3011979 Intro to Deep Learning for Medical Imaging

L3: Unsupervised learning – PCA and MDS

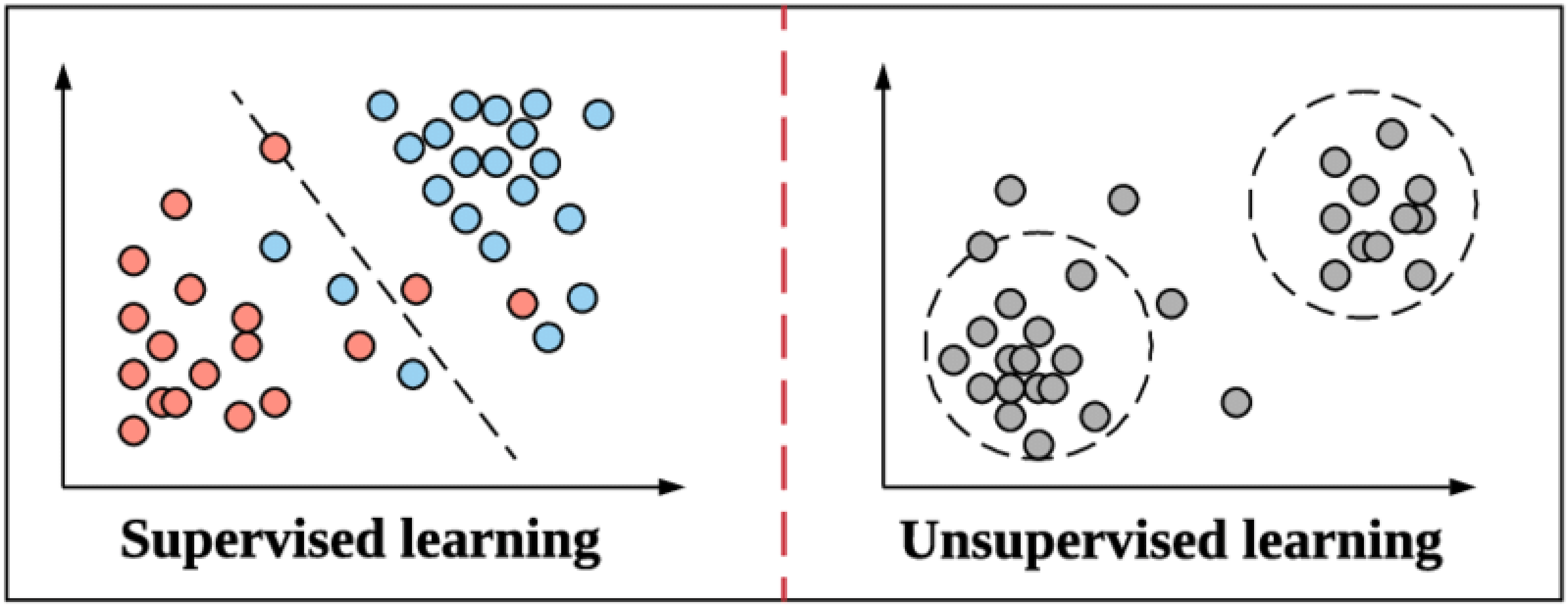
Feb 5th, 2021

Sira Sriswasdi, Ph.D.
Research Affairs, Faculty of Medicine
Chulalongkorn University

Today's objectives

- Introduction to unsupervised learning
- Principal component analysis
- Multidimensionality scaling

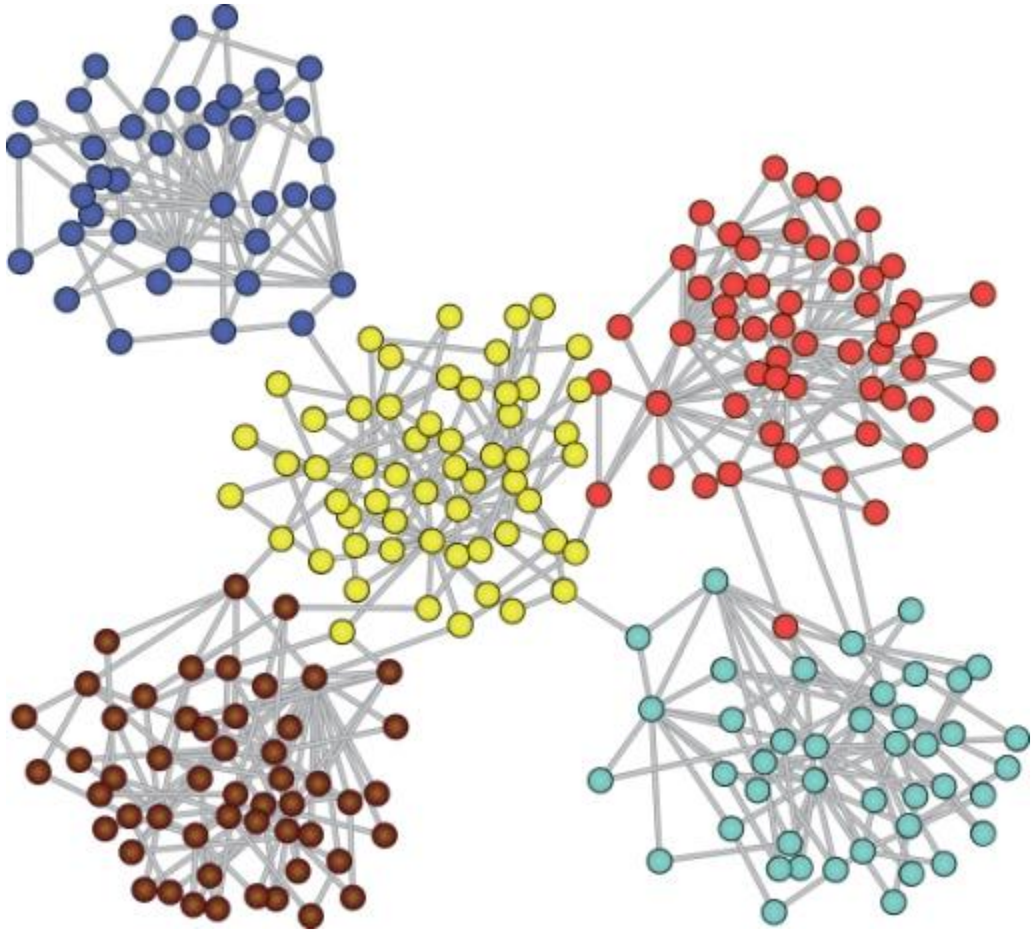
Unsupervised learning



Qian, B. et al. “Orchestrating the Development Lifecycle of Machine Learning-Based IoT Applications: A Taxonomy and Survey”

