SIMILARITY MEASURES AND THE DIMENSIONALITY CURSE

Claudio Silvestri

In the previous episode …

- Content-Based recommendation
- Collaborative Filtering
- Item-based Collaborative Filtering

- “Find something similar to this” problem:
  - Each has its own similarity function

Minkowski distances

- Given two ***N***-dimensional objects X and Y, where X = [x_0, …, x_{N-1}] and Y = [y_0, …, y_{N-1}]

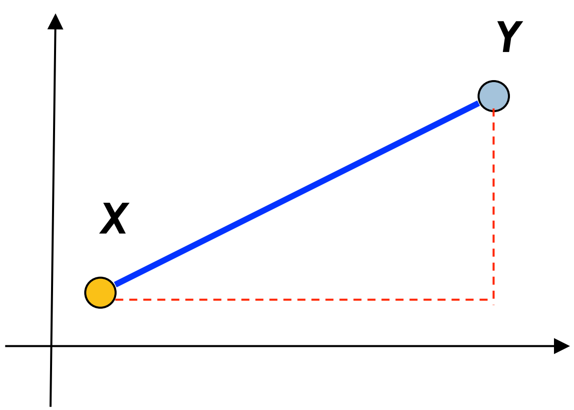
$$
\begin{aligned}
d(X,Y) &= \sqrt[q]{|x_0 - y_0|^q + |x_1 - y_1|^q + \ldots + |x_{N-1} - y_{N-1}|^q} \\
&= \sqrt[q]{\sum_{0 \leq i < N} |x_i - y_i|^q}
\end{aligned}
$$

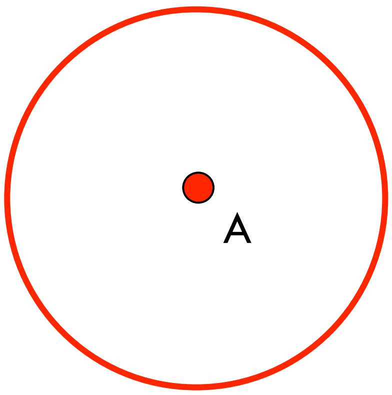
Euclidean Distance

- If *q=2*, *L2* norm or ***Euclidean Distance***:

$$d(X,Y) = \sqrt[2]{|x_0 - y_0|^2 + |x_1 - y_1|^2 + \ldots + |x_{N-1} - y_{N-1}|^2}$$

- It defines a metric space:
  - d(X,Y) >= 0 *(Positivity)*
  - d(X,Y) = d(Y,X) *(Symmetry)*
  - d(X,Y) <= d(X,Z) + d(Z,Y) *(Triangular inequality)*



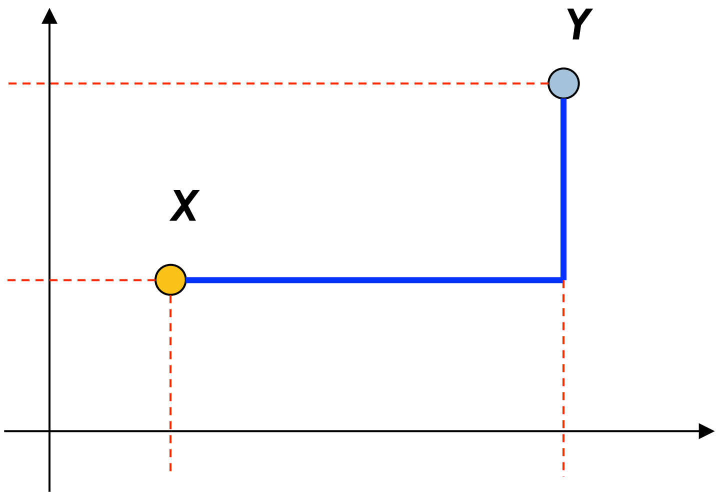


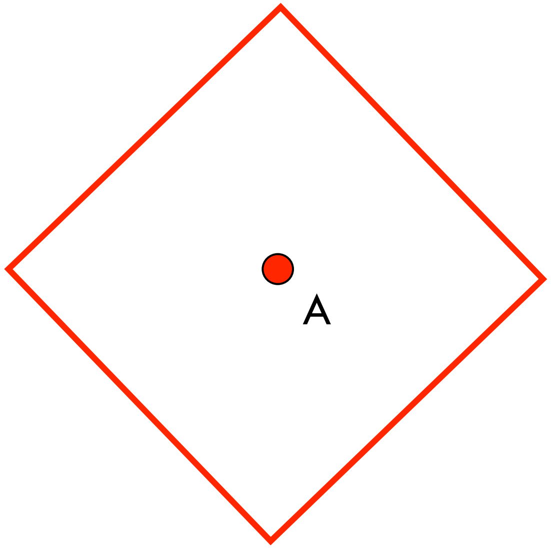
punti equidistanti da A

Manhattan Distance

- If *q=1*, *L1* norm or ***Manhattan*** or ***City-Block***:

$$d(X,Y) = |x_0 - y_0| + |x_1 - y_1| + \ldots + |x_{N-1} - y_{N-1}|$$



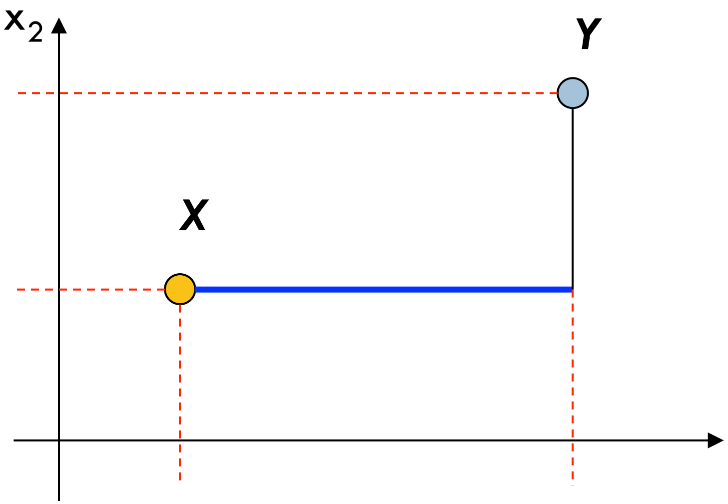


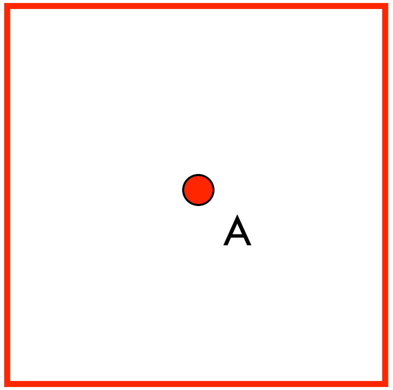
punti equidistanti da A

Chebyshev Distance

- If $q \to \infty$, L_∞ norm or ***Chebyshev***, ***Chessboard Distance***:

$$d(X,Y) = \max_i |x_i - y_i|$$





punti equidistanti da A

Chessboard Distance

Editing distance

- Sequences of chars
- Similarity: minimum number of insert/delete to transform one sequence into the other

Earth Mover's Distance

- A.k.a: Wasserstein metric
- (Usually discrete) probability distributions
- Distributions compared to a irregular lots of earth
- Distance: Minimum amount of earth to move to transform one lot into the other

- Optimal transportation problem. Easy to solve in one dimension.

Binary Vectors

- Document X = [0,1,0,1,0,1,0,1,1,1]
- Document Y = [1,0,1,1,1,0,1,0,1,1]
- *X[i] = 1 iff* the *i*-th term occurs in *X*
- Contingency table:

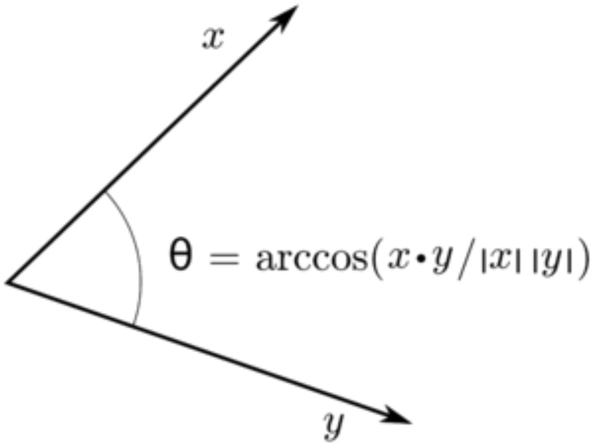
		X = 1	X = 0	sum
Y	1	q	r	$q+r$
	0	s	t	$s+t$
	sum	$q+s$	$r+t$	p

- *Simple matching*: (q+t)/p
- *Jaccard*: q/(q+r+s) or X ∩ Y/(X ∪ Y)

Cosine Similarity

- Document X = [0,0,0,3,0,5,0,14,7,9]
- Document Y = [1,0,2,2,4,0,10,0,3,11]
- *X[i]* is the number of time the *i*-th term occurs in X

$$\cos(X,Y) = \frac{X \cdot Y}{|X|\ |Y|} = \frac{\sum x_i y_i}{\sqrt{\sum x_i^2}\sqrt{\sum y_i^2}}$$



Cosine Similarity

- Example:
  - X = 3 2 0 5 0 0 0 2 0 0
  - Y = 1 0 0 0 0 0 0 1 0 2
  - X • Y = 3*1 + 2*0 + 0*0 + 5*0 + 0*0 + 0*0 + 0*0 + 2*1 + 0*0 + 0*2 = 5
  - |X| = (3*3+2*2+0*0+5*5+0*0+0*0+0*0+2*2+0*0+0*0)$^{0.5}$ = (42)$^{0.5}$ = 6.481
  - |Y| = (1*1+0*0+0*0+0*0+0*0+0*0+0*0+1*1+0*0+2*2)$^{0.5}$ = (6)$^{0.5}$ = 2.245
  - cos(X, Y) = 0.344

