

AIMS CDT - Signal Processing

Michaelmas Term 2023

Xiaowen Dong

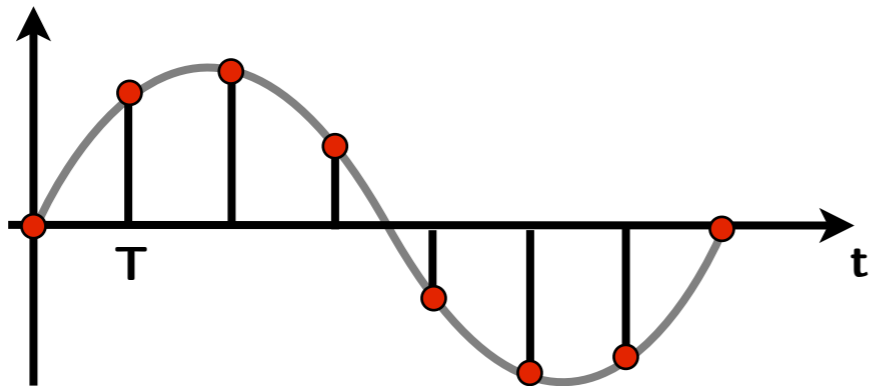
Department of Engineering Science



Representation of Signals

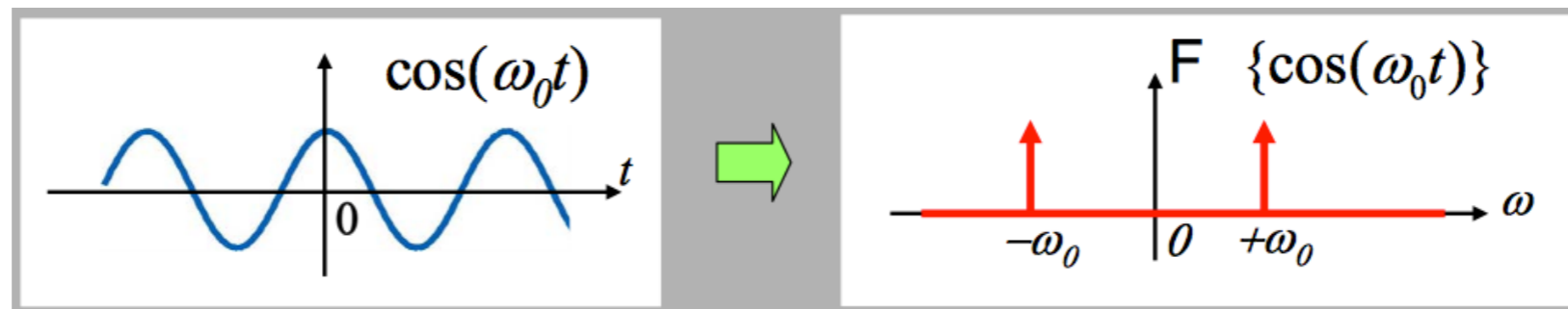
What is a representation of a signal?

- Sum of delta functions in time or space (sampling domain)
 - good for display or playback
 - not good for analysis (e.g., denoising, compression)



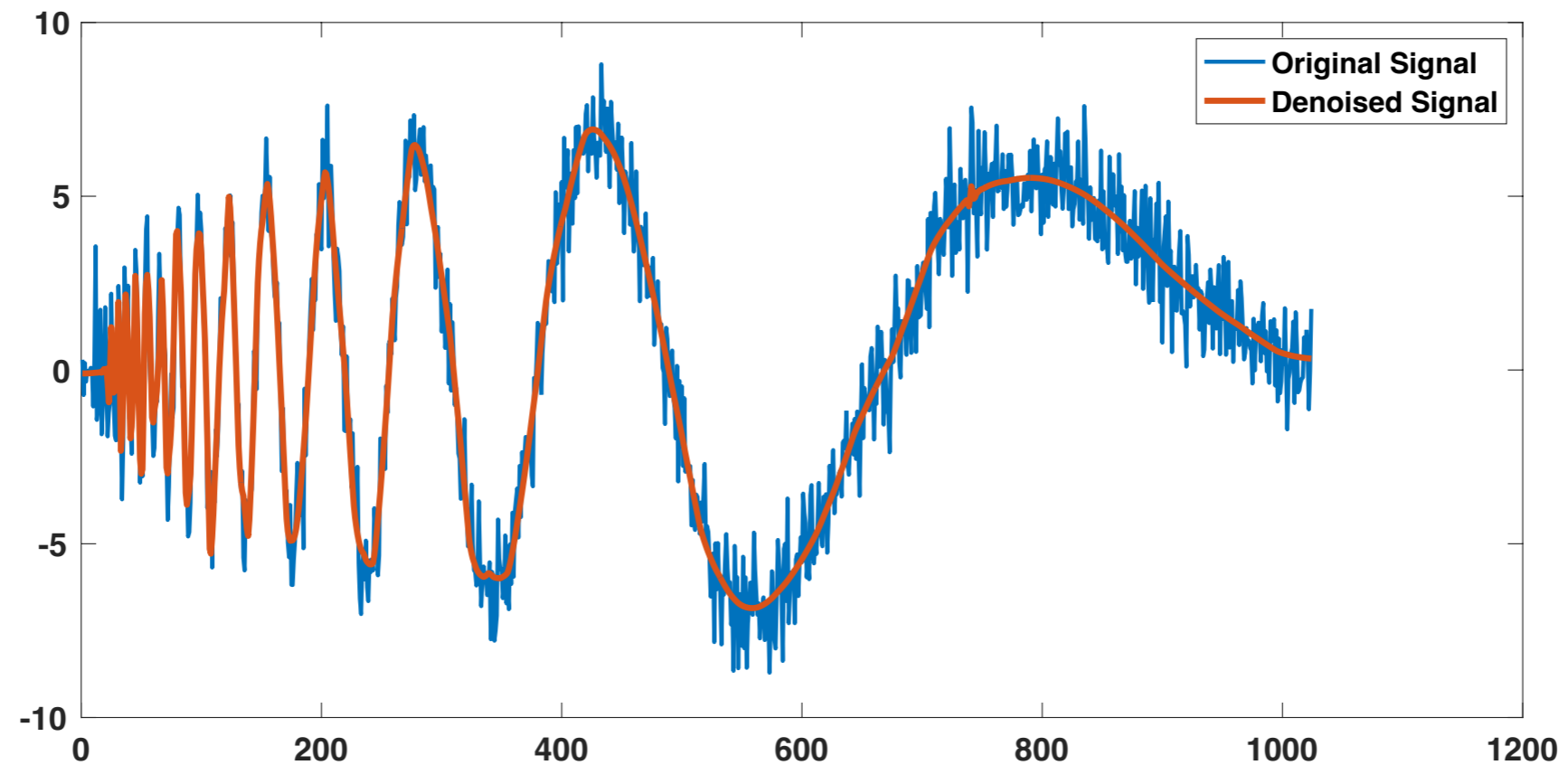
What is a representation of a signal?

- In a time-series setting, useful representation could be past samples
- More generally, it involves transformation of the signal into a new domain where signal characteristics are revealed
 - example: **Fourier coefficients** reveal **rate of change** of the signal



- Usefulness of the representation depends on the analysis goal
 - which may vary but all share the core desire for **simplification**

Example: Denoising



goal: recover signal from noisy observation

Example: Compression

original



JPEG 2000 (10% in size)



JPEG 2000 (1% in size)



goal: compress signal without sacrificing quality

Example: Recognition

samples

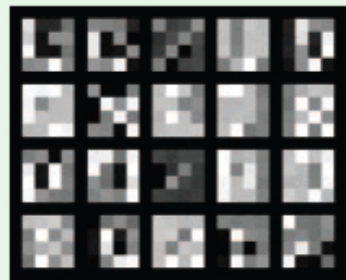
true "causes"

PCA

ICA

sparse coding

KSVD



(a)



(b)



(c)



(d)



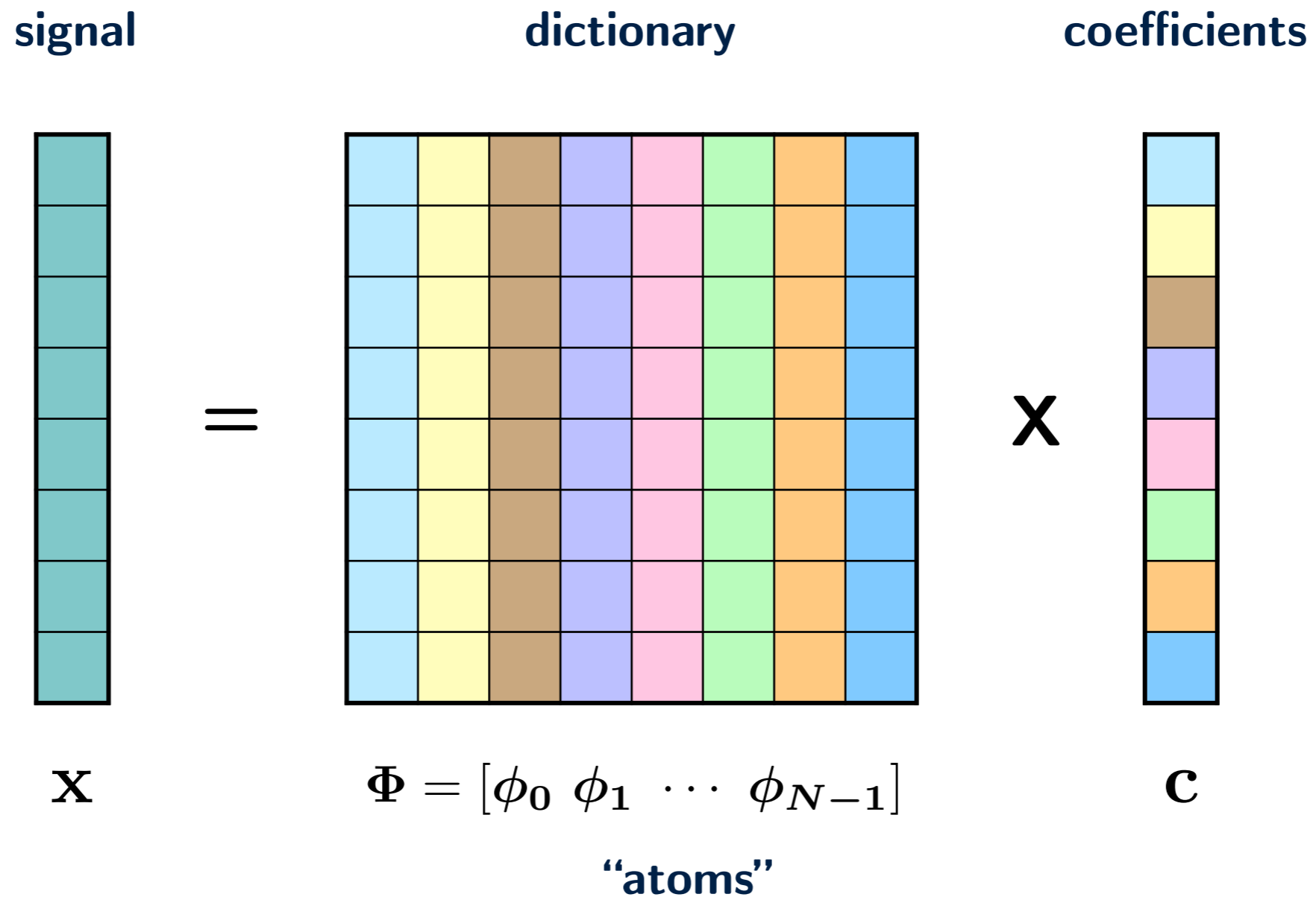
(e)



(f)

