Monday Lecture

Dimensionality Reduction:

Principal component analysis, Nonnegative matrix factorization, and Independent component analysis

Roman Akhmetshyn, Jennifer Glover, and Teo Lemay

Dimensionality

What is a “dimension”

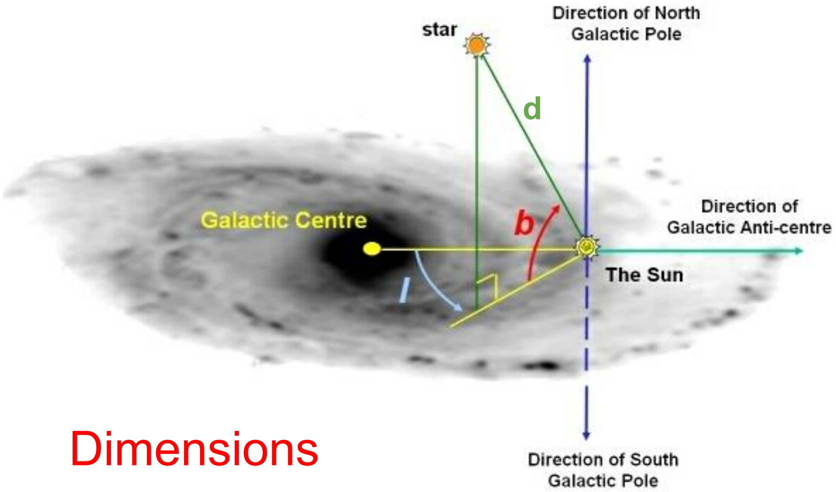
Each of the “things” we can measure about an object is a “**dimension**” or “**feature**” of that object

Let’s imagine we want to know the locations of stars in the galaxy.

Intuitively, we would measure 3 coordinates.

Dimensions

Star	Latitude	Longitude	Distance
1			
2			

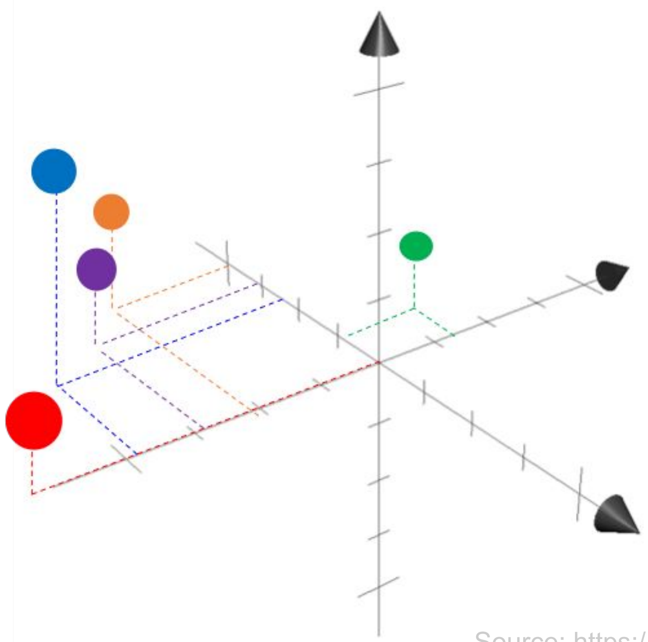


Source: COSMOS

$$\vec{x} = \overbrace{(\ell,\ b,\ d)}^{\text{Dimensions}}$$

Dimension of a Space

We need D pieces of information to describe the location of each object in a D-dimensional space



$$\vec{x} = (x, y, z)$$

Source: https://mbernste.github.io/posts/intrinsic_dimensionality/

Intrinsic Dimensionality

- The **intrinsic** dimensionality of a space is the minimum number of dimensions you need to describe a location in the space
- The dimensionality of your data is the number of dimensions you actually measure

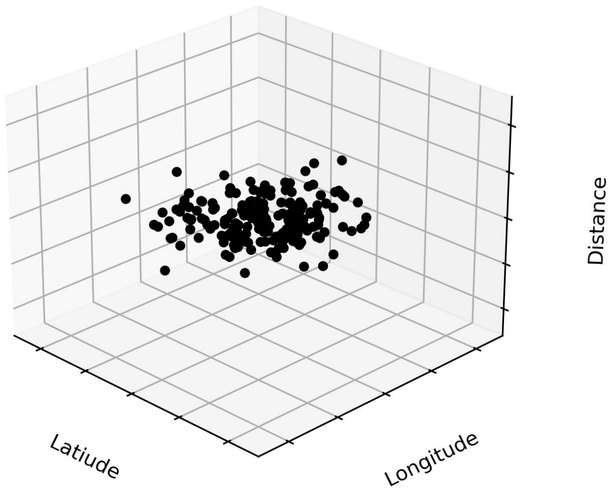
Dimensionality of Data = 6

Star	Latitude	Longitude	Distance	RA	DEC	Parallax
1						
2						

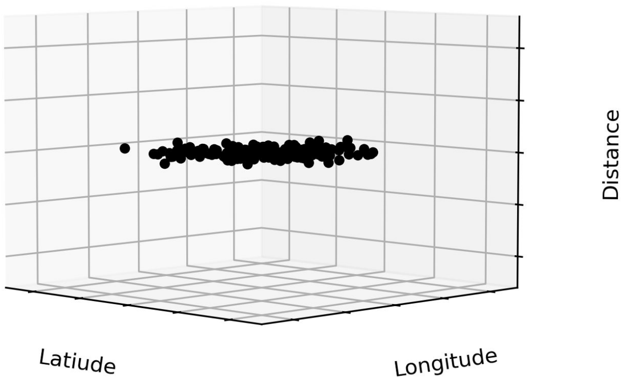
Intrinsic Dimensionality = 3 ?

What if your data looked like this?

View 1



View 2

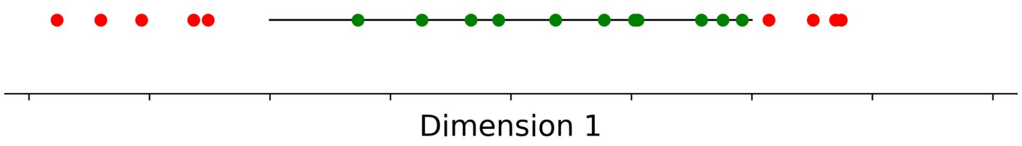


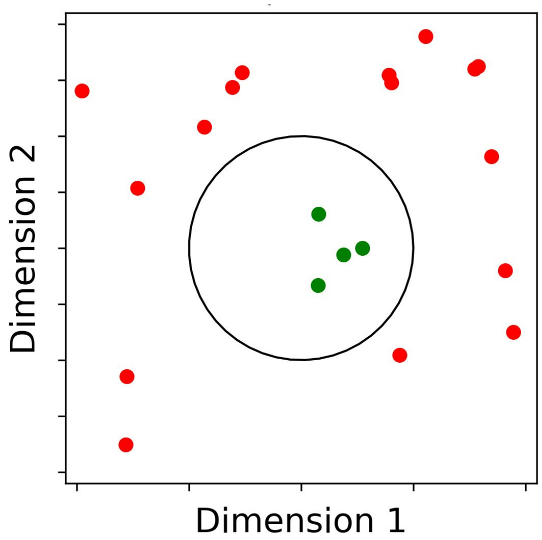
The “Curse of Dimensionality”

