

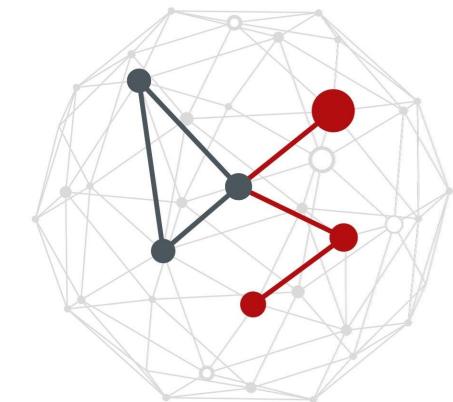
PRINCIPAL COMPONENT ANALYSIS (PCA)

Michele Rossi

michele.rossi@unipd.it

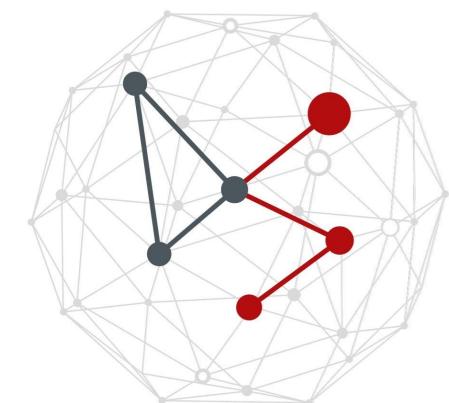
<http://www.dei.unipd.it/~rossi/>

University of Padova, IT



PCA - PART I

Intuition and Theory



What is it?

- A dimensionality reduction technique
 - Based on rigorous matrix algebra (matrix decomposition)
 - Deterministic result, given a dataset
- Applications
 - Lossy data compression
 - Automatic feature extraction
 - Clustering
 - Data visualization (exploratory data analysis)
- Used in many and diverse fields to
 - Extract relevant information in big and confusing data sets
 - Simple, non-parametric (**unsupervised**) method

Toy example (1/3)

- We are a physicist who is about to study the motion of
 - an *ideal spring*
 - i.e., a body of mass m attached to a **frictionless spring**
 - the spring is stretched, moving it away from its equilibrium point
 - It oscillates along the **x-axis** (forever) at a set frequency
- A single variable (x) would be needed to
 - Fully characterize the **law of motion**
- But we are ignorant...
 - we then resort to measuring the motion from three cameras
 - cameras are placed at **arbitrary angles** wrt the spring system
 - each camera measures a **projection** of the real motion
 - this is what we often do:
 - oversampling the signal, using a sub-optimal reference system

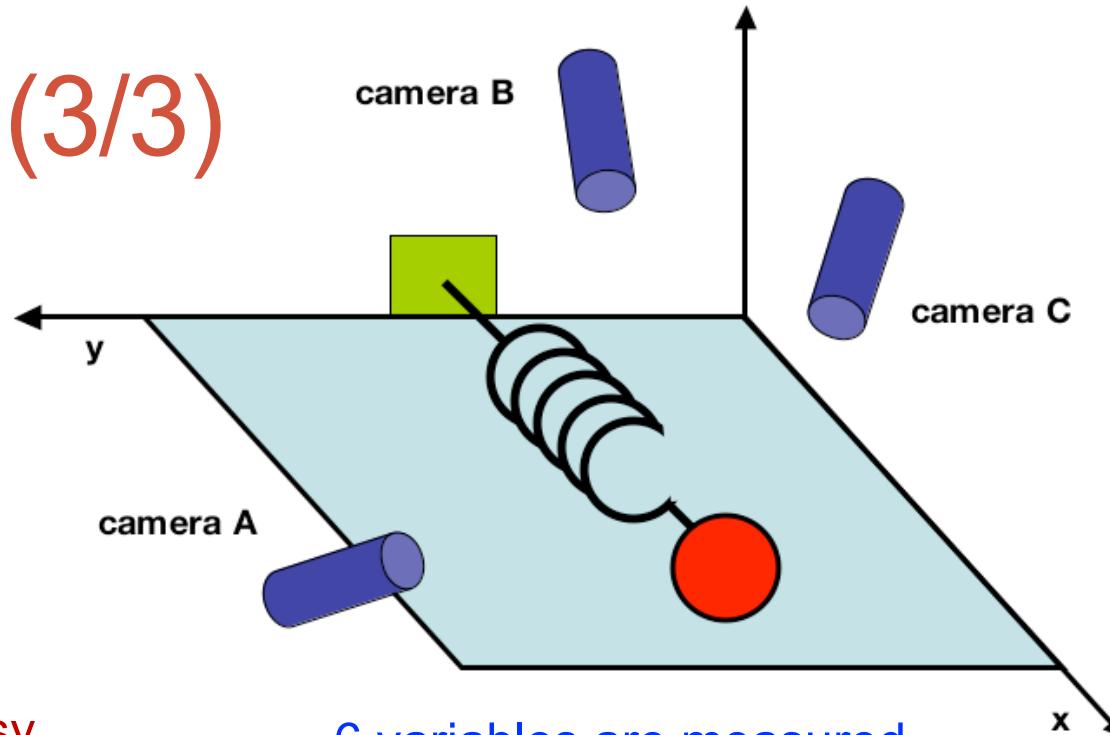
Toy example (2/3)

“We know a-priori that *if we were smart experimenters*, we would have just measured the position along the x-axis with one camera. But this is not what happens in the real world. We often do not know which measurements best reflect the dynamics of our system...

... furthermore, we sometimes record more dimensions than we actually need...” - text from [1]

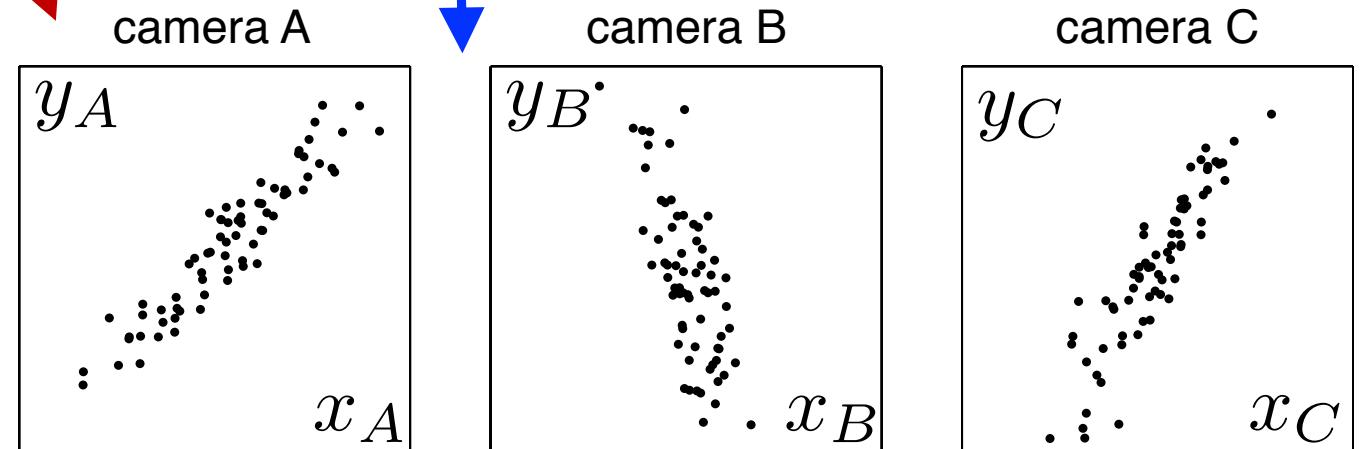
[1] Jonathon Shlens, “A tutorial on Principal Component Analysis,” arXiv:1404:1100v1 , April 7, 2014. [4577 citations]

Toy example (3/3)



measurements are noisy

6 variables are measured
(2 projections per camera)



The framework – “change of basis”

- Goal of PCA
 - Find a new basis to re-express a dataset
- Our hope
 - This new basis should
 - 1) filter out noise,
 - 2) reveal interesting structure
- In the spring example, the hope is
 - to determine that the unit length vector along the **x-axis**
 - is the (only) important dimension
 - this allows to discern between *informative data* and *noise*
 - allows using a single variable (magnitude along the x-axis)

Our measurements

- n measurements (time samples)
- each sample (at time i) is a *column vector*
 - Collecting all m measurement types at time i ($m=6$ in the toy example)

$$\mathbf{x}_i = (x_{1i}, x_{2i}, \dots, x_{mi})^T \in \mathbb{R}^m$$

- This data can be summarized through a matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$
- Each column represents a single (data) sample

data matrix $\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_n \end{bmatrix}$

- We assume (for now) that the data is zero mean
- If not, we subtract the mean, computed as: $\bar{\mathbf{x}} = \left(\sum_{i=1}^n \mathbf{x}_i \right) / n$

The naïve basis (natural basis for the Euclidean space)

- Is the basis we use to measure the original data points

$$B = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_m \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} = I_{m \times m}$$

- **Trivially:** each vector in our dataset can be expressed as a *linear combination* of the basis vectors (the combination coefficient being the data points themselves)

$$\mathbf{x}_i = B\mathbf{x}_i$$

Change of basis (1/3)

- **Question**
 - *Is there another basis, which is a **linear** combination of the original basis, that **better expresses** our dataset?*
- **Linearity**
 - PCA makes a *stringent* but (very) *powerful* assumption: **linearity**
 - This *greatly* simplifies the problem
- **Problem setup**
 - Original data points $\mathbf{X} \in \mathbb{R}^{m \times n}$
 - New basis $\mathbf{P} \in \mathbb{R}^{m \times m}$
 - Transformed data points $\mathbf{Y} \in \mathbb{R}^{m \times n}$

Change of basis (2/3)

- \mathbf{p}_i are the **rows** of $\mathbf{P} \in \mathbb{R}^{m \times m}$
- \mathbf{x}_i are the **columns** of $\mathbf{X} \in \mathbb{R}^{m \times n}$
- \mathbf{y}_i are the **columns** of $\mathbf{Y} \in \mathbb{R}^{m \times n}$

$$\mathbf{P}\mathbf{X} = \mathbf{Y}$$

- **Observations**
 - \mathbf{P} is a matrix that transforms \mathbf{X} into \mathbf{Y}
 - The **rows of \mathbf{P} are new basis vectors** to express the columns of \mathbf{X}

$$\mathbf{P}\mathbf{X} = \begin{bmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_m \end{bmatrix} \left[\begin{array}{c} \mathbf{x}_1 \cdots \mathbf{x}_n \\ \text{input data} \\ (\text{columns}) \end{array} \right] = \begin{bmatrix} \mathbf{p}_1\mathbf{x}_1 & \cdots & \mathbf{p}_1\mathbf{x}_n \\ \vdots & \ddots & \vdots \\ \mathbf{p}_m\mathbf{x}_1 & \cdots & \mathbf{p}_m\mathbf{x}_n \end{bmatrix} \triangleq \left[\begin{array}{c} \mathbf{y}_1 \cdots \mathbf{y}_n \\ \text{transformed data} \\ (\text{columns}) \end{array} \right]$$

Change of basis (3/3) $PX = Y$

$$PX = \begin{bmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_m \end{bmatrix} \left[\mathbf{x}_1 \cdots \mathbf{x}_n \right] = \begin{bmatrix} \mathbf{p}_1 \mathbf{x}_1 & \cdots & \mathbf{p}_1 \mathbf{x}_n \\ \vdots & \ddots & \vdots \\ \mathbf{p}_m \mathbf{x}_1 & \cdots & \mathbf{p}_m \mathbf{x}_n \end{bmatrix} \triangleq \left[\mathbf{y}_1 \cdots \mathbf{y}_n \right]$$

- \mathbf{y}_i , $i = 1, \dots, n$
 - is a projection onto the basis $\{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_m\}$
- The row vectors $\{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_m\}$ in the new basis are referred to as the **principal directions** of X

Open questions

- What is the best way to re-express \mathbf{X} ?
- What is a good choice of basis \mathbf{P} ?
- Answering these questions
 - Implies understanding what features we would like \mathbf{Y} to exhibit
 - This also implies adding additional assumptions beyond linearity
- Additional assumptions, have to do with
 - noise
 - redundancy

Noise (1/2)

- Measurement noise in any data set must be low
 - As otherwise no meaningful info on the data can be extracted
 - No matter which technique we use
- There exists no absolute scale for noise
 - We rather compare its power against that of the useful component
 - To do this we define the **Signal to Noise Ratio (SNR)** as the ratio of their variances, i.e.,

$$SNR = \frac{\sigma_{\text{signal}}^2}{\sigma_{\text{noise}}^2}$$

($SNR >> 1$ indicates a *high precision measure*)

Noise (2/2) – back to our toy example

- Single camera (A) measures a noisy trajectory
 - Still along a straight line (projected onto camera view)
 - Any spread deviating from straight-line is noise
- find the direction aligned with σ_{signal}^2

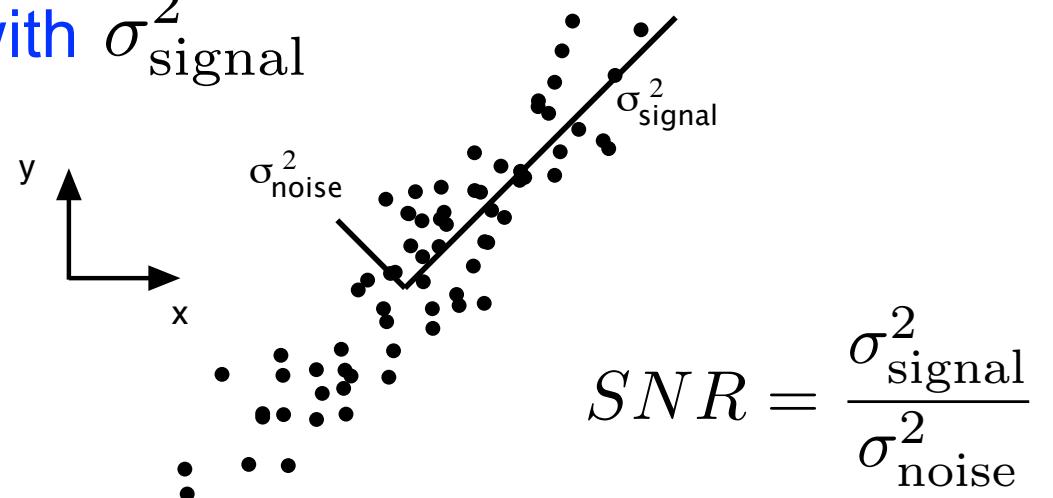


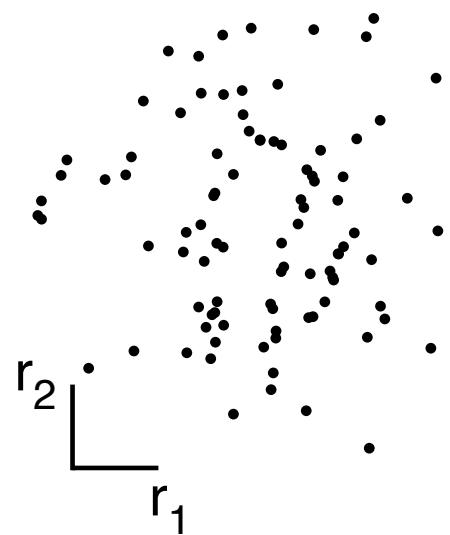
FIG. 2 Simulated data of (x, y) for camera A. The signal and noise variances σ_{signal}^2 and σ_{noise}^2 are graphically represented by the two lines subtending the cloud of data. Note that the largest direction of variance does not lie along the basis of the recording (x_A, y_A) but rather along the best-fit line.

