Machine Learning for Signal Processing

Fundamentals of Linear Algebra - 2

Class 3. 8 Sep 2015

Instructor: Bhiksha Raj
Overview

• Vectors and matrices
• Basic vector/matrix operations
• Various matrix types
• Projections

• More on matrix types
• Matrix determinants
• Matrix inversion
• Eigenanalysis
• Singular value decomposition
• Matrix Calculus
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

\[
A = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad B = \begin{bmatrix} u \\ v \\ w \end{bmatrix}
\]

\[
A \cdot B = 0 \quad \Rightarrow \quad xu + yv + zw = 0
\]
Orthogonal matrices

- Orthogonal Matrix: $AA^T = A^TA = I$
  - 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)

\[
\begin{pmatrix}
\sqrt{0.5} & -\sqrt{0.125} & \sqrt{0.375} \\
\sqrt{0.5} & \sqrt{0.125} & -\sqrt{0.375} \\
0 & \sqrt{0.75} & 0.5
\end{pmatrix}
\]
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
Orthogonal and Orthonormal Matrices

\[
\begin{bmatrix}
1 & -\sqrt{0.0675} & \sqrt{0.1875} \\
\sqrt{0.5} & \sqrt{0.125} & -\sqrt{0.375} \\
0 & \sqrt{0.75} & 0.5
\end{bmatrix}
\]

• If the vectors in the matrix are not unit length, it cannot be orthogonal
  – \( AA^T \neq I, \ A^T A \neq I \)
  – \( AA^T = \text{Diagonal} \) 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
  – \( AA^T = I \) or \( A^T A = I \), but not both
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 \textit{rank deficient} matrices
- The rank of the matrix is the dimensionality of the transformed version of a full-dimensional object

\[
P = \begin{bmatrix}
1.0000 & 0 & 0 \\
0 & 0.2500 & -0.4330 \\
0 & -0.4330 & 0.7500
\end{bmatrix}
\]

\[
P_2 = \begin{bmatrix}
0.5000 & -0.2500 & 0.4330 \\
-0.2500 & 0.1250 & -0.2165 \\
0.4330 & -0.2165 & 0.3750
\end{bmatrix}
\]

Rank = 2

Rank = 1
Projections are often examples of rank-deficient transforms

\[ P = W \left( W^T W \right)^{-1} W^T \; ; \text{Projected Spectrogram} = P \cdot M \]

- 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 \textit{linearly independent} bases
  - Rank of P is 4
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

\[
\begin{bmatrix}
  x_1 & x_2 & \ldots & x_N \\
  y_1 & y_2 & \ldots & y_N \\
\end{bmatrix}
\]
\[
\begin{bmatrix}
  .8 & .9 \\
  .1 & .9 \\
  .6 & 0 \\
\end{bmatrix}
\]
\[
\begin{bmatrix}
  \hat{x}_1 & \hat{x}_2 & \ldots & \hat{x}_N \\
  \hat{y}_1 & \hat{y}_2 & \ldots & \hat{y}_N \\
  \hat{z}_1 & \hat{z}_2 & \ldots & \hat{z}_N \\
\end{bmatrix}
\]

\[X = 2D \text{ data} \quad P = \text{transform} \quad PX = 3D, \text{ rank 2}\]
Non-square Matrices

• Non-square matrices add or subtract axes
  – More rows than columns \( \rightarrow \) add axes
    • But does not increase the dimensionality of the data
  – Fewer rows than columns \( \rightarrow \) reduce axes
    • May reduce dimensionality of the data

\[
X = \begin{bmatrix}
  x_1 & x_2 & \cdots & x_N \\
  y_1 & y_2 & \cdots & y_N \\
  z_1 & z_2 & \cdots & z_N \\
\end{bmatrix}
\]

\( X = 3D \) data, rank 3

\[
P = \begin{bmatrix}
  .3 & 1 & .2 \\
  .5 & 1 & 1 \\
\end{bmatrix}
\]

\( P = \) transform

\[
PX = \begin{bmatrix}
  \hat{x}_1 & \hat{x}_2 & \cdots & \hat{x}_N \\
  \hat{y}_1 & \hat{y}_2 & \cdots & \hat{y}_N \\
\end{bmatrix}
\]

\( PX = 2D, \) rank 2
The Rank of a Matrix

- The matrix rank is the dimensionality of the transformation of a full-dimensioned 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.

\[
\begin{bmatrix}
.3 & 1 & .2 \\
.5 & 1 & 1
\end{bmatrix}
\]

\[
\begin{bmatrix}
.8 & .9 \\
.1 & .9 \\
.6 & 0
\end{bmatrix}
\]
The Rank of Matrix

- Projected Spectrogram = \( P \times M \)
  - Every vector in it is a combination of only 4 bases
- The rank of the matrix is the \textit{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 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 text book
Matrix Determinant: Another Perspective

