MLSP linear algebra refresher: 3

Course Projects

- Projects will be done by teams of students
  - Ideal team size: 4
  - Find yourself a team
  - If you wish to work alone, that is OK
    - But we will not require less of you for this
  - If you cannot find a team by yourselves, you will be assigned to a team
  - Teams will be listed on the website
  - All currently registered students will be put in a team eventually
- Will require background reading and literature survey
  - Learn about the problem

Projects

- Teams must be formed by 17th Friday
- Teams must send us a preliminary project proposal by 30th September 2021
  - Please send us proposals earlier, so that we can vet them
  - The later you start, the less time you will have to work on the project

Quality of projects

- Project must include aspects of signal analysis and machine learning
  - Prediction, classification or compression of signals
  - Using machine learning techniques
- Several projects from previous years have led to publications
  - Conference and journal papers
  - Best paper awards
  - Doctoral and Masters' dissertations

Projects from past years: 2020

- Denoising EEG signals
- Kalman filters for motion prediction: Simulating an automatic driver
- Detecting urban noises from spatio-temporal context
- Music source separation using overcomplete dictionary representations
- Mars in Living Color: Up-Sampling CRISM Spectra Using HiRISE Images
- Fast fractal image compression
- Predicting shipment arrival times
- Drunk speech detection

Projects from past years: 2015

- Loop querier – searching the rhythmic pattern
- Vision-based montecarlo localization for autonomous vehicle
- Beatbox to drum conversion
- City localization on flikr videos using only audio
- Facial landmarks based video frontalization and its application in face recognition
- Audioshop: Modifying and editing singing voice
- Predicting and classifying RF signal strength in an environment with obstacles
- Realtime detection of basketball players

Projects from past years: 2014

- IMPROVING SPATIALIZATION ON HEADPHONES FOR STEREO MUSIC
- PREDICTING THE OUTCOME OF ROULETTE
- MOOD BASED CLASSIFICATION OF SONGS TO IDENTIFY ACOUSTIC FEATURES THAT ALLEVIATE DEPRESSION
- PERSON IDENTIFICATION THROUGH FOOTSTEP-INDUCED FLOOR VIBRATION

Projects from past years: 2013

- Automotive vision localization
- Lyric recognition
- Imaging without a camera
- Handwriting recognition with a Kinect
- Soccer tracking
- Image manipulation using patch transforms
- Audio classification
- Foreground detection using adaptive mixture models

Projects from previous years: 2012

- Skin surface input interfaces
  - Chris Harrison
- Visual feedback for needle steering system
- Clothing recognition and search
- Time of flight countertop
  - Chris Harrison
- Non-intrusive load monitoring using an EMF sensor
  - Mario Berges
- Blind sidewalk detection
- Detecting abnormal ECG rhythms
- Shot boundary detection (in video)
- Change detection using SVD for ultrasonic pipe monitoring
- Detecting Bonobo vocalizations
  - Alan Black
- Kinect gesture recognition for musical control

Projects from previous years: 2011

- Spoken word detection using seam carving on spectrograms
  - Rita Singh
- Automatic annotation and evaluation of solfege
- Left ventricular segmentation in MR images using a conditional random field
- Non-intrusive load monitoring
  - Mario Berges
- Velocity detection of speeding automobiles from analysis of audio recordings
- Speech and music separation using probabilistic latent component analysis and constant-Q transforms

Project Complexity

- Depends on what you want to do
- Complexity of the project will be considered in grading.
- Projects typically vary from cutting-edge research to reimplementation of existing techniques. Both are fine.
- Only caveat : The term "deep learning" must not relate to your project
  - Absolutely no DL/Nnets

Project presentations

- You will make a five minute video and upload to YouTube and send us a link
- You also submit a report

- Your classmates and the TAs/Instructor will evaluate the project
- You must "defend" you project (online) during a "review" period
- Final scores are based on Instructor/TA and peer review scores

Incomplete Projects

- Be realistic about your goals.

- Incomplete projects can still get a good grade if
  - You can demonstrate that you made progress
  - You can clearly show why the project is infeasible to complete in one semester

- Remember: You will be graded by peers

“Local” Projects..

- Several project ideas routinely proposed by various faculty/industry partners
  - Sarnoff labs, NASA, Mitsubishi, Adobe..
- Local faculty
  - Alan Black is usually good for a project or two
  - Roger Dannenberg is a world leader in computational music
  - Mario Berges has helped in the past
  - Rita Singh does nice work on speech forensics
  - Jim Baker invented automatic speech recognition as we know it
  - Others…

Questions

Overview

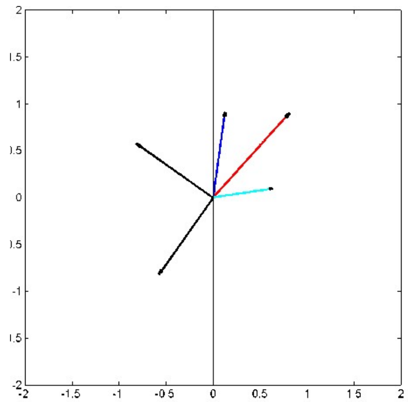
- Vectors and matrices
- Basic vector/matrix operations
- Various matrix types
- Matrix properties
  - Determinant
  - Inverse
  - Rank
- Solving simultaneous equations
- Projections
- **Eigen decomposition**
- **SVD**

Eigenanalysis

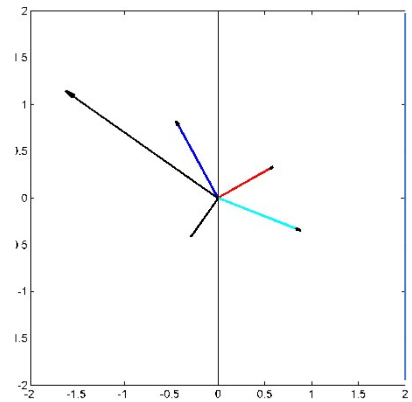
- If something can go through a process mostly unscathed in character it is an *eigen*-something
  – Sound example:
- A vector that can undergo a matrix multiplication and keep pointing the same way is an *eigenvector*
  – Its length can change though
- How much its length changes is expressed by its corresponding *eigenvalue*
  – Each eigenvector of a matrix has its eigenvalue
- Finding these “eigenthings” is called eigenanalysis

