbf{A}\mathbf{A}^\top$.
- [x] f The singular values in $\mathbf{S}$ are the square roots of the nonzero eigenvalues of $\mathbf{A}^\top\mathbf{A}$.

Explanation: $\mathbf{A}\mathbf{A}^\top = \mathbf{U}\mathbf{S}^2\mathbf{U}^\top$, implying that the columns of $\mathbf{U}$ are the eigenvectors of $\mathbf{A}\mathbf{A}^\top$ with corresponding eigenvalues along the diagonal of $\mathbf{S}^2$. Similarly, $\mathbf{A}^\top\mathbf{A} = \mathbf{V}\mathbf{S}^2\mathbf{V}^\top$, implying that the columns of $\mathbf{V}$ are the eigenvectors of $\mathbf{A}^\top\mathbf{A}$ with corresponding eigenvalues along the diagonal of $\mathbf{S}^2$.

PCA

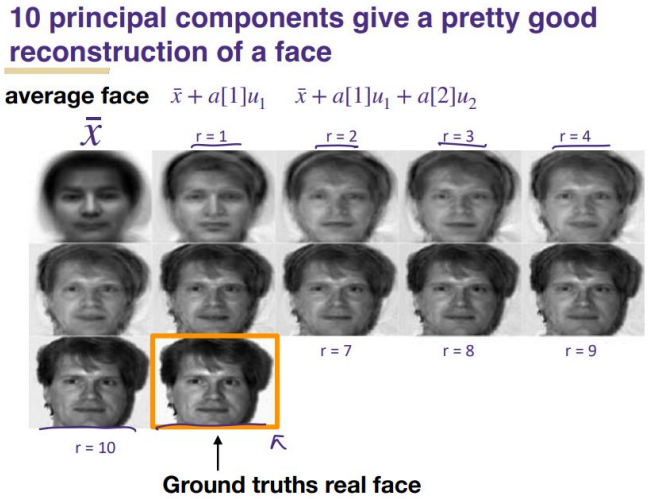
PCA: Dimensionality reduction

Dealing with high-dimensional data can lead to problems (curse of dimensionality)

Visualization is very hard for $d > 2$

Goal: Find a lower-dimensional representation of your data **X** with the least loss of information.

We need to find a subspace of the original data span **to project down to**. The **Principal Components** of our data can help us do this

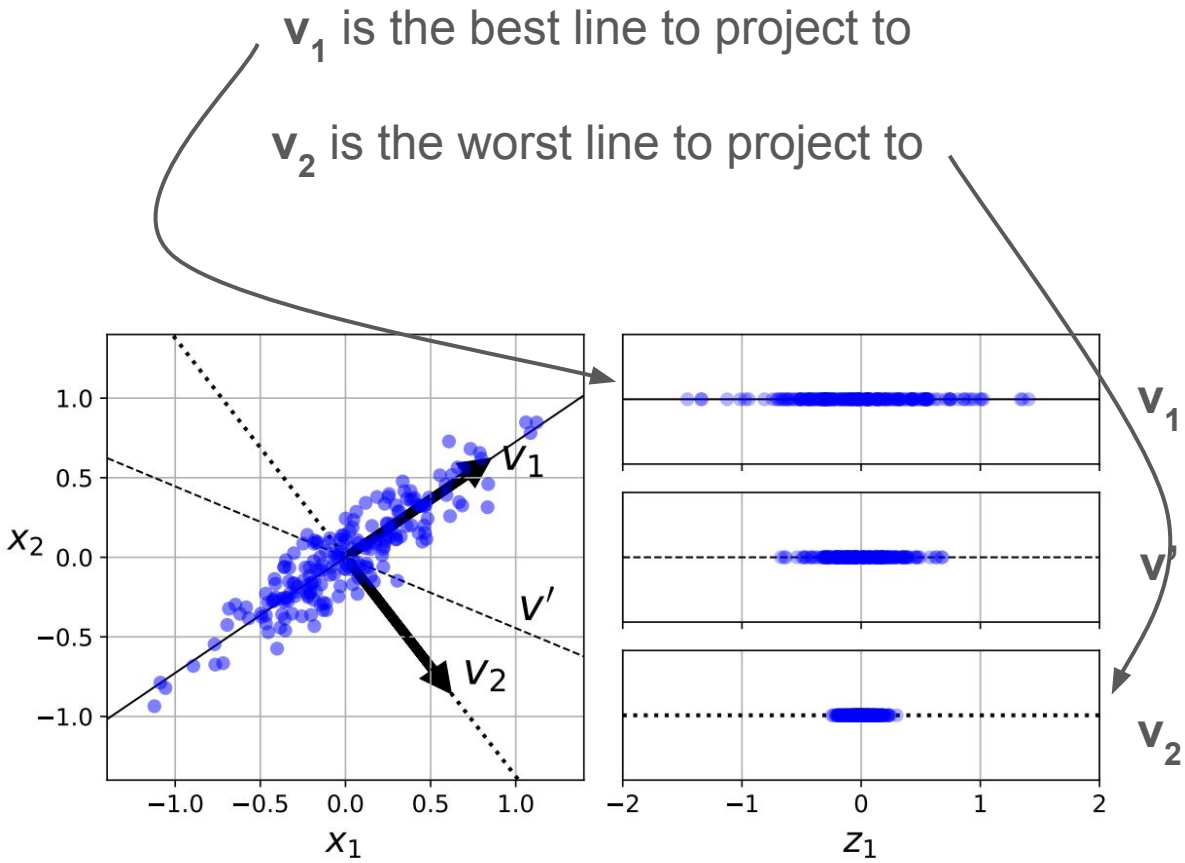


Variance maximization

One way to think about “information” in data is its spread, or variance.

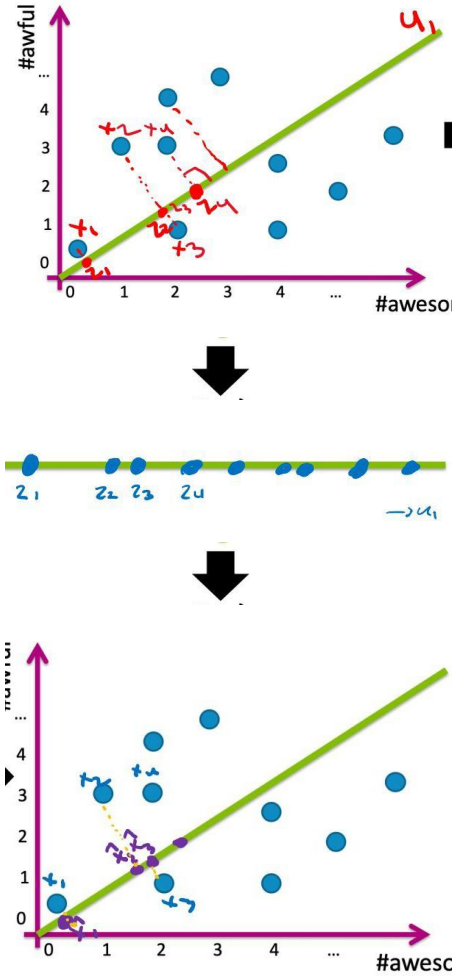
Projection data down to one single point = loss of all information.

So what is the opposite?



Principal Component Analysis

- **Idea**: Use a linear projection from d-dimensional data to k-dimensional data, where $d > k$
  - 1000-dimensional word vectors to 3-dimensional
- Choose the projection that minimizes reconstruction error
  - Intuition: The information lost if you were to ”undo” the projection can give us meaningful information
  - Choosing vectors that can capture the most **variation** (during projection)



Section Participation

What helps project high dimensional data down to lower dimensions?

A: PCA

Recall from lecture…

PCA: a high-fidelity linear projection

$$\min_{\mathbf{V}_q} \sum_{i=1}^{N} \left|\left|(x_i - \bar{x}) - \mathbf{V}_q \mathbf{V}_q^\top (x_i - \bar{x})\right|\right|_2^2$$

$\mathbf{V}_q \mathbf{V}_q^T$ is a *projection matrix* that minimizes error in basis of size q

$$\widehat{x}_i := \bar{x} + \mathbf{V}_q \mathbf{V}_q^\top (x_i - \bar{x}) = \bar{x} + \sum_{j=1}^{q} v_j v_j^\top (x_i - \bar{x})$$

Case when $q = 1$

$$v_1 = \arg \min_{v:\|v\|_2=1} \sum_{i=1}^{N} \|(x_i - \bar{x}) - vv^\top (x_i - \bar{x})\|_2^2$$

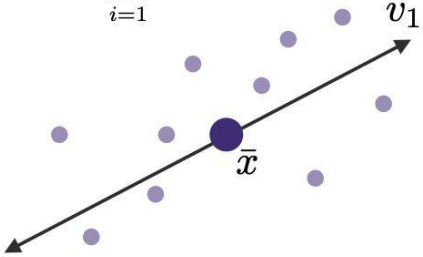
$$= \arg \min_{v:\|v\|_2=1} \sum_{i=1}^{N} \|x_i - \bar{x}\|_2^2 - 2(x_i - \bar{x})^\top vv^\top (x_i - \bar{x}) + (x_i - \bar{x})^\top vv^\top vv^\top (x_i - \bar{x})$$

$$= \arg \min_{v:\|v\|_2=1} \sum_{i=1}^{N} \|x_i - \bar{x}\|_2^2 - \sum_{i=1}^{N} (x_i - \bar{x})^\top vv^\top (x_i - \bar{x})$$

$$= \arg \max_{v:\|v\|_2=1} \sum_{i=1}^{N} (x_i - \bar{x})^\top vv^\top (x_i - \bar{x})$$

$$= \arg \max_{v:\|v\|_2=1} v^\top \Sigma v$$

$$\Sigma := \sum_{i=1}^{N} (x_i - \bar{x})(x_i - \bar{x})^T$$

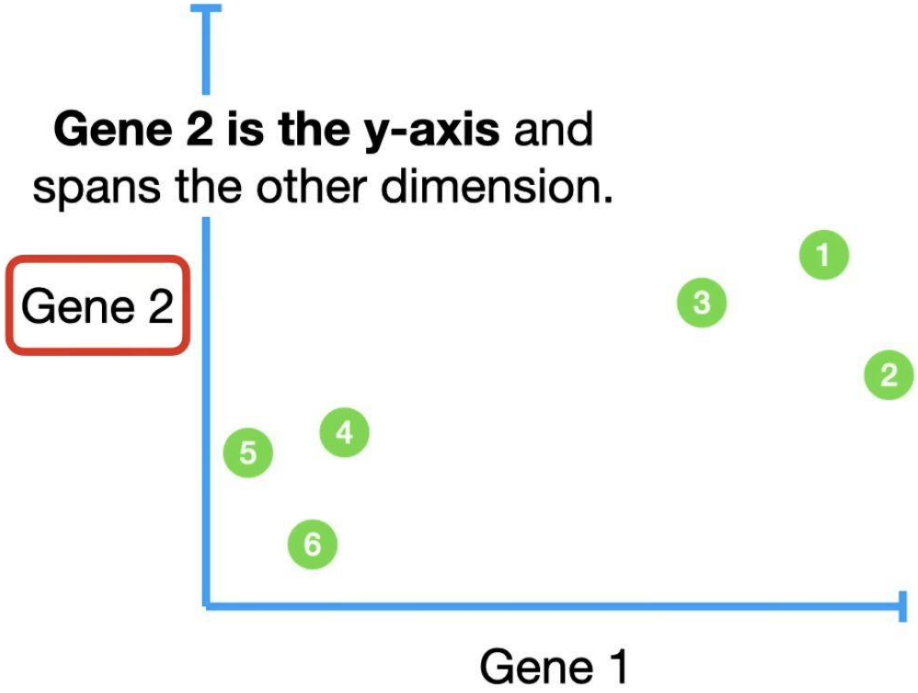


Let’s do some PCA Intuition Building

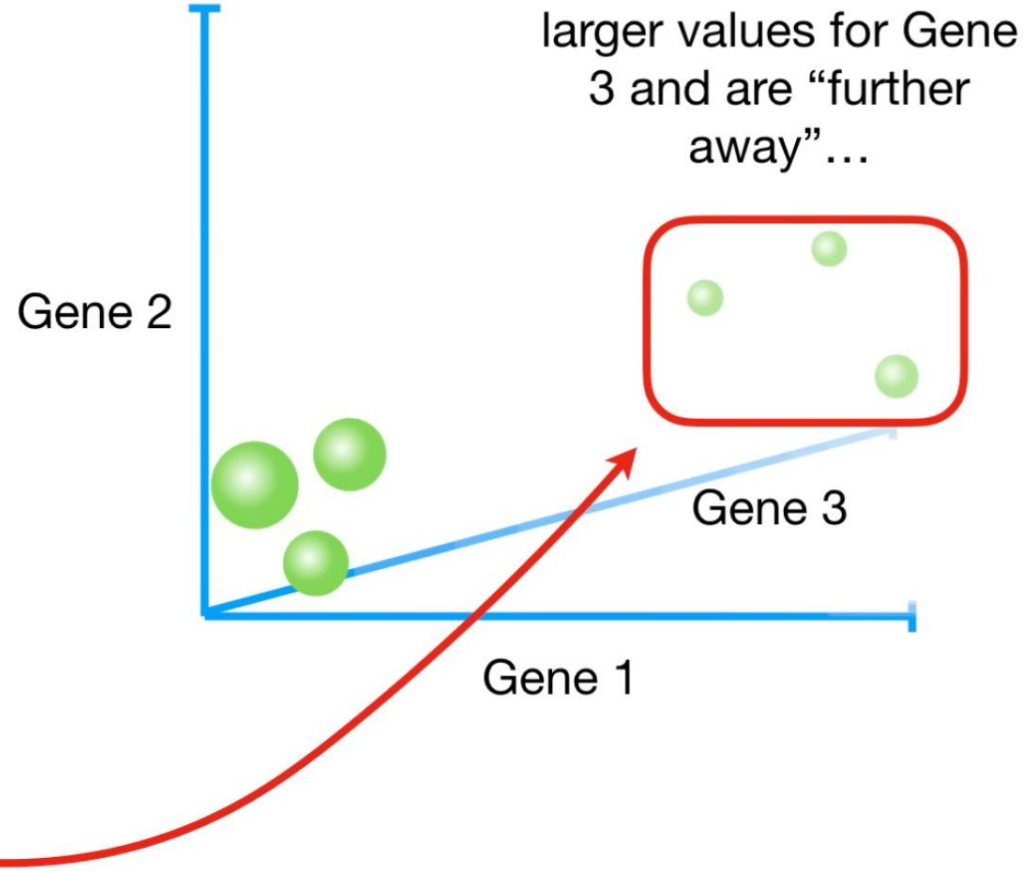
	Mouse 1	Mouse 2	Mouse 3	Mouse 4	Mouse 5	Mouse 6
Gene 1	10	11	8	3	1	2

