Unsupervised Learning
Dimension Reduction

Nezar Abdennur

(with slides adapted from Zhiping Weng)

Types of statistical learning problems

Unsupervised learning:

- Only predictor vector x_i is observed for every measurement.
- Understand the underlying structure of X, find patterns and relationship between features etc.

e.g.
Given gene expression data under different conditions, find out which genes are co-expressed.

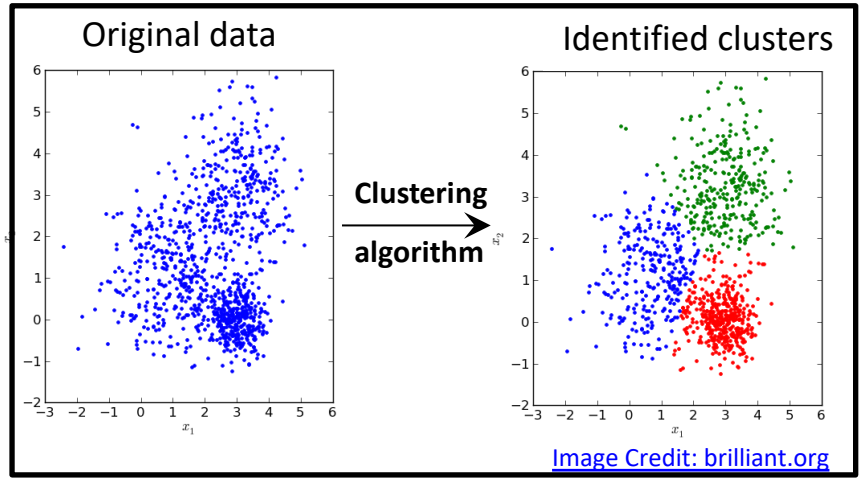
Supervised learning:

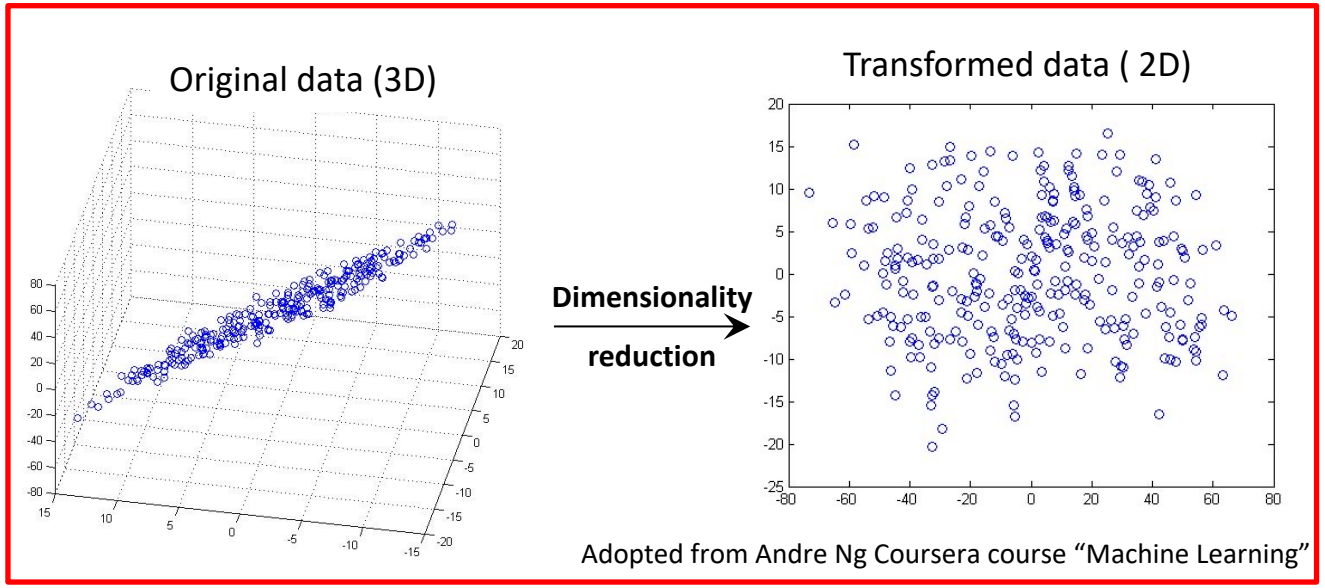
- For every predictor vector x_i the associated response y_i is known.
- Estimate f(x) such that $\hat{f}$:X → Y
- Use $\hat{f}$ to predict response y for new x or understand the relationship between x and y.

e.g.
Given biomarker levels under diseased and normal conditions, use the data to predict whether a patient has the disease or not.

Adapted from *An Introduction to Statistical Learning*

Unsupervised : Clustering, Dimensionality Reduction





Summary

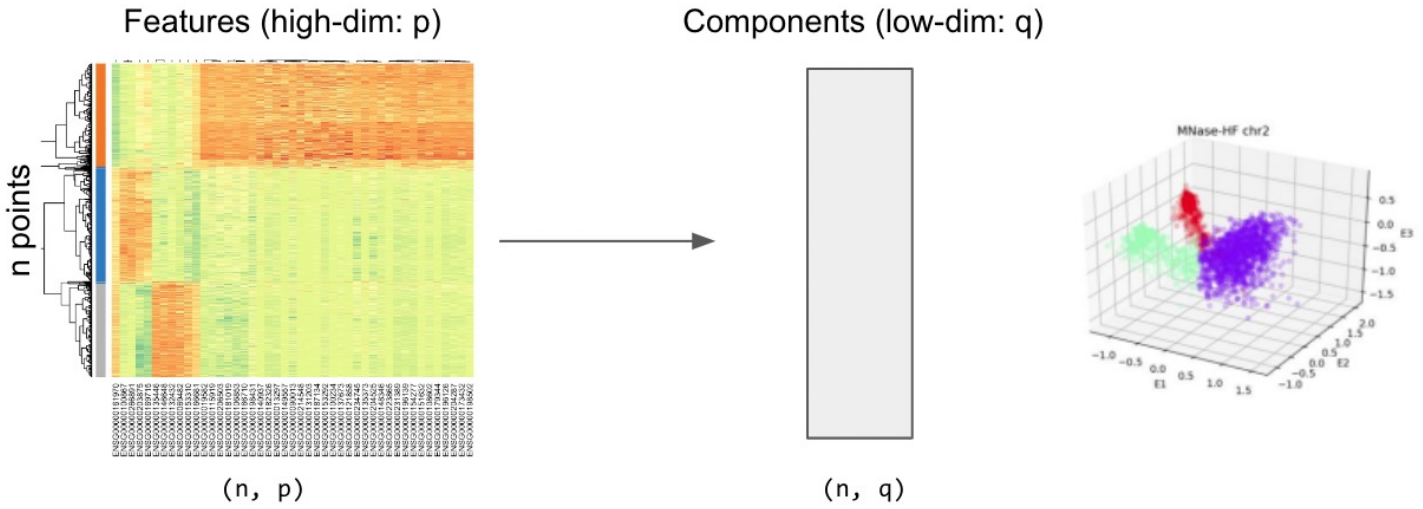
Response or Output (Y)	Available	Not available	Partially available
	Supervised	**Unsupervised**	**Semi-supervised**
Discrete / Categorical	Classification	Clustering	**Unsupervised techniques** to learn structure of data, **supervised techniques** to make best guess predictions
Continuous	Regression	Dimensionality reduction	
	Find function f: X → Y	Infer the underlying structure of X	

Adapted from *An Introduction to Statistical Learning*

Dimensionality reduction

Many modern data have huge numbers of features / dimensions

- Documents: thousands of words, millions of bigrams
- Images: thousands to millions of pixels
- Genomics: thousands of genes, millions of DNA polymorphisms



Why reduce dimensions?

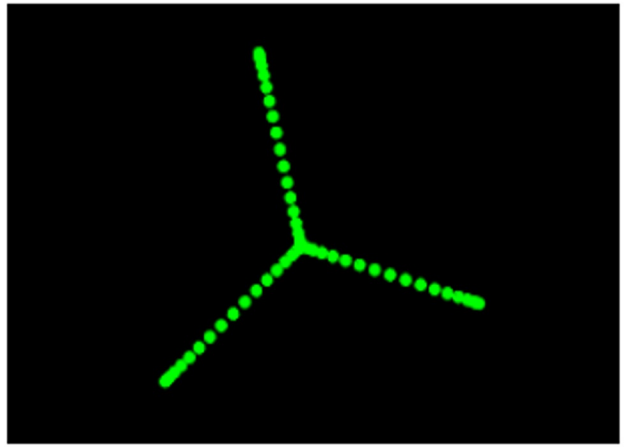
High dimensionality has many costs:

- Redundant and irrelevant features degrade performance of machine learning algorithms.
- Difficulty in interpretation and visualization
- Curse of dimensionality
- Computation may become prohibitive—what if your algorithm scales as $O(p^3)$ or $O(p^4)$?

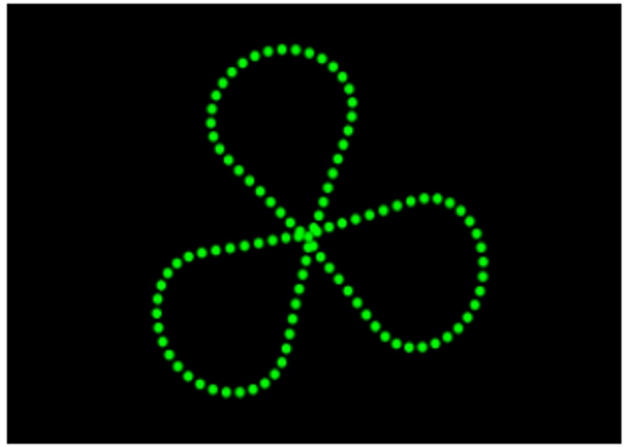
Dimension reduction for visualization

- Very difficult to visualize > 3-dimensional data
- Want to preserve as many higher order structures as possible
- Linear vs nonlinear

Original: 3 mutually perpendicular circles in 6-dimensional space, meeting at a point



(a) Projection by PCA does not preserve the structure of the dataset — it is unclear that it consists of three circles



(b) The Sammon mapping preserves the topological structure — while the circles become distorted, there are still three closed loops meeting at a single point

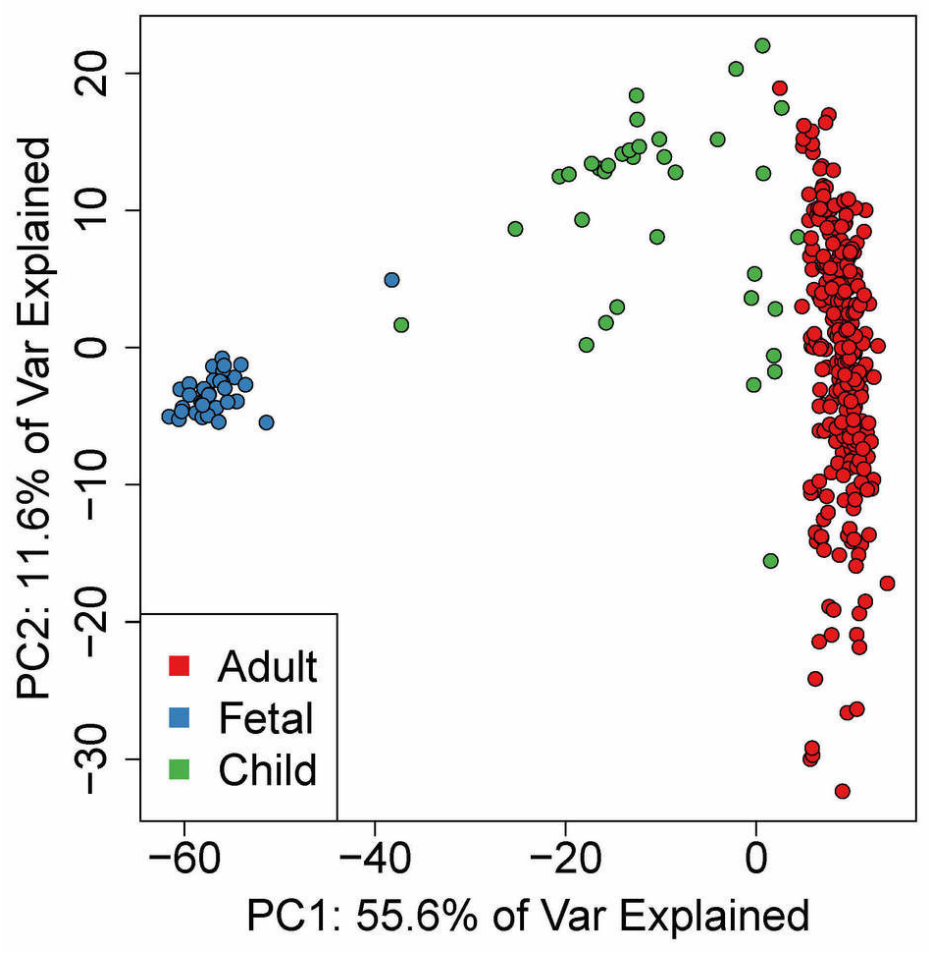
Part 1. Principal Component Analysis (PCA)