EigenVectors and EigenValues

Black vectors are eigen vectors



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

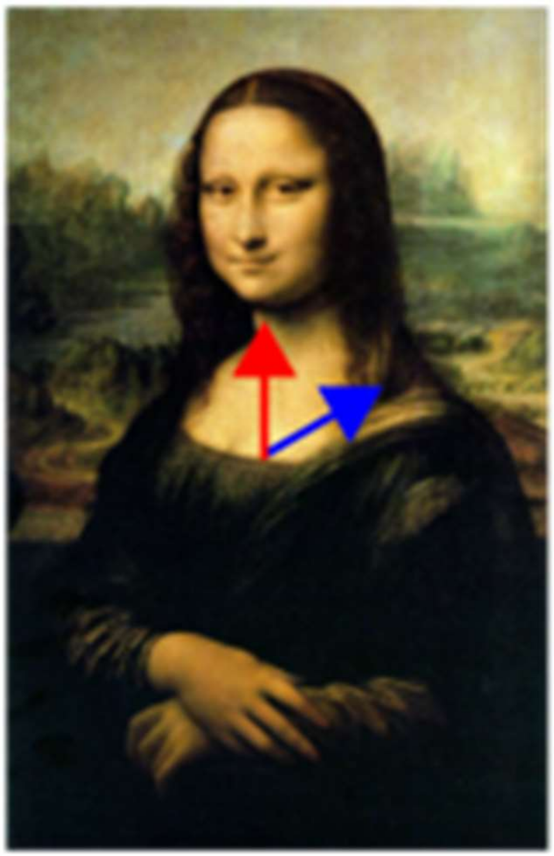


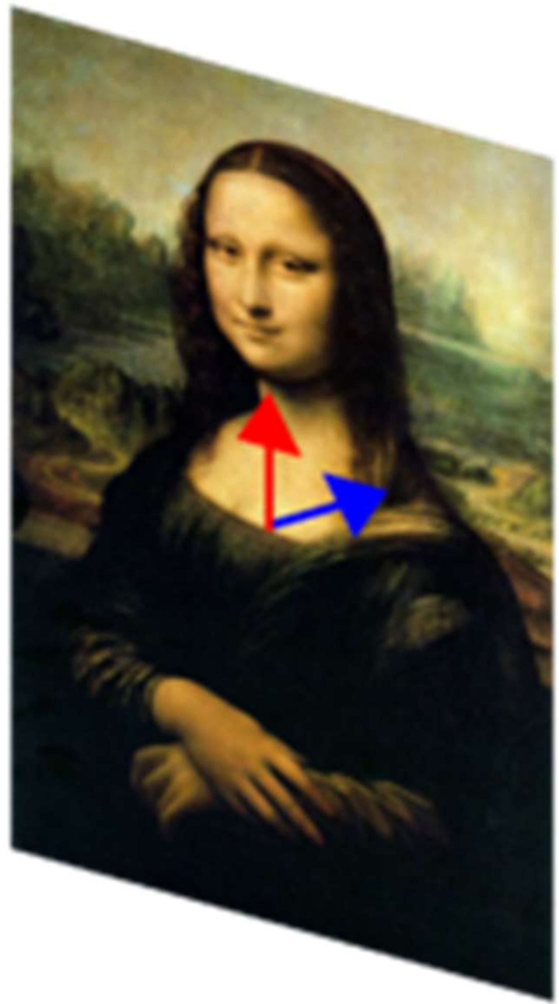
- Vectors that do not change angle upon transformation
  - They may change length