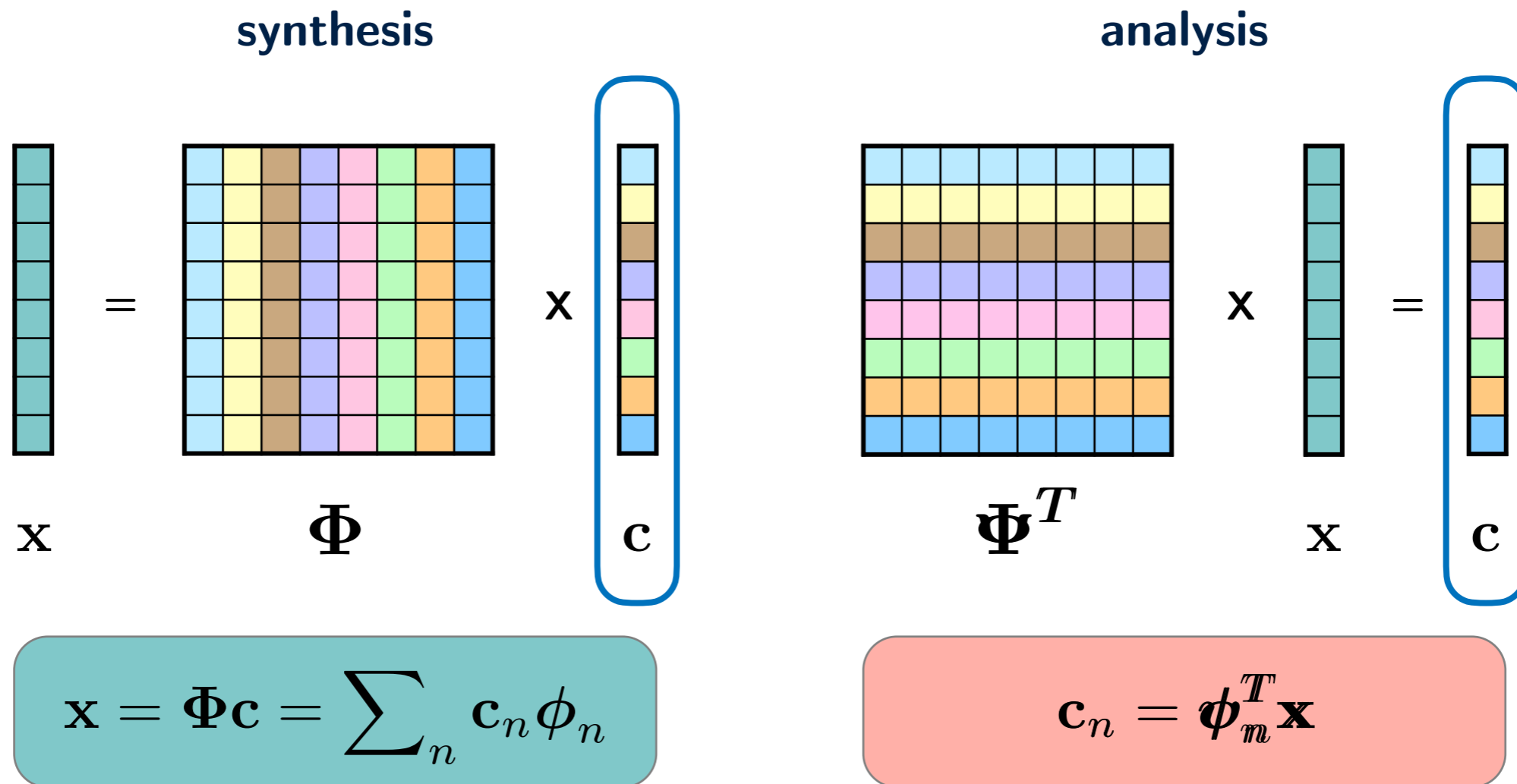
goal: capture true "causes" of signal

Signal representation via dictionaries



Signal representation via dictionaries

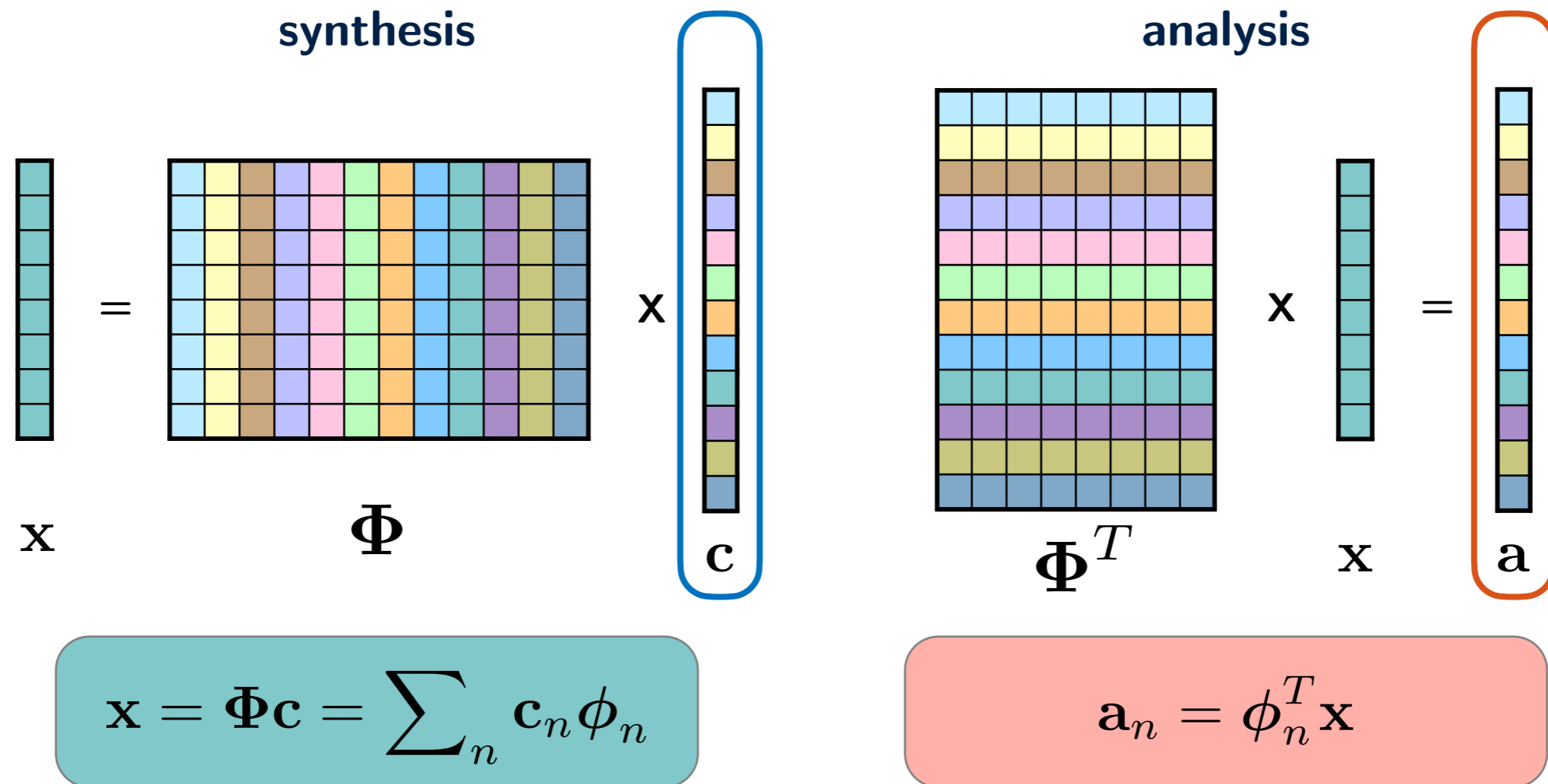
- Complete dictionaries



equivalent for complete dictionaries

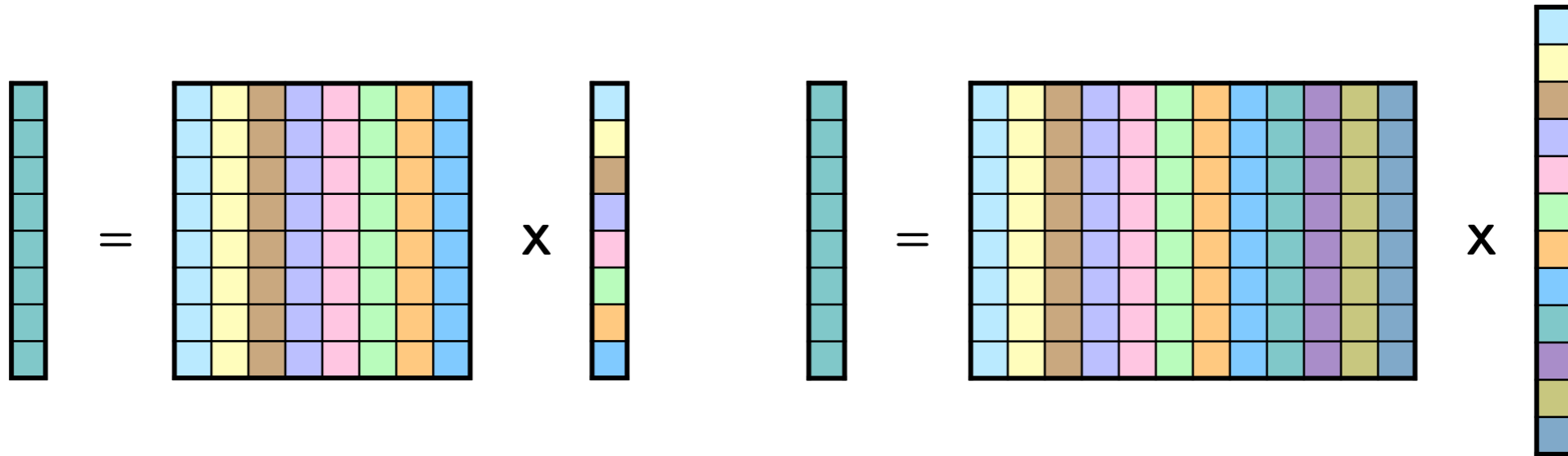
Signal representation via dictionaries

- Over-complete dictionaries



not equivalent for over-complete dictionaries

Signal representation via dictionaries



two sources of dictionary design

- mathematical modelling of data (transforms/analytic dictionaries)
- a set of realisations of data (dictionary learning)

Outline

- A historical overview of dictionary design techniques
 - signal representation via stochastic models
 - transforms & analytic dictionaries
 - trained dictionaries (dictionary learning)
- Discussion
 - applications
 - connection with deep learning

1920s-30s: Stochastic models

- Stochastic models
 - examples of parametric models
 - describe how data were generated
 - provide a special representations of signal from a time-series viewpoint
- Typical examples
 - autoregressive (AR) models
 - moving average (MA) models
 - autoregressive moving average (ARMA) models

Autocorrelation

- Autocovariance: covariance between signal and lagged version of itself

$$\sigma_{xx}(T) = \frac{1}{N-1} \sum_{t=1}^N \underbrace{(x_{t-T} - \mu_x)}_{\text{lagged version by T samples}} \underbrace{(x_t - \mu_x)}_{\text{signal}}$$

- Autocorrelation: normalised autocovariance

$$r_{xx}(T) = \frac{\sigma_{xx}(T)}{\sigma_{xx}(0)} \quad \text{where } \sigma_{xx}(0) = \sigma_x^2$$

- Both are **symmetric** or **even** functions

Autocorrelation

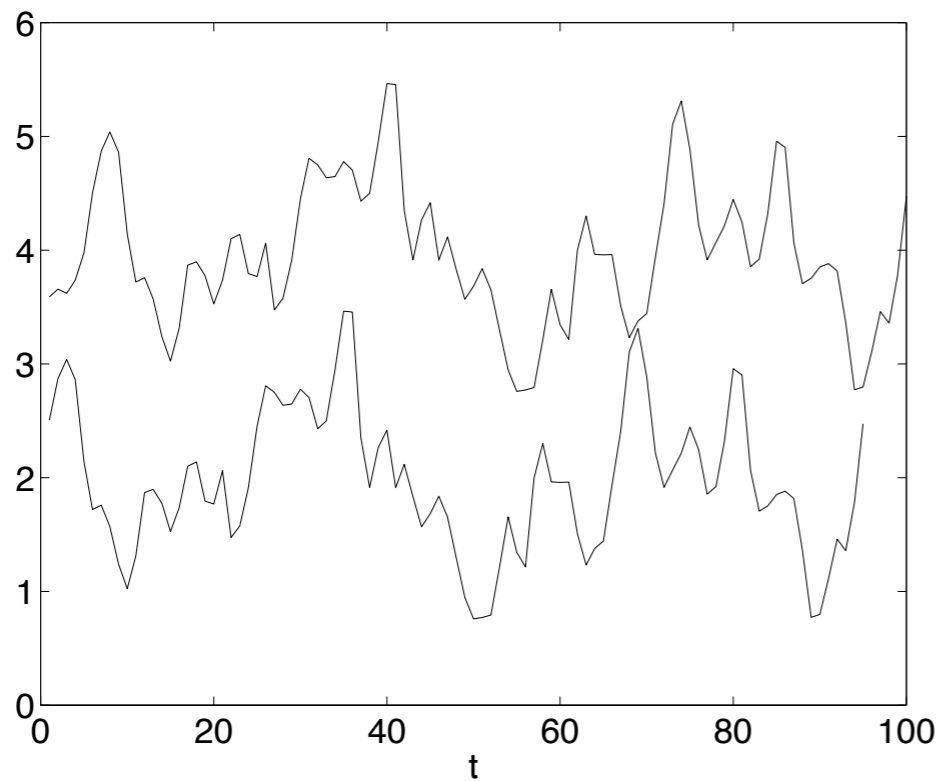


Figure 4.7: Signal x_t (top) and x_{t+5} (bottom). The bottom trace **leads** the top trace by 5 samples. Or we may say it **lags** the top by -5 samples.

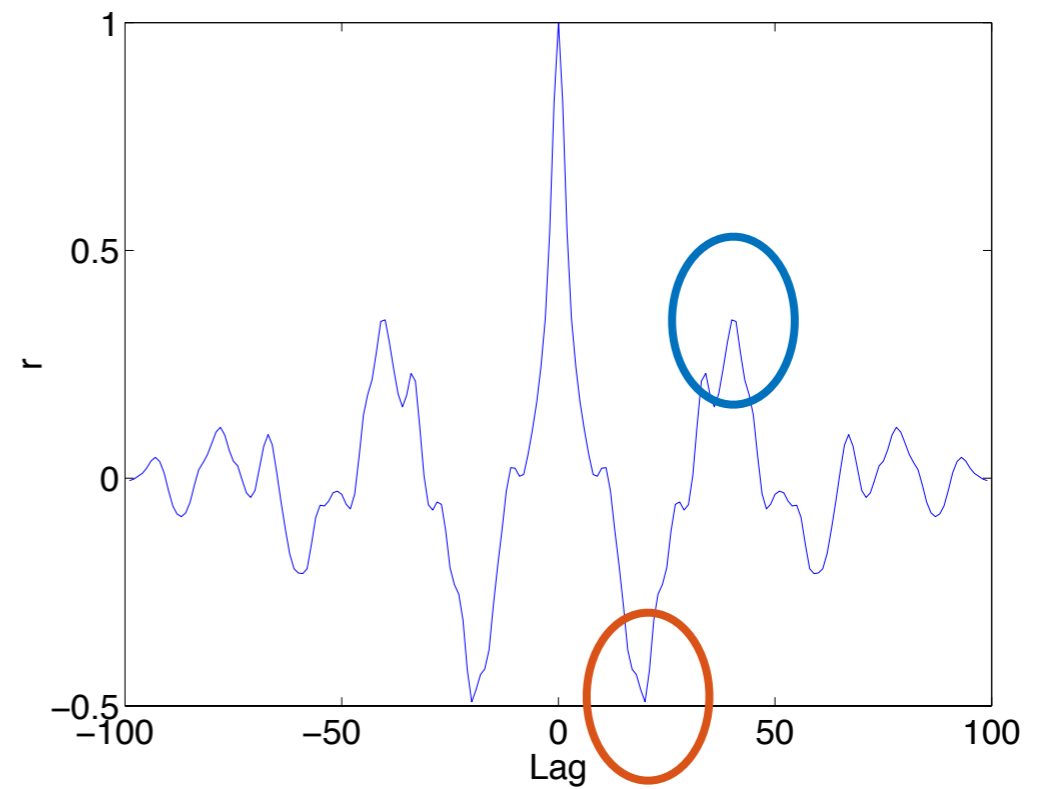


Figure 4.8: Autocorrelation function for x_t . Notice the negative correlation at lag 20 and positive correlation at lag 40. Can you see from Figure 4.7 why these should occur?

