

# COM SCI C121 Week 8

Aidan Jan

May 21, 2024

## Matrices

- A matrix can be used to store data
- Consider a matrix of dimension  $m \times n$
- The rows can be a high-dimensional sample of dimension  $n$
- That means that each column is *one* dimension across *all* samples,  $m$

The goal is to summarize the data in the matrix to see relationship across samples.

- The matrix itself is described by  $m \times n$  data points, which on its own is unwieldy
- Naturally, we would want a way to summarize  $n$  data points into an "intuitive" representation that we can make sense of
- One way to think about this is creating a "faithful" representation of  $m \times n$  data points into, say,  $n \times 2$  data points while preserving structure in the data.
  - Taking this to the extreme, we will look at linear algebra for inspiration
- In an ideal case, samples that are more "similar" will "cluster" together. For example, if I put tumor samples and normal samples, one would expect the tumor samples to be more similar to each other and the normal samples to be more similar to each other.

## What is a Matrix in Linear Algebra?

In short, a matrix is code for a linear function (to transform a set of vectors to another). For example,

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \rightarrow \begin{cases} s = x + y \\ t = 0x + y \end{cases}$$

where  $x$  and  $y$  are the  $x$  and  $y$  of the original vector, and  $s$  and  $t$  are the components of the new vector.

## Singular Value Decomposition (SVD)

What is singular value decomposition about?

- Linear transformations, and their corresponding matrices (which are rectangular tables filled with numbers), are seemingly complicated and arbitrary.
- The singular value decomposition (SVD) says that every matrix is essentially diagonal, i.e., "nice", provided the "right" bases are used for the domain and range spaces.
- By finding the "right" bases, the SVD provides fundamental insights into linear transformations and their accompanying matrix representations.

## SVD Algorithm

- **Input:** an  $m \times n$  matrix
- **Output:** a set of numbers called *singular values* and a two collection of vectors: a set of *right singular vectors* and another set of *left singular vectors*.

$$\begin{bmatrix} M \end{bmatrix} = \begin{bmatrix} U \end{bmatrix} \times \begin{bmatrix} \Sigma \end{bmatrix} \times \begin{bmatrix} V^* \end{bmatrix}$$

... where:

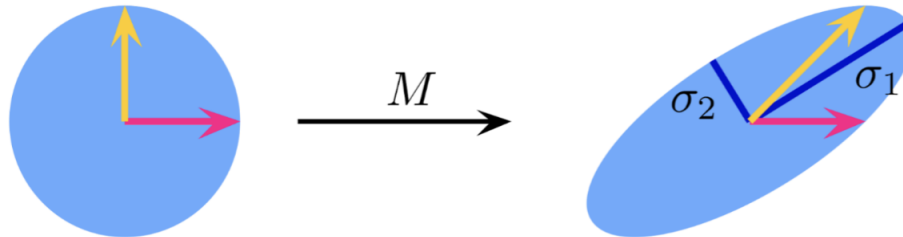
- $M$  is a  $m \times n$  matrix
- $U$  is a  $m \times m$  matrix
- $\Sigma$  is a  $m \times n$  matrix
- $V^*$  is a  $n \times n$  matrix

additionally

- $U$  and  $V^*$  have the property that their transposes equal their inverses.

## The meaning of *singular values*

- As a linear map, an  $m \times n$  matrix  $M$  can be thought of as mapping a vector  $x$  from  $R^n$  to  $R^m$ .
- A unit sphere in  $R^n$  is mapped to an ellipsoid in  $R^m$
- The non-zero *singular values* of  $M$  are the lengths of the *semi-axes* of the ellipsoid.



## Measuring directions of distortion

- The maximal singular value can therefore be understood to be the size of the vector that points in the direction in which the linear transformation corresponding to  $M$  has the largest effect.
- Formally, the maximal singular value, which is usually denoted as  $\sigma_1$  can be understood to be

$$\sigma_1 = \max_{x: \|x\|=1} \|Mx\|$$

- Similarly, the smallest non-zero singular value is the size of the smallest semi-axis of the ellipsoid that is the image of the unit sphere under  $M$ .