Assumption

- Based on previous example (intuition)
- We assume that
 - The dynamics of interests exist along directions with the highest variance (and presumably the highest SNR too!!!)
- This assumption suggests that the direction that maximizes the variance (aligned with the signal component in the previous figure) corresponds to the best-fit for the data cloud
- How do we generalize this for any dimension? (be patient...)

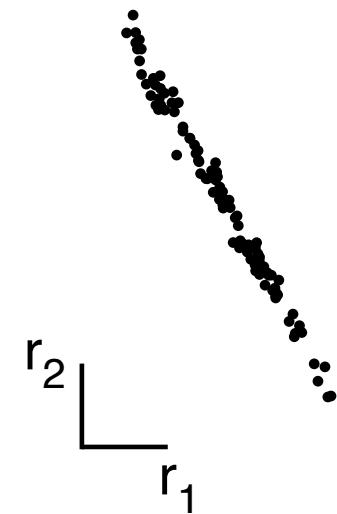
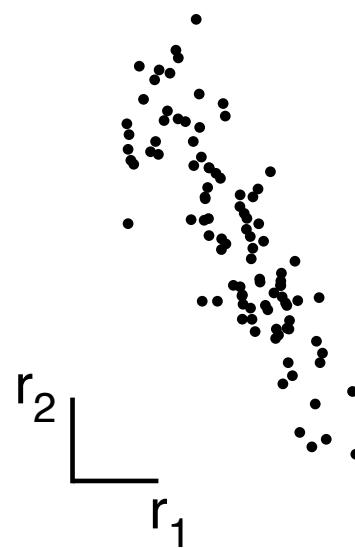
Redundancy

- Another confounding factor in the data is **redundancy**
- Means that some of the variables are **highly correlated**



low redundancy

(low correlation between r_1 and r_2)

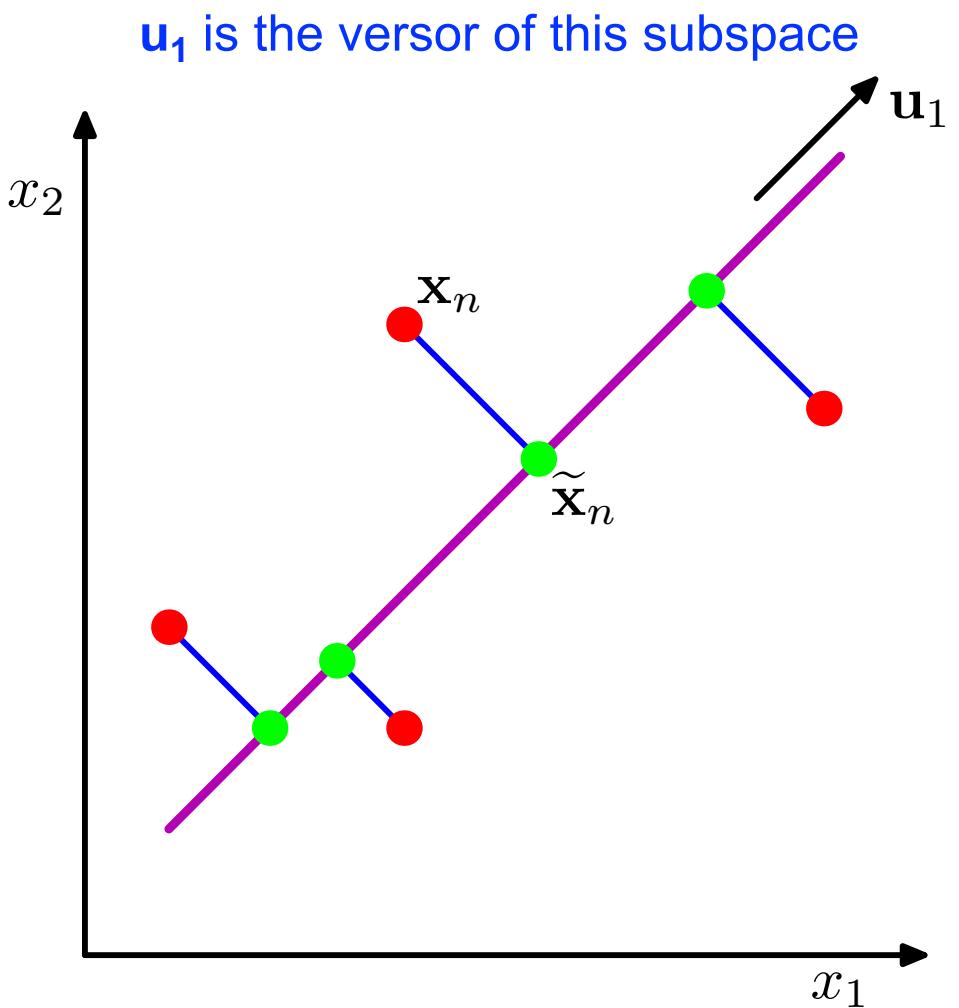


high redundancy

- high correlation between r_1 and r_2
- $r_1(r_2)$ can be used to predict $r_2(r_1)$
- central idea in dimensionality reduction

Let's get started...

PCA seeks a space of lower dimensionality, known as the **principal subspace** and denoted by the magenta line, such that the **orthogonal projection** of the data points (red dots) onto this subspace maximizes the variance of the projected points (green dots)

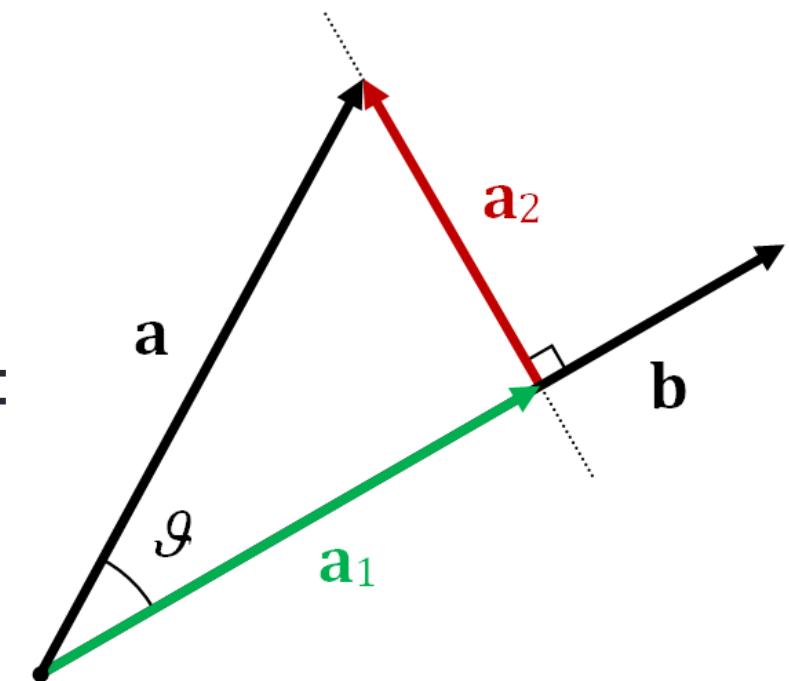


Projection into vector

- Projection of vector \mathbf{a} into vector \mathbf{b}
- It is a vector parallel to \mathbf{b} defined as:

$$\mathbf{a}_1 = a_1 \frac{\mathbf{b}}{|\mathbf{b}|} \triangleq a_1 \mathbf{b}_v$$

versor along direction \mathbf{b}
(it is unit length)



- The scalar projection a_1 along \mathbf{b} is found as:

$$\begin{aligned} a_1 &= |\mathbf{a}| \cos(\theta) = |\mathbf{a}| |\mathbf{b}_v| \cos(\theta) = \langle \mathbf{a}, \mathbf{b}_v \rangle = \\ &= \mathbf{a} \cdot \mathbf{b}_v = \mathbf{a}^T \mathbf{b}_v = \end{aligned}$$

inner (dot) product

$$= \mathbf{b}_v^T \mathbf{a} = \langle \mathbf{b}_v, \mathbf{a} \rangle \quad \leftarrow \text{commutative property of dot product}$$

Finding \mathbf{u}_1 (1/5)

- \mathbf{u}_1 is a *unit* vector (versor), that is: $\mathbf{u}_1^T \mathbf{u}_1 = 1$

- Sample mean vector of the input data is:

$$\bar{\mathbf{x}} = \left(\sum_{i=1}^n \mathbf{x}_i \right) / n$$

- Mean of the projected data (along direction \mathbf{u}_1) is: $\mathbf{u}_1^T \bar{\mathbf{x}}$

Finding \mathbf{u}_1 (2/5)

- Variance of the projected data (“projected variance”) is:

$$\frac{1}{n} \sum_{i=1}^n (\mathbf{u}_1^T \mathbf{x}_i - \mathbf{u}_1^T \bar{\mathbf{x}})^2 = \mathbf{u}_1^T C_X \mathbf{u}_1$$

↑ ↑
projection of data points projection of mean

- Where the data **covariance matrix** is defined as:

$$C_X = \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T$$

Finding \mathbf{u}_1 (2/5 bis)

- Variance of data projected onto direction \mathbf{u}_1


$$\begin{aligned}\sigma^2(\mathbf{u}_1) &= \frac{1}{n} \sum_{i=1}^n (\mathbf{u}_1^T \mathbf{x}_i - \mathbf{u}_1^T \bar{\mathbf{x}})^2 = \frac{1}{n} \sum_{i=1}^n (\mathbf{u}_1^T (\mathbf{x}_i - \bar{\mathbf{x}})) (\mathbf{u}_1^T (\mathbf{x}_i - \bar{\mathbf{x}}))^T = \\ &= \frac{1}{n} \sum_{i=1}^n \mathbf{u}_1^T (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \mathbf{u}_1 = \mathbf{u}_1^T \left[\frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \right] \mathbf{u}_1 = \\ &= \mathbf{u}_1^T C_{\mathbf{X}} \mathbf{u}_1\end{aligned}$$

- where:

$$C_{\mathbf{X}} = \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \quad \text{covariance matrix}$$

Finding \mathbf{u}_1 (3/5)

- The objective is to **maximize the projected variance**
 - With respect to \mathbf{u}_1
 - Subject to the normalization condition $\mathbf{u}_1^T \mathbf{u}_1 = 1$

$$\max_{\mathbf{u}_1} [\mathbf{u}_1^T C_X \mathbf{u}_1]$$

$$\text{subject to: } \mathbf{u}_1^T \mathbf{u}_1 = 1$$

optimization
problem

- We construct a **Lagrangian function $J(\mathbf{u}_1)$**
(Lagrangian multiplier λ_1)

$$J(\mathbf{u}_1) = \mathbf{u}_1^T C_X \mathbf{u}_1 + \lambda_1 (1 - \mathbf{u}_1^T \mathbf{u}_1)$$



objective



constraint

Intermission – gradient of a quadratic form

- Let α be the quadratic form (scalar): $\alpha \triangleq \mathbf{x}^T \mathbf{A} \mathbf{x}$
 - where: \mathbf{x} is $n \times 1$, $\mathbf{A} = [a_{ij}]$ is $n \times n$ and does not depend on \mathbf{x}
- We define the gradient of α as the row vector:

$$\Delta\alpha(\mathbf{x}) \triangleq \left(\frac{\partial\alpha}{\partial x_1}, \frac{\partial\alpha}{\partial x_2}, \dots, \frac{\partial\alpha}{\partial x_n} \right)$$

- Then, we have that (row vector form):

$$\Delta\alpha(\mathbf{x}) = \mathbf{x}^T (\mathbf{A} + \mathbf{A}^T)$$

Intermission – gradient of a quadratic form

- Proof. by definition,

$$\alpha = \sum_{j=1}^n \sum_{i=1}^n a_{ij} x_i x_j$$

- Differentiating with respect to the **k-th** element of **x** we get:

$$\frac{\partial \alpha}{\partial x_k} = \sum_{j=1}^n a_{kj} x_j + \sum_{i=1}^n a_{ik} x_i$$