- Pattern recognition through data density

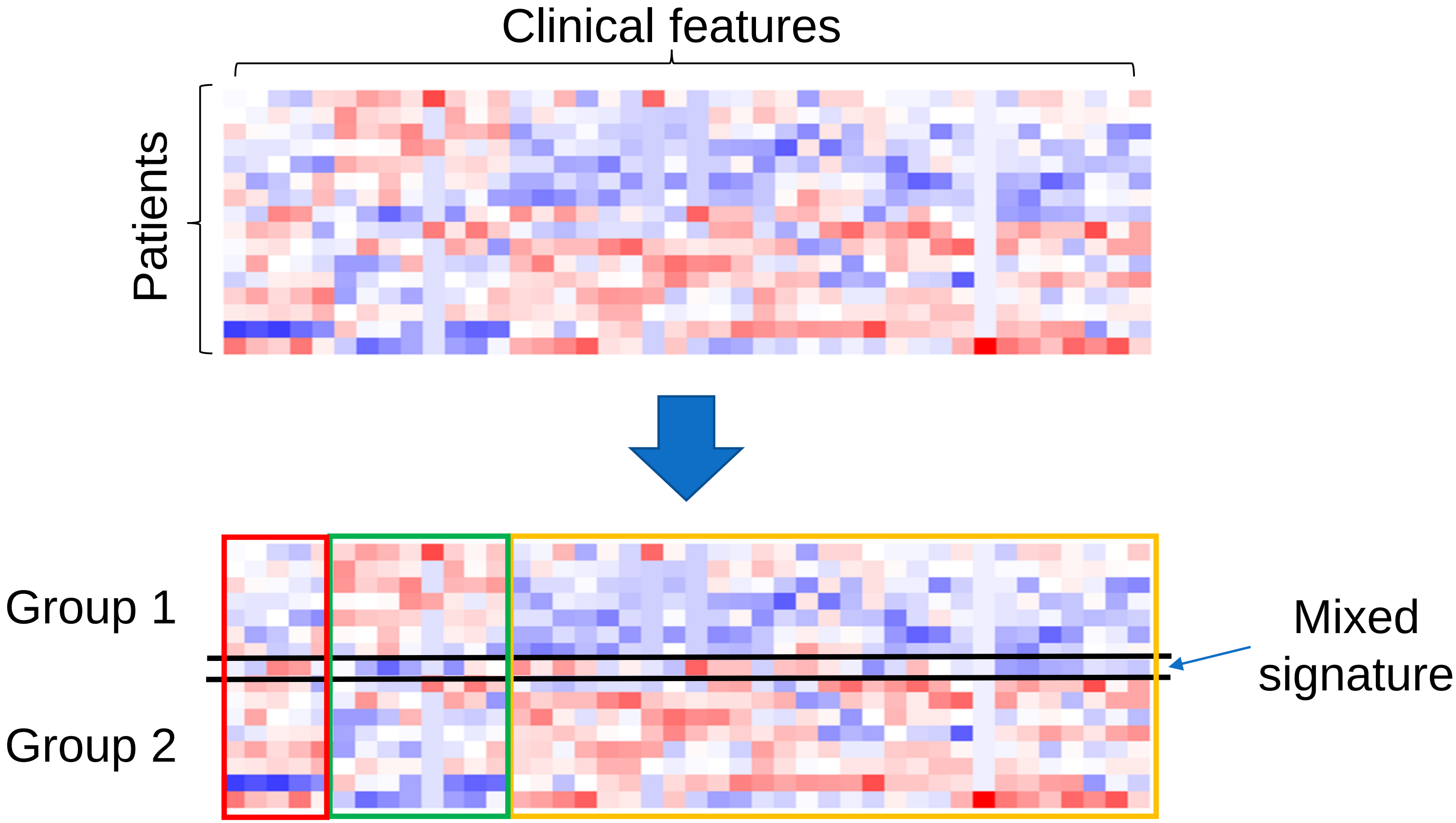
Unsupervised learning



https://github.com/benedekrozemberczki/awesome-community-detection

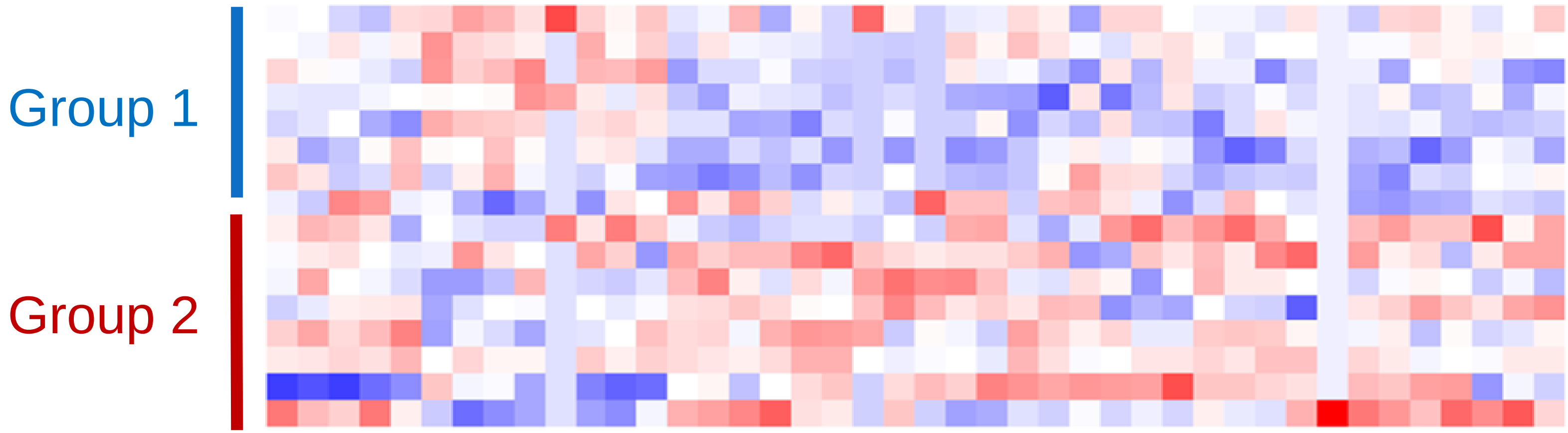
- Pattern recognition through data connectivity