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	1	2
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



Understanding the decomposition

$$\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top = \sum_{i=1}^{r} u_i \sigma_i v_i^\top$$

Rank-r approximation

Rank-2 approximation

Rank-1 approximation

$$\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top = u_1\sigma_1 v_1^\top + u_2\sigma_2 v_2^\top + \ldots + u_r\sigma_r v_r^\top$$

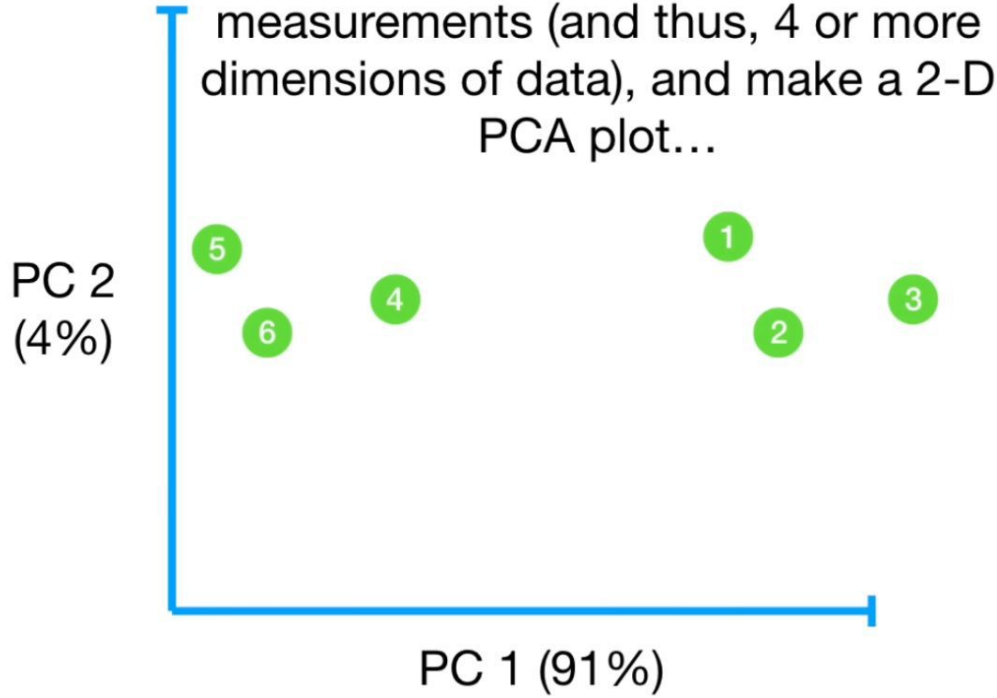
	Mouse 1	Mouse 2	Mouse 3	Mouse 4	Mouse 5	Mouse 6
Gene 1	10	11	8	3	2	1
Gene 2	6	4	5	3	2.8	1
Gene 3	12	9	10	2.5	1.3	2
Gene 4	5	7	6	2	4	7

	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.

	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

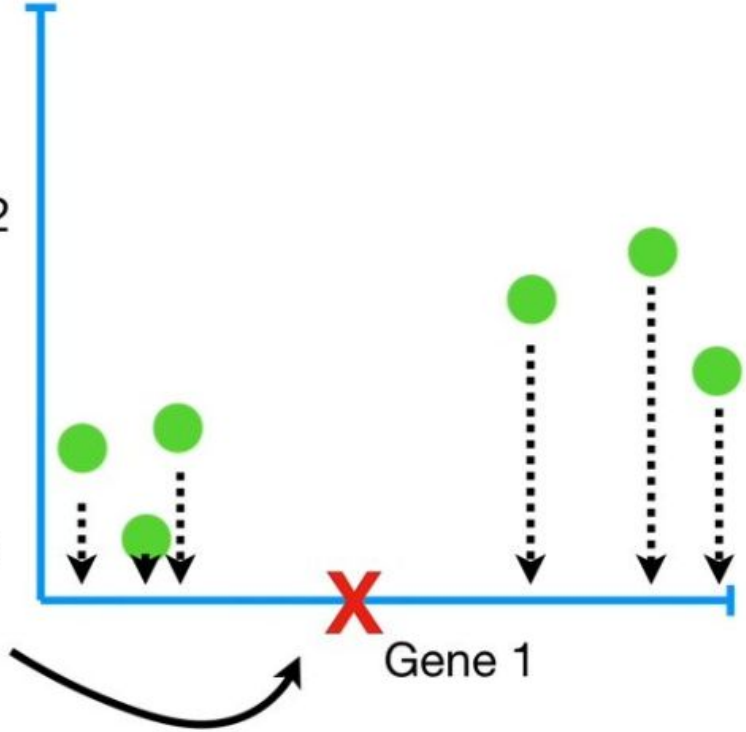
So we're going to talk about how PCA can take 4 or more gene measurements (and thus, 4 or more dimensions of data), and make a 2-D PCA plot…



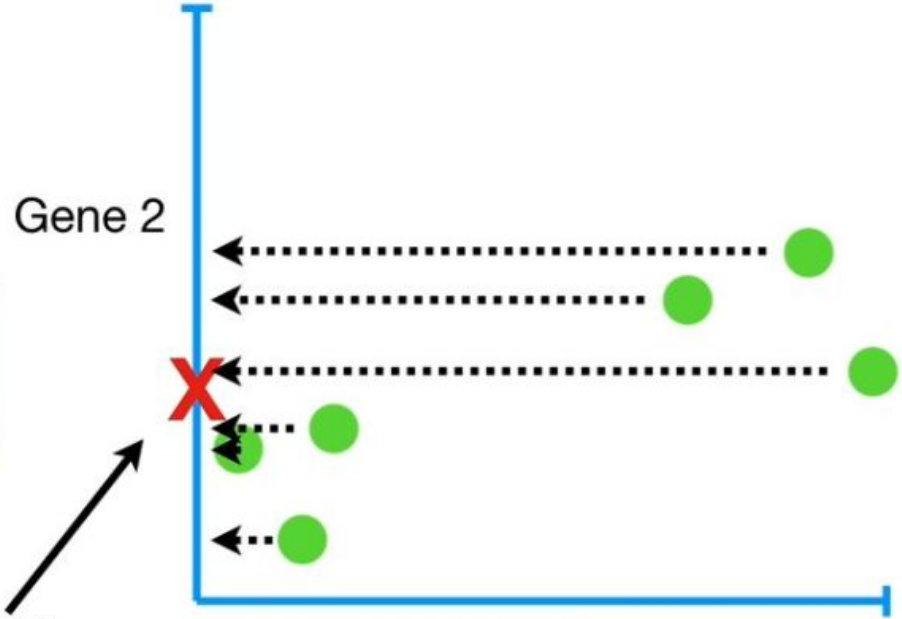
Step 1: Center the data

	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

Then we'll calculate the average measurement for Gene 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

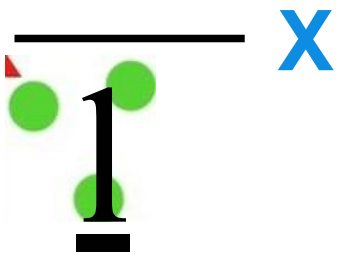


…and the average measurement for Gene 2.

	Mouse 1	Mouse 2	Mouse 3	Mouse 4	Mouse 5	Mouse 6
Gene 1	10	11	8	3	2	1
Gene2		4	5	3	2.8	1

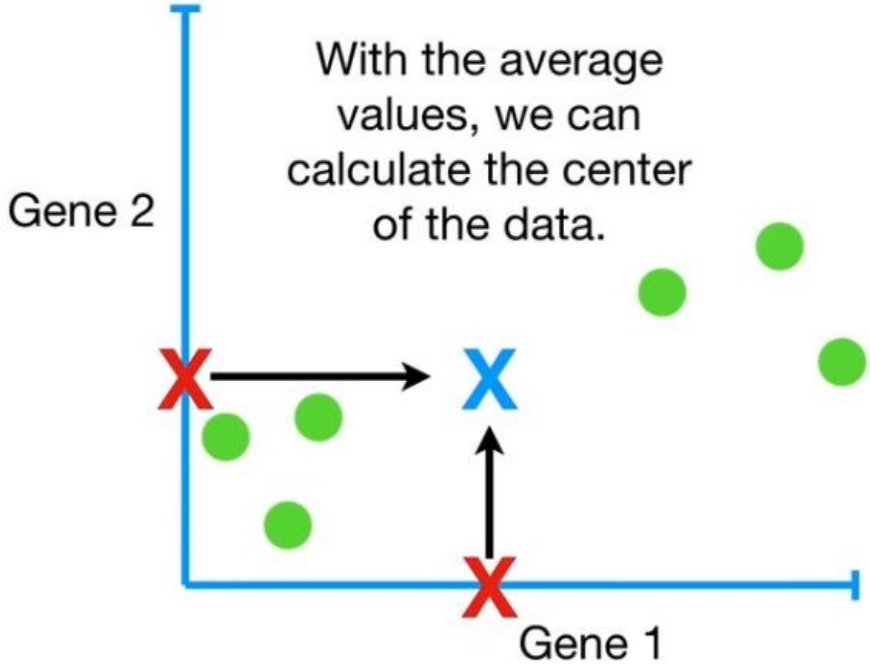
Gene2

With the average values, we can calculate the center of the data.