## Calculating SVD

- Let  $M = U\Sigma V^T$  and set  $M_k = \sum_{i=1}^k \sigma_i u_i v_i^T$ . The matrix  $M_k$  is a good low-rank matrix approximation of  $M$ . Specifically,

$$\min_{\text{rank}(X)=k} \|A - B\|_2 = \|A - A_k\|_2 = \sigma_{k+1}$$

- This theorem makes precise the intuition that the top singular values, which measure the sizes of the largest of the semi-axes of the ellipsoid resulting from mapping of a sphere by the linear transformation corresponding to a matrix, capture "most" of the transformation.

## A Centered Matrix can be Transformed to Summarize the Covariance of the Data

- Take a matrix  $A$  with  $m \times n$  data points. Create  $M$  by subtracting the mean of each column from that corresponding column

$$- M_{ij} = A_{ij} - \frac{1}{n} \sum_{k=1}^n a_{kj}$$

- $M$  is now a matrix that is *centered*

- $M^T M$  is a *covariance matrix* of  $A$  with the property:

$$- (M^T M)_{ij} = \text{Cov}(A_i, A_j)$$

- In words,  $M_{ij}$  represents how much samples  $i$  and  $j$  covary with each other

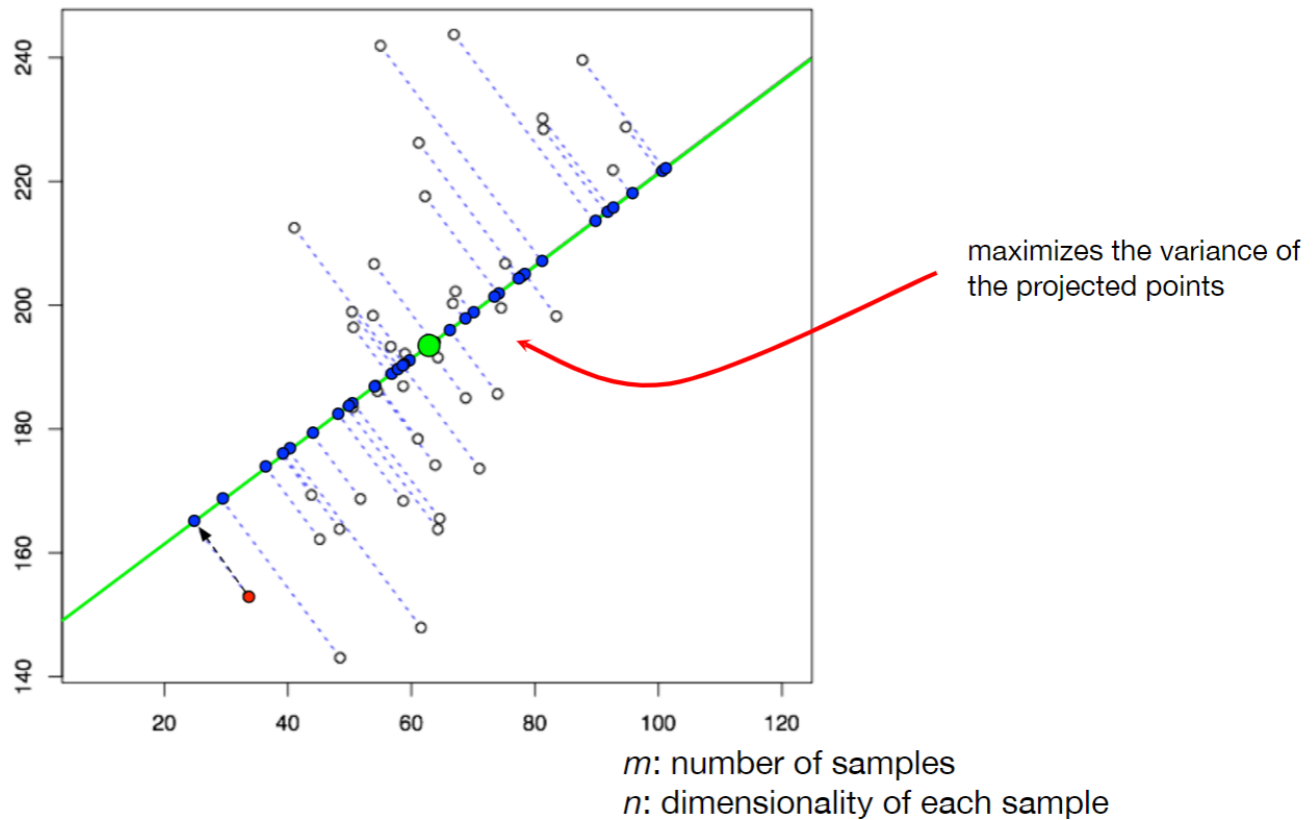
- \* If  $(M^T M)_{ij} > 0$ , samples  $i$  and  $j$  are positively related
- \* If  $(M^T M)_{ij} < 0$ , samples  $i$  and  $j$  are negatively related
- \* If  $(M^T M)_{ij} \approx 0$ , samples  $i$  and  $j$  are unrelated.