$$MV = \lambda V$$

  - V = eigen vector
  - λ = eigen value

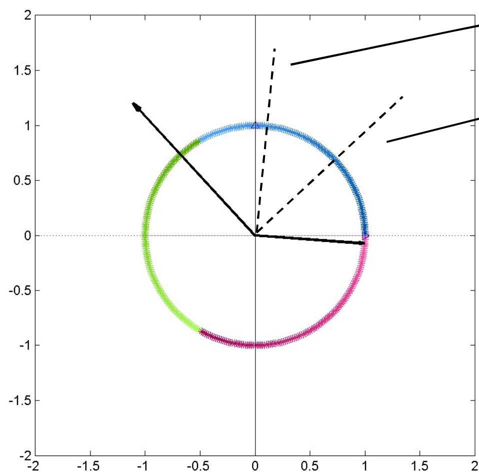
Eigen vector example



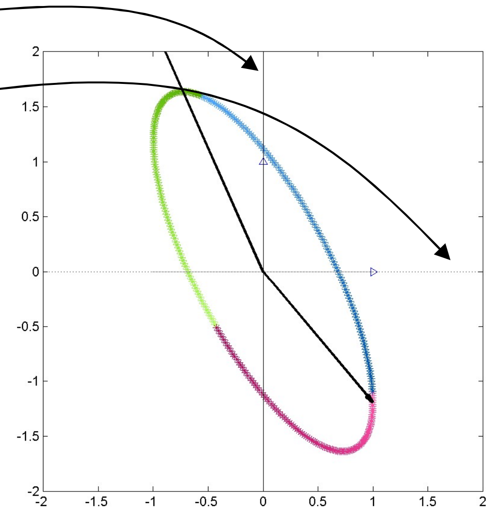


Defective matrix

Matrix multiplication revisited

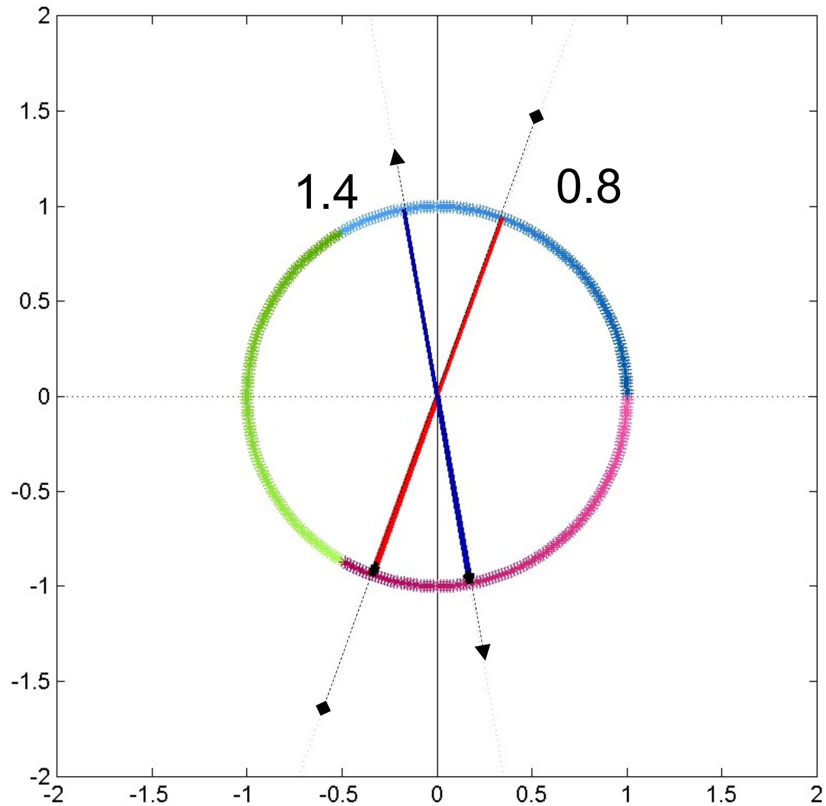


$$A = \begin{bmatrix} 1.0 & -0.07 \\ -1.1 & 1.2 \end{bmatrix}$$



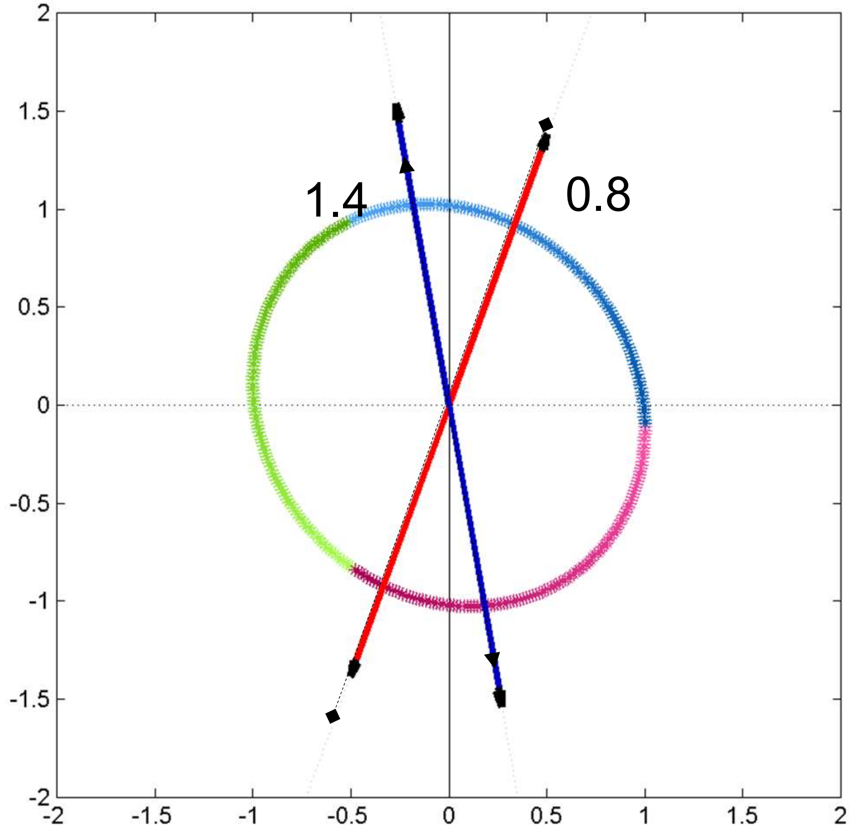
- Matrix transformation “transforms” the space
  - Warps the paper so that the normals to the two vectors now lie along the axes

A stretching operation



- Draw two lines
- Stretch / shrink the paper along these lines by factors λ_1 and λ_2
  - The factors could be negative – implies flipping the paper
- The result is a transformation of the space

A stretching operation



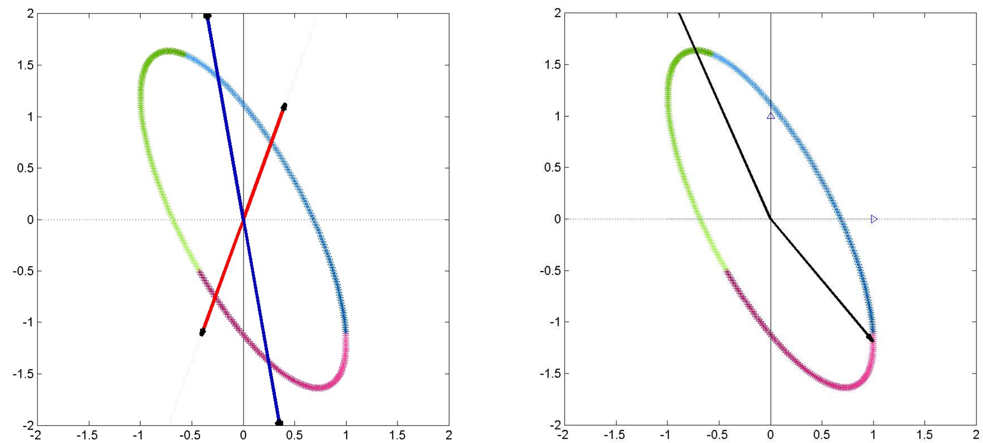
- Draw two lines
- Stretch / shrink the paper along these lines by factors λ_1 and λ_2
  - The factors could be negative – implies flipping the paper
- The result is a transformation of the space

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

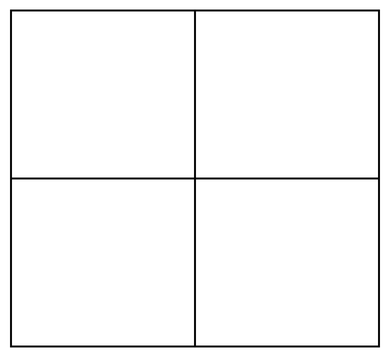
Eigen Analysis

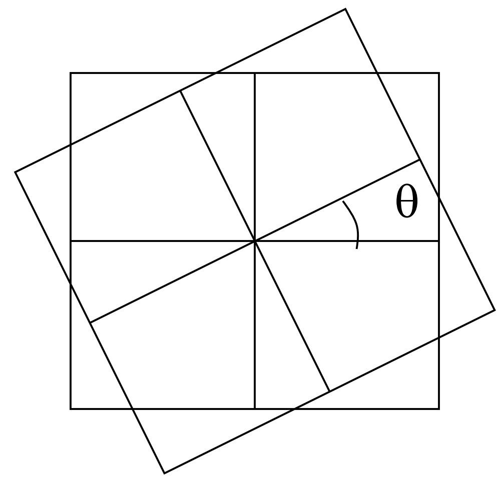
- Not all square matrices have nice eigen values and vectors
  - E.g. consider a rotation matrix

