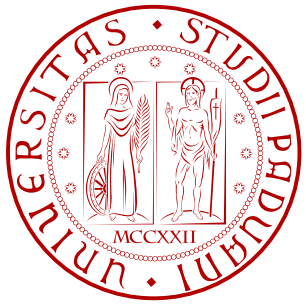
PRINCIPAL COMPONENT ANALYSIS (PCA)

Michele Rossi

rossi@dei.unipd.it

Dept. of Information Engineering

University of Padova, IT



What is it?

- A dimensionality reduction technique

- Applications
  - Lossy data compression
  - Feature extraction
  - Clustering
  - Data visualization
- Used in diverse fields to
  - Extract relevant information in big and confusing data sets
  - Simple, non-parametric 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

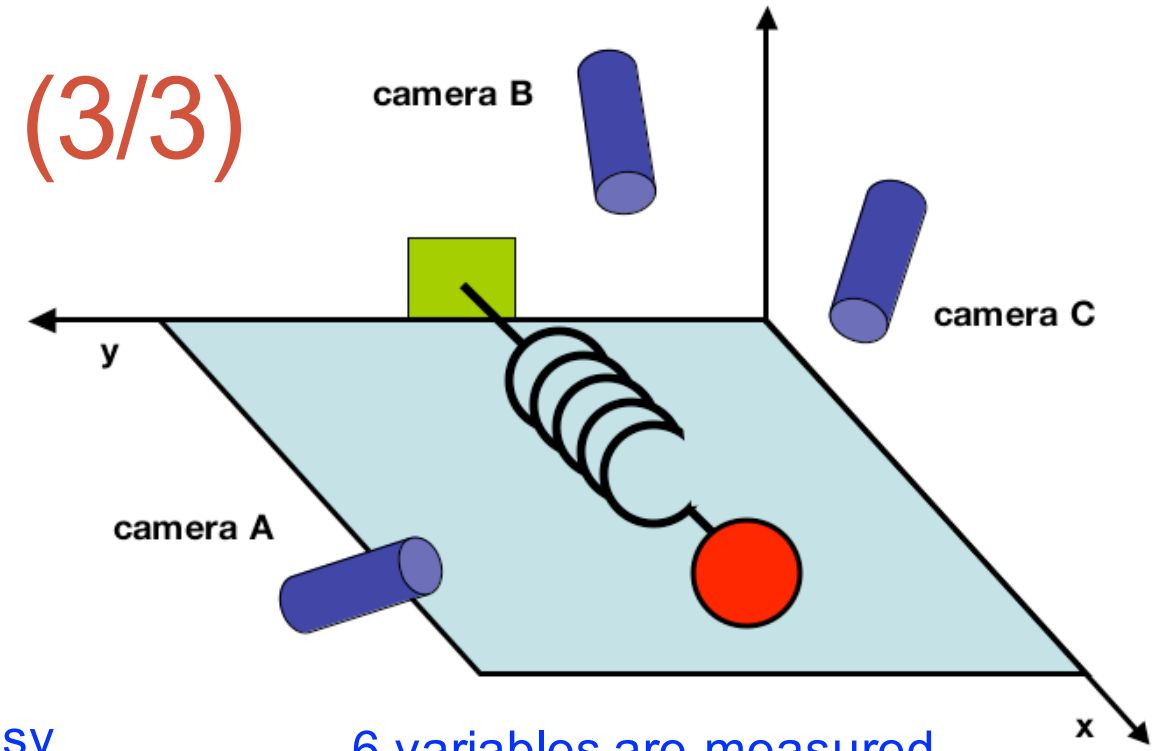
Toy example (2/3)

"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.

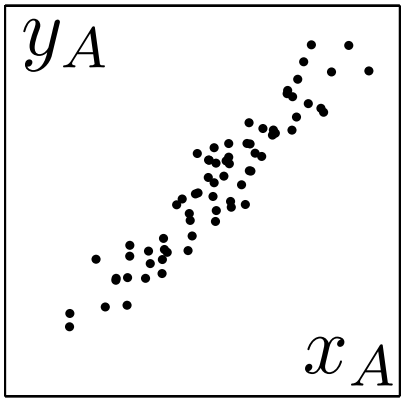
Toy example (3/3)



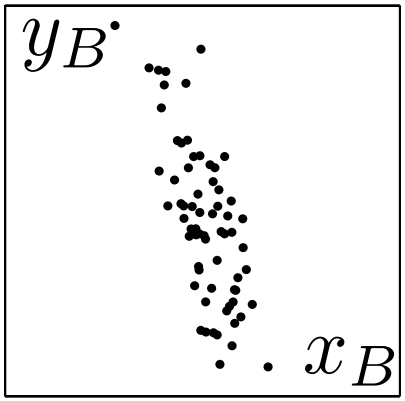
measurements are noisy

6 variables are measured
(2 projections per camera)

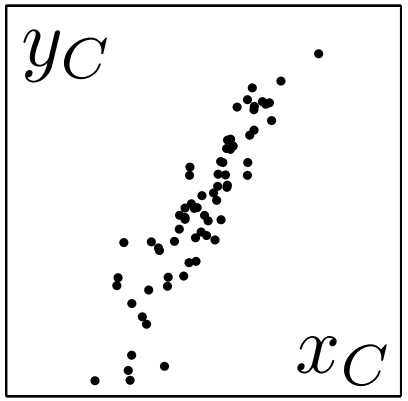
camera A



camera B



camera C



The framework – "change of basis"

- Goal of PCA
  - Find a new basis to re-express a dataset
- 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 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 (samples)
- each sample (at time i) is a *column vector*
  - Collecting all m measurement types at time i (m=6 in the toy example)

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

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

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \dots & \boldsymbol{x}_n \end{bmatrix}$$

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

The naïve basis (*natural basis* for the Euclidean space)

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

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

- 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)

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

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 *powerful* assumption: linearity
  - This *greatly* simplifies the problem

- Problem setup
  - Original data points $\boldsymbol{X} \in \mathbb{R}^{m \times n}$
  - New basis $\boldsymbol{P} \in \mathbb{R}^{m \times m}$
  - Transformed data points $\boldsymbol{Y} \in \mathbb{R}^{m \times n}$

Change of basis (2/3)

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

$$\boldsymbol{PX} = \boldsymbol{Y}$$

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

$$\boldsymbol{PX} = \begin{bmatrix} \boldsymbol{p}_1 \\ \vdots \\ \boldsymbol{p}_m \end{bmatrix} \underbrace{\begin{bmatrix} \boldsymbol{x}_1 \cdots \boldsymbol{x}_n \end{bmatrix}}_{\text{input data (columns)}} = \begin{bmatrix} \boldsymbol{p}_1\boldsymbol{x}_1 & \cdots & \boldsymbol{p}_1\boldsymbol{x}_n \\ \vdots & \ddots & \vdots \\ \boldsymbol{p}_m\boldsymbol{x}_1 & \cdots & \boldsymbol{p}_m\boldsymbol{x}_n \end{bmatrix} \triangleq \underbrace{\begin{bmatrix} \boldsymbol{y}_1 \cdots \boldsymbol{y}_n \end{bmatrix}}_{\text{transformed data (columns)}}$$

Change of basis (3/3) $\boldsymbol{PX} = \boldsymbol{Y}$

$$\boldsymbol{PX} = \begin{bmatrix} \boldsymbol{p}_1 \\ \vdots \\ \boldsymbol{p}_m \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \cdots \boldsymbol{x}_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{p}_1\boldsymbol{x}_1 & \cdots & \boldsymbol{p}_1\boldsymbol{x}_n \\ \vdots & \ddots & \vdots \\ \boldsymbol{p}_m\boldsymbol{x}_1 & \cdots & \boldsymbol{p}_m\boldsymbol{x}_n \end{bmatrix} \triangleq \begin{bmatrix} \boldsymbol{y}_1 \ldots \boldsymbol{y}_n \end{bmatrix}$$

- $\boldsymbol{y}_i\ ,\ i = 1, \ldots, n$
  - is a projection onto the basis $\{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_m\}$

- The row vectors $\{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_m\}$ in the new basis are referred to as the principal directions of $\boldsymbol{X}$

Open questions

- What is the best way to re-express **X**?
- What is a good choice of basis **P**?

- Answering these questions
  - Implies understanding what features we would like **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 (SRN) as the ratio of their variances, i.e.,

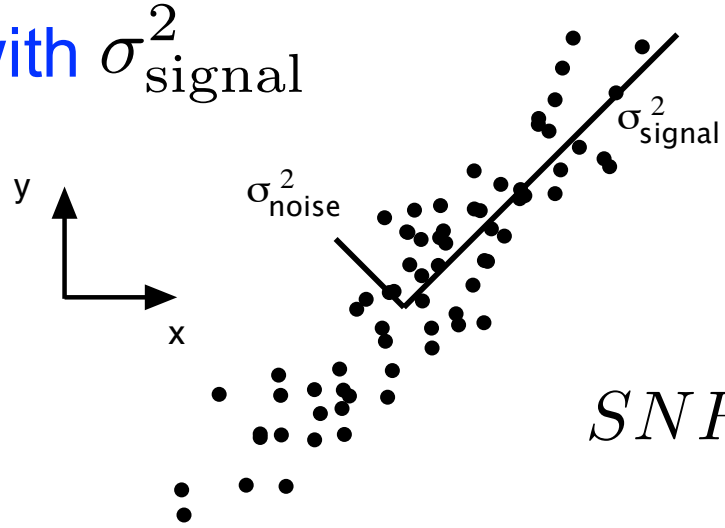
$$SNR = \frac{\sigma^2_{\text{signal}}}{\sigma^2_{\text{noise}}}$$

(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^2_{signal}



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

FIG. 2 Simulated data of (x, y) for camera A. The signal and noise variances σ^2_{signal} and σ^2_{noise} 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.