- for all **k=1, 2, ..., n**. This can be compactly written as:

$$\Delta \alpha(\mathbf{x}) = \mathbf{x}^T (\mathbf{A} + \mathbf{A}^T)$$

Intermission – gradient of a quadratic form

- Moreover, if matrix \mathbf{A} is symmetric ($\mathbf{A}=\mathbf{A}^T$), we have:

$$\begin{aligned}\Delta\alpha(\mathbf{x}) &= \mathbf{x}^T (\mathbf{A} + \mathbf{A}^T) = \\ &= \mathbf{x}^T (2\mathbf{A}) = 2\mathbf{x}^T \mathbf{A}\end{aligned}$$

- In column form, we get:

$$(\Delta\alpha(\mathbf{x}))^T = (2\mathbf{x}^T \mathbf{A})^T = 2\mathbf{A}^T \mathbf{x} = 2\mathbf{A}\mathbf{x}$$

- Note that the covariance matrix \mathbf{C}_x is symmetric
(this will be shown later)

Intermission – gradient of square norm

- Square norm-2:

$$\alpha = \mathbf{x}^T \mathbf{x} = \sum_{i=1}^n x_i^2$$

- Which leads to:

$$\frac{\partial \alpha}{\partial x_j} = 2x_j$$

- In (column) vector form: $(\Delta \alpha(\mathbf{x}))^T = 2\mathbf{x}$

- Example (using *chain rule of derivatives*):

$$g(\mathbf{x}) = -\cos(2\pi \mathbf{x}^T \mathbf{x}) + 2\mathbf{x}^T \mathbf{x}$$

$$\nabla g(\mathbf{x}) = 4\pi \sin(2\pi \mathbf{x}^T \mathbf{x})\mathbf{x} + 4\mathbf{x}$$

Finding \mathbf{u}_1 (4/5)

- Lagrangian function J

$$J(\mathbf{u}_1) = \mathbf{u}_1^T \mathbf{C}_X \mathbf{u}_1 + \lambda_1 (1 - \mathbf{u}_1^T \mathbf{u}_1)$$

- For a **stationary point**, we require that

$$(1) \quad \frac{\partial J(\mathbf{u}_1)}{\partial \mathbf{u}_1} = 0 \Rightarrow \mathbf{C}_X \mathbf{u}_1 - \lambda_1 \mathbf{u}_1 = 0$$

$$(2) \quad \frac{\partial J(\mathbf{u}_1)}{\partial \lambda_1} = 0 \Rightarrow 1 - \mathbf{u}_1^T \mathbf{u}_1 = 0 \Rightarrow \mathbf{u}_1^T \mathbf{u}_1 = 1$$

- If we left-multiply (1) by \mathbf{u}_1^T and make use of (2):

$$\mathbf{u}_1^T \mathbf{C}_X \mathbf{u}_1 = \lambda_1$$

Finding \mathbf{u}_1 (5/5)

- Our findings

$$\mathbf{C}_X \mathbf{u}_1 = \lambda_1 \mathbf{u}_1$$

- This says that \mathbf{u}_1 must be an *eigenvector* of matrix \mathbf{C}_X

$$\mathbf{u}_1^T \mathbf{C}_X \mathbf{u}_1 = \lambda_1 \leftarrow$$

- This says that the projected variance is maximized when \mathbf{u}_1 corresponds to eigenvector having the largest eigenvalue λ_1
- Also, the largest eigenvalue λ_1 equals the variance of the projected points (projected variance) – physical meaning!

Let's see this using linear algebra

- This procedure can be iterated for the second, third, fourth, etc. dimensions...
- A more computationally efficient algorithm is provided in the next slides...

Covariance matrix (1/4)

- With 2 variables (e.g., cameras) it is easy to identify the direction of best-fit → linear fitting
- In a more general setting this is not so obvious...
- Consider two sets of measurements with **zero mean**

$$A = \{a_1, a_2, \dots, a_n\} \quad B = \{b_1, b_2, \dots, b_n\}$$

- **Variances**

$$\sigma_A^2 = \frac{1}{n} \sum_{i=1}^n a_i^2 \quad \sigma_B^2 = \frac{1}{n} \sum_{i=1}^n b_i^2$$

- **Covariance of A and B**

$$\text{cov}(A, B) = \sigma_{A,B}^2 = \frac{1}{n} \sum_{i=1}^n a_i b_i$$

Covariance matrix (2/4)

- Covariance measures the degree of linear relationship between two variables
 - Large and positive covariance: means positively correlated data
 - Large and negative covariance: means negatively correlated data
- $\sigma_{A,B}^2 = 0$ if and only if A and B are uncorrelated
- $\sigma_{A,B}^2 = \sigma_A^2$ if and only if $A=B$
- We can express sets A and B using *row* vectors
 - Use inner product to compute their covariance

$$\mathbf{a} = [a_1, a_2, \dots, a_n] \quad \mathbf{b} = [b_1, b_2, \dots, b_n]$$

$$\text{cov}(\mathbf{a}, \mathbf{b}) = \sigma_{\mathbf{a}, \mathbf{b}}^2 = \frac{1}{n} \mathbf{a} \mathbf{b}^T$$

Covariance matrix (3/4)

- We can generalize from two vectors to an arbitrary number m
- Data matrix $X \in \mathbb{R}^{m \times n}$ (brought to zero mean form)

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}$$

data type 1
(n instants)

data sample (or pattern
or feature vector) at time 2

- X is the data matrix
 - row index $i=1, \dots, m$ is a particular data type
 - column index $j=1, \dots, n$ is the sample number (sampling time)

Covariance matrix (4/4)

- Covariance matrix of \mathbf{X} is:

$$C_{\mathbf{X}} = \frac{1}{n} \mathbf{X} \mathbf{X}^T$$

- Properties of $\mathbf{C}_{\mathbf{X}}$

- The covariance matrix is a square $m \times m$ symmetric matrix (*next slide*)
- The diagonal elements of $\mathbf{C}_{\mathbf{X}}$ are the variances of a measurement type
- The off-diagonal elements of $\mathbf{C}_{\mathbf{X}}$ capture the covariance between differing measurement types

- Covariance values

- Reflect the *noise and redundancy* in our measurements
- In the **diagonal terms**: high values mean **interesting structure**
- In the **off-diagonal terms**: large magnitudes mean **high redundancy**

Intermission - symmetry

- For any matrix A : AA^T and A^TA are symmetric

$$(AA^T)^T = (A^T)^T A^T = AA^T$$

$$(A^T A)^T = A^T (A^T)^T = A^T A$$

- These follows as (trivially)

$$(A^T)^T = A$$

Our objectives – revisited

- In summary, we want:

Obj1) To minimize the redundancy, measured by the covariances

Obj2) To maximize the signal power (SNR), measured by the variance

- Going back to our transformation

$$P\mathbf{X} = \mathbf{Y}$$


covariance matrices of \mathbf{X} and \mathbf{Y} : $C_{\mathbf{X}}$ $C_{\mathbf{Y}}$

- What would the optimized covariance matrix $C_{\mathbf{Y}}$ look like?
 - All its off-diagonal elements should be zero (\mathbf{Y} is uncorrelated) – Obj1
 - Each subsequent dimension in \mathbf{Y} should be
 - rank-ordered according to variance (from largest to smallest) – Obj2

Diagonalize C_Y

- Many methods exist
- PCA assumes that the basis vectors $\{p_1, p_2, \dots, p_m\}$
 - are orthonormal
 - i.e., that P is an orthonormal matrix (p_i are called principal directions)
- How PCA works

Step 1) Select a direction in the m -dimensional data space along which the variance of X is maximized. Save this direction as p_1

Step 2) Find another direction along whose variance is maximized, however, because of the orthonormality condition, restrict the search to all directions orthogonal to all previous selected ones. Save this vector as p_2 (resp. p_i)

Step 3) Repeat this procedure until m vectors are selected

- Method to judge the importance of *principal direction* p_i
 - Rank-ordered according to variance associated with dimension i

PCA: solution

- Goal: find a linear transformation matrix \mathbf{P} such that $\mathbf{Y}=\mathbf{PX}$ and the covariance matrix of \mathbf{Y} (\mathbf{C}_Y) is a diagonal matrix
- Relation between \mathbf{C}_X and \mathbf{C}_Y

$$\begin{aligned}\mathbf{C}_Y &= \frac{1}{n} \mathbf{Y} \mathbf{Y}^T = \frac{1}{n} (\mathbf{P} \mathbf{X}) (\mathbf{P} \mathbf{X})^T = \\ &= \frac{1}{n} \mathbf{P} \mathbf{X} \mathbf{X}^T \mathbf{P}^T = \mathbf{P} \left(\frac{1}{n} \mathbf{X} \mathbf{X}^T \right) \mathbf{P}^T = \mathbf{P} \mathbf{C}_X \mathbf{P}^T\end{aligned}$$

Intermission – Shur's decomposition

- **Theorem 1:** let \mathbf{A} be an $n \times n$ complex matrix. There exists a unitary matrix \mathbf{E} (i.e., $\mathbf{E}^* \mathbf{E} = \mathbf{I}_n$) and an *upper triangular* matrix \mathbf{M} whose diagonal elements are the eigenvalues of \mathbf{A} such that:

$$\mathbf{E}^* \mathbf{A} \mathbf{E} = \mathbf{M}$$

- **Note:** if \mathbf{E} is complex, it can be written as (\mathbf{X} and \mathbf{Y} are two real matrices, with $i^2=-1$)

$$\mathbf{E} = \mathbf{X} + i\mathbf{Y}$$

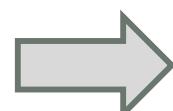
- Its *complex conjugate* is:

$$\mathbf{E}^* = \mathbf{X}^T - i\mathbf{Y}^T$$

Theorem: Eigenvector decomposition

- Theorem 2: for every $n \times n$ **real** and **symmetric** matrix \mathbf{A} . there exists an *orthonormal* $n \times n$ matrix \mathbf{E} (i.e., $\mathbf{E}^T\mathbf{E} = \mathbf{I}_n$) whose **columns** are eigenvectors of \mathbf{A} and a diagonal matrix \mathbf{D} whose elements are the (corresponding) eigenvalues of \mathbf{A} such that:

$$\mathbf{E}^T \mathbf{A} \mathbf{E} = \mathbf{D} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix}$$



$$\mathbf{A} = \mathbf{E} \mathbf{D} \mathbf{E}^T$$

(right- and left-multiplying by \mathbf{E}^T and \mathbf{E})

Eigenvector decomposition Proof

Using Shur's decomposition theorem, there exists a unitary matrix $\mathbf{E} = \mathbf{X} + i\mathbf{Y}$ with real \mathbf{X} and \mathbf{Y} and an upper triangular matrix \mathbf{M} such that $\mathbf{E}^* \mathbf{A} \mathbf{E} = \mathbf{M}$.

Hence, we can write:

$$\begin{aligned}\mathbf{M} &= \mathbf{E}^* \mathbf{A} \mathbf{E} = (\mathbf{X} - i\mathbf{Y})^T \mathbf{A} (\mathbf{X} + i\mathbf{Y}) \\ &= (\mathbf{X}^T \mathbf{A} \mathbf{X} + \mathbf{Y}^T \mathbf{A} \mathbf{Y}) + i(\mathbf{X}^T \mathbf{A} \mathbf{Y} - \mathbf{Y}^T \mathbf{A} \mathbf{X})\end{aligned}$$

Using the symmetry of \mathbf{A} , we have:

$$\mathbf{M} + \mathbf{M}^T = 2(\mathbf{X}^T \mathbf{A} \mathbf{X} + \mathbf{Y}^T \mathbf{A} \mathbf{Y})$$

It follows that $\mathbf{M} + \mathbf{M}^T$ is a real matrix and since \mathbf{M} is triangular, then \mathbf{M} must also be a real matrix

Eigenvector decomposition Proof

- Since \mathbf{M} is a real matrix we have: imaginary part must be zero

$$\begin{aligned}\mathbf{M} &= (\mathbf{X}^T \mathbf{A} \mathbf{X} + \mathbf{Y}^T \mathbf{A} \mathbf{Y}) + i(\cancel{\mathbf{X}^T \mathbf{A} \mathbf{Y}} - \cancel{\mathbf{Y}^T \mathbf{A} \mathbf{X}}) \\ &= \mathbf{X}^T \mathbf{A} \mathbf{X} + \mathbf{Y}^T \mathbf{A} \mathbf{Y} \quad (1)\end{aligned}$$

- From (1), since \mathbf{A} is symmetric, we get $\mathbf{M}^T = \mathbf{M}$, which means that \mathbf{M} is also symmetric
- However, since \mathbf{M} is also *upper triangular*, in order for it to be symmetric, it must also be diagonal !!!

Eigenvector decomposition: Proof

- Up to now we have:

$$\mathbf{E}^* \mathbf{A} \mathbf{E} = \mathbf{M} = \begin{bmatrix} m_1 & 0 & \dots & 0 \\ 0 & m_2 & \dots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \dots & m_n \end{bmatrix}$$

- Which implies (left multiply by E)

$$\mathbf{A} \mathbf{E} = \mathbf{E} \mathbf{M} = \begin{bmatrix} m_1 e_{11} & m_2 e_{12} & \dots & m_n e_{1n} \\ m_1 e_{21} & m_2 e_{22} & \dots & m_n e_{2n} \\ \vdots & \ddots & \ddots & \vdots \\ m_1 e_{n1} & m_2 e_{n2} & \dots & m_n e_{nn} \end{bmatrix}$$

- It means that the diagonal elements of \mathbf{M} (m_i) are *eigenvalues* of \mathbf{A} and the columns of \mathbf{E} (\mathbf{e}_i) are the corresponding *eigenvectors*

Eigenvector decomposition Proof

To conclude:

- The **columns of \mathbf{E}** are *eigenvectors* of **\mathbf{A}**
- The **diagonal elements of \mathbf{M}** are *eigenvalues* of **\mathbf{A}**

$$\mathbf{A}\mathbf{E} = \mathbf{E}\mathbf{M}$$

- since the **diagonal elements of \mathbf{M}** are **real** (proven before)
 - and **matrix \mathbf{A} is real** (by assumption)
 - without loss of generality, **matrix \mathbf{E}** can also be chosen to be **real**
-
- Setting $\mathbf{M} = \mathbf{D}$ concludes the proof.