The more dimensions a dataset has, the more data is required to constrain a model

The more dimensions you have:

- the more data you need to take
- the more computationally intensive things become
- the more difficult it becomes to visualize the data

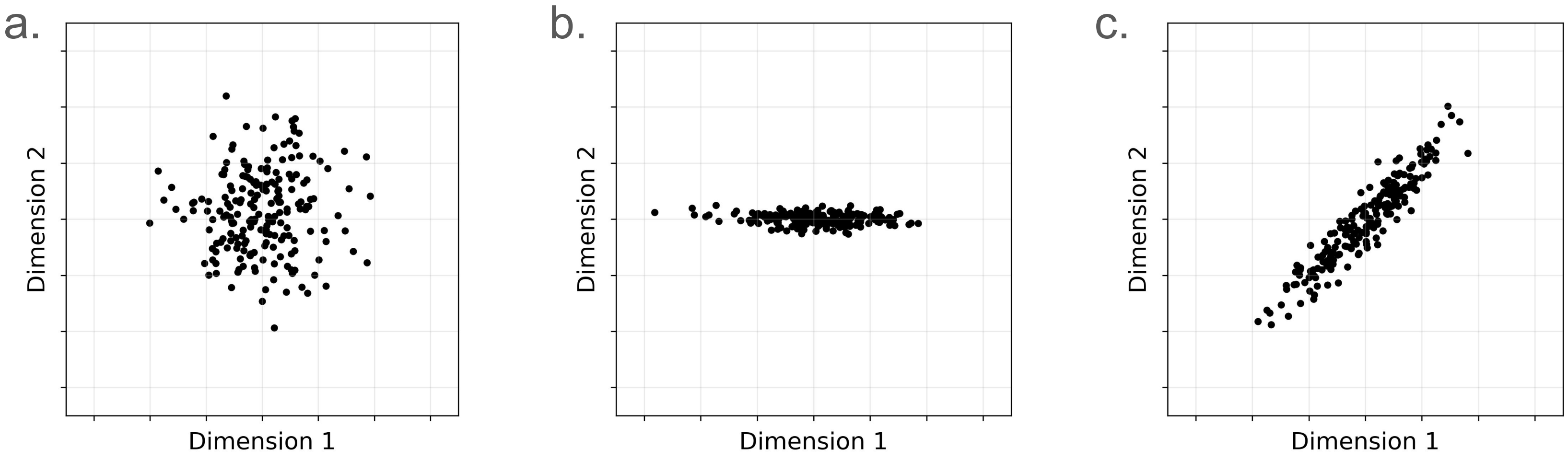




Breaking the “Curse of Dimensionality”

→ Dimension Reduction!

- Ideally, we want to use the minimum required number of dimensions
- There will be some dimensions of the data which capture the most information
    - In real data, many dimensions will be correlated



Dimension Reduction

Dimension Reduction

The goal of **“Dimension Reduction”** is to reduce the complexity of the data down to only the most important features

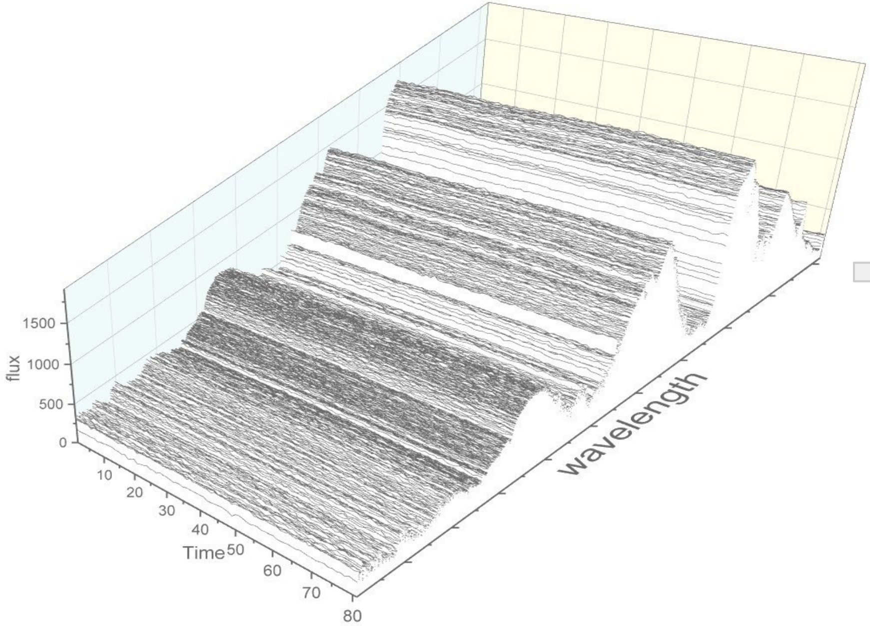
There are several ways to perform dimension reduction

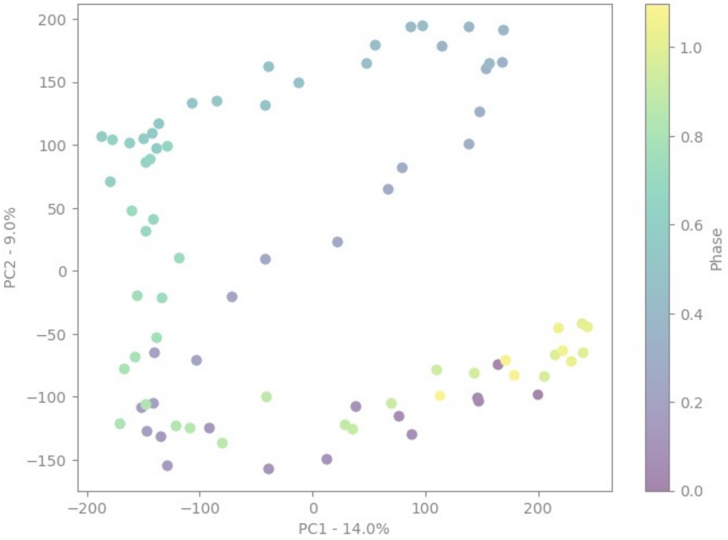
We will go over these three:

- PCA: Principal Component Analysis
- NMF: Nonnegative Component Analysis
- ICA: Independent Component Analysis

Dimension Reduction Method 1:

Principal Component Analysis (PCA)





PCA introduction

PCA – linear transformation of multidimensional data that defines orthogonal set of axes which capture most variance.

Generally used to: decorrelate variables, compress data, extract features, and reduce noise.