Unsupervised learning



- Pattern recognition through similarity

Patterns are defined by distances



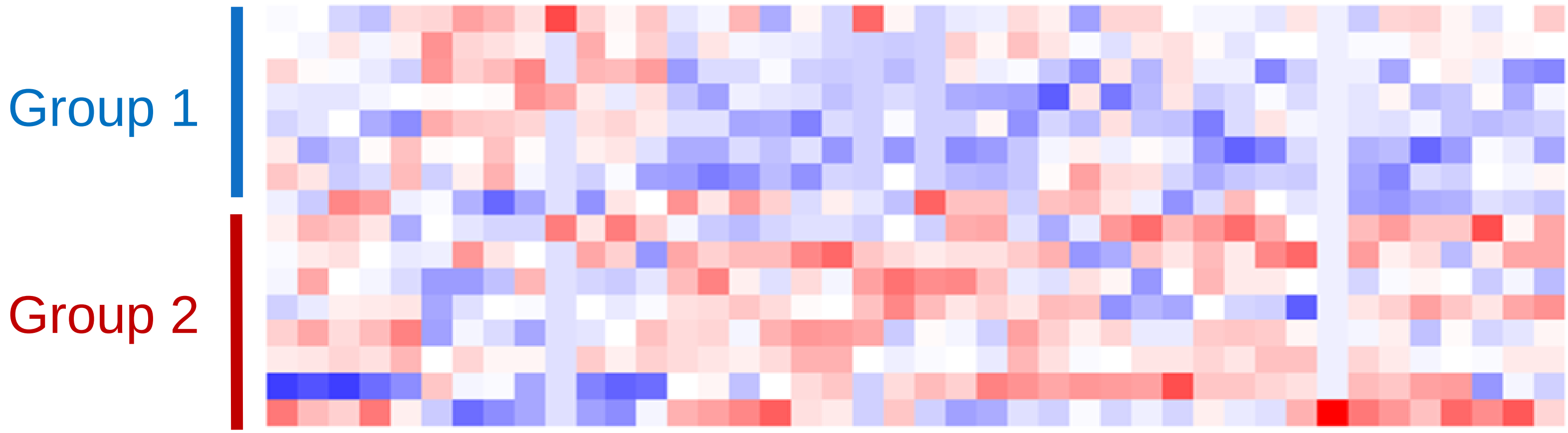
Distance

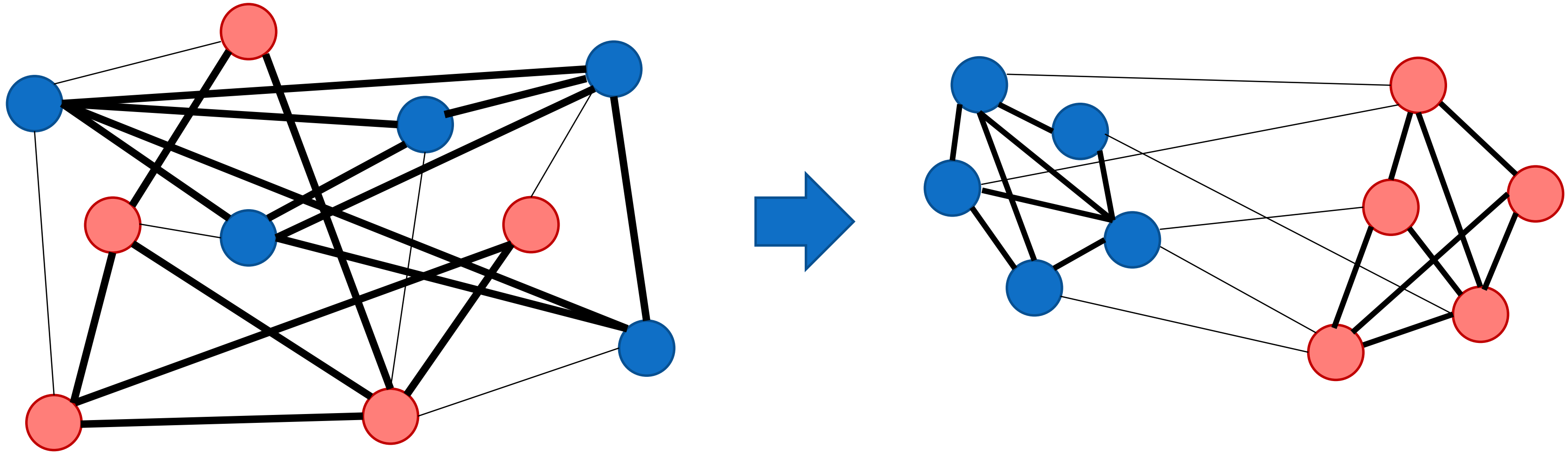
Small	Large
Large	Small

Similarity

High	Low
Low	High

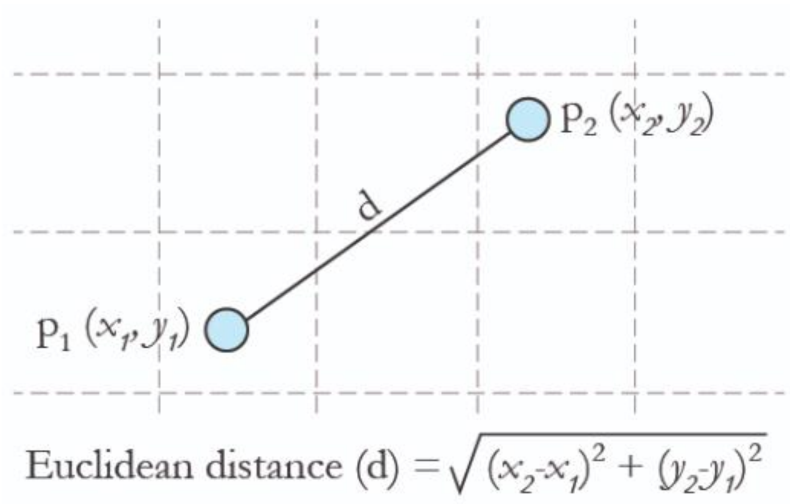
Patterns are defined by distances



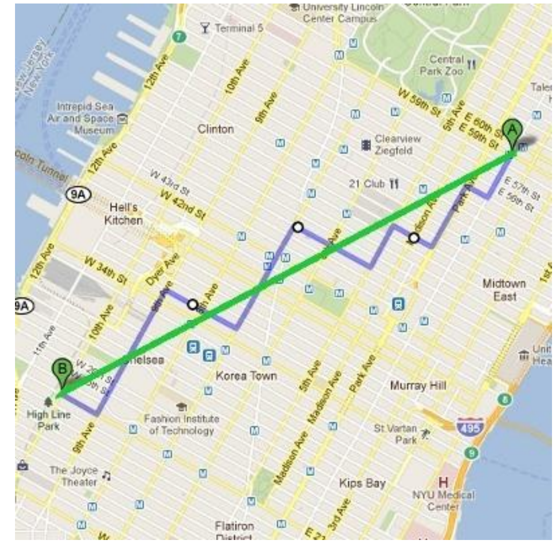


Thick edges = small distances = high similarities

Choices of distance measurement



www.tutorialexample.com



www.quora.com

- Euclidean distance = $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$
- Manhattan distance = $|x_1 - x_2| + |y_1 - y_2|$
- Pearson and Spearman correlation coefficients
- Cosine similarity = $\frac{\overrightarrow{u_1} \cdot \overrightarrow{u_2}}{|u_1||u_2|} = \frac{x_1 x_2 + y_1 y_2}{\sqrt{{x_1}^2 + {y_1}^2}\sqrt{{x_2}^2 + {y_2}^2}}$
- What would be an appropriate distance measurement between the clinical profiles of two patients?

Features can have different scales and units

study_id	age	sex	tumor_size	surg	tace	embo	cmt	total_dose	no_fx	dose_fx
2	53	0	3.7	0	0	0	0	30.0	10	3.0
3	54	1	2.0	0	1	0	0	30.0	10	3.0
4	53	1	7.4	0	0	0	0	45.0	25	1.8
5	41	1	5.8	1	1	0	0	30.0	10	3.0
6	54	1	14.4	0	1	0	0	30.0	10	3.0

- $d_{\mathrm{Euclidean}}(\mathrm{p}_1, \mathrm{p}_4) = \sqrt{(53-41)^2 + (0-1)^2 + \cdots + (3-3)^2}$
- What is the unit of distance?
- What would correlation between these data look like?

Data standardization

study_id	age	sex	tumor_size	surg	tace	embo	cmt	total_dose	no_fx	dose_fx
2	53	0	3.7	0	0	0	0	30.0	10	3.0
3	54	1	2.0	0	1	0	0	30.0	10	3.0
4	53	1	7.4	0	0	0	0	45.0	25	1.8
5	41	1	5.8	1	1	0	0	30.0	10	3.0
6	54	1	14.4	0	1	0	0	30.0	10	3.0