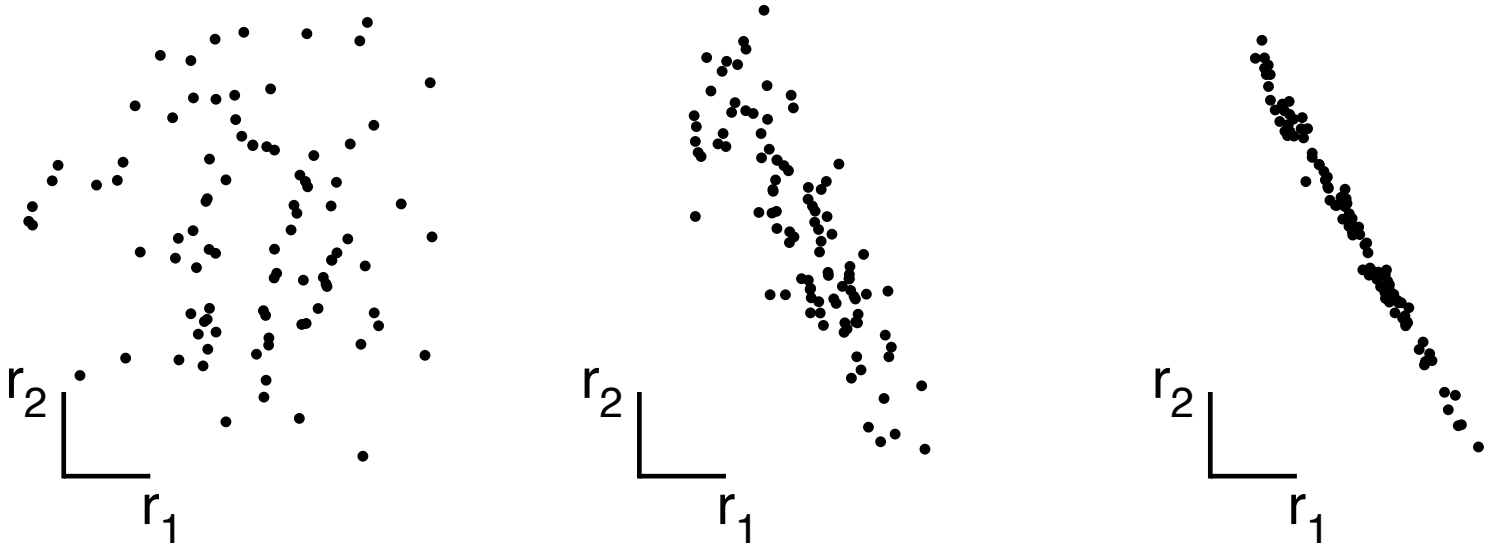
Assumption

- 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?

Redundancy

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



low redundancy

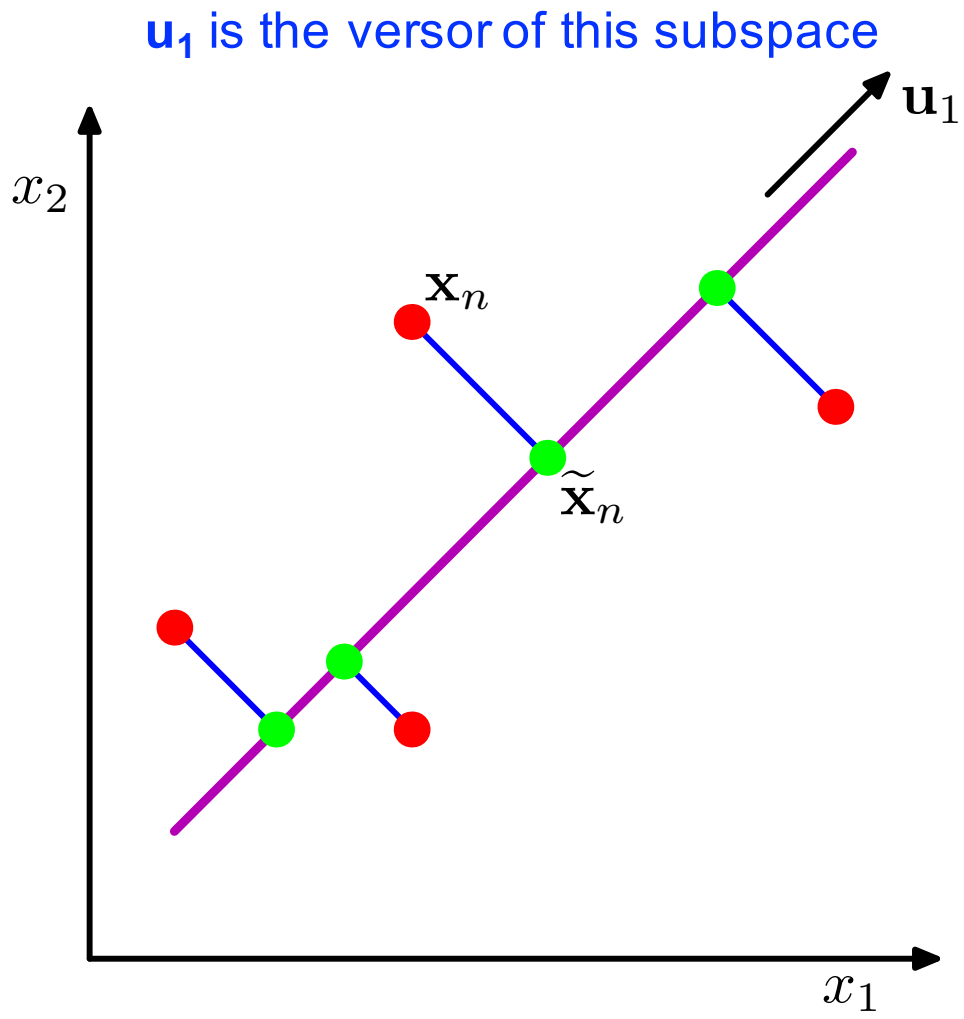
high redundancy

(low correlation between r_1 and r_2)

- 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)

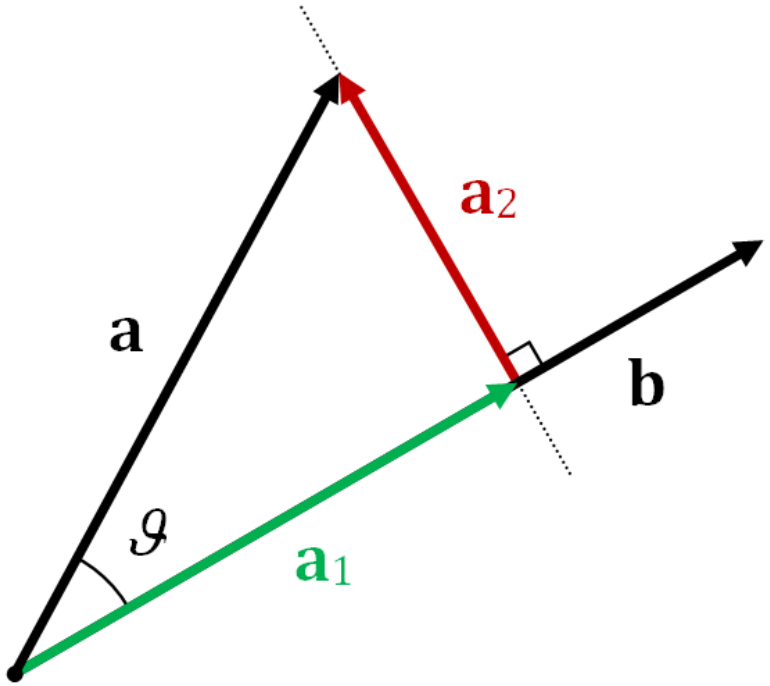


Projection into vector

- Projection of vector **a** into vector **b**
- It is a vector parallel to **b** defined as:

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

versor along direction **b**
(it is unit length)



- The scalar projection a_1 along **b** is found as:

$$a_1 = |\boldsymbol{a}| \cos(\theta) = |\boldsymbol{a}||\boldsymbol{b}_v| \cos(\theta) = < \boldsymbol{a}, \boldsymbol{b}_v > =$$

inner (dot) product

$$= \boldsymbol{a} \cdot \boldsymbol{b}_v = \boldsymbol{a}^T \boldsymbol{b}_v =$$

$$= \boldsymbol{b}_v^T \boldsymbol{a} = < \boldsymbol{b}_v, \boldsymbol{a} >$$

commutative property
of dot product

Finding $\mathbf{u}_1$ (1/4)

- $\boldsymbol{u}_1$ is a *unit* vector (versor), that is: $\boldsymbol{u}_1^T \boldsymbol{u}_1 = 1$
- Sample mean vector of the input data is:

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

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

Finding $\mathbf{u}_1$ (2/4)