AR models

- An AR model **predicts** the value of a time-series from previous values

$$x_t = \sum_{i=1}^p x_{t-i} a_i + e_t$$

AR coefficients prediction error $e_t \sim \mathcal{N}(0, \sigma_e^2)$

- Matrix form

$$\mathbf{M} = \begin{bmatrix} x_4 & x_3 & x_2 & x_1 \\ x_5 & x_4 & x_3 & x_2 \\ \dots & \dots & \dots & \dots \\ x_{N-1} & x_{N-2} & x_{N-3} & x_{N-4} \end{bmatrix} \rightarrow \begin{bmatrix} x_5 \\ x_6 \\ \dots \\ x_N \end{bmatrix} = \begin{bmatrix} x_4 & x_3 & x_2 & x_1 \\ x_5 & x_4 & x_3 & x_2 \\ \dots & \dots & \dots & \dots \\ x_{N-1} & x_{N-2} & x_{N-3} & x_{N-4} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} + \begin{bmatrix} e_5 \\ e_6 \\ \dots \\ e_N \end{bmatrix}$$

embedding matrix

$$\mathbf{x} = \mathbf{M}\mathbf{a} + \mathbf{e}$$

Estimation of AR: Least-squares method

- The AR model is a special case of the multivariate regression model
- It also provides a special **representation** of the signal
- To compute AR coefficients and predictions

$$\mathbf{x} = \mathbf{M}\mathbf{a} + \mathbf{e} \quad \longrightarrow \quad \begin{aligned} \hat{\mathbf{a}} &= (\mathbf{M}^T \mathbf{M})^{-1} \mathbf{M}^T \mathbf{x} \\ \hat{\mathbf{x}} &= \mathbf{M}\hat{\mathbf{a}} \\ \mathbf{e} &= \mathbf{x} - \hat{\mathbf{x}} \end{aligned}$$

Estimation of AR: Least-squares method

- Use an AR(4) model to analyse data shown before:

$$\hat{\mathbf{a}} = [1.46, -1.08, 0.60, -0.186]^T \quad \sigma_e^2 = 0.079 \quad \sigma_x^2 = 0.3882$$

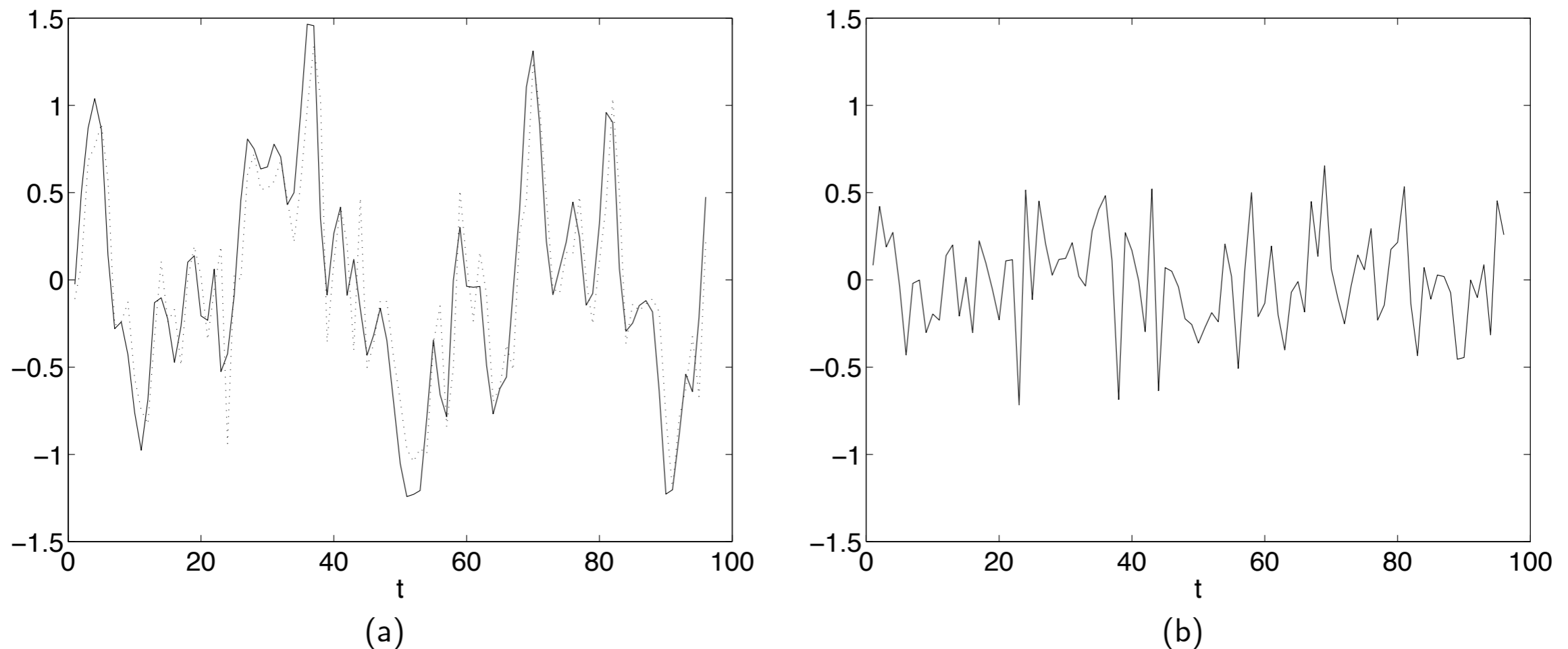


Figure 4.9: (a) Original signal (solid line), \mathbf{X} , and predictions (dotted line), $\hat{\mathbf{X}}$, from an AR(4) model and (b) the prediction errors, \mathbf{e} . Notice that the variance of the errors is much less than that of the original signal.


Estimation of AR: Yule-Walker method

- Relation to autocorrelation


$$x_t = a_1 x_{t-1} + a_2 x_{t-2} + \dots + a_p x_{t-p} + e_t$$

 multiply by x_{t-k}

$$x_t x_{t-k} = a_1 x_{t-1} x_{t-k} + a_2 x_{t-2} x_{t-k} + \dots + a_p x_{t-p} x_{t-k} + e_t x_{t-k}$$

 sum over t and divide by N-1

$$\sigma_{xx}(k) = a_1 \sigma_{xx}(k-1) + a_2 \sigma_{xx}(k-2) + \dots + a_p \sigma_{xx}(k-p) + \sigma_{e,x}$$

 divide by signal variance

$$r_{xx}(k) = a_1 r_{xx}(k-1) + a_2 r_{xx}(k-2) + \dots + a_p r_{xx}(k-p)$$

Estimation of AR: Yule-Walker method

- For an AR(4) model

$$\begin{bmatrix} r_{xx}(1) \\ r_{xx}(2) \\ r_{xx}(3) \\ r_{xx}(4) \end{bmatrix} = \begin{bmatrix} r_{xx}(0) & r_{xx}(-1) & r_{xx}(-2) & r_{xx}(-3) \\ r_{xx}(1) & r_{xx}(0) & r_{xx}(-1) & r_{xx}(-2) \\ r_{xx}(2) & r_{xx}(1) & r_{xx}(0) & r_{xx}(-1) \\ r_{xx}(3) & r_{xx}(2) & r_{xx}(1) & r_{xx}(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} \quad \text{Yule-Walker relations}$$

$$\mathbf{r} = \mathbf{R}\mathbf{a}$$

- More efficient way to estimate AR coefficients: $\mathbf{a} = \mathbf{R}^{-1}\mathbf{r}$

$$\mathbf{R} = \begin{bmatrix} 1 & r_{xx}(1) & r_{xx}(2) & r_{xx}(3) \\ r_{xx}(1) & 1 & r_{xx}(1) & r_{xx}(2) \\ r_{xx}(2) & r_{xx}(1) & 1 & r_{xx}(1) \\ r_{xx}(3) & r_{xx}(2) & r_{xx}(1) & 1 \end{bmatrix}$$

- autocorrelation matrix is symmetric and Toeplitz
- efficient computation via a recursive estimation technique (Levinson-Durbin)

Outline

- A historical overview of dictionary design techniques
 - signal representation via stochastic models
 - transforms & analytic dictionaries
 - trained dictionaries (dictionary learning)
- Discussion
 - applications
 - connection with deep learning

1960s: Fourier basis and DFT

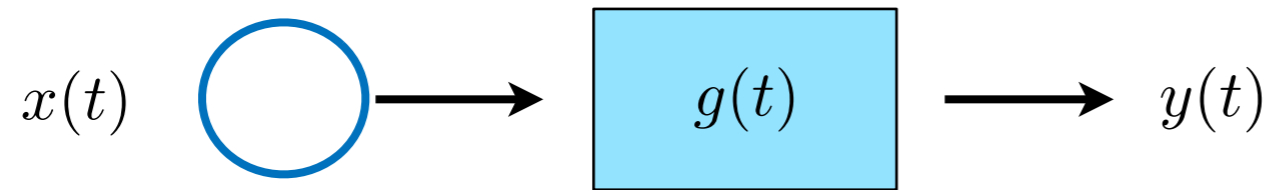
discrete
Fourier
transform



1960s

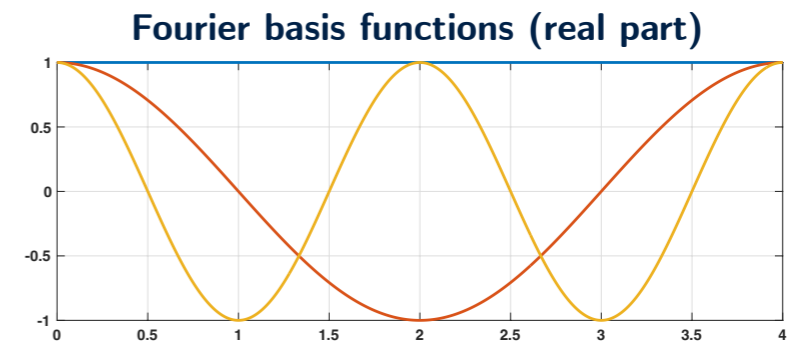


- recall the LTI system



$$y(t) = \int_{-\infty}^{\infty} e^{j\omega(t-\tau)} g(\tau) d\tau = \underbrace{e^{j\omega t}}_{\text{Fourier basis}} G(\omega)$$

$$X(\omega) \longrightarrow G(\omega) \longrightarrow Y(\omega) = X(\omega)G(\omega)$$



**Fourier basis diagonalises
convolution operator**

1960s: Fourier basis and DFT

discrete
Fourier
transform



1960s



- Fourier basis describes a signal in terms of its **global** frequency content and hence is good at representing **uniformly smooth** signals
- discrete Fourier transform (DFT) provides an **orthogonal** dictionary: $\phi_n(k) = e^{j\frac{2\pi}{N}nk}$

$$\begin{pmatrix} x[0] \\ x[1] \\ x[2] \\ \vdots \\ x[N-1] \end{pmatrix} = \frac{1}{N} \begin{pmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & W & W^2 & W^3 & \dots & W^{N-1} \\ 1 & W^2 & W^4 & W^6 & \dots & W^{N-2} \\ 1 & W^3 & W^6 & W^9 & \dots & W^{N-3} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & W^{N-1} & W^{N-2} & W^{N-3} & \dots & W \end{pmatrix} \begin{pmatrix} X[0] \\ X[1] \\ X[2] \\ \vdots \\ X[N-1] \end{pmatrix} \quad \text{with } W = e^{j\frac{2\pi}{N}}$$

- fast Fourier transform (FFT) reduces complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(N\log N)$

1960s: Fourier basis and DFT

discrete
Fourier
transform



1960s



- DFT produces complex coefficients (“wasteful” for real signals)
- DFT assumes periodic extension (discontinuity at boundary)

Fourier transform $X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t} dt = \int_{-\infty}^{\infty} x(t)[\cos(\omega t) - j\sin(\omega t)]dt$

$x(t) = x(-t)$ \longrightarrow $X(\omega) = \int_{-\infty}^{\infty} x(t)\cos(\omega t)dt$ **cosine transform**

- a real and even signal leads to a **real** cosine transform

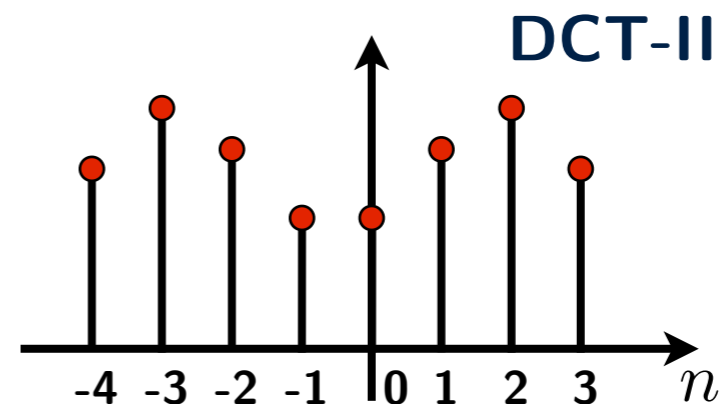
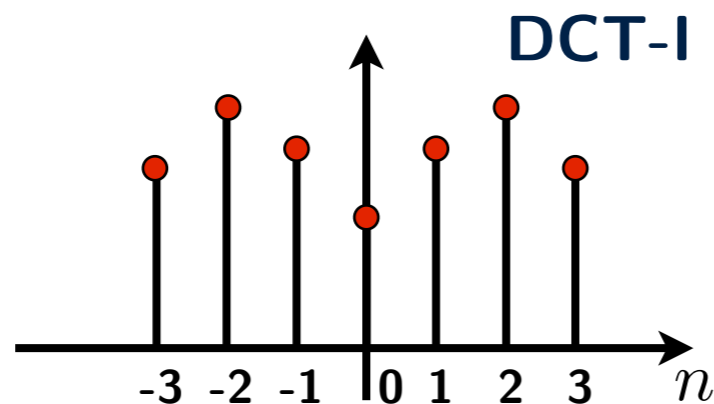
1970s: DCT

*discrete
Fourier
transform* *discrete
cosine
transform*

1960s

1974

- the discrete version is called the discrete cosine transform (DCT)
- several variants of symmetric extension, which all make the signal **even** and lead to **smoother boundary**



- DCT-II provides a real dictionary: $\phi_n(k) = \cos\left[\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right]$
- DCT-II is behind the JPEG image compression standard

1970s-80s: KLT and PCA

discrete
Fourier
transform



1960s

discrete
cosine
transform



1974

Karhunen-
Loève
transform



1970s-80s



- projection onto a fixed subset of DFT or DCT atoms leads to **compaction**

$$\mathbf{x} \approx \sum_{n \in \mathcal{S}_k} (\Psi_n^T \mathbf{x}) \Phi_n$$

- but **data** themselves can also be a source of compaction
- Karhunen-Loève transform (KLT) or principal component analysis (PCA) fits a low-dimensional subspace to data

$$\Sigma = \Phi \Lambda \Phi^T$$

known/empirical covariance

eigenvectors (dictionary atoms as k largest eigenvectors)

- representation is efficient (maximally compacts energy) but expensive to compute