QED (“quod erat demonstrandum”)

Remark no. 1

From Schur's decomposition theorem, we know that

$$AE = EM$$

- with E being complex, that is $E = X + iY$
- Which means:

$$A(X + iY) = (X + iY)M$$

$$AX + iAY = XM + YM$$

- Which implies

$$\begin{cases} AX = XM \\ AY = YM \end{cases}$$

this also holds setting $Y = 0$

Remark no. 2

Since $A = EDE^T$

The diagonal matrix D can be decomposed as:

$$D = D_{1/2}D_{1/2}$$

where $D_{1/2}$ contains the square root of the eigenvalues

So, the following decomposition also holds:

$$A = EDE^T = ED_{1/2}D_{1/2}E^T = ED_{1/2}(ED_{1/2})^T = CC^T$$

with $C = ED_{1/2}$

Diagonalize C_Y

- New basis transformation $P = \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \vdots \\ \mathbf{p}_m \end{bmatrix}$
(remember: \mathbf{p}_i are row vectors)
- TRICK: we pick **row \mathbf{p}_i of P** as an eigenvector of $C_X = \frac{1}{n}XX^T$
(we choose $\mathbf{A} = C_X$ in the *eigenvalue decomposition theorem*)
- Thus, we have (wrt eigenvalue decomposition): $P = E^T$
- Note also that P is an orthogonal matrix, which means that:

$$P^T P = I_m \Rightarrow P^T = P^{-1}$$

Evaluate \mathbf{C}_Y

- With this choice of \mathbf{P} :

$$\begin{aligned}\mathbf{C}_Y &= \mathbf{P} \mathbf{C}_X \mathbf{P}^T = \mathbf{P} \left(\frac{1}{n} \mathbf{X} \mathbf{X}^T \right) \mathbf{P}^T = \\ &= \mathbf{P} (\mathbf{E} \mathbf{D} \mathbf{E}^T) \mathbf{P}^T = \xrightarrow{\text{apply eigenvector decomposition as } \mathbf{C}_X \text{ is symmetric}} \\ &= \mathbf{P} (\mathbf{P}^T \mathbf{D} \mathbf{P}) \mathbf{P}^T = \mathbf{P} \mathbf{P}^T \mathbf{D} \mathbf{P} \mathbf{P}^T = \xrightarrow{\text{as } \mathbf{P} = \mathbf{E}^T} \\ &= \mathbf{D} \xrightarrow{\text{D is diagonal and contains the eigenvalues of } \mathbf{C}_X \text{ i.e., the variances along each principal direction}}\end{aligned}$$

- This choice of \mathbf{P} diagonalizes $\mathbf{C}_Y!!!$
- In practice, PCA amounts to: (1) subtracting off the mean of each measurement type and (2) computing the eigenvectors of \mathbf{C}_X (computational complexity is $O(m^3)$, note that in general $m \ll n$)

Summary of PCA

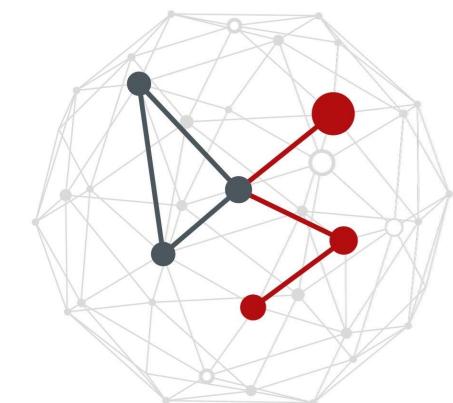
1. Organize data into an $m \times n$ matrix \mathbf{X}
 - m : number of measurement types (*feature vector size*)
 - n : number of samples
2. Compute data (sample) mean vector $\bar{\mathbf{x}}$
3. Subtract off mean vector from dataset $\mathbf{x}_i \leftarrow \mathbf{x}_i - \bar{\mathbf{x}}$
4. Calculate sample covariance matrix \mathbf{C}_x
5. Calculate eigenvectors of matrix $\mathbf{C}_x \rightarrow$ obtain matrix \mathbf{P}
6. Apply change of base $\mathbf{P}\mathbf{X} = \mathbf{Y}$

End - Y is the transformed data matrix

$$\bar{\mathbf{x}} = \left(\sum_{i=1}^n \mathbf{x}_i \right) / n$$

PCA - PART II

Applications



PCA for lossy data compression

- There are m original dimensions in the dataset
- After applying PCA we can decide to retain $K < m$ of these

The setting is as follows [Bishop2011]

- Input (data matrix) values $X = (x_{ij})$
- Output (after PCA) values $Y = (y_{ij})$
- The principal directions are:
 - transpose: \mathbf{p}_i are row vectors, \mathbf{u}_i are column vectors
 - \mathbf{u}_i are the new basis vectors composing \mathbf{P}

$$\mathbf{u}_i = \mathbf{p}_i^T, i = 1, \dots, m$$

[Bishop2011] Christopher Bishop, *Pattern Recognition and Machine Learning*, Springer Nature, 2011.

Applying the transformation

- Let n be the number of input vectors
- Let the *original* vectors be $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ with $\mathbf{x}_i \in \mathbb{R}^m$
 - No mean vector subtracted yet from them
- Once matrix \mathbf{P} is found, for each input vector \mathbf{x}_i
 - the corresponding output (transformed) vector is

$$\mathbf{y}_i = \mathbf{P}(\mathbf{x}_i - \bar{\mathbf{x}}) = \mathbf{P}\mathbf{x}'_i \quad \mathbf{y}_i \in \mathbb{R}^m$$

subtraction of mean

- From which, its j -th element is expressed as

$$y_{ij} = \mathbf{p}_j \mathbf{x}'_i = \mathbf{u}_j^T \mathbf{x}'_i$$

\mathbf{p}_j is a row vector

Expressing data in the new basis

- Alternatively, each original data vector can be **expressed exactly using** the new coordinate system provided by the principal directions \mathbf{u}_i (the new basis vectors)
- This is

$$\mathbf{x}'_i = \sum_{k=1}^m \alpha_{ik} \mathbf{u}_k$$

- Where the α_{ik} coefficients (scalars) *differ for each vector*
- This is simply a rotation of the original coordinate system onto a new system defined by the \mathbf{u}_k
- We want to find the α_{ik} coefficients...

Finding the α coefficients

- Left multiply both terms by \mathbf{u}_j^T

$$\mathbf{u}_j^T \mathbf{x}'_i = \sum_{k=1}^m \alpha_{ik} \mathbf{u}_j^T \mathbf{u}_k$$

- And using the **orthonormality** property

$$\mathbf{u}_j^T \mathbf{u}_k = \delta_{j,k} = \begin{cases} 1 & j = k \\ 0 & j \neq k \end{cases}$$

Kronecker delta

- We obtain

$$\mathbf{u}_j^T \mathbf{x}'_i = \alpha_{ij} \Rightarrow \alpha_{ij} = y_{ij}$$

Retrieving original data from \mathbf{y}

- The original data can be **exactly retrieved** from \mathbf{y} as

$$\mathbf{x}'_i = \sum_{k=1}^m y_{ik} \mathbf{u}_k \quad \text{original vector retrieved from transformed vector } \mathbf{y}$$

- Adding back the mean vector, we get

$$\mathbf{x}_i = \mathbf{x}'_i + \bar{\mathbf{x}} = \sum_{k=1}^m y_{ik} \mathbf{u}_k + \bar{\mathbf{x}}$$

Approximating original data

- Now, the original data can be **approximated**
 - By retaining **the first K elements** in the transformed vector \mathbf{y}
 - With **K< m**, it follows that

$$\tilde{\mathbf{x}}_i = \tilde{\mathbf{x}}'_i + \bar{\mathbf{x}} = \sum_{k=1}^K y_{ik} \mathbf{u}_k + \bar{\mathbf{x}}$$

- and, in turn (gap between original and approx.)

$$\mathbf{x}_i - \tilde{\mathbf{x}}_i = \sum_{k=K+1}^m y_{ik} \mathbf{u}_k$$

Distortion measure J

- Squared distance between *original* and *approximated* data

$$J = \frac{1}{n} \sum_{j=1}^n \| \mathbf{x}_j - \tilde{\mathbf{x}}_j \|^2$$

- For given (arbitrary) K
 - if we use the previous expression for $\tilde{\mathbf{x}}_j$
 - And the first K components are those with largest eigenvalues
- Then J is minimized and we also obtain

$$J = \sum_{k=K+1}^m \lambda_k \quad \text{sum of eigenvalues of discarded dimensions}$$

$$\begin{aligned}
J &= \frac{1}{n} \sum_{i=1}^n \|\mathbf{x}_i - \tilde{\mathbf{x}}_i\|^2 = \frac{1}{n} \sum_{i=1}^n \left\| \sum_{k=K+1}^m y_{ik} \mathbf{u}_k \right\|^2 \\
&= \frac{1}{n} \sum_{i=1}^n \left(\sum_{k=K+1}^m y_{ik} \mathbf{u}_k \right)^T \left(\sum_{r=K+1}^m y_{ir} \mathbf{u}_r \right) \quad \text{transpose of sum = sum of transposes} \\
&= \frac{1}{n} \sum_{i=1}^n \left(\sum_{k=K+1}^m y_{ik} \mathbf{u}_k^T \right) \left(\sum_{r=K+1}^m y_{ir} \mathbf{u}_r \right) \\
&= \frac{1}{n} \sum_{i=1}^n \left(\sum_{k,r=K+1}^m y_{ik} y_{ir} \mathbf{u}_k^T \mathbf{u}_r \right) \xleftarrow{\mathbf{u}_k^T \mathbf{u}_r = \delta_{k,r}} \\
&= \frac{1}{n} \sum_{i=1}^n \sum_{k=K+1}^m y_{ik} y_{ik} \quad \xleftarrow{\text{replace } y_{ik} = \mathbf{u}_k^T \mathbf{x}'_i = (\mathbf{x}'_i)^T \mathbf{u}_k} \\
&= \frac{1}{n} \sum_{i=1}^n \sum_{k=K+1}^m \mathbf{u}_k^T \mathbf{x}'_i (\mathbf{x}'_i)^T \mathbf{u}_k \quad \xrightarrow{\frac{1}{n} \sum_{i=1}^n \mathbf{x}'_i (\mathbf{x}'_i)^T = \mathbf{C_X}} \\
&= \sum_{k=K+1}^m \mathbf{u}_k^T \left(\frac{1}{n} \sum_{i=1}^n \mathbf{x}'_i (\mathbf{x}'_i)^T \right) \mathbf{u}_k = \sum_{k=K+1}^m \lambda_k
\end{aligned}$$

QED

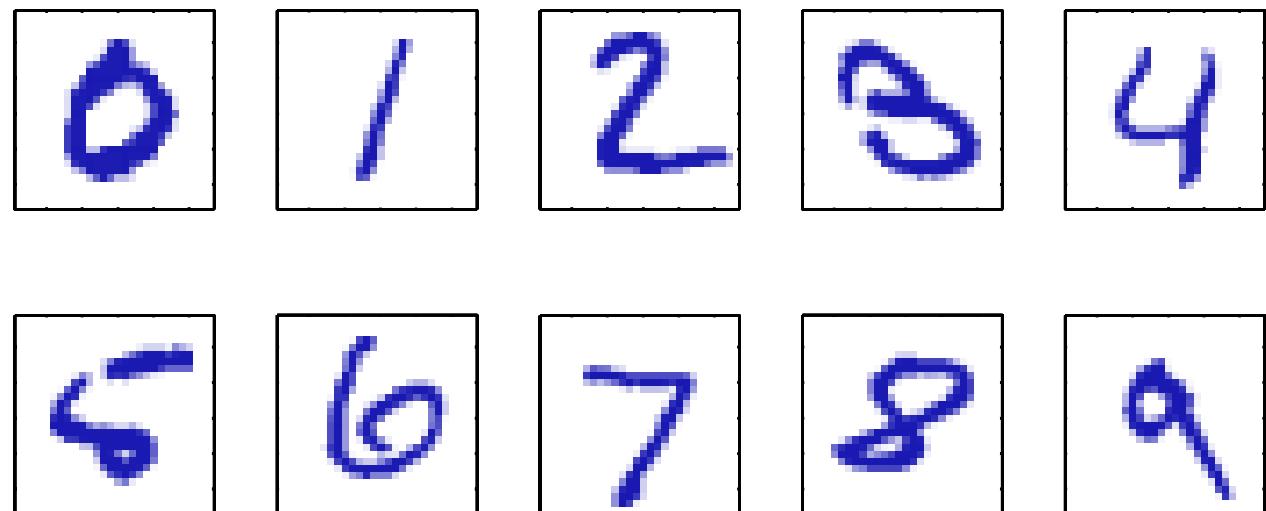
Summary of PCA assumptions

- **Linearity:** linearity frames the problem as a *change of basis*. Several areas of research have explored how extending these notions to *nonlinear regimes* (see, e.g., **autoencoders**)
- **Large variance reveals important structure:** this assumption encompasses the belief that the **useful data structure has a high SNR**. Hence, those principal components with larger associated variances represent interesting structure, while those with lower variances represent noise
- **The principal components are orthogonal:** this assumption provides a simplification that makes PCA (exactly) *solvable with linear algebra decomposition techniques*

Application example – handwritten digits

- Handwritten digits dataset
 - Digits are translated and scaled
 - Each one is contained in a box of the same size
 - Each digit is a 28 x 28 pixel 2D image (784 real numbers)
 - See MNIST handwritten digits dataset 70,000 images
<http://yann.lecun.com/exdb/mnist/>