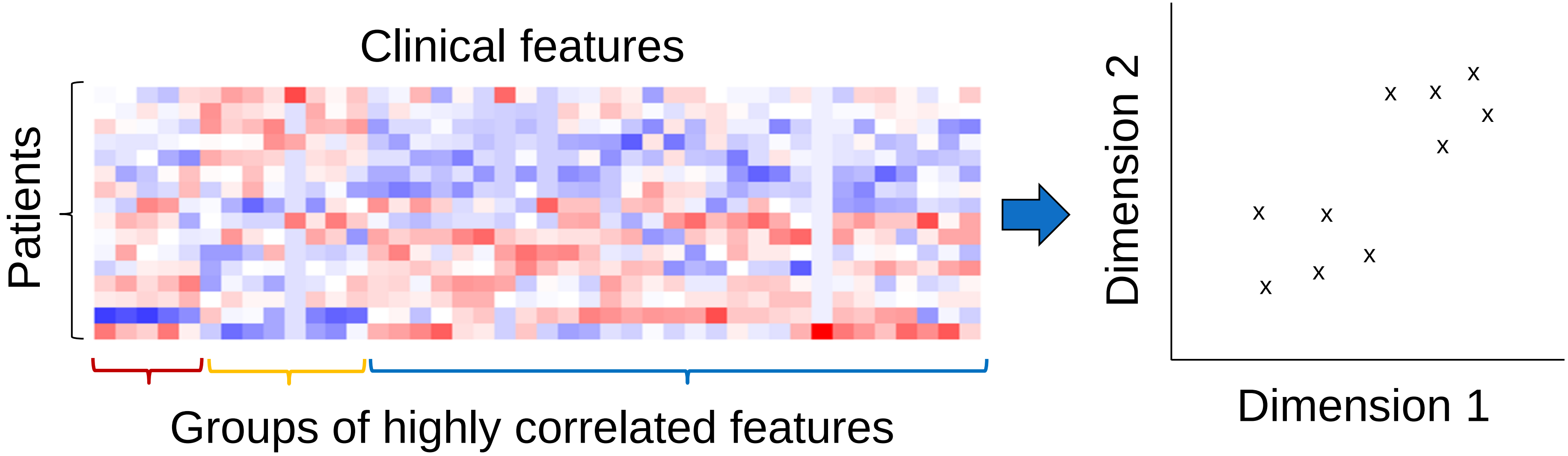
study_id	age	sex	tumor_size	surg	tace	embo	cmt	total_dose	no_fx	dose_fx
2	-0.712257	-2.186163	-0.816166	-0.362132	-1.336953	-0.148093	-0.25287	-0.733512	-0.259310	-0.416553
3	-0.626058	0.455451	-1.170211	-0.362132	0.744746	-0.148093	-0.25287	-0.733512	-0.259310	-0.416553
4	-0.712257	0.455451	-0.045598	-0.362132	-1.336953	-0.148093	-0.25287	0.835401	2.025241	-0.977328
5	-1.746647	0.455451	-0.378817	2.749521	0.744746	-0.148093	-0.25287	-0.733512	-0.259310	-0.416553
6	-0.626058	0.455451	1.412233	-0.362132	0.744746	-0.148093	-0.25287	-0.733512	-0.259310	-0.416553

- $x_{\text{standardized}} = \dfrac{x - \text{mean}}{\text{s.d.}}$ $\qquad \text{s. d.} = \sqrt{\dfrac{\sum(x_i - \text{mean})^2}{N}}$
- What’s the mean of a standardized feature?
- What’s the standard deviation of a standardized feature?

Some derivations

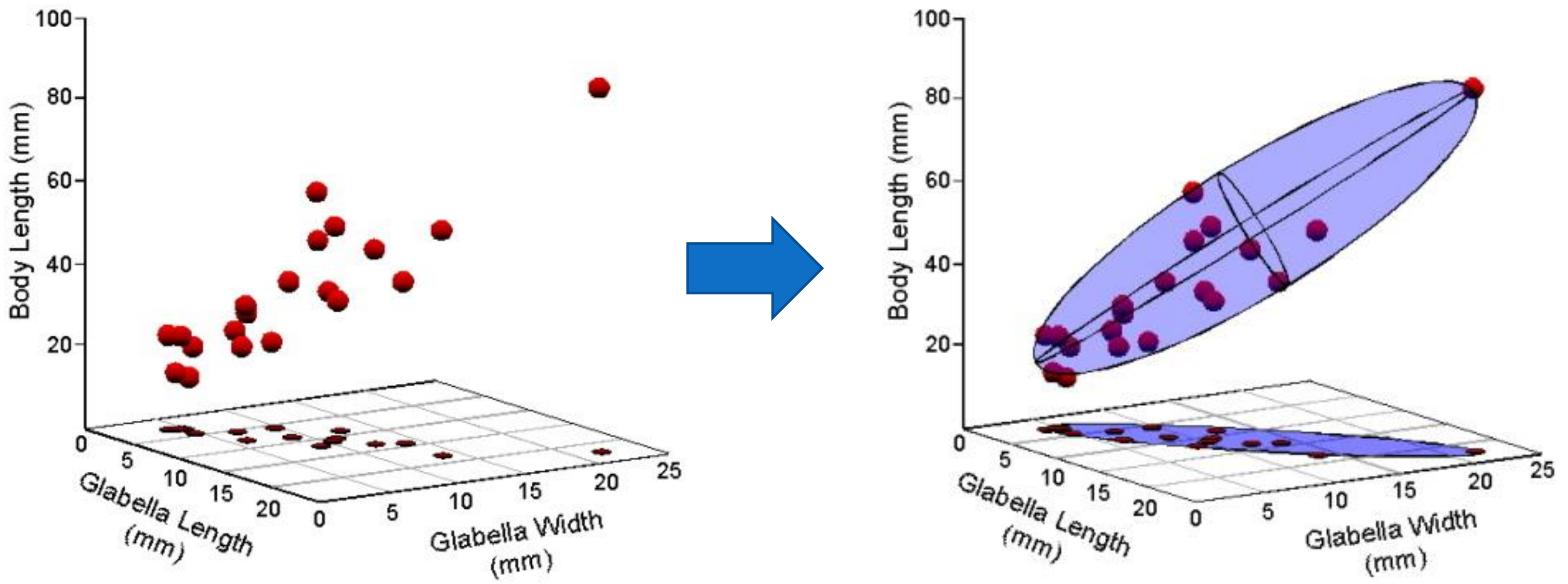
- $z = \frac{x-\mu}{\sigma}$ $\sigma = \sqrt{\frac{\sum(x_i-\mu)^2}{N}}$ $\mu = \frac{\sum x_i}{N}$
- Mean of z's $= \frac{\sum z_i}{N} = \frac{\sum \frac{x_i-\mu}{\sigma}}{N} = \frac{\sum(x_i-\mu)}{N\sigma} = \frac{(\sum x_i)-N\mu}{N\sigma} = 0$
- S.D. of z's $= \sqrt{\frac{\sum {z_i}^2}{N}} = \sqrt{\frac{\sum\left(\frac{x_i-\mu}{\sigma}\right)^2}{N}} = \sqrt{\frac{\sum(x_i-\mu)^2}{N\sigma^2}} = \frac{1}{\sigma}\sqrt{\frac{\sum(x_i-\mu)^2}{N}} = 1$
- What is the unit of z's?

Dimensionality reduction



- Highly correlated / redundant features should be collapsed
- The major pattern (two groups of patients) can be sufficiently represented with just a few features
- Visualization on 2D or 3D for human eyes

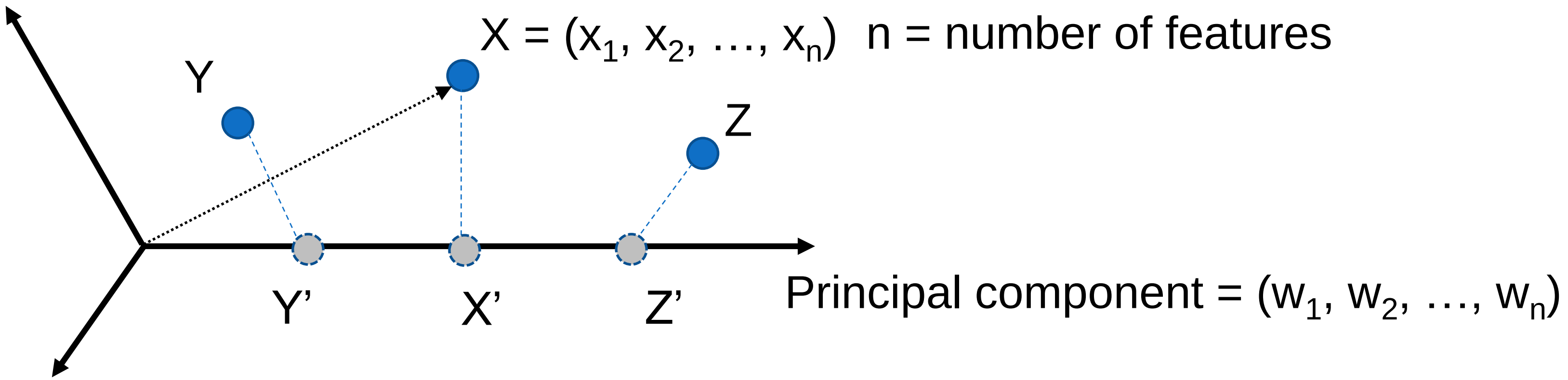
Principal component analysis (PCA)