- Variance of the projected data ("projected variance") is:

$$\frac{1}{n}\sum_{i=1}^{n}\left(\boldsymbol{u}_1^T\boldsymbol{x}_i - \boldsymbol{u}_1^T\bar{\boldsymbol{x}}\right)^2 = \boldsymbol{u}_1^T\boldsymbol{C}_{\boldsymbol{X}}\boldsymbol{u}_1$$

projection of data points — projection of mean

- Where the data covariance matrix is defined as:

$$\boldsymbol{C}_{\boldsymbol{X}} = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T$$

Finding $\mathbf{u}_1$ (2/4 bis)

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

scalar

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

$$= \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{u}_1^T(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T\boldsymbol{u}_1 = \boldsymbol{u}_1^T\left[\frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T\right]\boldsymbol{u}_1 =$$

$$= \boldsymbol{u}_1^T\boldsymbol{C}_{\boldsymbol{X}}\boldsymbol{u}_1$$

- where:

$$\boldsymbol{C}_{\boldsymbol{X}} = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T$$

Finding $\mathbf{u}_1$ (3/4)

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

$$\max_{\boldsymbol{u}_1} [\boldsymbol{u}_1 \boldsymbol{C}_{\boldsymbol{X}} \boldsymbol{u}_1]$$

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

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

$$J(\boldsymbol{u}_1) = \boldsymbol{u}_1^T \boldsymbol{C}_{\boldsymbol{X}} \boldsymbol{u}_1 + \lambda_1 (1 - \boldsymbol{u}_1^T \boldsymbol{u}_1)$$

Intermission – gradient of a quadratic form

- Let α be the quadratic form: $\alpha \triangleq \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x}$
  - where: $\boldsymbol{x}$ is nx1, $\boldsymbol{A} = [a_{ij}]$ is nxn and does not depend on $\boldsymbol{x}$

- We define the gradient of α as the row vector:

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

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

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

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. In compact form, this can be written as:

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

Intermission – gradient of a quadratic form

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

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

- In column form, we get:

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

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

Intermission – gradient of square norm

- Square norm-2: $$\alpha = \boldsymbol{x}^T \boldsymbol{x} = \sum_{i=1}^{n} x_i^2$$
- Which leads to: $$\frac{\partial \alpha}{\partial x_j} = 2x_j$$
- In (column) vector form: $$(\Delta \alpha(\boldsymbol{x}))^T = 2\boldsymbol{x}$$
- So we have (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 $\mathbf{u}_1$ (4/4)

- Lagrangian function J

$$J(\boldsymbol{u}_1) = \boldsymbol{u}_1^T \boldsymbol{C}_{\boldsymbol{X}} \boldsymbol{u}_1 + \lambda_1 (1 - \boldsymbol{u}_1^T \boldsymbol{u}_1)$$

$$\frac{dJ(\boldsymbol{u}_1)}{d\boldsymbol{u}_1} = 0 \Rightarrow \boldsymbol{C}_{\boldsymbol{X}} \boldsymbol{u}_1 = \lambda_1 \boldsymbol{u}_1$$

- This says that $\mathbf{u}_1$ must be an *eigenvector* of matrix $\mathbf{C}_\mathbf{x}$
- If we left-multiply by $\mathbf{u}_1^T$ and make use of the constraint:

$$\boldsymbol{u}_1^T \boldsymbol{C}_{\boldsymbol{X}} \boldsymbol{u}_1 = \lambda_1$$

- Which says that the variance is maximized when we set $\mathbf{u}_1$ equal to the eigenvector having the largest eigenvalue λ_1

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

$$A = \{a_1, a_2, \ldots, a_n\} \qquad B = \{b_1, b_2, \ldots, b_n\}$$

- Variances

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

- Covariance of **a** and **b**

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

Covariance matrix (2/4)

- Covariance measures the degree of linear relationship between two variables
  - Large and positive variance: positively correlated data
  - Large and negative variance: negatively correlated data
  - $\sigma^2_{A,B} = 0$ if and only if A and B are uncorrelated
  - $\sigma^2_{A,B} = \sigma^2_A$ if and only if A=B
- We can convert A and B into *row* vectors
  - Use inner product to compute their covariance

$$\boldsymbol{a} = [a_1, a_2, \ldots, a_n] \qquad \boldsymbol{b} = [b_1, b_2, \ldots, b_n]$$

$$\mathrm{cov}(\boldsymbol{a}, \boldsymbol{b}) = \sigma^2_{\boldsymbol{a},\boldsymbol{b}} = \frac{1}{n}\boldsymbol{a}\boldsymbol{b}^T$$

Covariance matrix (3/4)

- We can generalize from two vectors to an arbitrary number m
- Data matrix $\boldsymbol{X} \in \mathbb{R}^{m \times n}$

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

data type 1 (n instants)

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

- X is the data matrix
  - row index i=1,…,m is a particular data type
  - column index j=1,…,n is the sample number

Covariance matrix (4/4)

- Covariance matrix of **X** is:

$$C_X = \frac{1}{n} X X^T$$

- Properties of $\mathbf{C_X}$
  - The covariance matrix is a square m x m symmetric matrix (*next slide*)
  - The diagonal elements of $\mathbf{C_X}$ are the variances of a measurement type
  - The off diagonal elements of $\mathbf{C_X}$ capture the covariance between 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: A^TA and AA^T are symmetric

$$(\boldsymbol{A}\boldsymbol{A}^T)^T = (\boldsymbol{A}^T)^T\boldsymbol{A}^T = \boldsymbol{A}\boldsymbol{A}^T$$

$$(\boldsymbol{A}^T\boldsymbol{A})^T = \boldsymbol{A}^T(\boldsymbol{A}^T)^T = \boldsymbol{A}^T\boldsymbol{A}$$

- These follows as (trivially)

$$(\boldsymbol{A}^T)^T = \boldsymbol{A}$$

Our objectives – revisited

- In summary, we want:
  1) To minimize the redundancy, measured by the covariances
  2) To maximize the signal power, measured by the variance
- Going back to our transformation

$$PX = Y$$

covariance matrices of **X** and **Y**: C_X C_Y

- What would the optimized covariance matrix C_Y look like?
  - All its off-diagonal elements should be zero (Y is decorrelated)
  - Each successive dimension in Y should be
    - rank-ordered according to variance (from largest to smallest)

Diagonalize C_Y

- Many methods exist
- PCA assumes that the basis vectors $\{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_m\}$
  - are orthonormal
  - i.e., that **P** is an orthonormal matrix ($\mathbf{p}_i$ are called principal directions)
- How PCA works
  1) Select a direction in the m-dimensional data space along which the variance of **X** is maximized. Save this direction as $\mathbf{p}_1$
  2) Find another direction along which 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 $\mathbf{p}_i$
  3) Repeat this procedure until m vectors are selected
- Method to judge the importance of *principal direction* $\mathbf{p}_i$
  - Rank-ordered according to variance associated with dimension i

PCA: solution

- Goal: find a linear transformation matrix **P** such that **Y=PX** and the covariance matrix of **Y** (C_Y) is a diagonal matrix

- Relation between $\mathbf{C_X}$ and $\mathbf{C_Y}$

$$\boldsymbol{C_Y} = \frac{1}{n}\boldsymbol{YY}^T = \frac{1}{n}(\boldsymbol{PX})(\boldsymbol{PX})^T =$$

$$= \frac{1}{n}\boldsymbol{PXX}^T\boldsymbol{P}^T = \boldsymbol{P}\left(\frac{1}{n}\boldsymbol{XX}^T\right)\boldsymbol{P}^T = \boldsymbol{PC_X P}^T$$

Intermission – Shur's decomposition

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

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

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

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

- Its *complex conjugate* is:

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

Theorem: Eigenvector decomposition

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

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

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

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

Eigenvector decomposition Proof

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

Hence, we can write:

$$\boldsymbol{M} = \boldsymbol{E}^* \boldsymbol{A} \boldsymbol{E} = (\boldsymbol{X} - i\boldsymbol{Y})^T \boldsymbol{A} (\boldsymbol{X} + i\boldsymbol{Y})$$

$$= (\boldsymbol{X}^T \boldsymbol{A} \boldsymbol{X} + \boldsymbol{Y}^T \boldsymbol{A} \boldsymbol{Y}) + i(\boldsymbol{X}^T \boldsymbol{A} \boldsymbol{Y} - \boldsymbol{Y}^T \boldsymbol{A} \boldsymbol{X})$$

Using the symmetry of **A**, we have:

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

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

Eigenvector decomposition Proof

- Since **M** is a real matrix we have:

imaginary part must be zero

$$\boldsymbol{M} = (\boldsymbol{X}^T\boldsymbol{A}\boldsymbol{X} + \boldsymbol{Y}^T\boldsymbol{A}\boldsymbol{Y}) + i(\boldsymbol{X}^T\boldsymbol{A}\boldsymbol{Y} - \boldsymbol{Y}^T\boldsymbol{A}\boldsymbol{X})$$

$$= \boldsymbol{X}^T\boldsymbol{A}\boldsymbol{X} + \boldsymbol{Y}^T\boldsymbol{A}\boldsymbol{Y} \quad (1)$$

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

Eigenvector decomposition: Proof

- Up to now we have:

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

- Which implies (left multiply by E)

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

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

Eigenvector decomposition Proof

To conclude:

- The columns of **E** are *eigenvectors* of **A**
- The diagonal elements of **M** are *eigenvalues* of **A**

$$\boldsymbol{AE = EM}$$

  - Since the diagonal elements of **M** are real (proven before)
  - And matrix **A** is real (by assumption)
  - Without loss of generality, matrix **E** can also be chosen to be real

- Setting **M** = **D** concludes the proof.

q.e.d.

