Fundamentals of Linear Algebra

Class 2-3.  6 Sep 2011

Instructor: Bhiksha Raj
Administrivia

- **TA Times:**
  - Anoop Ramakrishna: Thursday 12.30-1.30pm
  - Manuel Tragut: Friday 11am – 12pm.

- **HW1:** On the webpage
Projections

- What would we see if the cone to the left were transparent if we looked at it along the normal to the plane?
  - The plane goes through the origin
  - Answer: the figure to the right
- How do we get this? Projection
Consider any plane specified by a set of vectors $W_1, W_2$.

- Or matrix $[W_1 \ W_2 \ldots]$
- Any vector can be projected onto this plane
- The matrix $A$ that rotates and scales the vector so that it becomes its projection is a projection matrix
Given a set of vectors $W_1$, $W_2$, which form a matrix $W = [W_1 \ W_2 \ldots]$.

The projection matrix that transforms any vector $X$ to its projection on the plane is $P = W (W^T W)^{-1} W^T$.

We will visit matrix inversion shortly.

Magic – any set of vectors from the same plane that are expressed as a matrix will give you the same projection matrix $P = V (V^T V)^{-1} V^T$. 
Projections

- HOW?
Projections

- Draw any two vectors $W_1$ and $W_2$ that lie on the plane
  - *ANY two so long as they have different angles*
- Compose a matrix $W = [W_1 \ W_2]$
- Compose the projection matrix $P = W \ (W^T W)^{-1} \ W^T$
- Multiply every point on the cone by $P$ to get its projection
- View it 😊
  - I’m missing a step here – what is it?

6 Sep 2011
The projection actually projects it onto the plane, but you’re still seeing the plane in 3D

- The result of the projection is a 3-D vector
- \( P = W (W^T W)^{-1} W^T = 3 \times 3, \ P \times \text{Vector} = 3 \times 1 \)
- The image must be rotated till the plane is in the plane of the paper
  - The Z axis in this case will always be zero and can be ignored
  - How will you rotate it? (remember you know W1 and W2)
Projection matrix properties

- The projection of any vector that is already on the plane is the vector itself
  - $P_x = x$ if $x$ is on the plane
  - If the object is already on the plane, there is no further projection to be performed

- The projection of a projection is the projection
  - $P(P_x) = P_x$
  - That is because $P_x$ is already on the plane

- Projection matrices are *idempotent*
  - $P^2 = P$
  - Follows from the above
Perspective

- The picture is the equivalent of “painting” the viewed scenery on a glass window

- Feature: The lines connecting any point in the scenery and its projection on the window merge at a common point
  - The eye
An aside on Perspective..

- Perspective is the result of convergence of the image to a point
- Convergence can be to multiple points
  - Top Left: One-point perspective
  - Top Right: Two-point perspective
  - Right: Three-point perspective
Central Projection

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).

\[
\frac{x}{x'} = \frac{y}{y'} = \frac{z}{z'} \quad \text{Property of a line through origin}
\]

\[
\alpha = \frac{z}{z'} \quad x = \alpha x' \\
y = \alpha y'
\]
Projections: A more physical meaning

- Let $W_1, W_2 \ldots W_k$ be “bases”
- We want to explain our data in terms of these “bases”
  - We often cannot do so
  - But we can explain a significant portion of it

- The portion of the data that can be expressed in terms of our vectors $W_1, W_2, \ldots W_k$, is the projection of the data on the $W_1 \ldots W_k$ (hyper) plane
  - In our previous example, the “data” were all the points on a cone
  - The interpretation for volumetric data is obvious
Projection: an example with sounds

- The spectrogram (matrix) of a piece of music

- How much of the above music was composed of the above notes
  - I.e. how much can it be explained by the notes
Projection: one note

M = spectrogram; W = note

\[ P = W (W^T W)^{-1} W^T \]

Projected Spectrogram = \( P \ast M \)

M = The spectrogram (matrix) of a piece of music
Projection: one note – cleaned up

M =

- The spectrogram (matrix) of a piece of music

W =

- Floored all matrix values below a threshold to zero
Projection: multiple notes

\[ M = \ldots \]

- The spectrogram (matrix) of a piece of music

\[ W = \]

- \[ P = W (W^T W)^{-1} W^T \]
- Projected Spectrogram = \( P \times M \)
Projection: multiple notes, cleaned up

\[ M = \]

- The spectrogram (matrix) of a piece of music

\[ W = \]

- \[ P = W(W^TW)^{-1} W^T \]
- Projected Spectrogram = \( P \times M \)
Projection and Least Squares

- Projection actually computes a *least squared error* estimate
- For each vector \( V \) in the music spectrogram matrix
  - Approximation: \( V_{\text{approx}} = a*\text{note}1 + b*\text{note}2 + c*\text{note}3.. \)

\[
V_{\text{approx}} = \begin{bmatrix} \text{note}1 & \text{note}2 & \text{note}3 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}
\]

- Error vector \( E = V - V_{\text{approx}} \)
- Squared error energy for \( V \): \( e(V) = \| E \|^2 \)
- Total error = \( \sum_{V} e(V) \)