DCT vs KLT

*discrete
Fourier
transform*

*discrete
cosine
transform*

*Karhunen-
Loève
transform*

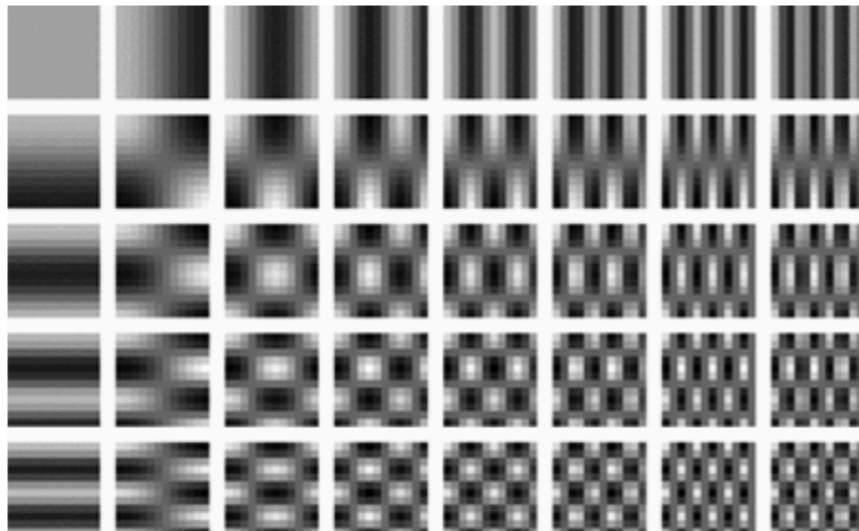
1960s

1974

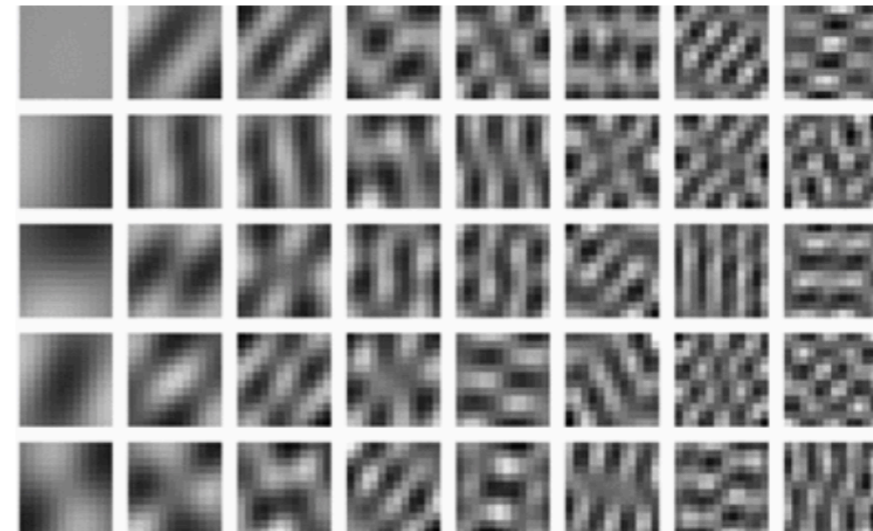
1970s-80s

- DCT atoms (12x12) vs. KLT atoms (trained using 12x12 image patches)

DCT



KLT



The need for sparsity

*discrete
Fourier
transform*

*discrete
cosine
transform*

*Karhunen-
Loève
transform*

1960s

1974

1970s-80s

- simplicity motivates **sparsity**: signal as linear combination of a few atoms
- sparsity requires shift from linear to **nonlinear** approximation

$$\mathbf{x} \approx \sum_{\mathcal{K}} c_k \phi_k \rightarrow \text{subset of atoms (different for each } \mathbf{x})$$

- sparsity requires **localisation**: atoms with concentrated supports
 - allow more flexible representations based on local characteristics
 - limit effects of irregularities (a main source of large coefficients)

Time-frequency representation

time localisation

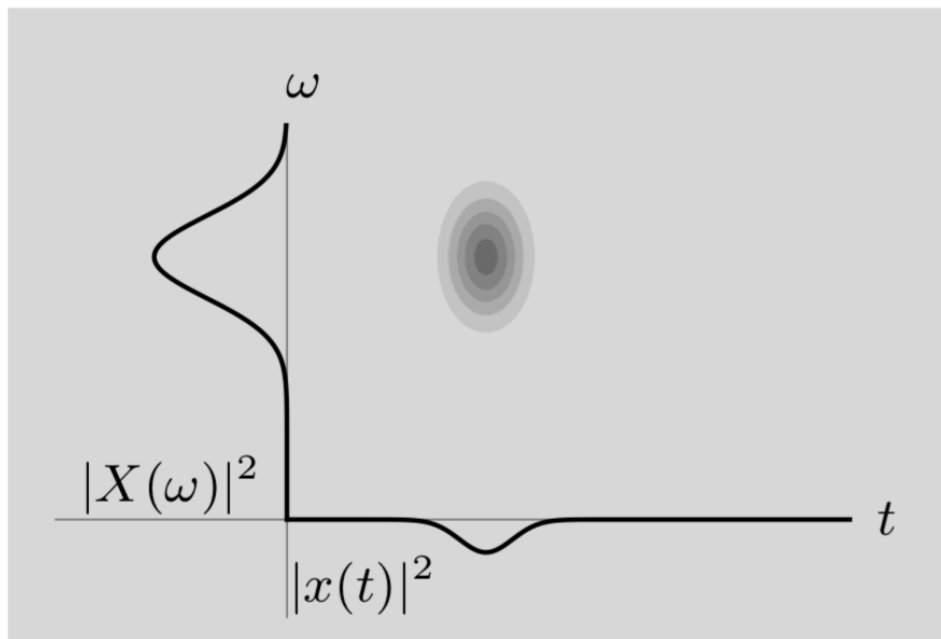
$$\mu_t = \frac{1}{\|x\|^2} \int_{-\infty}^{\infty} t |x(t)|^2 dt$$

$$\Delta_t = \left(\frac{1}{\|x\|^2} \int_{-\infty}^{\infty} (t - \mu_t)^2 |x(t)|^2 dt \right)^{\frac{1}{2}}$$

frequency localisation

$$\mu_f = \frac{1}{2\pi \|x\|^2} \int_{-\infty}^{\infty} \omega |X(\omega)|^2 d\omega$$

$$\Delta_f = \left(\frac{1}{2\pi \|x\|^2} \int_{-\infty}^{\infty} (\omega - \mu_f)^2 |X(\omega)|^2 d\omega \right)^{\frac{1}{2}}$$



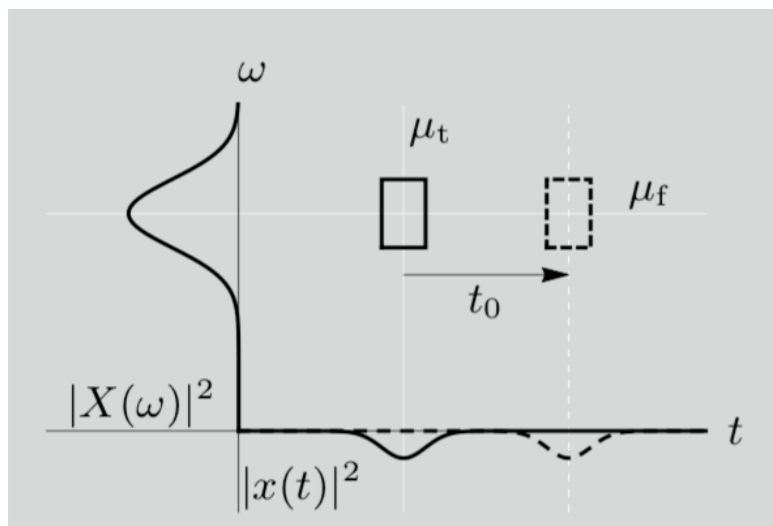
time-frequency tile
(Heisenberg box)

time-frequency
plane

Time-frequency representation

- Consider three basic operations

shift in time

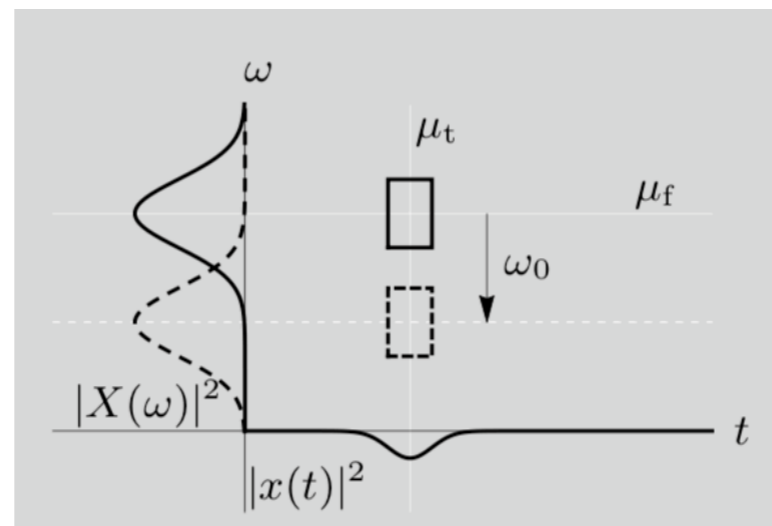


$$y(t) = x(t - t_0)$$

FT ↓

$$Y(\omega) = e^{-j\omega t_0} X(\omega)$$

shift in frequency

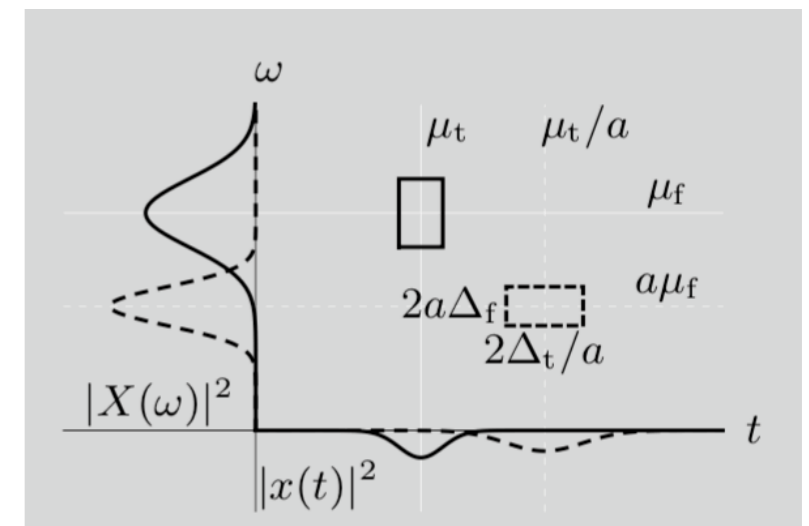


$$y(t) = e^{j\omega_0 t} x(t)$$

FT ↓

$$Y(\omega) = X(\omega - \omega_0)$$

scaling in time



$$y(t) = \sqrt{a} x(at)$$

FT ↓

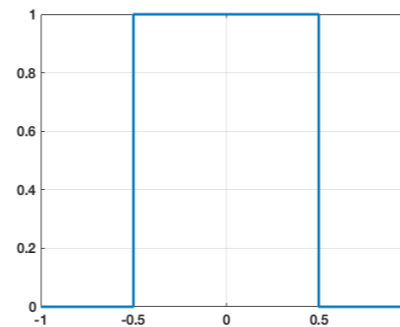
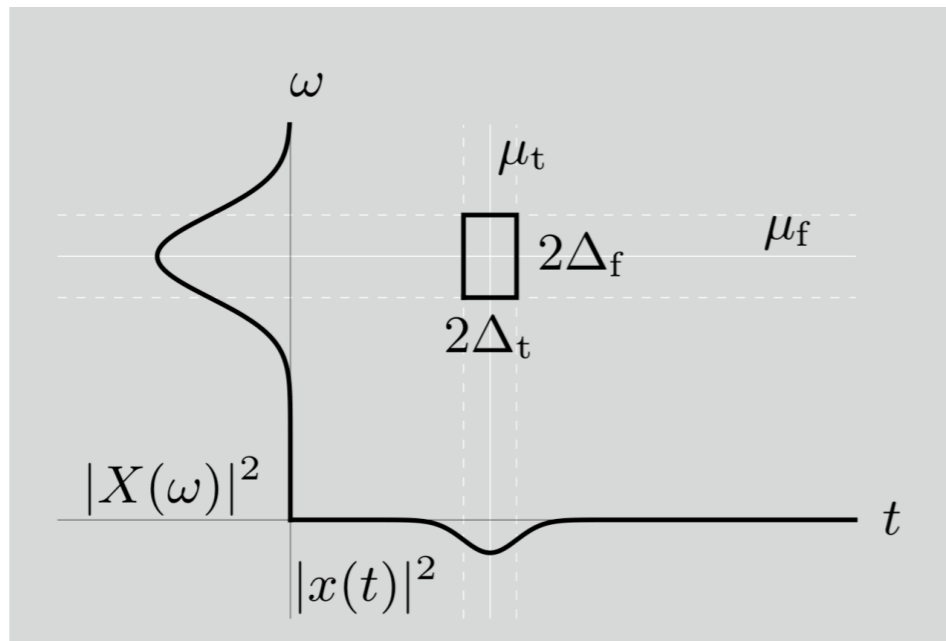
$$Y(\omega) = \frac{1}{\sqrt{a}} X\left(\frac{\omega}{a}\right)$$

Time-frequency representation

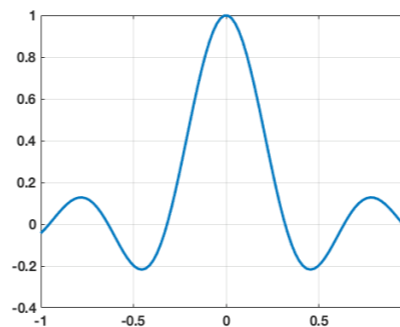
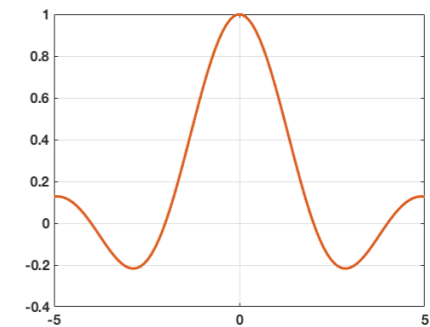
Heisenberg's uncertainty principle

$$\text{Let } x \in \mathcal{L}^2(\mathbb{R}), \text{ then } \Delta_t \Delta_f \geq \frac{1}{2}$$