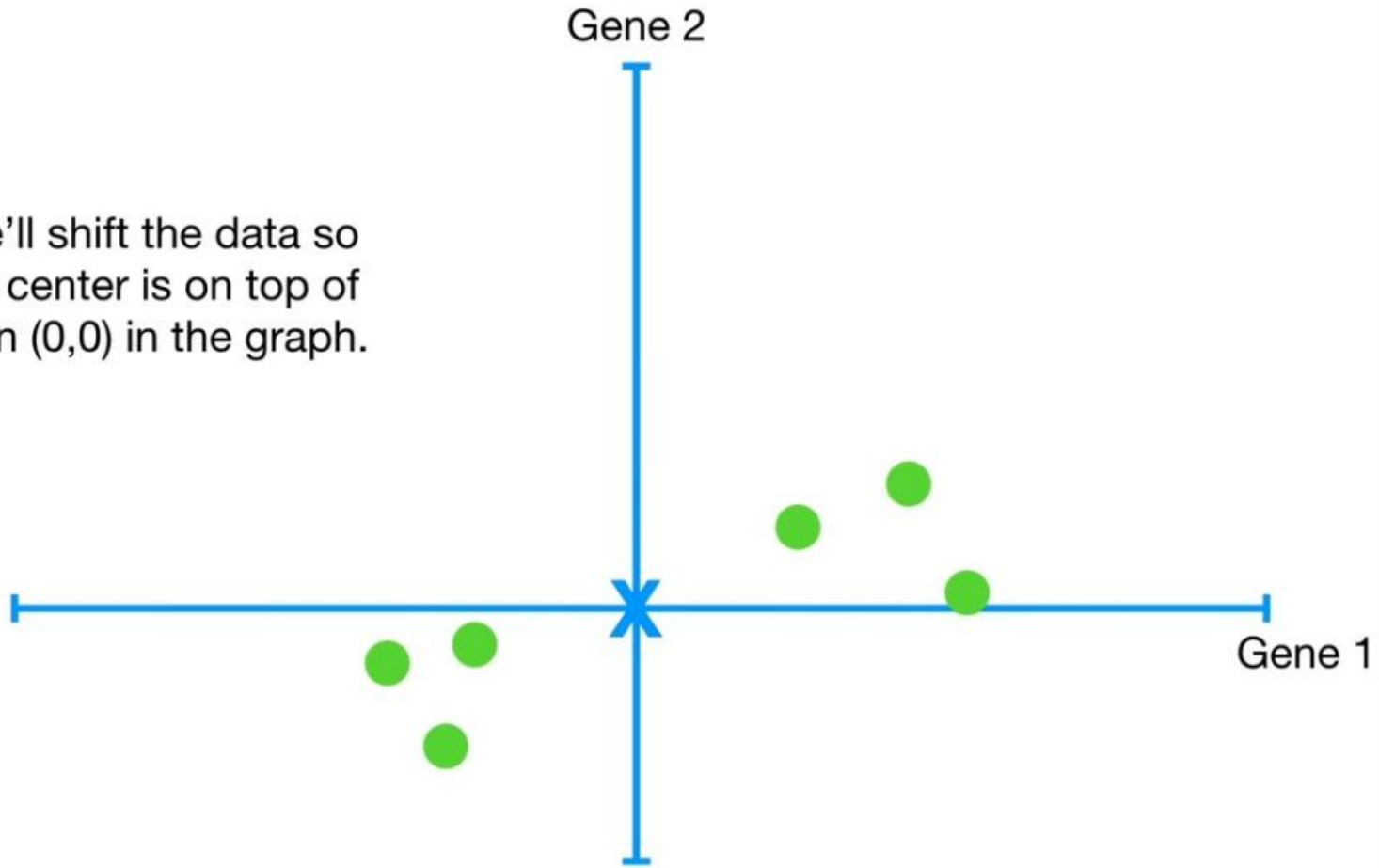


Gene 1

	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

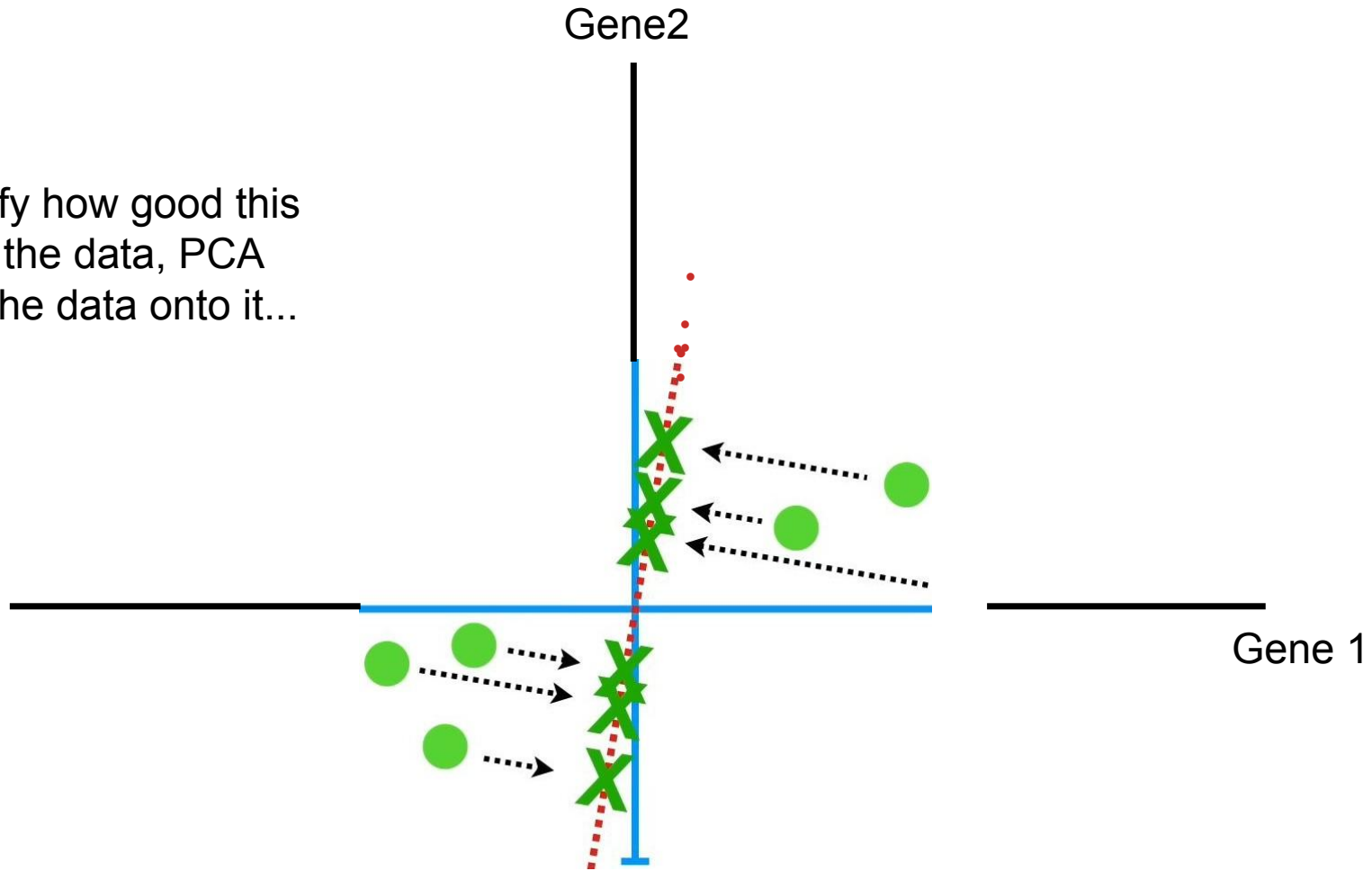


Now we'll shift the data so that the center is on top of the origin (0,0) in the graph.