## PCA with SVD

- The SVD of a (centered)  $M$ , given by  $M = U\Sigma V^T$ , yields a decomposition of  $M^T M$  and  $M^T M = V\Sigma^2 V^{-1}$ , i.e., eigendecomposition of the covariance matrix  $M^T M$  can be performed by SVD of  $M$ . Set  $V_k$  to be the first  $k$  columns of  $V$ , i.e.,  $V_k = [v_1, v_2, \dots, v_k]$ . Then the projection of the points in  $M$  by  $V_k$ , i.e.,  $\text{PCA}(k) = MV_k$  has numerous useful properties.

## An Example of a PCA Projection

- Each dot represents a sample
- Remember,  $M$  is  $m \times n$ ,  $V$  is  $n \times n$ .
- If I take the first two columns of  $V$ :
  - $\text{PCA}(2) = MV_2$  results in a  $m \times 2$  matrix.

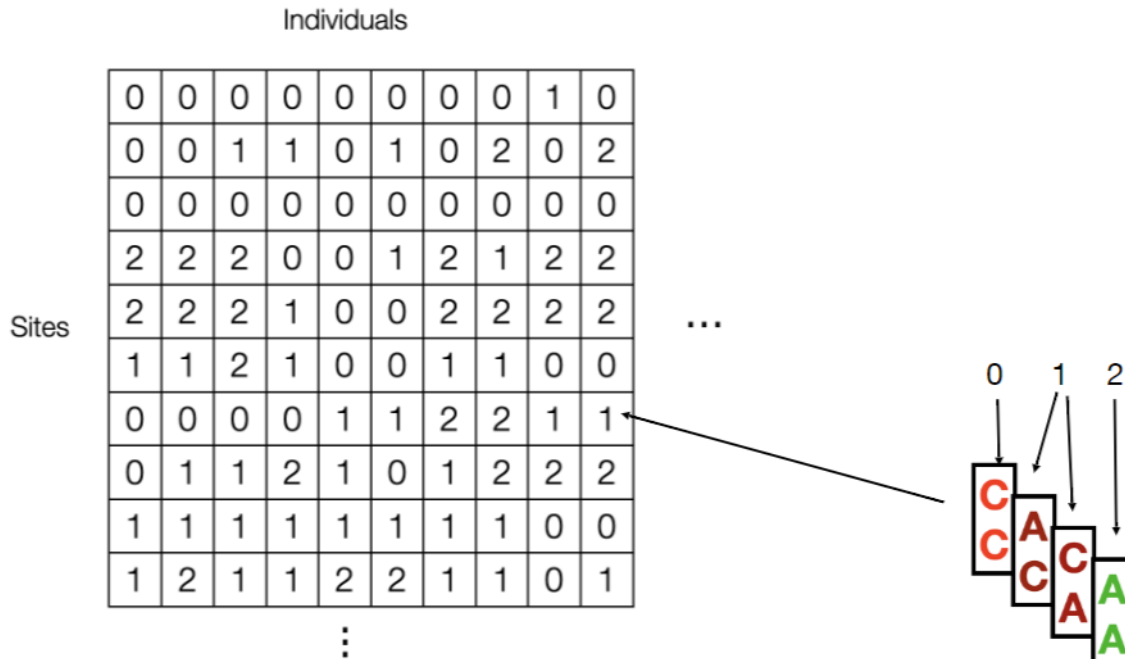


PCA Steps:

- Start with a data matrix  $A$ .
- Center  $A$  to get  $M$ .
- $M$  has a singular value decomposition that is derived from viewing  $M$  as a linear transformation.  
 $M = U\Sigma V^T$ .
- The matrix  $V$  consists of the eigenvectors which diagonalize the covariance matrix  $M^T M$ .
- Compute  $V$  from  $M$  using the SVD.
- Let  $V_k$  be the truncation of  $V$  to its first  $k$  columns. We know from linear algebra that this is a meaningful restriction because  $M_k = U_k \Sigma V_k^T$  is a good low-rank approximation to  $M$ .
- Project the data matrix  $M$  with  $V_k$  to obtain a new set of points:  $MV_k$ .