Source: the paleontological association

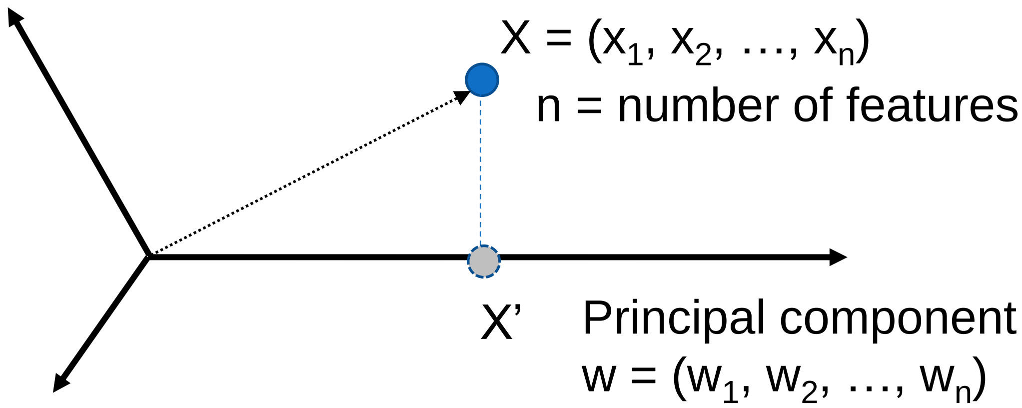
- Fit an *n*-dimensional ellipsoid to the data cloud
  - Axes of the ellipsoid are the principal components (dimensions)
  - Axes are orthogonal
  - Larger axis = more variance of data along that axis

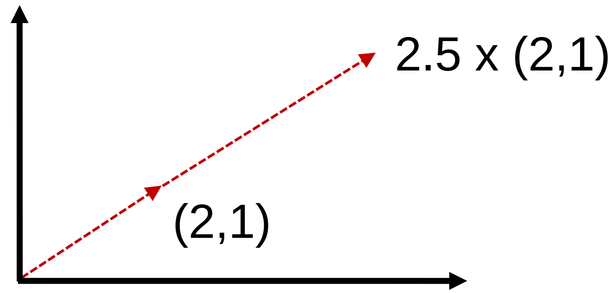
Variance of data along an axis



- Original data belong to an *n*-dimensional space
- A principal component is a direction in *n*-dimensional space and can be characterized by (w_1, …, w_n)
- Data points can be projected onto this 1D axis and the variance of the projection can be calculated

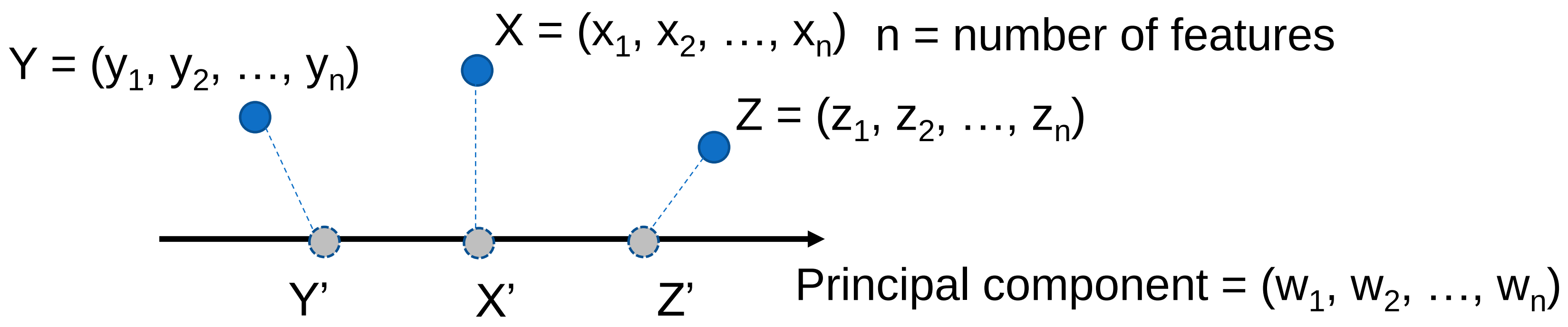
Projection of point onto a line





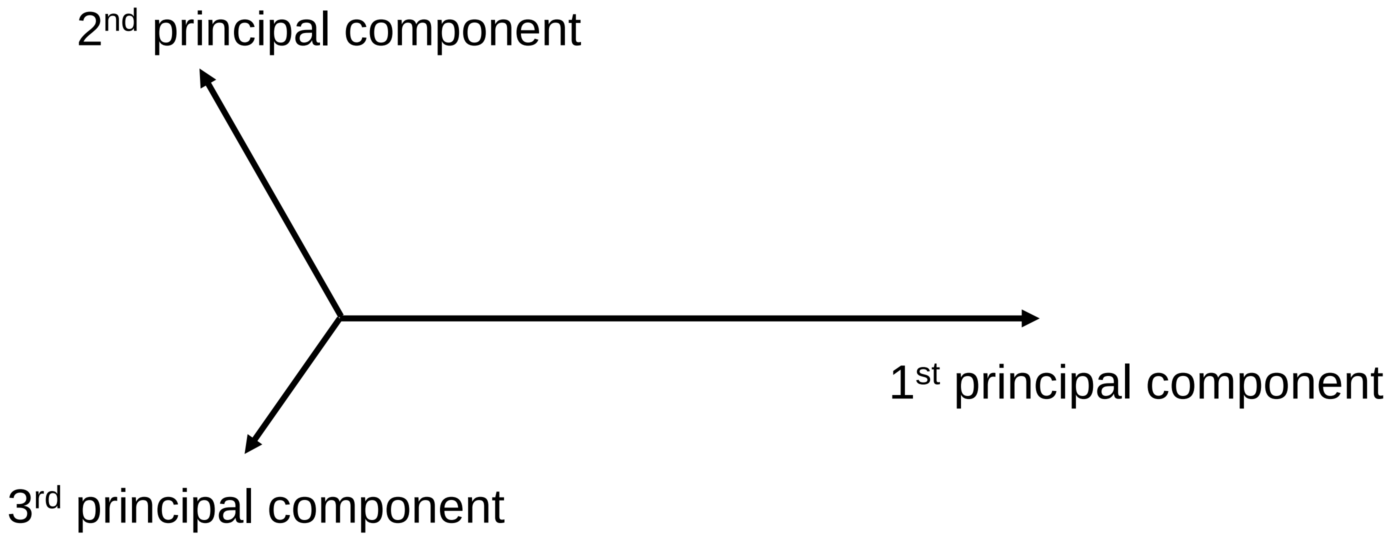
- $X' = (\alpha w_1, \alpha w_2, \ldots, \alpha w_n)$, α is a real number
- $\overrightarrow{XX'} = (\alpha w_1 - x_1, \alpha w_2 - x_2, \ldots, \alpha w_n - x_n)$
- The vector XX' is orthogonal to the principal component
  - Angle between XX' and $(w_1, w_2, \ldots, w_n)$ is 90 degree
  - Dot product between XX' and $(w_1, w_2, \ldots, w_n) = 0$
  - $(\alpha w_1 - x_1)w_1 + (\alpha w_2 - x_2)w_2 + \cdots + (\alpha w_n - x_n)w_n = 0$
  - $\alpha \sum w_i^2 - \sum w_i x_i = 0$
  - $\alpha = \sum w_i x_i / \sum w_i^2$ ← Let's consider $\sum w_i^2 = 1$, so $\alpha = \sum w_i x_i = w \cdot X$
- $X' = (w \cdot X)w$