$$\mathbf{R}_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

$$X = \begin{bmatrix} x \\ y \end{bmatrix}$$

$$X_{new} = \begin{bmatrix} x' \\ y' \end{bmatrix}$$



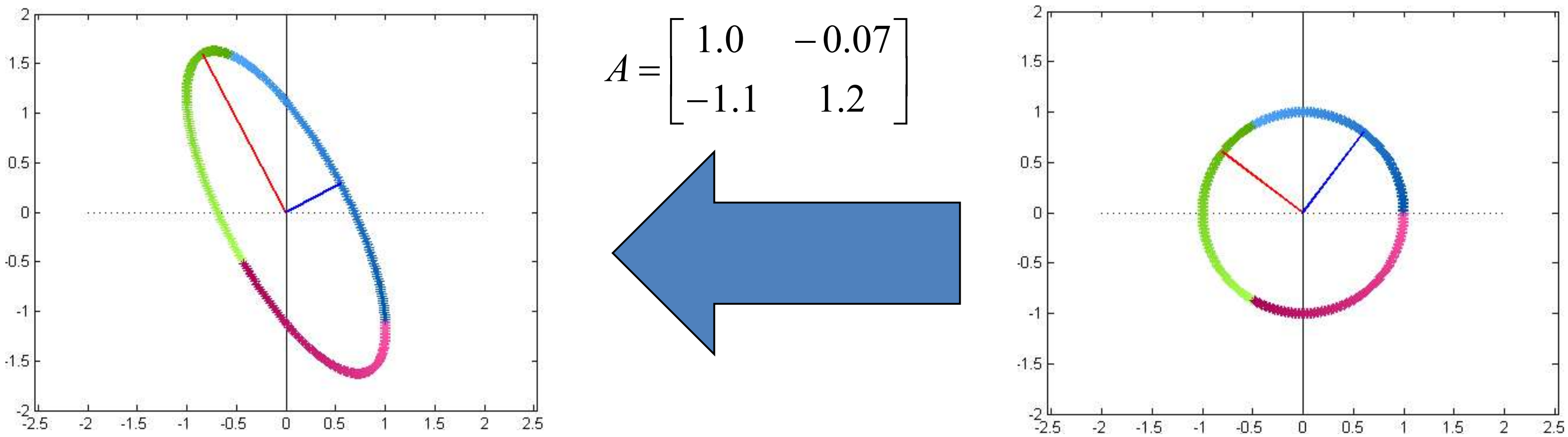


  - This rotates every vector in the plane
    - No vector that remains unchanged

- In these cases the Eigen vectors and values are complex
  - Actually complex conjugate pairs

Poll 1

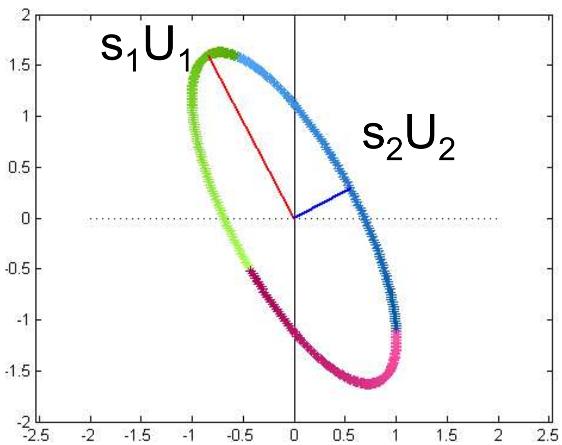
Singular Value Decomposition



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

- Matrix transformations convert circles to ellipses
- 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!

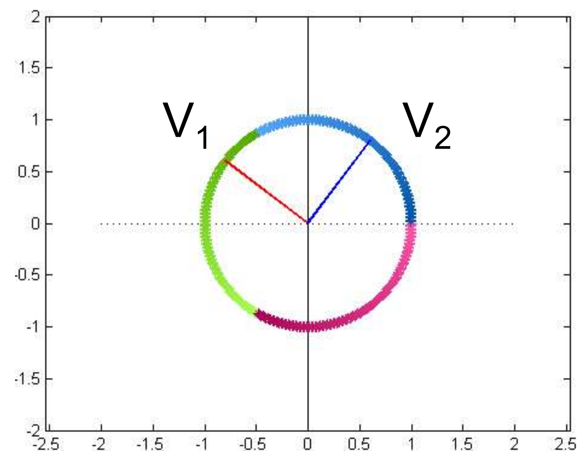
Singular Value Decomposition



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

$\boxed{A = U\,S\,V^T}$

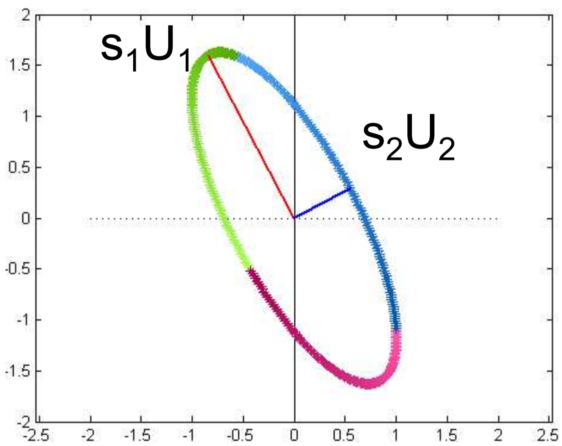
matlab:
[U,S,V] = svd(A)



- U and V are orthonormal matrices
  - Columns are orthonormal vectors
- S is a diagonal matrix

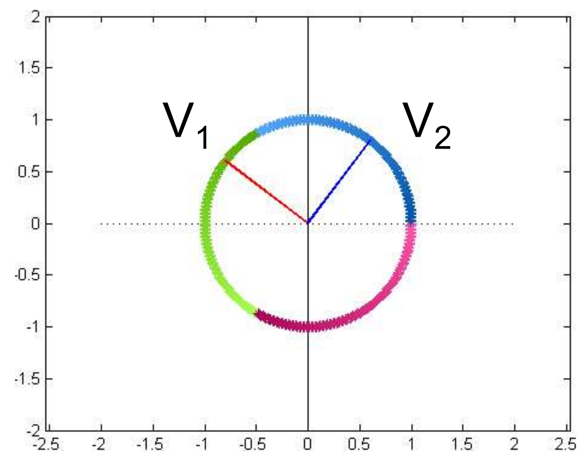
- The *right singular vectors* in V are transformed to the *left singular vectors* in U
  - Scaled by the *singular values* that are the diagonal entries of S

Singular Value Decomposition



$A = U\,S\,V^T$

$A^T = V\,S\,U^T$



- A matrix $\boldsymbol{A}$ converts *right* singular vectors $\boldsymbol{V}$ to *left* singular vectors $\boldsymbol{U}$
- $\boldsymbol{A}^{\mathrm{T}}$ converts $\boldsymbol{U}$ to $\boldsymbol{V}$

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

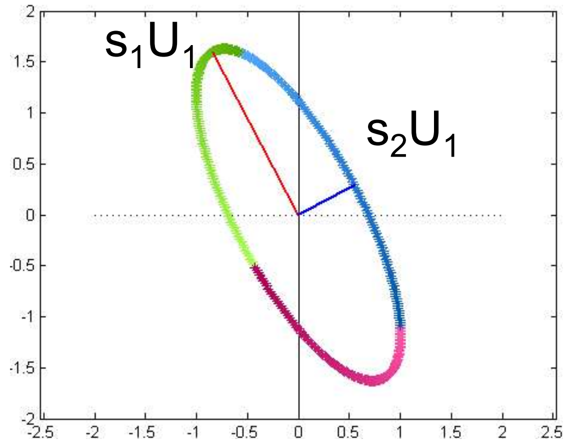
Poll 1

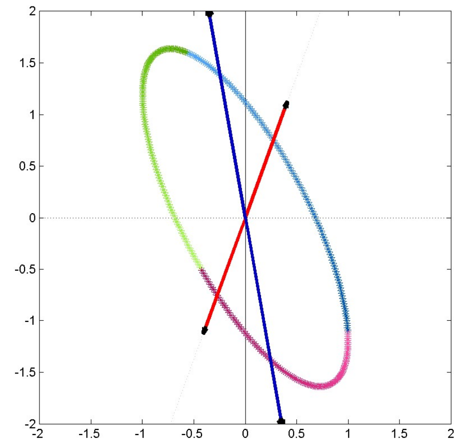
Poll 1

- Q1: Which of the following are true regarding the matrices U, Sigma, and V that result from singular value decomposition? Please note, the "." character represents matrix multiplication and "^T" denotes transposition.
  - **U . U^T = I**
  - **Sigma . Sigma^T = I**
  - **V . V^T = I**
  - **U . V = I**