Volume = $V_1$

Volume = $V_2$

\[
\begin{bmatrix}
0.8 & 0 & 0.7 \\
1.0 & 0.8 & 0.8 \\
0.7 & 0.9 & 0.7 \\
\end{bmatrix}
\]

- The determinant is the ratio of N-volumes
  - If $V_1$ is the volume of an N-dimensional sphere “O” in N-dimensional space
    - O is the complete set of points or vertices that specify the object
  - If $V_2$ is the volume of the N-dimensional ellipsoid 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 \(\neq\) 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

\[
T = \begin{bmatrix}
0.8 & 0 & 0.7 \\
1.0 & 0.8 & 0.8 \\
0.7 & 0.9 & 0.7 \\
\end{bmatrix}
\]

\[
Q = \begin{bmatrix}
? & ? & ? \\
? & ? & ? \\
? & ? & ? \\
\end{bmatrix} = T^{-1}
\]
Matrix Inversion

The product of a matrix and its inverse is the identity matrix

\[ T^{-1}TD = D \Rightarrow T^{-1}T = I \]

- 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

\[ TT^{-1}D = D \Rightarrow TT^{-1} = I \]
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
Rank Deficient Matrices

The projection matrix is rank deficient

You cannot recover the original spectrogram from the projected one.
Revisiting Projections and Least Squares

• Projection computes a least squared error estimate
• For each vector V in the music spectrogram matrix
  – Approximation: $V_{\text{approx}} = a*\text{note1} + b*\text{note2} + c*\text{note3}..$
  
  \[
  T = \begin{bmatrix}
  \text{note1} \\
  \text{note2} \\
  \text{note3}
  \end{bmatrix}
  \quad \quad \quad \quad
  V_{\text{approx}} = T \begin{bmatrix}
  a \\
  b \\
  c
  \end{bmatrix}
  
  \]
  
  – Error vector $E = V - V_{\text{approx}}$
  – Squared error energy for $V$ $e(V) = \text{norm}(E)^2$

• 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)

\[ V_{\text{approx}} = T \begin{bmatrix} a \\ b \\ c \end{bmatrix} \Rightarrow V \approx T \begin{bmatrix} a \\ b \\ c \end{bmatrix} = PINV(T) * V \]

- 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 give 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: \( P = W (W^T W)^{-1} W^T \), Projected Spectrogram = \( P \times M \)

Approximation: \( M = W \times X \)

The amount of \( W \) in each vector = \( X = PINV(W) \times M \)

\( W \times PINV(W) \times M = \) Projected Spectrogram

\( PINV(W) = (W^T W)^{-1} W^T \)
Explanation with multiple notes

\[ X = \text{Pinv}(W) \times M; \quad \text{Projected matrix} = W \times X = W \times \text{Pinv}(W) \times M \]
How about the other way?

\[ W = M \mathbf{P}^{-\text{inv}}(V) \]
\[ U = WV \]

- \( WV \approx M \)
- \( W = M \mathbf{P}^{-\text{inv}}(V) \)
- \( U = WV \)
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!
  \[ A \cdot B = C, \quad A = C \cdot B^{-1}, \quad B = A^{-1} \cdot C \]
• If matrix is not square use a matrix pseudoinverse:
  \[ A \cdot B \approx C, \quad A = C \cdot B^{+}, \quad B = A^{+} \cdot C \]
Eigenanalysis

- If something can go through a process mostly unscathed in character it is an *eigen*-something
  - Sound example: 🎧 🎧 🎧 🎧

- 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

$$ MV = \lambda V $$

- $V$ = eigen vector
- $\lambda$ = eigen value

$$ M = \begin{bmatrix} 1.5 & -0.7 \\ -0.7 & 1.0 \end{bmatrix} $$
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

\[ A = \begin{bmatrix} 1.0 & -0.07 \\ -1.1 & 1.2 \end{bmatrix} \]
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
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
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
Physical interpretation of eigen vector

\[
V = \begin{bmatrix} V_1 & V_2 \end{bmatrix} \\
\Lambda = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \\
M = V\Lambda V^{-1}
\]

- 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
    
    \[
    R_\theta = \begin{bmatrix}
    \cos \theta & -\sin \theta \\
    \sin \theta & \cos \theta
    \end{bmatrix}
    \]
    
    \[
    X = \begin{bmatrix}
    x \\
    y
    \end{bmatrix}
    \]
    
    \[
    X_{new} = \begin{bmatrix}
    x' \\
    y'
    \end{bmatrix}
    \]
    
    – 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?

\[
A = \begin{bmatrix} 1.0 & -0.07 \\ -1.1 & 1.2 \end{bmatrix}
\]
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!