- Projection computes \( V_{\text{approx}} \) for all vectors such that Total error is minimized
  - It does not give you “a”, “b”, “c”.. Though
    - That needs a different operation – the inverse / pseudo inverse
Orthogonal and Orthonormal matrices

- **Orthogonal Matrix**: $A A^T =$ diagonal
  - Each row vector lies exactly along the normal to the plane specified by the rest of the vectors in the matrix

- **Orthonormal Matrix**: $A A^T = A^T A = I$
  - In addition to being orthogonal, each vector has length exactly $= 1.0$
  - Interesting observation: In a square matrix if the length of the row vectors is 1.0, the length of the column vectors is also 1.0

\[
\begin{bmatrix}
1 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
0.707 & -0.354 & 0.612 \\
0.707 & 0.354 & -0.612 \\
0 & 0.866 & 0.5 \\
\end{bmatrix}
\]
Orthogonal and Orthonormal Matrices

- Orthonormal matrices will retain the relative angles between transformed vectors
  - Essentially, they are combinations of rotations, reflections and permutations
  - Rotation matrices and permutation matrices are all orthonormal matrices
  - The vectors in an orthonormal matrix are at 90 degrees to one another.

- Orthogonal matrices are like Orthonormal matrices with stretching
  - The product of a diagonal matrix and an orthonormal matrix
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

\[ M = \begin{bmatrix} p_1 & p_2 & \cdots & p_n \end{bmatrix} \begin{bmatrix} w_1 & w_2 & \cdots & w_n \end{bmatrix} \begin{bmatrix} w_1 \ T & w_2 \ T & \cdots & w_n \ T \end{bmatrix}^{-1} \begin{bmatrix} w_1 \ T & w_2 \ T & \cdots & w_n \ T \end{bmatrix} \]

\[ W = \begin{bmatrix} w_1 & w_2 & \cdots & w_4 \end{bmatrix} \]

- \( P = W (W^T W)^{-1} W^T \); Projected Spectrogram = \( P \ast M \)
- 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 4 \textit{independent} bases
  - Rank of \( P \) is 4
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

$\begin{bmatrix} x_1 & x_2 & \ldots & x_N \\ y_1 & y_2 & \ldots & y_N \end{bmatrix}$

$\begin{bmatrix} .8 & .9 \\ .1 & .9 \\ .6 & 0 \end{bmatrix}$

$\begin{bmatrix} x_1 & x_2 & \ldots & x_N \\ y_1 & y_2 & \ldots & y_N \\ z_1 & z_2 & \ldots & z_N \end{bmatrix}$

$X = 2D$ data  \hspace{1cm} P = \text{transform}  \hspace{1cm} PX = 3D, \text{rank} 2$
Non-square Matrices

Non-square matrices add or subtract axes

- More rows than columns → add axes
- Fewer rows than columns → reduce axes
- May reduce dimensionality of the data

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

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

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

\(P = \text{transform}\)

\[
\begin{bmatrix}
  x_1 & x_2 & \ldots & x_N \\
  y_1 & y_2 & \ldots & y_N
\end{bmatrix}
\]

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

\[ M = \]

- 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-D object is compressed to a K-D object by a matrix, it will also be compressed to a K-D object by the transpose of the matrix.
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
The determinant is the ratio of N-volumes

- If $V_1$ 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 $V_2$ 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-D objects into zero-volume N-D 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-D 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!!!
    \[ A \cdot B \neq B \cdot A \]
  - \textit{left multiplications} \neq \textit{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
  \[ |B + C| \neq |B| + |C| \]
  The volume of the parallelepiped formed by row vectors of the sum of two matrices is not the sum of the volumes of the parallelepipeds formed by the original matrices

- Commutative for square matrices!!
  \[ |A \cdot B| = |B \cdot A| = |A| \cdot |B| \]
  The order in which you scale the volume of an object is irrelevant
Matrix Inversion

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

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

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
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: $V_{\text{approx}} = a \cdot \text{note1} + b \cdot \text{note2} + c \cdot \text{note3}$.

\[
T = \begin{bmatrix}
\text{note1} \\
\text{note2} \\
\text{note3}
\end{bmatrix} \quad V_{\text{approx}} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}
\]

- Error vector $E = V - V_{\text{approx}}$
- Squared error energy for $V$: $e(V) = \|E\|^2$
- Total error = Total error + $e(V)$

- Projection computes $V_{\text{approx}}$ for all vectors such that Total error is minimized

- *But WHAT ARE “a” “b” and “c”?*
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 \textit{pseudo inverse}.

- This give us a \textit{LEAST SQUARES} solution.

