Overview

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

Central Projection

\[ \begin{align*}
\frac{x}{x'} &= \frac{y}{y'} = \frac{z}{z'} \\
\end{align*} \]

- The positions on the “window” are scaled along the line
- To compute \((x,y)\) position on the window, we need \(z\) (distance of window from eye), and \((x',y',z')\) (location being projected)

Homogeneous Coordinates

- Represent points by a triplet
  - Using yellow window as reference:
    - \((x,y) = (x,y,1)\)
    - \((x',y') = (x,y,c')\) \(c' = \frac{a}{a'}\)
- Locations on line generally represented as \((x,y,c)\)
  \[ \begin{align*}
  x' &= \frac{x}{c'} \\
  y' &= \frac{y}{c'} \\
  \end{align*} \]

Homogeneous Coordinates in 3-D

- Points are represented using FOUR coordinates
  - \((X,Y,Z,c)\)
  - “c” is the “scaling” factor that represents the distance of the actual scene
- Actual Cartesian coordinates:
  - \(X_{\text{actual}} = X/c, Y_{\text{actual}} = Y/c, Z_{\text{actual}} = Z/c\)
Orthogonal/Orthonormal vectors

- Two vectors are orthogonal if they are perpendicular to one another
  - \( \mathbf{A} \cdot \mathbf{B} = 0 \)
  - 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: \( \mathbf{A} \mathbf{A}^T = \mathbf{I} \)
  - 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 matrices
- If the entries of the matrix are not unit length, it cannot be orthogonal
  - \( \mathbf{A} \mathbf{A}^T = \mathbf{I} \) or \( \mathbf{A} \mathbf{A}^T = \mathbf{I} \), but not both
  - \( \mathbf{A} \mathbf{A}^T = \text{Diagonal} \) or \( \mathbf{A} \mathbf{A}^T = \text{Diagonal} \), but not both
  - If all the entries are the same length, we can get \( \mathbf{A} \mathbf{A}^T = \mathbf{A} \mathbf{A}^T = \text{Diagonal} \), though
  - A non-square matrix cannot be orthogonal
- \( \mathbf{A} \mathbf{A}^T = \mathbf{I} \), 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

Projections are often examples of rank-deficient transforms

- Projections are often examples of rank-deficient transforms
- Such matrices can never be orthogonal
- They can be used to eliminate dimensions
- Each projection matrix is symmetric
- The greatest number of independent projections is equal to the rank
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

4 Sep 2012 17

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

4 Sep 2012 15

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

4 Sep 2012 16

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!

4 Sep 2012 14

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

4 Sep 2012 15

Matrix Determinant

- The determinant is the “volume” of a matrix
- Actually the volume of a parallelepiped formed from its row vectors
  - Also the volume of the parallelepiped formed from its column vectors
- Standard formula for determinant: in textbook

4 Sep 2012 13
Matrix Determinant: Another Perspective

- The determinant is the ratio of N-volumes
  - If \( V_2 \) is the volume of an N-dimensional object "O" in N-dimensional space
  - \( O \) is the complete set of points or vertices that specify the object
  - If \( V_1 \) is the volume of the N-dimensional object specified by \( A \* O \), where \( A \) is a matrix that transforms the space
  - \( |A| = V_2 / V_1 \)

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-volumed 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-volumed 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
    \[ (A \cdot B)^T = B^T \cdot A^T \]

Determinant 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 ≠ sum of Volumes
  \[ |B + C| \neq |B| + |C| \]
- 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

Matrix Inversion

- The product of a matrix and its inverse is the identity matrix
  - Transforming an object, and then inverse transforming it gives us back the original object
  \[ T^{-1} \cdot T = T \cdot T^{-1} = I \]
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

Revisiting Projections and Least Squares

- Projection computes a least squared error estimate
- For each vector V in the music spectrogram matrix
  - Approximation: \( V_{\text{approx}} = a^*\text{note1} + b^*\text{note2} + c^*\text{note3} \).
  - Error vector \( E = V - V_{\text{approx}} \)
  - Squared error energy for \( V \): \( e(V) = \|E\|^2 \)
  - Projection computes \( V_{\text{approx}} \) 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

- Explanation with multiple notes
- How about the other way?
Pseudo-inverse (PINV)

- Pinv() applies to non-square matrices
- Pinv ( Pinv (A))) = A
- A*Pinv(A)= projection matrix!
  - Projection onto the columns of A
- If A = K x N matrix and K > N, A projects N-D vectors into a higher-dimensional K-D space
  - Pinv(A) = NxK matrix
  - Pinv(A)*A = I in this case
- Otherwise A * Pinv(A) = I

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!
- If matrix not square use a matrix pseudoinverse:
- MATLAB syntax: inv(a), pinv(a)

Eigenanalysis

- If something can go through a process mostly unscathed in character it is an eigen-something
  - Sound example: Vectors (black) that can undergo a transformation and keep pointing the same way is an eigenvector
- 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

EigenVectors and EigenValues

- Vectors that do not change angle upon transformation
  - They may change length
  - V = eigen vector
  - \lambda = eigen value
  - Matlab ab: [V, L] = eig(A)
  - L is a diagonal matrix whose entries are the eigen values
  - V is a matrix whose columns are the eigen vectors

Eigen vector example

Matrix multiplication revisited

- 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

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
    - 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

- 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 major and minor axes of the transformed ellipse define the ellipse
  - They are at right angles
- These are transformations of right-angled vectors on the original circle!