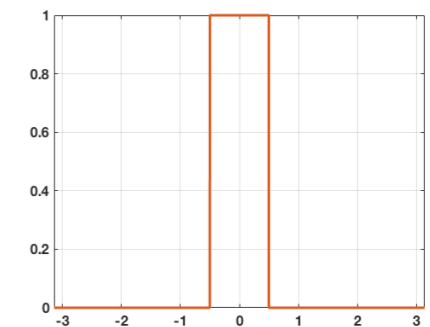
examples



FT
→



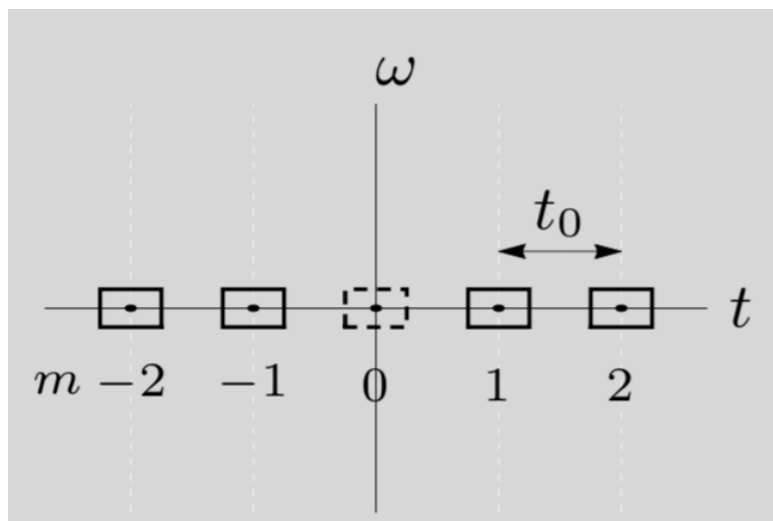
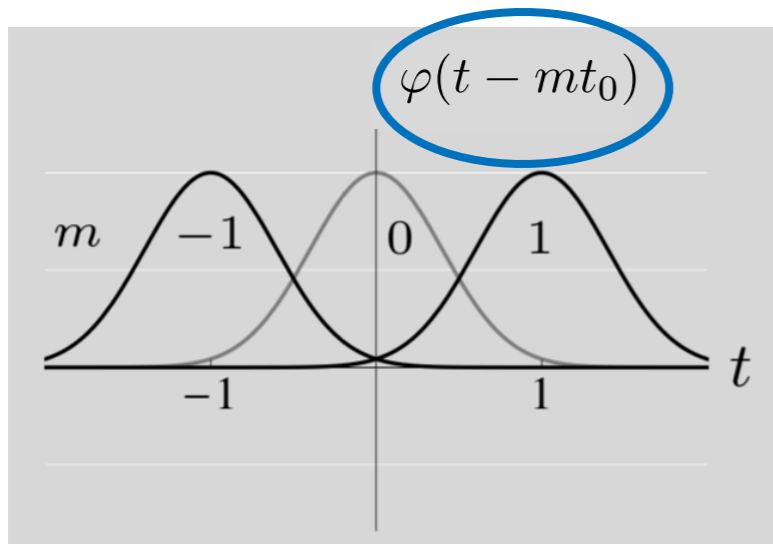
FT
→



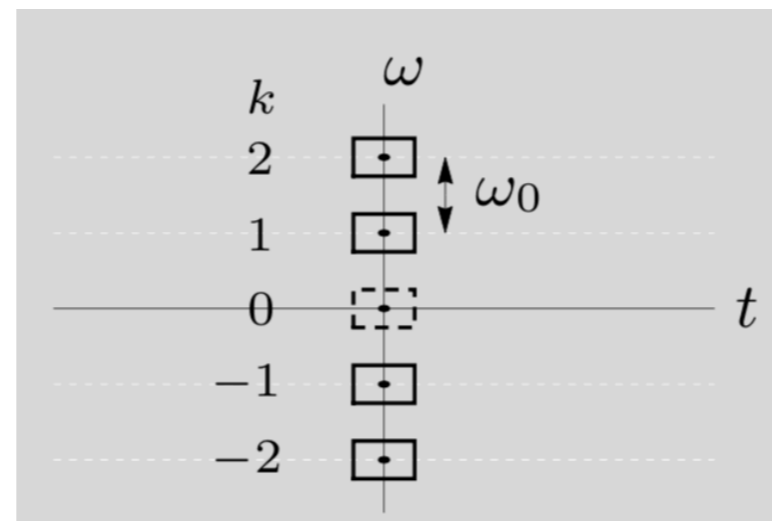
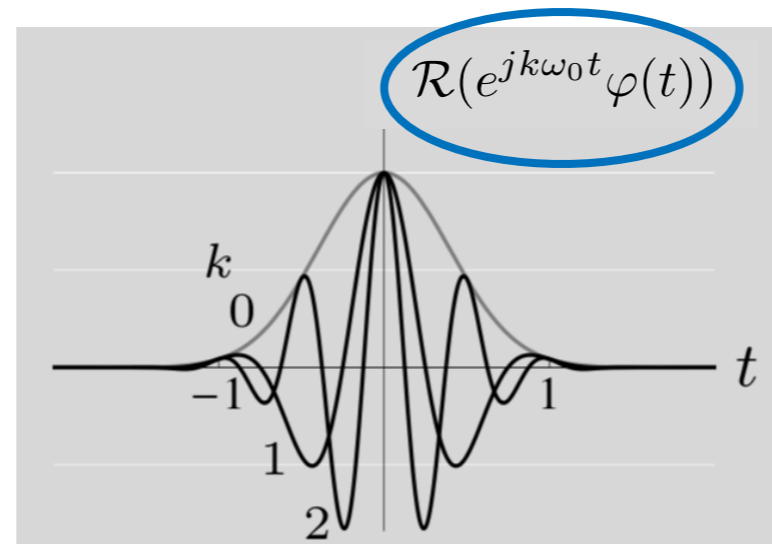
Time-frequency representation

- Consider three structured sets of functions

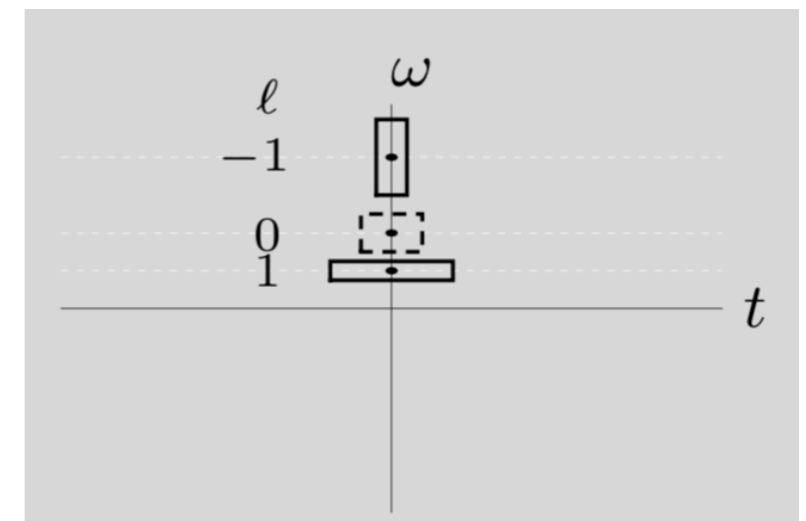
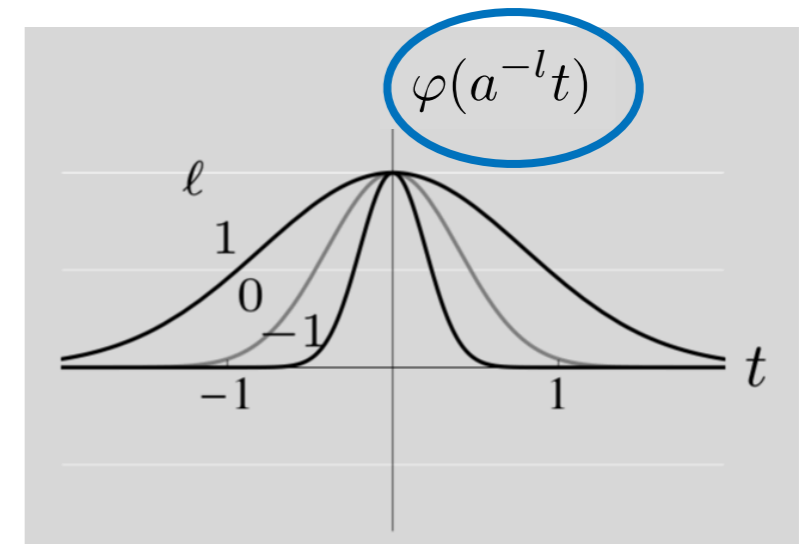
shift in time



shift in frequency

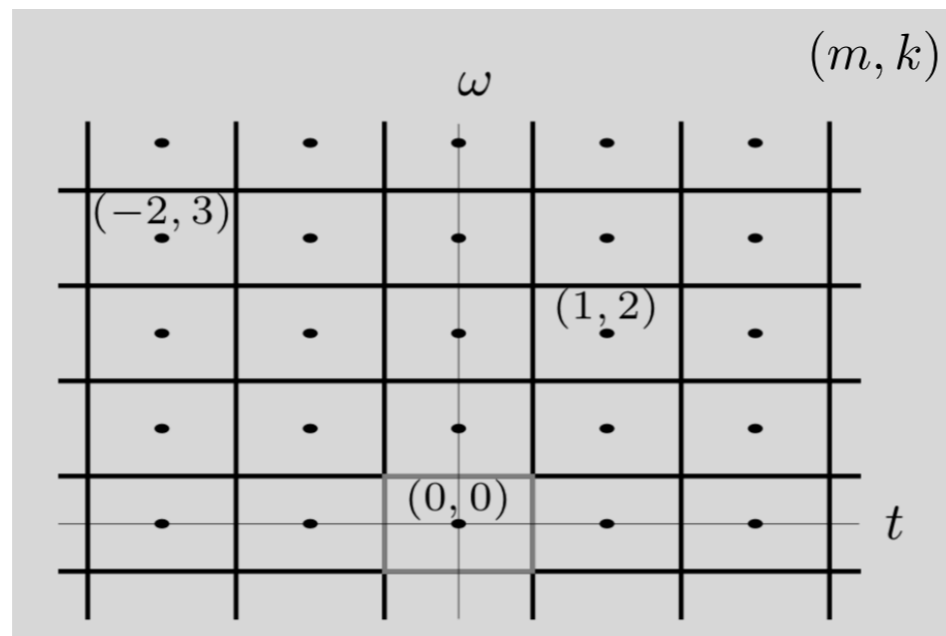


scaling in time

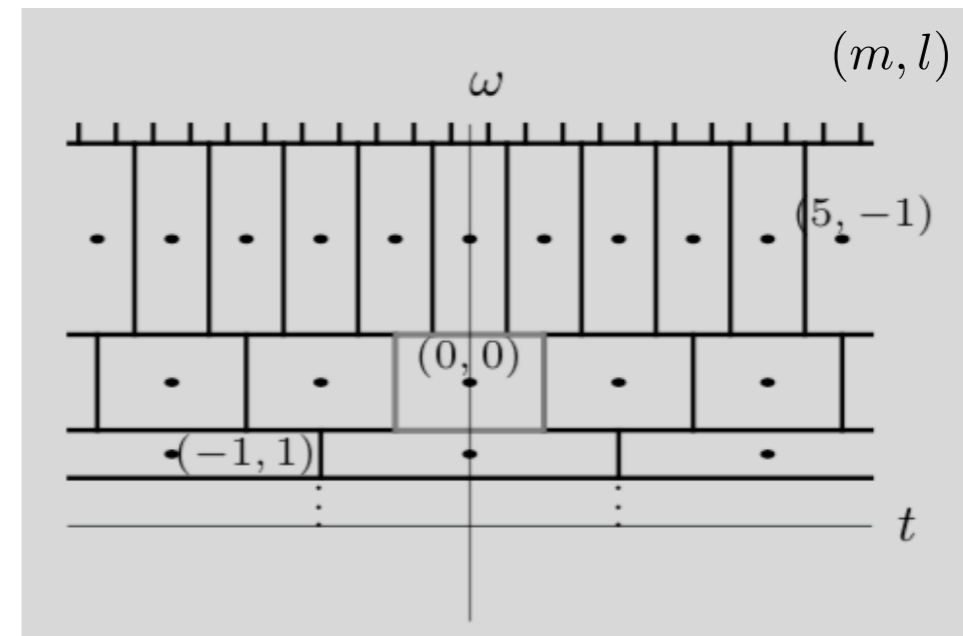


Time-frequency representation

time shift and modulation



time shift and scaling



$$\varphi_{k,m}(t) = e^{jk\omega_0 t} \varphi(t - mt_0) \quad k, m \in \mathbb{Z}$$

$$\varphi_{l,m}(t) = \varphi(a^{-l}t - mt_0) \quad l, m \in \mathbb{Z}$$

↓

$$\varphi(a^{-l}(t - ma^l t_0))$$

1970s-80s: STFT

*discrete
Fourier
transform*

*discrete
cosine
transform*

*Karhunen-
Loève
transform*



1960s

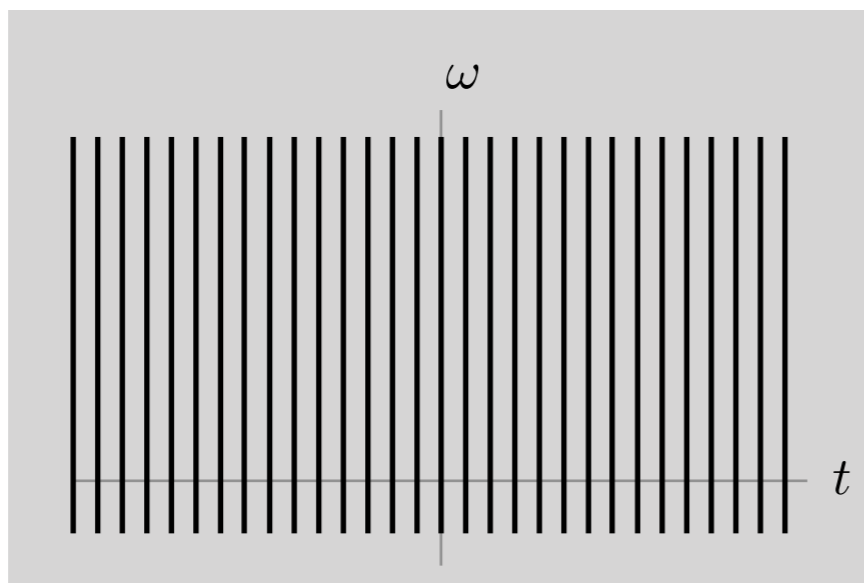
1974

1970s-80s

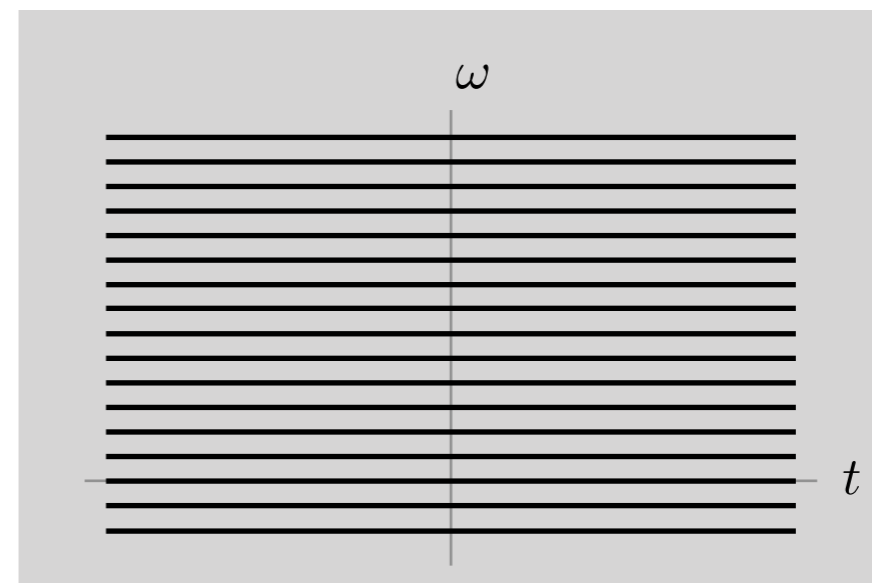


- delta functions are not localised in frequency
- Fourier basis functions (complex exponentials) are not localised in time

time-domain



frequency-domain



1970s-80s: STFT

discrete Fourier transform discrete cosine transform Karhunen-Loève transform short-time Fourier transform