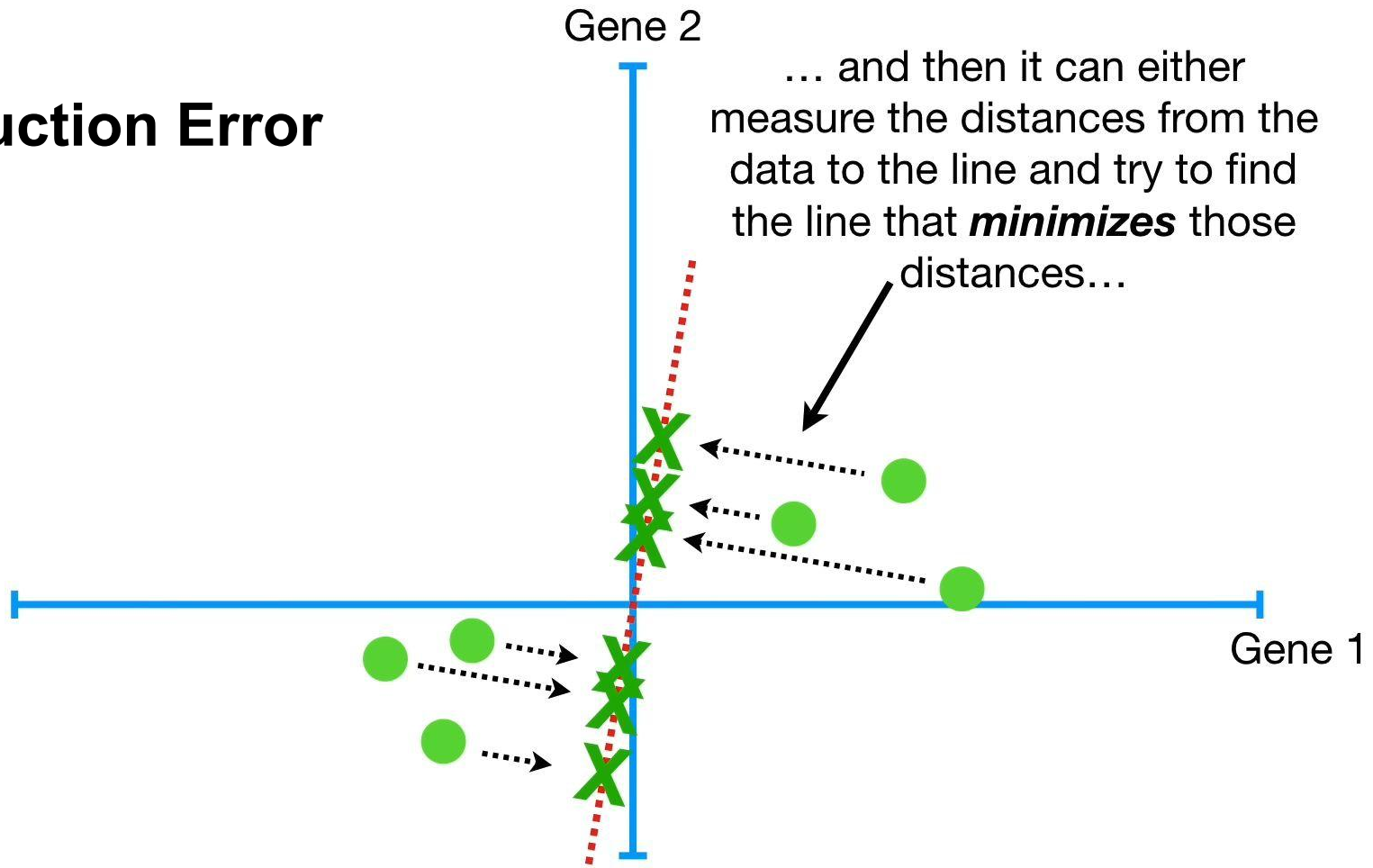


Step 2: Find Principal Components

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

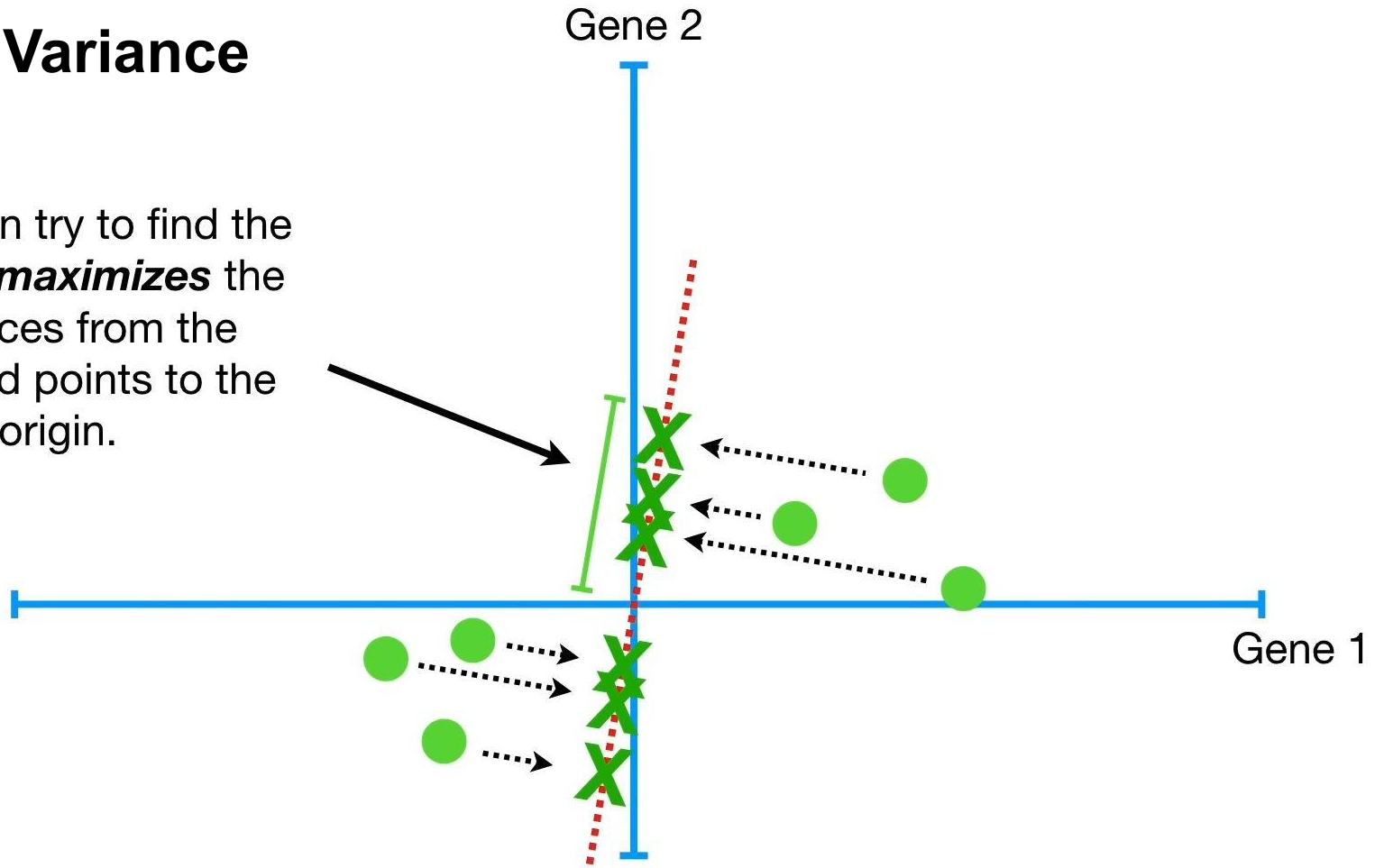


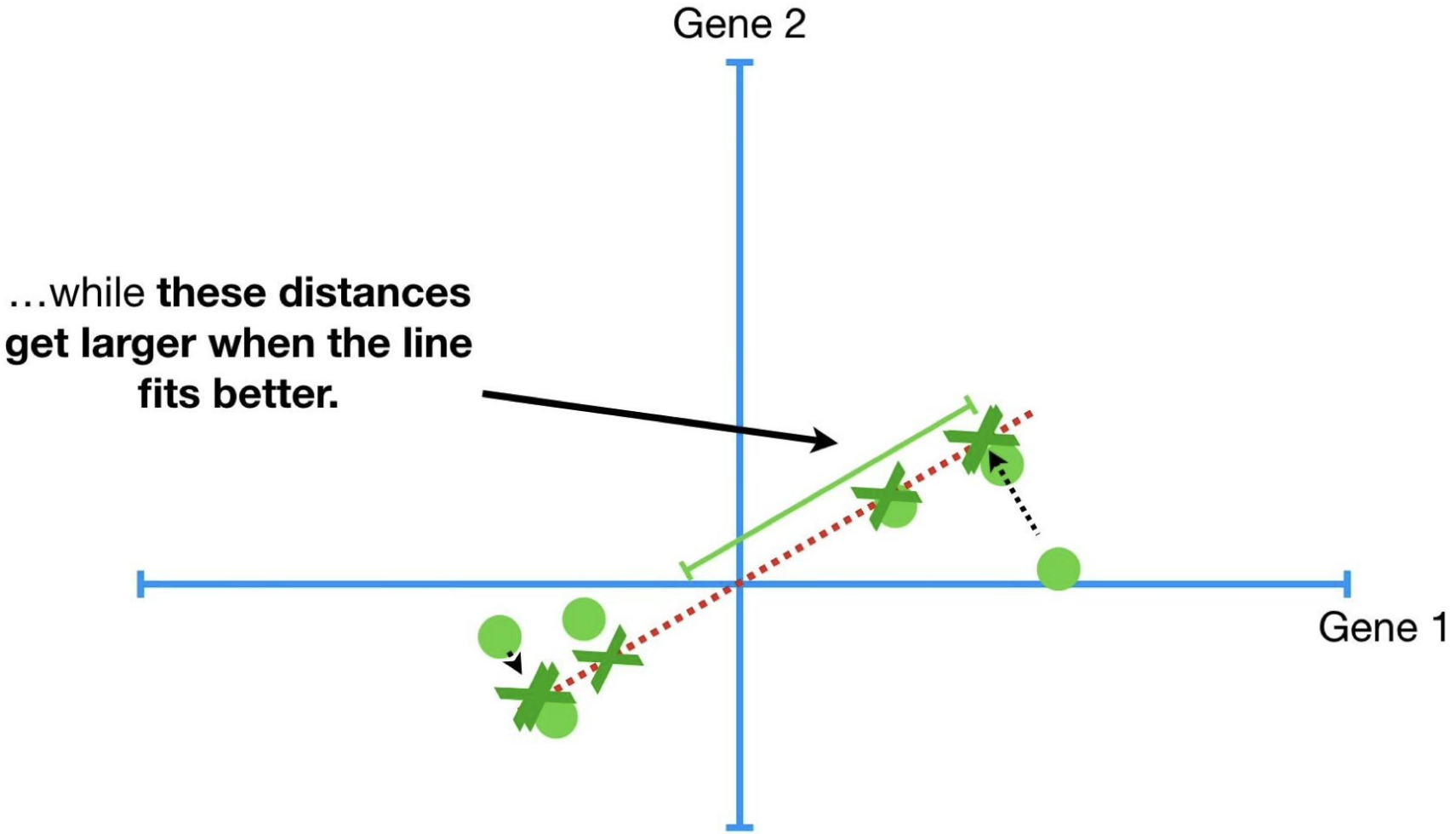
Minimize Reconstruction Error



Maximize Variance

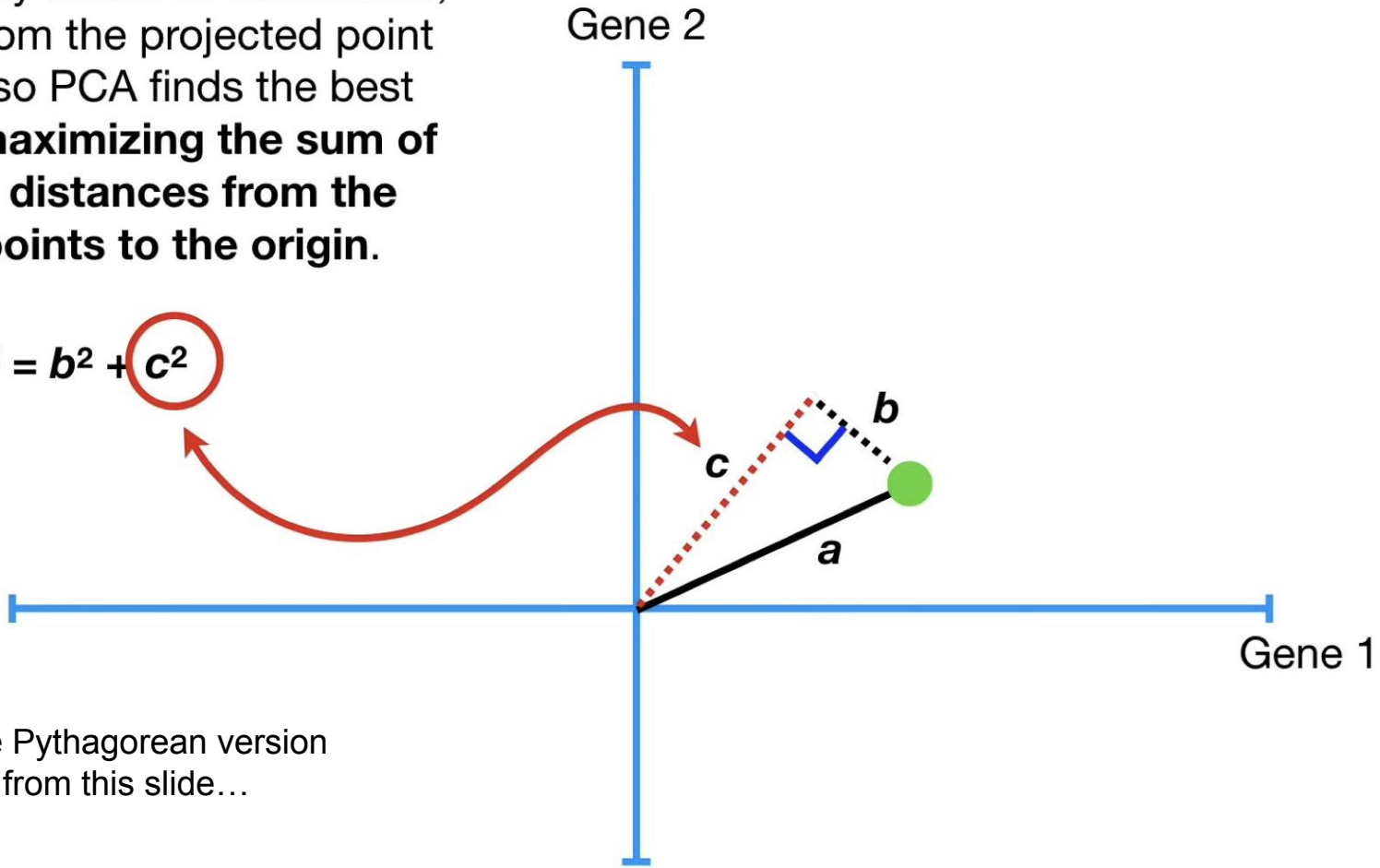
…or it can try to find the line that ***maximizes*** the distances from the projected points to the origin.




Gene 2
…while these distances get larger when the line fits better.
Gene 1

…but it's actually easier to calculate ***c***, the distance from the projected point to the origin, so PCA finds the best fitting line by **maximizing the sum of the squared distances from the projected points to the origin**.

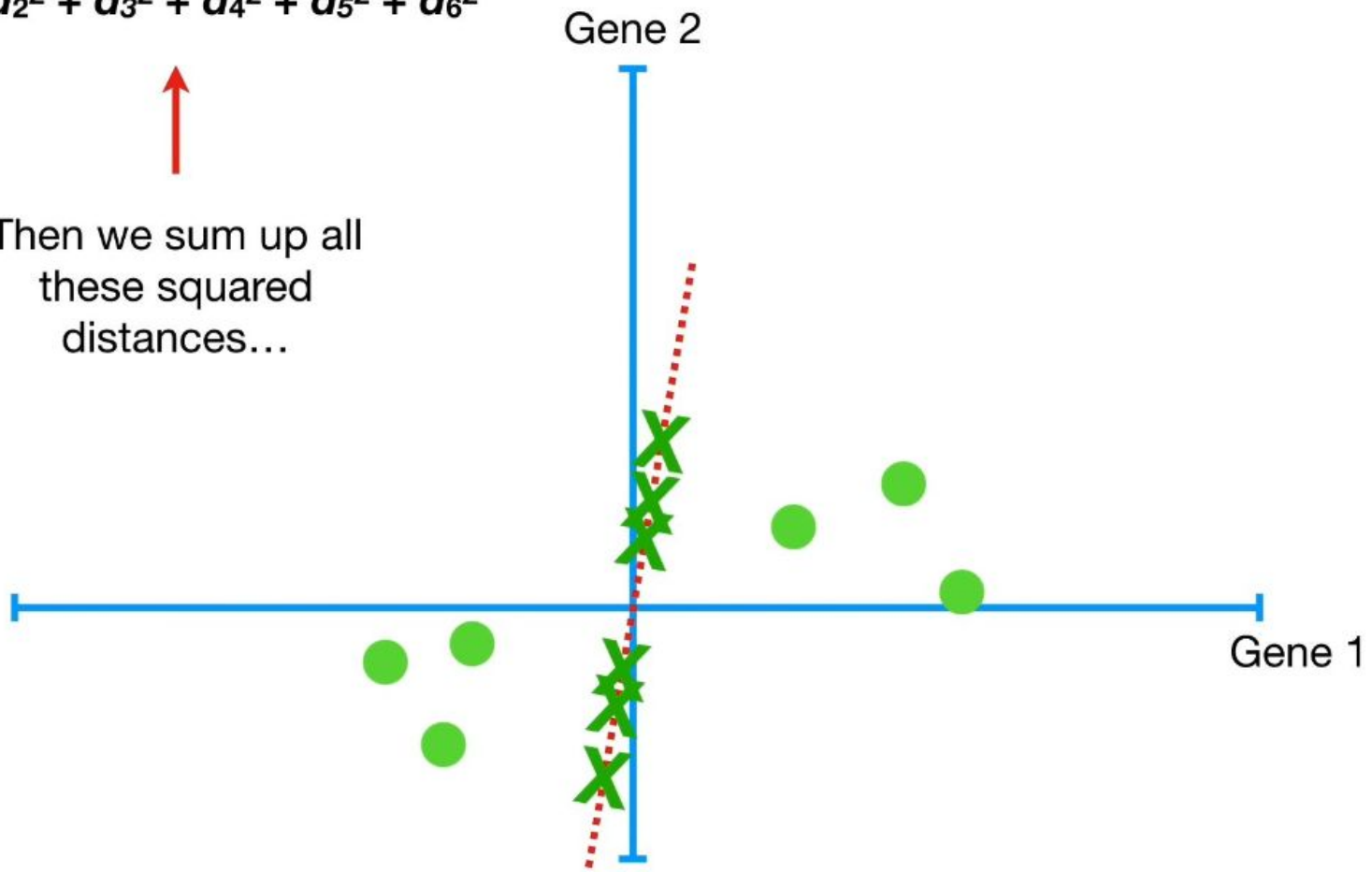
$a^2 = b^2 + c^2$



This is just the Pythagorean version of the algebra from this slide…

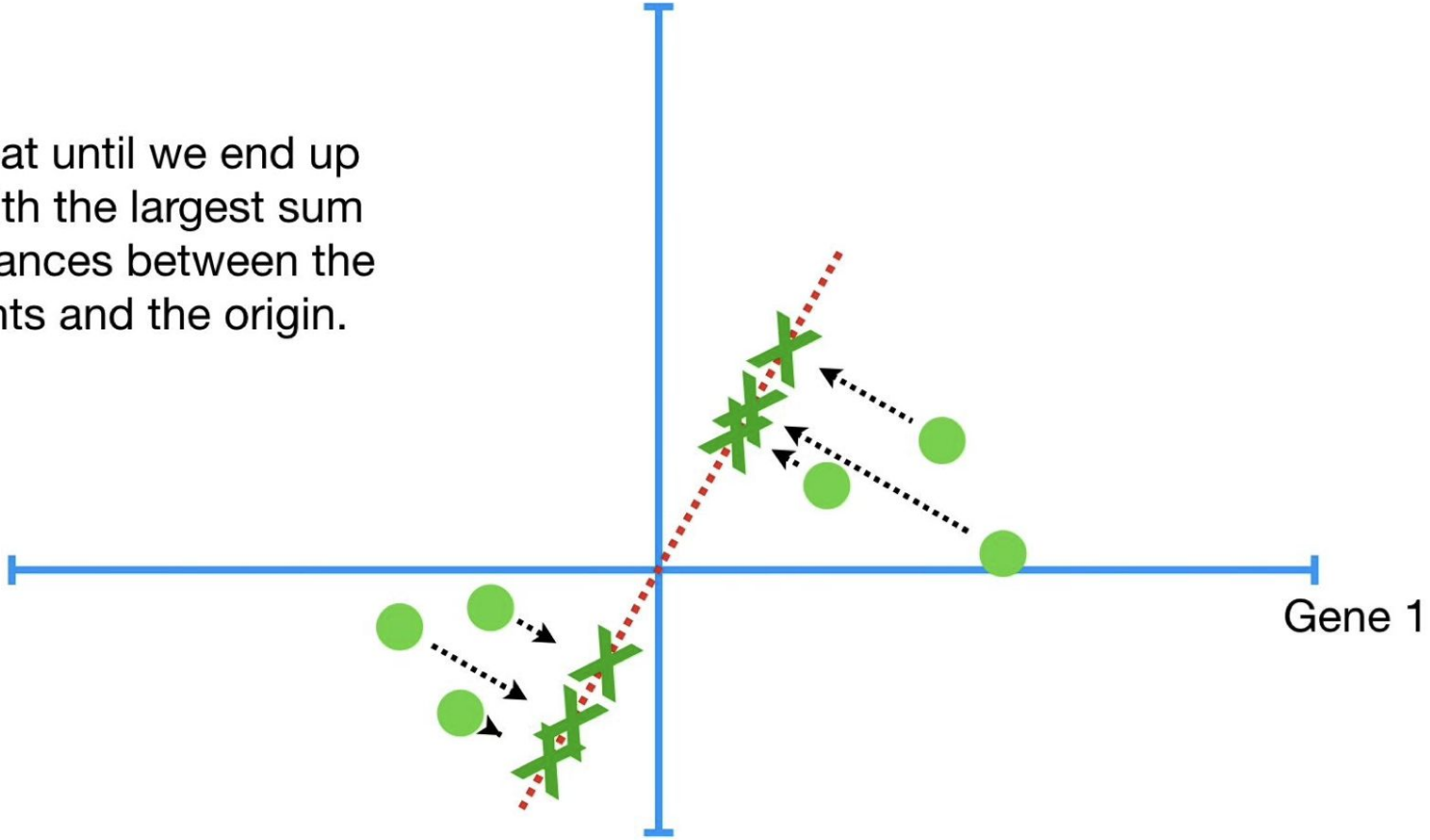
$d_1^2 + d_2^2 + d_3^2 + d_4^2 + d_5^2 + d_6^2$