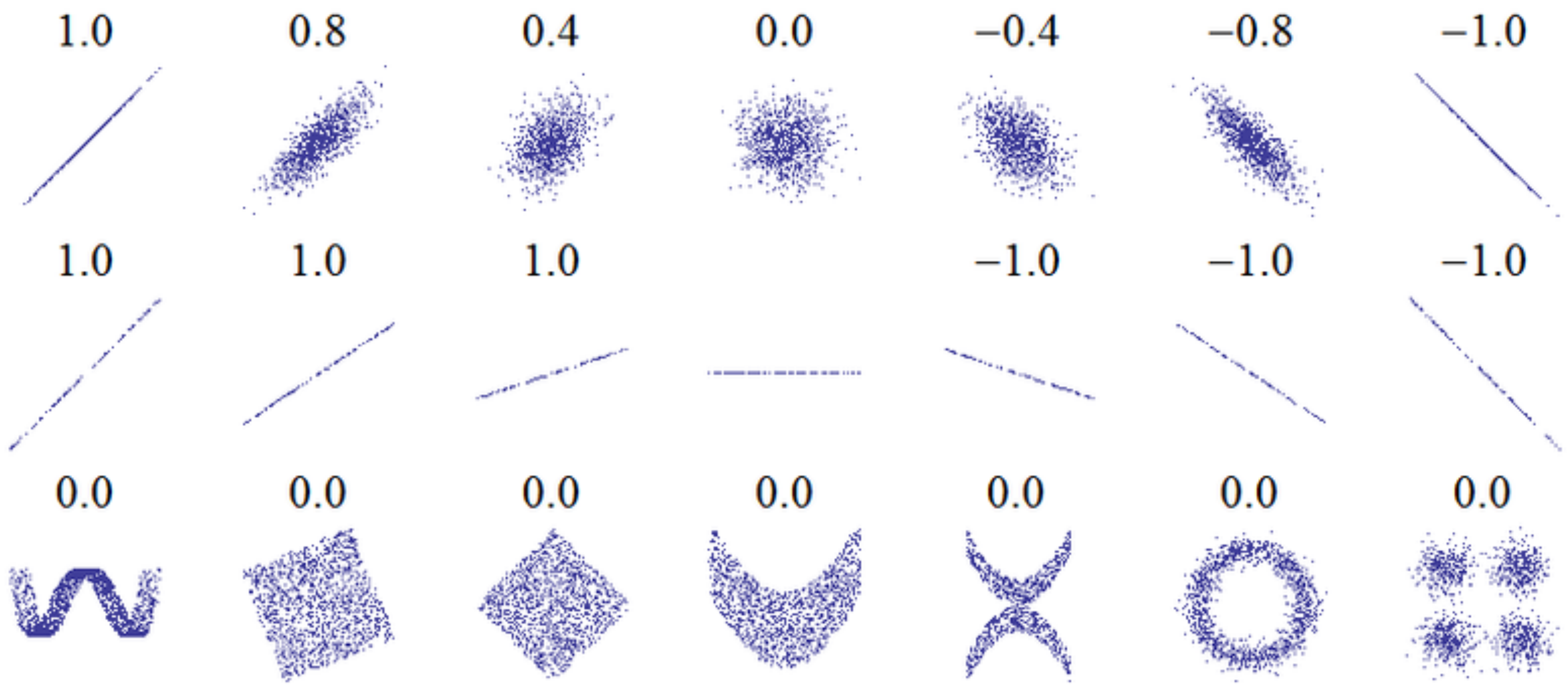
Pearson Correlation

- Linear dependency between variables
  - Does X increase when Y increases ?
  - Is there any correlation between income and degree ?
- Standardize and multiply:

$$X_i = \frac{x_i - \overline{X}}{\sqrt{\sum (x_i - \overline{X})^2}} \qquad Y_i = \frac{y_i - \overline{Y}}{\sqrt{\sum (y_i - \overline{Y})^2}}$$

$$\rho(X, Y) = \sum X_i Y_i$$

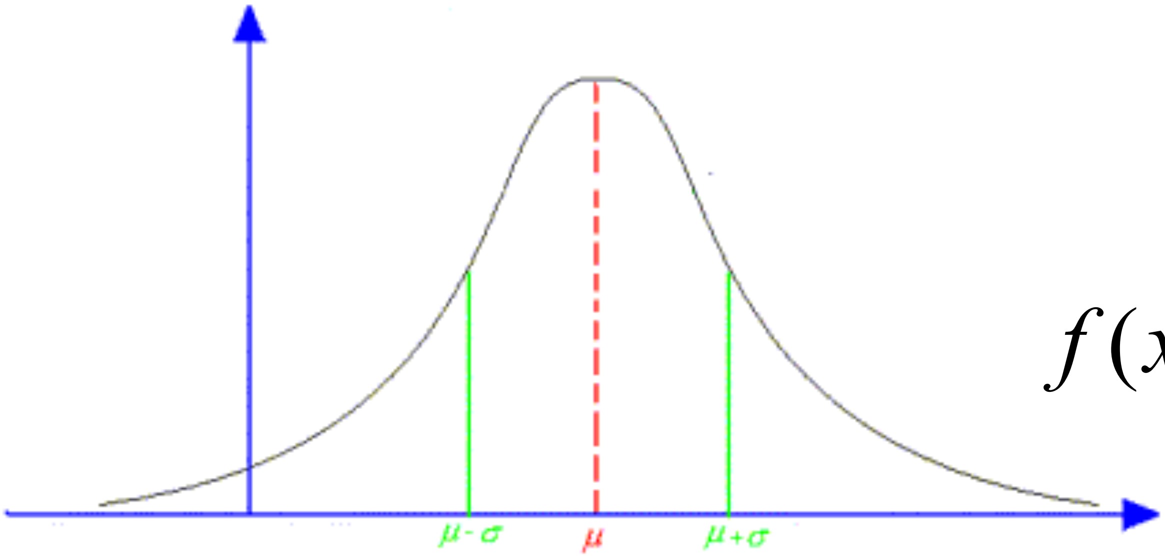
Correlation plots



Standardization ?

- Most data has Gaussian Distribution:
  - Average height of people
  - Number of head in coins flipping
  - [Central Limit Theorem]

$$X_i = \frac{x_i - \overline{X}}{\sqrt{\sum (x_i - \overline{X})^2}}$$

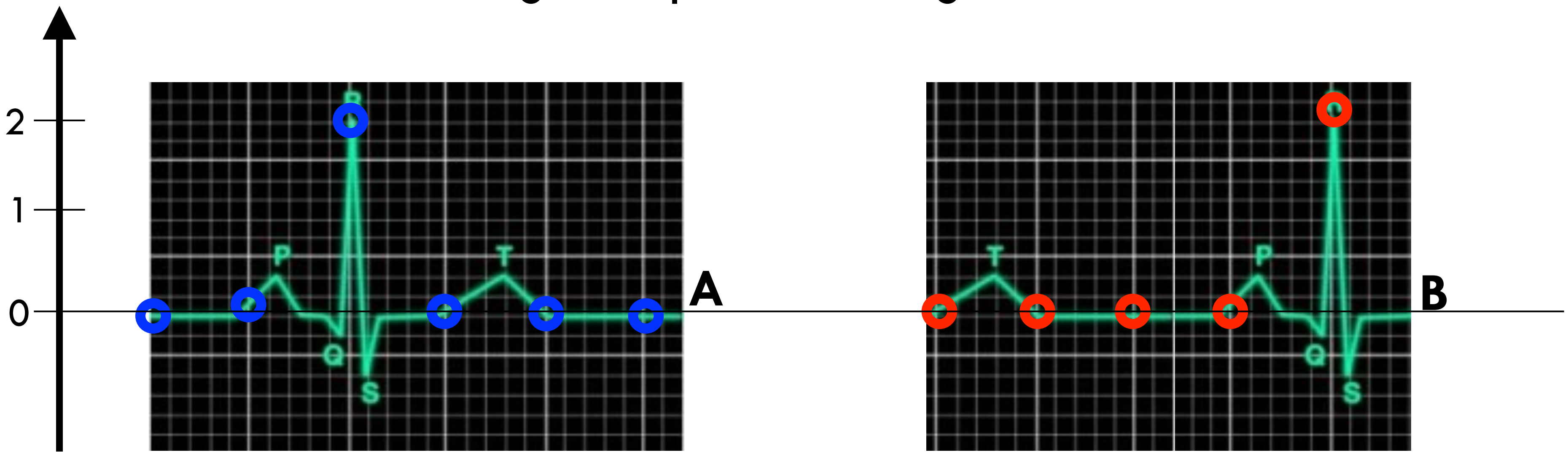


$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

- **68.2**% of points is at distance at most **one** standard deviation from the mean
- **95,5**% of points is at distance at most **two** standard deviation from the mean

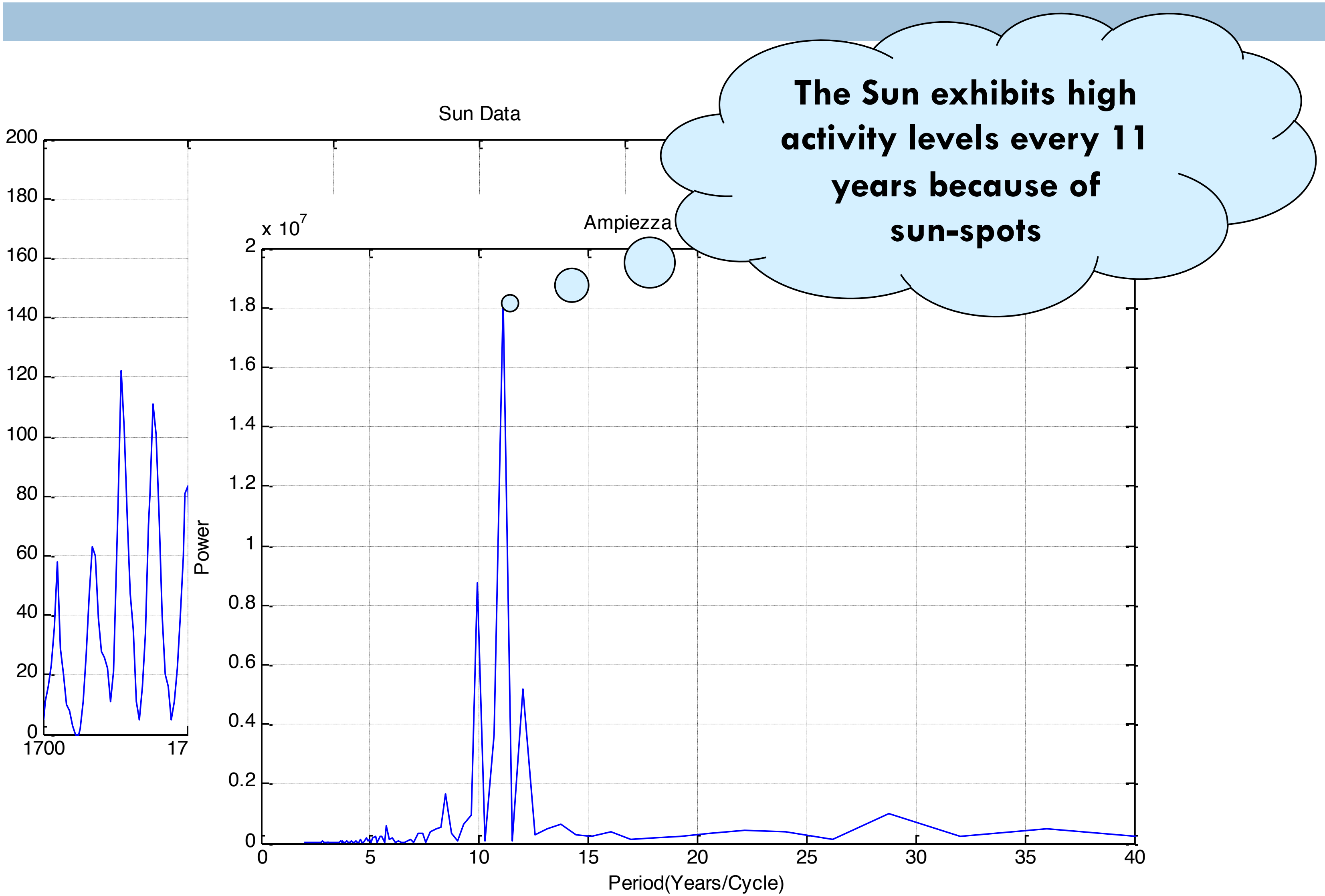
Similarity of time-series

- Examples:
  - ECG, stocks, temperatures, stars luminosity, ecc.
- Euclidean Distance ?
  - Not robust against phase changes



$$d(A,B) = \sqrt{(0-0)^2 + (0-0)^2 + (2-0)^2 + (0-0)^2 + (0-2)^2} = \sqrt{4+4} = 2.82$$