1960s

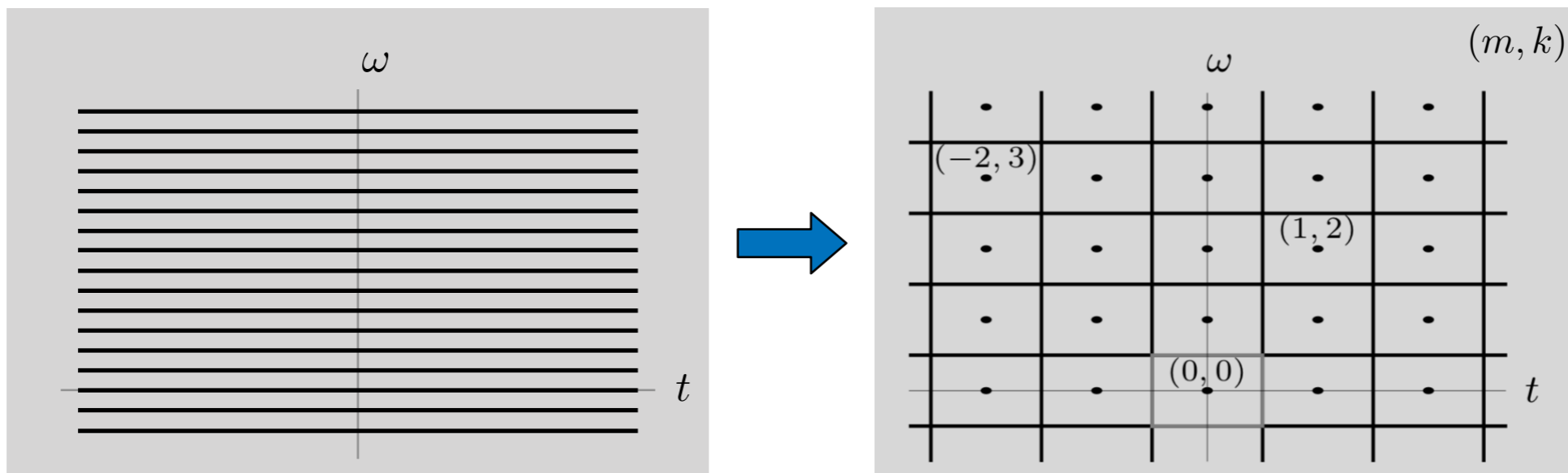
1974

1970s-80s

1970s-80s

- consider a set of shifted and modulated versions of a low-pass function

$$\varphi_{k,m}(t) = e^{jk\omega_0 t} \varphi(t - mt_0) \quad k, m \in \mathbb{Z}$$



1970s-80s: STFT

discrete Fourier transform discrete cosine transform **Karhunen-Loève transform** short-time Fourier transform

1960s

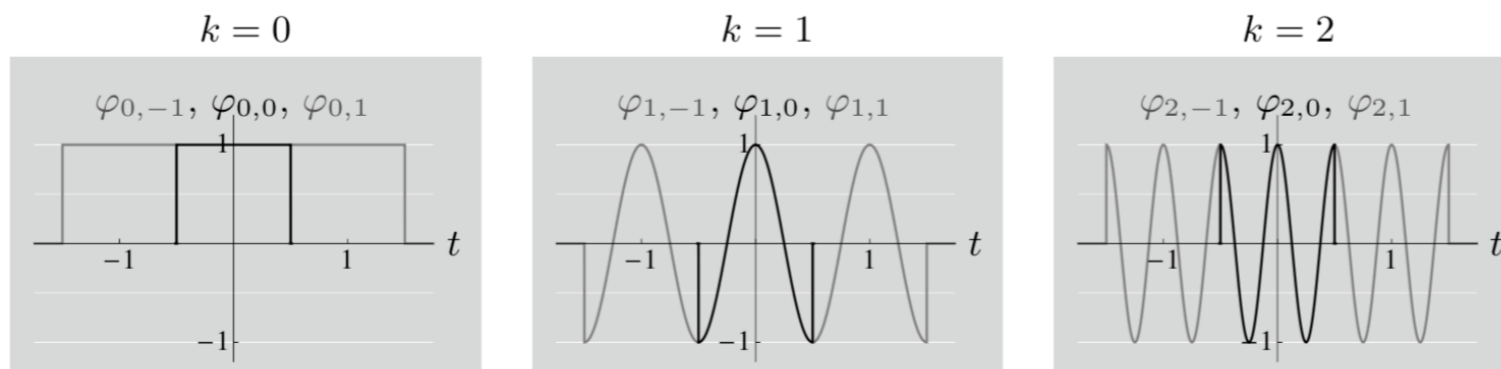
1974

1970s-80s

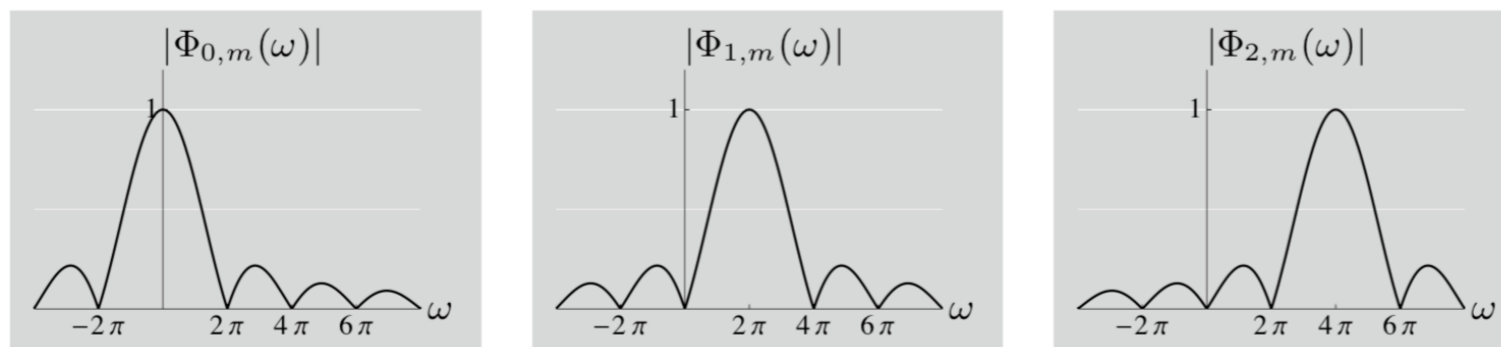
1970s-80s

- example: consider a box function and $t_0 = 1, \omega_0 = 2\pi$

$$\varphi_{k,m}(t) = e^{jk2\pi t} \varphi(t - m), \quad \varphi(t) = \begin{cases} 1, & \text{for } |t| \leq \frac{1}{2}; \\ 0, & \text{otherwise.} \end{cases}$$

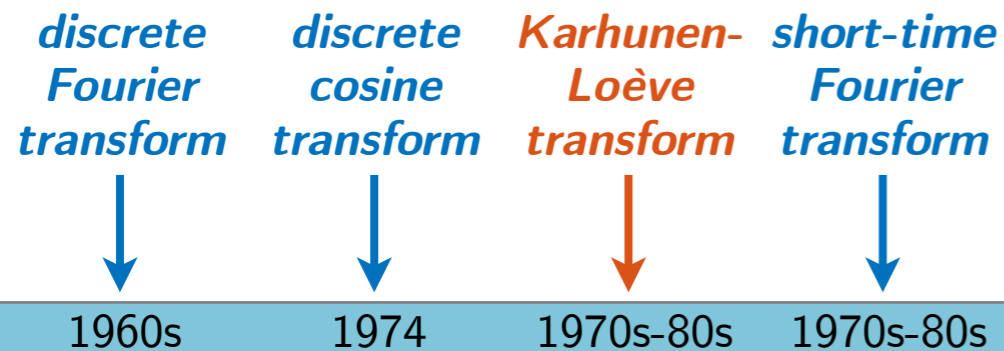


Basis functions (real parts only).



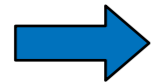
Magnitudes of the Fourier transform.

1970s-80s: STFT



- we can define the following transform

$$X_{k,m} = \int_{-\infty}^{\infty} x(t) \varphi_{k,m}^*(t) dt = \int_{-\infty}^{\infty} x(t) \varphi(t - mt_0) e^{-jk\omega_0 t} dt$$


$$X(\omega, \tau) = \int_{-\infty}^{\infty} x(t) \varphi_{\omega, \tau}^*(t) dt = \int_{-\infty}^{\infty} x(t) \varphi(t - \tau) e^{-j\omega t} dt$$

- applying **time-localised** window to the signal before taking Fourier transform: windowed or short-time Fourier transform (STFT)
- Gaussian window achieves **localisation in frequency**: Gabor transform
- STFT maps a 1-D function into a 2-D function (**redundant**)

DCT vs STFT

discrete Fourier transform *discrete cosine transform* *Karhunen-Loève transform* *short-time Fourier transform*

1960s

1974

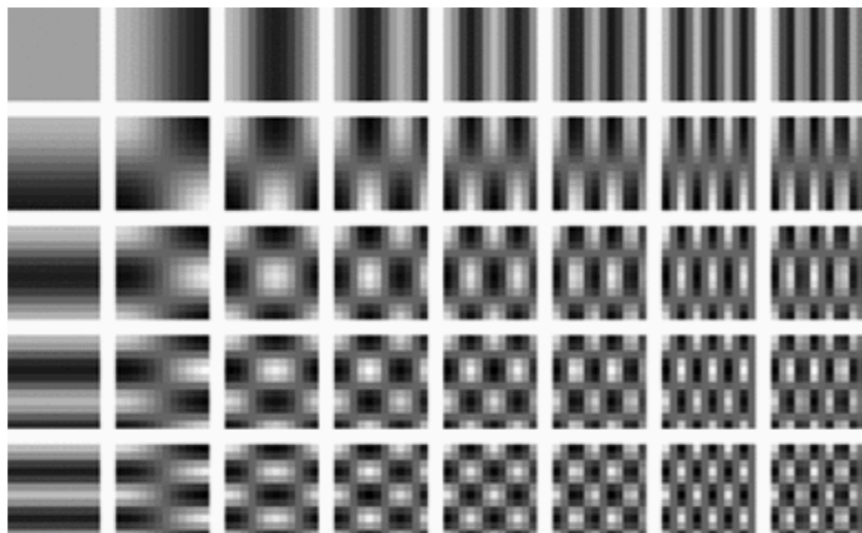
1970s-80s

1970s-80s

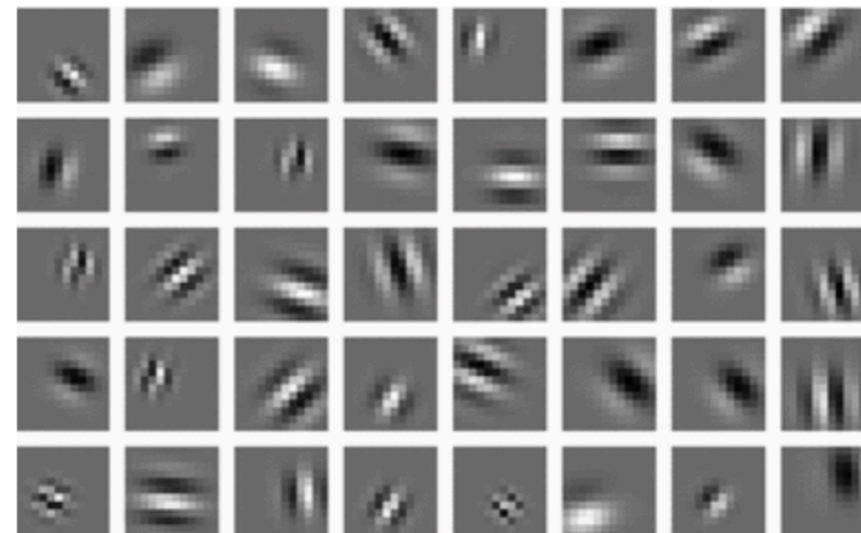
- discrete STFT provides an **over-complete** dictionary