Principal Component Analysis (PCA)

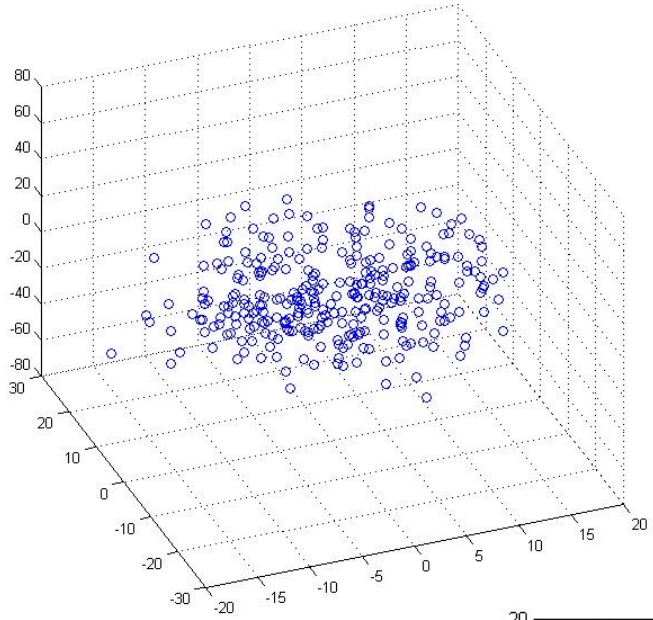
- Linear decomposition
- Use a small number of representative variables (the largest PCs) to collectively explain most of the variance in the original data
- The focus is on keeping distant data points far apart

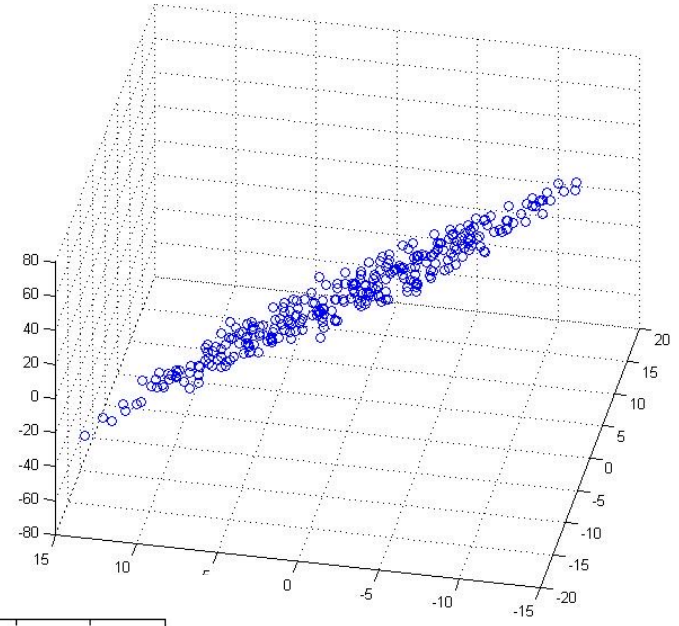
Mapping DNA methylation across development, genotype and schizophrenia in the human frontal cortex
Andrew E Jaffe, …, Joel E Kleinman
Nature Neuroscience 19, 40–47 (2016) doi:10.1038/nn.4181

Suppl. Fig. 1. Principal component analysis (PCA) demonstrates genome-wide changes in DNAm comparing pre- and post-natal samples

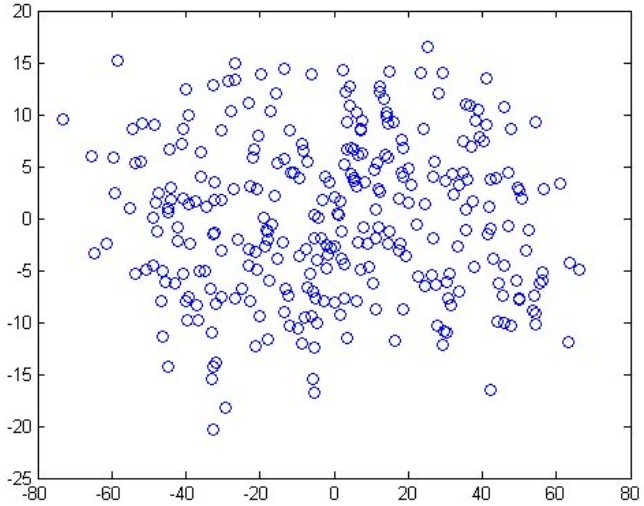


What is PCA? 3D to 2D





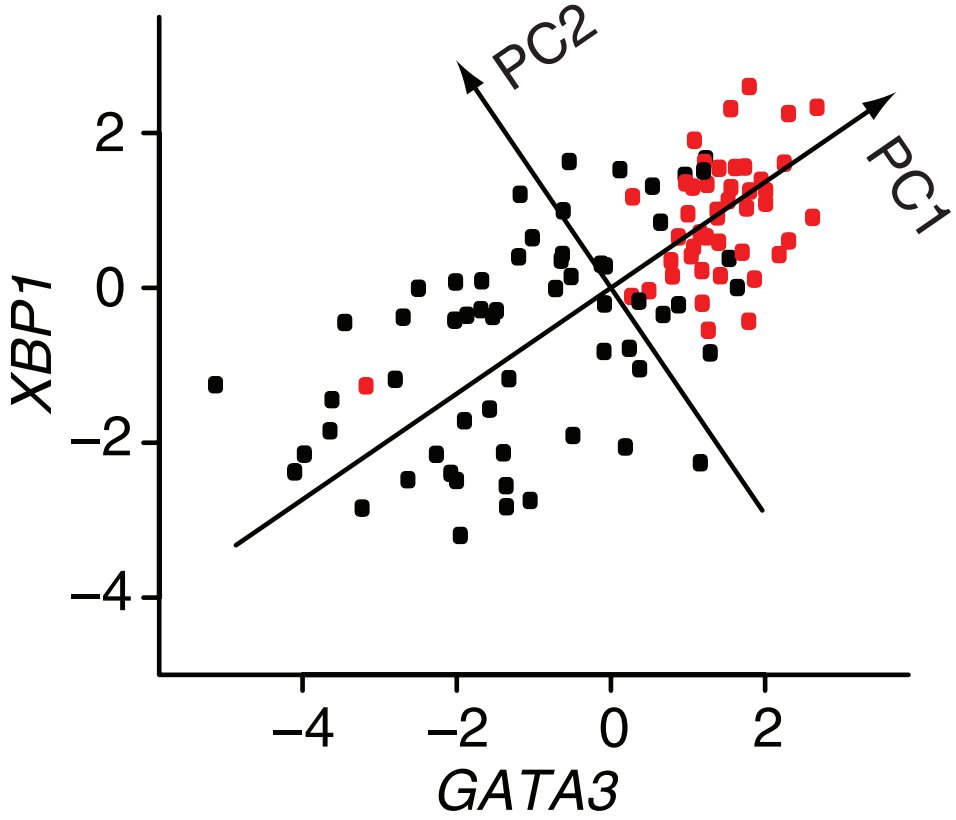
Data points can be found concentrated on a 2D plane.



Adopted from Andre Ng Coursera course “Machine Learning”

What is PCA?

Data points can be found concentrated along a single line.



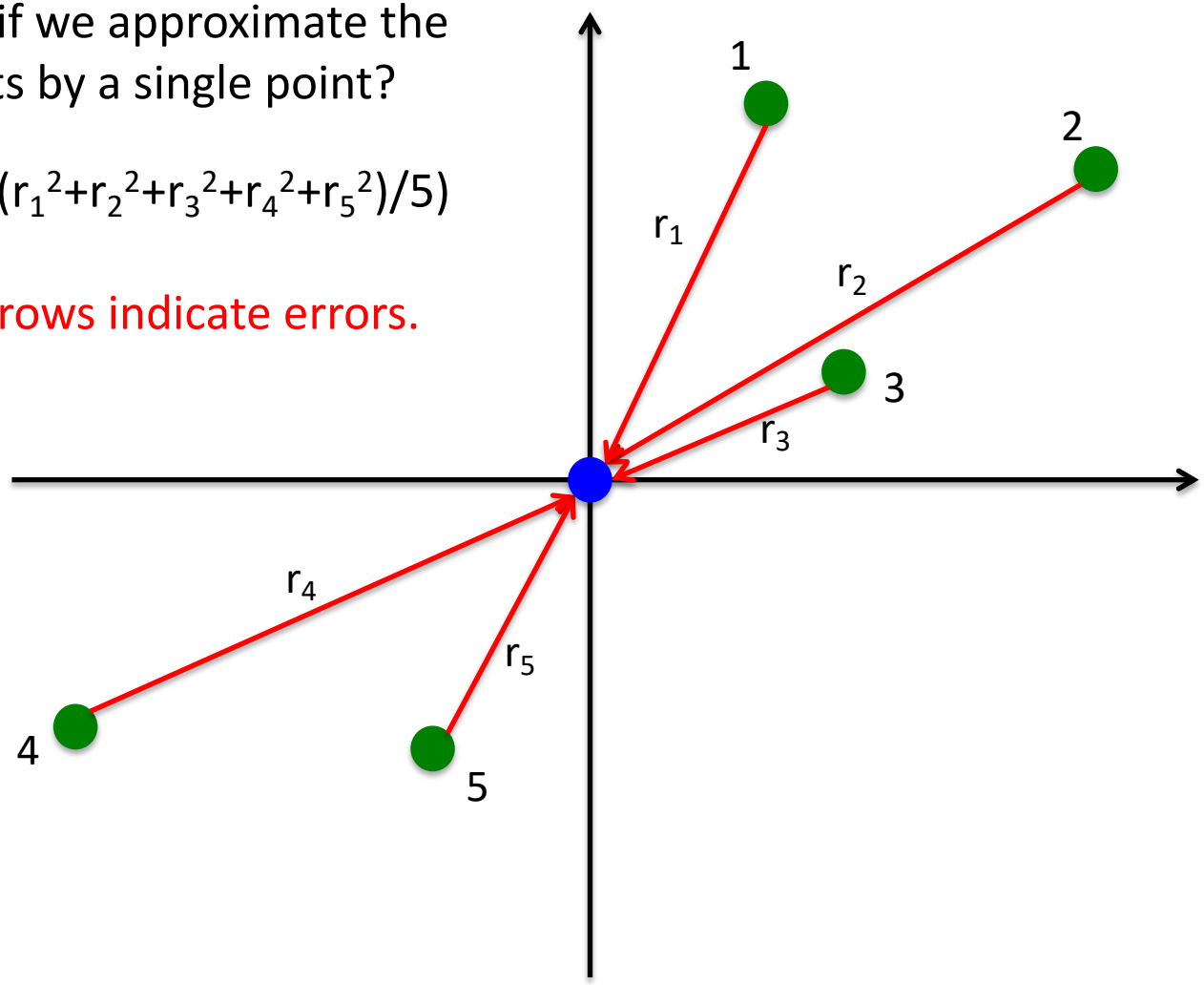
Markus Ringner, Nature Biotechnology, vol. 26, p. 303-304, 2008

PCA conceptually: simplify data

What is the error if we approximate the cloud of five points by a single point?