Periodic Distance



Periodic Distance

- Fourier Transform:
  - "Understands" the important ***frequencies*** in a signal, in terms of ***Amplitude*** and ***Phase***
    - AX = [100, 80, 70, 10, 0, 0, 0]
    - AY = [99, 80, 50, 20, 10, 0, 0]

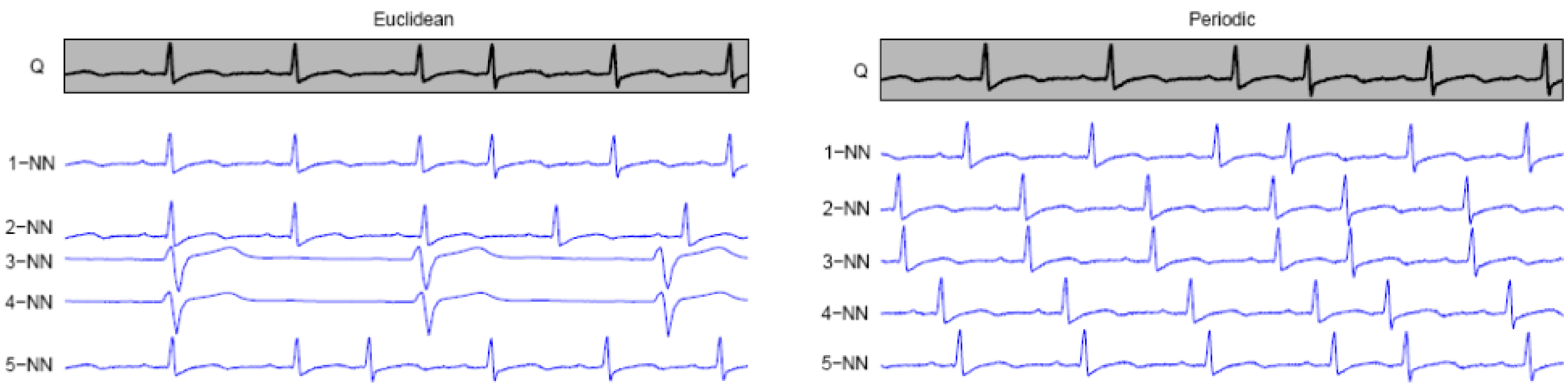


Fig. 1. 5-NN euclidean and periodic matches on an ECG dataset.

FFT demo

- FFT is an algorithm to compute DFT
- http://www.falstad.com/fourier/

The curse of dimensionality

- Originally used to address optimization problems:

- Suppose you want to find the optimum value of $x \in \{1,2,3,4,5,6,7,8,9,10\}$.
  - Try every value and check the function to optimize.
- Suppose you have two variables $x,y \in \{1,2,3,4,5,6,7,8,9,10\}$.
  - You may need to try *100* cases:
    - x=1 & y=1, x=1 & y=2, x=1 & y=3, etc. etc.
- Suppose you have *n* such variables, the search space grows up to 10^n.

- Problems are considered intractable starting from $n=10$.

Not only optimization problems

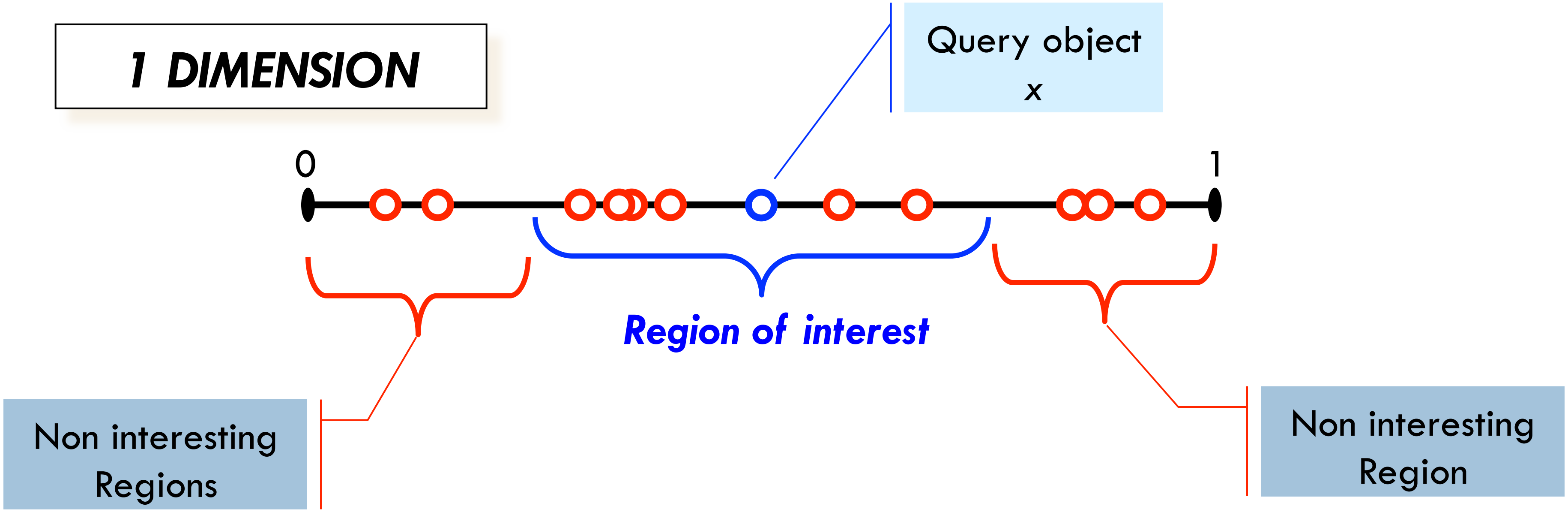
- Anytime you have objects with a large number of attributes (variables)
- In our case:
  - Objects are documents
  - Variables are term occurrence counts
  - Minimize similarity

Similarity

- Suppose objects are identically independently distributed at random in the (search) space
- Every dimension has values in the interval [0, 1]
- Find Objects at distance <0.25 from x.

Similarity

- Suppose objects are identically independently distributed at random in the (search) space
- Every dimension has values in the interval [0, 1]
- Find Objects at distance <*0.25* from *x*.

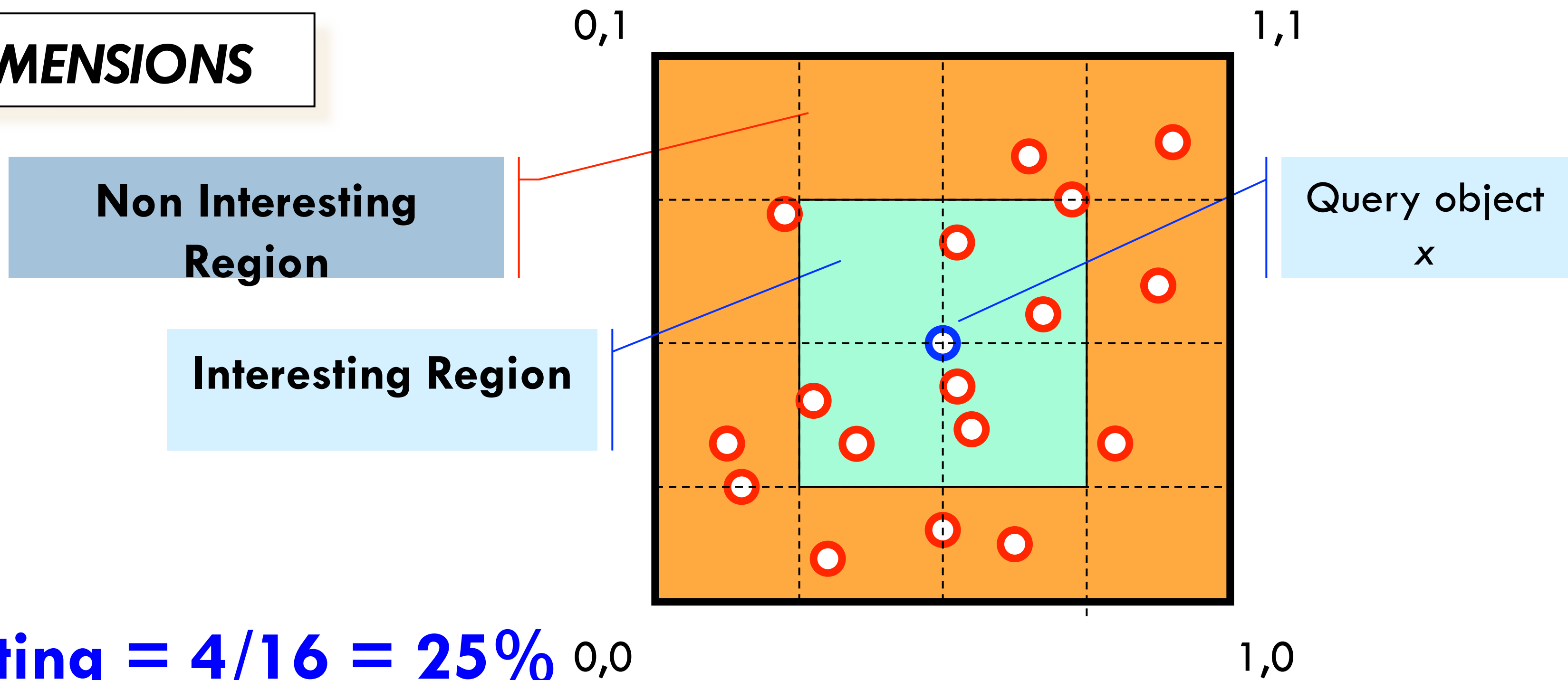


Interesting space = ½ = 50%

Similarity

- Suppose objects are identically independently distributed at random in the (search) space
- Every dimension has values in the interval [0, 1]
- Find Objects at distance <0.25 from x.