Finding the "best" principal component



- $X' = w \cdot X$, $Y' = w \cdot Y$, and $Z' = w \cdot Z$
- Mean of X', Y', Z' $= w \cdot \left(\frac{x_1+y_1+z_1}{3}, \ldots, \frac{x_n+y_n+z_n}{3}\right) = w \cdot \mu = 0$
- Variance of X', Y', Z' $= \frac{(w \cdot X)^2 + (w \cdot Y)^2 + (w \cdot Z)^2}{3}$
- The best principal component is the *w* that maximize the variance $\frac{(w \cdot X)^2 + (w \cdot Y)^2 + (w \cdot Z)^2}{3}$ subject to $\sum w_i^2 = 1$

Finding the next best principal components



- The second-best principal component is the *w* that maximize the variance $\frac{(w \cdot X)^2 + (w \cdot Y)^2 + (w \cdot Z)^2}{3}$ subject to $\sum w_i^2 = 1$ and is orthogonal to the best principal component
- And so on…
- In practice, finding the principal components is equivalent to finding the eigenvectors and eigenvalues of the data matrix

PCA on raw data

study_id	age	sex	tumor_size	surg	tace	embo	cmt	total_dose	no_fx	dose_fx
2	53	0	3.7	0	0	0	0	30.0	10	3.0
3	54	1	2.0	0	1	0	0	30.0	10	3.0
4	53	1	7.4	0	0	0	0	45.0	25	1.8
5	41	1	5.8	1	1	0	0	30.0	10	3.0
6	54	1	14.4	0	1	0	0	30.0	10	3.0

- What would likely be the first principal component?
  - **Hint**: The first principal component is the direction that maximize the variance of the data points
  - **Hint**: Variance scales linearly with magnitude of data value

Key behaviors of PCA

- The projection $X' = (w \cdot X)w$ is a linear combination of the original *n* features
  - We can look at *w* to interpret feature-level contribution
  - *w* = (–10, 1, 0.2, 3) means that the first feature is quite important here

- PCA = rotation of the original axes
  - PCA preserve Euclidean distances between data points
  - PCA does not work well when Euclidean distance is inappropriate

- Highly correlated features tend to be grouped together in the same principal component

- PCA is a good initial step for more-complex algorithm

- PCA is generally deterministic

PCA in Python

sklearn.decomposition.PCA

class `sklearn.decomposition.PCA`(*n_components=None, *, copy=True, whiten=False, svd_solver='auto', tol=0.0, iterated_power='auto', random_state=None*) [source]

Returned values that you can use

Attributes:

components_ : *ndarray of shape (n_components, n_features)*

Principal axes in feature space, representing the directions of maximum variance in the data. The components are sorted by `explained_variance_`.

explained_variance_ : *ndarray of shape (n_components,)*

The amount of variance explained by each of the selected components.

Equal to n_components largest eigenvalues of the covariance matrix of X.

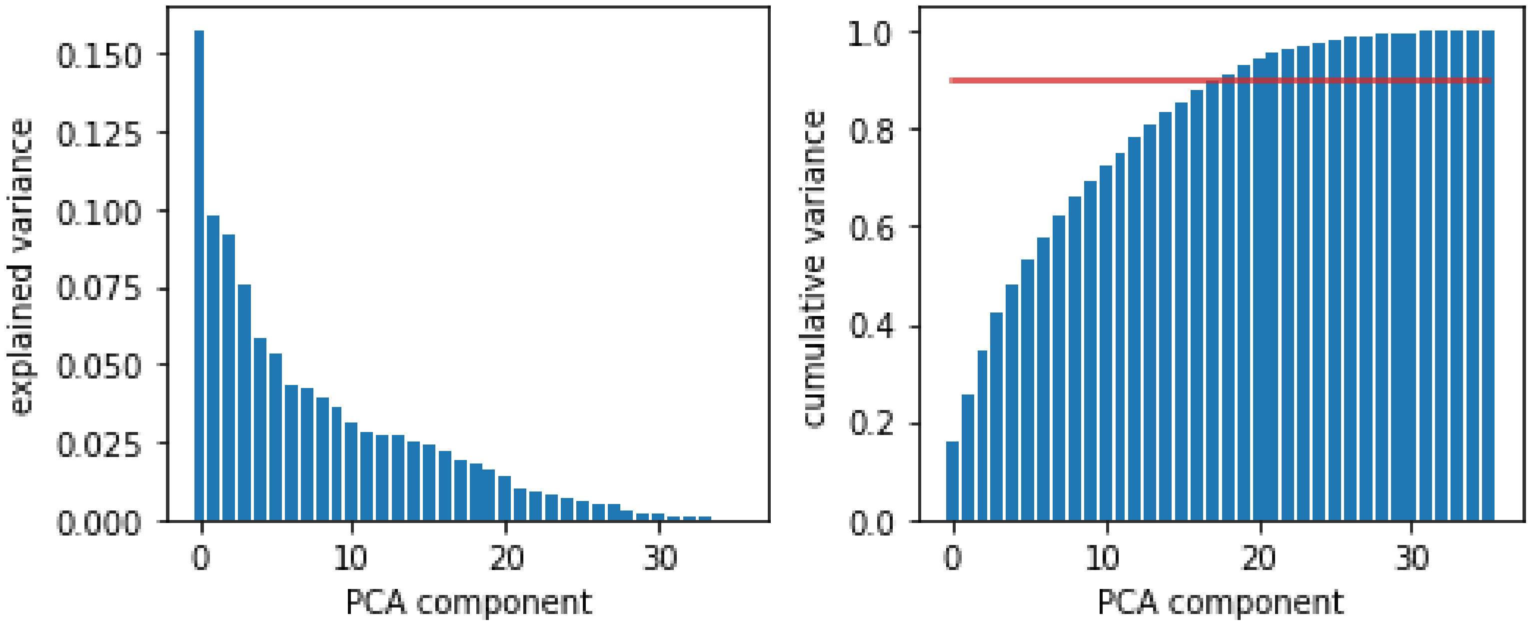
New in version 0.18.

explained_variance_ratio_ : *ndarray of shape (n_components,)*

Percentage of variance explained by each of the selected components.