- Q2: A vector V of length L is transformed by a matrix M to give a new vector V' of length L'. What can we say of the scaling factor L' / L
  - It could be arbitrarily large
  - It could be arbitrarily small
  - **It cannot be larger than the largest singular value of M**
  - **It cannot be smaller than the smallest singular value of M**

The Singular Values

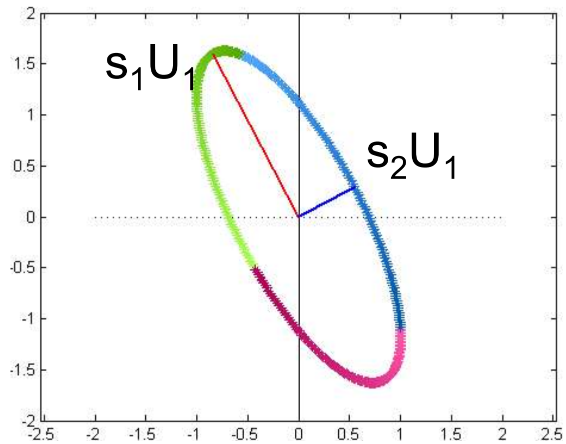


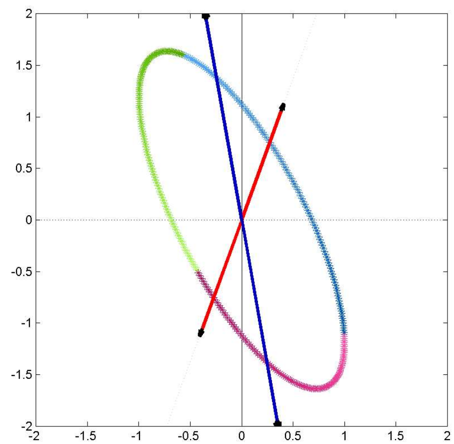


- Square matrices: product of singular values = determinant of the matrix
  - This is also the product of the *eigen* values

- 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

SVD vs. Eigen Analysis

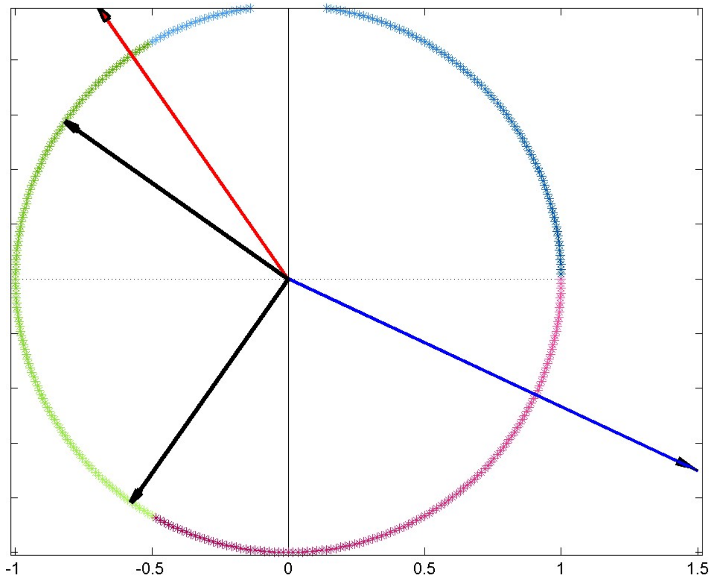




- Eigen analysis of a matrix **A**:
  - Find vectors such that their absolute directions are not changed by the transform
    - Eigen vectors are transformed into scaled versions of themselves!
- SVD of a matrix **A**:
  - Find orthogonal set of vectors such that the *angle* between them is not changed by the transform
    - One set of singular vectors is transformed into another
- For one class of matrices, these two operations are the same

Symmetric Matrices

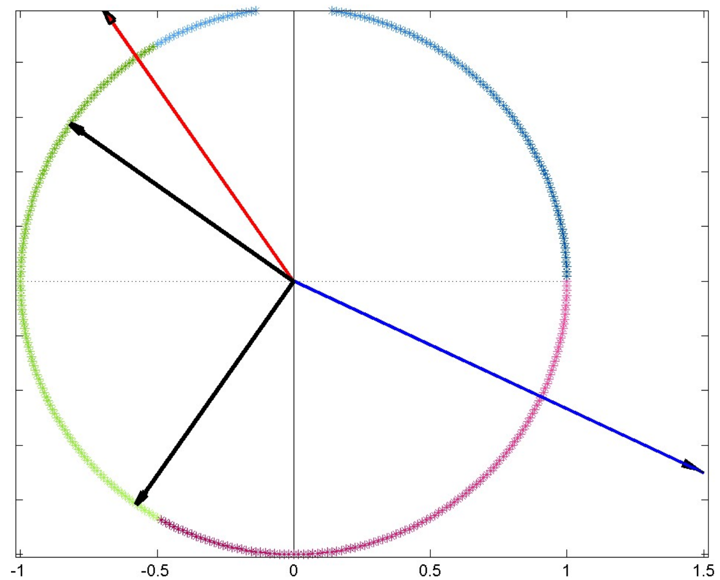
$$\begin{bmatrix} 1.5 & -0.7 \\ -0.7 & 1 \end{bmatrix}$$



- 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 do not rotate the space
  - Only scaling and, if Eigen values are negative, reflection