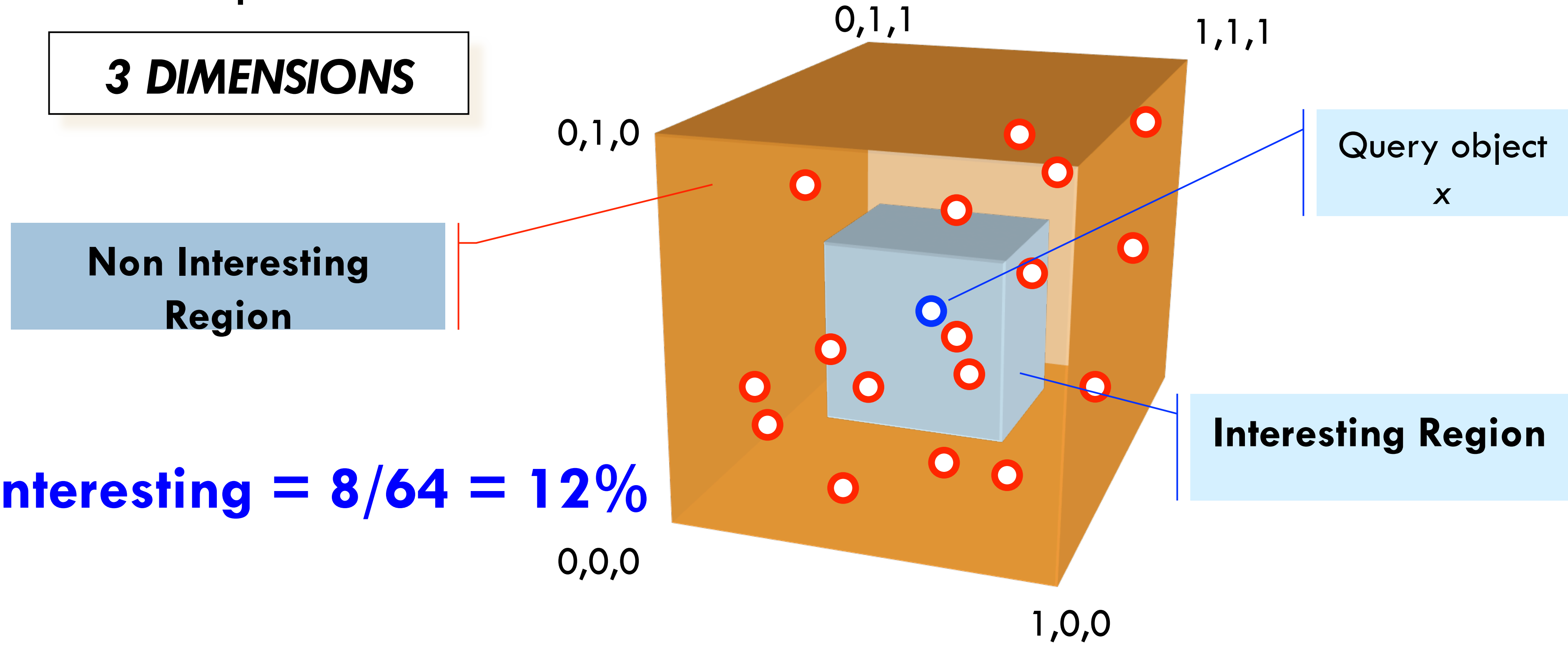
2 DIMENSIONS



interesting = 4/16 = 25%

Similarity

- Suppose objects are identically independently distributed at random in the (search) space
- Every dimension has values in the interval [0, 1]
- Find Objects at distance <*0.25* from *x*.



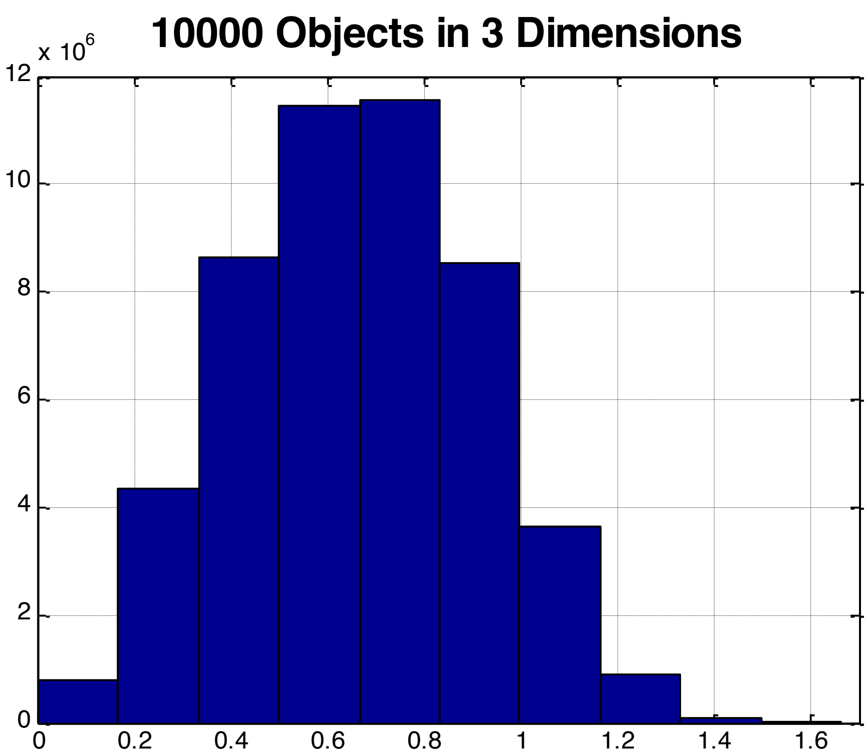
interesting = 8/64 = 12%

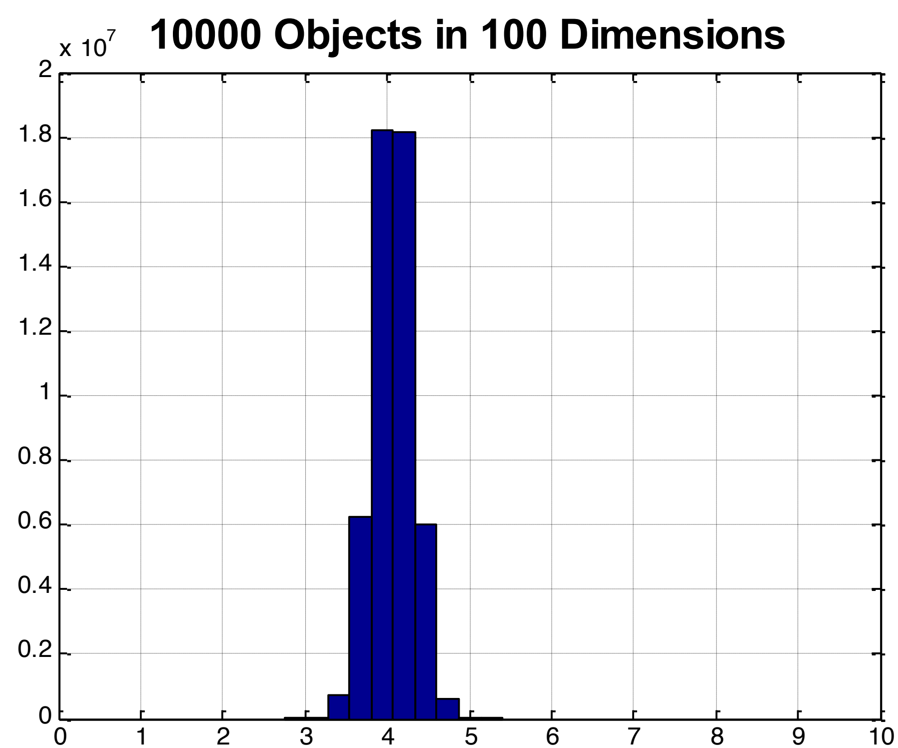
What does it mean ?

- The region of interest halves when increasing the number of dimensions
  - 50%, 25%, 12.5%, …
- Consequently, the number of interesting objects gets smaller and smaller
- For large values of *n* there will be no results
- You need to significantly increase the search radius to get some objects, but, you'll likely get everything !
- Anything is similar or un-similar to anything

Curse of Dimentionality

- Everything is at the same distance.

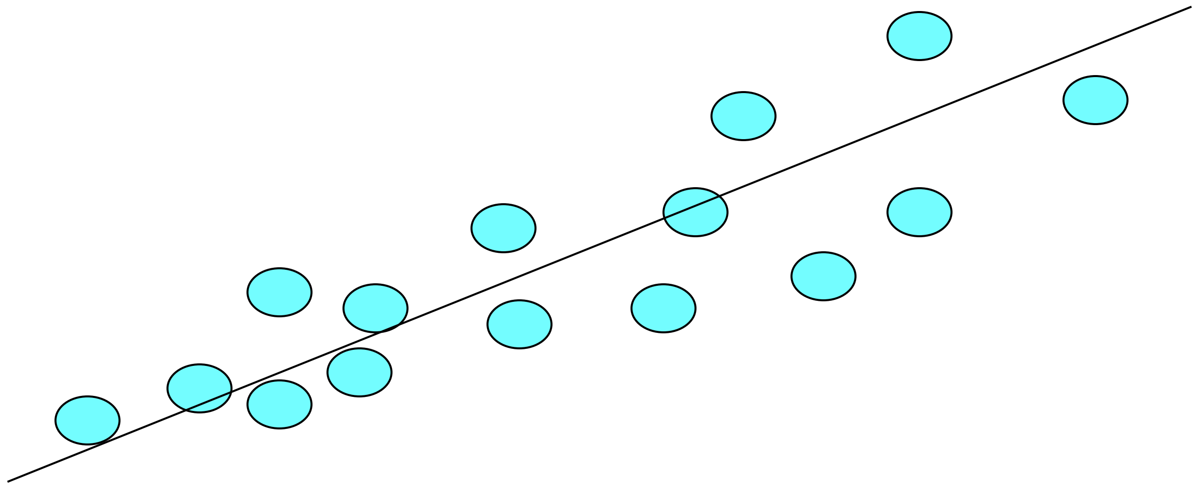




- How to overcome the dimensionality curse ?
- Try to understand what is useful,
  and what is not !
- Dimensionality reduction !