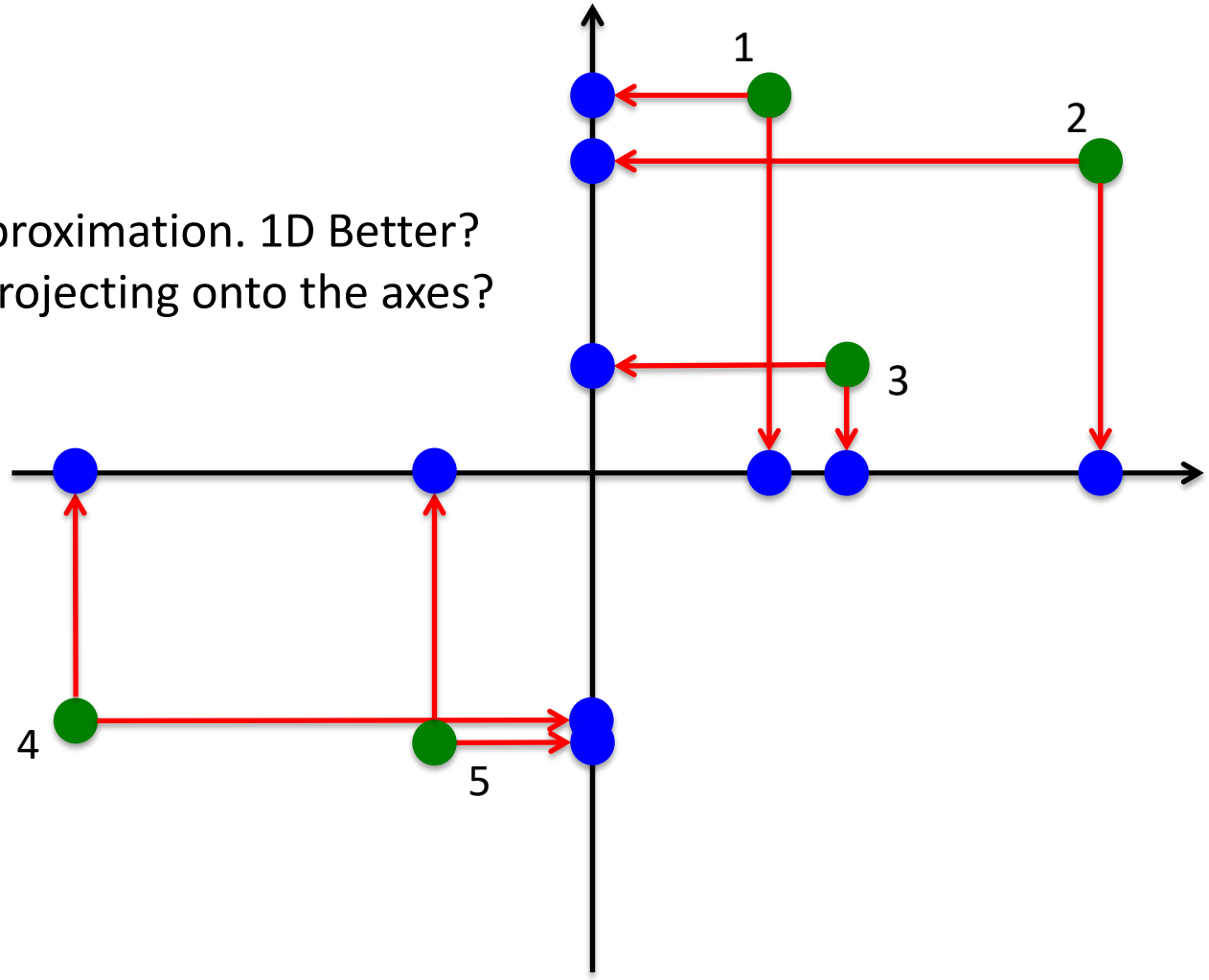
RMSE = Square root($(r_1^2+r_2^2+r_3^2+r_4^2+r_5^2)/5$)

The lengths of red arrows indicate errors.



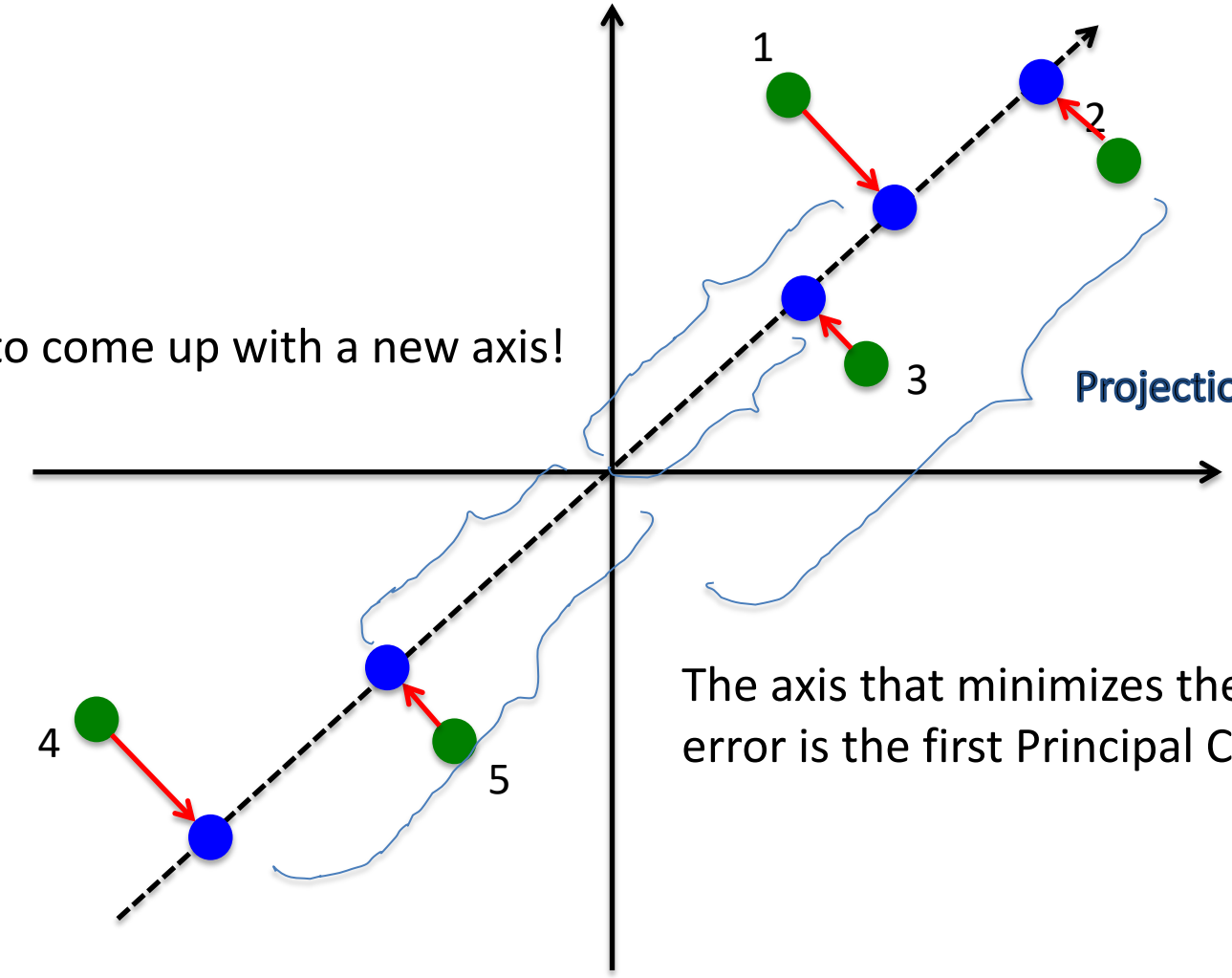
PCA conceptually: simplify data

0D is bad approximation. 1D Better?
How about projecting onto the axes?



PCA conceptually: simplify data

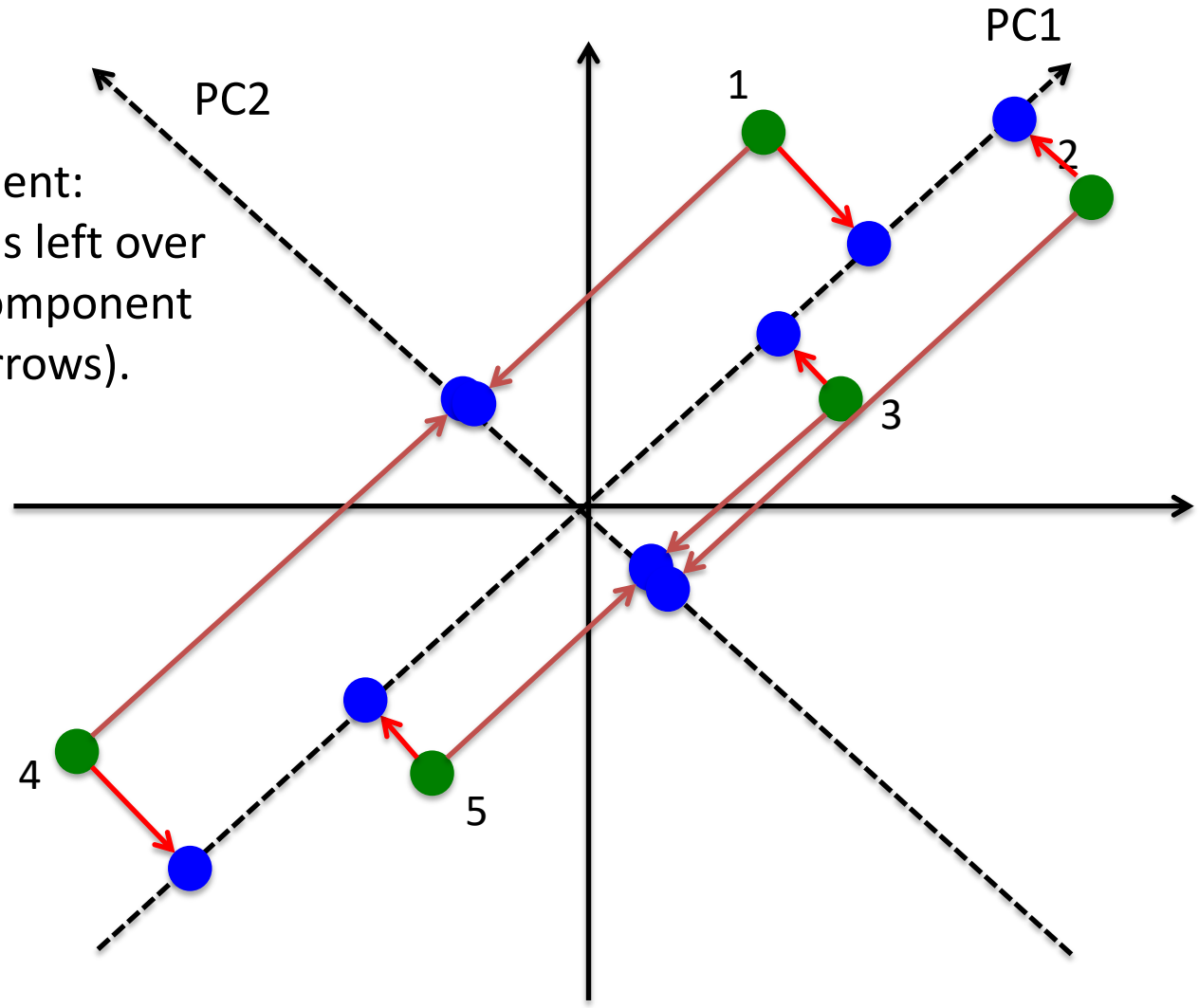
Better idea to come up with a new axis!



