Overview

- Vectors and matrices
- Basic vector/matrix operations
- Various matrix types
- Projections
- More on matrix types
- Matrix determinants
- Matrix inversion
- Eigenanalysis
- Singular value decomposition

Orthogonal/Orthonormal vectors

- Two vectors are orthogonal if they are perpendicular to one another
  - \( A \cdot B = 0 \)
  - A vector that is perpendicular to a plane is orthogonal to every vector on the plane

- Two vectors are orthonormal if
  - They are orthogonal
  - The length of each vector is 1.0
  - Orthogonal vectors can be made orthonormal by normalizing their lengths to 1.0

Orthogonal matrices

- Orthogonal Matrix: \( AA^T = A^T A = I \)
  - The matrix is square
  - All row vectors are orthonormal to one another
  - Every vector is perpendicular to the hyperplane formed by all other vectors
  - All column vectors are also orthonormal to one another
  - Observation: In an orthogonal matrix if the length of the row vectors is 1.0, the length of the column vectors is also 1.0
  - Observation: In an orthogonal matrix no more than one row can have all entries with the same polarity (+ve or -ve)

Orthogonal and Orthonormal Matrices

- Orthogonal matrices will retain the length and relative angles between transformed vectors
  - Essentially, they are combinations of rotations, reflections and permutations
  - Rotation matrices and permutation matrices are all orthonormal

- If the vectors in the matrix are not unit length, it cannot be orthogonal
  - \( AA^T = I, A^T A = I \)
  - \( AA^T = \text{Diagonal} \) or \( A^T A = \text{Diagonal} \), but not both
  - If all the entries are the same length, we can get \( AA^T = A^T A = \text{Diagonal} \), though
  - A non-square matrix cannot be orthogonal
  - \( AA^T \) or \( A^T A \), but not both

Matrix Rank and Rank-Deficient Matrices

- Some matrices will eliminate one or more dimensions during transformation
  - These are rank-deficient matrices
  - The rank of the matrix is the dimensionality of the transformed version of a full-dimensional object
Matrix Rank and Rank-Deficient Matrices

- Some matrices will eliminate one or more dimensions during transformation
  - These are rank-deficient matrices
  - The rank of the matrix is the dimensionality of the transformed version of a full-dimensional object

Non-square Matrices

- Non-square matrices add or subtract axes
  - More rows than columns → add axes
  - But does not increase the dimensionality of the data
  - Fewer rows than columns → reduce axes
  - May reduce dimensionality of the data

Projections are often examples of rank-deficient transforms

M =

W =

- P = W (W^T W)^{-1} W^T; Projected Spectrogram = P * M
- The original spectrogram can never be recovered
- P is rank-deficient
- P explains all vectors in the new spectrogram as a mixture of only the 4 vectors in W
  - There are only a maximum of 4 independent bases
  - Rank of P is 4

The Rank of a Matrix

- The matrix rank is the dimensionality of the transformation of a full-dimensional object in the original space
- The matrix can never increase dimensions
  - Cannot convert a circle to a sphere or a line to a circle
- The rank of a matrix can never be greater than the lower of its two dimensions

Non-square Matrices

- Non-square matrices add or subtract axes
  - More rows than columns → add axes
  - But does not increase the dimensionality of the data

The Rank of Matrix

- Projected Spectrogram = P * M
  - Every vector in it is a combination of only 4 bases
  - The rank of the matrix is the smallest no. of bases required to describe the output
  - E.g., if note no. 4 in P could be expressed as a combination of notes 1, 2, and 3, it provides no additional information
  - Eliminating note no. 4 would give us the same projection
  - The rank of P would be 3!
Matrix rank is unchanged by transposition

- If an N-dimensional object is compressed to a K-dimensional object by a matrix, it will also be compressed to a K-dimensional object by the transpose of the matrix.

Matrix Determinants

- Matrix determinants are only defined for square matrices.
  - They characterize volumes in linearly transformed space of the same dimensionality as the vectors.
- Rank deficient matrices have determinant 0.
  - Since they compress full-volume N-dimensional objects into zero-volume N-dimensional objects.
    - E.g., a 3-D sphere into a 2-D ellipse: The ellipse has 0 volume (although it does have area).
- Conversely, all matrices of determinant 0 are rank deficient.
  - Since they compress full-volume N-dimensional objects into zero-volume objects.