Diagonalize C_Y

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

$$\boldsymbol{P}^T\boldsymbol{P} = \boldsymbol{I}_n \Rightarrow \boldsymbol{P}^T = \boldsymbol{P}^{-1}$$

Evaluate C_Y

- With this choice of **P**:

$$\boldsymbol{C_Y} = \boldsymbol{P}\boldsymbol{C}_X\boldsymbol{P}^T = \boldsymbol{P}\left(\frac{1}{n}\boldsymbol{X}\boldsymbol{X}^T\right)\boldsymbol{P}^T =$$

$$= \boldsymbol{P}\left(\boldsymbol{E}\boldsymbol{D}\boldsymbol{E}^T\right)\boldsymbol{P}^T =$$

→ eigenvector decomposition as $\mathbf{C}_X$ is symmetric

$$= \boldsymbol{P}\left(\boldsymbol{P}^T\boldsymbol{D}\boldsymbol{P}\right)\boldsymbol{P}^T = \boldsymbol{P}\boldsymbol{P}^T\boldsymbol{D}\boldsymbol{P}\boldsymbol{P}^T =$$

→ as $\mathbf{P} = \mathbf{E}^T$

$$= \boldsymbol{D}$$

→ D is diagonal and contains the eigenvalues of $\mathbf{C}_X$ i.e., the variances along each principal direction

- This choice of **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 $\mathbf{C}_X$

Summary of PCA

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

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

End - Y is the transformed data matrix

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

- Input (data matrix) values $\boldsymbol{X} = (x_{ij})$
- Output (after PCA) values $\boldsymbol{Y} = (y_{ij})$

- The principal directions are: $\boldsymbol{u}_i = \boldsymbol{p}_i^T \ , \ i = 1, \ldots, m$

  (**NOTE:** $\mathbf{p}_i$ are *row* vectors, $\mathbf{u}_i$ are *column* vectors)

PCA for lossy data compression

- Point (i,j) is a projection onto i-th principal direction

$$y_{ij} = \boldsymbol{x}_j^T \boldsymbol{u}_i$$

- Original data re-expressed in new basis $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_m\}$

$$\boldsymbol{x}_j = \sum_{i=1}^{m} y_{ij} \boldsymbol{u}_i = \sum_{i=1}^{m} (\boldsymbol{x}_j^T \boldsymbol{u}_i) \boldsymbol{u}_i$$

- If we only retain K<m dimensions in the transformed **Y**

$$\tilde{\boldsymbol{x}}_j = \sum_{i=1}^{K} (\boldsymbol{x}_j^T \boldsymbol{u}_i) \boldsymbol{u}_i + \sum_{i=K+1}^{m} (\bar{\boldsymbol{x}}^T \boldsymbol{u}_i) \boldsymbol{u}_i$$

strongest K directions are retained

remaining m-K use average information

Distortion measure J

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

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

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

$$J = \sum_{i=K+1}^{m} \lambda_i$$

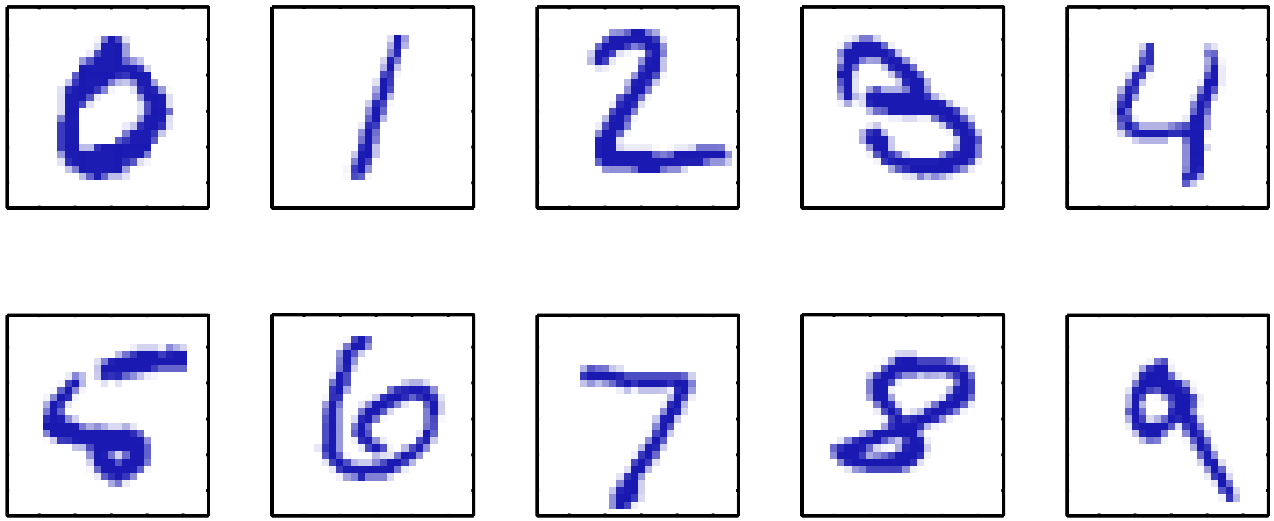
sum of discarded eigenvalues

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*
- Large variance reveal important structure: this assumption encompasses the belief that the data 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 *solvable with linear algebra decomposition techniques*

Application example – handwritten digits

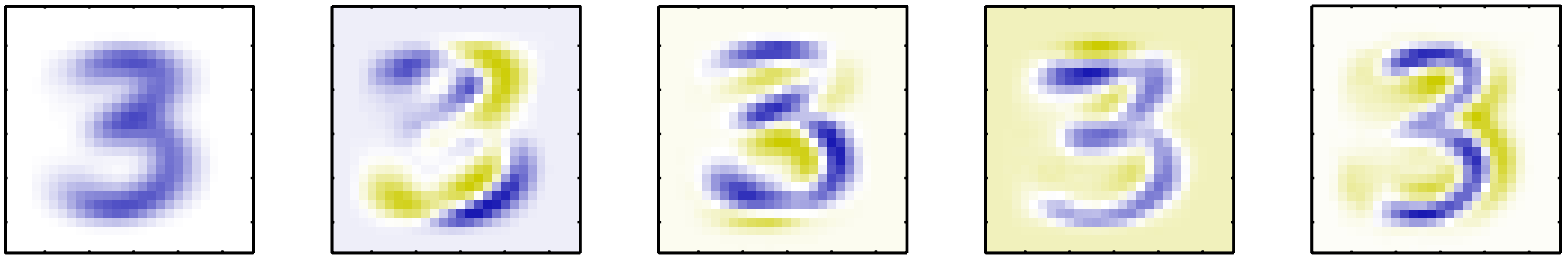
- Handwritten digits dataset
  - Digits are translated and scaled
  - Each one is contained in a box of the same size
  - Each digit is a 28 x 28 pixel 2D image (784 real numbers)
  - See MNIST handwritten digits dataset 70,000 images
    http://yann.lecun.com/exdb/mnist/



Application example – results (1/2)

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

strongest eigenvalues

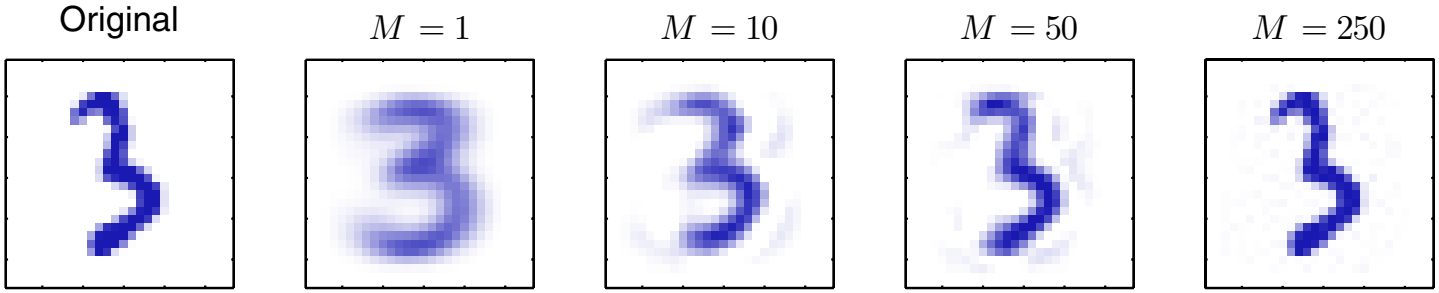


principal directions

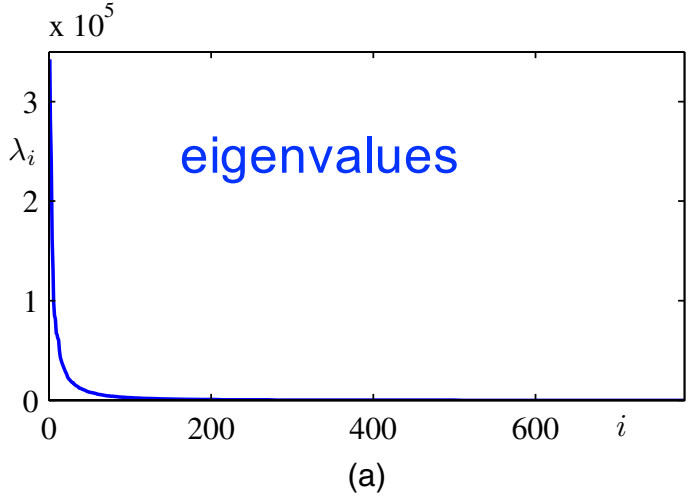
reveal important patterns in the data

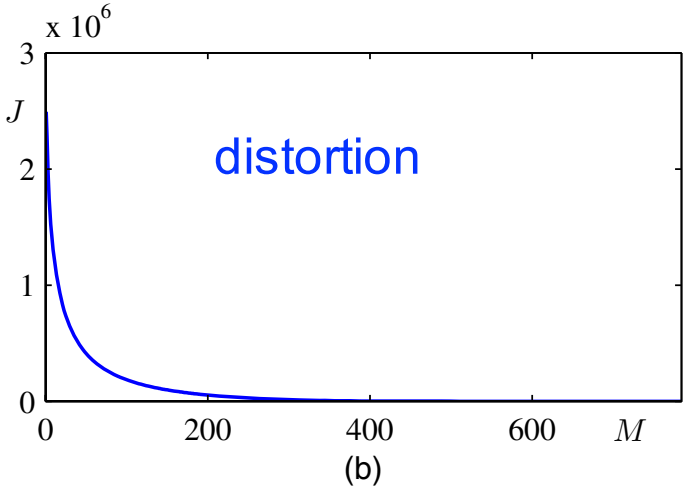
Application example – results (2/2)

- 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 $M = D = 28 \times 28 = 784$.