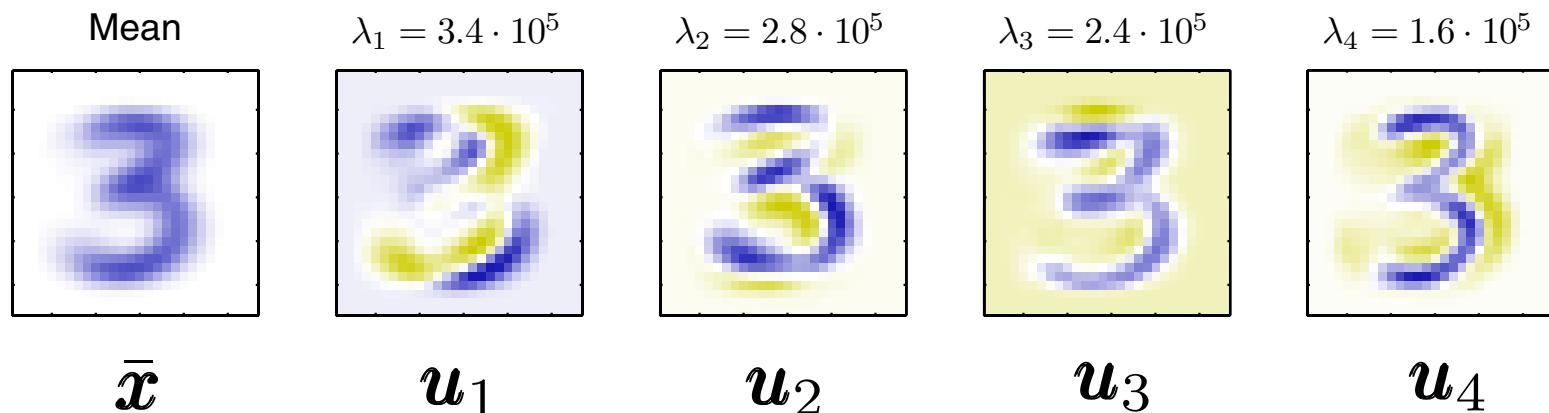
7 2 1 0 4 1 4 9 5 9
0 6 9 0 1 5 9 7 3 4
9 6 6 5 4 0 7 4 0 1
3 1 3 4 7 2 7 1 2 1
1 7 4 2 3 5 1 2 4 4
6 3 5 5 6 0 4 1 9 5
7 8 9 3 7 4 6 4 3 0
7 0 2 9 1 7 3 2 9 7
7 6 2 7 8 4 7 3 6 1
3 6 9 3 1 4 1 7 6 9



Application example – results (1/2)

- Let us consider number 3 and apply PCA on the corresponding data samples

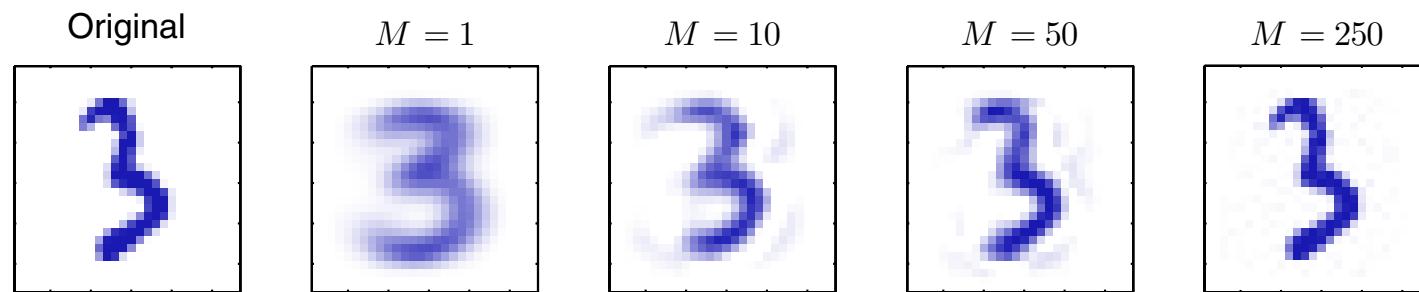
four strongest eigenvalues & corresponding eigenvectors



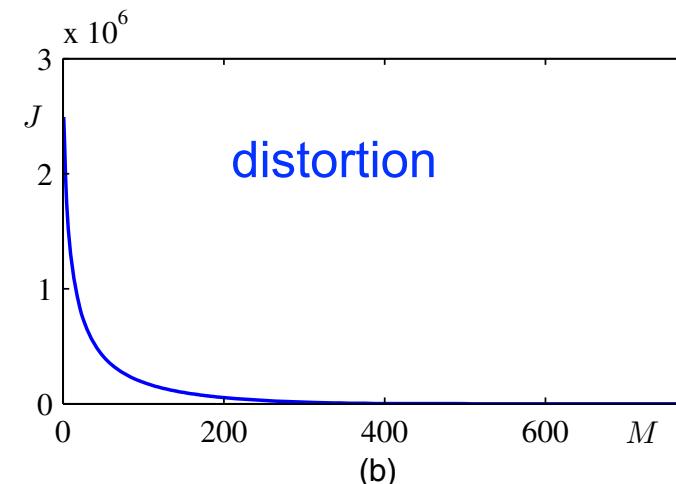
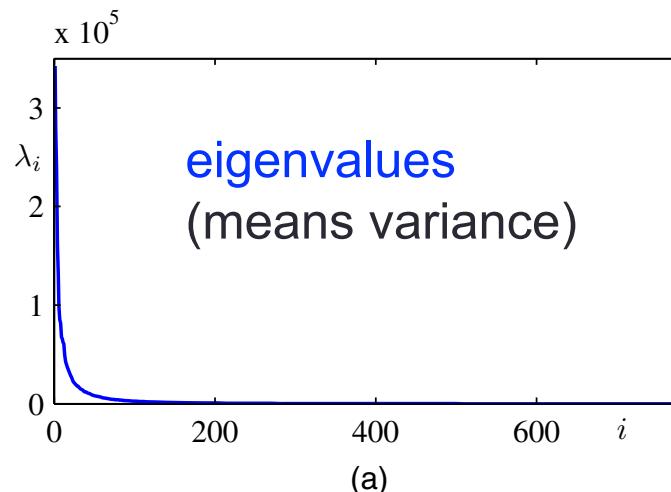
principal directions
reveal important patterns in the data

Application example – results (2/2)

- Reconstruction vs number of retained principal components M



An original example from the off-line digits data set together with its PCA reconstructions obtained by retaining M principal components for various values of M . As M increases the reconstruction becomes more accurate and would become perfect when $M = D = 28 \times 28 = 784$.



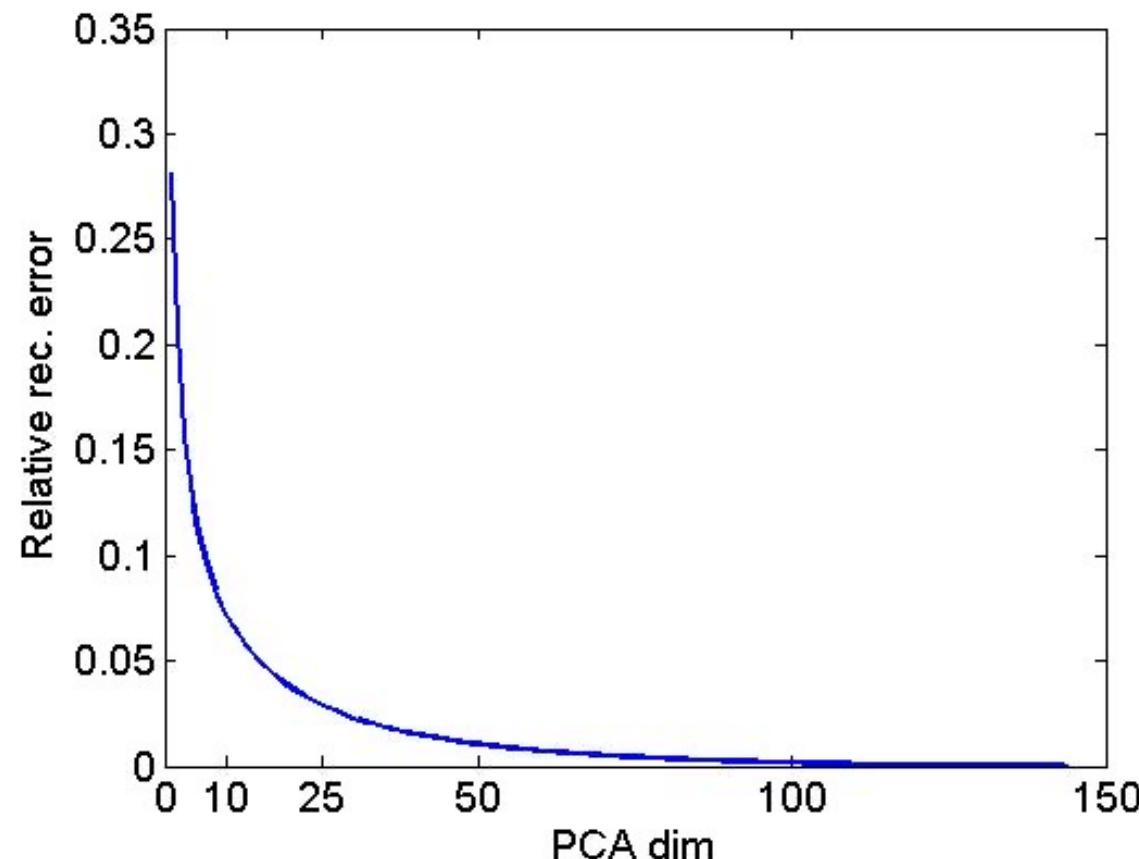
Application – image compression

- Original image (372 x 492 pixels): divide it into patches
 - Each patch is 12 x 12 pixels, view these as a 144D vector



L_2 error vs retained principal directions

- We get 1,271 patches (144 pixels each \rightarrow 144D vectors)
 - Apply PCA on such patches
 - Distortion vs number of retained dimensions