Multiplication properties

- Properties of vector/matrix products:
  - Associative: \( A \cdot (B \cdot C) = (A \cdot B) \cdot C \)
  - Distributive: \( A \cdot (B + C) = A \cdot B + A \cdot C \)
    - NOT commutative!!!
      - \( A \cdot B \neq B \cdot A \)
        - left multiplications ≠ right multiplications
  - Transposition:
    - \( (A \cdot B)^T = B^T \cdot A^T \)

Determinate properties

- Associative for square matrices: \( |A \cdot B \cdot C| = |A| \cdot |B| \cdot |C| \)
  - Scaling volume sequentially by several matrices is equal to scaling once by the product of the matrices.
- Volume of sum \( \pm \) sum of Volumes: \( |B + C| = |B| + |C| \)
- Commutative:
  - The order in which you scale the volume of an object is irrelevant:
    - \( |A \cdot B| = |B \cdot A| = |A| \cdot |B| \)
Matrix Inversion

- A matrix transforms an N-dimensional object to a different N-dimensional object
- What transforms the new object back to the original? — The inverse transformation
- The inverse transformation is called the matrix inverse

Revisiting Projections and Least Squares

- Projection computes a least squared error estimate
- For each vector \( V \) in the music spectrogram matrix
  - Approximation: \( \mathbf{V}_{\text{approx}} = a \mathbf{v}_1 + b \mathbf{v}_2 + c \mathbf{v}_3 \)
  
  \[
  T = \begin{bmatrix}
  a \\
  b \\
  c
  \end{bmatrix}
  \]

  \[
  \mathbf{V}_{\text{approx}} = T \mathbf{b}
  \]

  - Error vector \( E = V - \mathbf{V}_{\text{approx}} \)
  - Squared error energy for \( V \): \( \mathbf{e}(V) = \|E\|^2 \)
- Projection computes \( \mathbf{V}_{\text{approx}} \) for all vectors such that Total error is minimized
- But WHAT ARE “a”, “b” and “c”?

The Pseudo Inverse (PINV)

- We are approximating spectral vectors \( V \) as the transformation of the vector \( [a \ b \ c]^T \)
  - Note — we’re viewing the collection of bases in \( T \) as a transformation
- The solution is obtained using the pseudo inverse
  - This gives us a LEAST SQUARES solution
    - If \( T \) were square and invertible \( \text{Pinv}(T) = T^{-1} \), and \( V = V_{\text{approx}} \)

Explaining music with one note

- Recap: \( P = W \ (W^T W)^{-1} \) — Projected Spectrogram = \( P^* M \)
- Approximation: \( M = W^* X \)
- The amount of \( W \) in each vector = \( X = \text{Pinv}(W)^* M \)
- \( W^* \text{Pinv}(W)^* M = \) Projected Spectrogram
- \( W^* \text{Pinv}(W) = \) Projection matrix\( ! \)

Inverting rank-deficient matrices

- Rank deficient matrices “flatten” objects
  - In the process, multiple points in the original object get mapped to the same point in the transformed object
- It is not possible to go “back” from the flattened object to the original object
  - Because of the many-to-one forward mapping
- Rank deficient matrices have no inverse
**Explanation with multiple notes**

\[ M = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \]

\[ W = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \]

\[ X = \text{Pinv}(W) \cdot M; \quad \text{Projected matrix} = W^T X = W^T \text{Pinv}(W) \cdot M \]

**Matrix inversion (division)**

- The inverse of matrix multiplication
  - Not element-wise division!!
- Provides a way to "undo" a linear transformation
  - Inverse of the unit matrix is itself
  - Inverse of a diagonal is diagonal
  - Inverse of a rotation is a (counter)rotation (its transpose!)
  - Inverse of a rank deficient matrix does not exist!
    - But pseudoinverse exists
- For square matrices: Pay attention to multiplication side!
  \[ A \cdot B = C, \quad A = C \cdot B^{-1}, \quad B = A^{-1} \cdot C \]
- If matrix is not square use a matrix pseudoinverse:
  \[ A \cdot B = C, \quad A = C \cdot B^+, \quad B = A^+ \cdot C \]
- MATLAB syntax: \text{inv}(a), \text{pinv}(a)