Then we sum up all these squared distances…



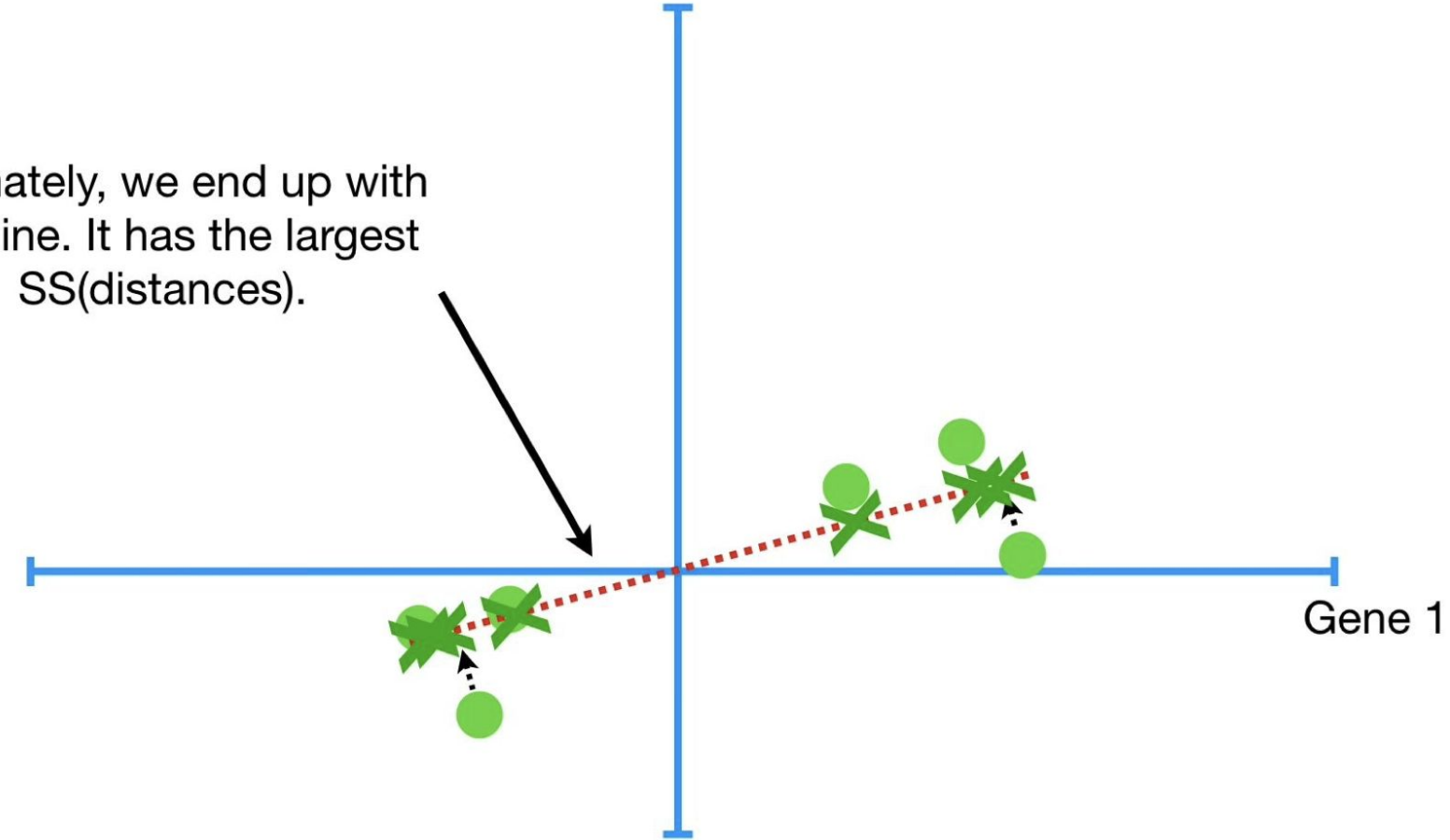
$d_1^2 + d_2^2 + d_3^2 + d_4^2 + d_5^2 + d_6^2$ = sum of squared distances = SS(distances)

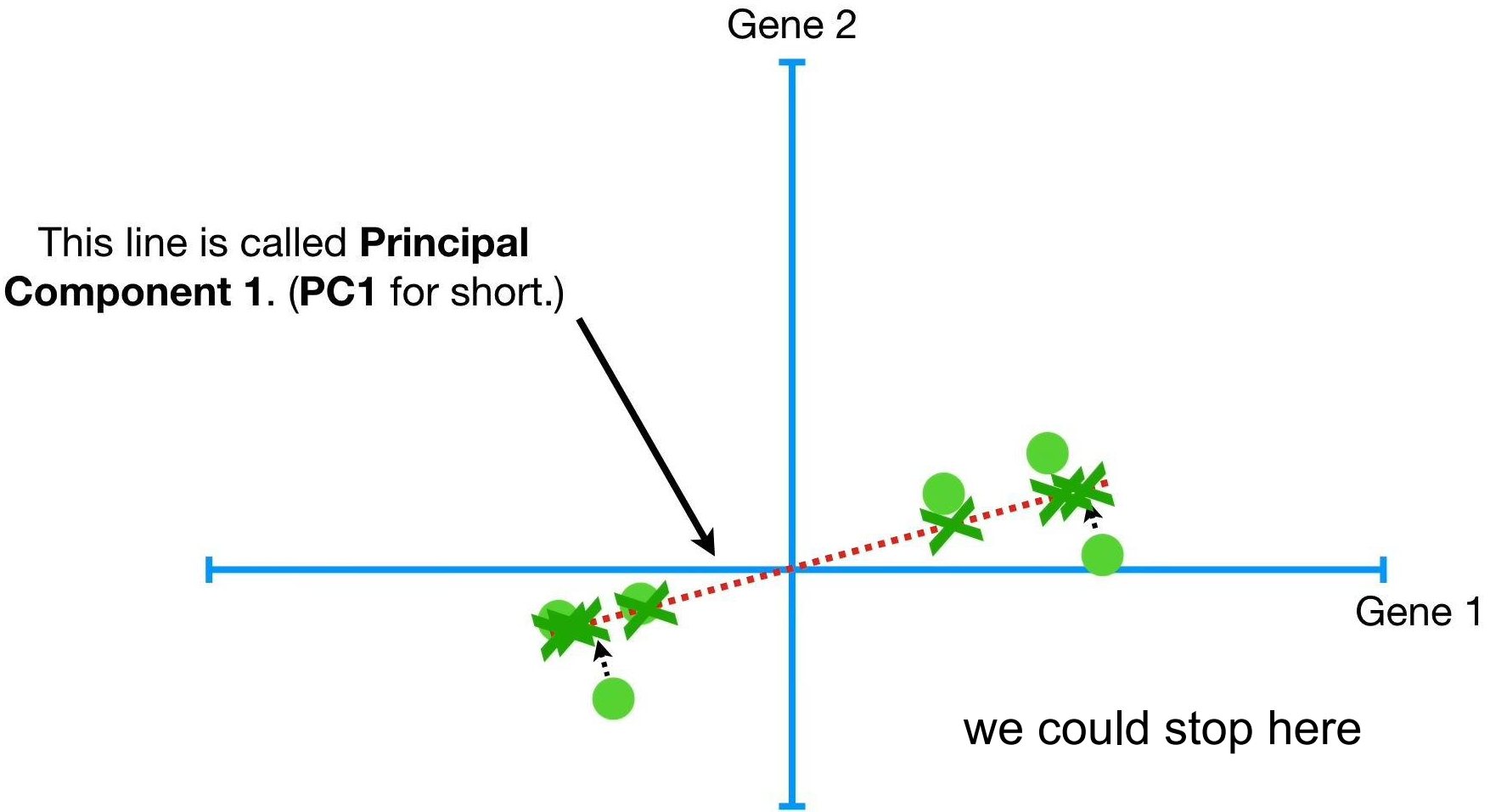
…and we repeat until we end up with the line with the largest sum of squared distances between the projected points and the origin.



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

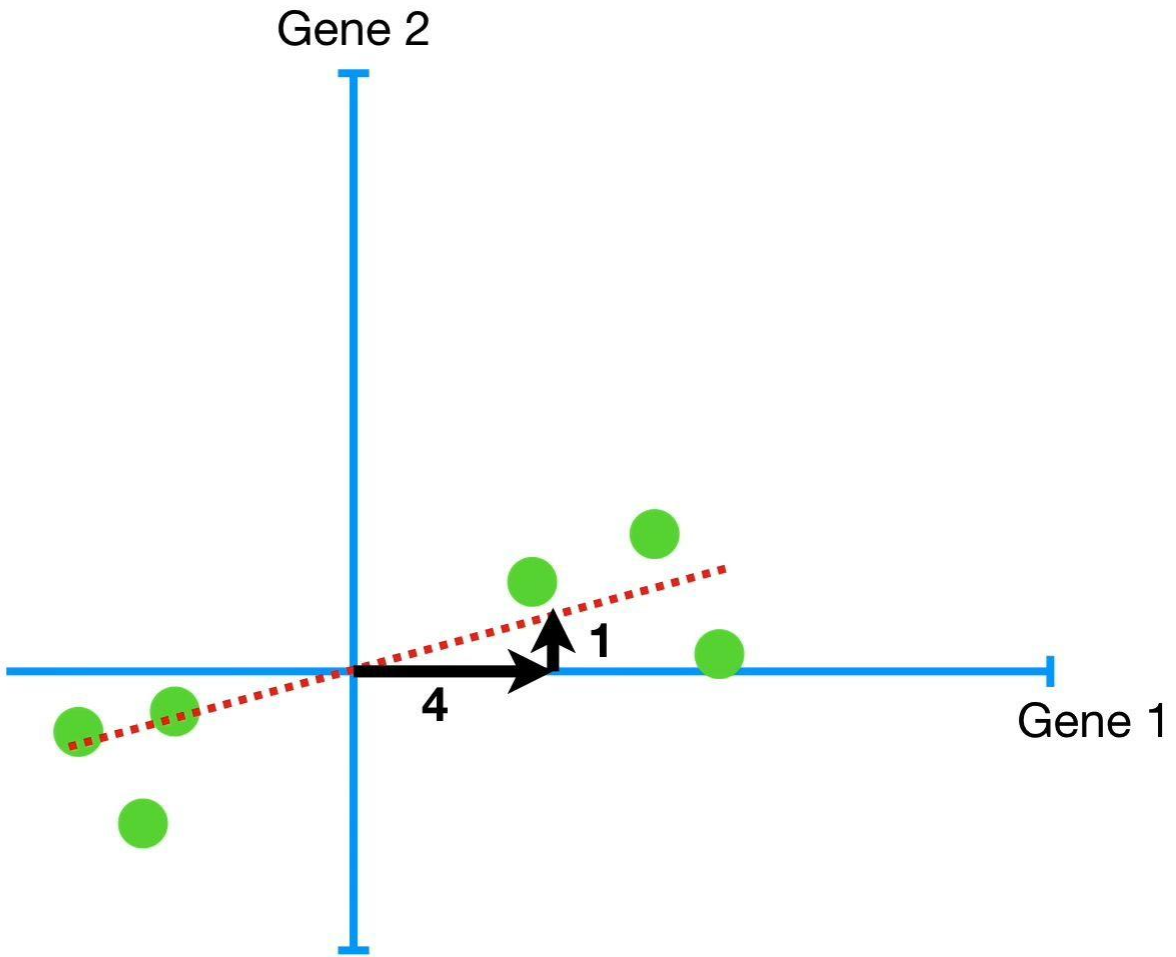
Ultimately, we end up with this line. It has the largest SS(distances).