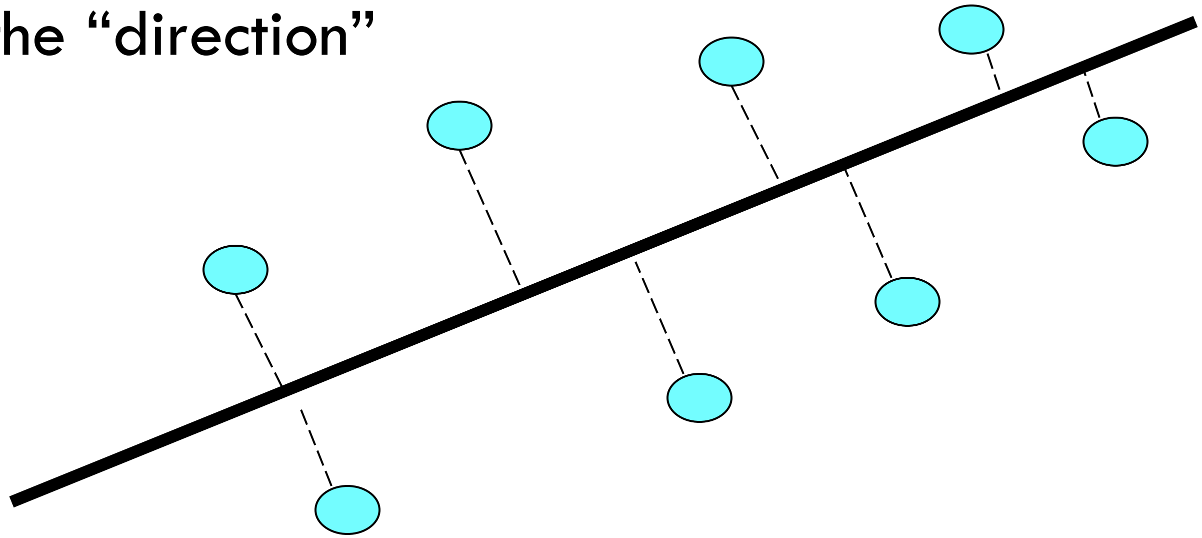
Principal Component Analysis (1901)

- Find the “main trend” in the data



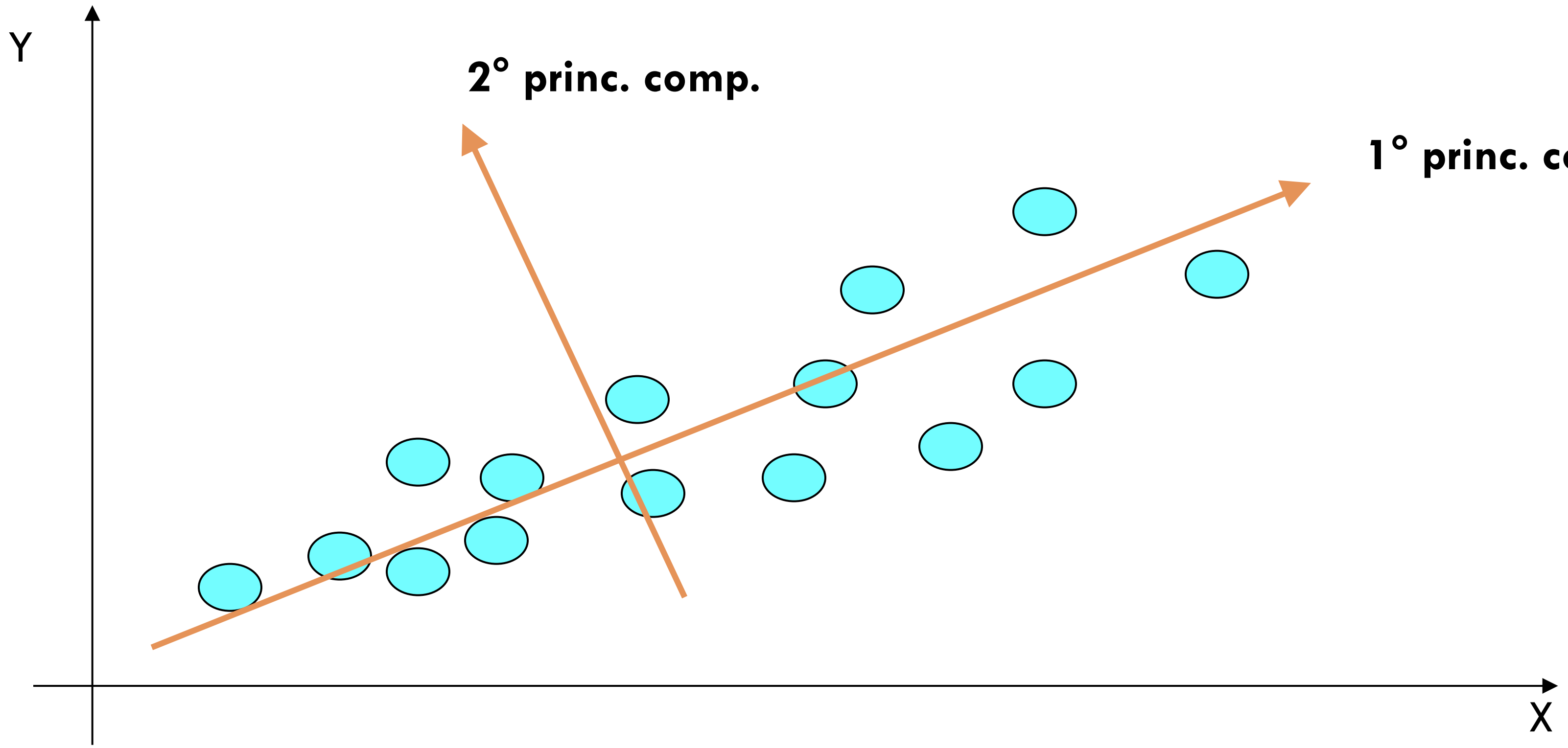
Principal Component Analysis

- Minimize the Mean Squared Error:
  - That is the (squared) distances of the given points from the "direction"



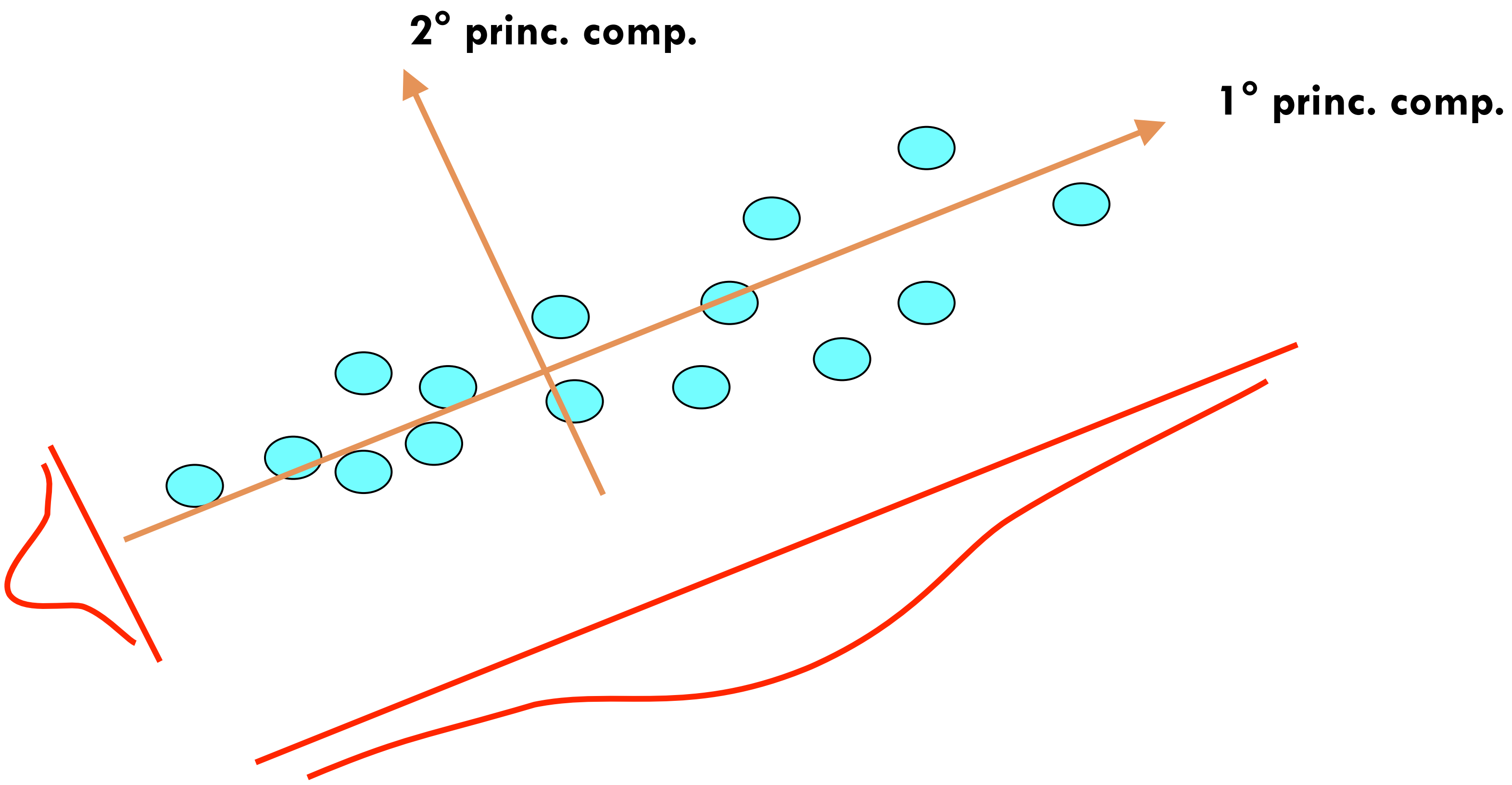
Principal Component Analysis

- In general it finds as many directions as the number of dimensions
  - They must be orthogonal