Important:

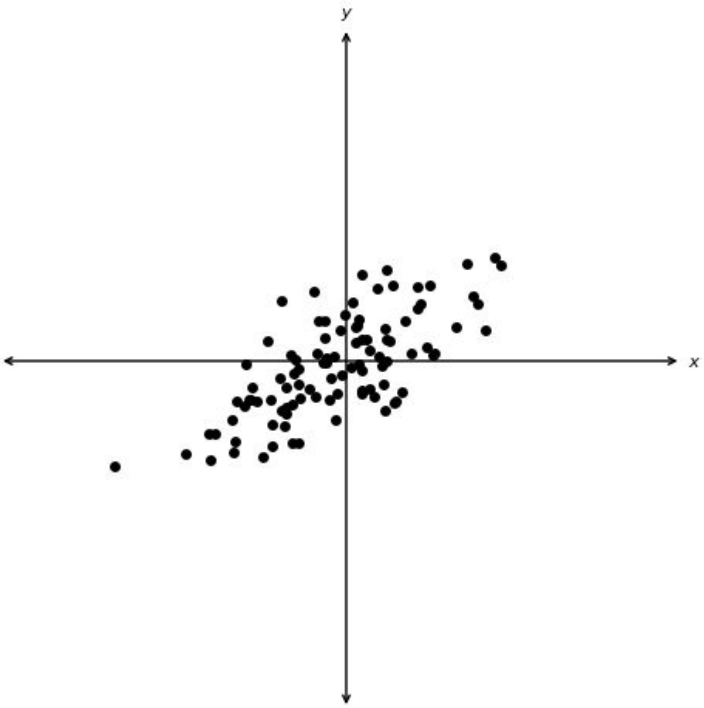
- pre-PCA data must be centered,
- number of PCs = min (measurements, features)
- PCs are unphysical
- Data is usually projected on PCs

How does PCA work?

Before we start, let's center our data by removing the mean.

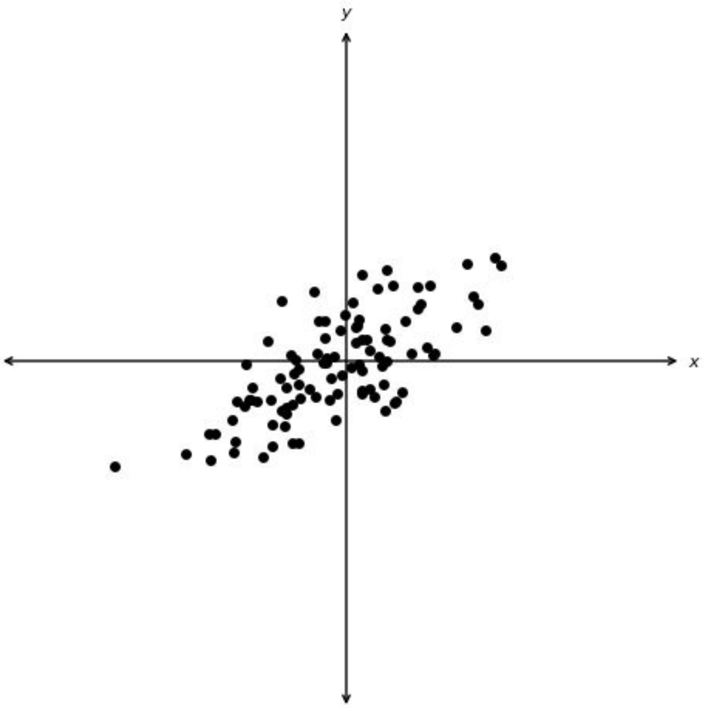
If your dimensions have different units, we should also scale them:

X = (data - mean) / std



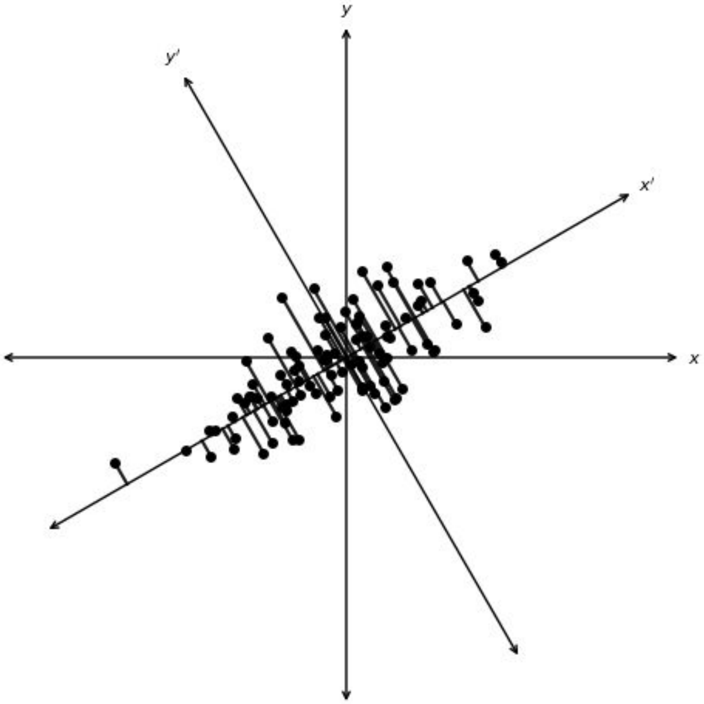
Information we can infer:

- Correlation of X and Y
- Variance of data
- Clustering
- Patterns



We rotate axes to align with data's correlation, i.e. minimize orthogonal distances of data points to the axes, i.e. maximize variance along the axes.

- Eigenvectors: unit-vectors, our "new" axes, principal components
- Eigenvalues: variance along principal components



What’s next?

PCA is a linear transformation and is based on data projection (orthogonal)

We can:

- Project data onto principal components
- Visualize and explore data with principal component planes
- Reconstruct whole data using few top PCs
- Reconstruct single data using PCP coordinates
- Reconstruct data into less dimensions

How can we use that in astronomy?

We can:

- Remove low amplitude noise from data (reconstruct by top PCs)
- Remove high amplitude noise (reconstruct by bottom PCs)
- Get a basis to describe objects, standardize parameters
- Optimize models and simulations (reconstruct to lower dimension)
- Prepare data for classification, clustering and ML algorithms
- Interpolate across missing data
- And probably much more!