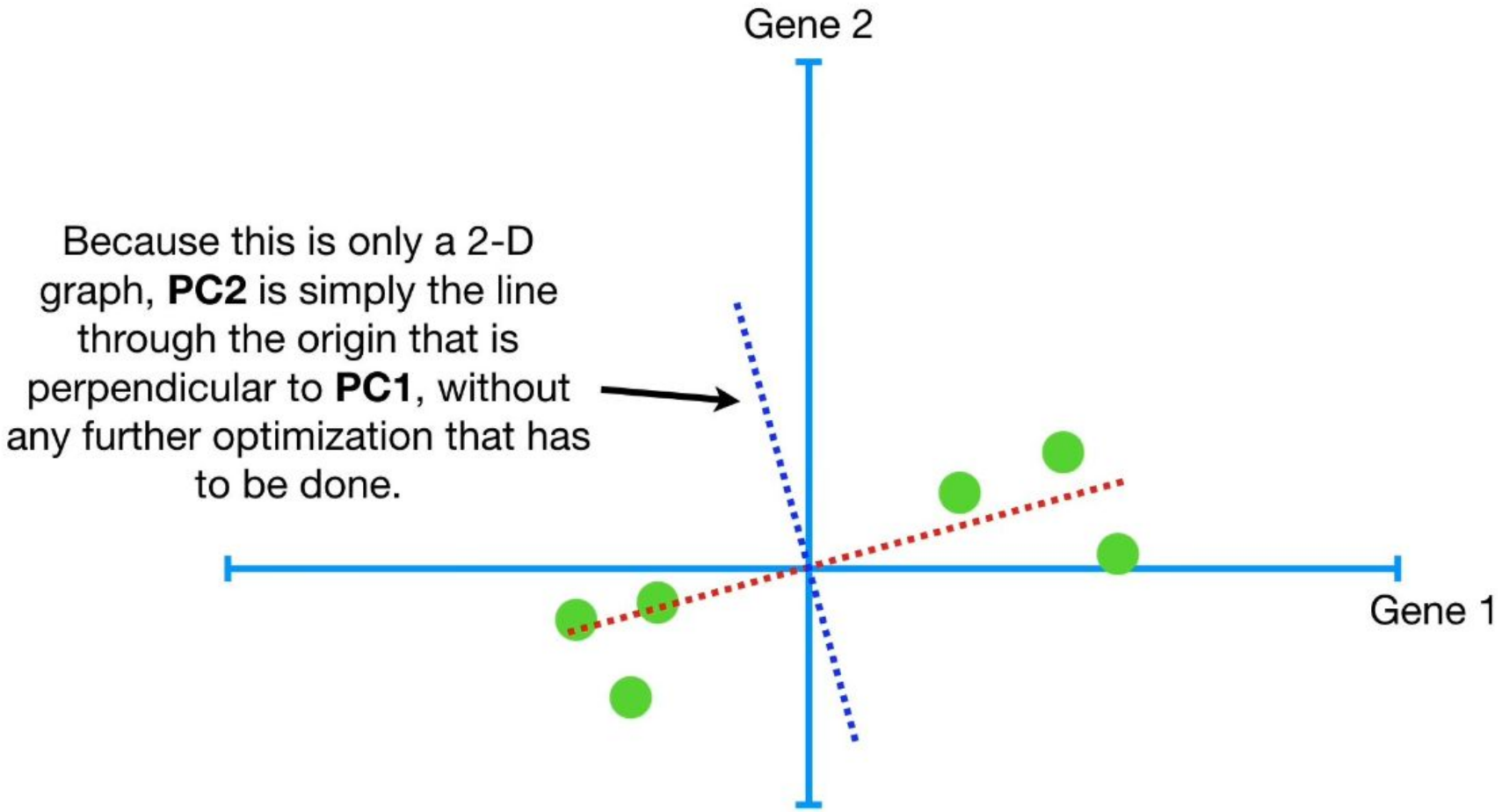


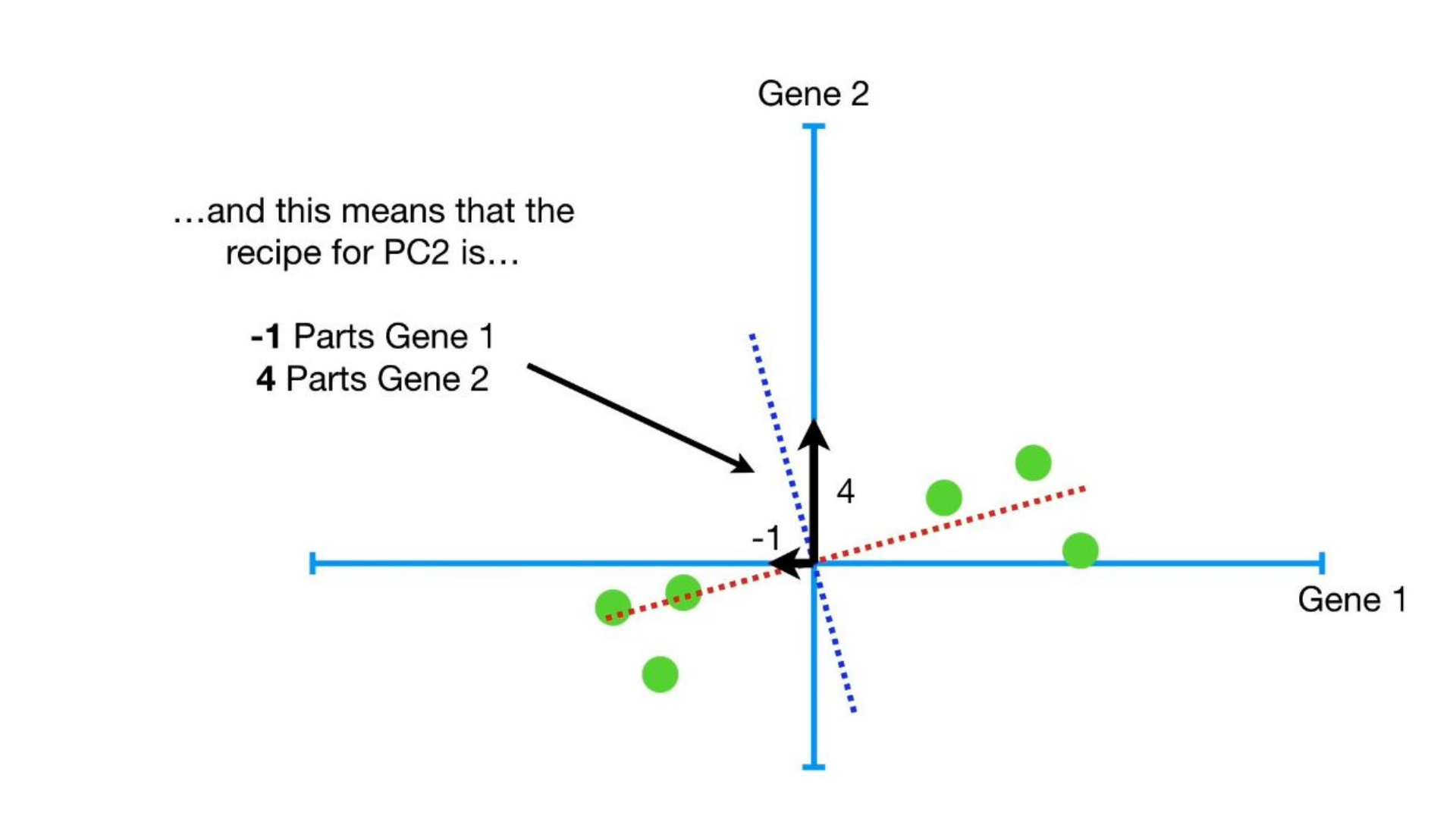

Gene 2
This line is called **Principal Component 1**. (**PC1** for short.)
Gene 1
we could stop here

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

The ratio of Gene 1 to Gene 2 tells you that Gene 1 is more important when it comes to describing how the data are spread out..

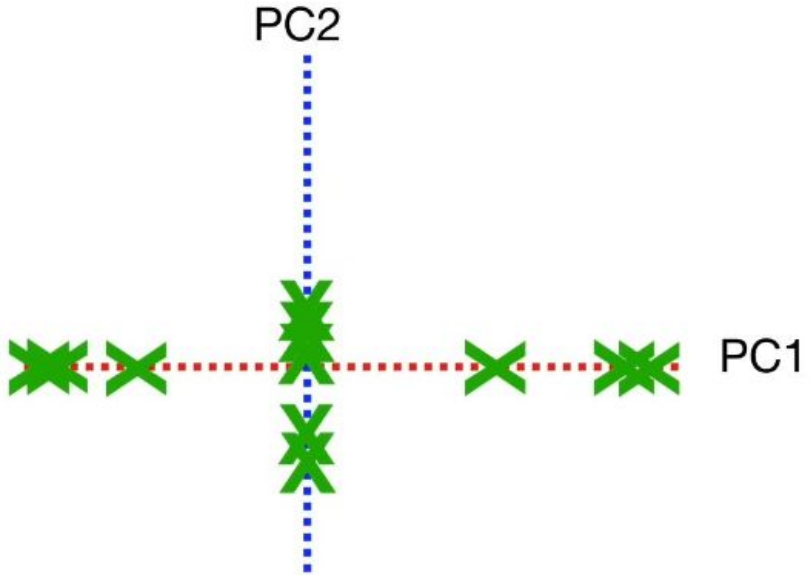


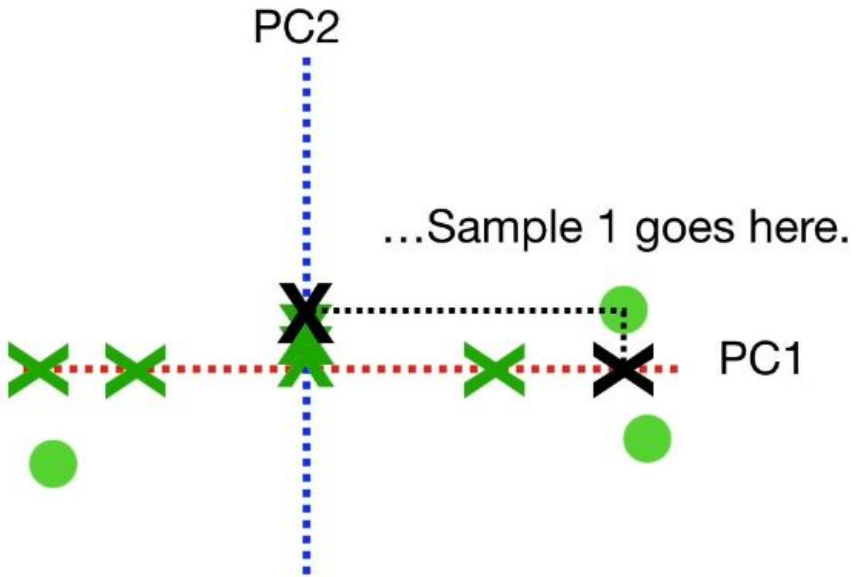

Gene 2
Because this is only a 2-D graph, **PC2** is simply the line through the origin that is perpendicular to **PC1**, without any further optimization that has to be done.
Gene 1


Gene 2
…and this means that the recipe for PC2 is…
-1 Parts Gene 1
4 Parts Gene 2
4
-1
Gene 1

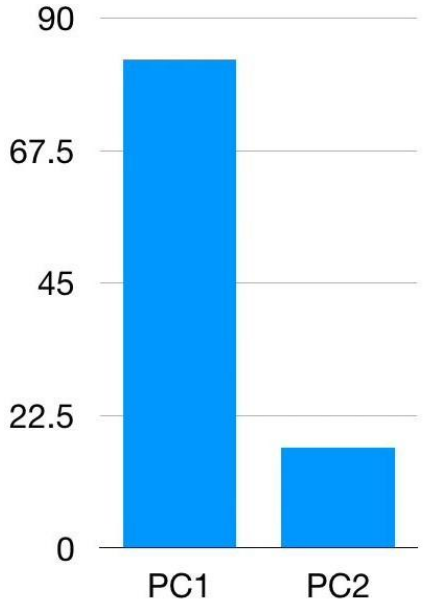
Step 3: Change of Basis

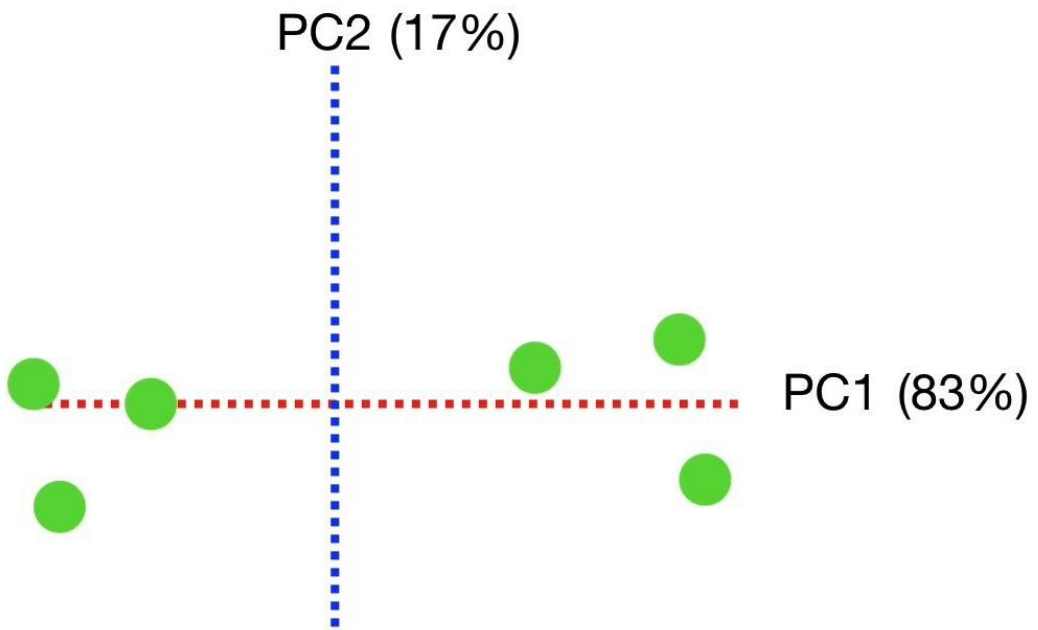
…then we use the projected points to find where the samples go in the PCA plot.