Compression 144D → 60D



Compression 144D → 16D



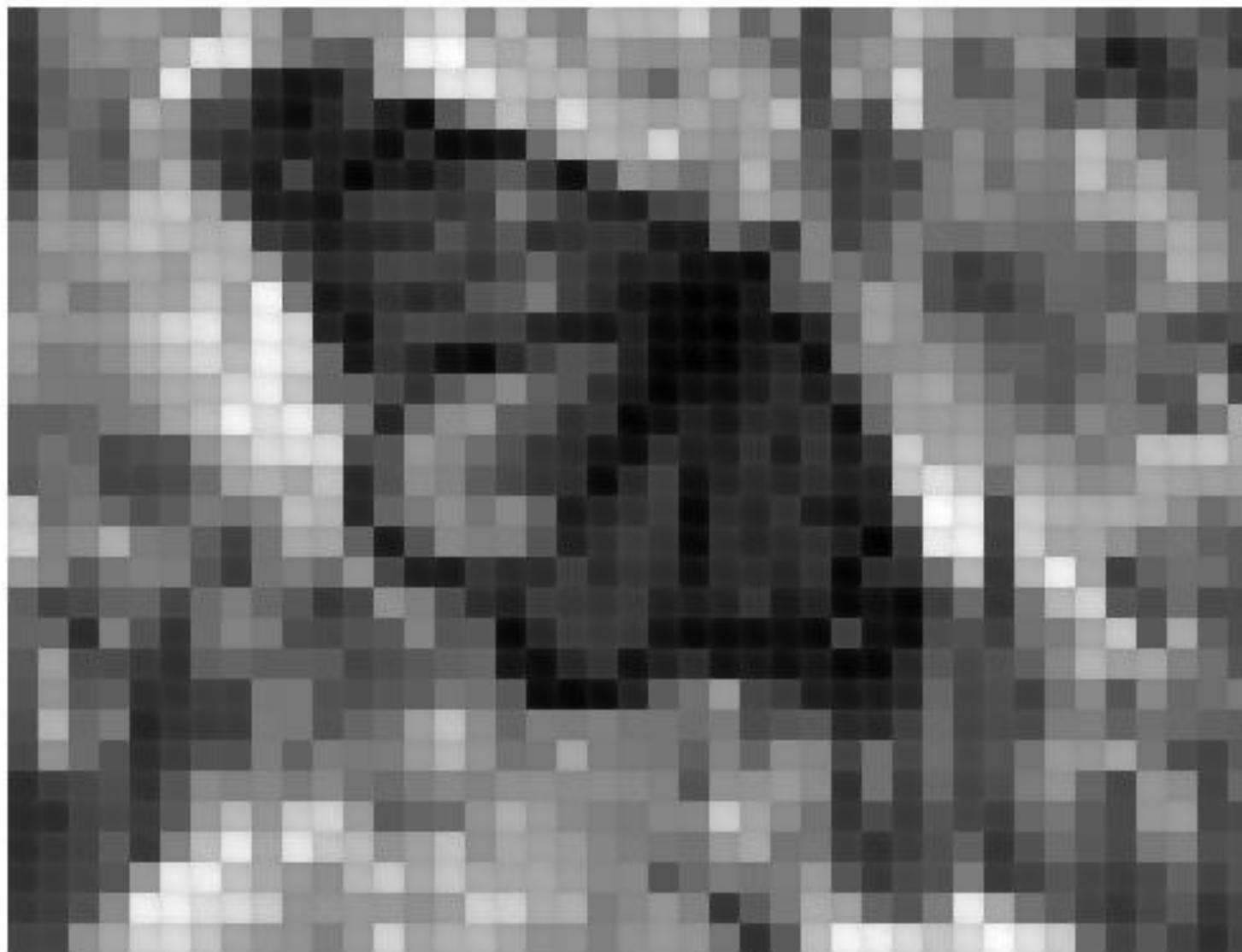
Compression 144D → 6D



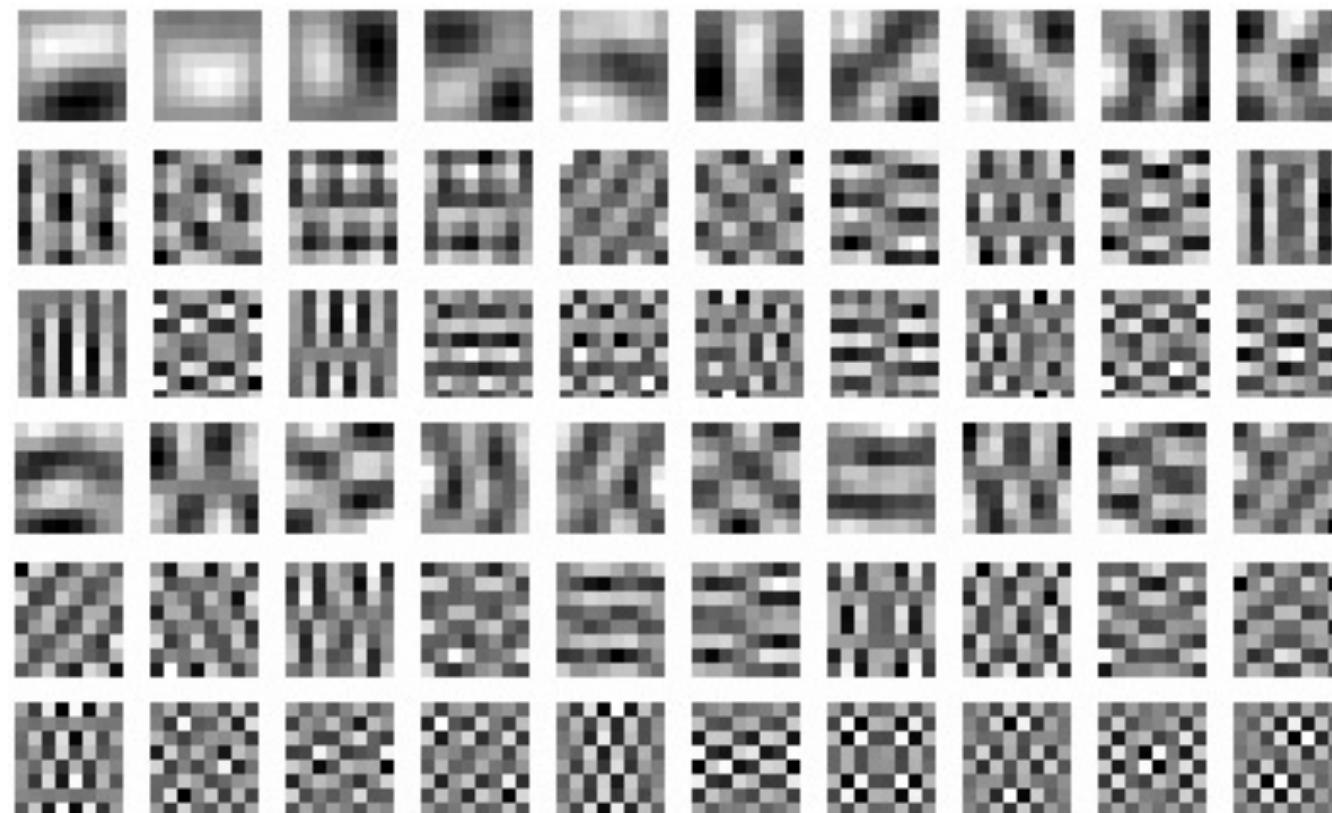
Compression 144D → 3D



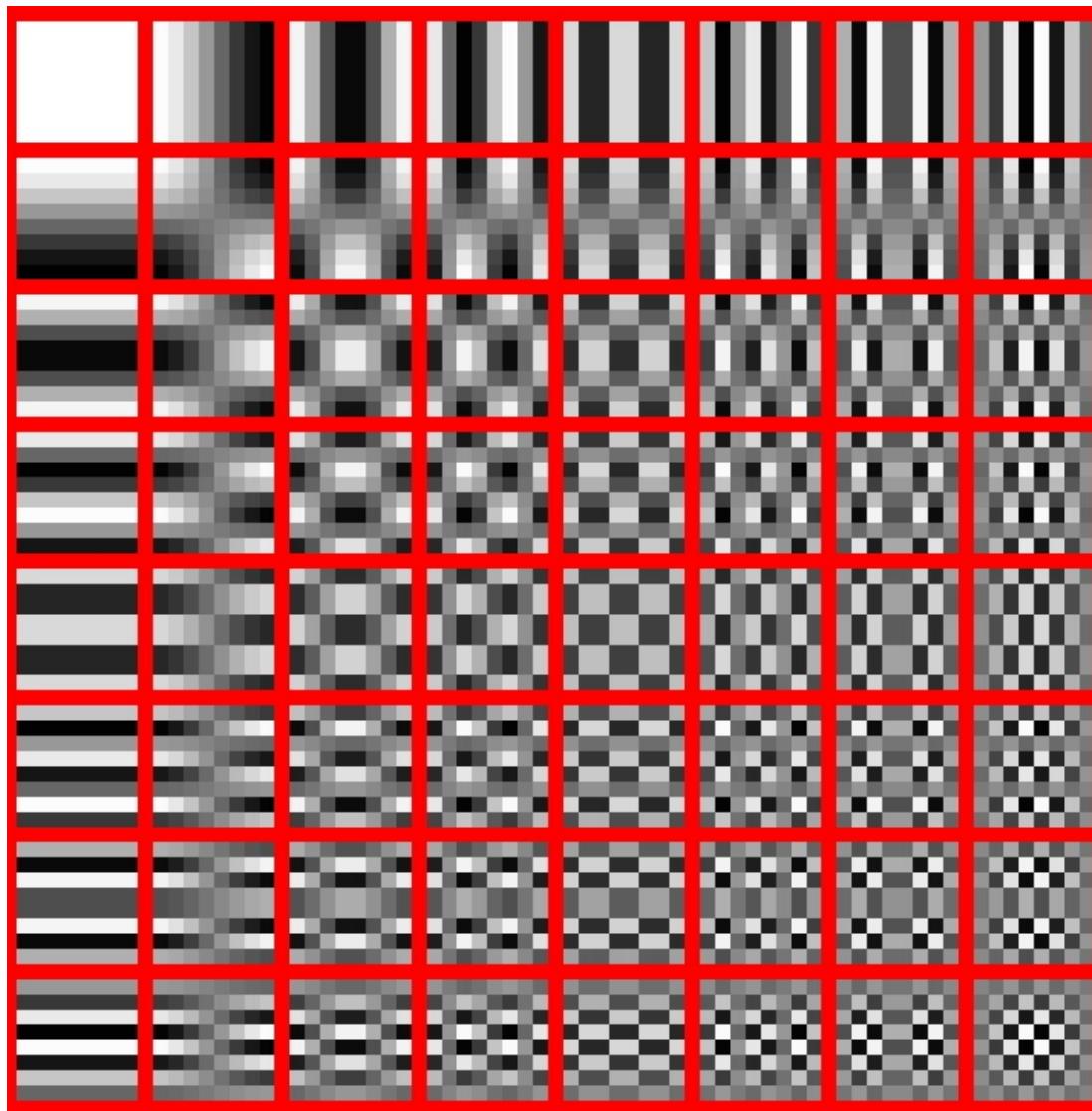
Compression 144D → 1D



60 most important eigenvectors

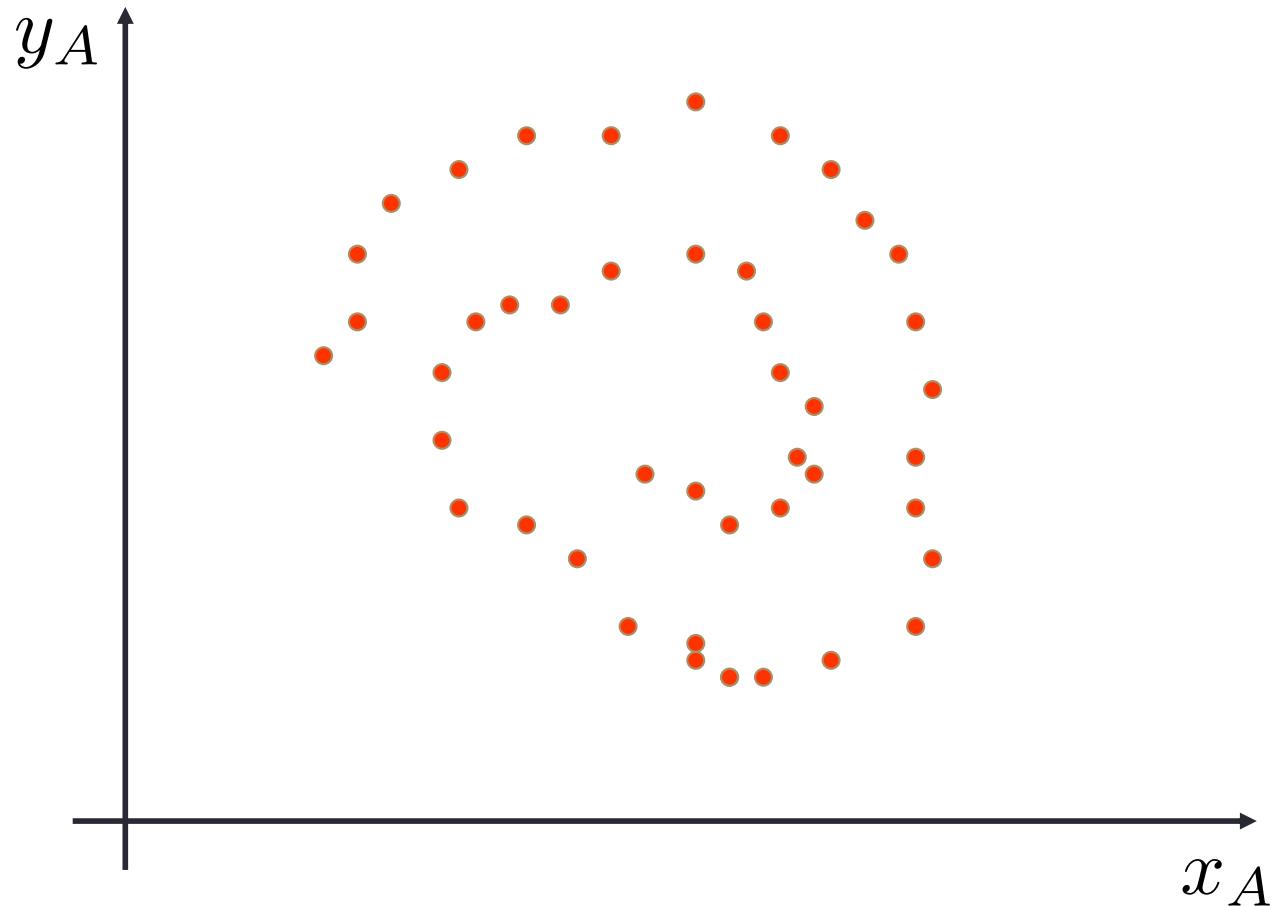


Discrete Cosine (DCT) basis for jpg



Looks pretty similar

A problematic dataset



- PCA is unable to capture nonlinear structure!!!

Where PCA fails

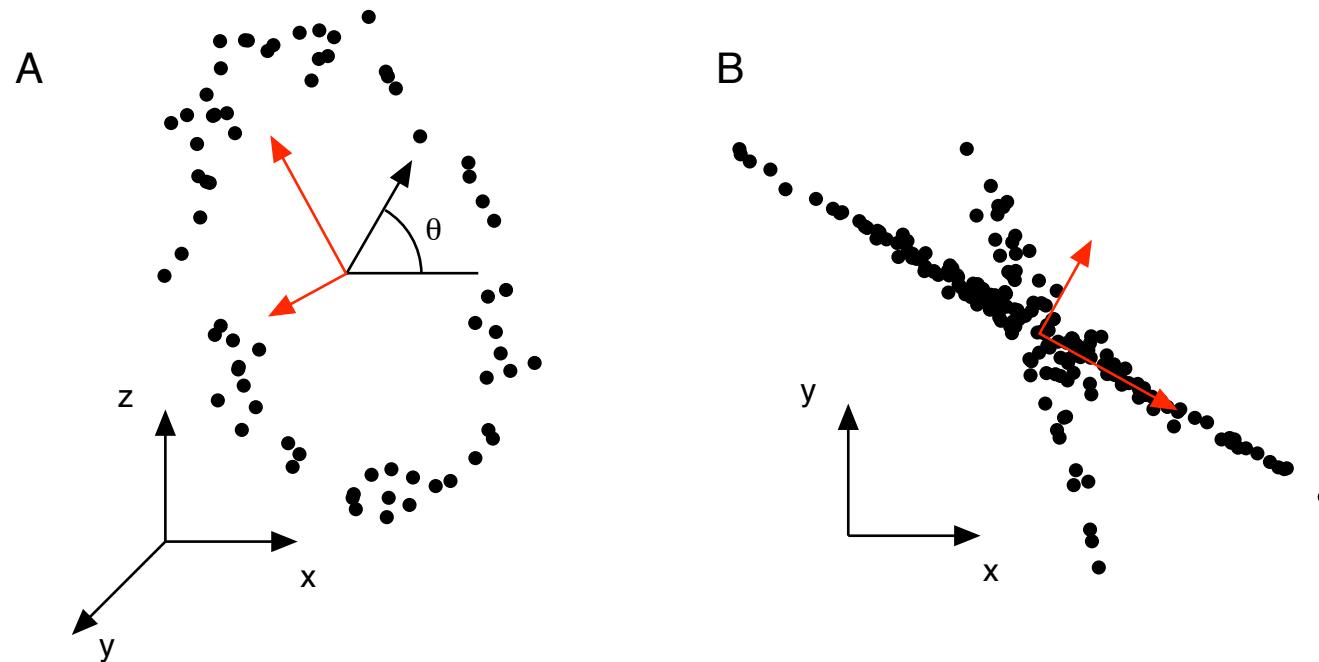


FIG. 6 Example of when PCA fails (red lines). (a) Tracking a person on a ferris wheel (black dots). All dynamics can be described by the phase of the wheel θ , a non-linear combination of the naive basis. (b) In this example data set, non-Gaussian distributed data and non-orthogonal axes causes PCA to fail. The axes with the largest variance do not correspond to the appropriate answer.