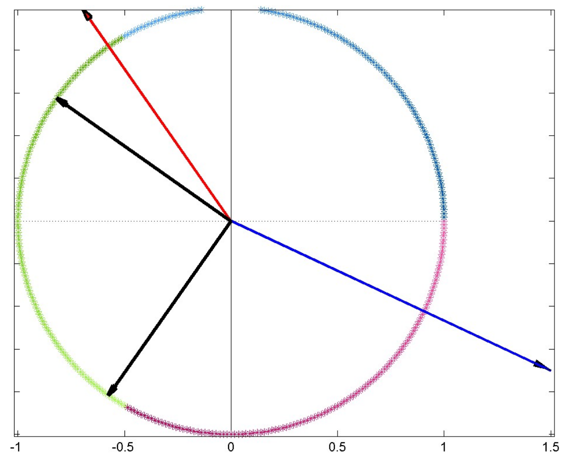
Symmetric Matrices

$$\begin{bmatrix} 1.5 & -0.7 \\ -0.7 & 1 \end{bmatrix}$$

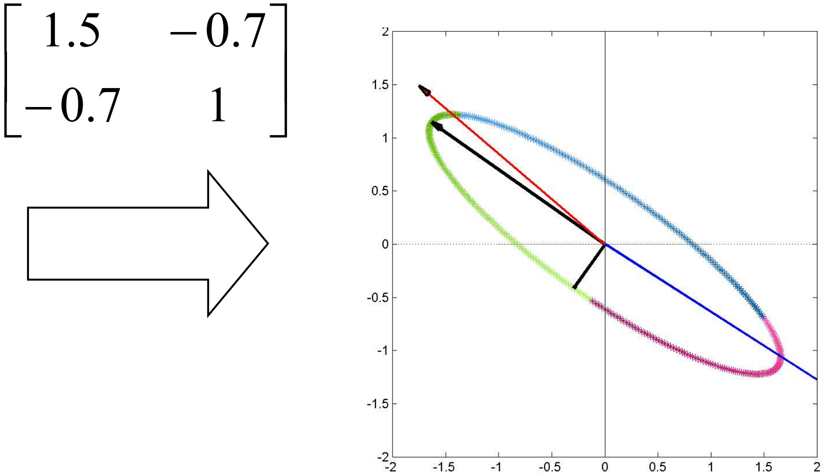


- Matrices that do not change on transposition
  - Row and column vectors are identical
- Symmetric matrix: Eigen vectors and Eigen values are always real
- Eigen vectors are always orthogonal
  - At 90 degrees to one another

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
  - $V_i^T V_i = 1$
  - $V_i^T V_j = 0$, $i \,!\!= j$
- 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 : $M V = V \Lambda$
  - Λ is a diagonal matrix with all eigen values
- $M = V \Lambda V^T$

Definiteness..

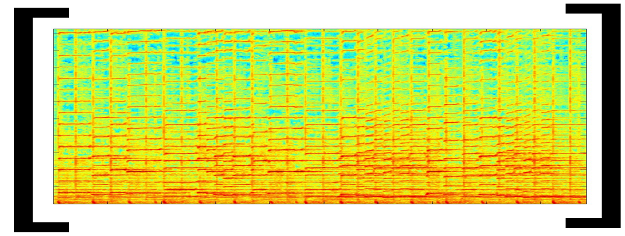
- SVD: Singular values are always positive!
- Eigen Analysis: Eigen values can be real or imaginary
  - Real, positive Eigen values represent stretching of the space along the Eigen vector
  - Real, *negative* Eigen values represent stretching and *reflection* (across origin) of Eigen vector
  - Complex Eigen values occur in conjugate pairs and represent rotation

- 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** along orthogonal axes
    - Transformation has no permutation or 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

SVD on data-container matrices

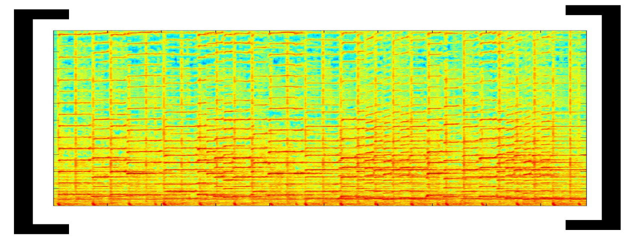


$$\mathbf{X} = [X_1 \ X_2 \ \ \cdots X_N]$$

$$\mathbf{X} = \mathbf{USV}^{\mathrm{T}}$$

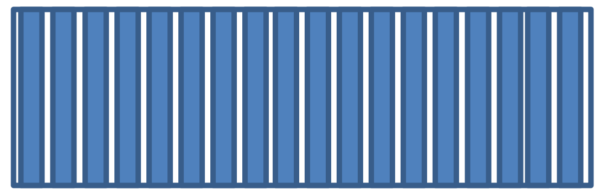
- We can also perform SVD on matrices that are *data containers*
- **S** is a d x N rectangular matrix
  - N vectors of dimension d
- **U** is an orthogonal matrix of d vectors of size d
  - All vectors are length 1
- **V** is an orthogonal matrix of N vectors of size N
- **S** is a d x N diagonal matrix with non-zero entries only on diagonal

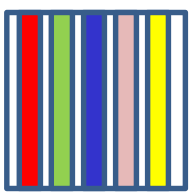
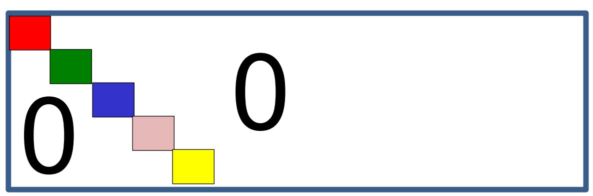
SVD on data-container matrices



$$\mathbf{X} = [X_1 \ X_2 \ \cdots X_N]$$

$$\mathbf{X} = \mathbf{USV}^{\mathrm{T}}$$

$\mathbf{X} =$ 

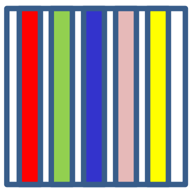
$\mathbf{U} =$  $\mathbf{S} =$ 

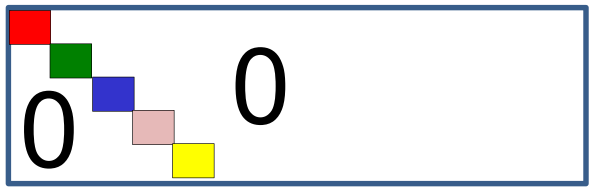
$\mathbf{V}^{\mathrm{T}} =$ 

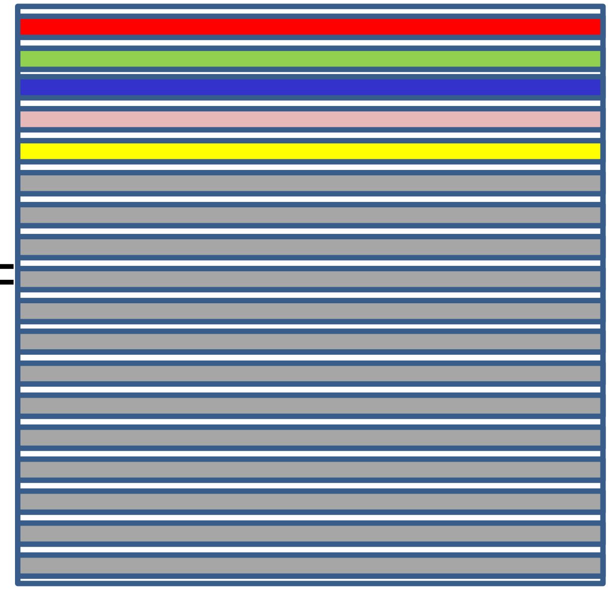
$|U_{\mathrm{i}}| = 1.0$ for every vector in $\mathbf{U}$