**How about the other way?**

\[ M = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \]

\[ V = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \]

\[ W = \begin{bmatrix} ? \\ ? \end{bmatrix} \]

\[ WV = M \quad W = M \text{Pinv}(V) \quad U = WV \]

**Eigenanalysis**

- If something can go through a process mostly unscathed in character it is an eigen-something
  - Sound example: \[ \text{sound}(\max) \]
- A vector that can undergo a matrix multiplication and keep pointing the same way is an eigenvector
  - Its length can change though
- How much its length changes is expressed by its corresponding eigenvalue
  - Each eigenvector of a matrix has its eigenvalue
- Finding these "eigenthings" is called eigenanalysis

**Pseudo-inverse (PINV)**

- \text{Pinv()} applies to non-square matrices
- \text{Pinv (Pinv(A))} = A
- \[ A^* \text{Pinv}(A) = \text{projection matrix!} \]
  - Projection onto the columns of A
- If \[ A = K \times N \text{ matrix and } K > N, \text{ A projects N-D vectors into a higher-dimensional K-D space} \]
  - \text{Pinv}(A) = N\times K matrix
  - \text{Pinv}(A)^*A = I \text{ in this case}
- Otherwise \[ A^* \text{ Pinv}(A) = I \]

**EigenVectors and EigenValues**

- Vectors that do not change angle upon transformation
  - They may change length
  \[ MV = \lambda V \]
  - \( V \) = eigen vector
  \[ \lambda \] = eigen value
**Eigen vector example**

- Matrix transformation “transforms” the space
  - Warps the paper so that the normals to the two vectors now lie along the axes

**A stretching operation**

- Draw two lines
- Stretch / shrink the paper along these lines by factors $\lambda_1$ and $\lambda_2$
  - The factors could be negative – implies flipping the paper
- The result is a transformation of the space

**Matrix multiplication revisited**

- The result of the stretching is exactly the same as transformation by a matrix
- The axes of stretching/shrinking are the eigenvectors
  - The degree of stretching/shrinking are the corresponding eigenvalues
- The EigenVectors and EigenValues convey all the information about the matrix

**Physical interpretation of eigen vector**

- The result of the stretching is exactly the same as transformation by a matrix
- The axes of stretching/shrinking are the eigenvectors
  - The degree of stretching/shrinking are the corresponding eigenvalues
- The EigenVectors and EigenValues convey all the information about the matrix
**Eigen Analysis**

- Not all square matrices have nice eigen values and vectors
  - E.g., consider a rotation matrix
    
    ![Rotation Matrix Diagram]
    
    - This rotates every vector in the plane
    - No vector that remains unchanged
- In these cases, the Eigen vectors and values are complex

---

**Singular Value Decomposition**

- U and V are orthonormal matrices
  - Columns are orthonormal vectors
- S is a diagonal matrix
- The right singular vectors in V are transformed to the left singular vectors in U
  - And scaled by the singular values that are the diagonal entries of S

---

**Singular Value Decomposition**

- Matrix transformations convert circles to ellipses
- Eigen vectors are vectors that do not change direction in the process
- There is another key feature of the ellipse to the left that carries information about the transform
  - Can you identify it?

---

**Singular Value Decomposition**

- The left and right singular vectors are not the same
  - If A is not a square matrix, the left and right singular vectors will be of different dimensions
- The singular values are always real
- The largest singular value is the largest amount by which a vector is scaled by A
  - $\max (|Ax| / |x|) = \sigma_{\text{max}}$
- The smallest singular value is the smallest amount by which a vector is scaled by A
  - $\min (|Ax| / |x|) = \sigma_{\text{min}}$
  - This can be 0 (for low-rank or non-square matrices)

---

**The Singular Values**

- Square matrices: product of singular values = determinant of the matrix
  - This is also the product of the eigenvalues
  - I.e., there are two different sets of axes whose products give you the area of an ellipse
- For any "broad" rectangular matrix A, the largest singular value of any square submatrix B cannot be larger than the largest singular value of A
  - An analogous rule applies to the smallest singular value
  - This property is utilized in various problems, such as compressive sensing
SVD vs. Eigen Analysis