PC2
…Sample 1 goes here.
PC1

Explained Variance

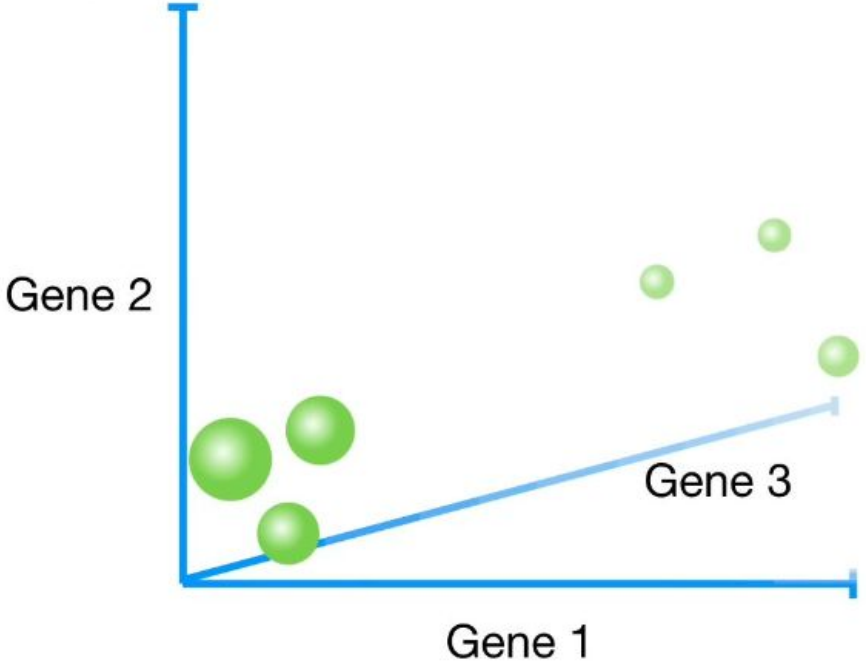




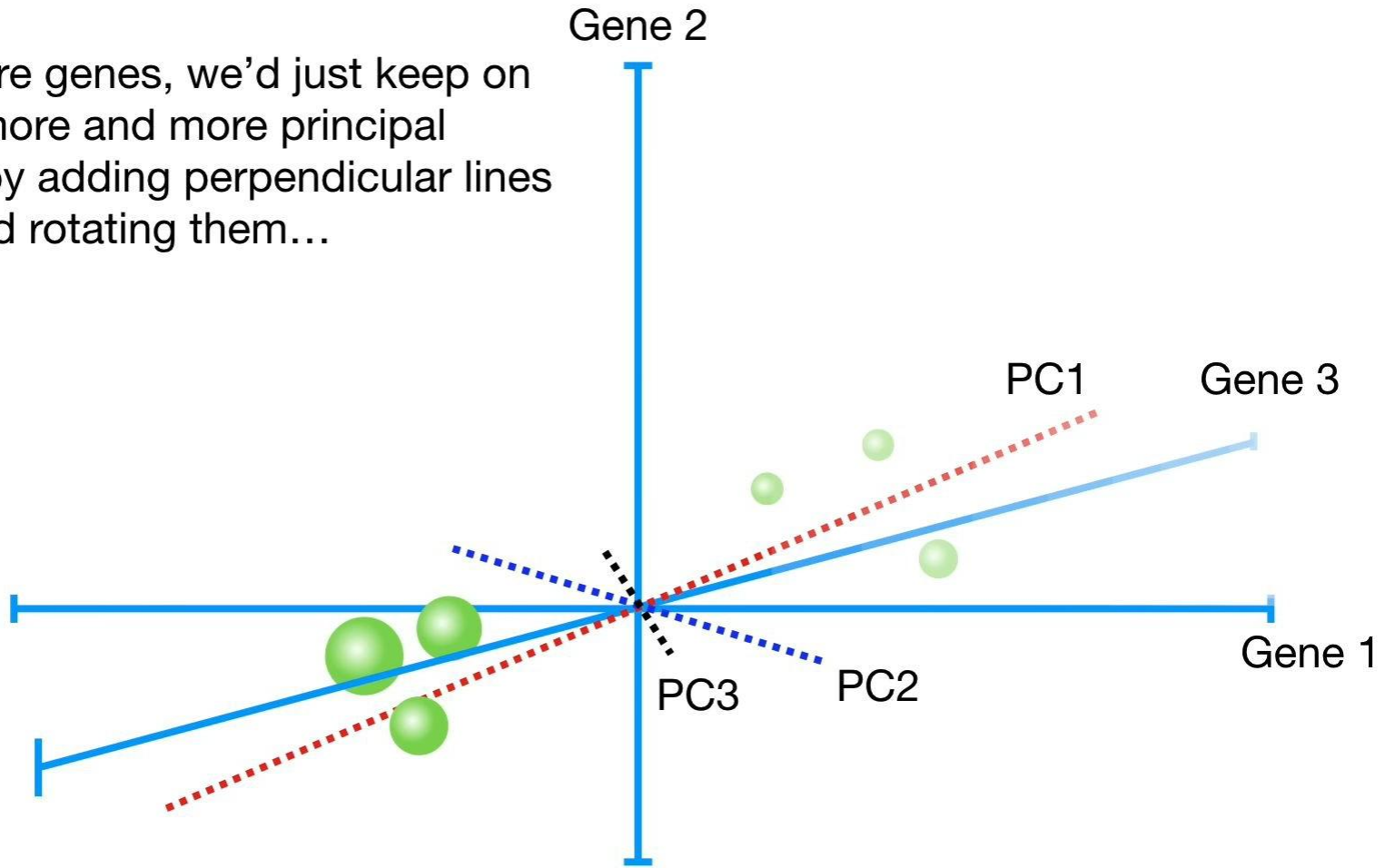
Example w 3 Genes
→ 3 Dimensions

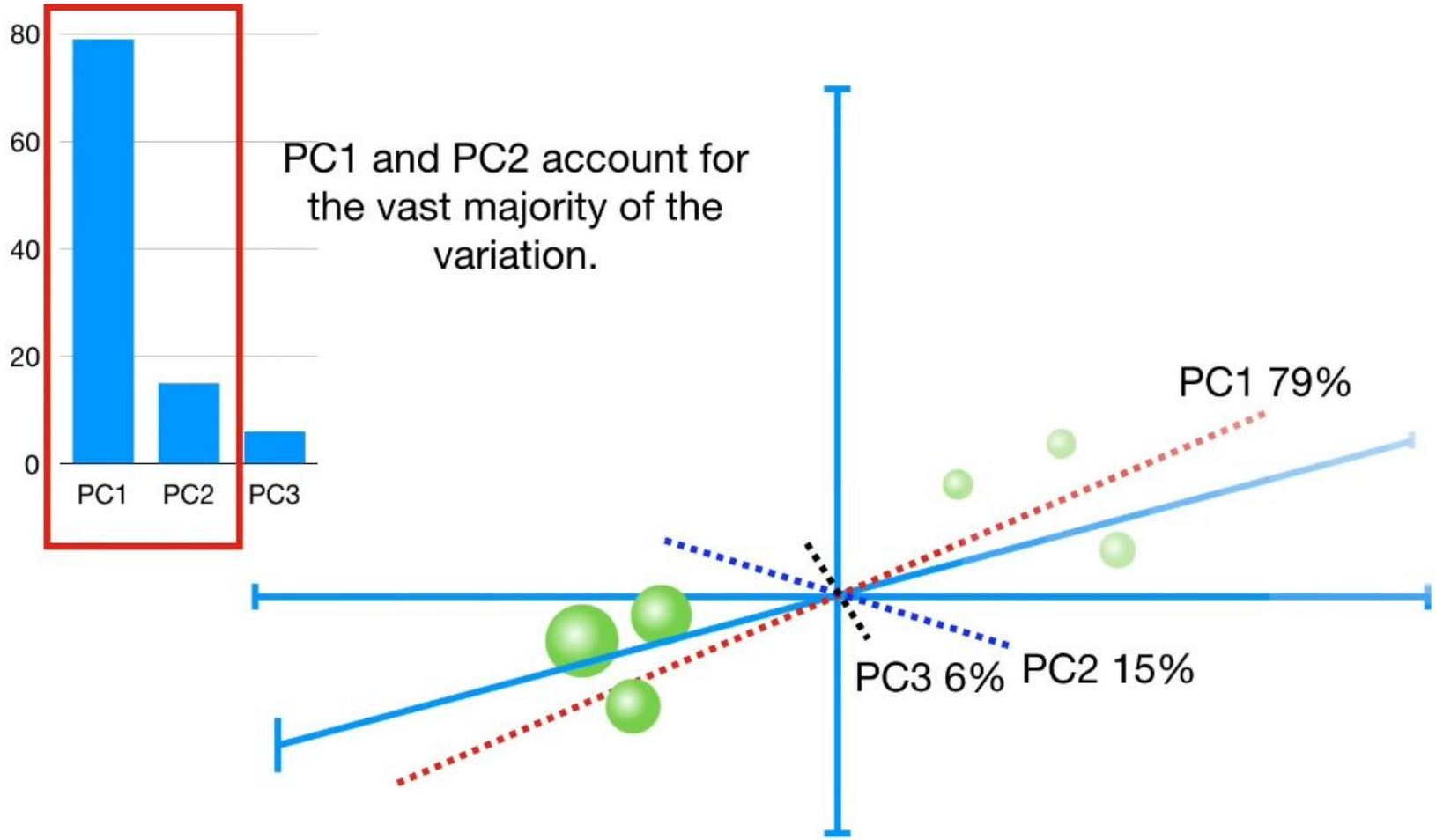
PCA with 3 variables (in this case, that means 3 genes) is pretty much the same as 2 variables…

	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

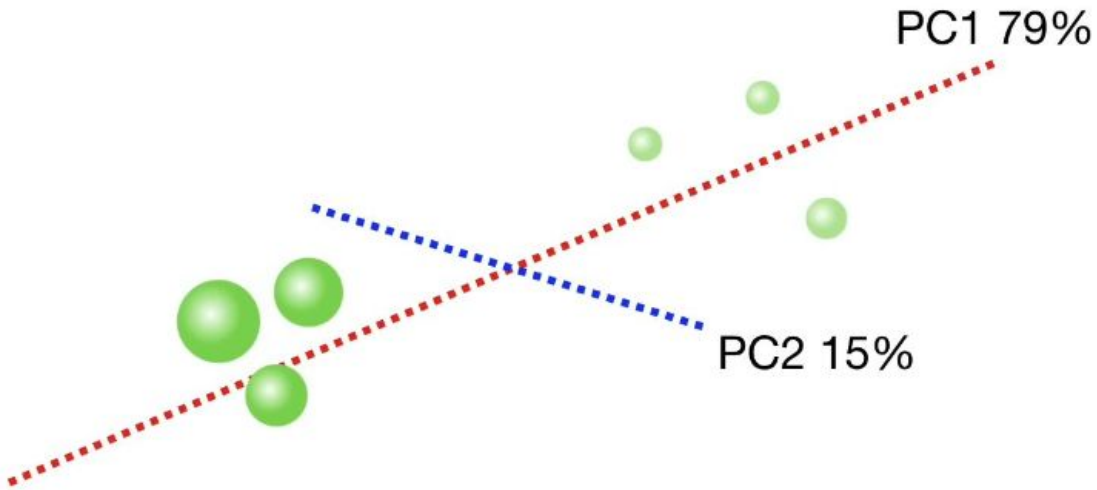


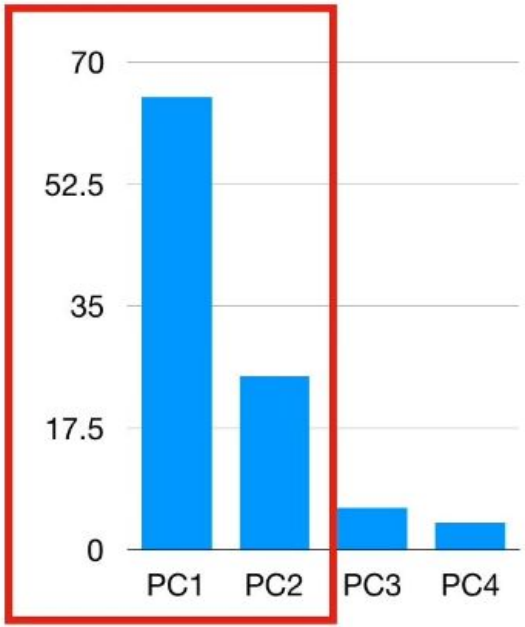
If we had more genes, we'd just keep on finding more and more principal components by adding perpendicular lines and rotating them…



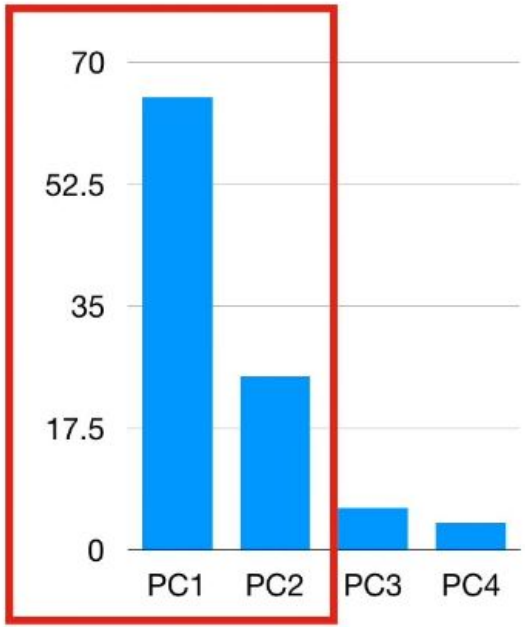

80
60
40
20
0
PC1
PC2
PC3
PC1 and PC2 account for the vast majority of the variation.
PC1 79%
PC3 6%
PC2 15%

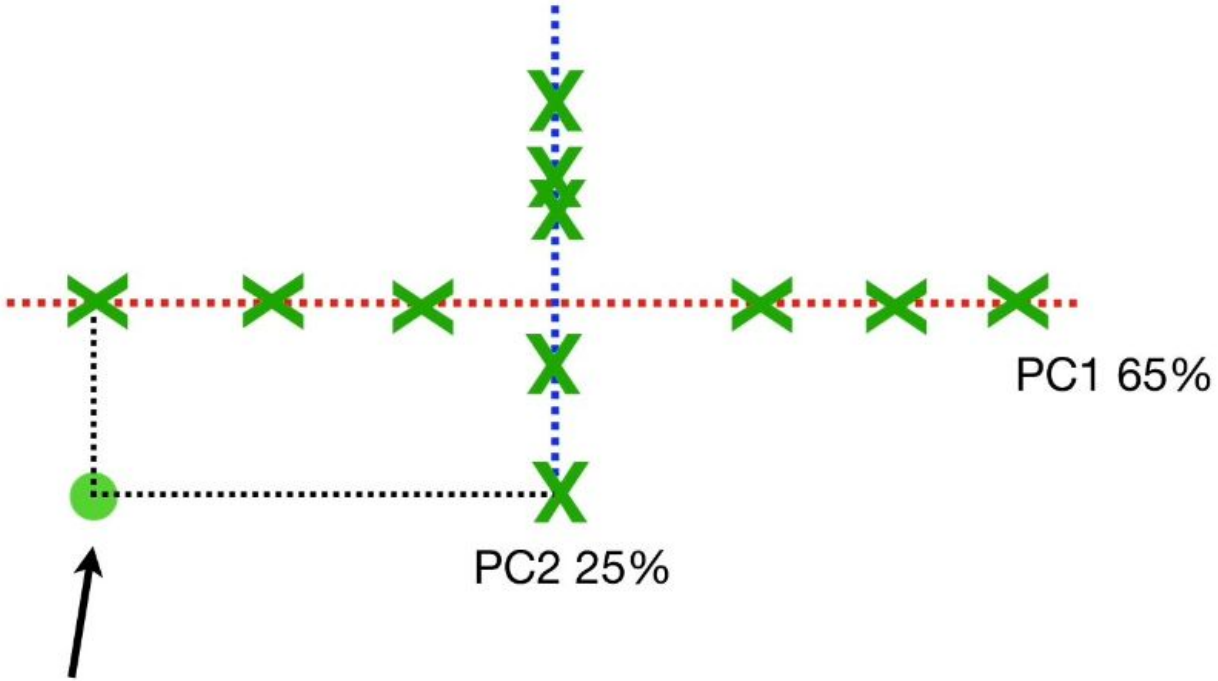
To convert the 3-D graph into a 2-D PCA graph, we just strip away everything but the data and PC1 and PC2…





…in this case, PC1 and PC2 account for 90% of the variation, so we can just use those to draw a 2-dimensional PCA graph.