PCA pros/cons

Pros

- Simple
- Computationally efficient
- PCs can be informative
- Handles multicollinearity between variables
- Minimizes reconstruction error

Cons

- Not scaling or unit invariant
- Need to decide about best number of PCs
- PCs are not always informative
- Fails to capture correlated and non-linear patterns
- Sensitive to outliers
- Not optimal for categorical data

Monday Exercises:

1. Investigate the “Curse of Dimensionality”

2. Write your own PCA

PCA Exercises Key steps:

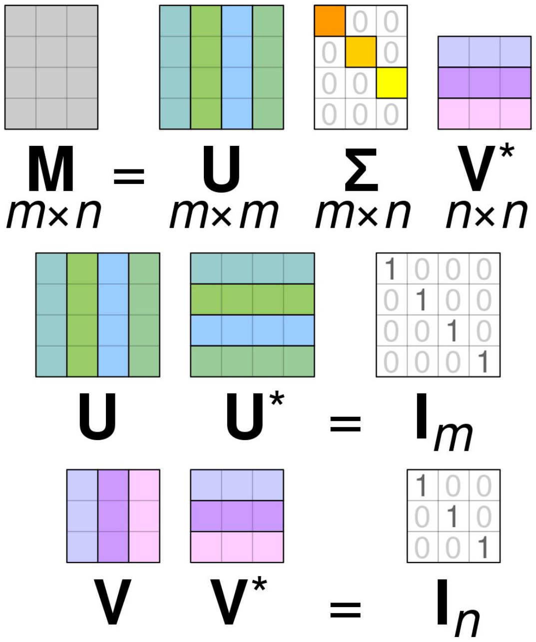
We will use SDSS rest frame spectra of 4000 galaxies that span across 1000 wavelength. Each wavelength is a dimension.

1. We will center data by subtracting mean spectra
2. Use Singular Value Decomposition to retrieve principal components.

SVD

A method that decomposes a matrix into a product of matrices. A Basis of PSA.

$$U\Sigma V^T = \frac{1}{\sqrt{N-1}}X$$



Source: Wikipedia

SVD $$U\Sigma V^T = \frac{1}{\sqrt{N-1}}X$$

U - hierarchically arranged basis-columns that describe variance in X-columns

Σ - hierarchically arranged non-negative diagonal singular values

V - hierarchically arranged variance-columns of each U-column across data / mixture-columns of U to create X-columns

U and V are unitary ($UU^T=U^TU=I_{nxn}$, $VV^T=V^TV=I_{mxm}$), columns are orthonormal.

SVD

$$U\Sigma V^T = \frac{1}{\sqrt{N-1}} X$$

$$X = \underbrace{\begin{bmatrix} | & | & & | \\ x_1, & x_2, & \cdots & x_m \\ | & | & & | \end{bmatrix}}_{m} \Big\} n = U \Sigma V^T = \overset{n \times n}{\begin{bmatrix} | & | & & | \\ u_1, & u_2, & \cdots & u_n \\ | & | & & | \end{bmatrix}} \overset{n \times m}{\begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_m \\ & & 0 & \end{bmatrix}} \overset{m \times m}{\begin{bmatrix} - & v_1^T & - \\ - & v_2^T & - \\ & \vdots & \\ & v_m^T & \end{bmatrix}}$$

$$= \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T \cdots \sigma_m u_m v_m^T$$

SVD

$$U\Sigma V^T = \frac{1}{\sqrt{N-1}}X$$

$$X = U\Sigma V^T$$

$$X^TX = V\Sigma^T \underbrace{U^T U}_{1}\Sigma V^T = V\Sigma^T\Sigma V^T = V\Sigma^2 V^T \quad (\times V)$$

$$\Rightarrow (X^TX)V = V\Sigma^2$$

$$\begin{bmatrix} X^TX \end{bmatrix} \begin{bmatrix} | & | & & | \\ v_1 & v_2 & \cdots & v_m \\ | & | & & | \end{bmatrix} = \begin{bmatrix} | & | & & | \\ v_1 & v_2 & \cdots & v_m \\ | & | & & | \end{bmatrix} \begin{bmatrix} \sigma^2 & & & \\ & \sigma^2 & & \\ & & \ddots & \\ & & & \sigma_D^2 \end{bmatrix}$$

SVD

$$U\Sigma V^T = \frac{1}{\sqrt{N-1}}X$$

$$X = U\Sigma V^T$$

$$X^TX = V\Sigma^T \underbrace{U^T U}_{1} \Sigma V^T = V\Sigma^T\Sigma V^T = V\Sigma^2 V^T \qquad (\times V)$$

$$\Rightarrow (X^TX)V = V\Sigma^2$$

$$\begin{bmatrix} X^TX \end{bmatrix} \overset{\text{Ei\_vecs}}{\begin{bmatrix} | & | & & | \\ v_1 & v_2 & \cdots & v_m \\ | & | & & | \end{bmatrix}} = \begin{bmatrix} | & | & & | \\ v_1 & v_2 & \cdots & v_m \\ | & | & & | \end{bmatrix} \overset{\text{Ei\_vals}}{\begin{bmatrix} \sigma^2 & & & \\ & \sigma^2 & & \\ & & \ddots & \\ & & & \sigma_D^2 \end{bmatrix}}$$

SVD

$$U\Sigma V^T = \frac{1}{\sqrt{N-1}} X$$

$$X = U\Sigma V^T$$

$$X^T X = V\Sigma^T \underbrace{U^T U}_{1} \Sigma V^T = V\Sigma^T \Sigma V^T = V\Sigma^2 V^T \qquad (\times V)$$