$$\phi_{k,m}(n) = e^{j\frac{2\pi}{N}nk} \varphi(n - mN)$$

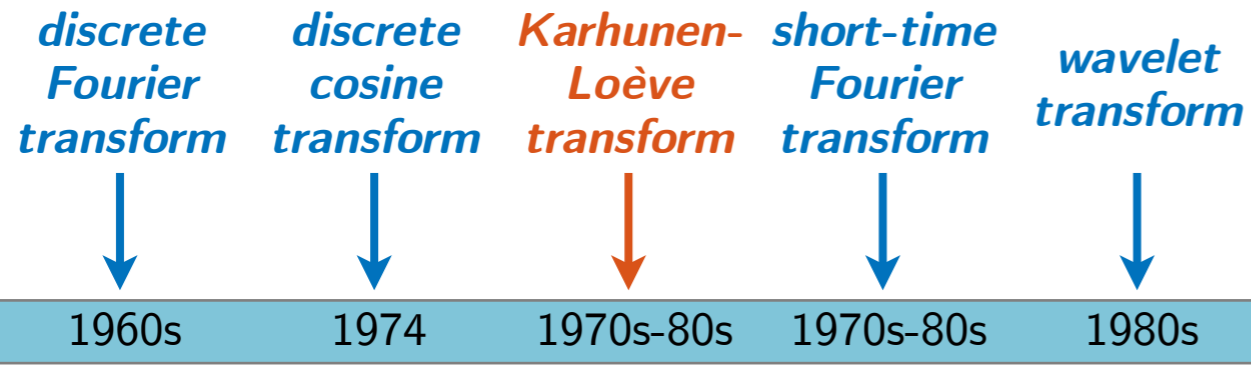
DCT



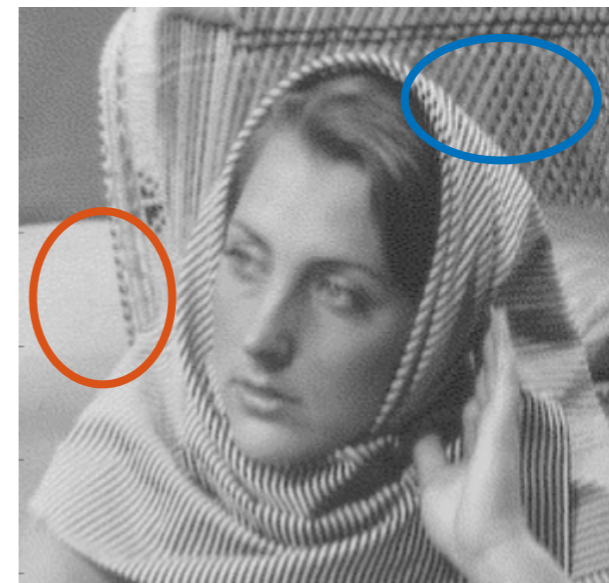
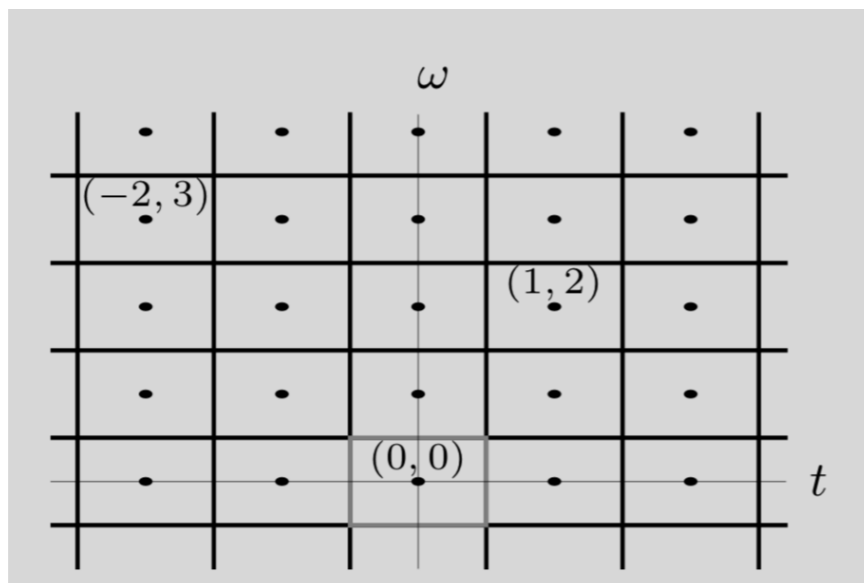
STFT



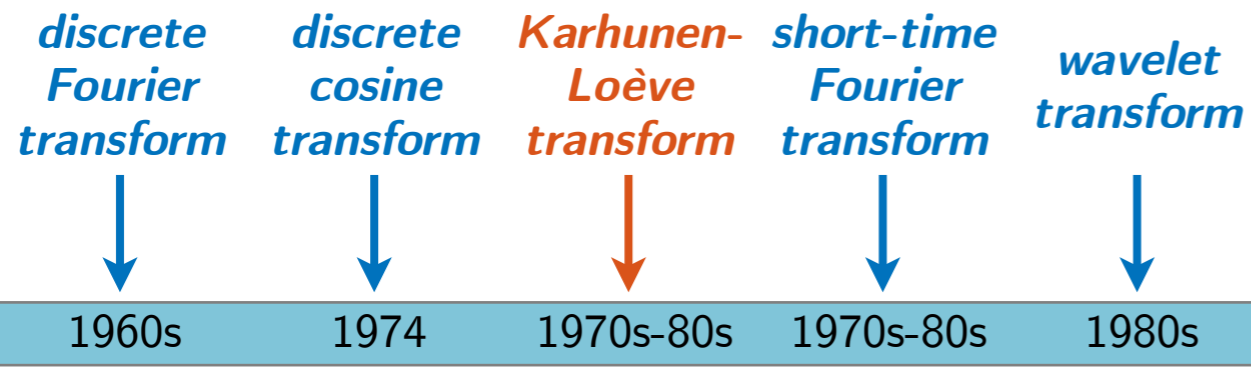
1980s: wavelet transform



- STFT atoms have **fixed time-frequency resolution**
- often times a **multi-resolution** representation is needed to capture various scales in natural signals

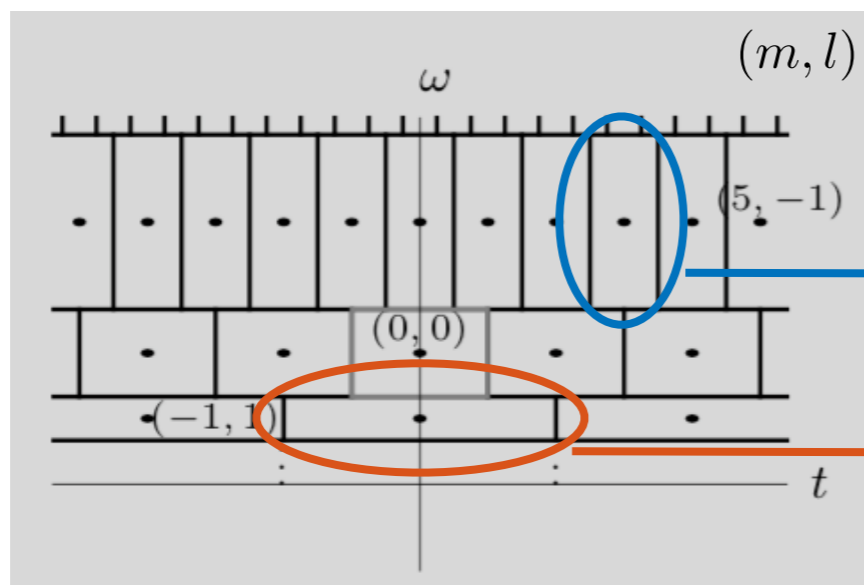


1980s: wavelet transform



- consider a set of shifted and scaled versions of a band-pass function

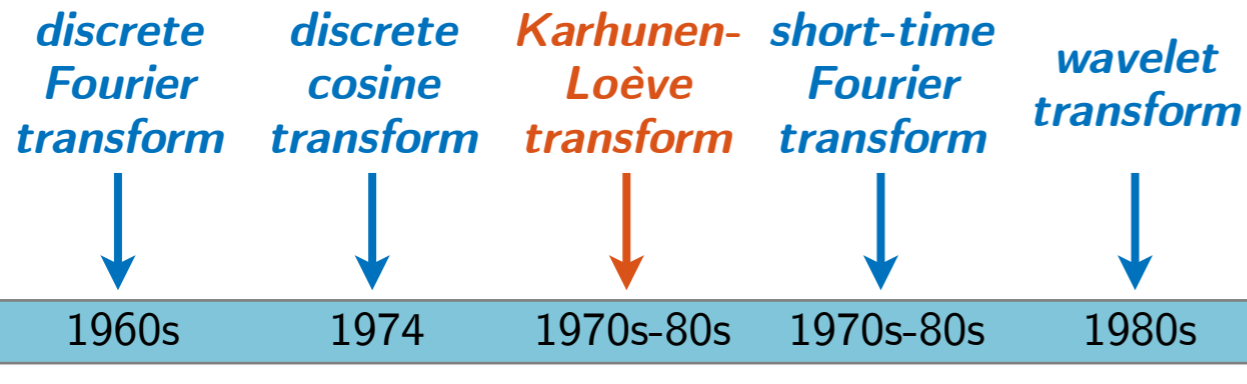
$$\varphi_{l,m}(t) = \varphi(a^{-l}t - mt_0) = \varphi\left(\frac{t - ma^l t_0}{a^l}\right) \quad l, m \in \mathbb{Z}$$



good time resolution (short-term)

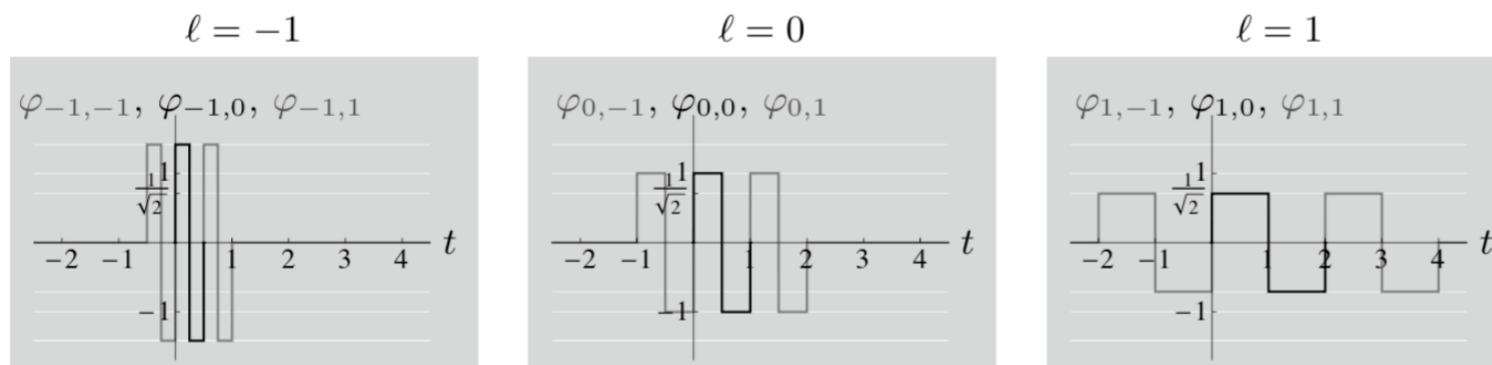
good frequency resolution (long-term)

1980s: wavelet transform

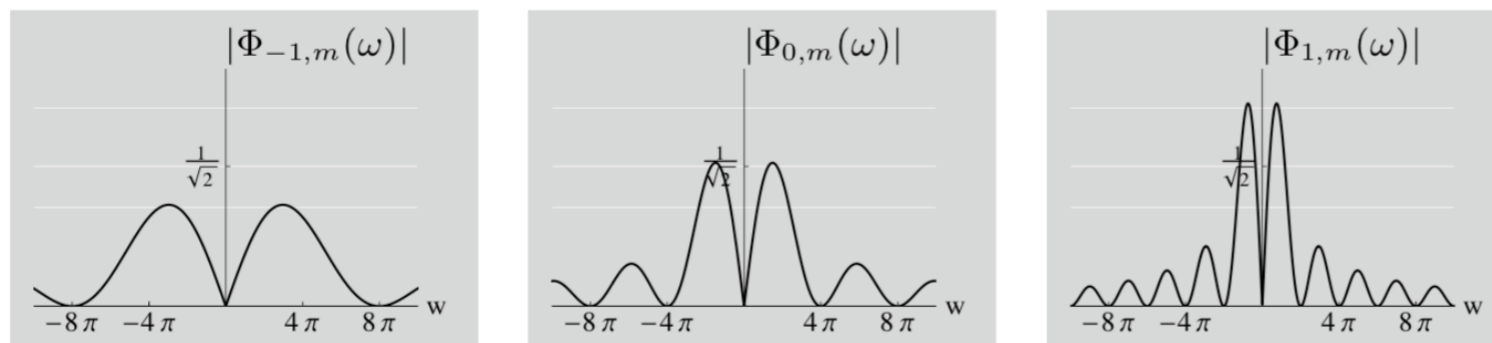


- example: consider a square wave function and $t_0 = 1, a = 2$

$$\varphi_{l,m}(t) = \varphi\left(\frac{t - 2^l m}{2^l}\right), \quad \varphi(t) = \begin{cases} 1, & \text{for } 0 \leq t < \frac{1}{2}; \\ -1, & \text{for } \frac{1}{2} \leq t < 1; \\ 0, & \text{otherwise.} \end{cases}$$

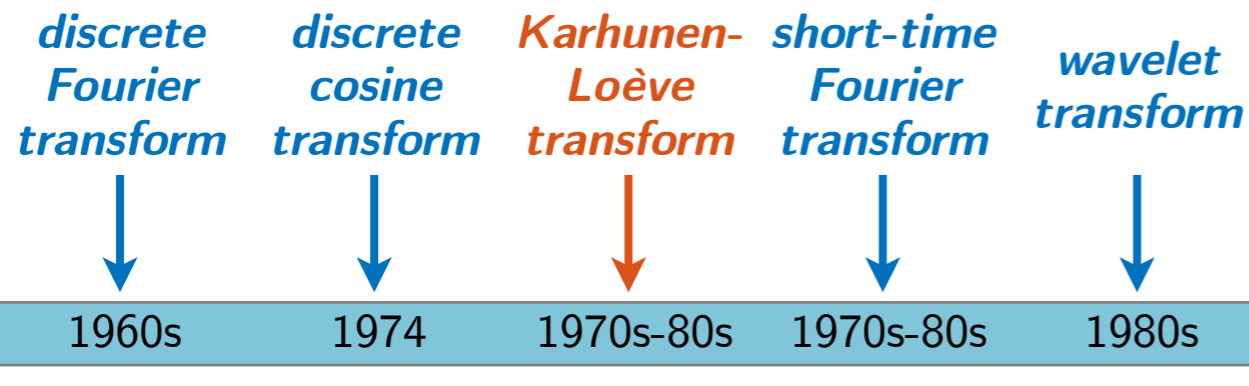


Basis functions.



Magnitudes of the Fourier transform.