Final remarks (1/3)

- Principal component analysis (PCA) has widespread applications because it reveals simple underlying structures in complex data sets **using analytical solutions from linear algebra**
- A primary benefit of PCA arises from **quantifying the importance of each dimension** for describing the variability of a data set
- The value of the **variance along each principal component** provides a means for comparing the relative importance of each dimension
- An implicit **hope behind employing this method** is that the variance along a small number of principal components (i.e., fewer than the number of measurement types) provides a reasonable characterization of the complete data set → this is the precise intuition behind any *dimensionality reduction* method

Final remarks (2/3)

- PCA is completely non-parametric: any data set can be plugged in and an answer comes out, requiring no parameters to tweak and no regard for how the data was recorded
- This means that PCA is an unsupervised technique
- PCA de-correlates a dataset (\mathbf{Y} is de-correlated), this can be useful when a further classification task has to be applied on the data (also known as data “whitening”)
- Under the L_2 norm (common loss function), the objective of PCA (dimensionality reduction) is to approximate the input signal through a reduced set of variables. It can be proven that PCA provides the optimal reduced representation of the input data under the L_2 norm

Final remarks (3/3)

- Further extensions, see Chapter 12 of [2]
 - Online PCA (for large dataset)
 - Kernel PCA
 - Bayesian PCA
 - Independent Component Analysis
- Many more application domains
 - Eigenfaces, see [3]
 - Motion tracking mobile systems [4]

[2] Christopher Bishop, "Pattern Recognition and Machine Learning," Springer, 2006.

[3] M. Turk and A. Pentland, "Eigenfaces for Recognition", Journal of Cognitive Neuroscience, vol. 3, no. 1, 1991.

[4] Matteo Gadaleta, Michele Rossi, "IDNet: Smartphone-based Gait Recognition with Convolutional Neural Networks," Pattern Recognition, Volume 74, Pages 25–37, 2018.

References

- [Shlens14] Jonathon Shlens, “A tutorial on Principal Component Analysis,” arXiv:1404:1100v1, April 7, 2014.
- [Bishop11] Christopher Bishop, “Pattern Recognition and Machine Learning,” Springer, 2011. [Chapter 12](#).
- [Magnus99] Jan R. Magnus, “Matrix Differential Calculus with Applications in Statistics and Econometrics,” *John Wiley & Sons Ltd*, 1999.
- [cookbook-2012] Kaare Brandt Petersen, Michael Syskind Pedersen, “The Matrix Cookbook,” Technical Report, Nov. 15, 2012.
- [Quer12] Giorgio Quer, Riccardo Masiero, Gianluigi Pillonetto, Michele Rossi and Michele Zorzi, ”Sensing, Compression and Recovery for WSNs: Sparse Signal Modeling and Monitoring Framework,” *IEEE Transactions on Wireless Communications* Vol. 11, No. 10, October 2012.

Appendix 1 – quadratic forms (1/3)

- **Definition:** Let \mathbf{A} be an $n \times n$ matrix and \mathbf{x} be a $n \times 1$ vector
- **A quadratic form** is defined as: $\alpha = \mathbf{x}^T \mathbf{A} \mathbf{x}$
- In a quadratic form, we may assume *without loss of generality* that \mathbf{A} is **symmetric**, since we can always replace it with $(\mathbf{A}^T + \mathbf{A})/2$, the **Proof.** follows:

$$\begin{aligned}\mathbf{x}^T \frac{(\mathbf{A} + \mathbf{A}^T)}{2} \mathbf{x} &= \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} + \frac{1}{2} \mathbf{x}^T \mathbf{A}^T \mathbf{x} = \\ &= \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} + \frac{1}{2} (\mathbf{x}^T \mathbf{A} \mathbf{x})^T = \\ &= \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} + \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} = \mathbf{x}^T \mathbf{A} \mathbf{x}\end{aligned}$$

scalar

Appendix 1 – quadratic forms (2/3)

- Thus, let \mathbf{A} be a symmetric matrix. We say that \mathbf{A} is
 - positive definite: if $\mathbf{x}^T \mathbf{A} \mathbf{x} > 0$ for all $\mathbf{x} \neq 0$
 - positive semi-definite: if $\mathbf{x}^T \mathbf{A} \mathbf{x} \geq 0$ for all \mathbf{x}
 - negative definite: if $\mathbf{x}^T \mathbf{A} \mathbf{x} < 0$ for all $\mathbf{x} \neq 0$
 - negative semi-definite: if $\mathbf{x}^T \mathbf{A} \mathbf{x} \leq 0$ for all \mathbf{x}
 - indefinite: if $\mathbf{x}^T \mathbf{A} \mathbf{x} > 0$ for some \mathbf{x} ,
if $\mathbf{x}^T \mathbf{A} \mathbf{x} < 0$ for some \mathbf{x}

Appendix 1 – quadratic forms (3/3)

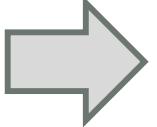
- **Theorem 3:** covariance matrices \mathbf{C}_x are always **positive semi-definite**
- **Proof.** For an arbitrary direction \mathbf{u} we have that, the projected variance of the data \mathbf{X} onto direction \mathbf{u} is obtained as:

$$\begin{aligned}\sigma^2(\mathbf{u}) &= \frac{1}{n} \sum_{i=1}^n (\mathbf{u}^T \mathbf{x}_i - \mathbf{u}^T \bar{\mathbf{x}})^2 = \frac{1}{n} \sum_{i=1}^n (\mathbf{u}^T (\mathbf{x}_i - \bar{\mathbf{x}})) (\mathbf{u}^T (\mathbf{x}_i - \bar{\mathbf{x}}))^T = \\ &= \frac{1}{n} \sum_{i=1}^n \mathbf{u}^T (\mathbf{x}_i - \bar{\mathbf{x}}) (\mathbf{x}_i - \bar{\mathbf{x}})^T \mathbf{u} = \mathbf{u}^T \left[\frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}}) (\mathbf{x}_i - \bar{\mathbf{x}})^T \right] \mathbf{u} = \\ &= \mathbf{u}^T \mathbf{C}_x \mathbf{u}\end{aligned}$$

- This holds for any \mathbf{u} , and for any covariance matrix \mathbf{C}_x
- But the variance is by definition a non-negative scalar $\sigma^2(\mathbf{u}) \geq 0$
- But this means that: $\mathbf{u}^T \mathbf{C}_x \mathbf{u} \geq 0, \forall \mathbf{u}$

QED

Appendix 2 – symmetric matrices (1/3)

- **Theorem 4.** A symmetric matrix is positive definite (semi-definite) if and only if its eigenvalues are positive (non-negative)
 - **Proof.**  **(sufficiency)**
-
- Assume \mathbf{A} is positive definite and write: $\mathbf{Ax} = \lambda\mathbf{x}$
 - Pre-multiplying by \mathbf{x}^T , we get: $\mathbf{x}^T \mathbf{Ax} = \lambda \mathbf{x}^T \mathbf{x}$
 - Since by assumption, $\mathbf{x}^T \mathbf{Ax} > 0$, $\mathbf{x} \neq 0$
 - And likewise, it must be (quadratic norm): $\mathbf{x}^T \mathbf{x} > 0$, $\mathbf{x} \neq 0$
 - Then, the eigenvalues **must be positive** $\lambda > 0$

Appendix 2 – symmetric matrices (2/3)

- **Theorem 4.** A symmetric matrix is positive definite (semi-definite) if and only if its eigenvalues are positive (non-negative)
- **Proof.**  (necessity)

- Now, assume all the eigenvalues of \mathbf{A} are positive, $\lambda_i > 0$
- From the eigenvector decomposition theorem we have:

$$\mathbf{E}^T \mathbf{A} \mathbf{E} = \mathbf{D} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix}$$

$\mathbf{E}^T \mathbf{A} \mathbf{E} = \mathbf{D}$ columns of \mathbf{E} are eigenvectors of \mathbf{A}

Appendix 2 – symmetric matrices (3/3)

- Proof. (continued). Recall that:

- \mathbf{E} is an orthonormal basis of eigenvectors of \mathbf{A} : $\mathbf{E}^T \mathbf{A} \mathbf{E} = \mathbf{D}$
- Moreover, *this relation is equivalent to*:

$$\mathbf{A} = \mathbf{E} \mathbf{D} \mathbf{E}^T$$

- Let us now apply a *change of variable*:

$$\mathbf{u} = \mathbf{E} \mathbf{v}$$

- With this, we have that:

$$\mathbf{u}^T \mathbf{A} \mathbf{u} = \mathbf{v}^T \mathbf{E}^T \mathbf{E} \mathbf{D} \mathbf{E} \mathbf{E}^T \mathbf{E} \mathbf{v} = \mathbf{v}^T \mathbf{D} \mathbf{v}$$

- This shows that $\mathbf{u}^T \mathbf{A} \mathbf{u}$ is positive for any non-zero \mathbf{u} only if \mathbf{D} is positive for any non-zero \mathbf{v} , i.e., only if \mathbf{D} is positive definite. Moreover, the diagonal matrix \mathbf{D} is positive definite only if each element of the diagonal (i.e., each eigenvalue of \mathbf{A}) is positive. Since this is true by assumption the theorem is proven. The same holds for the semi-positive definite case.

QED

Appendix 3 – covariance matrices

- We have learned that
 - A matrix of the form (**mxm**): $C_X = \frac{1}{n}XX^T$
 - Is symmetric
 - Is positive semi-definite
 - remember: its eigenvalues are variances
 - hence, its non-zero eigenvalues must be greater than zero
 - The same properties apply to (although it is size **nxn**)

$$X^T X$$

Appendix 4 – eigenvectors & eigenvalues

- **Theorem 5:** the eigenvectors of a *symmetric matrix* are *orthogonal*
- **Proof.** Let λ_1 and λ_2 be two distinct eigenvalues

$$A\mathbf{u}_1 = \lambda_1 \mathbf{u}_1 \quad A\mathbf{u}_2 = \lambda_2 \mathbf{u}_2 , \text{ with: } \lambda_1 \neq \lambda_2$$

$$\lambda_1 \mathbf{u}_1 \cdot \mathbf{u}_2 = (\lambda_1 \mathbf{u}_1)^T \mathbf{u}_2 =$$

$$= (A\mathbf{u}_1)^T \mathbf{u}_2 =$$

$$= \mathbf{u}_1^T A^T \mathbf{u}_2 =$$

$$= \mathbf{u}_1^T (A\mathbf{u}_2) =$$

$$= \mathbf{u}_1^T (\lambda_2 \mathbf{u}_2) = \lambda_2 \mathbf{u}_1 \cdot \mathbf{u}_2$$

- This implies that:

$$(\lambda_1 - \lambda_2) \mathbf{u}_1 \cdot \mathbf{u}_2 = 0 \Rightarrow \mathbf{u}_1 \cdot \mathbf{u}_2 = 0 \quad \text{QED}$$

Appendix 5 – useful relations

- **Theorem 6:** Let \mathbf{A} ($m \times n$) and \mathbf{B} ($m \times p$) be two matrices and \mathbf{x} ($n \times 1$) be a column vector. Then, the following relations hold:

$$(a) \quad \mathbf{Ax} = \mathbf{0} \Leftrightarrow \mathbf{A}^T \mathbf{Ax} = \mathbf{0}$$

$$(b) \quad \mathbf{AB} = \mathbf{0} \Leftrightarrow \mathbf{A}^T \mathbf{AB} = \mathbf{0}$$

- **Proof.** (a) Clearly $\mathbf{Ax} = \mathbf{0} \Rightarrow \mathbf{A}^T \mathbf{Ax} = \mathbf{0}$

- Conversely, if

$$\mathbf{A}^T \mathbf{Ax} = \mathbf{0}$$

- Then, we also have:

$$(\mathbf{Ax})^T (\mathbf{Ax}) = \mathbf{x}^T \mathbf{A}^T \mathbf{Ax} = \mathbf{0}$$

- The last equality is a square norm, which implies $\mathbf{Ax} = \mathbf{0}$
- Part (b) immediately follows from (a) by applying (a) to each column of matrix \mathbf{B} .