- Eigen analysis of a matrix $A$:
  - Find two vectors such that their absolute directions are not changed by the transform
- SVD of a matrix $A$:
  - Find two vectors such that the angle between them is not changed by the transform
- For one class of matrices, these two operations are the same

Symmetric Matrices

- Matrices that do not change on transposition
  - Row and column vectors are identical
- Symmetric matrix: Eigen vectors and Eigen values are always real
  - $A$ is orthogonal
- Eigen vectors are always orthogonal
  - At 90 degrees to one another

A matrix vs. its transpose

- Multiplication by matrix $A$:
  - Transforms right singular vectors in $V$ to left singular vectors $U$
- Multiplication by its transpose $A^T$:
  - Transforms left singular vectors $U$ to right singular vector $V$
- $A^T$: Converts $V$ to $U$, then brings it back to $V$
  - Result: Only scaling

Symmetric Matrices

- Eigen vectors $V$ are orthonormal
  - $V_i V_j = 1$
  - $V_i V_j = 0$, $i \neq j$
- Listing all eigen vectors in matrix form $V$
  - $V^T V = I$
  - $V V^T = I$
  - $V V^T = I$
- $M V = \lambda V$
- In matrix form: $M V = V \Lambda$
  - $\Lambda$ is a diagonal matrix with all eigen values
- $M = V \Lambda V^T$
Square root of a symmetric matrix

\[ C = VAV^T \]

\[ \text{Sqrt}(C) = V \cdot \text{Sqrt}(\Lambda) \cdot V^T \]

\[ \text{Sqrt}(C) = V \cdot \text{Sqrt}(\Lambda) \cdot V^T \]

\[ \text{Sqrt}(\Lambda) \cdot V^T = VAV^T = C \]

- The square root of a symmetric matrix is easily derived from the Eigen vectors and Eigen values
  - The Eigen values of the square root of the matrix are the square roots of the Eigen values of the matrix
  - For correlation matrices, these are also the “singular values” of the data set

Definiteness..

- SVD: Singular values are always positive!
- Eigen Analysis: Eigen values can be real or imaginary
  - Real, positive Eigen values represent stretching of the space along the Eigen vector
  - Real, negative Eigen values represent stretching and reflection (across origin) of Eigen vector
  - Complex Eigen values occur in conjugate pairs
- A square (symmetric) matrix is positive definite if all Eigen values are real and positive, and are greater than 0
  - Transformation can be explained as stretching and rotation
  - If any Eigen value is zero, the matrix is positive semi-definite

Positive Definiteness..

- Property of a positive definite matrix: Defines inner product norms
  - \( x^T A x \) is always positive for any vector \( x \) if \( A \) is positive definite
- Positive definiteness is a test for validity of Gram matrices
  - Such as correlation and covariance matrices
  - We will encounter other gram matrices later

The Correlation and Covariance Matrices

- Consider a set of column vectors ordered as a DxN matrix \( A \)
- The correlation matrix is
  - \( C = (1/N)AA^T \)
    - If the average (mean) of the vectors in \( A \) is subtracted out of all vectors,
      \( C \) is the covariance matrix
    - Covariance = correlation * mean * mean
- Diagonal elements represent average of the squared value of each dimension
  - Off diagonal elements represent how two components are related
  - How much knowing one lets us guess the value of the other

The Correlation Matrix

- Projections along the N Eigen vectors with the largest Eigen values represent the N greatest “energy-carrying” components of the matrix
- Conversely, N “bases” that result in the least square error are the N best Eigen vectors

Square root of the Covariance Matrix
An audio example

- The spectrogram has 974 vectors of dimension 1025
- The covariance matrix is size 1025 x 1025
- There are 1025 eigenvectors

Eigenvalues and Eigenvectors

- The vectors in the spectrogram are linear combinations of all 1025 Eigen vectors
- The Eigen vectors with low Eigen values contribute very little
  - The average value of $a_i$ is proportional to the square root of the Eigenvalue
  - Ignoring these will not affect the composition of the spectrogram