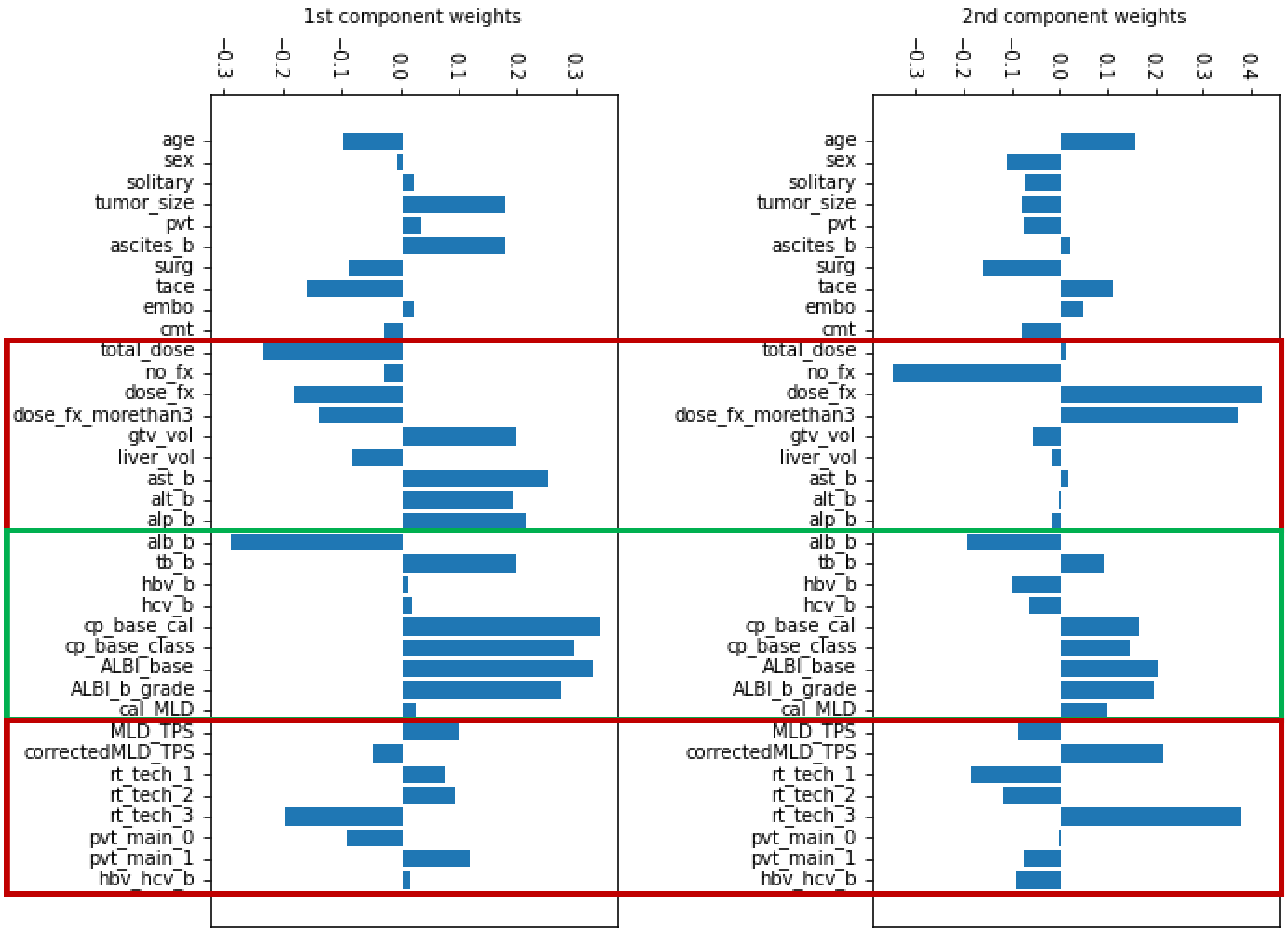
If `n_components` is not set then all components are stored and the sum of the ratios is equal to 1.0.

Explained variance ratio

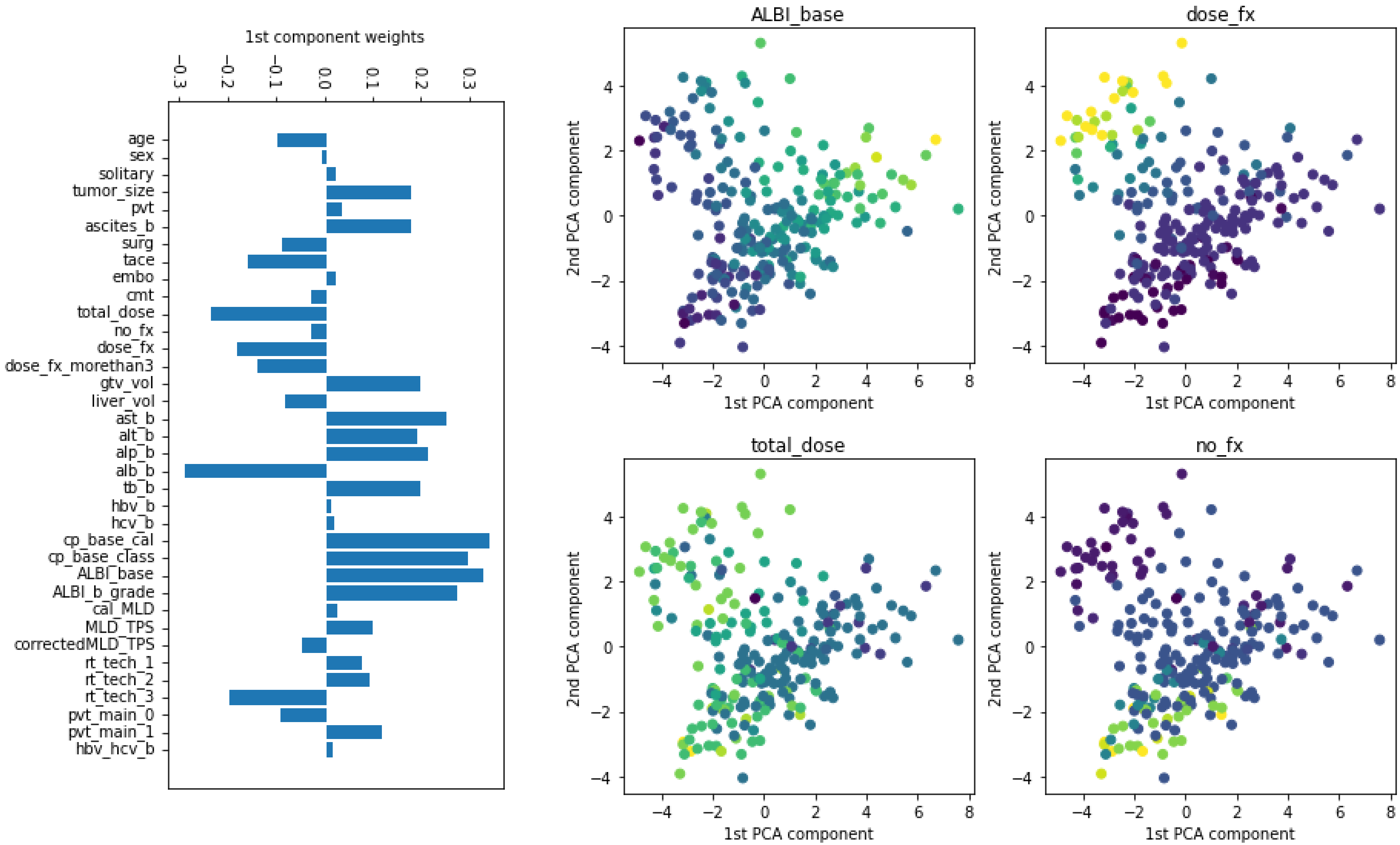


- Components that capture high variances are typically useful but not always
- Explained variance trend suggests the true dimension of data