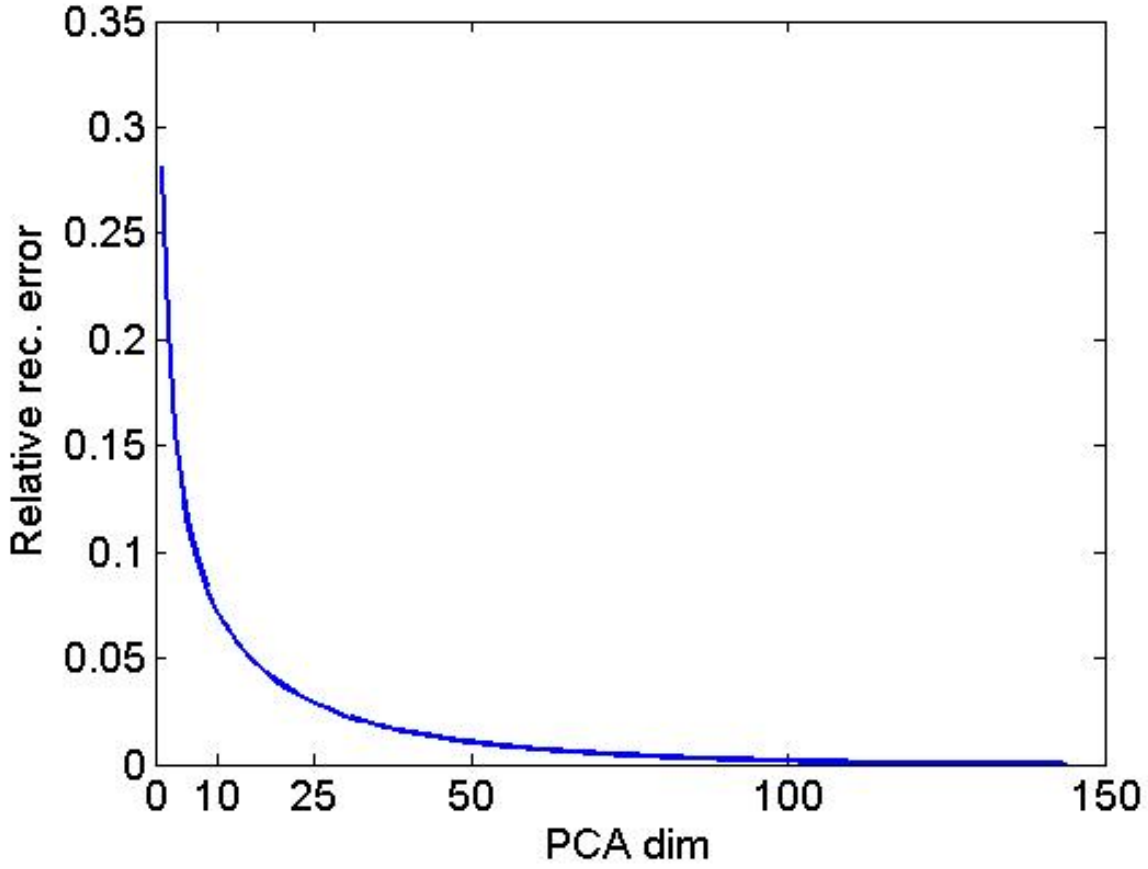




Application – image compression

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

L_2 error *vs* retained directions



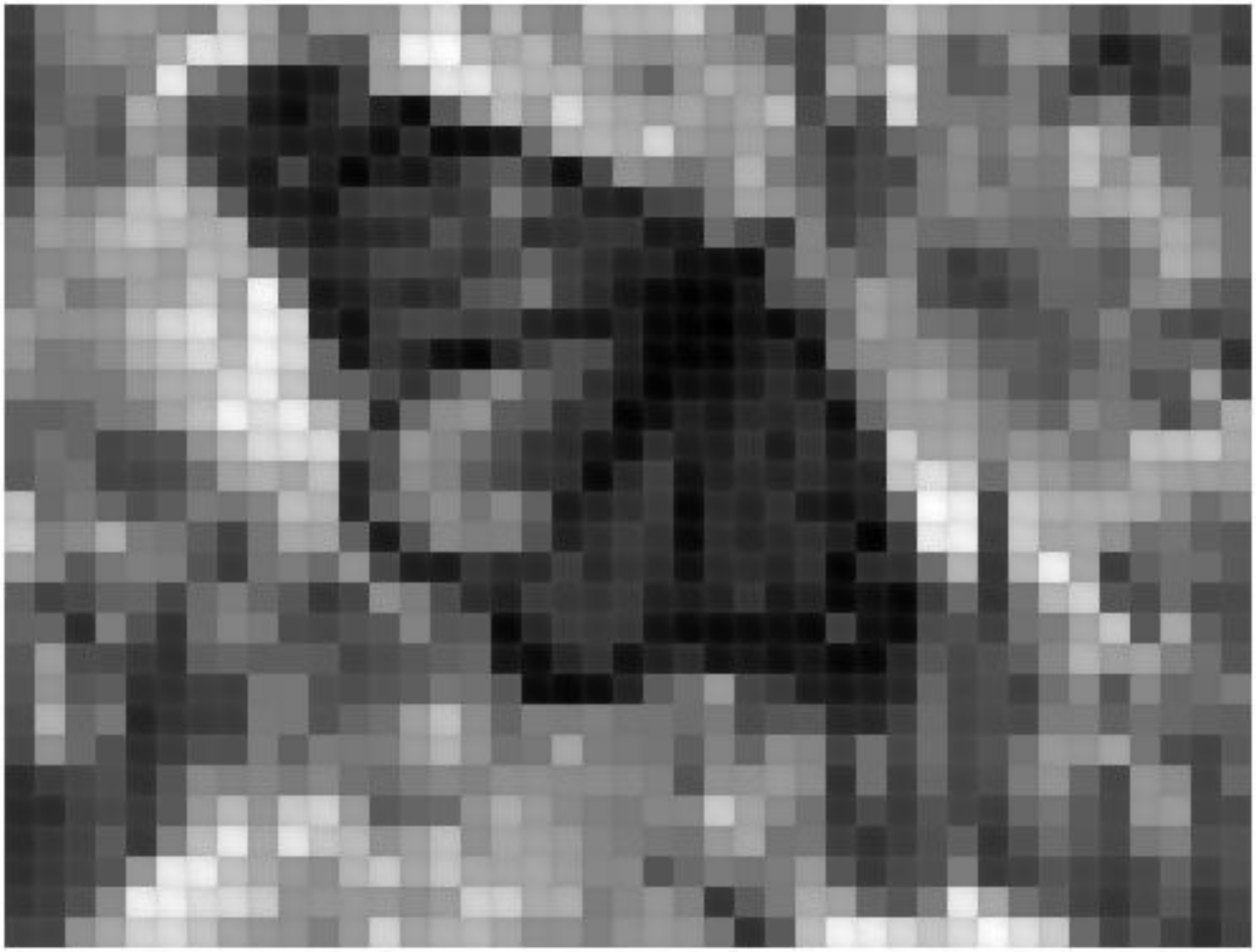
Compression 144D → 60D

Compression 144D → 16D

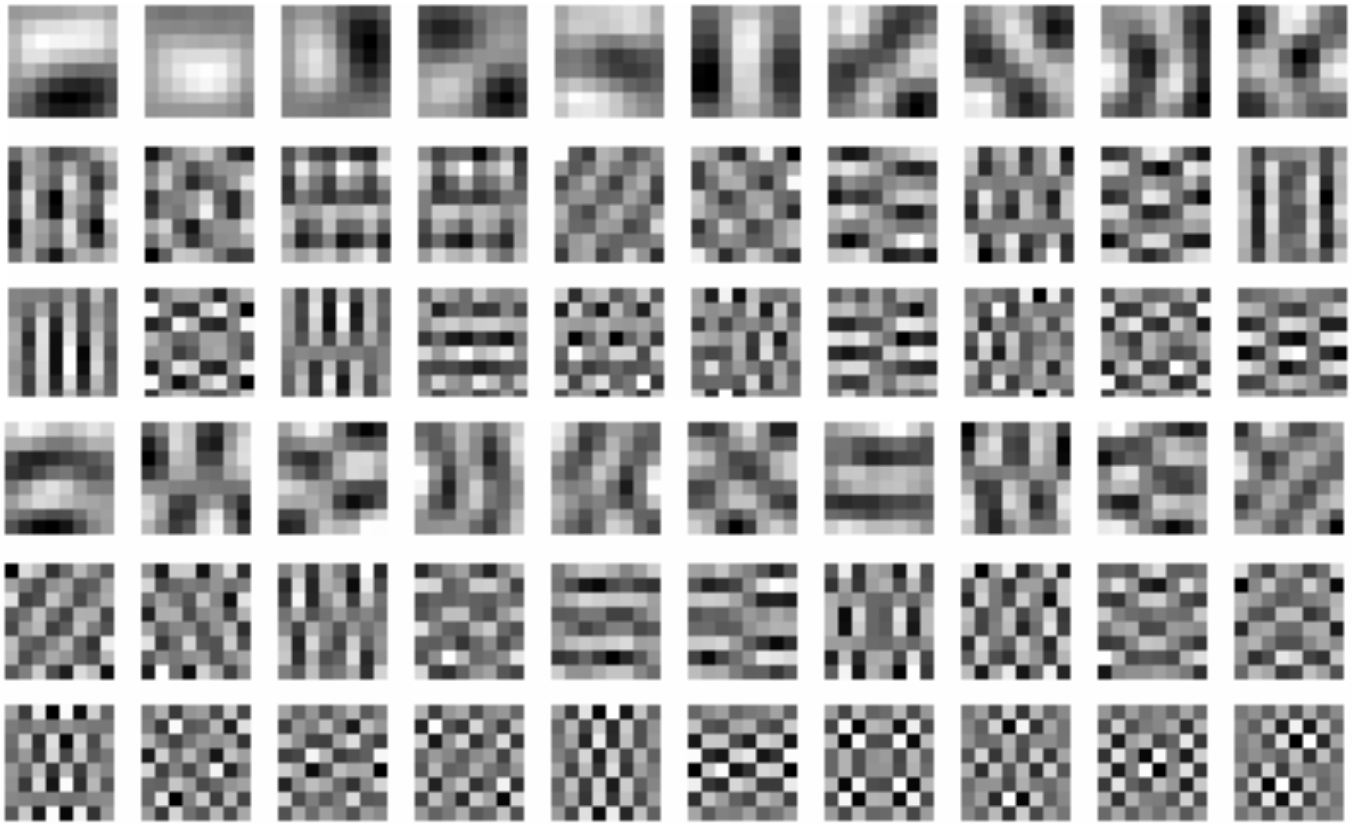