- The projection has the property that it will maximize the variance of the projected points.

### An Application of PCA: the Human Genotype Matrix

- Differences between any pair of human genomes are largely in the same sites, and consist of single nucleotide polymorphisms (SNPs).
- Most human SNPs are biallelic.



## "Genes Mirror Geography Within Europe"

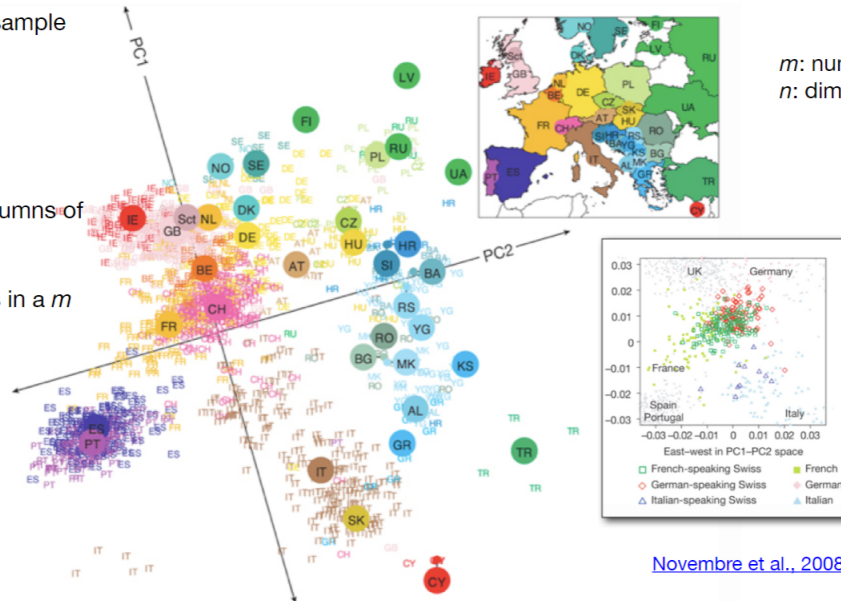
Each dot represents a sample (person) with  $n$  SNPs

$M$  is  $m \times n$

$V$  is  $n \times n$

If I take the first two columns of  $V$ :

$PCA(2) = MV_2$  results in a  $m \times 2$  matrix



## Some Properties of the PCA

- Each subsequent dimension explains less variance than the previous
  - i.e., there are *diminishing returns* by including additional PCs
- The singular values,  $\sigma_i$  are related to how much variance each dimension explains
  - Proportion of variance explained:  $\sigma_p^2 / \sum_{j=1}^n \sigma_j^2$

## Choosing $k$ in K-means Clustering

As a review, K-means minimizes the following loss function:

$$L(\mu, \alpha) = \sum_{k=1}^K \sum_{i=1}^n \|x_i - \mu_k\|_2^2 \mathbb{1}\{\alpha_i = k\}$$

where

- $x_i$  is the data
- $\mu_k$  is the cluster center (mean of the data)
- $\mathbb{1}$  is a function that returns 1 if the condition is true. In this case, 1 if the data belongs to cluster  $k$ , 0 otherwise.