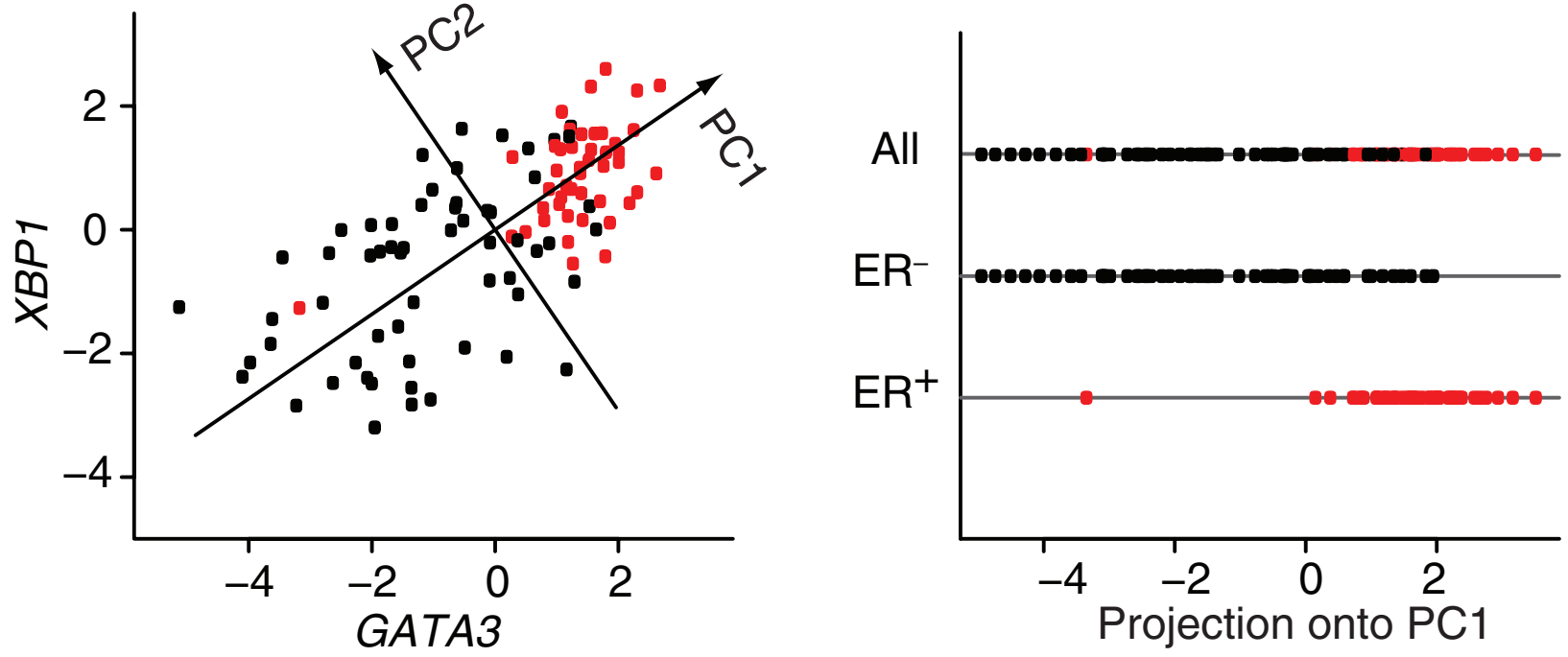
The axis that minimizes the projection error is the first Principal Component.

PCA conceptually: simplify data

Second Principal Component:
Minimizes the error that is left over from the first Principal Component (lengths of the maroon arrows).



Cancer and control samples segregate on PC1



From: Markus Ringner, Nature Biotechnology, vol. 26, p. 303-304, 2008

Variance

- Measure of the spread in a data set.

$$s^2 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{(n-1)}$$

- Can be calculated along a specific axis.
- The vector that maximizes the variance, minimizes the error.
- Thus finding PC is finding vectors that maximize variance.

Covariance

- How do two datasets (vectors X & Y) vary with respect to each other?
  - Sample covariance is

$$cov(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)}$$

  - Sample variance can be rewritten as cov(X,X)

$$var(X) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(X_i - \bar{X})}{(n-1)}$$

Covariance matrix

- The covariances between all pairs of features

$$C = \begin{pmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{pmatrix}$$

- Can use matrix operations to simplify the equation:

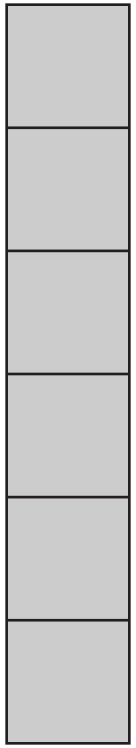
$$C = \left(\frac{1}{n-1}\right) X^T X$$

Matrix Algebra

Linear Algebra: Vectors

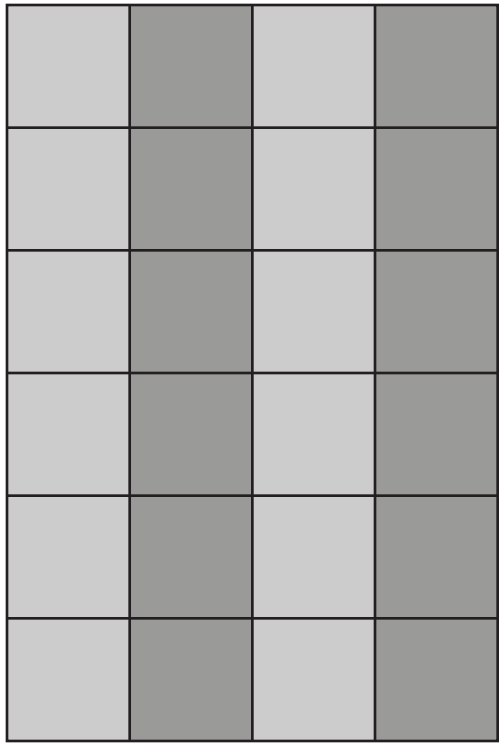
$$\mathbf{x} = (x_1, x_2, x_3, \ldots, x_n)$$

$$\mathbf{x}^T = (x_1, x_2, x_3, \ldots, x_n)^T$$



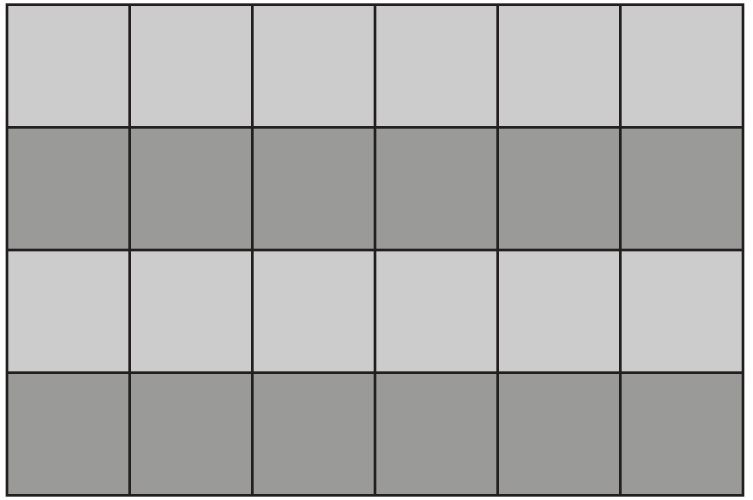
Linear Algebra: Matrices

A



(6, 4)

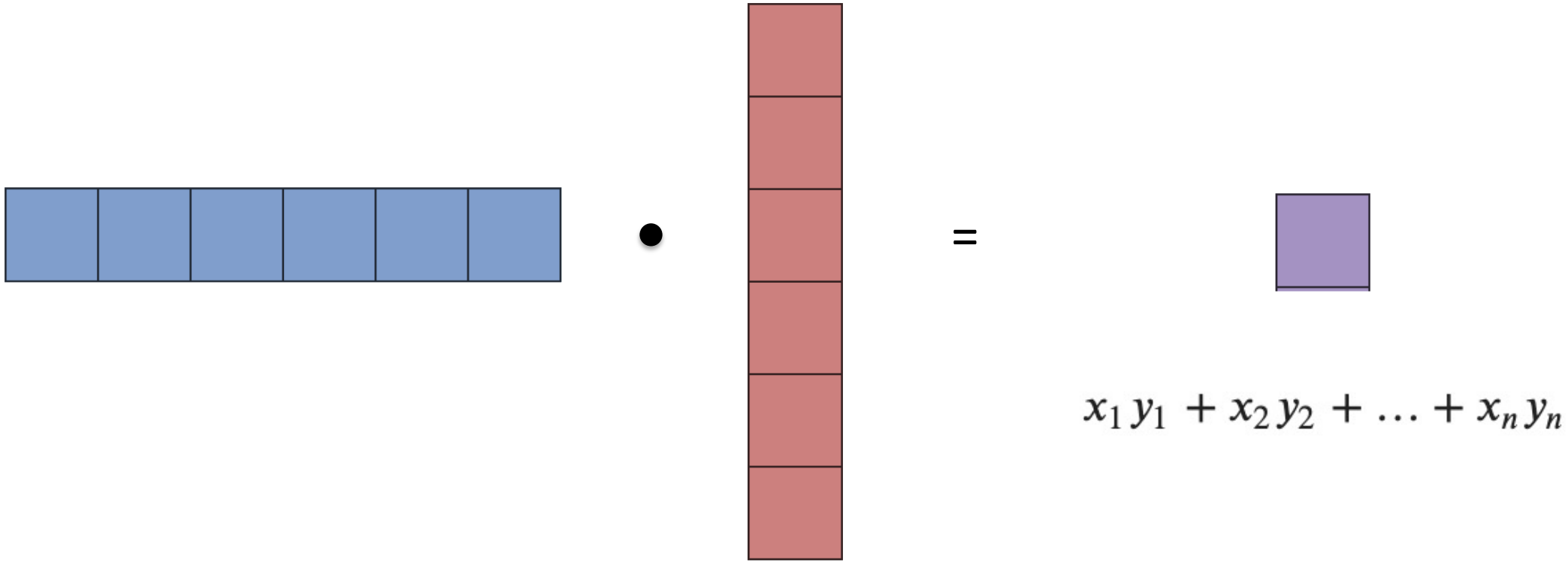
A^T



(4, 6)