Compression 144D → 6D

Compression 144D → 3D

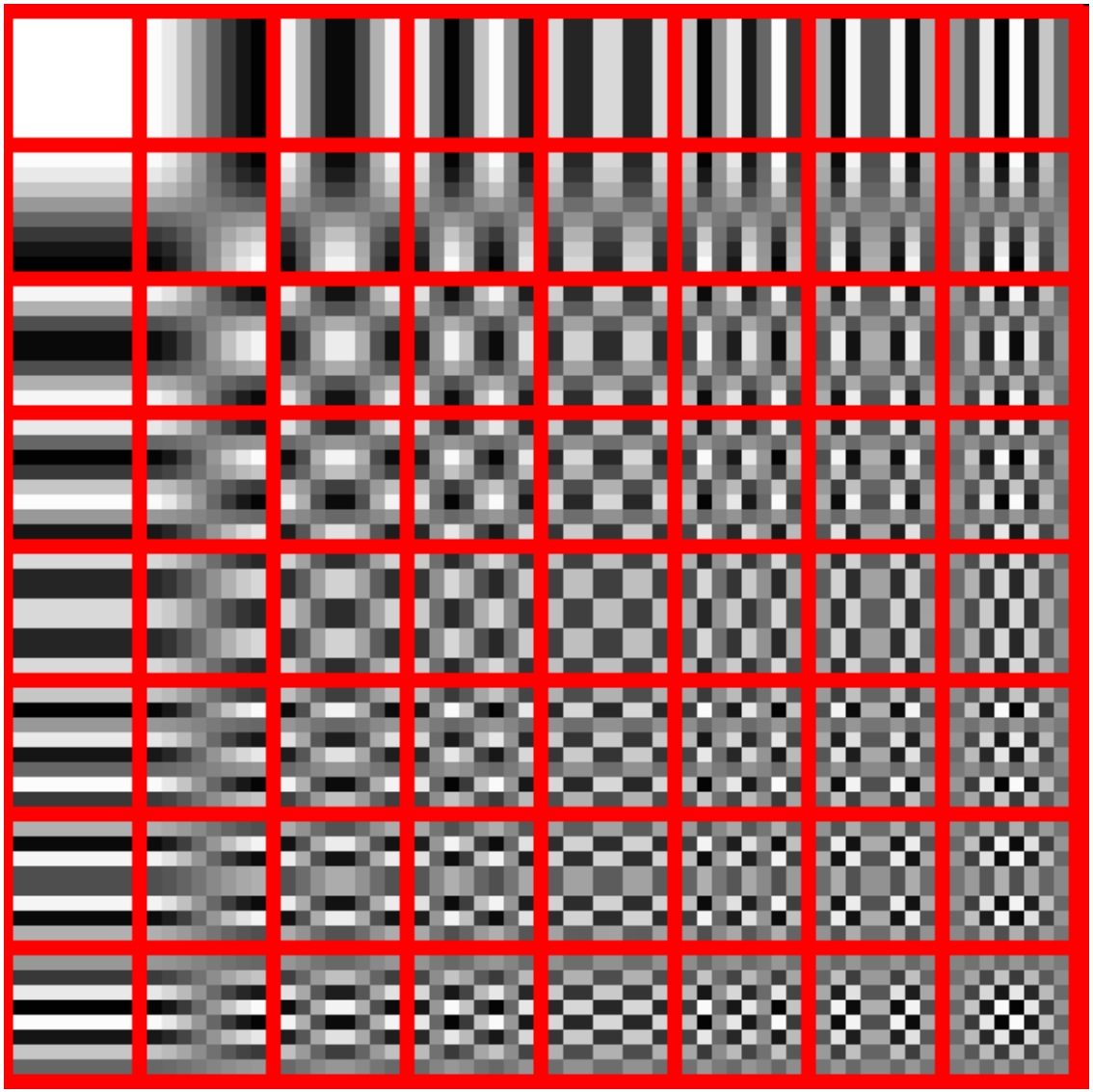
Compression 144D → 1D



60 most important eigenvectors

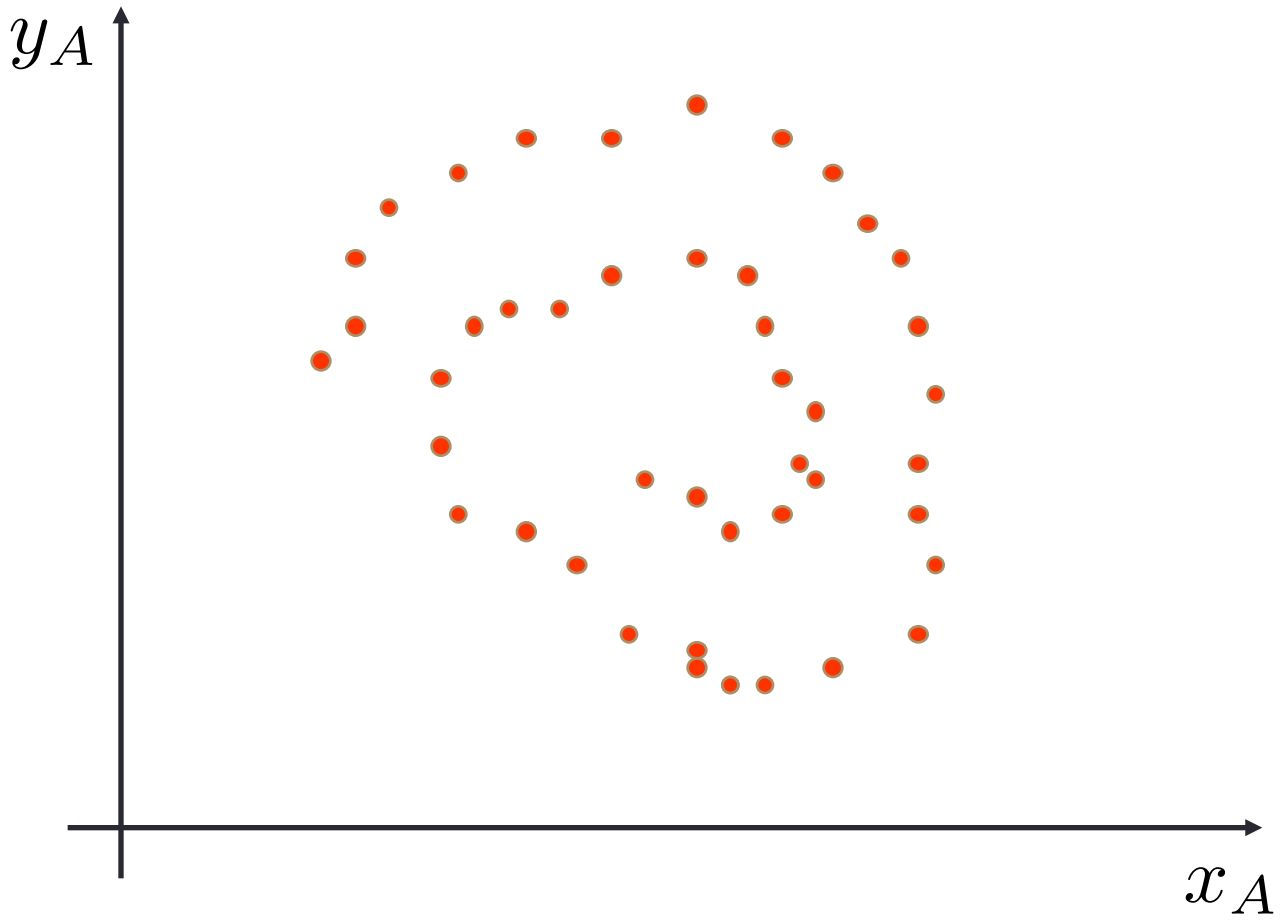


Discrete Cosine (DCT) basis for jpg



Looks pretty similar

A problematic dataset



- PCA is unable to capture nonlinear structure!!!