\[ A = \begin{bmatrix} 1.0 & -0.07 \\ -1.1 & 1.2 \end{bmatrix} \]
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 $S$

$$A = \begin{bmatrix} 1.0 & -0.07 \\ -1.1 & 1.2 \end{bmatrix}$$

matlab:

```matlab
[U, S, V] = svd(A)
A = U * S * V'
```
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 $A$
  – $\text{Max} \left( \frac{|Ax|}{|x|} \right) = s_{\text{max}}$

• The smallest singular value is the smallest amount by which a vector is scaled by $A$
  – $\text{Min} \left( \frac{|Ax|}{|x|} \right) = s_{\text{min}}$
  – 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
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 $V$

- $A A^T$ : Converts $V$ to $U$, then brings it back to $V$
  - Result: Only scaling

$$ A = \begin{bmatrix} .7 & 0 \\ -0.1 & 1 \end{bmatrix} $$
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 do not rotate the space
- Only scaling and, if Eigen values are negative, reflection
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

\[
\begin{bmatrix}
1.5 & -0.7 \\
-0.7 & 1
\end{bmatrix}
\]
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 eigen values are the lengths of the axes
Symmetric matrices

- Eigen vectors $V_i$ are orthonormal
  - $V_i^T V_i = 1$
  - $V_i^T V_j = 0, \ i \neq j$

- Listing all eigen vectors in matrix form $V$
  - $V^T = V^{-1}$
  - $V^T V = I$
  - $V V^T = I$

- $M V_i = \lambda V_i$

- 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

\[ C = V \Lambda V^T \]

\[ \sqrt{C} = V \cdot \sqrt{\Lambda} \cdot V^T \]

\[ \sqrt{C} \cdot \sqrt{C} = V \cdot \sqrt{\Lambda} \cdot V^T \cdot V \cdot \sqrt{\Lambda} \cdot V^T \]

\[ = V \cdot \sqrt{\Lambda} \cdot \sqrt{\Lambda} V^T = V\Lambda V^T = C \]
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 these and 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
  \[ C = \frac{1}{N} A A^T \]
  - Represents the directions in which the “energy” in the signal lies
• If the average (mean) of the vectors in A is subtracted out of all vectors, C is the **covariance** matrix
  \[ \text{covariance} = \text{correlation} + \text{mean} * \text{mean}^T \]
  - Represents the directions in which the “spread” of the signal lies
• Diagonal elements represent the energy/spread of individual components
  - Off diagonal elements represent how two components are related
    • How much knowing one lets us guess the value of the other

\[ (1/N)\sum_i a_{1,i}^2 \]
\[ (1/N)\sum_i a_{k,i} a_{k,j} \]
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.
  - “Modes” in direction of scatter.
The Correlation Matrix

Any vector \( V = a_{V,1} \times \text{eigenvec1} + a_{V,2} \times \text{eigenvec2} + \ldots \)

\[ \sum V a_{V,i} = \text{eigenvalue}(i) \]

- 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

\[ M = \text{spectrogram} \quad 1025 \times 1000 \]

\[ C = M \cdot M^T \quad 1025 \times 1025 \]

\[ V = 1025 \times 1025 \]

\[ [V, L] = \text{eig}(C) \]

\[ V_{\text{reduced}} = [V_1 \quad \ldots \quad V_{25}] \quad 1025 \times 25 \]

\[ M_{\text{lowdim}} = P\text{inv}(V_{\text{reduced}})M \quad 25 \times 1000 \]

\[ M_{\text{reconstructed}} = V_{\text{reduced}} M_{\text{lowdim}} \quad 1025 \times 1000 \]

- 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”

\[ M = \text{spectrogram} \]
\[ C = M \cdot M^T \]
\[ [V, L] = \text{eig} (C) \]
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

$$\text{Vec} = a_1 \cdot \text{eigenvec1} + a_2 \cdot \text{eigenvec2} + a_3 \cdot \text{eigenvec3} \ldots$$
An audio example

\[ V_{reduced} = [V_1 \ldots V_{25}] \]
\[ M_{lowdim} = Pinv(V_{reduced})M \]

- 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
An audio example

\[ M_{\text{reconstructed}} = V_{\text{reduced}} M_{\text{lowdim}} \]

- 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)
SVD vs. Eigen decomposition

- SVD cannot in general be derived directly from the Eigen analysis and vice versa
- But for matrices of the form $M = DD^T$, the Eigen decomposition of $M$ is related to the SVD of $D$
  - SVD: $D = U S V^T$
  - $DD^T = U S V^T V S U^T = U S^2 U^T$

- The “left” singular vectors are the Eigen vectors of $M$
  - Show the directions of greatest importance

- The corresponding singular values of $D$ are the square roots of the Eigen values of $M$
  - Show the importance of the Eigen vector
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
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

Eigenfaces
Using a linear transform of the above “eigenvectors” we can compose various faces
Trace