Linear Algebra: Inner Product

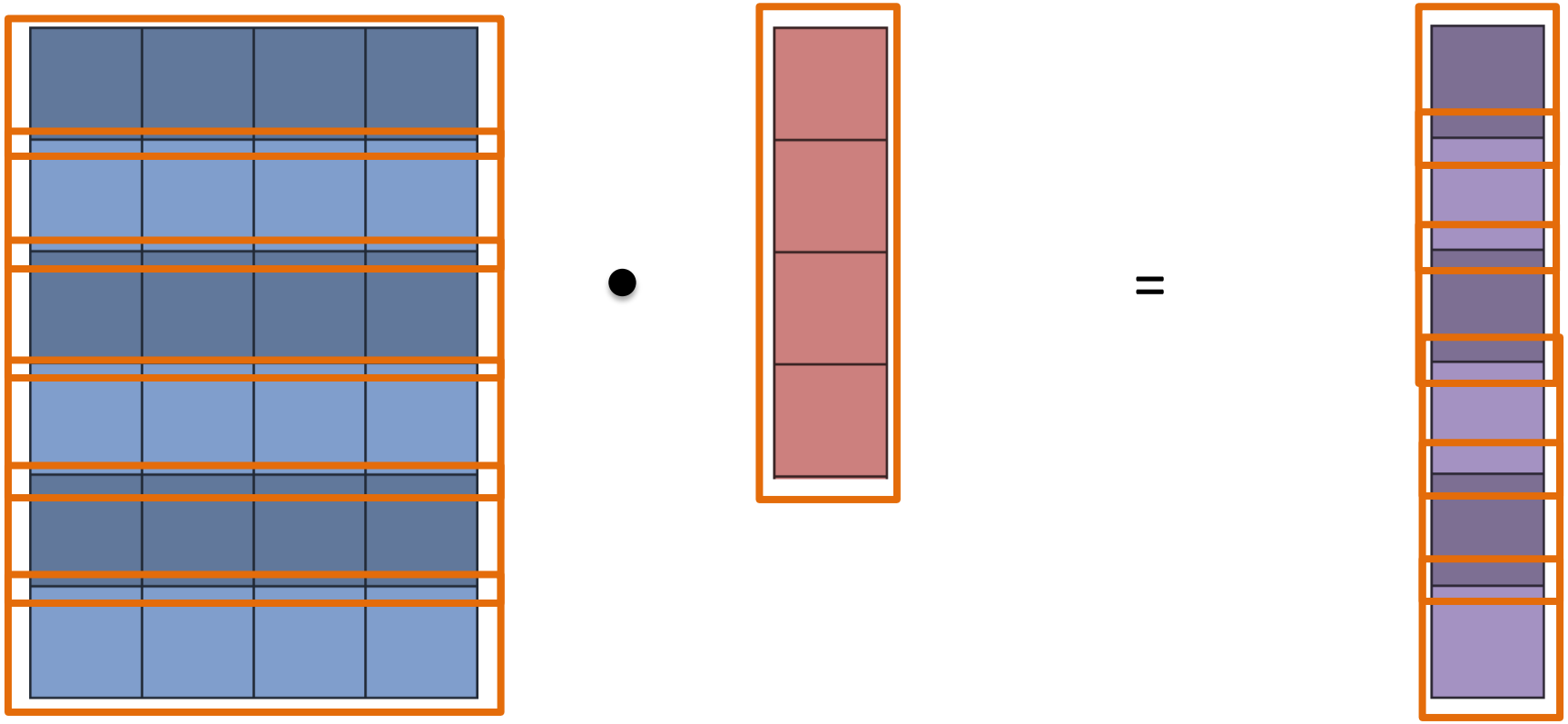
$$\mathbf{x}^T \cdot \mathbf{y}$$



$$x_1 y_1 + x_2 y_2 + \ldots + x_n y_n$$

Linear Algebra: Matrix-Vector Product

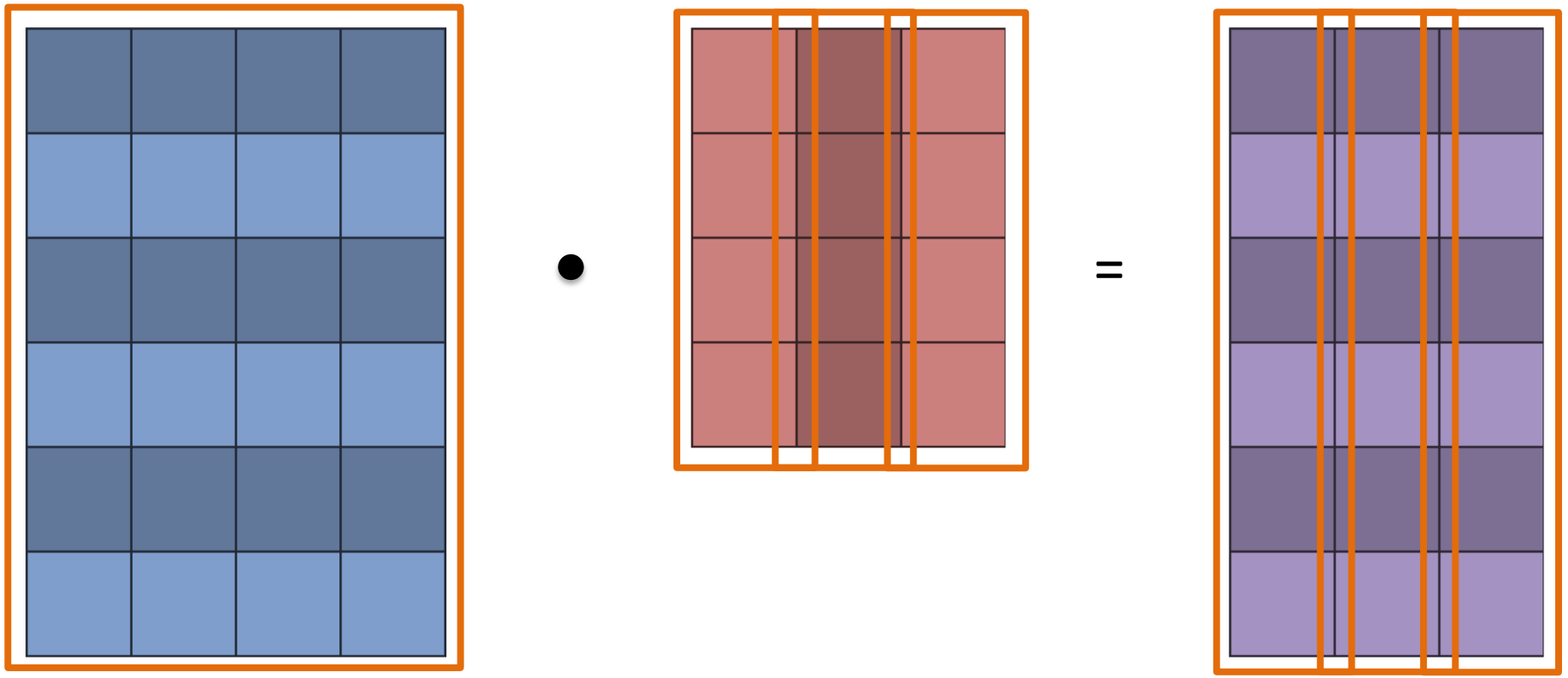
Ax



(n, p) (p, 1) (n, 1)

Linear Algebra: Matrix-Matrix Product

$$AB$$



(n, p) (p, q) (n, q)

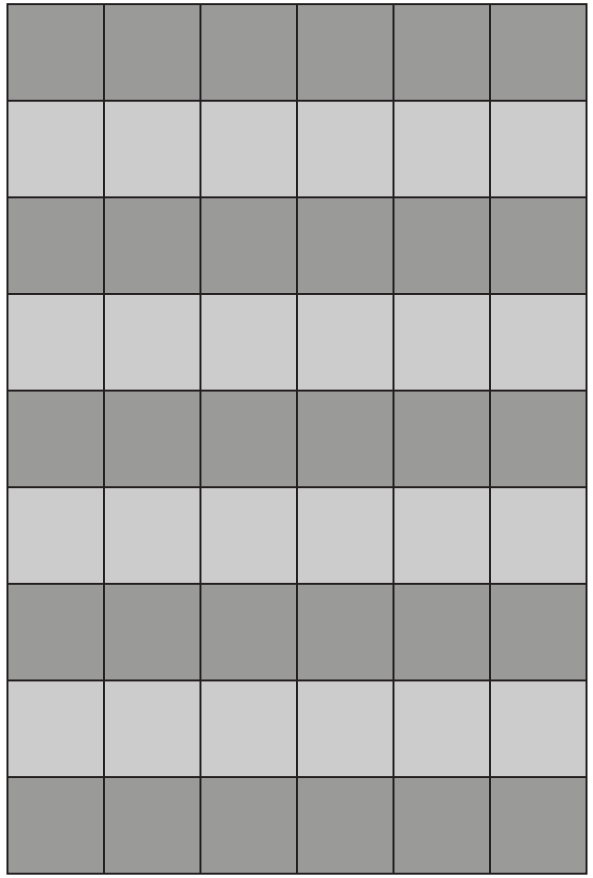
And we’re back!

PCA Step 1: Mean-centering

Data Matrix *A*

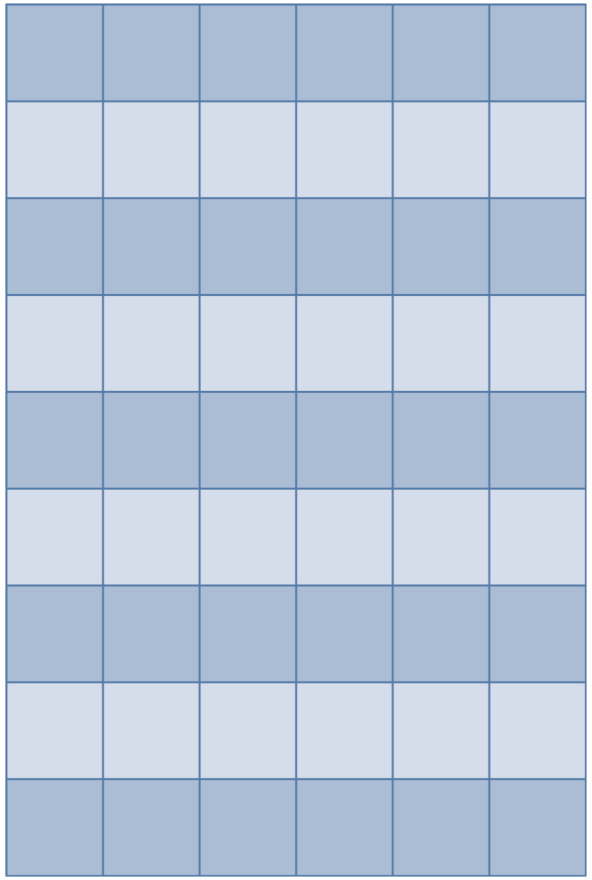
p features

n observations



Subtract column mean from each column

Mean-centered *X*



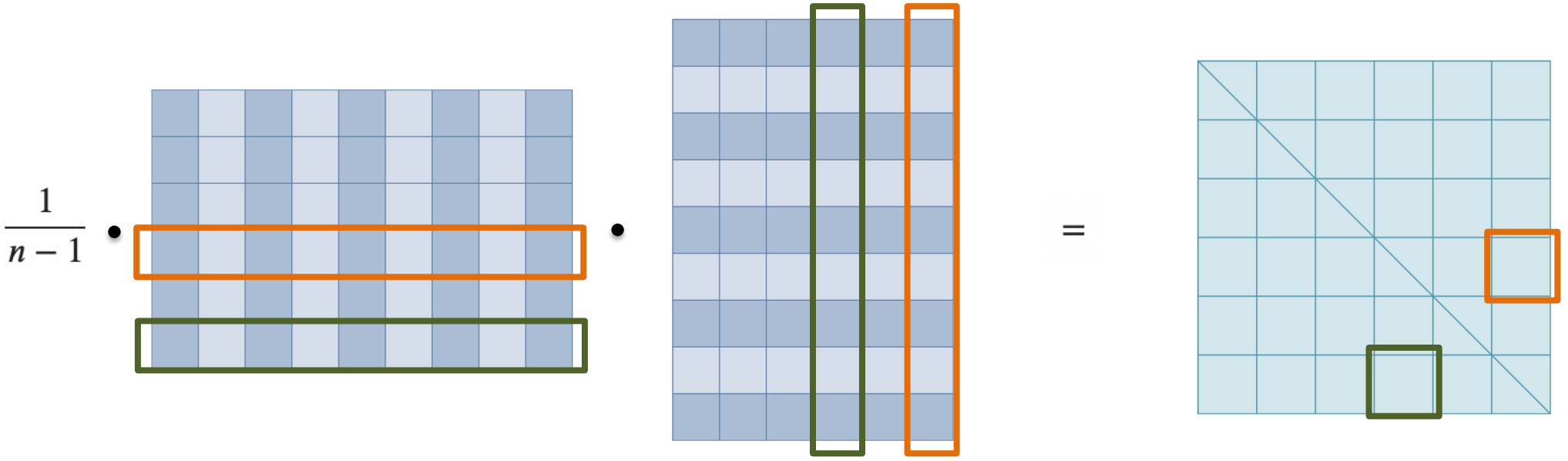
PCA Step 2: Calculate Feature Covariance Matrix

$$\frac{1}{n-1} X^T X = C$$

X^T (p, n)

X (n, p)

Feature Covariance Matrix ***C*** (p, p)



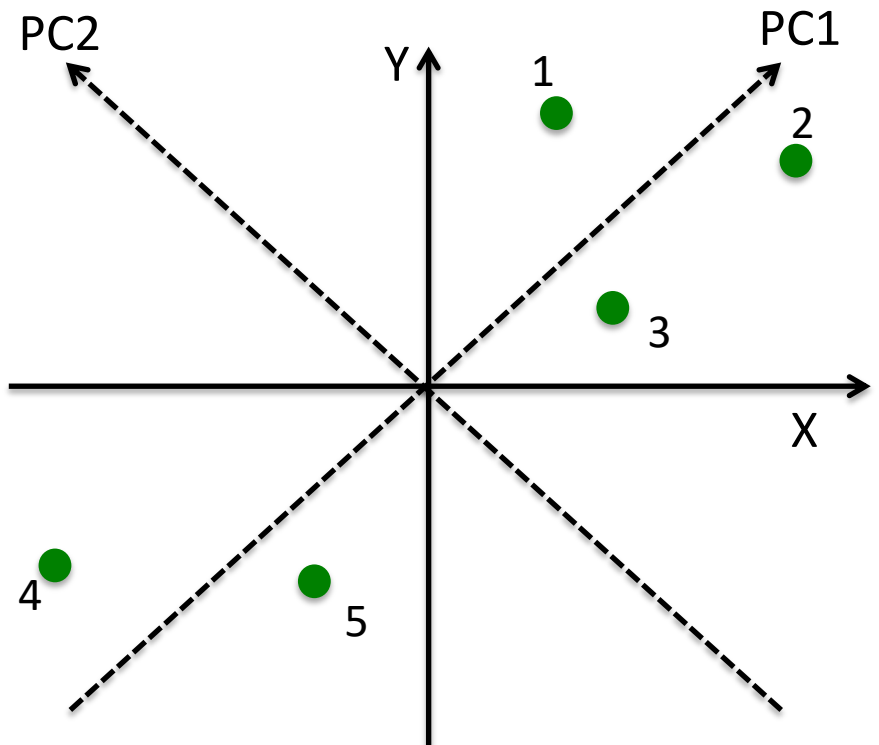
$$cov(\mathbf{a}, \mathbf{b}) = \frac{1}{n-1}(\mathbf{a} - \bar{a}) \cdot (\mathbf{b} - \bar{b})$$

Properties of the covariance matrix

- Diagonal elements contain the variance along each axis
- Total variance: Sum of the diagonal elements
- When the axes are rotated (for example going from original framework to PCs), the total variance stays the same!