  - If \( T \) were square and invertible \( \text{Pinv}(T) = T^{-1} \), and \( V = V_{\text{approx}} \).
Explaining music with one note

$M = \text{PINV}(W) \cdot M$

$W = \text{Recap: } P = W \cdot (W^TW)^{-1} \cdot W^T, \text{Projected Spectrogram} = P \cdot M$

$\text{Approximation: } M = W \cdot X$

The amount of $W$ in each vector $= X = \text{PINV}(W) \cdot M$

$W \cdot \text{PINv}(W) \cdot M = \text{Projected Spectrogram}$

$\text{PINV}(W) = (W^TW)^{-1}W^T$

6 Sep 2011
M =

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

\[ M = \]

\[ V = \]

\[ W = ? \quad U = ? \]

- \( WV \approx M \)
  - \( W = M \times Pinv(V) \)
  - \( U = WV \)

6 Sep 2011

11-755/18-797
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)*A = I in this case
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
- Pay attention to multiplication side!
  \[ A \cdot B = C, \quad A = C \cdot B^{-1}, \quad B = A^{-1} \cdot C \]
- Matrix inverses defined for square matrices only
  - If matrix not square use a matrix pseudoinverse:
    \[ A \cdot B = C, \quad A = C \cdot B^+, \quad B = A^+ \cdot C \]
- MATLAB syntax: `inv(a), pinv(a)`
What is the Matrix?

- Duality in terms of the matrix identity
  - Can be a container of data
    - An image, a set of vectors, a table, etc ...
  - Can be a **linear** transformation
    - A process by which to transform data in another matrix

- We’ll usually start with the first definition and then apply the second one on it
  - Very frequent operation
  - Room reverberations, mirror reflections, etc ...

- Most of signal processing and machine learning are a matrix multiplication!
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

- Black vectors are eigen vectors

- Vectors that do not change angle upon transformation
  - They may change length

\[ \mathbf{M} \mathbf{V} = \lambda \mathbf{V} \]

- \( \mathbf{V} = \) eigen vector
- \( \lambda = \) eigen value

Matlab: \([\mathbf{V}, \mathbf{L}] = \text{eig}(\mathbf{M})\)
- \(\mathbf{L}\) is a diagonal matrix whose entries are the eigen values
- \(\mathbf{V}\) 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

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

\[ V = \begin{bmatrix} V_1 & V_2 \end{bmatrix} \]

\[ L = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \]

\[ M = VLV^{-1} \]

- 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 eigenvalues and eigenvectors.
  - E.g. consider a rotation matrix

\[ R_\theta = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \]

\[ X = \begin{bmatrix} x \\ y \end{bmatrix}, \quad X_{\text{new}} = \begin{bmatrix} x' \\ y' \end{bmatrix} \]

- This rotates every vector in the plane.
  - No vector that remains unchanged.

- In these cases the Eigen vectors and values are complex.

- Some matrices are special however..
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 right 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!
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**

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

**matlab:**

\[
[U, S, V] = \text{svd}(A)
\]
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 $A$
  - $\text{Max } (|Ax| / |x|) = s_{\text{max}}$

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

- **Square matrices:** The product of the singular values is the determinant of the matrix
  - This is also the product of the *eigen* values
  - 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

\[
\begin{bmatrix}
1.5 & -0.7 \\
-0.7 & 1
\end{bmatrix}
\]

- 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

\[
\begin{bmatrix}
  1.5 & -0.7 \\
  -0.7 & 1
\end{bmatrix}
\]

- 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

\[
\begin{bmatrix}
1.5 & -0.7 \\
-0.7 & 1
\end{bmatrix}
\]

- 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 $V$
  - $V^T = V^{-1}$
  - $V^T V = I$
  - $V V^T = I$

- $C V_i = \lambda V_i$

- In matrix form: $C V = V L$
  - $L$ is a diagonal matrix with all eigen values

- $C = V L V^T$
The Correlation and Covariance Matrices

Consider a set of column vectors represented as a DxN matrix M

The correlation matrix is

- $C = (1/N) MM^T$
  - If the average value (mean) of the vectors in M is 0, C is called the covariance matrix
  - covariance = correlation + mean * mean^T

Diagonal elements represent average value 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
Correlation / Covariance Matrix

\[ C = VLV^T \]

\[ Sqrt(C) = V \cdot Sqrt(L) \cdot V^T \]

\[ Sqrt(C) \cdot Sqrt(C) = V \cdot Sqrt(L) \cdot V^T \cdot V \cdot Sqrt(L) \cdot V^T = V \cdot Sqrt(L) \cdot Sqrt(L) \cdot V^T = VLV^T = C \]

- 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 Covariance Matrix

Any vector $V = a_{V,1} \cdot \text{eigenvec1} + a_{V,2} \cdot \text{eigenvec2} + ..$

$\Sigma_V a_{V,i} = \text{eigenvalue}(i)$

- 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

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

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

\[ V = 1025 \times 1025 \]

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

\[ V_{\text{reduced}} = \begin{bmatrix} V_1 & \cdots & V_{25} \end{bmatrix} \quad 1025 \times 25 \]

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

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

- Compute the Covariance/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”

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

(Vec = $a_1 \cdot \text{eigenvec}_1 + a_2 \cdot \text{eigenvec}_2 + a_3 \cdot \text{eigenvec}_3$ ...)
An audio example

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

- 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

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
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.
- The “right” 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.
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 eigen vectors corresponding to non-zero singular values are computed

- **Reduced SVD**: Only compute the columns of $U$ corresponding to the $K$ highest singular values
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