Eigen Reduction

$$M = \text{spectrogram}$$

$$C = MM^T$$

$$V = \text{1025x1025}$$

$$V_{\text{reduced}} = [V_1, \ldots, V_{25}]$$

$$M_{\text{reduced}} = \text{Pinv}(V_{\text{reduced}})M$$

- Compute the Correlation
- Compute Eigen vectors and values
- Create matrix from the 25 Eigen vectors corresponding to 25 highest Eigen values
- Compute the weights of the 25 eigenvectors
- To reconstruct the spectrogram – compute the projection on the 25 Eigen vectors

An audio example

- The same spectrogram projected down to the 25 eigen vectors with the highest eigen values
  - Only the 25-dimensional weights are shown
  - The weights with which the 25 eigen vectors must be added to compose a least squares approximation to the spectrogram

Eigenvalues and Eigenvectors

- Left panel: Matrix with 1025 eigen vectors
- Right panel: Corresponding eigen values
  - Most Eigen values are close to zero
    - The corresponding eigenvectors are "unimportant"

An audio example

- The same spectrogram constructed from only the 25 Eigen vectors with the highest Eigen values
  - Looks similar
    - With 100 Eigenvectors, it would be indistinguishable from the original
  - Sounds pretty close
    - But now sufficient to store 25 numbers per vector (instead of 1024)
With only 5 eigenvectors

- The same spectrogram constructed from only the 5 Eigen vectors with the highest Eigen values
  - Highly recognizable

SVD vs. Eigen decomposition

- Singular value decomposition is analogous to the Eigen decomposition of the correlation matrix of the data
  - SVD: \( D = U S V^T \)
  - \( D^2 = U S V^T U V^T = U S^2 U^T \)

- The "left" singular vectors are the Eigen vectors of the correlation matrix
  - Show the directions of greatest importance

- The corresponding singular values are the square roots of the Eigen values of the correlation matrix
  - Show the importance of the Eigen vector

Correlation vs. Covariance Matrix

- Correlation:
  - The N Eigen vectors with the largest Eigen values represent the N greatest "energy-carrying" components of the matrix
  - Conversely, N "bases" that result in the least square error are the N best Eigen vectors
    - Projections onto these Eigen vectors retain the most energy

- Covariance:
  - the N Eigen vectors with the largest Eigen values represent the N greatest "variance-carrying" components of the matrix
  - Conversely, N "bases" that retain the maximum possible variance are the N best Eigen vectors

Thin SVD, compact SVD, reduced SVD

- SVD can be computed much more efficiently than Eigen decomposition
- Thin SVD: Only compute the first N columns of U
  - All that is required if \( N < M \)
- Compact SVD: Only the left and right singular vectors corresponding to non-zero singular values are computed

Eigenvectors, Eigenvalues and Covariances/Correlations

- The eigenvectors and eigenvalues (singular values) derived from the correlation matrix are important
- Do we need to actually compute the correlation matrix?
  - No
- Direct computation using Singular Value Decomposition

Why bother with Eigens/SVD

- Can provide a unique insight into data
  - Strong statistical grounding
  - Can display complex interactions between the data
  - Can uncover irrelevant parts of the data we can throw out
- Can provide basis functions
  - A set of elements to compactly describe our data
  - Indispensable for performing compression and classification
- Used over and over and still perform amazingly well
**Trace**

\[
A = \begin{bmatrix}
  a_{11} & a_{12} & a_{13} & a_{14} \\
  a_{21} & a_{22} & a_{23} & a_{24} \\
  a_{31} & a_{32} & a_{33} & a_{34} \\
  a_{41} & a_{42} & a_{43} & a_{44}
\end{bmatrix}
\]

\[
\text{Tr}(A) = a_{11} + a_{22} + a_{33} + a_{44}
\]

- The trace of a matrix is the sum of the diagonal entries.
- It is equal to the sum of the Eigen values:

\[
\text{Tr}(A) = \sum a_{ii} = \sum \lambda_i
\]