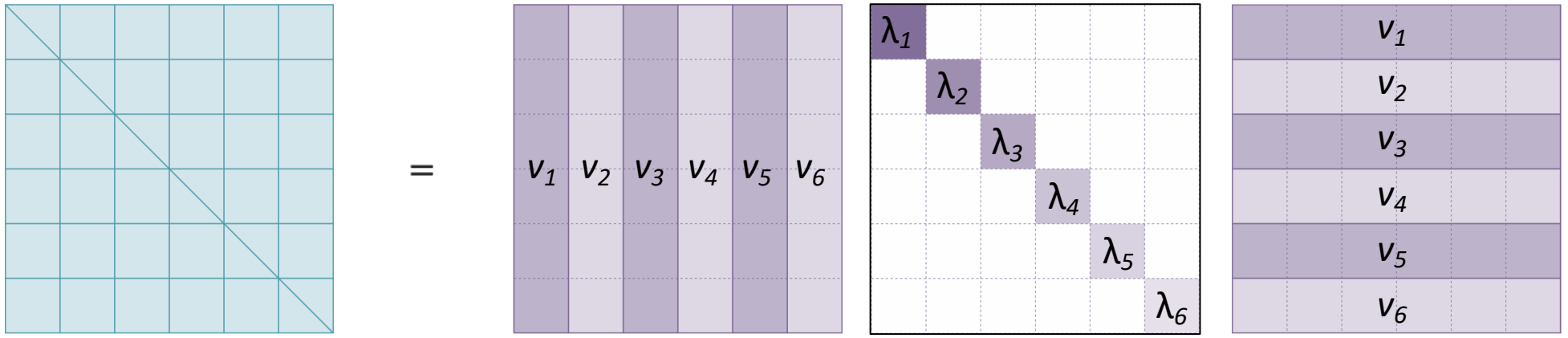
How are PCs related to covariance?

- Key Idea: When we rotate the axes in such way that the covariance between the axes disappears, we have obtained the principal components!



How do we rotate the axes?

- Every square symmetric matrix can be decomposed to give precisely the rotation into a new coordinate system that makes cross-covariances disappear!
- This is called **diagonalization** or **eigendecomposition**



C V Λ V^T

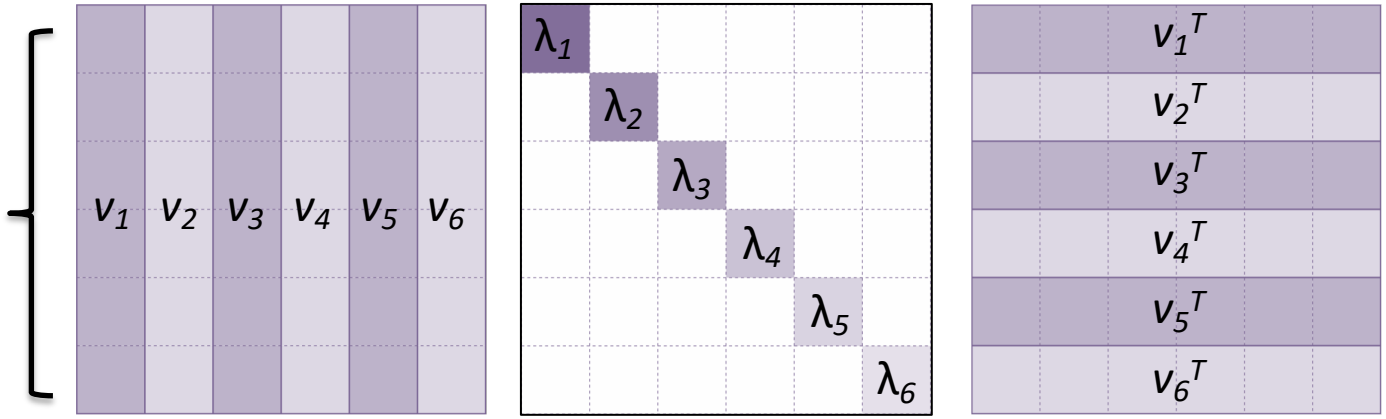
PCA Step 3: Diagonalize the Feature Covariance Matrix

V Λ V^T

Principal Components (directions) – a.k.a. eigenvectors of *C*

Variance Explained – a.k.a. eigenvalues of *C*

Factor "Loadings"



- unit vectors (length = 1)
- mutually orthogonal
- sign is arbitrary

$\sum_{i=1}^{p} \lambda_i$ = total variance

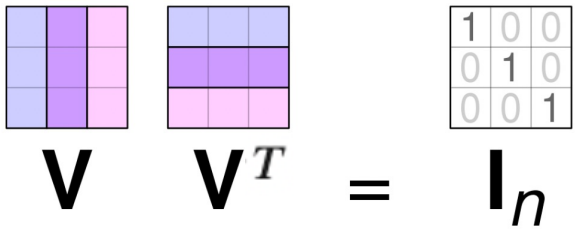
Decomposition is unique up to sign of the eigenvectors and ordering of the eigenvalues.

We choose the order $\lambda_1 > \lambda_2 > \ldots > \lambda_p$.

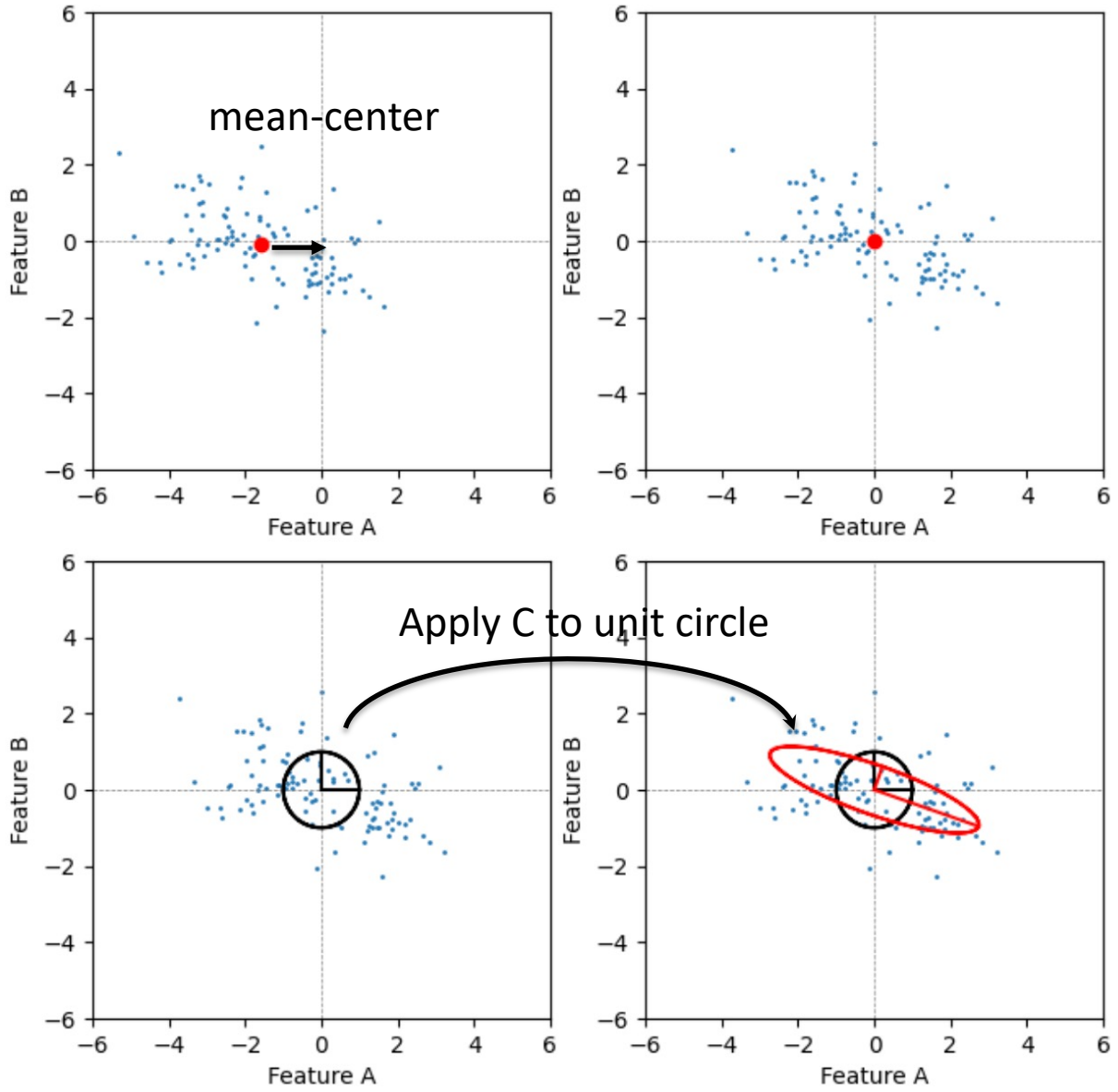
Properties of the PCs

- Norm is always 1
- Direction is not defined, i.e., you may flip the sign of the PC loading vector.
- PCs (columns of V) are always orthogonal to each other

Summarized as: $VV^T = V^TV = I$



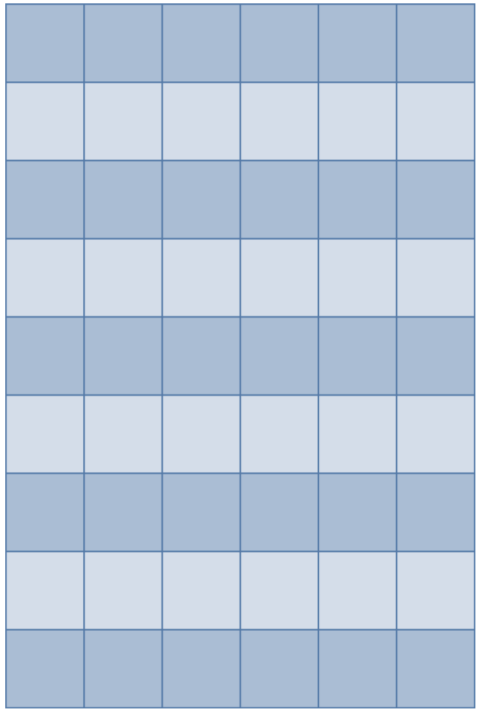
Covariance Matrix as a Linear Transformation of Feature Space



PCA Step 4: Project the data onto the PC axes

Centered Data Matrix

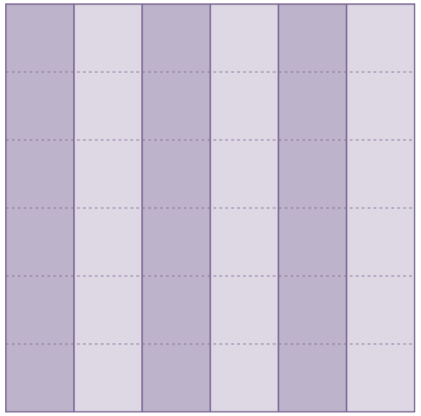
X



(n, p)

PCs (loadings)

V

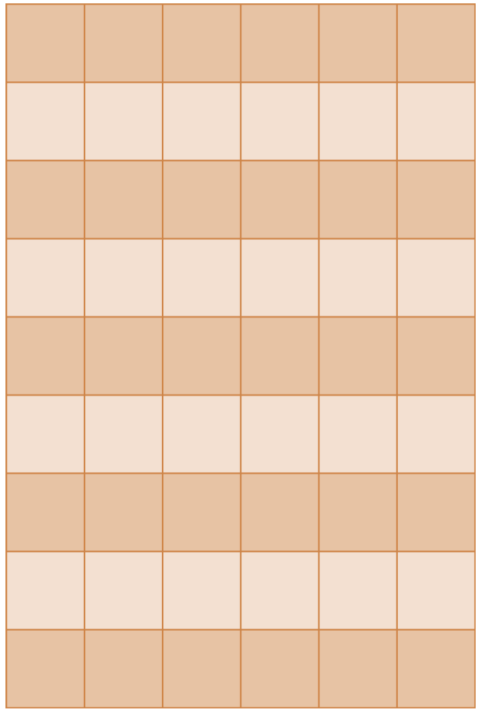


(p, p)