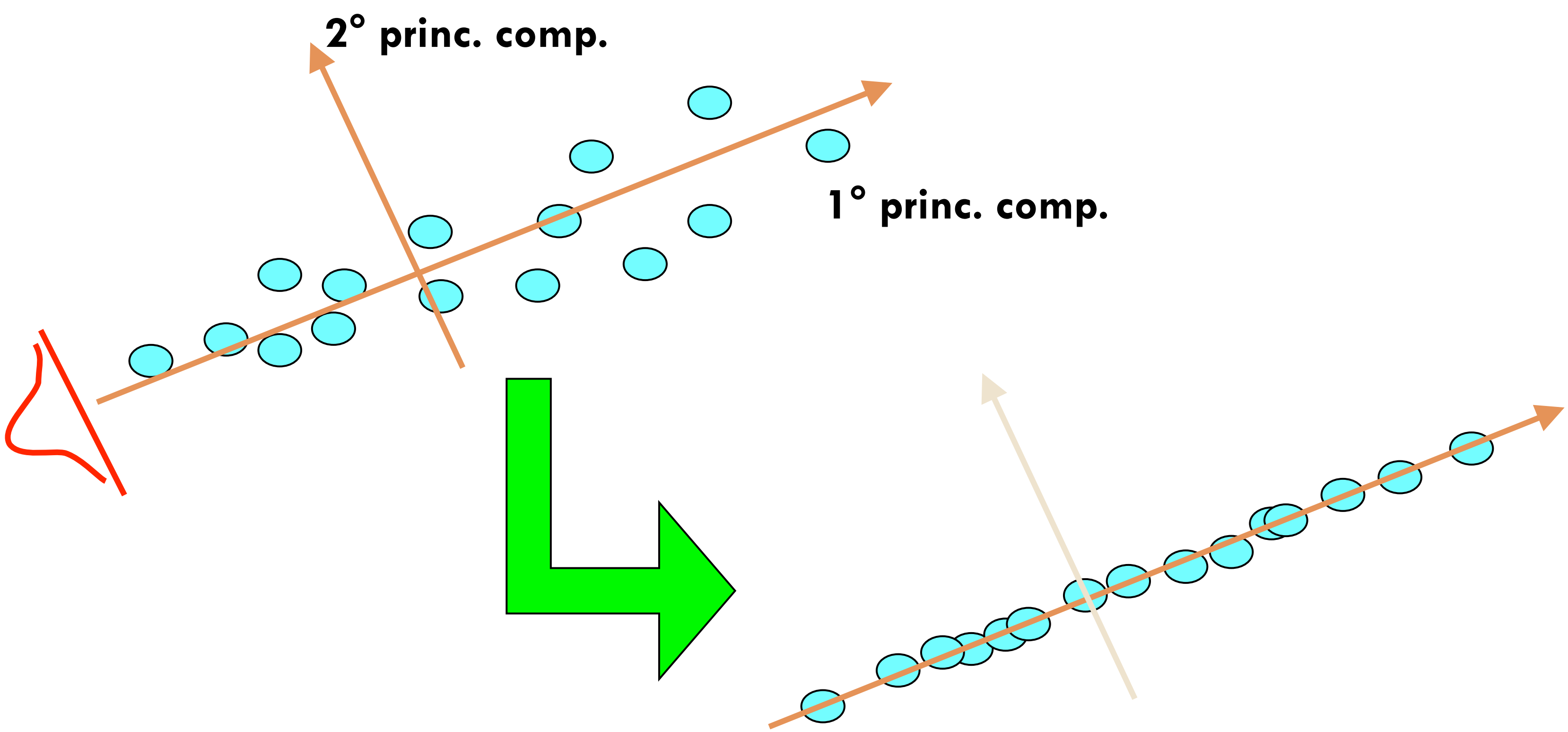
Principal Component Analysis

- It finds the direction of maximum variance



PCA e dimensionality reduction

- We can ignore lest significant components



One dimension / one coordinate

In matrices …

- Given a data set X
  - A column is an object, each row is a coordinate
- Compute covariance matrix $X \cdot X^T$
- Compute eigenvectors of $X \cdot X^T$
  - e is an eigenvector if and only if $X \cdot X^T \cdot e = \sigma \cdot e$
    - σ is an eigenvalue

- Let E be a matrix such that its rows are the eigenvectors

- Create a new dataset by removing least significant eigenvectors
  - $X^{PCA} = E^k X$
    where E^k has only the first k rows of E (eigenvectors of $X \cdot X^T$)

Singular Value Decomposition

- Any matrix X can be expressed as the product

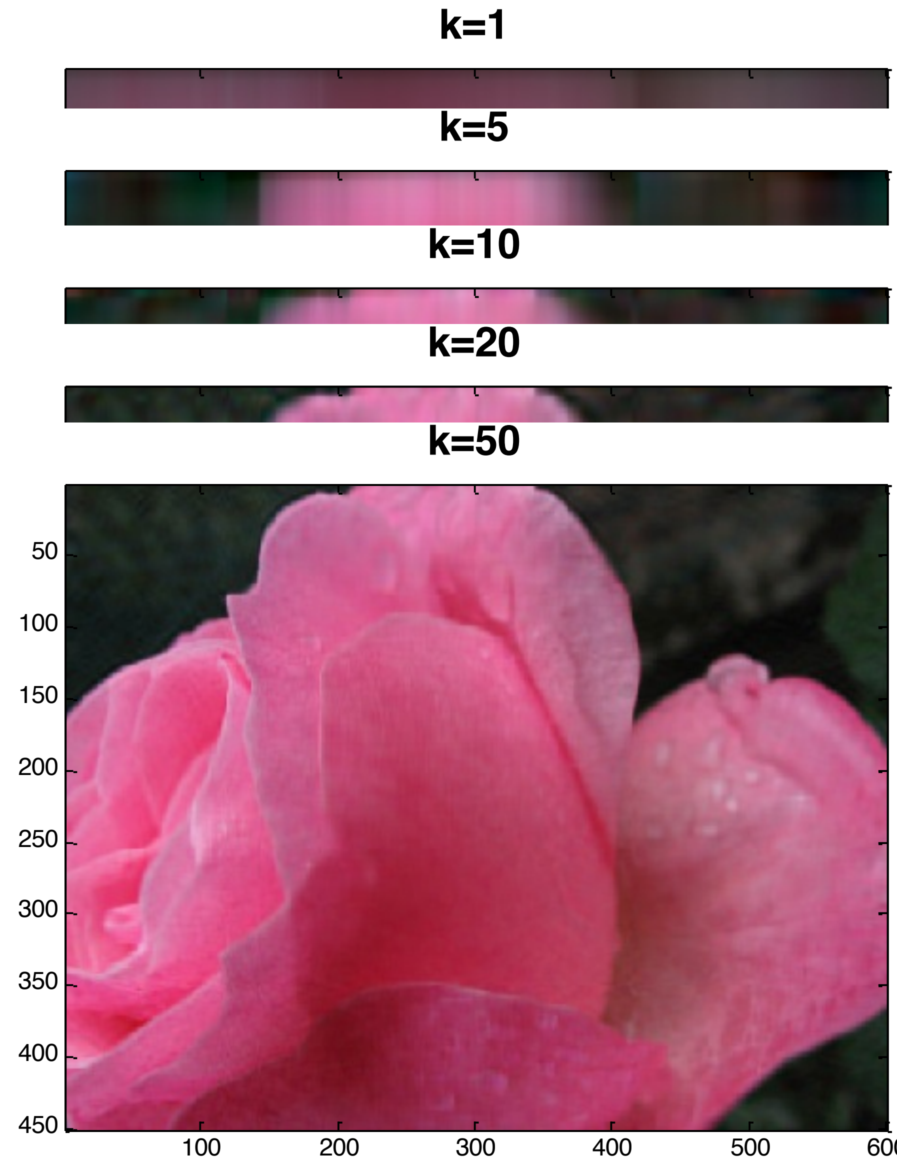
$$X = U\,S\,V$$

- where:
  - S has elements on the diagonal only
  - U 's columns are the eigenvectors of $X^T X$
  - V 's rows are the eigenvectors of $X X^T$

- We can discard the lest significant eigenvectors of U and V, and the corresponding values of S:
  - To reduce the dimensionality of X: $X^{SVD} = V^k\ X$
  - To approximate X: $X^{*} = U^k\ S^k\ V^k$

Image Compression with SVD

ORIGINAL



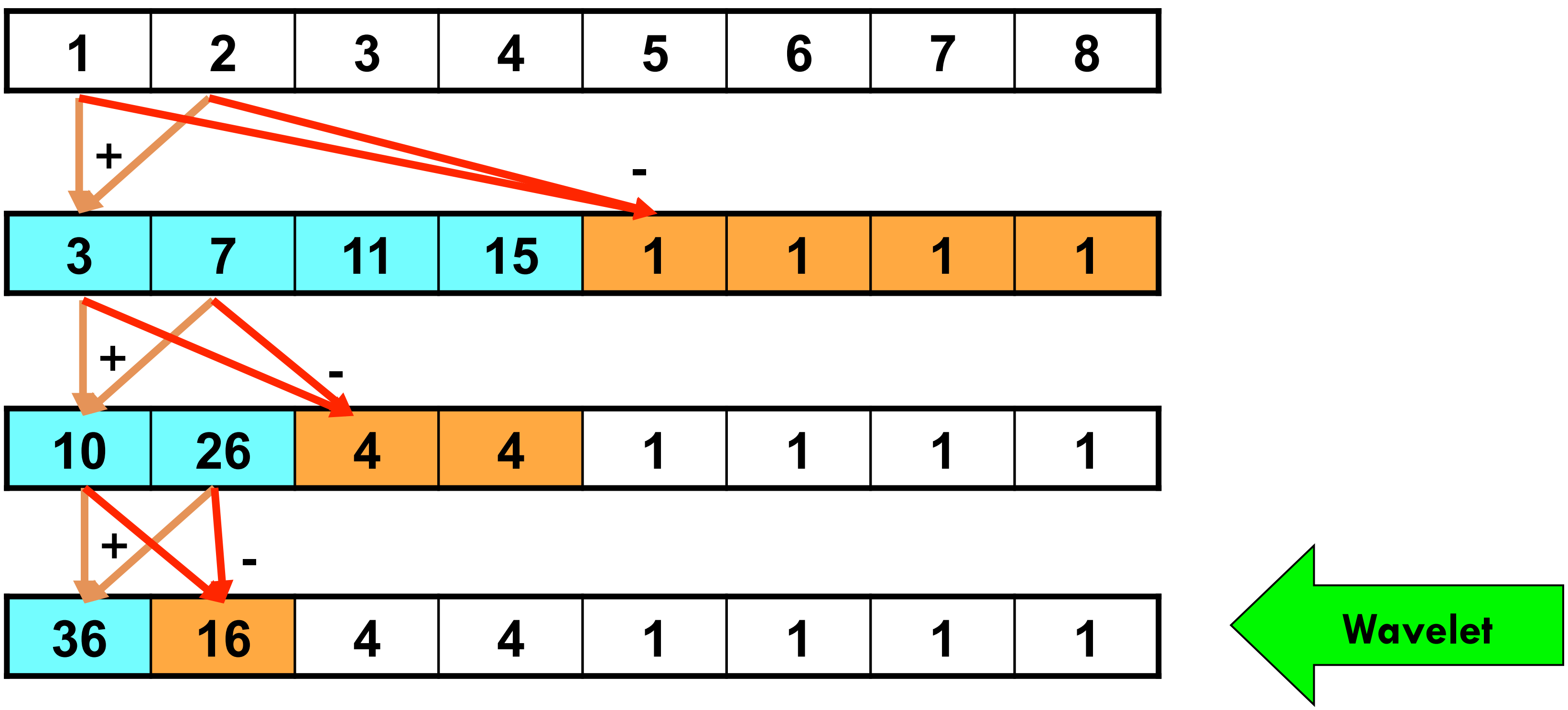
SVD

- NB: When applied to documents, it is usually called *LSI: latent semantic indexing*

- Advantages:
  - Discovers topics
  - Partially solves problems of synonymy and polysemy