**Decompositions of matrices**

- Square A: LU decomposition
  - Decompose A = LU
  - L is a lower triangular matrix
  - All elements above diagonal are 0
  - R is an upper triangular matrix
  - All elements below diagonal are zero
  - Cholesky decomposition: A is symmetric, L = U^T

- QR decompositions: A = QR
  - Q is orthogonal: QQ^T = I
  - R is upper triangular

- Generally used as tools to compute Eigen decomposition or least square solutions

**Making vectors and matrices in MATLAB**

- Make a row vector: \( a = [1 \ 2 \ 3] \)
- Make a column vector: \( a = [1; 2; 3] \)
- Make a matrix: \( A = [1 \ 2 \ 3; 4 \ 5 \ 6] \)
- Combine vectors: \( A = [b \ c] \text{ or } A = [b; c] \)
- Make a random vector/matrix: \( c = \text{rand}(n,n) \)
- Make an identity matrix: \( I = \text{eye}(n) \)
- Make a sequence of numbers: \( c = 1:10 \) or \( c = 1:0.5:10 \) or \( c = 10:1:100 \)
- Make a ramp: \( c = \text{linspace}(0, 1, 100) \)

**Indexing**

- To get the \( i \)-th element of a vector: \( a(i) \)
- To get the \( i \)-th \( j \)-th element of a matrix: \( A(i,j) \)
- To get from the \( i \)-th to the \( j \)-th element: \( a(i:j) \)
- To get a sub-matrix: \( A(i:j, k:l) \)
- To get segments: \( a([i:j \ k:l \ m]) \)

**Properties of a Trace**

- Linearity: \( \text{Tr}(A+B) = \text{Tr}(A) + \text{Tr}(B) \)
  \( \text{Tr}(cA) = c \cdot \text{Tr}(A) \)

- Cycling invariance:
  - \( \text{Tr}(ABCD) = \text{Tr}(DABC) = \text{Tr}(CDAB) = \text{Tr}(BCDA) \)
  - \( \text{Tr}(AB) = \text{Tr}(BA) \)

- Frobenius norm: \( F(A) = \sum_{i,j} a_{ij}^2 = \text{Tr}(AA^T) \)
Arithmetic operations

- Addition/Subtraction
  \[ C = A + B \text{ or } C = A - B \]
- Vector/Matrix multiplication
  \[ C = A \times B \]
  - Operands must match!
- Element-wise operations
  - Multiplication/division
    \[ C = A \times B \text{ or } C = A ./ B \]
  - Exponentiation
    \[ C = A.^B \]
  - Elementary functions
    \[ C = \sin(A) \text{ or } C = \sqrt{A}, \ldots \]

Getting help with functions

- The help function
  - Type `help` followed by a function name
- Things to try
  - `help help`
  - `help +`
  - `help eig`
  - `help svd`
  - `help plot`
  - `help bar`
  - `help imagesc`
  - `help surf`
  - `help ops`
  - `help matfun`
- Also check out the tutorials and the mathworks site

Linear algebra operations

- Transposition
  \[ C = A' \]
  - If A is complex also conjugates use \[ C = A.' \] to avoid that
- Vector norm
  \[ \text{norm}(x) \] (also works on matrices)
- Matrix inversion
  \[ C = \text{inv}(A) \text{ if } A \text{ is square} \]
  \[ C = \text{pinv}(A) \text{ if } A \text{ is not square} \]
  - A might not be invertible, you'll get a warning if so
- Eigenanalysis
  \[ [u,d] = \text{eig}(A) \]
  - u is a matrix containing the eigenvectors
  - d is a diagonal matrix containing the eigenvalues
- Singular Value Decomposition
  \[ [u,s,v] = \text{svd}(A) \text{ or } [u,s,v] = \text{svd}(A,0) \]
  - "thin" versus regular SVD
  - s is diagonal and contains the singular values

Plotting functions

- 1-d plots
  \[ \text{plot}(x) \]
  - If x is a vector will plot all its elements
  - If x is a matrix will plot all its columns
  \[ \text{bar}(x) \]
  - Ditto but makes a bar plot
- 2-d plots
  \[ \text{imagesc}(x) \]
  - plots a matrix as an image
  \[ \text{surf}(x) \]
  - makes a surface plot