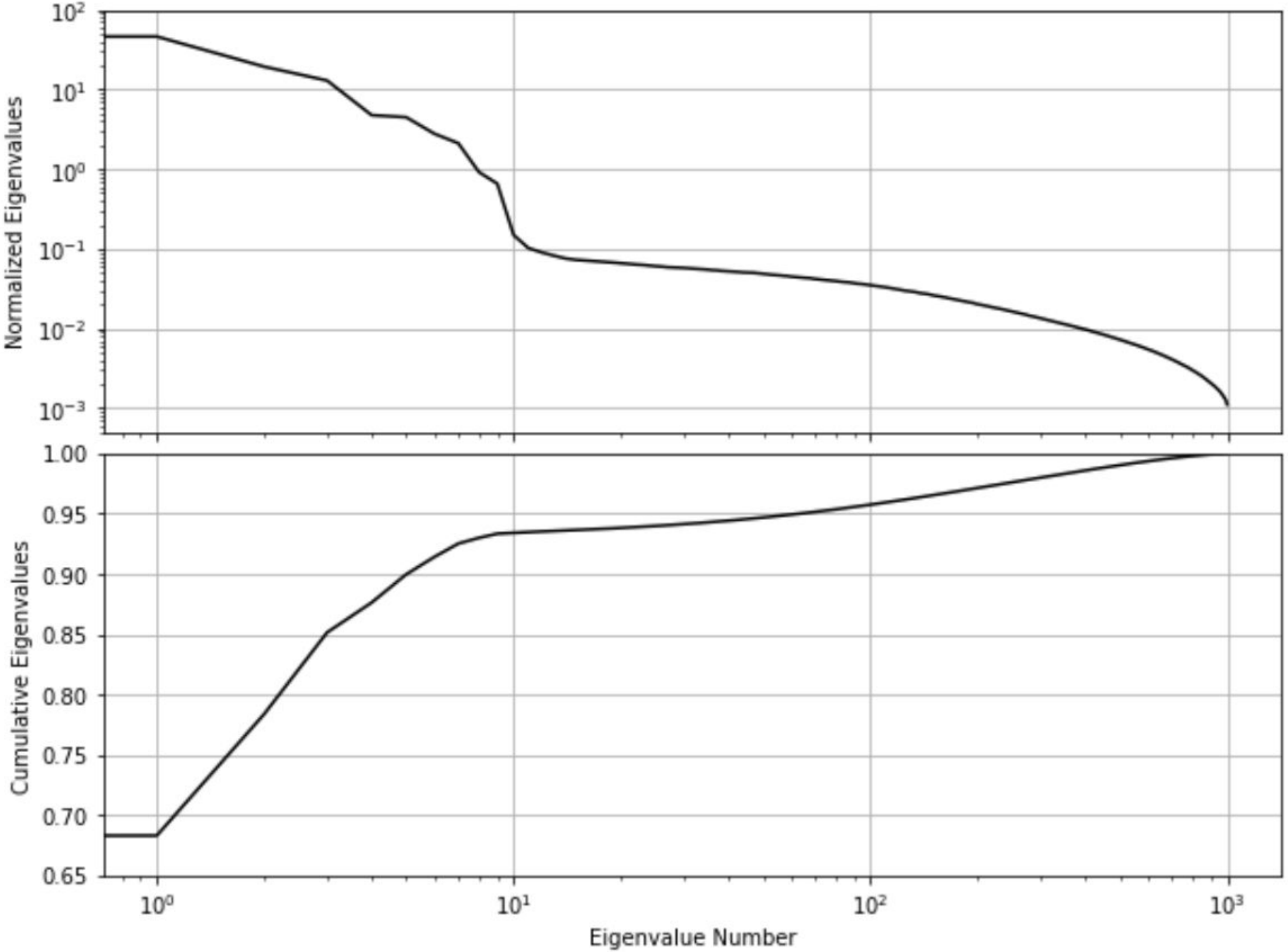
$$\Rightarrow (X^T X) V = V\Sigma^2$$

$$\overset{C}{\begin{bmatrix} X^T X \end{bmatrix}} \overset{\text{Ei\_vecs}}{\begin{bmatrix} | & | & & | \\ v_1 & v_2 & \cdots & v_m \\ | & | & & | \end{bmatrix}} = \begin{bmatrix} | & | & & | \\ v_1 & v_2 & \cdots & v_m \\ | & | & & | \end{bmatrix} \overset{\text{Ei\_vals}}{\begin{bmatrix} \sigma^2 & & & \\ & \sigma^2 & & \\ & & \ddots & \\ & & & \sigma_D^2 \end{bmatrix}}$$

PCA Exercises Key steps:

We will use SDSS rest frame spectra of 4000 galaxies that span across 1000 wavelength. Each wavelength is a dimension.

1. We will center data by subtracting mean spectra
2. Use Single Value Decomposition to retrieve principal components.
3. Scree plot



Normalized Eigenvalues
10^2
10^1
10^0
10^{-1}
10^{-2}
10^{-3}
Cumulative Eigenvalues
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
10^0
10^1
10^2
10^3
Eigenvalue Number

PCA Exercises Key steps:

We will use SDSS rest frame spectra of 4000 galaxies that span across 1000 wavelength. Each wavelength is a dimension.

1. We will center data by subtracting mean spectra
2. Use Single Value Decomposition to retrieve principal components.
3. Scree plot
4. Project data onto PCP
5. Reconstruct data

Wednesday Lecture

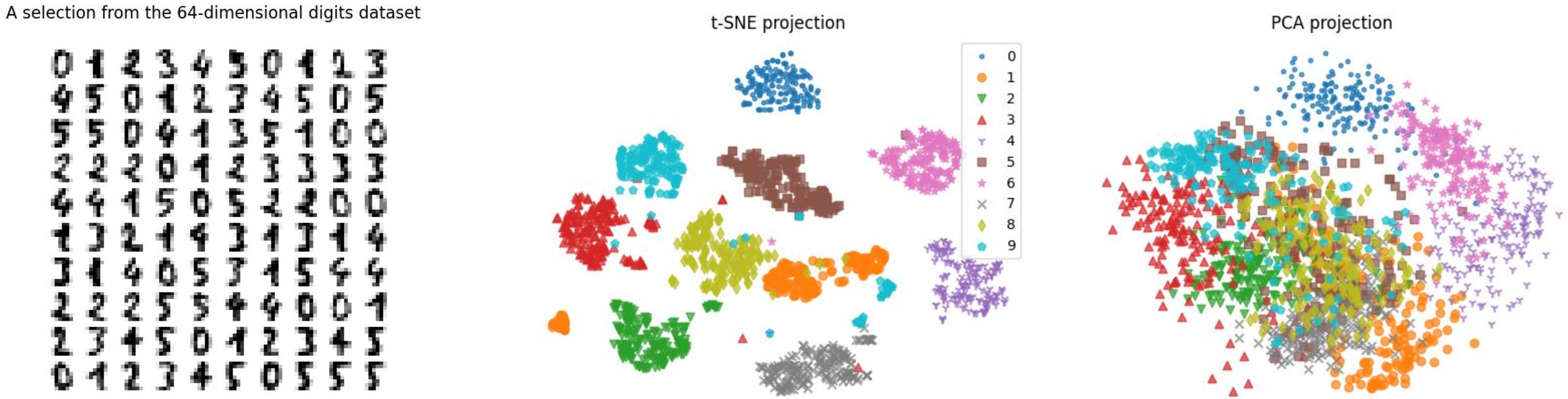
Dimensionality reduction:

Principal component analysis, Nonnegative matrix factorization, and Independent component analysis

Roman Akhmetshyn, Jennifer Glover, and Teo Lemay

PCA is not the only option

- Nonnegative Matrix Factorization (NMF)
- Independent component analysis (ICA)
- Manifold learning
- And more! (Kernel methods, Autoencoder, t-SNE, etc.)