**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 \( A \)
- The smallest singular value is the smallest amount by which a vector is scaled by \( A \)
  - \( \min \left\{ \frac{||Ax||}{||x||} \right\} = s_{\text{min}} \)
  - This can be 0 (for low-rank or non-square matrices)

**Singular Value Decomposition**

- \( U \) and \( V \) are orthonormal matrices
- Columns are orthonormal vectors
- \( S \) is a diagonal matrix
- The right singular vectors of \( V \) 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**

\[
A = U S V^T
\]

```
[U,S,V] = svd(A)
```

**The Singular Values**

- Square matrices: The product of the singular values is the 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

**Symmetric Matrices**

- Matrices that do not change on transposition
  - Row and column vectors are identical
- The left and right singular vectors are identical
  - \( U = V \)
  - \( A = U S U^T \)
  - They are identical to the eigen vectors of the matrix

**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
  - Eigen vectors are always orthogonal
  - At 90 degrees to one another
Symmetric Matrices

- Eigen vectors point in the direction of the major and minor axes of the ellipsoid resulting from the transformation of a spheroid.
- The eigenvalues are the lengths of the axes.

The Correlation and Covariance Matrices

- Consider a set of column vectors represented as a D x N matrix A.
- The correlation matrix is:
  \[ C = \frac{1}{N} A^T A \]
- Diagonal elements represent the average 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.

Correlation / Covariance Matrix

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

- The correlation / covariance matrix is symmetric.
- Has orthonormal eigen vectors and real, non-negative eigen values.
- The square root of a correlation or covariance matrix is easily derived from the eigen vectors and eigen values.
- The eigen values of the square root of the covariance matrix are the square roots of the eigen values of the covariance matrix.
- These are also the “singular values” of the data set.

Square root of the Covariance Matrix

- The square root of the covariance matrix represents the elliptical scatter of the data.
- The eigenvectors of the matrix represent the major and minor axes.

The Correlation Matrix

- Any vector \( V = \lambda_1 \cdot \text{eigvec}_1 + \lambda_2 \cdot \text{eigvec}_2 + \ldots \)
- \( \lambda_1, \lambda_2, \) etc.
- 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.
An audio example

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

Eigen Reduction

\[ M = \text{spectrogram} \]
\[ C = MM^T \]
\[ V = \text{eig}(C) \]
\[ M_{\text{reduced}} = \text{reduced}(V_{\text{25}})M \]

- 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

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"

Vec = \[ a_1 \cdot \text{eigenvec1} + a_2 \cdot \text{eigenvec2} + a_3 \cdot \text{eigenvec3} \ldots \]

- 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 \( a \) is proportional to the square root of the eigenvalue
  - Ignoring these will not affect the composition of the spectrogram

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

- 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 eigenvectors with the highest eigenvalues
  - Highly recognizable

Correlation vs. Covariance Matrix

- Correlation:
  - The \(N\) eigenvectors with the largest eigenvalues represent the \(N\) greatest "energy-carrying" components of the matrix
  - Highly recognizable

- Covariance:
  - The \(N\) eigenvectors with the largest eigenvalues represent the \(N\) greatest "variance-carrying" components of the matrix

- Conclusively, \(N\) "bases" that retain the maximum possible variance are the \(N\) best eigenvectors

Eigenvectors, Eigenvalues and Covariances

- 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

SVD vs. Eigen decomposition

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

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

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

Thin SVD, compact SVD, reduced SVD

- 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

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
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
  \[ A = [b; c] \] or \[ A = [b; c] \]
- Make a random vector/matrix:
  \[ r = rand(m,n) \]
- Make an identity matrix:
  \[ I = eye(n) \]
- Make a sequence of numbers
  \[ c = 1:10 \] or \[ c = 1:0.5:10 \] or \[ c = 100:-2:50 \]
- Make a ramp
  \[ c = linspace(0, 1, 100) \]

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
  \[ a([i:j k:l m]) \]

Arithmetic operations

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

Linear algebra operations

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

Plotting functions

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

Getting help with functions

- The \text{help} function
  \( \text{Type help followed by a function name} \)
- Things to try
  \[ \text{help help} \]
  \[ \text{help +} \]
  \[ \text{help eig} \]
  \[ \text{help svd} \]
  \[ \text{help plot} \]
  \[ \text{help bar} \]
  \[ \text{help imagesc} \]
  \[ \text{help surf} \]
  \[ \text{help ops} \]
  \[ \text{help matfun} \]
- Also check out the tutorials and the mathworks site