$|V_{\mathrm{i}}| = 1.0$ for every vector in $\mathbf{V}$

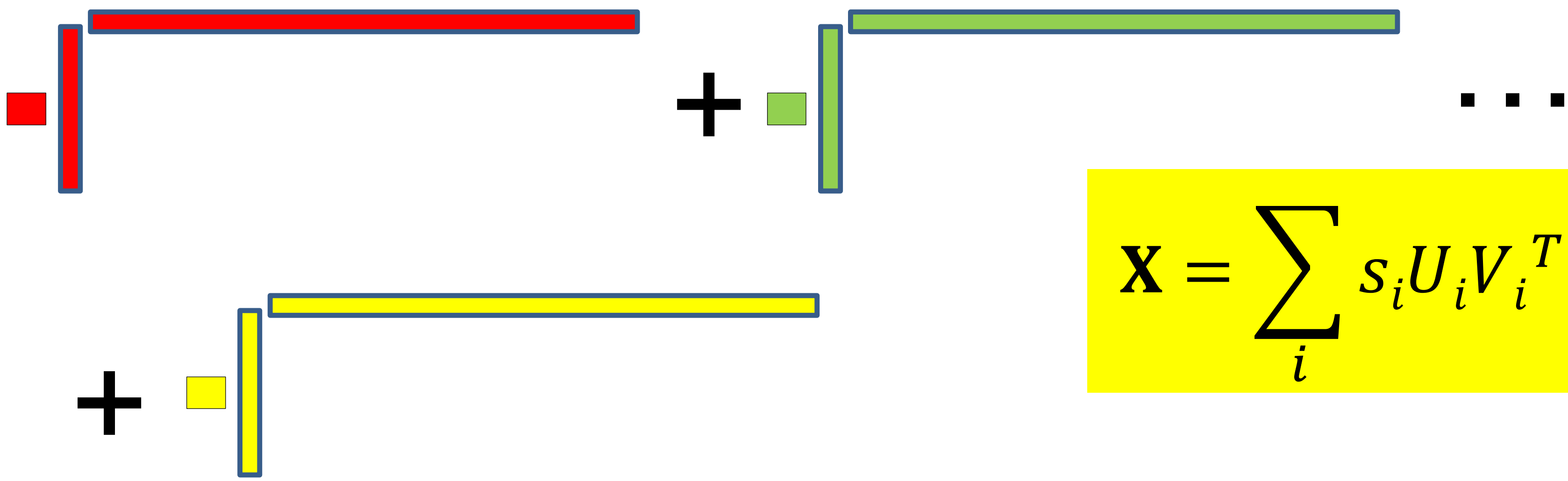
SVD on data-container matrices

$\mathbf{U} =$ 

$\mathbf{S} =$ 

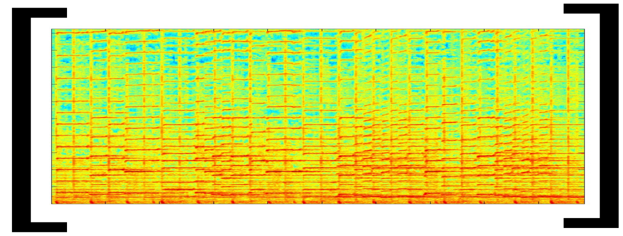
$\mathbf{V}^{\mathrm{T}} =$ 

$\mathbf{X} = \mathbf{USV}^{\mathrm{T}} =$



$$\mathbf{X} = \sum_i s_i U_i V_i^T$$

Expanding the SVD



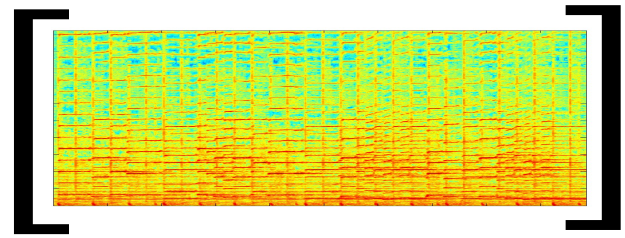
$$\mathbf{X} = [X_1\ X_2\ \ \cdots X_N]$$

$$\mathbf{X} = \mathbf{USV}^{\mathrm{T}}$$

$$\mathbf{X} = s_1 U_1 V_1^T + s_2 U_2 V_2^T + s_3 U_3 V_3^T + s_4 U_4 V_4^T + \ldots$$

- Each left singular vector and the corresponding right singular vector contribute on "basic" component to the data
- The "magnitude" of its contribution is the corresponding singular value

Expanding the SVD



$$\mathbf{X} = [X_1 \ X_2 \ \ \cdots X_N]$$

$$\mathbf{X} = \mathbf{USV}^{\mathrm{T}}$$

$$\mathbf{X} = s_1 U_1 V_1^T + s_2 U_2 V_2^T + s_3 U_3 V_3^T + s_4 U_4 V_4^T + \ldots$$

magnitude

basis

modulation

- Each left singular vector and the corresponding right singular vector contribute on “basic” component to the data

- The “magnitude” of its contribution is the corresponding singular value

Expanding the SVD

$$\mathbf{X} = s_1 U_1 V_1^T + s_2 U_2 V_2^T + s_3 U_3 V_3^T + s_4 U_4 V_4^T + \ldots$$

- Each left singular vector and the corresponding right singular vector contribute on "basic" component to the data

- The "magnitude" of its contribution is the corresponding singular value

- Low singular-value components contribute little, if anything
  - Carry little information
  - Are often just "noise" in the data