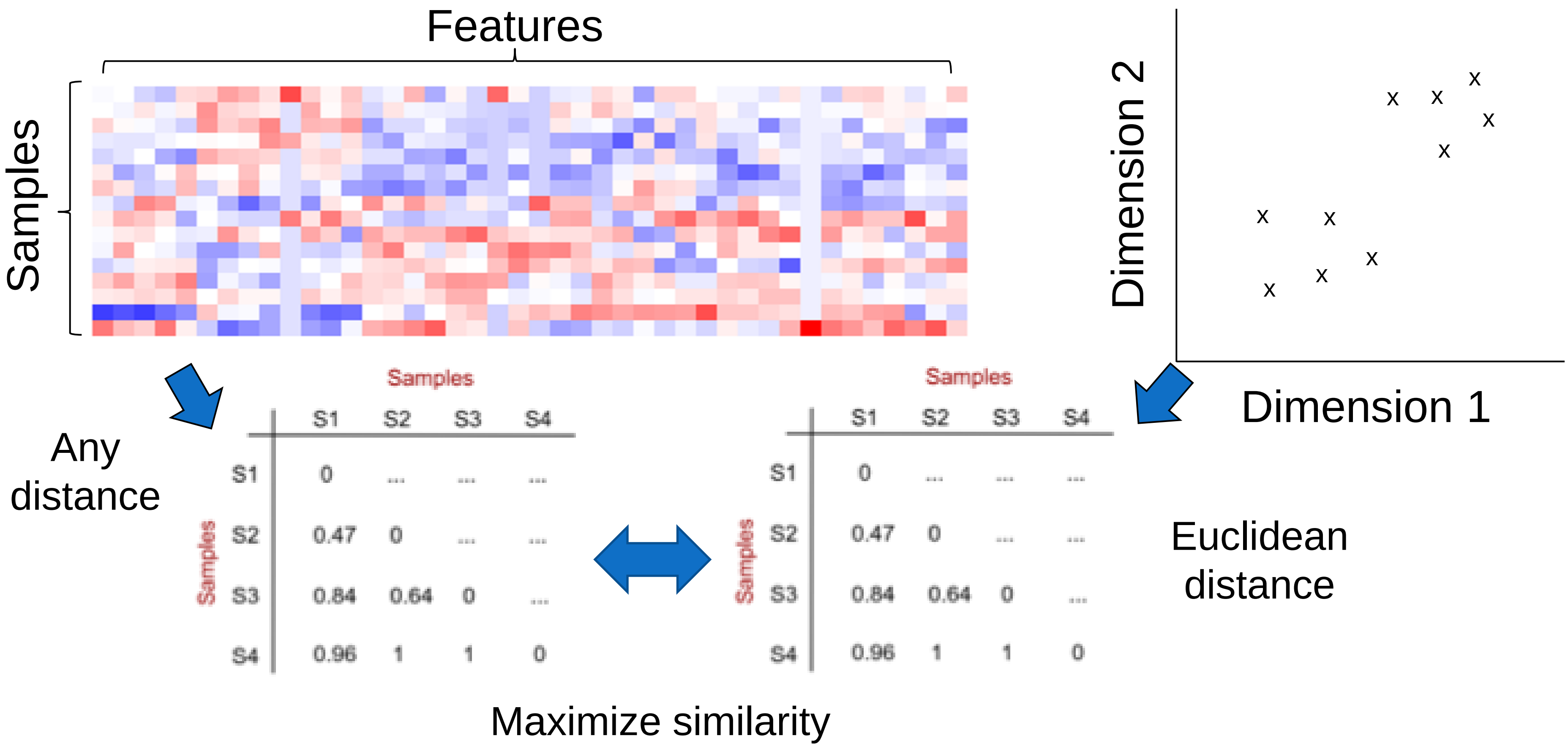
Principal component's weights



PCA-transformed data



Multidimensional scaling (MDS)

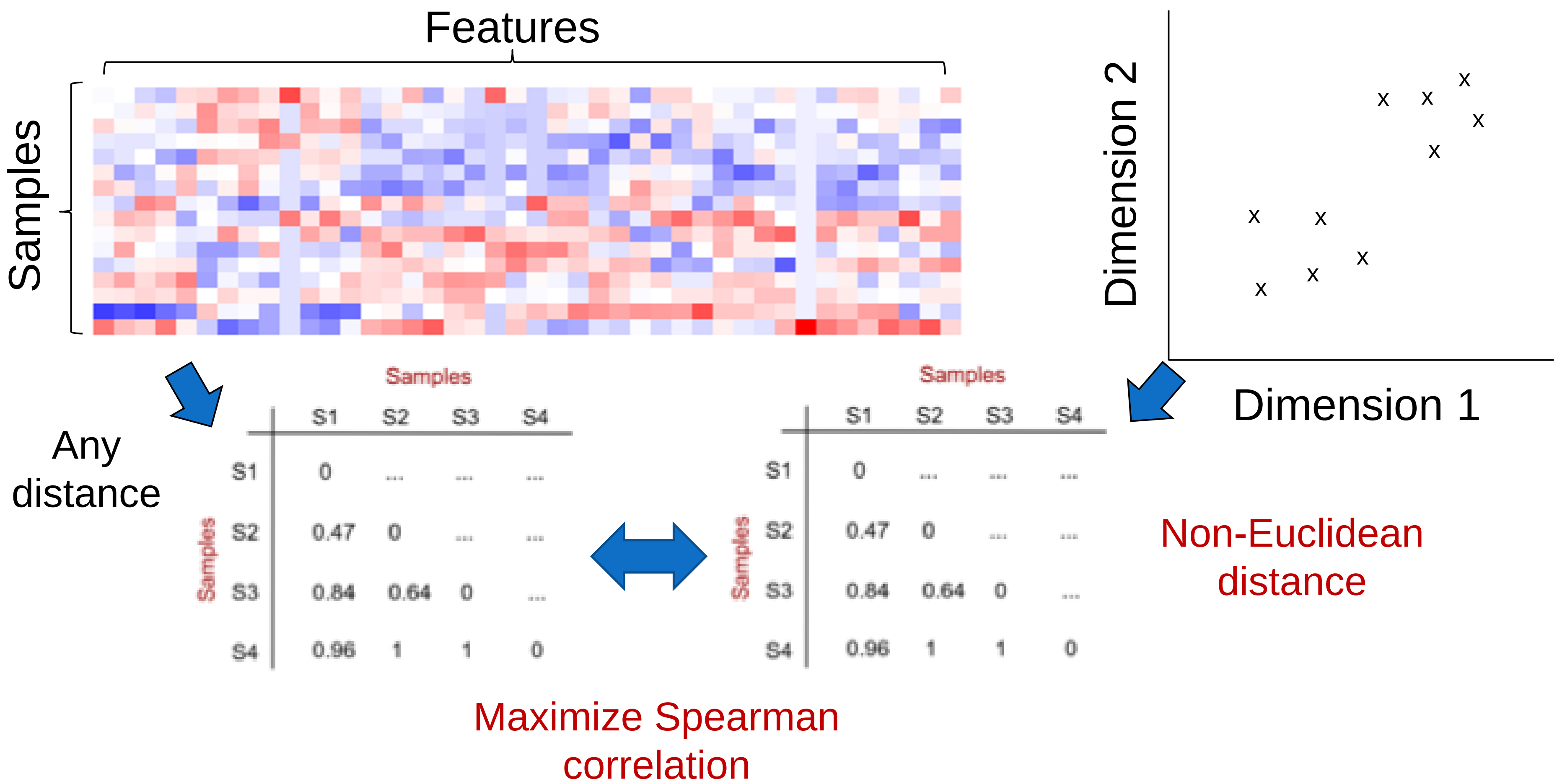


- MDS projects data points onto new dimensions while trying to preserve the similarity between two distance matrices
  - **For example**: Maximize Pearson or Spearman correlation

Principal Coordinate Analysis (PCoA)

- Also called Classical MDS
- Users provide distance matrix $d(X_i, X_j)$
- Let Y_i be a projection of X_i onto a new *k*-dimensional space
  - This induces Euclidean distances $d_{\text{Euclidean}}(Y_i, Y_j)$
- Pearson correlation between $d(X_i, X_j)$ and $d_{\text{Euclidean}}(Y_i, Y_j)$ can be calculated as a function of Y_i's
  - Solve for Y_i's that maximize this
- In practice, finding Y_i's is related to finding the eigenvectors and eigenvalues of some matrix (related to $d(X_i, X_j)$)

Non-classical MDS



- Non-metric MDS
- Generalized MDS

MDS in Python

sklearn.manifold.MDS

class `sklearn.manifold.MDS`(*n_components=2, *, metric=True, n_init=4, max_iter=300, verbose=0, eps=0.001, n_jobs=None, random_state=None, dissimilarity='euclidean'*) [source]

Parameters:

n_components : *int, default=2*
Number of dimensions in which to immerse the dissimilarities.

metric : *bool, default=True*
If `True`, perform metric MDS; otherwise, perform nonmetric MDS.

dissimilarity : *{'euclidean', 'precomputed'}, default='euclidean'*
Dissimilarity measure to use:

- **'euclidean':**
  Pairwise Euclidean distances between points in the dataset.
- **'precomputed':**
  Pre-computed dissimilarities are passed directly to `fit` and `fit_transform`.

- Default (metric = True) is Principal Coordinate Analysis

Any question?