QED

Appendix 6 – Singular Value Decomposition

- Theorem 7: Let \mathbf{A} be a matrix ($m \times n$) with $\text{rank}(\mathbf{A})=r>0$. Then, there exist:
(i) an orthonormal matrix \mathbf{U} ($m \times r$), **(ii)** an orthonormal matrix \mathbf{V} ($n \times r$), and
(iii) a diagonal matrix Σ ($r \times r$) with positive diagonal elements, such that:

$$\mathbf{A} = \mathbf{U}\Sigma^{1/2}\mathbf{V}^T$$

- where (orthonormal matrices):

$$\mathbf{U}^T\mathbf{U} = \mathbf{I}_r \quad \mathbf{V}^T\mathbf{V} = \mathbf{I}_r$$

- Σ contains the **r non-zero eigenvalues** of $\mathbf{A}\mathbf{A}^T$
- \mathbf{U} contains the (orthonormal) **eigenvectors** of $\mathbf{A}\mathbf{A}^T$
- \mathbf{V} contains the (orthonormal) **eigenvectors** of $\mathbf{A}^T\mathbf{A}$

Appendix 6 – Singular Value Decomposition

- Proof.
- Note that $\mathbf{A}\mathbf{A}^T$ is a real $m \times m$ symmetric matrix
- From Theorem 3, this matrix is:
 - positive semi-definite
- From Theorem 4:
 - its r non-zero eigenvalues are $\lambda_i > 0$
- Moreover, a property of the rank is:

$$\text{rank}(\mathbf{A}) = \text{rank}(\mathbf{A}^T) = \text{rank}(\mathbf{A}^T \mathbf{A}) = \text{rank}(\mathbf{A}\mathbf{A}^T)$$

Appendix 6 – Singular Value Decomposition

- Proof. (continued)

- Since AA^T is a real $m \times m$ and **symmetric** matrix

- From **Theorem 2**:

$$(AA^T)E = ED$$

matrix Σ

$$D = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 & 0 \\ 0 & \lambda_2 & \dots & 0 & 0 \\ 0 & 0 & \lambda_r & \dots & 0 \\ 0 & 0 & 0 & \ddots & \vdots \\ 0 & 0 & 0 & \ddots & 0 \end{bmatrix}$$

\uparrow \downarrow
 \uparrow \downarrow
 r non-zero eigenvalues of AA^T
 $m-r$ zero eigenvalues of AA^T

- We then re-express matrix E ($m \times m$) separating the **first r columns** and the following **$m-r$ ones**:

$$E \triangleq \begin{bmatrix} U & U_* \end{bmatrix}_{m \times r \quad m \times (m-r)}$$

Appendix 6 – Singular Value Decomposition

- Proof. (continued)

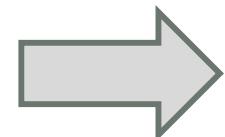
- Since $E \triangleq [U | U_*]$

$$D = \begin{bmatrix} & & & \text{matrix } \Sigma \\ \lambda_1 & 0 & \dots & 0 & 0 \\ 0 & \lambda_2 & \dots & 0 & 0 \\ 0 & 0 & \lambda_r & \dots & 0 \\ 0 & 0 & 0 & \ddots & \vdots \\ 0 & 0 & 0 & \ddots & 0 \end{bmatrix}$$

↑
r non-zero eigenvalues of $\mathbf{A}\mathbf{A}^T$
↓
m-r zero eigenvalues of $\mathbf{A}\mathbf{A}^T$

- Due to the block structure of D

$$(\mathbf{A}\mathbf{A}^T)E = ED = [U|U_*]D = [U\Sigma|0]$$



$$(\mathbf{A}\mathbf{A}^T)[U|U_*] = [U\Sigma|0]$$

Appendix 6 – Singular Value Decomposition

- Proof. (continued)
- Since matrix \mathbf{E} ($m \times m$) is orthonormal, it holds:

$$[\mathbf{U} | \mathbf{U}_*][\mathbf{U} | \mathbf{U}_*]^T = \mathbf{U}\mathbf{U}^T + \mathbf{U}_*\mathbf{U}_*^T = I_m$$

- Matrix \mathbf{U}_*
 - From Theorem 2 (previous slide), we have: $(\mathbf{A}\mathbf{A}^T)\mathbf{U}_* = \mathbf{0}$
 - Using Theorem 6(b), this also implies: $\mathbf{A}^T\mathbf{U}_* = \mathbf{0} \rightarrow \mathbf{U}_*^T\mathbf{A} = \mathbf{0}$
- Matrix \mathbf{U}
 - From Theorem 2 (previous slide), for the non-zero eigenvalues, it holds:

$$\mathbf{A}\mathbf{A}^T\mathbf{U} = \mathbf{U}\Sigma \quad (1)$$

Appendix 6 – Singular Value Decomposition

- Proof. (continued)
- (Th 2) matrix Σ contains the r positive eigenvalues of AA^T
- Let us define a new matrix:

$$V = A^T U \Sigma^{-1/2} \quad (2)$$

- Given these facts, we write:

$$\begin{aligned} A^T A V &= A^T A A^T U \Sigma^{-1/2} && \xleftarrow{\text{using (2)}} \quad \xleftarrow{\text{using (1)}} \\ &= A^T U \Sigma \Sigma^{-1/2} = \\ &= A^T U \Sigma^{1/2} = V \Sigma \end{aligned}$$

- We also have (easy to see using (1)): $V^T V = I_r$

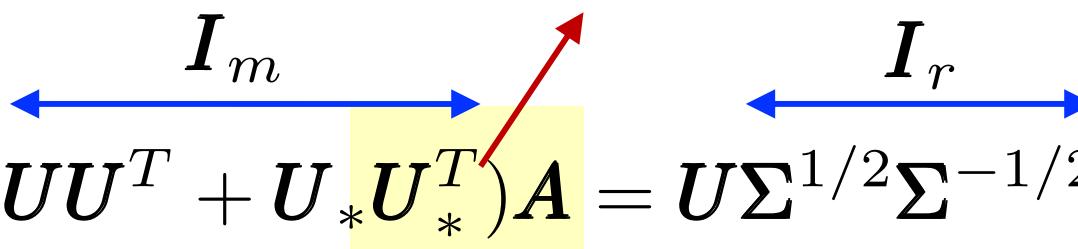
Appendix 6 – Singular Value Decomposition

- Proof. (continued)
- Given the previous expressions, we use the following trick:

this is zero – Theorem 6(b)

$$\begin{aligned} A &= (UU^T + U_*U_*^T)A = U\Sigma^{1/2}\Sigma^{-1/2}U^TA = \\ &= U\Sigma^{1/2}(A^TU\Sigma^{-1/2})^T = U\Sigma^{1/2}V^T \end{aligned}$$

$\xleftarrow[V]{\quad\quad\quad}$



- Hence, matrix A can be decomposed into:

$$A = U\Sigma^{1/2}V^T$$

Appendix 6 – Singular Value Decomposition

- Proof. (continued)
- Note that, in this process we have found that:

$$(AA^T)U = U\Sigma \quad (3)$$

$$(A^TA)V = V\Sigma \quad (4)$$

- (3) reveals that Σ contains the **r non-zero eigenvalues** of AA^T these eigenvalues correspond to those of A^TA
- U contains the (orthonormal) **eigenvectors** of AA^T
- V contains the (orthonormal) **eigenvectors** of A^TA
- $\Sigma^{1/2}$ contains the square root of the r non-zero eigenvalues

QED

Appendix 7 - PCA vs SVD (1/3)

- They are intimately related
- Let \mathbf{X} ($m \times n$) be our data matrix (mean has been removed)

- Define a new matrix \mathbf{Z} as:

$$\mathbf{Z} \triangleq \frac{1}{\sqrt{n}} \mathbf{X}$$

- Hence, we have that: $\mathbf{Z}\mathbf{Z}^T = \frac{1}{n} \mathbf{X}\mathbf{X}^T = \mathbf{C}_{\mathbf{X}}$
- Note that $\mathbf{Z}\mathbf{Z}^T$ is the **covariance matrix** of our data and corresponds to $\mathbf{A}\mathbf{A}^T$ in the SVD theory (taking $\mathbf{Z}=\mathbf{A}$), in the **SVD Theorem 7**

Appendix 7 - PCA vs SVD (2/3)

- Applying SVD to $\mathbf{Z}\mathbf{Z}^T$:
 - The columns of \mathbf{U} contains the eigenvectors associated with the non-zero eigenvalues of $\mathbf{Z}\mathbf{Z}^T = \mathbf{C}_x$
 - These columns form an orthonormal basis
 - The columns of \mathbf{U} are the principal directions (components) of \mathbf{C}_x
- Matrix $[\mathbf{U} \mid \mathbf{U}_*]$ corresponds to matrix \mathbf{E} in Theorem 2
 - We know that the columns of matrix \mathbf{E} are the eigenvectors of the covariance matrix \mathbf{C}_x (all eigenvalues)
 - From the SVD Proof. we also know:
$$\mathbf{Z}^T \mathbf{U}_* = (\mathbf{U}_*^T \mathbf{Z})^T = \mathbf{0}$$
 - This means that \mathbf{U}_* does not contribute to relevant (non-zero) elements in the transformed PCA space

Appendix 7 - PCA vs SVD (3/3)

- 1) Organize data vectors into a matrix \mathbf{X} (mxn), where n : number of samples (input vectors), m : number of measurement types
- 2) Subtract the mean from each data vector
- 3) Calculate SVD: $\mathbf{X} = \mathbf{U}\Sigma^{1/2}\mathbf{V}^T$
 - columns of \mathbf{U} are the eigenvectors of the r positive eigenvalues of $\mathbf{Z}\mathbf{Z}^T = \mathbf{C}_x$ (we have, $\text{rank}(\mathbf{C}_x) = r$). From the previous equation, pre-multiplying both sides by \mathbf{U}^T , we get:

$$\mathbf{U}^T \mathbf{X} = \mathbf{U}^T \mathbf{U} \Sigma^{1/2} \mathbf{V}^T = \Sigma^{1/2} \mathbf{V}^T = \mathbf{Y}$$

- Rows of matrix \mathbf{U}^T are the eigenvectors of \mathbf{C}_x and $\mathbf{U}^T = \mathbf{P}$ is the PCA transform. With this formulation, the transformed data \mathbf{Y} contains r rows (the rank of the covariance matrix) and n columns

Appendix 7 - Summing up

The transformed data points after PCA are:

$$U^T X = Y$$

where:

$$X \in \mathbb{R}^{m \times n} \quad Y \in \mathbb{R}^{r \times n}, \text{ with: } r \leq m$$

- only r eigenvectors are sufficient to represent Y with no information loss
- $m-r$: represents the number of linearly dependent variables
 - These are automatically found by PCA
 - The result is that in the PCA space r orthogonal variables suffice to represent the signal
- **PCA is useful for**
 - Energy compaction (in transformed space)
 - Dimensionality reduction (**from m to r vars, or fewer if we accept loss**)

Appendix 8 – eigenvectors & eigenvalues

- **Theorem 8:** eigenvectors associated with distinct eigenvalues are *linearly independent*

- **Proof.**

- Let $A\mathbf{u}_1 = \lambda_1 \mathbf{u}_1$ $A\mathbf{u}_2 = \lambda_2 \mathbf{u}_2$, with: $\lambda_1 \neq \lambda_2$

- Assume \mathbf{u}_1 and \mathbf{u}_2 are **linearly dependent**, i.e.,

$$\mathbf{u}_2 = \beta \mathbf{u}_1, \text{ with } \beta \neq 0$$

- If this is true, the following equalities hold:

$$\beta \lambda_1 \mathbf{u}_1 = \beta A\mathbf{u}_1 = A\mathbf{u}_2 = \lambda_2 \mathbf{u}_2 = \beta \lambda_2 \mathbf{u}_1$$

- This implies:

$$\beta(\lambda_1 - \lambda_2) \mathbf{u}_1 = 0$$

Since β is non-zero and the eigenvalues also differ, this equality implies $\mathbf{u}_1=0$. This also implies $\mathbf{u}_2=0$ and, in turn, $\lambda_1=\lambda_2$, a contradiction. **QED**

Appendix 9

- **Theorem 9:** a quadratic form $f(\mathbf{x}) = \mathbf{x}^T Q \mathbf{x}$ where Q is a symmetric positive semi-definite matrix **is convex**
- **Proof.** A necessary and sufficient condition (if and only if) for convexity is, for all points in the domain of the function

$$f(\mathbf{x}_1) - f(\mathbf{x}_2) \geq (\mathbf{x}_1 - \mathbf{x}_2)^T \nabla f(\mathbf{x}_2), \quad \forall \mathbf{x}_1, \mathbf{x}_2$$

- As we have seen, if Q is symmetric and positive semi-definite, there exists a matrix C such that the following decomposition holds

$$Q = C^T C$$

- Now, consider the following square norm, which is non-negative

$$\begin{aligned}\|C\mathbf{x}_1 - C\mathbf{x}_2\|^2 &= (C\mathbf{x}_1 - C\mathbf{x}_2)^T (C\mathbf{x}_1 - C\mathbf{x}_2) = \\ &= \mathbf{x}_1^T C^T C \mathbf{x}_1 - 2\mathbf{x}_1^T C^T C \mathbf{x}_2 + \mathbf{x}_2^T C^T C \mathbf{x}_2 = \\ &= \mathbf{x}_1^T Q \mathbf{x}_1 - 2\mathbf{x}_1^T Q \mathbf{x}_2 + \mathbf{x}_2^T Q \mathbf{x}_2 \geq 0\end{aligned}$$

Appendix 9

- The previous inequality can be rewritten as

$$\mathbf{x}_1^T \mathbf{Q} \mathbf{x}_1 - 2\mathbf{x}_1^T \mathbf{Q} \mathbf{x}_2 + \mathbf{x}_2^T \mathbf{Q} \mathbf{x}_2 \geq 0$$

$$\mathbf{x}_1^T \mathbf{Q} \mathbf{x}_1 - 2\mathbf{x}_1^T \mathbf{Q} \mathbf{x}_2 + 2\mathbf{x}_2^T \mathbf{Q} \mathbf{x}_2 - \mathbf{x}_2^T \mathbf{Q} \mathbf{x}_2 \geq 0$$

$$\mathbf{x}_1^T \mathbf{Q} \mathbf{x}_1 - \mathbf{x}_2^T \mathbf{Q} \mathbf{x}_2 \geq 2\mathbf{x}_1^T \mathbf{Q} \mathbf{x}_2 - 2\mathbf{x}_2^T \mathbf{Q} \mathbf{x}_2 = (\mathbf{x}_1 - \mathbf{x}_2)^T 2\mathbf{Q} \mathbf{x}_2$$

- But for a quadratic form with symmetric matrix it holds that

$$f(\mathbf{x}) = \mathbf{x}^T \mathbf{Q} \mathbf{x} \Rightarrow \nabla f(\mathbf{x}) = 2\mathbf{Q} \mathbf{x}$$

- Using this in the RHS of the previous inequality leads to the desired results

$$f(\mathbf{x}_1) - f(\mathbf{x}_2) \geq (\mathbf{x}_1 - \mathbf{x}_2)^T \nabla f(\mathbf{x}_2), \quad \forall \mathbf{x}_1, \mathbf{x}_2$$

QED

PRINCIPAL COMPONENT ANALYSIS (PCA)

Michele Rossi

michele.rossi@unipd.it

<http://www.dei.unipd.it/~rossi/>

University of Padova, IT



DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE



DIPARTIMENTO
MATEMATICA



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