Where PCA fails

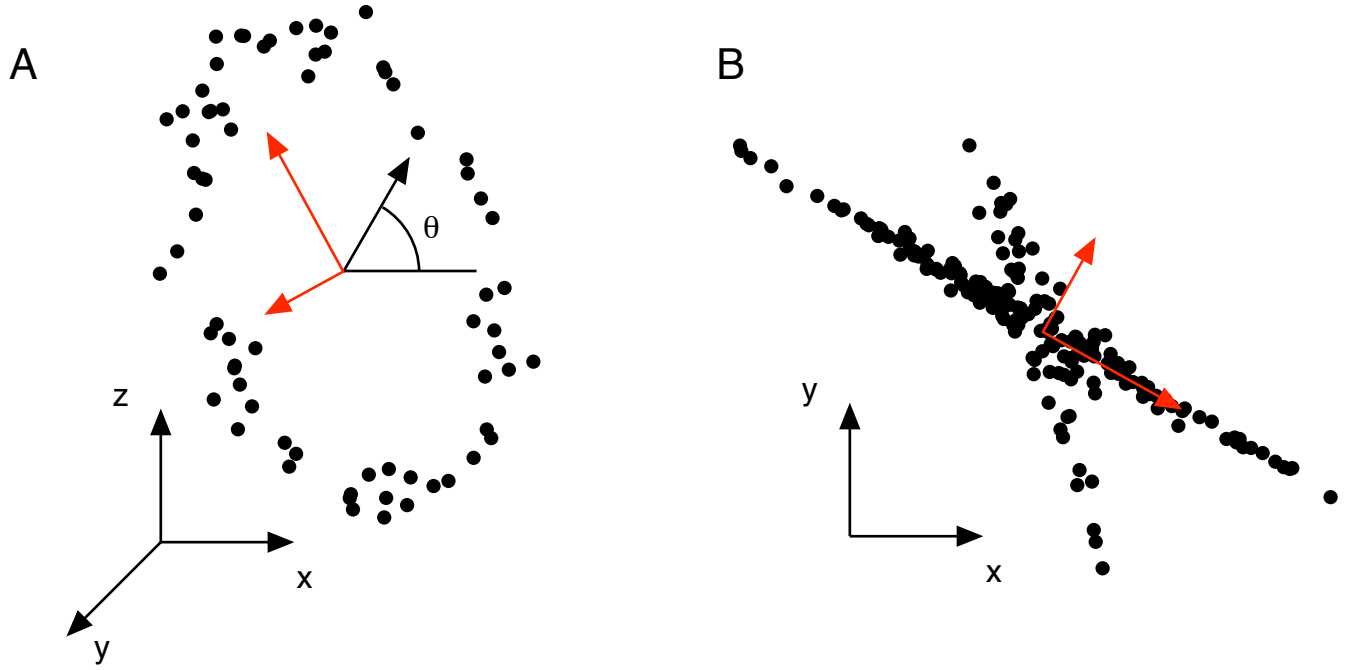


FIG. 6 Example of when PCA fails (red lines). (a) Tracking a person on a ferris wheel (black dots). All dynamics can be described by the phase of the wheel θ, a non-linear combination of the naive basis. (b) In this example data set, non-Gaussian distributed data and non-orthogonal axes causes PCA to 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 method
- PCA de-correlates a dataset (**Y** is de-correlated), this can be useful when a further classification task has to be applied on the data (aka 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.

[Bishop06] Christopher Bishop, “Pattern Recognition and Machine Learning,” Springer, 2006. Chapter 12.

[Magnus99] Jan R. Magnus, “Matrix Differential Calculus with Applications in Statistics and Econometrics,” *John Wiley & Sons Ltd*, 1999.

[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 **A** be an nxn matrix and **x** be a nx1 vector
- A quadratic form is defined as: $\alpha = \boldsymbol{x}^T \boldsymbol{A}\boldsymbol{x}$
- In a quadratic form, we may assume *without loss of generality* that **A** is symmetric, since we can always replace it with ($\mathbf{A}^T$+**A**)/2, the Proof. follows:

$$\boldsymbol{x}^T \frac{(\boldsymbol{A}+\boldsymbol{A}^T)}{2}\boldsymbol{x} = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} + \frac{1}{2}\boldsymbol{x}^T\boldsymbol{A}^T\boldsymbol{x} =$$

$$= \frac{1}{2}\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} + \frac{1}{2}(\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x})^T =$$

$\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}$ ⟷ scalar

$$= \frac{1}{2}\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} + \frac{1}{2}\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} = \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}$$

Appendix 1 – quadratic forms (2/3)

- Thus, let **A** be a symmetric matrix. We say that **A** is

$$\begin{aligned}
\text{positive definite:} &\quad \text{if } \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} > 0 \text{ for all } \boldsymbol{x} \neq 0 \\
\text{positive semi-definite:} &\quad \text{if } \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} \geq 0 \text{ for all } \boldsymbol{x} \\
\text{negative definite:} &\quad \text{if } \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} < 0 \text{ for all } \boldsymbol{x} \neq 0 \\
\text{negative semi-definite:} &\quad \text{if } \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} \leq 0 \text{ for all } \boldsymbol{x} \\
\text{indefinite:} &\quad \text{if } \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} > 0 \text{ for some } \boldsymbol{x}, \\
&\quad \text{if } \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} < 0 \text{ for some } \boldsymbol{x}
\end{aligned}$$

Appendix 1 – quadratic forms (3/3)

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

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

$$= \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{u}^T(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T\boldsymbol{u} = \boldsymbol{u}^T\left[\frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T\right]\boldsymbol{u} =$$

$$= \boldsymbol{u}^T\boldsymbol{C_X}\boldsymbol{u}$$

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

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 **A** is positive definite and write: $\boldsymbol{Ax} = \lambda \boldsymbol{x}$
- Pre-multiplying by $\mathbf{x}^T$, we get: $\boldsymbol{x}^T \boldsymbol{Ax} = \lambda \boldsymbol{x}^T \boldsymbol{x}$
- Since by assumption, $\boldsymbol{x}^T \boldsymbol{Ax} > 0\,,\ \boldsymbol{x} \neq 0$
- And likewise, it must be (quadratic norm): $\boldsymbol{x}^T \boldsymbol{x} > 0\,,\ \boldsymbol{x} \neq 0$
- 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 **A** are positive, $\lambda_i > 0$
- From the eigenvector decomposition theorem we have:

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

columns of E are eigenvectors of A

Appendix 2 – symmetric matrices (3/3)

- Proof. (continued). Recall that:
  - **E** is an orthonormal basis of eigenvectors of **A**: $\boldsymbol{E^T A E = D}$
  - Moreover, *this relation is equivalent to:*

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

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

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

  - With this, we have that:

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

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

Appendix 1 – covariance matrices

- We have learned that
  - A matrix of the form (mxm): $\boldsymbol{C_X} = \frac{1}{n}\boldsymbol{X}\boldsymbol{X}^T$
  - Is symmetric
  - Is positive semi-definite
    - Its non-zero eigenvalues are greater than zero
  - The same properties apply to (although it is size nxn)

$$\boldsymbol{X}^T\boldsymbol{X}$$

Appendix 3 – eigenvectors & eigenvalues

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

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

$$\begin{aligned}
\lambda_1 \boldsymbol{u}_1 \cdot \boldsymbol{u}_2 &= (\lambda_1 \boldsymbol{u}_1)^T \boldsymbol{u}_2 = \\
&= (\boldsymbol{A}\boldsymbol{u}_1)^T \boldsymbol{u}_2 = \\
&= \boldsymbol{u}_1^T \boldsymbol{A}^T \boldsymbol{u}_2 = \\
&= \boldsymbol{u}_1^T (\boldsymbol{A}\boldsymbol{u}_2) = \\
&= \boldsymbol{u}_1^T (\lambda_2 \boldsymbol{u}_2) = \lambda_2 \boldsymbol{u}_1 \cdot \boldsymbol{u}_2
\end{aligned}$$

- This implies that:

$$(\lambda_1 - \lambda_2)\boldsymbol{u}_1 \cdot \boldsymbol{u}_2 = 0 \Rightarrow \boldsymbol{u}_1 \cdot \boldsymbol{u}_2 = 0$$

QED

Appendix 4 – useful relations

- Theorem 6: Let **A** (mxn) and **B** (mxp) be two matrices and **x** (nx1) be a column vector. Then, the following relations hold:

$$(\mathrm{a})\ \boldsymbol{Ax} = \mathbf{0} \Leftrightarrow \boldsymbol{A}^T\boldsymbol{Ax} = \mathbf{0}$$
$$(\mathrm{b})\ \boldsymbol{AB} = \mathbf{0} \Leftrightarrow \boldsymbol{A}^T\boldsymbol{AB} = \mathbf{0}$$

- Proof. (a) Clearly $\boldsymbol{Ax} = \mathbf{0} \Rightarrow \boldsymbol{A}^T\boldsymbol{Ax} = \mathbf{0}$
- Conversely, if

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

- Then, we also have:

$$(\boldsymbol{Ax})^T(\boldsymbol{Ax}) = 0$$

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

QED

Appendix 5 – Singular Value Decomposition

- Theorem 7: Let **A** be a matrix (mxn) with rank(**A**)=r>0. Then, there exist: **(i)** an orthonormal matrix **U** (mxr), **(ii)** an orthonormal matrix **V** (nxr), and **(iii)** a diagonal matrix Σ (rxr) with positive diagonal elements, such that:

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

- where (orthonormal matrices):

$$\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}_r \qquad \boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}_r$$

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

Appendix 5 – Singular Value Decomposition

- Proof.