So Sample 2 goes here.

Our goal is to find a projection matrix $\mathbf{V}_q \in \mathbb{R}^{d \times q}$ that minimizes the reconstruction error of our demeaned data. Mathematically, we want to find:

$$\min_{\mathbf{V}_q} \sum_{i=1}^{N} ||(x_i - \bar{x}) - \mathbf{V}_q \mathbf{V}_q^\top (x_i - \bar{x})||_2^2$$

Turns out, the $\mathbf{V}_q$ that minimizes this equation is **the first q eigenvectors of $\tilde{\mathbf{X}}^\top \tilde{\mathbf{X}}$**. There is a proof in the lecture notes, but here is the inuition:

Relating PCA & SVD

From lecture notes:

Theorem [SVD]: Let $A \in \mathbb{R}^{m \times n}$ with rank $r \leq \min\{m, n\}$. Then $A = USV^T$ where $S \in \mathbb{R}^{r \times r}$ is diagonal with non-negative entries, $U^TU = I$, and $V^TV = I$.

$\mathbf{V}$ are the first r eigenvectors of $\mathbf{A}^T\mathbf{A}$ with eigenvalues $\text{diag}(\mathbf{S}^2)$

$\mathbf{U}$ are the first r eigenvectors of $\mathbf{A}\mathbf{A}^T$ with eigenvalues $\text{diag}(\mathbf{S}^2)$

So how do we find the eigenvectors of $\tilde{\mathbf{X}}^\top\tilde{\mathbf{X}}$?

$\text{SVD}(\tilde{\mathbf{X}}) = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ $\therefore$ *(definition of SVD)*

$\mathbf{V}$ contains the first $\mathbf{r}$ eigenvectors of $\tilde{\mathbf{X}}^\top\tilde{\mathbf{X}}$, $\therefore$ *(fact from lecture notes)*

the principal components of $\mathbf{X}$ exist in $\mathbf{V}$ from the SVD of $\tilde{\mathbf{X}}$ *(PCs of* $\mathbf{X}$ *= eigenvectors of* $\tilde{\mathbf{X}}^\top\tilde{\mathbf{X}}$*)*

Remember this?:

$$\sigma_1 > \sigma_2 > \ldots > \sigma_r$$

Selecting the first *q* vectors from **V** gives us the directions with the GREATEST variance… **principal components!**

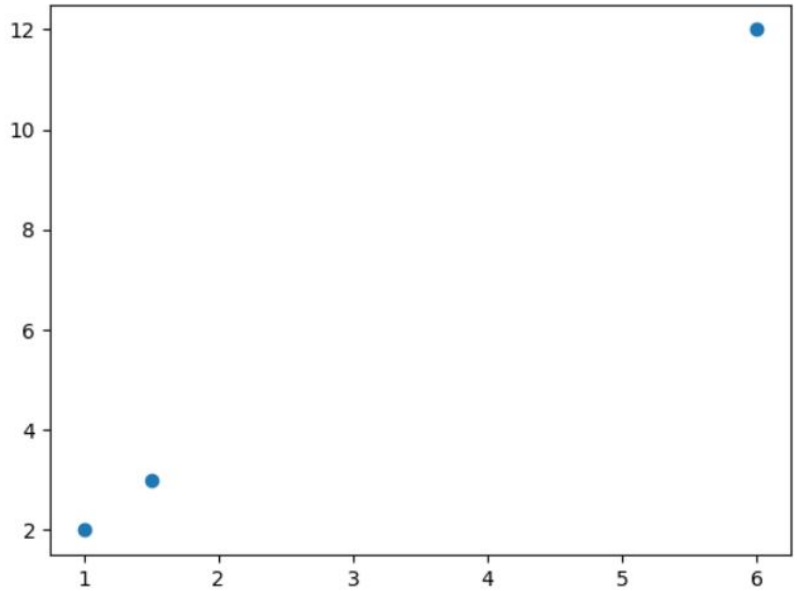
Problem 1ab

(a) What is the first principal component vector, v_1?

Consider the following dataset, which is represented as three points in $\mathbb{R}^2$. Note that in this problem we will **not** demean the dataset.

$$\begin{bmatrix} 1 & 2 \\ 1.5 & 3 \\ 6 & 12 \end{bmatrix}$$

(b) What is the second principal component, v_2?



1. Compute X^TX

$$X = \begin{bmatrix} 1 & 2 \\ 1.5 & 3 \\ 6 & 12 \end{bmatrix}$$

$$X^TX = \begin{bmatrix} 1 & 1.5 & 6 \\ 2 & 3 & 12 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 1.5 & 3 \\ 6 & 12 \end{bmatrix} = \begin{bmatrix} 39.25 & 78.5 \\ 78.5 & 157 \end{bmatrix}$$

2. Find eigenvalues

$$\det(X^TX - \lambda I) = 0$$

$$(39.25 - \lambda)(157 - \lambda) - (78.5^2) = 0$$
$$\lambda^2 - 196.25\lambda + (39.25 \times 157 - 6162.25) = 0$$
$$\lambda^2 - 196.25\lambda = 0$$
$$\lambda(\lambda - 196.25) = 0$$

$$\lambda_1 = 196.25, \lambda_2 = 0$$

3. Find singular values

$\boldsymbol{\sigma} = \sqrt{\boldsymbol{\lambda}}$

$\boldsymbol{\sigma}_1 = \sqrt{\boldsymbol{\lambda}}_1 = \sqrt{196.25}, \boldsymbol{\sigma}_2 = \sqrt{\boldsymbol{\lambda}}_2 = 0$

4. Find the singular vectors

Solve $(X^TX - \boldsymbol{\lambda}I)\mathbf{v} = 0$, $\boldsymbol{\lambda}_1 = 196.25$

$$\begin{bmatrix} 39.25 - 196.25 & 78.5 \\ 78.5 & 157 - 196.25 \end{bmatrix} \mathbf{v} = \begin{bmatrix} -157 & 78.5 \\ 78.5 & -39.25 \end{bmatrix} \mathbf{v} = 0$$

$-157v_1 + 78.5v_2 = 0$

$v_2 = 2v_1 \longrightarrow [1,2]^T$ is our eigenvector

$$v_1 = \begin{bmatrix} 1/\sqrt{5} \\ 2/\sqrt{5} \end{bmatrix}$$

(a) What is the first principal component vector, v_1?

Each point has second coordinate twice the first, so every point is on the line $y = 2x$, or equivalently is a multiple of the vector $[1, 2]$.

That direction, normalized, is the first principal component, so $v_1 = [1/\sqrt{5}, 2/\sqrt{5}] \approx [0.45, 0.89]$.

(b) What is the second principal component, v_2?

Since every data is in the span of the first principal component, any unit norm vector perpendicular to v_1 is an acceptable choice. One such vector is $[-2/\sqrt{5}, 1/\sqrt{5}] \approx [-0.89, 0.45]$.

(c) If we use only the first principal component to compress the dataset, what will the representation of each point be?

The first point is $\sqrt{5}v_1$, the second one is $1.5 \cdot \sqrt{5}v_1$, and the third one is $6 \cdot \sqrt{5}v_1$.

(d) Will this representation be lossy, or perfectly preserve the dataset?

In this particular dataset, we perfectly preserve this dataset (the points are all multiples of v_1).

24. 3 points

Consider a dataset $X \in \mathbb{R}^{n \times p}$ with n observations and p features, and with corresponding covariance matrix Σ. Let $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_p$ be the eigenvalues of Σ in descending order. Express the total variance explained by the first k principal components (obtained by performing Principal Component Analysis (PCA) on X) as a fraction of the total variance in the original data.

Answer: Fraction of total variance explained = _____

Explanation: $\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{p} \lambda_i}$. The fraction of total variance explained by the first k principal components in PCA can be expressed as the ratio of the sum of the first k eigenvalues to the sum of all eigenvalues of the covariance matrix Σ.

26. 4 points | Select All That Apply

You are applying PCA to a training dataset of $n = 1024$ grayscale images that are each 16×16 pixels (256 pixels per image). Consider reshaping each image into a vector $x_i \in \mathbb{R}^{256}$ and then composing a data matrix $X \in \mathbb{R}^{1024 \times 256}$, where the i^{th} row is $x_i^\top$. Let $\hat{x}_{i,k} \in \mathbb{R}^{256}$ be the PCA reconstruction of image x_i using the top k principal component directions in the data. Let $R(k)$ be the average reconstruction error on the training data using k principal components, $R(k) = \frac{1}{n} \sum_{i=1}^{n} \|x_i - \hat{x}_{i,k}\|_2^2$. Which of the following statements are true?

(a) $R(k)$ is monotonically decreasing as k increases, up to $k = 1024$. That is, if $0 < k_1 < k_2 \leq 1024$, then $R(k_1) > R(k_2)$. Note the strict equality.

(b) If $k < \text{rank}(X)$, then $R(k) > 0$.

(c) If $k \geq \text{rank}(X)$, then $R(k) = 0$.

(d) For $k \geq 1$, let $\delta(k) = R(k-1) - R(k)$ be the decrease in reconstruction error by going from $k-1$ to k principal components. (When $k = 0$, define the reconstruction of x_i to simply be the mean image $\bar{x}$.) Then, $\delta(k)$ is monotonically non-increasing as k increases.

Explanation:

a) False. The number of principal components cannot exceed the rank of X, which is $\min(n, 256)$. Since X is 1024×256, its rank is at most 256. Thus, $R(k)$ is only guaranteed to monotonically decrease for $k \leq 256$, not $k \leq 1024$.

b) True. The reconstruction error is non-zero when the number of principal components k is less than the rank of X, as there are remaining variations in X not captured by the top k components.

c) True. When k is greater than or equal to the rank of X, the PCA reconstruction captures all the variation in X, resulting in zero reconstruction error.

d) True. Each additional principal component explains the maximum remaining variance, so the decrease in reconstruction error ($R(k)$) diminishes as k increases, making $R(k)$ monotonically non-increasing.

Final Review

Trees/Bootstrap

15. 4 points | Select All That Apply

Assume we are given a fixed dataset $D = \{x^{(1)}, x^{(2)}, \ldots, x^{(n)}\}$ drawn i.i.d. (independently and identically distributed) from an underlying distribution $P(x)$. We use the bootstrap to draw bootstrap samples $\tilde{D} = \{\tilde{x}^{(1)}, \tilde{x}^{(2)}, \ldots\}$ from a bootstrap distribution $Q(x)$. Which of the following statements are true?

(a) The bootstrap samples in $\tilde{D}$ are drawn by sampling with replacement from D.

(b) The bootstrap samples in $\tilde{D}$ are drawn by sampling without replacement from D.

(c) The distribution of bootstrap samples in $\tilde{D}$ is always identical to the underlying data distribution P.

(d) The bootstrap samples in $\tilde{D}$ are independently and identically distributed.

a) True. The bootstrap distribution is created by sampling with replacement from the fixed dataset.

b) False. Inverse of option (a)

c) False. Bootstrap samples are not guaranteed to be identical to population distribution.

d) True. By construction of the bootstrap method.

Neural Networks

19. 4 points | Select All That Apply

In a neural network, the number of layers is an important hyperparameter. Which of these statements are true about adding layers to a neural network (keeping all other aspects of the model and training process the same)?

(a) Hyperparameters are independent, i.e., adding more layers will not affect the optimal choice of step size for gradient descent or the amount of regularization needed.

(b) We cannot use cross-validation to select hyperparameters that directly affect model architecture, such as the number of layers.

(c) Adding more layers generally decreases the training loss.

(d) Adding more layers generally increases the ability of the model to overfit the data.

Explanation:

a) False. Adding layers can affect optimal learning rates and regularization needs.

b) False. Cross-validation can be used to select architecture-related hyperparameters like the number of layers.

c) True. More layers improve representational capacity, reducing training loss.

d) True. Deeper networks can overfit without proper regularization.

22. 2 points | One Answer

Consider a fully connected neural network (MLP) with an input layer, a hidden layer, and an output layer. The input layer has n units, the hidden layer has h units, and the output layer has m units. Assume there are no bias units/terms. Which of the following statements about the number of trainable parameters is true?

a) The total number of trainable parameters is $n \cdot h \cdot m$.

b) The total number of trainable parameters is $n \cdot h + h \cdot m$.

c) The total number of trainable parameters is $(n+1) \cdot h + (h+1) \cdot m$.

d) The total number of trainable parameters is $n + h + m$.

Explanation: Connections between the input and the hidden layer: $n \cdot h$; connections between the hidden and the output layer: $h \cdot m$.