=

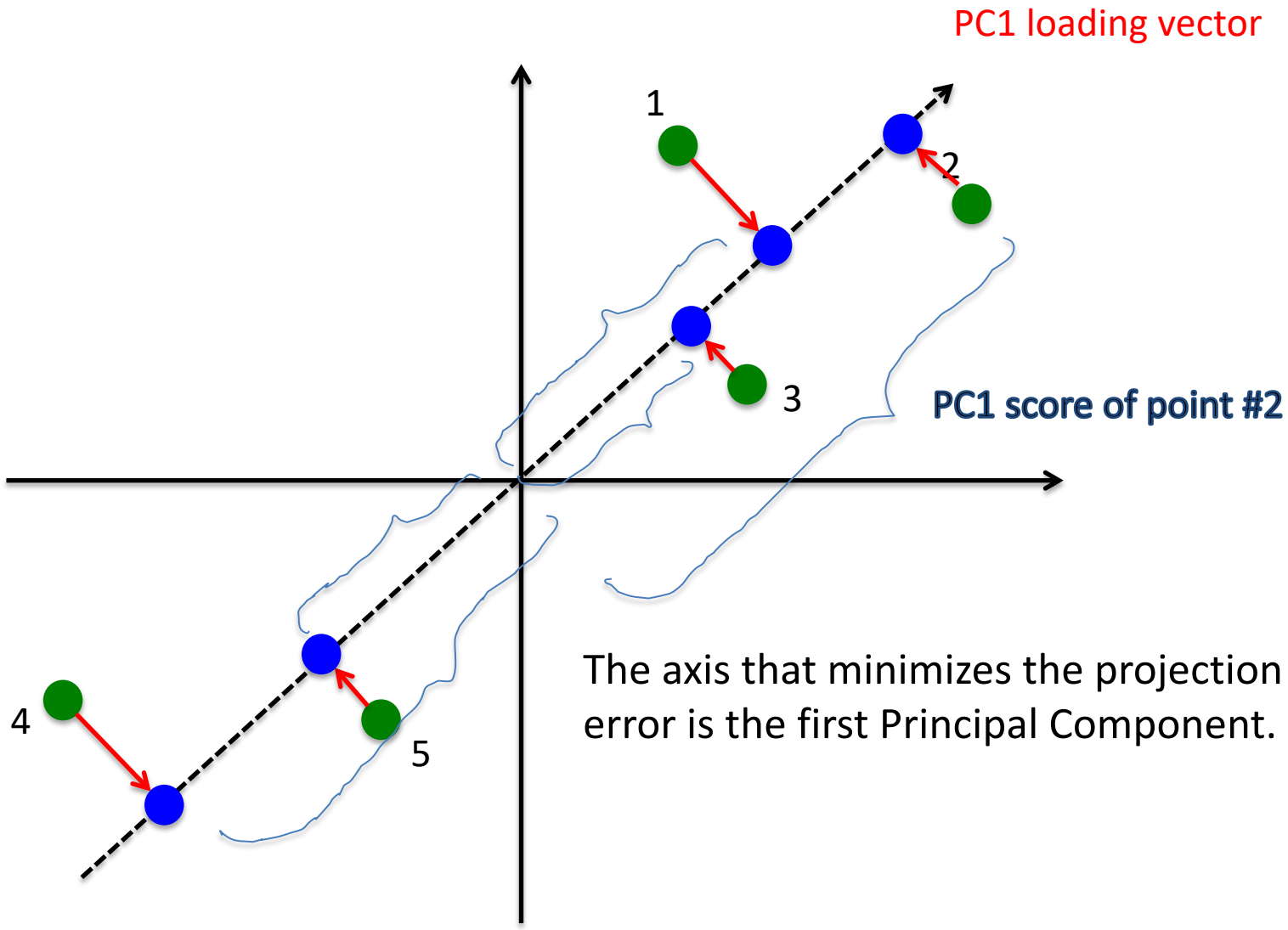
PC projections (scores)

$\tilde{X}$



(n, p)

Revisiting our previous example



The axis that minimizes the projection error is the first Principal Component.

Where's the reduction?

$$C = \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T + \lambda_2 \mathbf{v}_2 \mathbf{v}_2^T + \ldots + \lambda_p \mathbf{v}_p \mathbf{v}_p^T$$

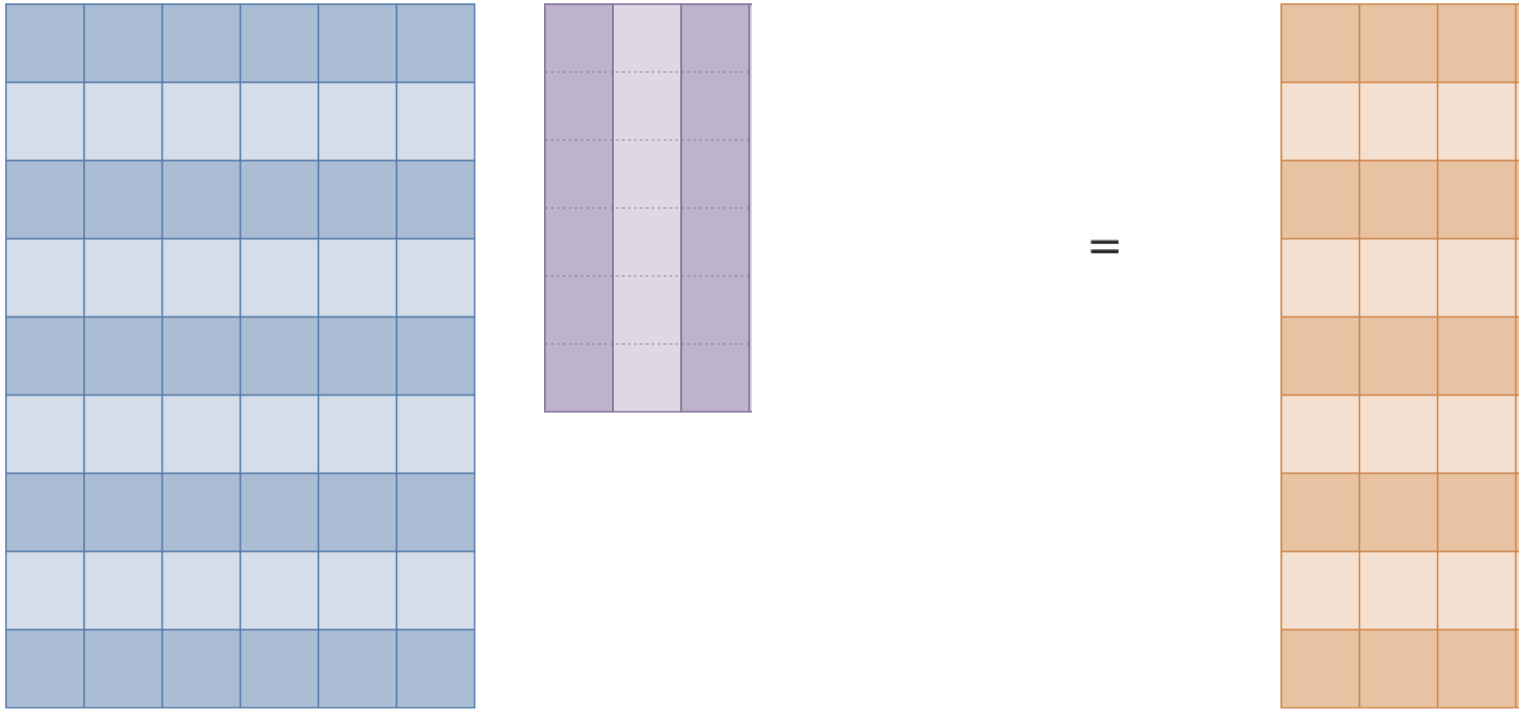
$$C \approx \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T + \lambda_2 \mathbf{v}_2 \mathbf{v}_2^T + \ldots + \lambda_r \mathbf{v}_r \mathbf{v}_r^T \qquad r < p$$

PCA Step 4: Project the data onto the PC axes

Centered Data Matrix X

PCs V

PC projections (scores) $\tilde{X}$



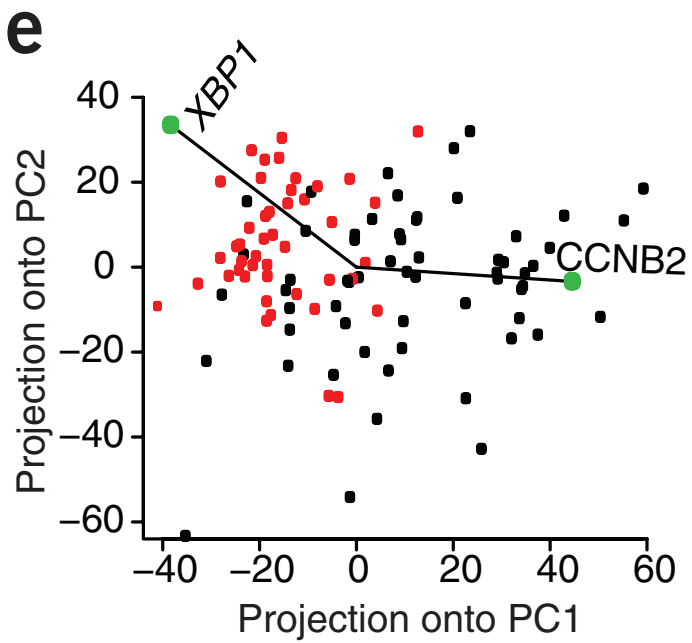
(n, p)

(p, r)

(n, r)

How to use PCs

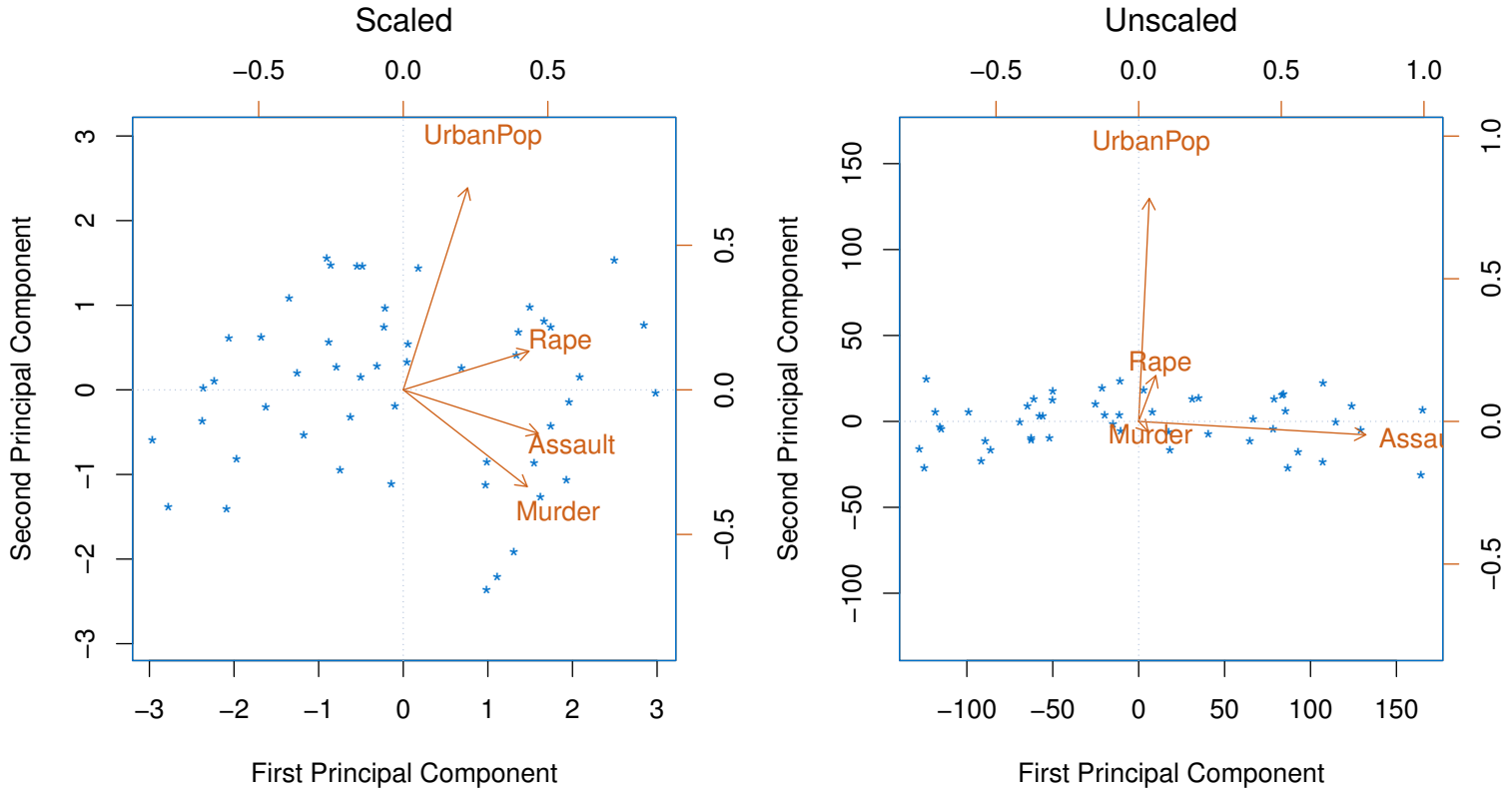
Visualization of complex datasets (with many features) in a lower dimension (biplot = score plot + loading plot):