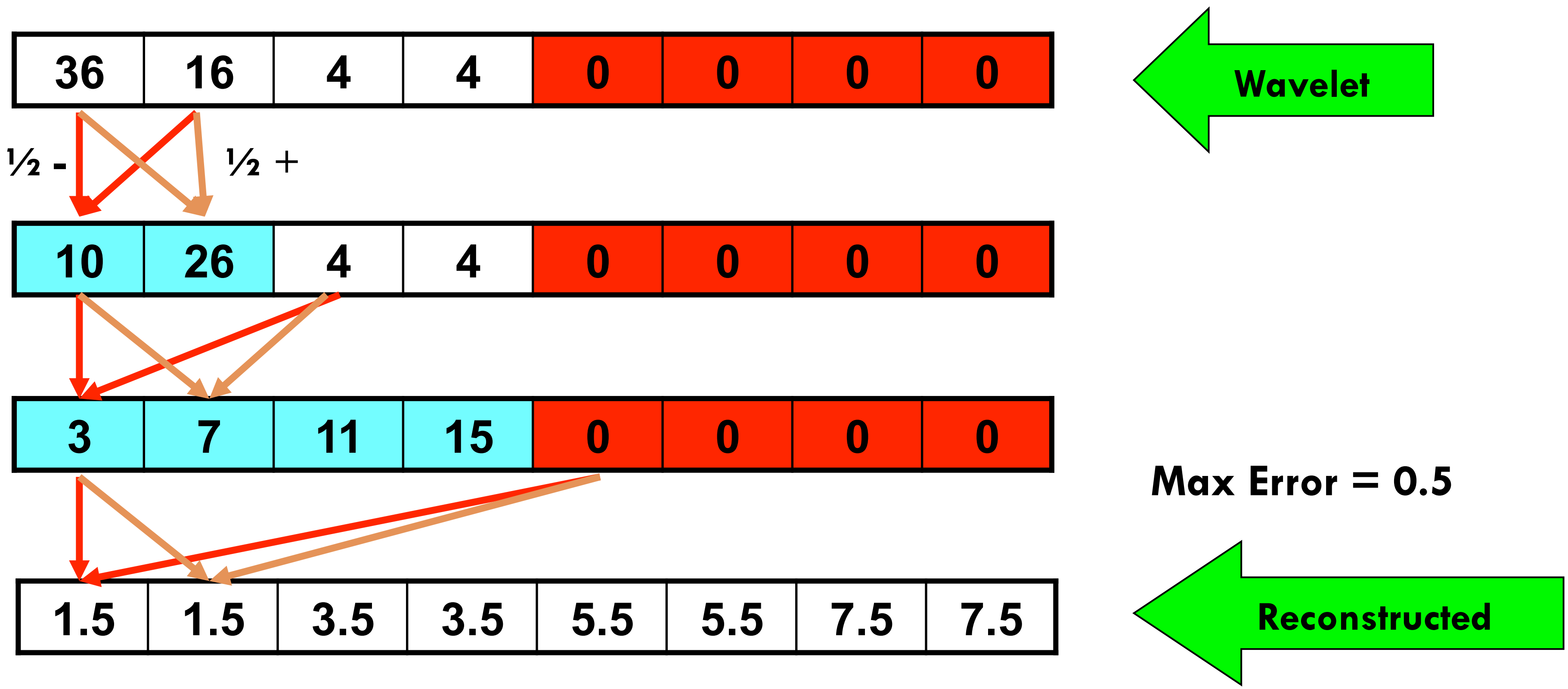
Wavelet

- Recursively apply addition and subtraction functions
  - The first models high frequencies
  - The second low frequencies



Wavelet

- Dimensionality reduction is achieved by:
  - Ignoring low frequencies
  - Other techniques such as quantization

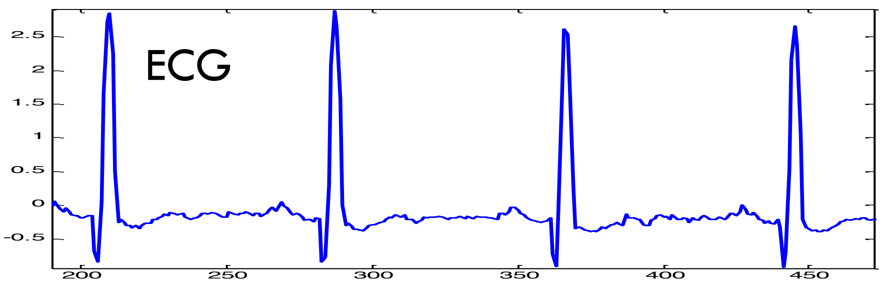


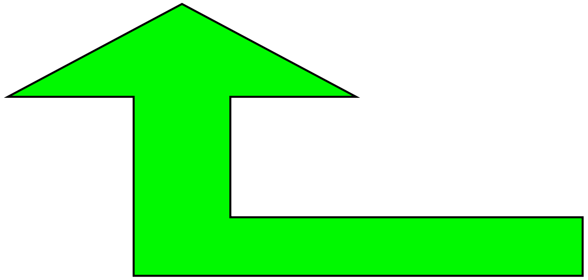
Wavelet image compression

- http://brain.cc.kogakuin.ac.jp/~kanamaru/WaveletJava/Compress/Compress-e.html

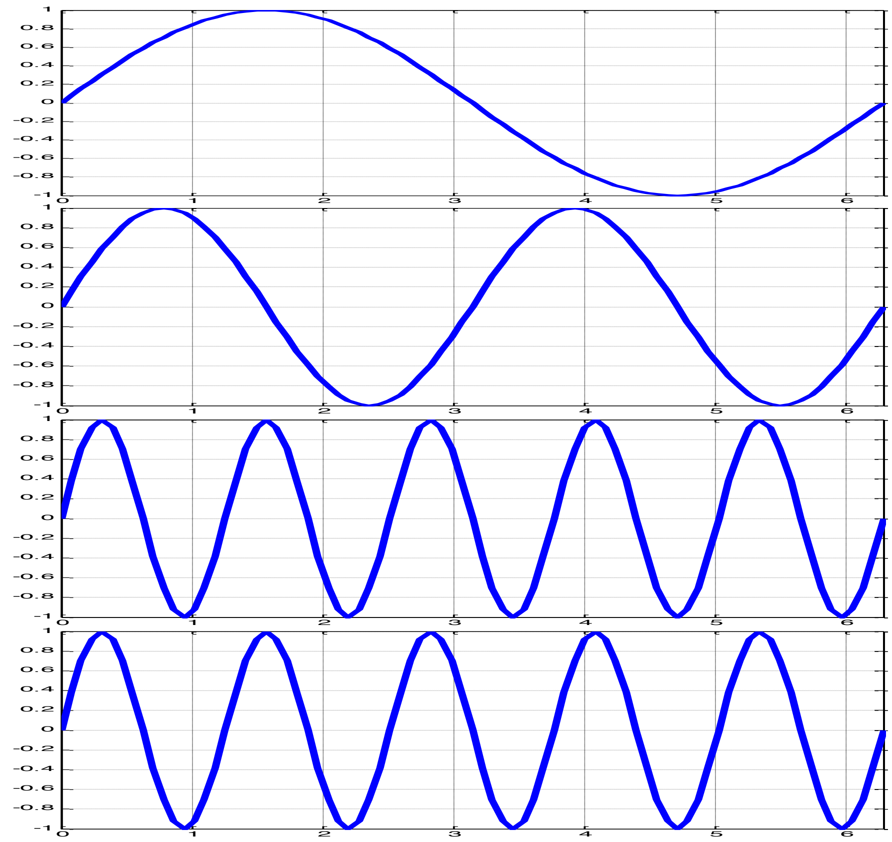
From Fourier Transform to JPEG

- Fourier Transform:
  - Every signal is the sum of various frequencies





Weighted Sum



Discrete Fourier transform (DFT)

- The weights assigned to each frequency are called Fourier coefficients
- In general, high frequency coefficients correspond to noise
  - Sometimes to abrupt changes
- How to reduce dimensionality ?
  - High frequency coefficient are usually close to zero
  - Consider low frequencies only
  - Can use Inverse DFT to get back to an approximation of the original data

JPEG Compression

- Split the image into blocks of 8x8 pixels
- Apply the Discrete Cosine Transform, similar to Fourier Transform

JPEG Compression

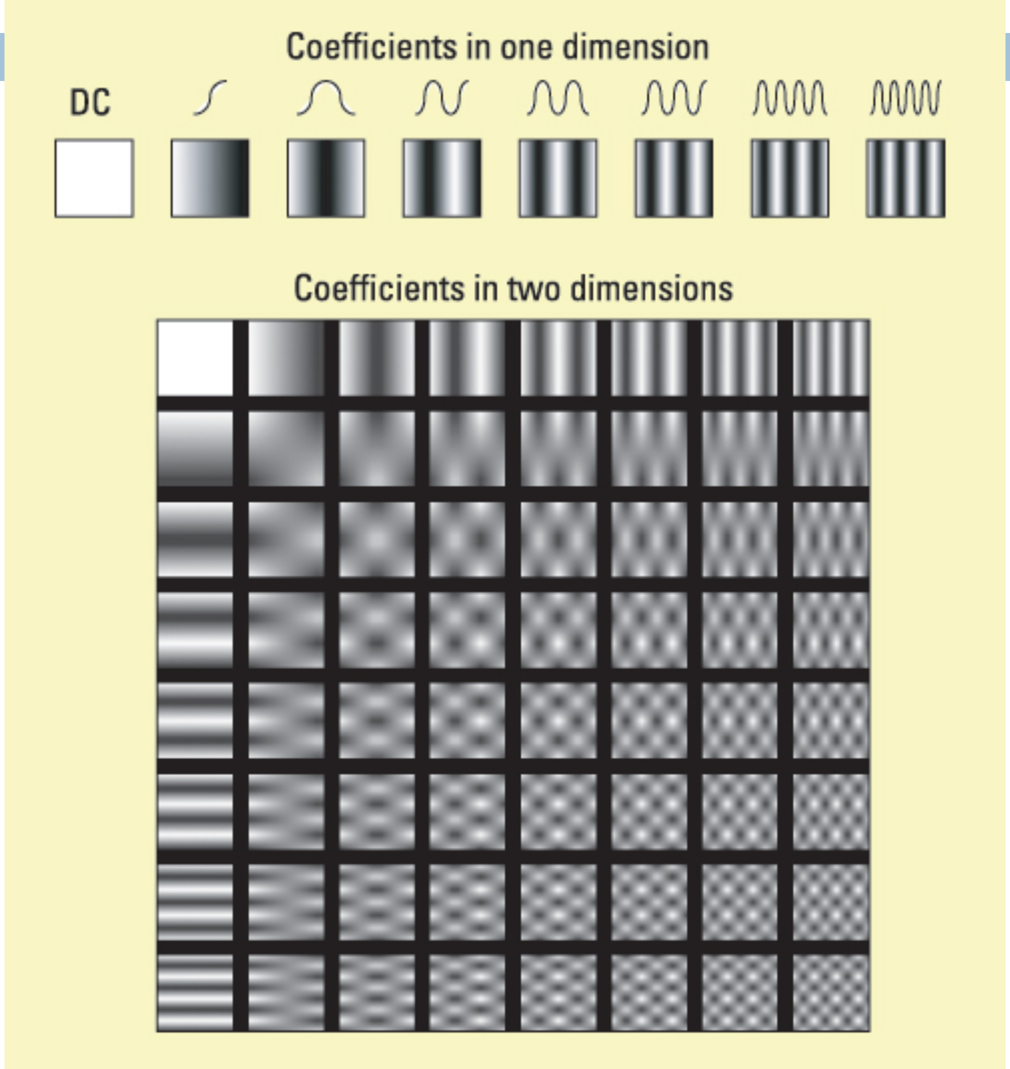


Figure 4. DCT coefficients

JPEG Compression

- Split the image into blocks of 8x8 pixels
- Apply the Fourier Transform
- Get a frequency coefficients matrix