Expanding the SVD

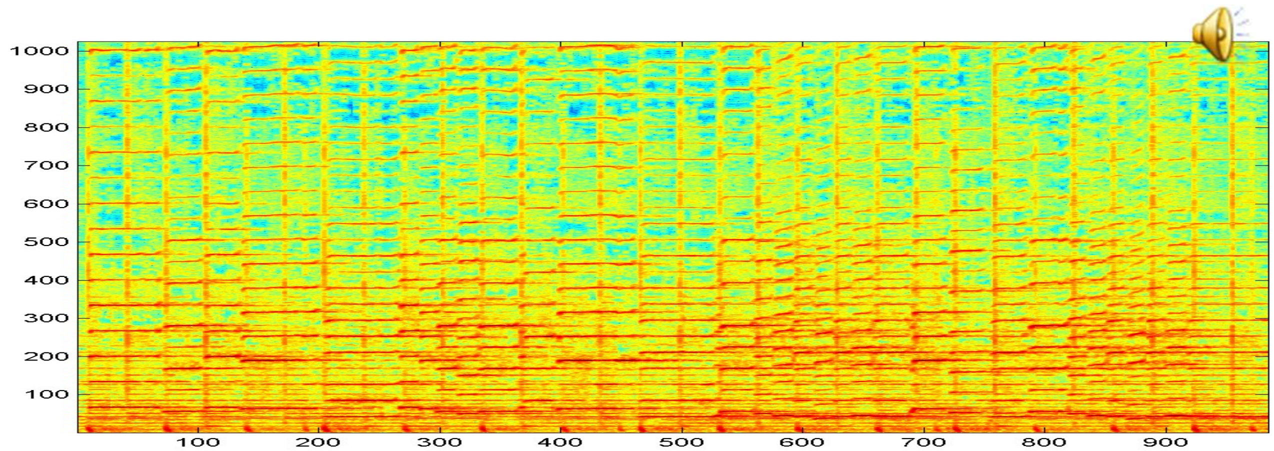
$$\mathbf{X} = s_1 U_1 V_1^T + s_2 U_2 V_2^T + s_3 U_3 V_3^T + s_4 U_4 V_4^T + \ldots$$

$$\mathbf{X} \approx s_1 U_1 V_1^T + s_2 U_2 V_2^T$$

- Low singular-value components contribute little, if anything
  - Carry little information
  - Are often just "noise" in the data

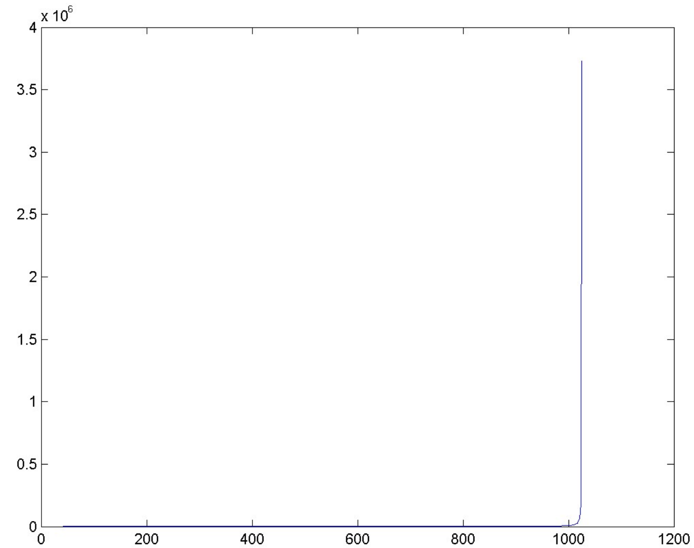
- Data can be recomposed using only the "major" components with minimal change of value
  - Minimum squared error between original data and recomposed data
  - Sometimes eliminating the low-singular-value components will, in fact "clean" the data

An audio example



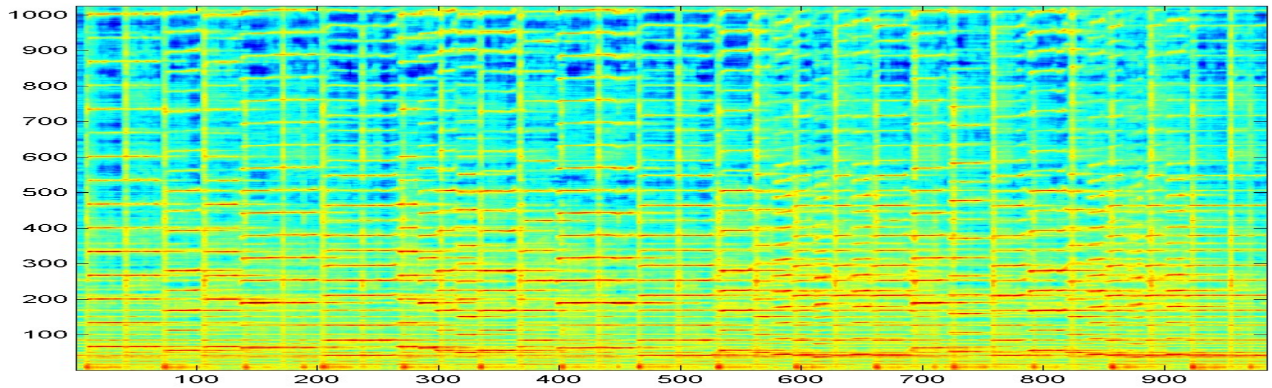
- The spectrogram has 974 vectors of dimension 1025
  - A 1024x974 matrix!
- Decompose: $\mathbf{M} = \mathbf{USV}^T = \Sigma_i \, s_i U_i \, V_i^T$
- **U** is 1024 x 1024
- **V** is 974 x 974
- There are 974 non-zero singular values S_i

Singular Values



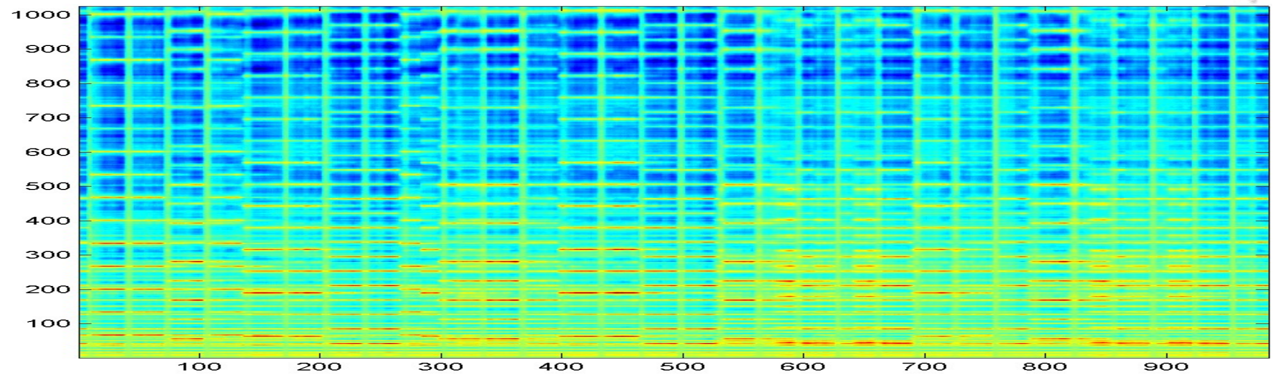
- Singular values for spectrogram **M**
  - Most Singluar values are close to zero
  - The corresponding components are "unimportant"

An audio example



- The same spectrogram constructed from only the 25 highest singular-value components
  - Looks similar
    - With 100 components, it would be indistinguishable from the original
  - Sounds pretty close
  - Background "cleaned up"

With only 5 components



- The same spectrogram constructed from only the 5 highest-valued components
  - Corresponding to the 5 largest singular values
  - Highly recognizable
  - Suggests that there are actually only 5 significant unique note combinations in the music

- Next up: A brief trip through optimization..