- Note that $\boldsymbol{A}\boldsymbol{A}^T$ is a real mxm 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:

$$\mathrm{rank}(\boldsymbol{A}) = \mathrm{rank}(\boldsymbol{A}^T) = \mathrm{rank}(\boldsymbol{A}^T\boldsymbol{A}) = \mathrm{rank}(\boldsymbol{A}\boldsymbol{A}^T)$$

Appendix 5 – Singular Value Decomposition

- Proof. (continued)
- Since $\boldsymbol{AA}^T$ is a real mxm and symmetric matrix
- From Theorem 2:

$$(\boldsymbol{AA}^T)\boldsymbol{E} = \boldsymbol{ED} = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 & 0 \\ 0 & \lambda_2 & \dots & 0 & 0 \\ 0 & 0 & \lambda_r & \dots & 0 \\ 0 & 0 & 0 & \ddots & \vdots \\ 0 & 0 & 0 & \ddots & 0 \end{bmatrix}$$

matrix $\boldsymbol{\Sigma}$

r non-zero eigenvalues of $\mathbf{AA}^T$

m-r zero eigenvalues of $\mathbf{AA}^T$

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

$$\boldsymbol{E} \triangleq [\underset{\text{mxr}}{\boldsymbol{U}} \mid \underset{\text{mx(m-r)}}{\boldsymbol{U}_*}]$$

Appendix 5 – Singular Value Decomposition

- Proof. (continued)
- Since matrix **E** (mxm) is orthonormal, it holds:

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

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

$$\boldsymbol{A}\boldsymbol{A}^T\boldsymbol{U} = \boldsymbol{U}\boldsymbol{\Sigma} \qquad (1)$$

Appendix 5 – Singular Value Decomposition

- Proof. (continued)
- (Th 2) matrix $\boldsymbol{\Sigma}$ contains the r positive eigenvalues of $\boldsymbol{AA}^T$
- Let us define a new matrix:

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

- Given these facts, we write:

using (2) — using (1)

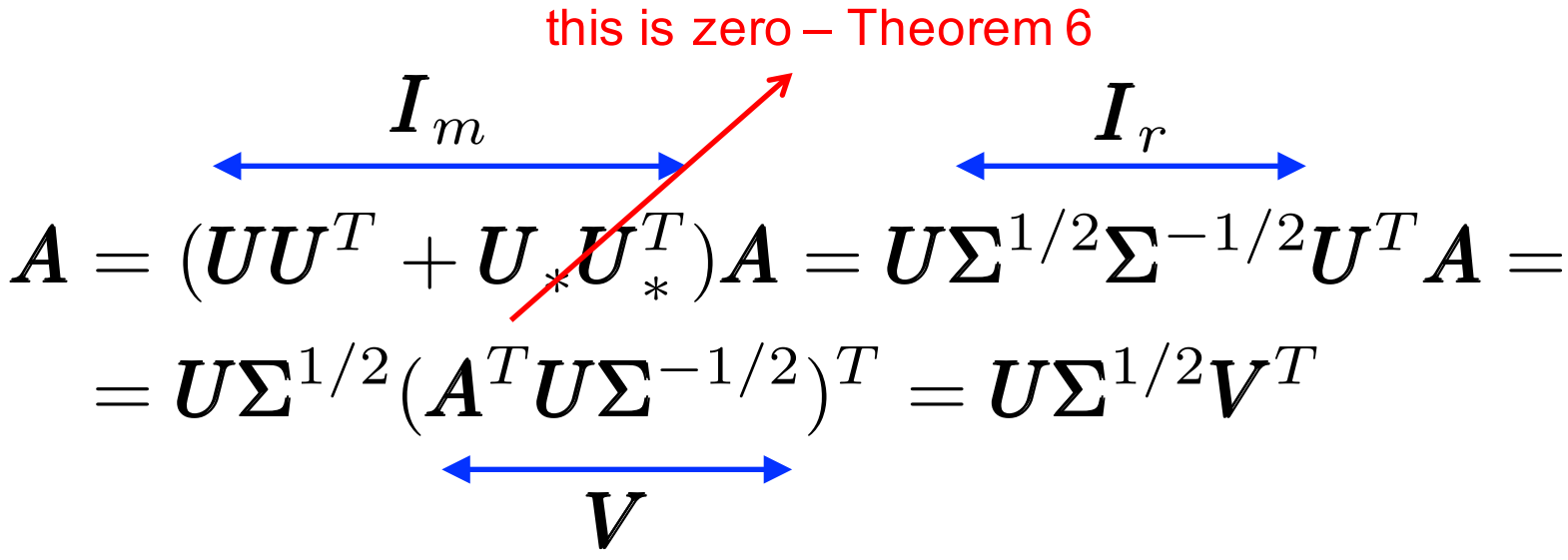
$$\boldsymbol{A}^T \boldsymbol{A} \boldsymbol{V} = \boldsymbol{A}^T \boldsymbol{A} \boldsymbol{A}^T \boldsymbol{U} \boldsymbol{\Sigma}^{-1/2} = \boldsymbol{A}^T \boldsymbol{U} \boldsymbol{\Sigma} \boldsymbol{\Sigma}^{-1/2} =$$

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

- We also have (easy to see using (1)): $\boldsymbol{V}^T \boldsymbol{V} = \boldsymbol{I}_r$

Appendix 5 – Singular Value Decomposition

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



- Hence:

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

Appendix 5 – Singular Value Decomposition

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

$$(\boldsymbol{A}\boldsymbol{A}^T)\boldsymbol{U} = \boldsymbol{U}\boldsymbol{\Sigma} \quad (3)$$

$$(\boldsymbol{A}^T\boldsymbol{A})\boldsymbol{V} = \boldsymbol{V}\boldsymbol{\Sigma} \quad (4)$$

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

QED

PCA *vs* SVD (1/3)

- They are intimately related
- Let **X** (mxn) be our data matrix (mean has been removed)
- Define a new matrix **Z** as:

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

- Hence, we have that:

$$\boldsymbol{Z}\boldsymbol{Z}^T = \frac{1}{n}\boldsymbol{X}\boldsymbol{X}^T = \boldsymbol{C}_{\boldsymbol{X}}$$

- Note that $\mathbf{ZZ}^T$ is the *covariance matrix* of our data and corresponds to $\mathbf{AA}^T$ (taking **Z**=**A**) in the SVD Theorem 7

PCA *vs* SVD (2/3)

- Applying SVD to $\mathbf{ZZ}^T$:
  - The columns of **U** contains the eigenvectors associated with the non-zero eigenvalues of $\mathbf{ZZ}^T=\mathbf{C_X}$
  - These columns form an orthonormal basis
  - The columns of **U** are the principal directions (components) of $\mathbf{C_X}$

- Matrix $[\boldsymbol{U} \mid \boldsymbol{U}_*]$ corresponds to matrix **E** in Theorem 2
  - We know that the columns of matrix **E** are the eigenvectors of the covariance matrix $\mathbf{C_X}$ (all eigenvalues)
  - From the SVD Proof. we also know: $$\boldsymbol{Z}^T\boldsymbol{U}_* = (\boldsymbol{U}_*^T\boldsymbol{Z})^T = \mathbf{0}$$
  - This means that $\boldsymbol{U}_*$ does not contribute to relevant (non-zero) elements in the transformed PCA space

PCA *vs* SVD (3/3)

1) Organize data vectors into a matrix **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: $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}^{1/2}\boldsymbol{V}^{T}$

   - columns of **U** are the eigenvectors of the r positive eigenvalues of $\mathbf{ZZ}^{T}=\mathbf{C}_{X}$ (we have, rank($\mathbf{C}_{X}$=r)). From the previous equation, pre-multiplying both sides by $\mathbf{U}^{T}$, we get:

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

   - 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 **Y** contains r rows and n columns

Summing up

The transformed data points after PCA are:

$$\boldsymbol{U}^T\boldsymbol{X} = \boldsymbol{Y}$$

where:

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

- 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*)

PRINCIPAL COMPONENT ANALYSIS (PCA)

Michele Rossi

rossi@dei.unipd.it

Dept. of Information Engineering

University of Padova, IT