10	20	…					
20	15	…					
13	…						
…							
						…	…
					…	3	2
				…	1	5	0
			…	2	0	0	1

JPEG Compression

- Split the image into blocks of 8x8 pixels
- Apply the Fourier Transform
- Get a frequency coefficients matrix
- Quantize

a. Low compression

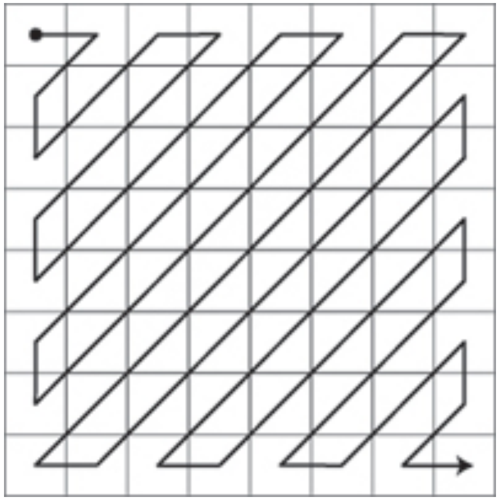
1	1	1	1	1	2	2	4
1	1	1	1	1	2	2	4
1	1	1	1	2	2	2	4
1	1	1	1	2	2	4	8
1	1	2	2	2	2	4	8
2	2	2	2	2	4	8	8
2	2	2	4	4	8	8	16
4	4	4	4	8	8	16	16

b. High compression

1	2	4	8	16	32	64	128
2	4	4	8	16	32	64	128
4	4	8	16	32	64	128	128
8	8	16	32	64	128	128	256
16	16	32	64	128	128	256	256
32	32	64	128	128	256	256	256
64	64	128	128	256	256	256	256
128	128	128	256	256	256	256	256

JPEG Compression

- Split the image into blocks of 8x8 pixels
- Apply the Fourier Transform
- Get a frequency coefficients matrix
- Quantize
- Reorder and compress.



Differences between Wavelet and FFT

- Wavelet are usually better
- Every single coefficient of the Fourier Transform affects the whole image (or block)
- Wavelet coefficient are local
- (Some type of) Wavelet are sensitive to phase shift

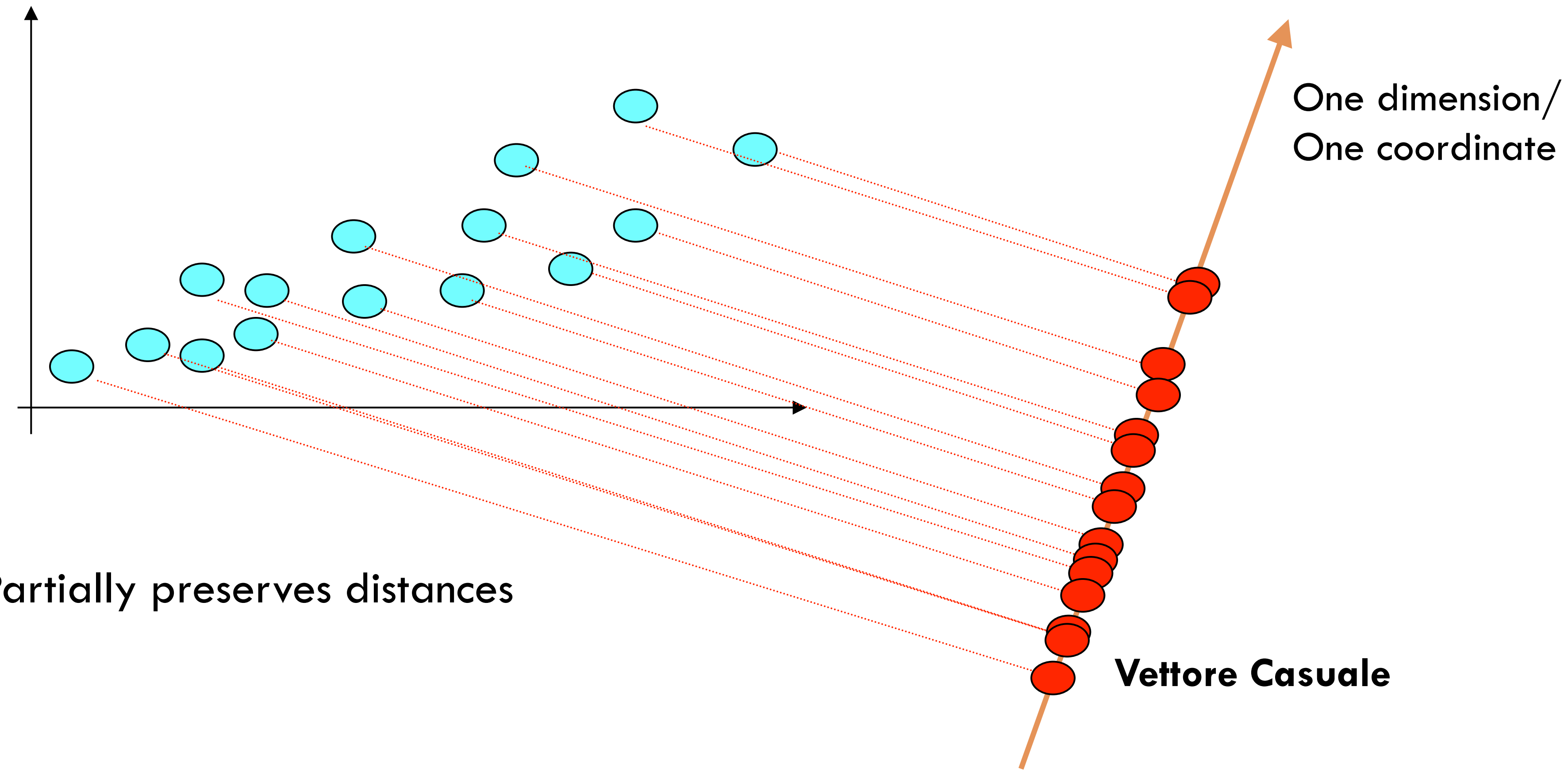
MP3 Compression (MPEG1-Layer3)

- Fourier Transform
- Remove unimportant frequencies

- Psycho-acoustic model:
  - A frequency with large amplitude may cover close frequencies in the *same* time frame
  - A frequency with large amplitude may cover other frequencies in the *next* time frames
  - simultaneous masking vs. temporal masking

Random Projection

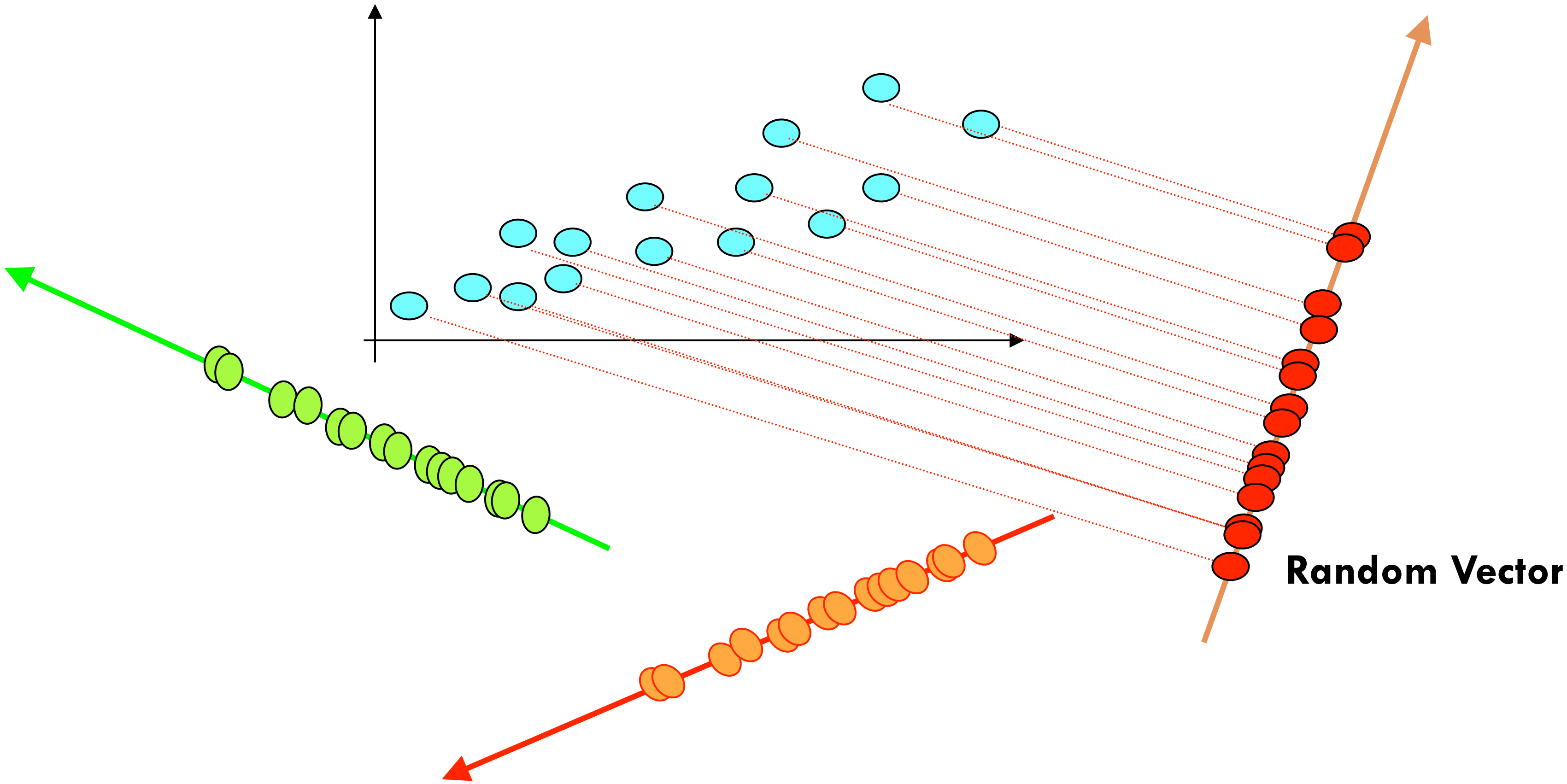
- Project on a new random coordinate system: $X^{RND} = R^k\ X$



- Partially preserves distances

Nearest Neighbor search with RP

- Project on multiple random vectors
- Merge (and filter) results from each "space"



Random Projection applications

- Face recognition