“Important Component” methods

- **Nonnegative Matrix Factorization (NMF)**
- **Independent component analysis (ICA)**
- Manifold learning
- And more! (Kernel methods, Autoencoder, t-SNE, etc.)

$$\text{Data} \; = \mathbf{X} \approx \mathbf{WH}$$

$$\mathbf{W} \rightarrow \; \text{Components}$$

$$\mathbf{H} \rightsquigarrow \; \text{Weights}$$

Dimension Reduction Method 2:

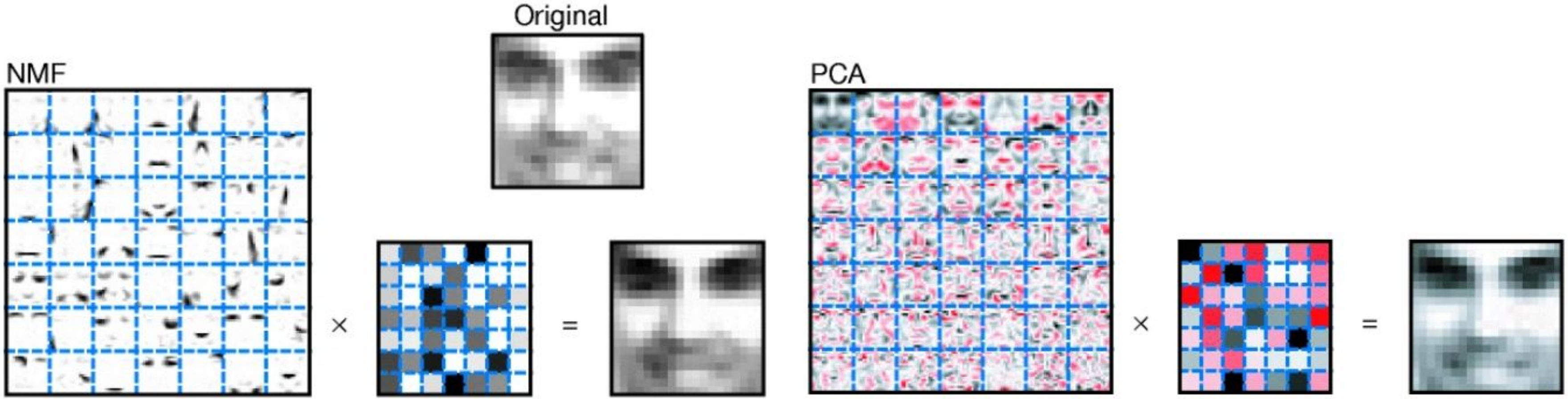
Nonnegative Matrix Factorization (NMF)

Nonnegative Matrix Factorization (NMF)

- Very similar to PCA
  - Dimensionality reduction
  - Sometimes informative components
- Constrain solution with known characteristics of the data
- A lot of astronomical data is non-negative …

Nonnegative Matrix Factorization (NMF)

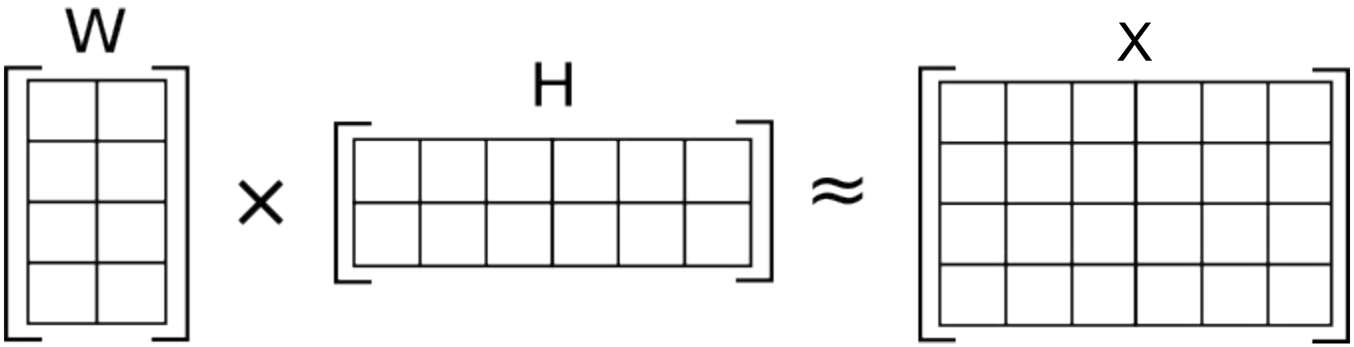
- NMF constraints mean that different components get selected
- PCA tends towards holistic features, NMF can pick out important parts



D. D. Lee and H. S. Seung, *Learning the Parts of Objects by Non-Negative Matrix Factorization*, Nature **401**, 788 (1999).

How does NMF work?

- Choose number of intrinsic dimensions < actual dimensionality
- Guess initial values, calculate distance metric



$$\chi^2 = \|(\boldsymbol{X} - \boldsymbol{W}\boldsymbol{H})\|^2$$
$$= \sum_{ij} \left(X_{ij} - \sum_{k} W_{ik} H_{kj} \right)^2$$

How does NMF work?

- Choose number of intrinsic dimensions < actual dimensionality
- Guess initial values, calculate distance metric
- **Iterative update rule**

$$\chi^2 = \|(\boldsymbol{X} - \boldsymbol{W}\boldsymbol{H})\|^2$$
$$= \sum_{ij} \left(X_{ij} - \sum_{k} W_{ik} H_{kj} \right)^2$$

$$\boldsymbol{H} \leftarrow \boldsymbol{H} \circ \frac{\boldsymbol{W}^T \boldsymbol{X}}{\boldsymbol{W}^T \boldsymbol{W} \boldsymbol{H}}$$

$$\boldsymbol{W} \leftarrow \boldsymbol{W} \circ \frac{\boldsymbol{X} \boldsymbol{H}^T}{\boldsymbol{W} \boldsymbol{H} \boldsymbol{H}^T}$$

How does NMF work?

- Choose number of intrinsic dimensions < actual dimensionality
- Guess initial values, calculate distance metric
- Iterative update rule
- **NMF is sensitive to local minima**
- **Multiple random initializations**

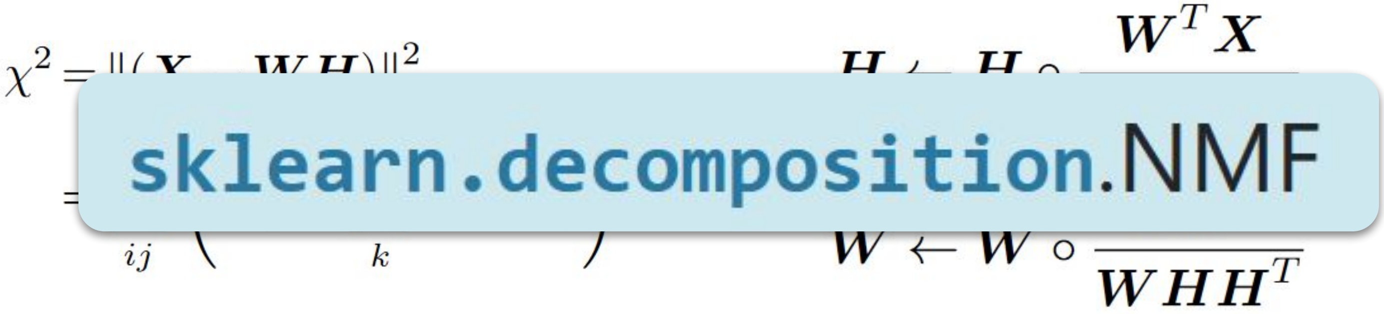
$$\chi^2 = \|(\boldsymbol{X} - \boldsymbol{W}\boldsymbol{H})\|^2$$
$$= \sum_{ij}\left(X_{ij} - \sum_k W_{ik}H_{kj}\right)^2$$

$$\boldsymbol{H} \leftarrow \boldsymbol{H} \circ \frac{\boldsymbol{W}^T\boldsymbol{X}}{\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{H}}$$

$$\boldsymbol{W} \leftarrow \boldsymbol{W} \circ \frac{\boldsymbol{X}\boldsymbol{H}^T}{\boldsymbol{W}\boldsymbol{H}\boldsymbol{H}^T}$$

How does NMF work?