From: Markus Ringner, Nature Biotechnology, vol. 26, p. 303-304, 2008

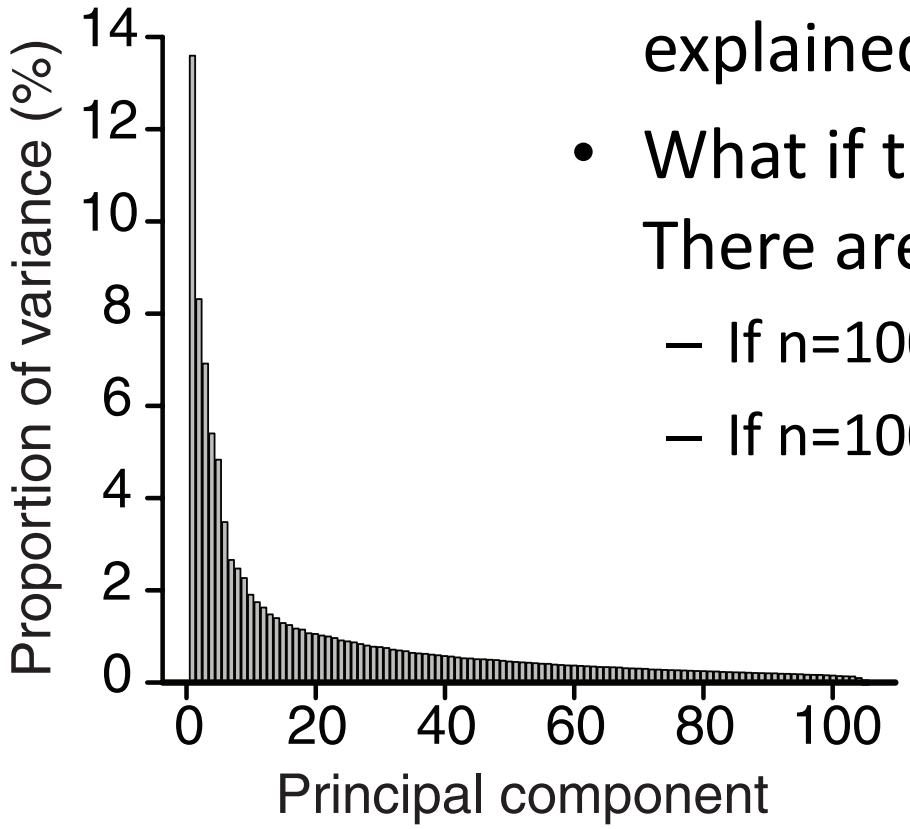
Practical considerations

Scaling the features affect the results of PCA.



"An Introduction to Statistical Learning, with applications in R" (Springer, 2013)

Practical considerations

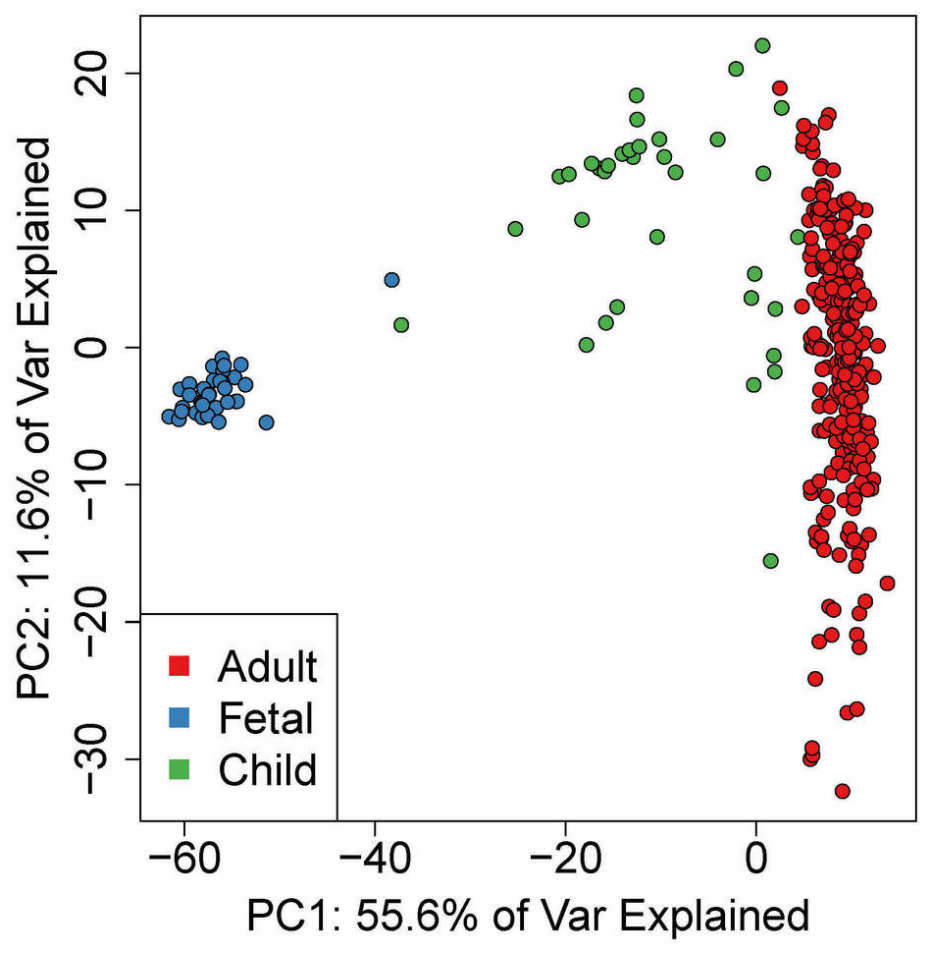


- The Scree Plot shows the proportion of variance explained by each PC.
- What if there are more features than samples (n<p)? There are at most min(n-1, p) PCs, for example:
  - If n=100 and p=2000, there are at most 99 PCs
  - If n=1000 and p=100, there are at most 100 PCs

Markus Ringner, Nature Biotechnology, vol. 26, p. 303-304, 2008

Mapping DNA methylation across development, genotype and schizophrenia in the human frontal cortex
Andrew E Jaffe, …, Joel E Kleinman
Nature Neuroscience 19, 40–47 (2016) doi:10.1038/nn.4181

Suppl. Fig. 1. Principal component analysis (PCA) demonstrates genome-wide changes in DNAm comparing pre- and post-natal samples



To keep in mind

- Covariance matrix is square, and its dimension depends on the number of features or axes (p), not the number of samples or data points (n).
- Principal Component vectors: Derived from the covariance matrix, therefore dimension depends on the number of features (again, not the data points).
- Diagonalization is computationally expensive (scales cubically in p), but for typical number of features this is not a problem.

PCA Resources

Review article

Principal component analysis: a review and recent developments

Ian T. Jolliffe and Jorge Cadima

Published: 13 April 2016 | https://doi.org/10.1098/rsta.2015.0202

https://stats.stackexchange.com/a/140579

Published: March 2008

What is principal component analysis?

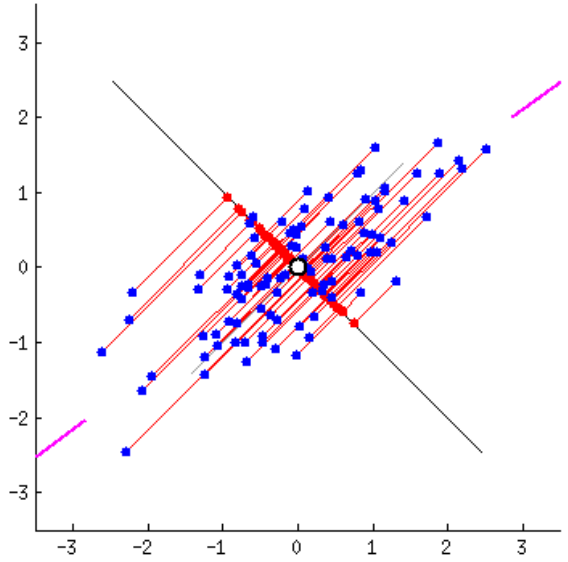
Markus Ringnér

Nature Biotechnology **26**, 303–304 (2008) | Cite this article

answered Mar 6, 2015 at 0:30

amoeba

96k 31 283 323



Published: 29 June 2017

Points of Significance

Principal component analysis

Jake Lever, Martin Krzywinski & Naomi Altman

Nature Methods **14**, 641–642 (2017) | Cite this article

Python Data Science Handbook (Jake Vanderplas)

Article | Open Access | Published: 29 August 2022

Principal Component Analyses (PCA)-based findings in population genetic studies are highly biased and must be reevaluated

Eran Elhaik

Scientific Reports **12**, Article number: 14683 (2022) | Cite this article

https://setosa.io/ev/principal-component-analysis/

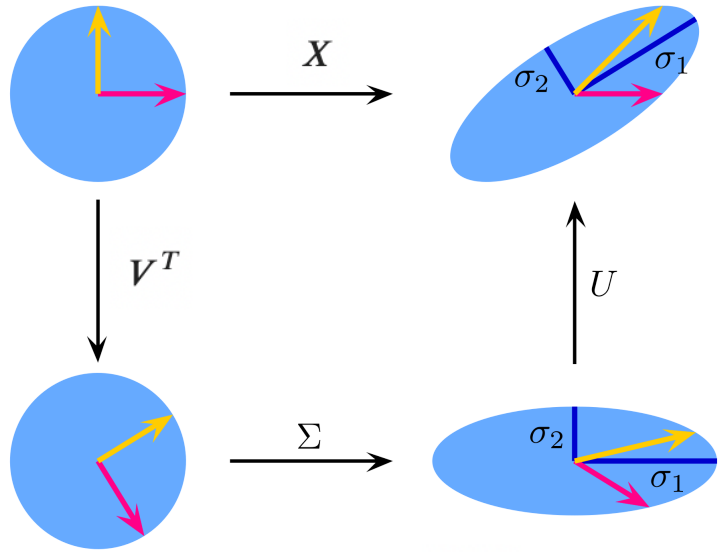
Technical Note

Most implementations don't actually compute the feature covariance matrix.

There is a more fundamental factorization called the **Singular Value Decomposition**.

PCA is equivalent to the SVD of a centered data matrix ***X***.

The columns of ***V*** (right singular vectors) are the eigenvectors of $\boldsymbol{X^TX}$
The columns of ***U*** (left singular vectors) are the eigenvectors of $\boldsymbol{XX^T}$
$\boldsymbol{X^TX}$ and $\boldsymbol{XX^T}$ always have the same eigenvalues!
$\boldsymbol{\Sigma}$ is diagonal and its elements are the square roots of the eigenvalues.



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

Therefore:
The explained variances are the squares of the singular values, i.e. $\boldsymbol{\Sigma^2}$.
The PCs (loadings) are right singular vectors of X, i.e. columns of ***V***.
The PC projections of the data are the left singular vectors of X scaled by the singular values, i.e. rows of $\boldsymbol{U\Sigma}$.

$$V^TV = I$$
$$U^TU = I$$

$$XV = U\Sigma$$