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

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

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

\[
\text{Tr}(A) = \sum_{i} a_{i,i}
\]
Trace

- Often appears in Error formulae

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

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

\[
E = D - C \quad \text{error} = \sum_{i,j} E^2_{i,j}
\]

- Useful to know some properties..
Properties of a Trace

• Linearity: \( \text{Tr}(A+B) = \text{Tr}(A) + \text{Tr}(B) \)
  \( \text{Tr}(c.A) = c.\text{Tr}(A) \)

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

• Frobenius norm \( F(A) = \sum_{i,j} a_{ij}^2 = \text{Tr}(AA^T) \)
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
Calculus of Matrices

• Derivative of scalar w.r.t. vector
• For any scalar $z$ that is a function of a vector $\mathbf{x}$
• The dimensions of $\frac{dz}{dx}$ are the same as the dimensions of $\mathbf{x}$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}$$

$$\frac{dz}{dx} = \begin{bmatrix} \frac{dz}{dx_1} \\ \vdots \\ \frac{dz}{dx_N} \end{bmatrix}$$

$N \times 1$ vector
Calculus of Matrices

• Derivative of scalar w.r.t. matrix
• For any scalar $z$ that is a function of a matrix $X$
• The dimensions of $\frac{dz}{dX}$ are the same as the dimensions of $X$

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \end{bmatrix}$$

$$\frac{dz}{dX} = \begin{bmatrix} \frac{dz}{dx_{11}} & \frac{dz}{dx_{12}} & \frac{dz}{dx_{13}} \\ \frac{dz}{dx_{21}} & \frac{dz}{dx_{22}} & \frac{dz}{dx_{23}} \end{bmatrix}$$

N x M matrix
Calculus of Matrices

• Derivative of vector w.r.t. vector
• For any Mx1 vector $y$ that is a function of an Nx1 vector $x$
• $\frac{dy}{dx}$ is an MxN matrix

$$y = \begin{bmatrix} y_1 \\ \vdots \\ y_M \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}, \quad \frac{dy}{dx} = \begin{bmatrix} \frac{dy_1}{dx_1} & \cdots & \frac{dy_1}{dx_N} \\ \vdots & \ddots & \vdots \\ \frac{dy_M}{dx_1} & \cdots & \frac{dy_M}{dx_N} \end{bmatrix}$$

M x N matrix
Calculus of Matrices

- Derivative of vector w.r.t. matrix
- For any Mx1 vector $y$ that is a function of an NxL matrix $X$
- $\frac{dy}{dX}$ is an MxLxN tensor (note order)

$y = \begin{bmatrix} y_1 \\ \vdots \\ y_M \end{bmatrix}$

$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \end{bmatrix}$

\[
\frac{dy}{dX} = \begin{bmatrix} \frac{dy_1}{dx_{1,j}} & \frac{dy_1}{dx_{2,j}} & \frac{dy_1}{dx_{3,j}} \\ \vdots & \vdots & \vdots \\ \frac{dy_M}{dx_{1,j}} & \frac{dy_M}{dx_{2,j}} & \frac{dy_M}{dx_{3,j}} \end{bmatrix}
\]

(i,j,k)th element = $\frac{dy_i}{dx_{k,j}}$
Calculus of Matrices

• Derivative of matrix w.r.t. matrix
• For any $M \times K$ vector $\mathbf{Y}$ that is a function of an $N \times L$ matrix $\mathbf{X}$
• $\frac{d\mathbf{Y}}{d\mathbf{X}}$ is an $M \times K \times L \times N$ tensor (note order)

$$\mathbf{Y} = \begin{bmatrix} y_{11} & y_{12} & y_{13} \\ y_{21} & y_{22} & y_{23} \end{bmatrix}$$

(i,j)th element $= \frac{dy_{11}}{dx_{j,i}}$
In general

• The derivative of an $N_1 \times N_2 \times N_3 \times \ldots$ tensor w.r.t to an $M_1 \times M_2 \times M_3 \times \ldots$ tensor

• Is an $N_1 \times N_2 \times N_3 \times \ldots \times M_L \times M_{L-1} \times \ldots \times M_1$ tensor
Compound Formulae

• Let \( Y = f ( g ( h ( X ) ) ) \)

• Chain rule (note order of multiplication)

\[
\frac{dY}{dX} = \frac{dh(X)}{dX} \# \frac{dg(h(X))}{dh(X)} \# \frac{df(g(h(X))}{dg(h(X))}
\]

• The \# represents a transposition operation
  – That is appropriate for the tensor
Example

\[ z = \| y - Ax \|^2 \]

- \( y \) is \( N \times 1 \)
- \( x \) is \( M \times 1 \)
- \( A \) is \( N \times M \)

- Compute \( d_z/dA \)
  - On board