- Choose number of intrinsic dimensions < actual dimensionality
- Guess initial values, calculate distance metric
- Iterative update rule
- **NMF is sensitive to local minima**
- **Multiple random initializations**

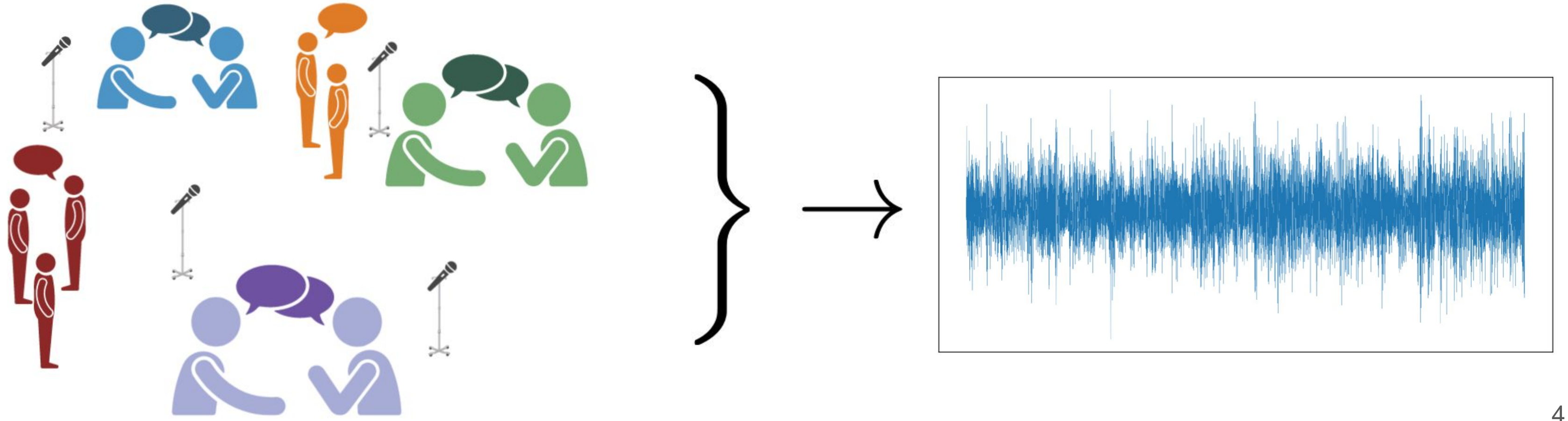


Dimension Reduction Method 3:

Independent Component Analysis (ICA)

Independent Component Analysis (ICA)

- Another matrix decomposition approach
  - Emphasis on important components
  - Exploration of patterns in data
- Assume data is a combination of signals from independent sources



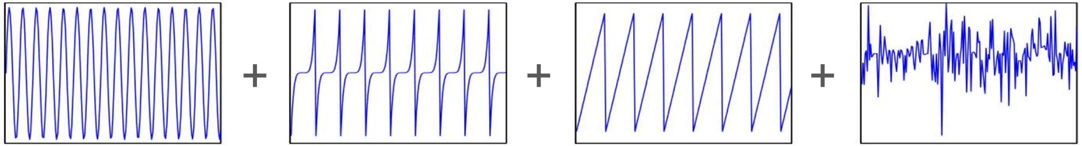
Independent Component Analysis (ICA)

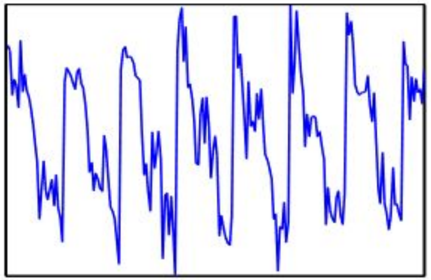
ICA

- Find statistically independent “source signals”
- Physically interpretable Independent Components?

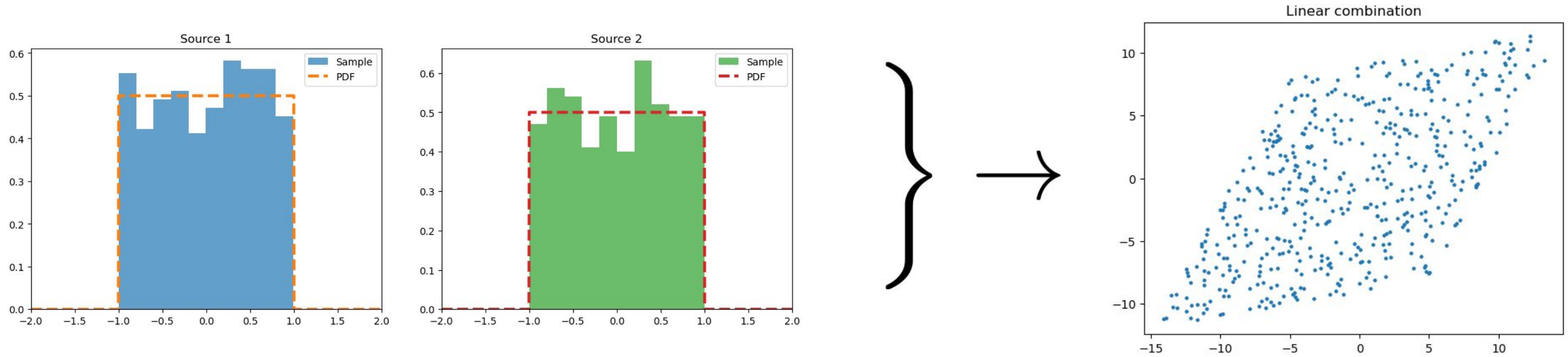
PCA

- Maximize variance along principle components
- Eigenvectors of covariance matrix for the data