What happens if  $k$  increases and approaches  $n$ ?

### More thoughts

- Conceptually, you want a clustering that satisfies the *good clustering principle*:
  - "Every pair of points from the same cluster should be close to each other than any pair of points from different clusters."
- Does setting  $k = n$  satisfy this principle?

### In Reality

Technically yes. But then the data would not be useful.

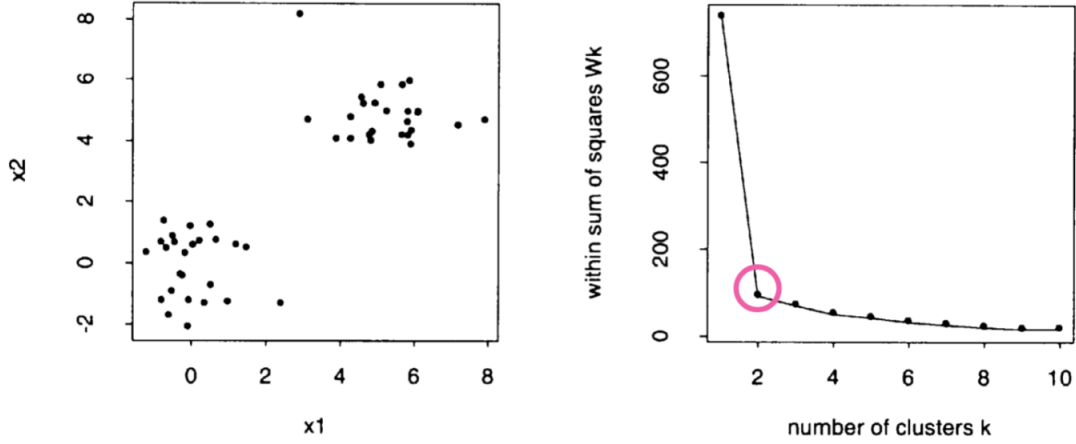
- We want to satisfy the good clustering principle with the *simplest* model that sufficiently captures the complexity of the data

### The Good Clustering Principle Mathematically

- Let  $d_{ii'} = \|x_i - x_{i'}\|_2^2 = \sum_{j=1}^p (x_{ij} - x_{i'j})^2$ , the pairwise distance between two points.
- Let  $D_r = \sum_{i,i' \in C_r} d_{ii'}$  be the pairwise distance between all points in a cluster  $C_r$ .
- Then

$$W_k = \sum_{r=1}^K \frac{1}{2n_r} D_r$$

- where  $n_r$  is the number of points in cluster  $r$ .
- What is k-means with respect to  $W$ ? K-means minimizes  $W$ . ( $W$  is the same objective function as the loss function above.)



Graphically, we are looking for the value of  $k$  at the "elbow". Unfortunately, a lot of the times, the graph isn't that clean.

## Gap Statistic

A more principled approach: the Gap statistic is the difference in the expected  $W$  under a *reasonable null distribution* and the observed  $W$ .

$$\text{Gap}_n(k) = E_n^*\{\log(W_k)\} - \log(W_k)$$

To see how to find an appropriate reference distribution, consider for a moment the population version corresponding to the gap statistic in the case of K-means clustering:

$$g(k) = \log \left\{ \frac{\text{MSE}_{X^*}(k)}{\text{MSE}_{X^*}(1)} \right\} - \log \left\{ \frac{\text{MSE}_X(k)}{\text{MSE}_X(1)} \right\}$$