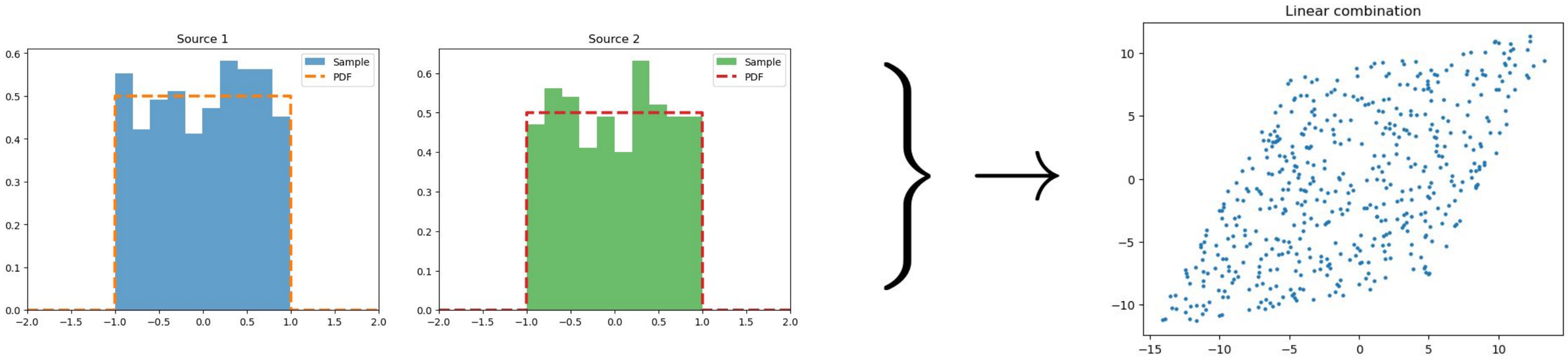


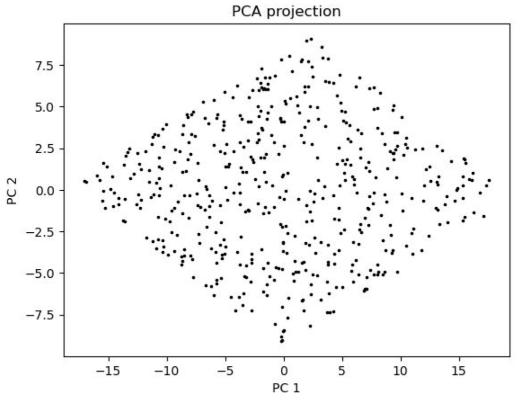


Independent Component Analysis (ICA)

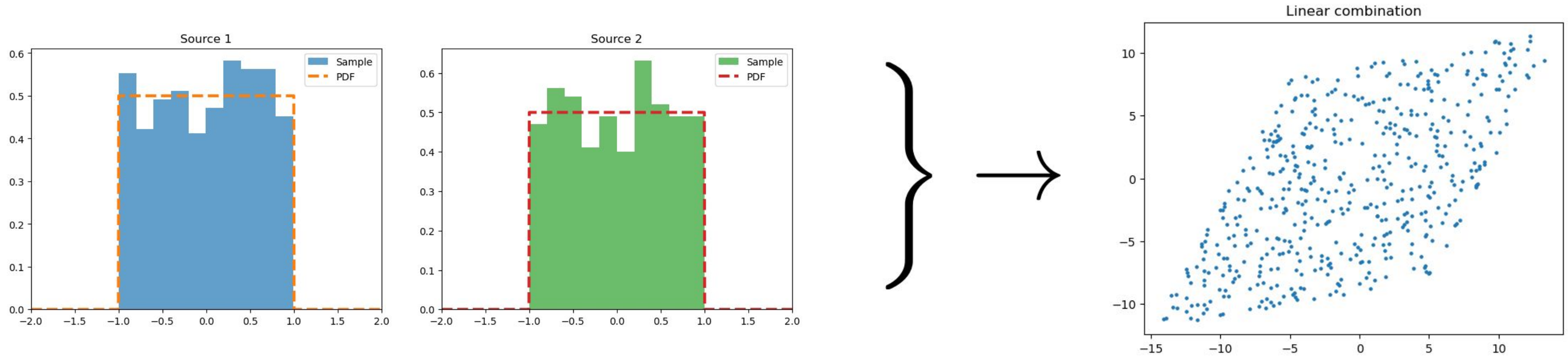


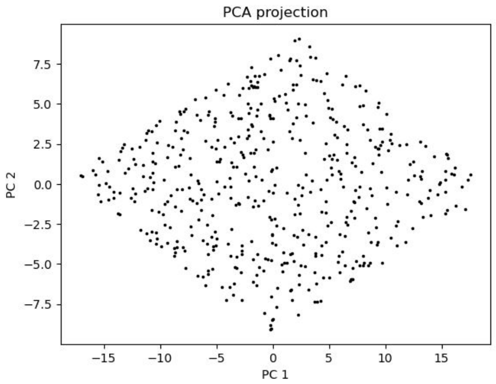
Independent Component Analysis (ICA)



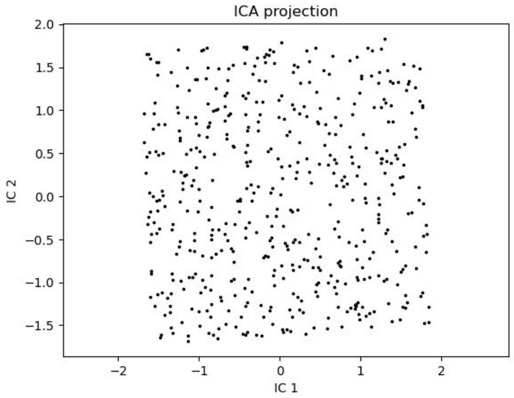


Independent Component Analysis (ICA)





vs.



Independent Component Analysis (ICA)

Assumptions

- Sources in the data are independent
- Sources are non-gaussian

$$p(y_a, y_b, y_c, ...) = p(y_a)p(y_b)p(y_c)\ ...$$

Notes

- No measure of variance explained by each component
- No hierarchy of independent components
- Should center data (mean subtraction)

How does ICA work?

- Central limit theorem!
- Combinations of random variables tends towards a gaussian
- Assume data should be more gaussian than the sources
- Choose the number Independent components to look for
- Try to find the maximally **non**-gaussian matrix decomposition

$$\mathbf{X} = \mathbf{W}\mathbf{H}$$

$$PCA \quad \mathbf{W}^{-1}\mathbf{X} \quad \text{maximizes variance}$$

$$ICA \quad \mathbf{W}^{-1}\mathbf{X} \quad \text{maximizes non-gaussianness}$$

Independent Component Analysis (ICA)

- A few different measures of non-gaussian-ness
    - Kurtosis, higher power cumulants
    - Neg-entropy
    - Mutual information minimization
- Practical implementations (fastICA) combine parts of different measures and make useful approximations
- Requires setting the number of ICs to look for

Independent Component Analysis (ICA)

- A few different measures of non-gaussian-ness
  - Kurtosis, higher power cumulants
  - Neg-entropy
  - Mutual information minimization
- Practical implementations (fastICA) combine parts of different measures and make useful approximations
- Requires setting the number of ICs to look for

`sklearn.decomposition.FastICA`

PCA is not the only option!

- PCA is the simplest, and most widely applicable of these methods

- What do you know about your data a priori?
  - Noisy combination of signals?
  - Nonlinear high dimensional shape?

- What are you trying to achieve with dimensionality reduction?
  - Do the components matter?
  - Should you believe the components?

Wednesday Exercise: Play around with parameters

- NMF example with Olivetti faces dataset

- ICA mixtures of waveforms

- Find sources in a mystery dataset