1980s: wavelet transform

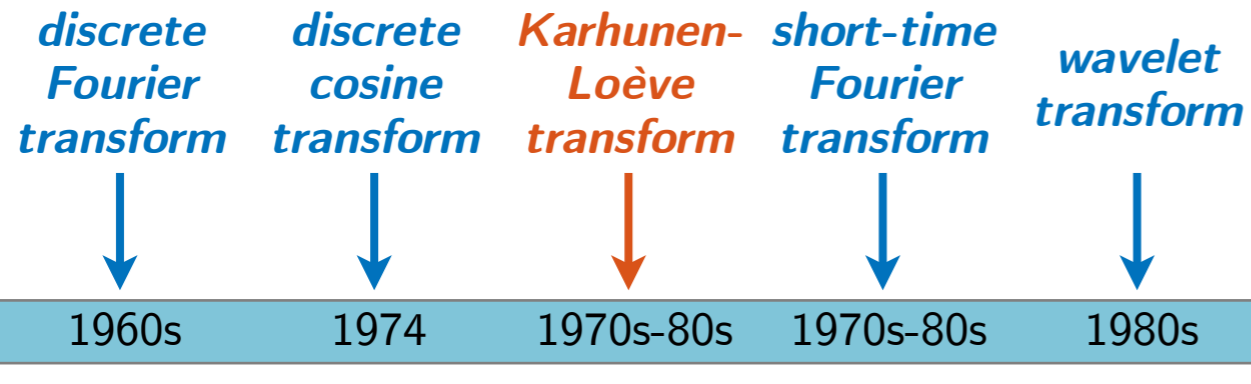


- consider a more general function and define the following transform

$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{s}} \psi\left(\frac{t-\tau}{s}\right) \longrightarrow X(s, \tau) = \int_{-\infty}^{\infty} x(t) \psi_{s,\tau}^*(t) dt = \int_{-\infty}^{\infty} x(t) \frac{1}{\sqrt{s}} \psi^*\left(\frac{t-\tau}{s}\right) dt$$

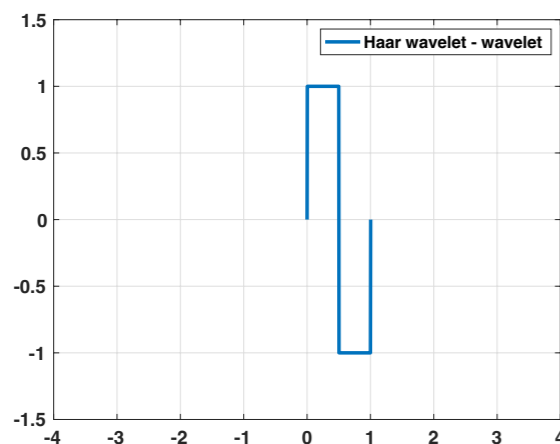
- the prototype function $\psi(t)$
 - has a compact support (small or “-let”)
 - is band-pass with zero mean (“wave”): $\int_{-\infty}^{\infty} \psi(t) dt = 0$
- this is called the **continuous wavelet transform (CWT)**
- CWT maps a 1-D function into a 2-D function (**redundant**)

1980s: wavelet transform

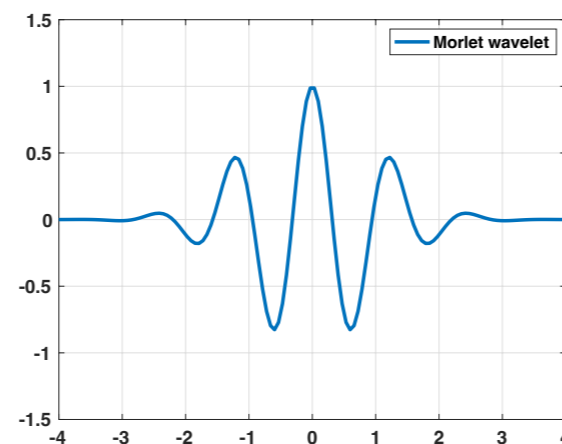


- examples of prototype function (mother wavelet)

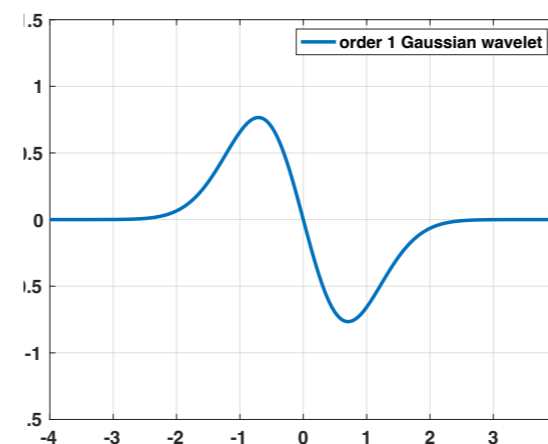
Haar



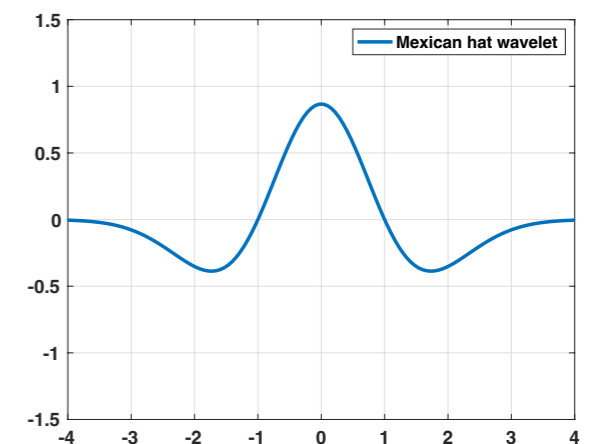
Morlet



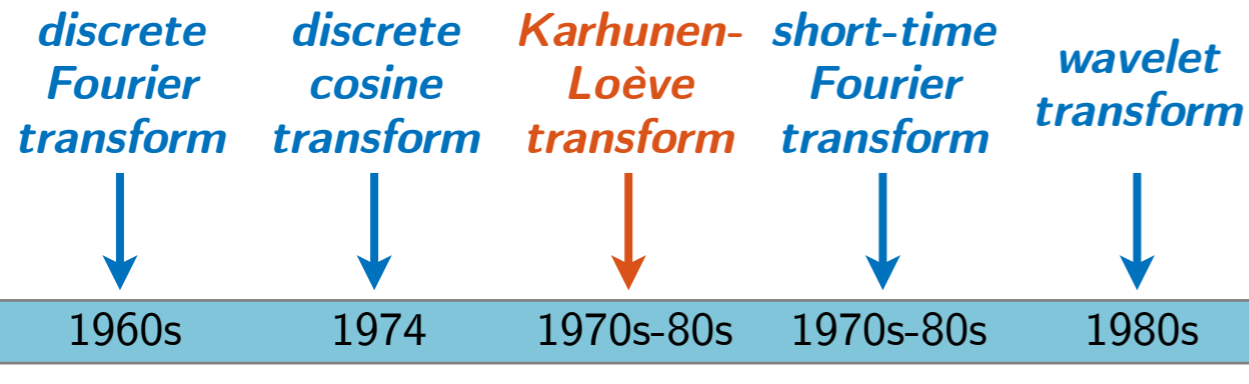
Derivative of Gaussian



Marr (Mexican hat)



1980s: wavelet transform

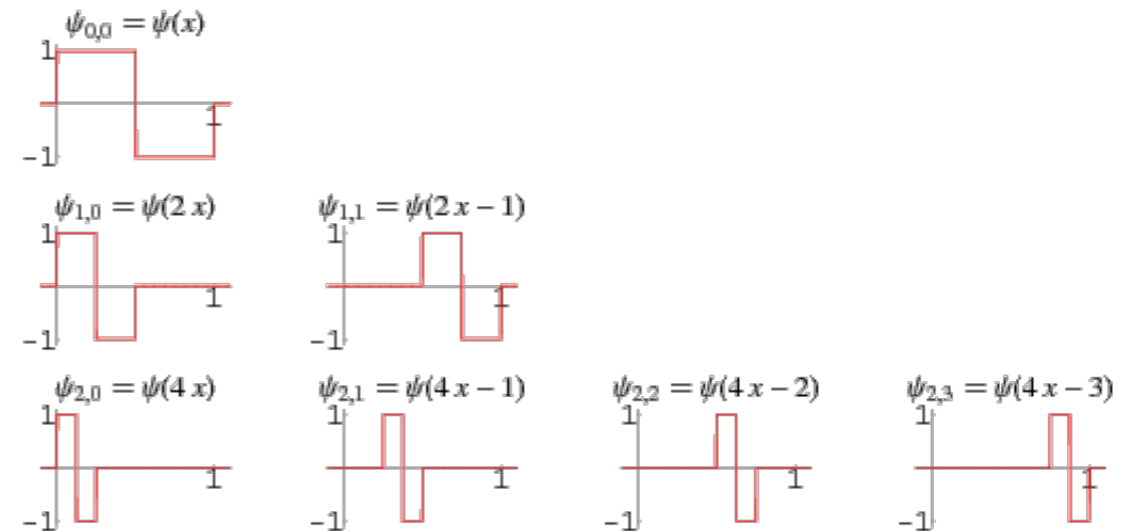


- CWT is a redundant transform; however, unlike STFT, we can design an **orthogonal** wavelet transform through a **multi-resolution analysis**

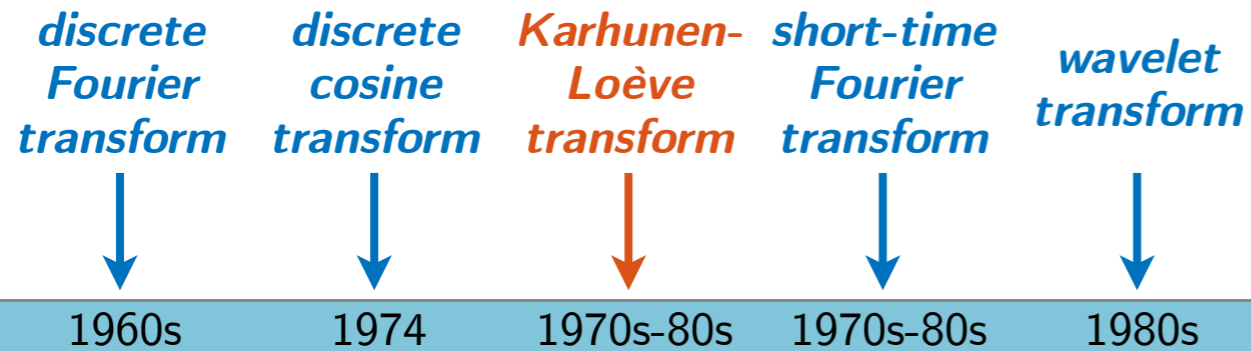
$$\psi_{l,m}(t) = \frac{1}{\sqrt{2^l}} \psi\left(\frac{t - 2^l m}{2^l}\right)$$

design principle

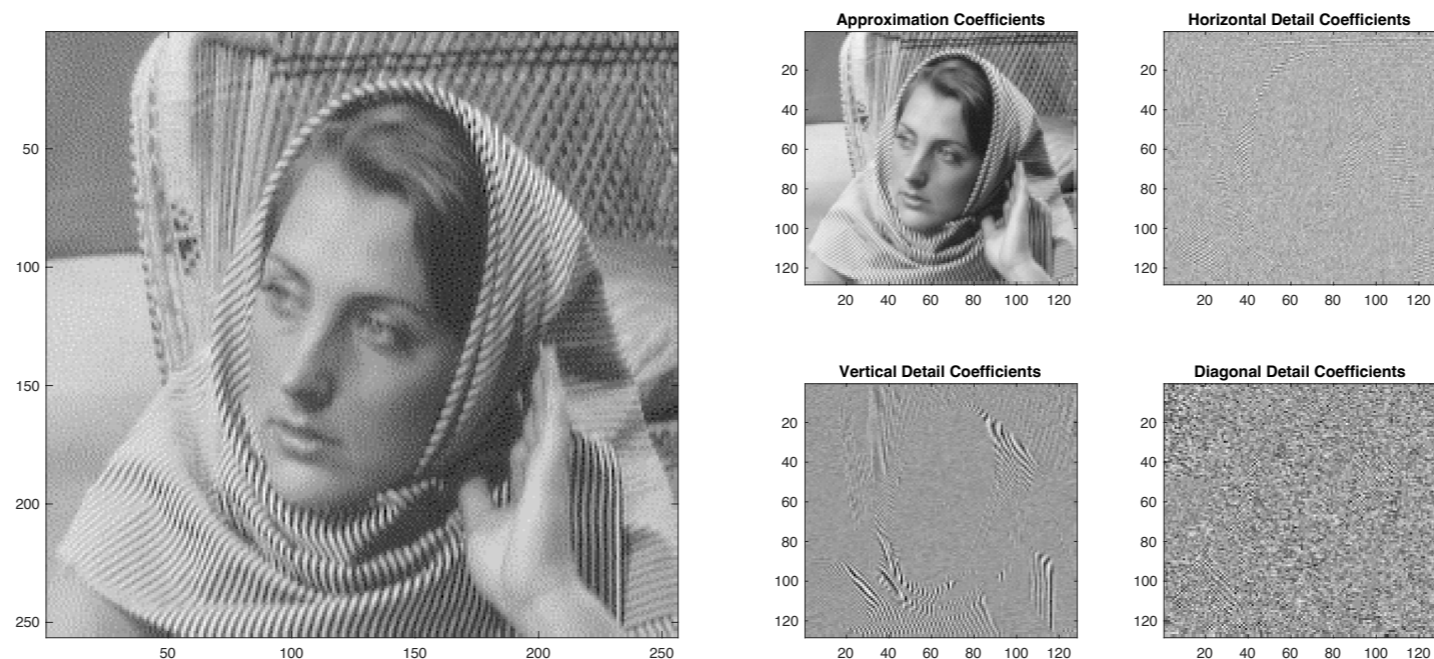
- functions at given scale l form an orthogonal basis of a space at l
- all functions across different scales are also orthogonal



1980s: wavelet transform

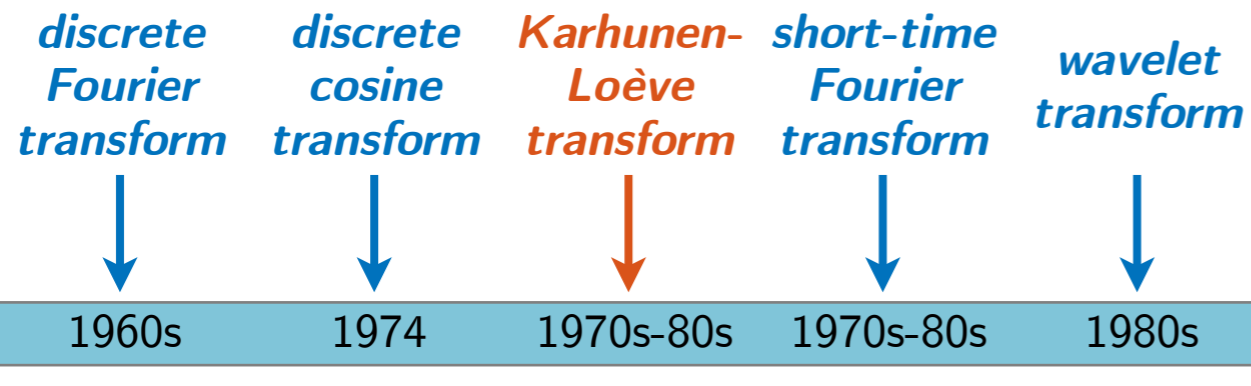


- this leads to the discrete wavelet transform (DWT) which provides an **orthogonal** dictionary

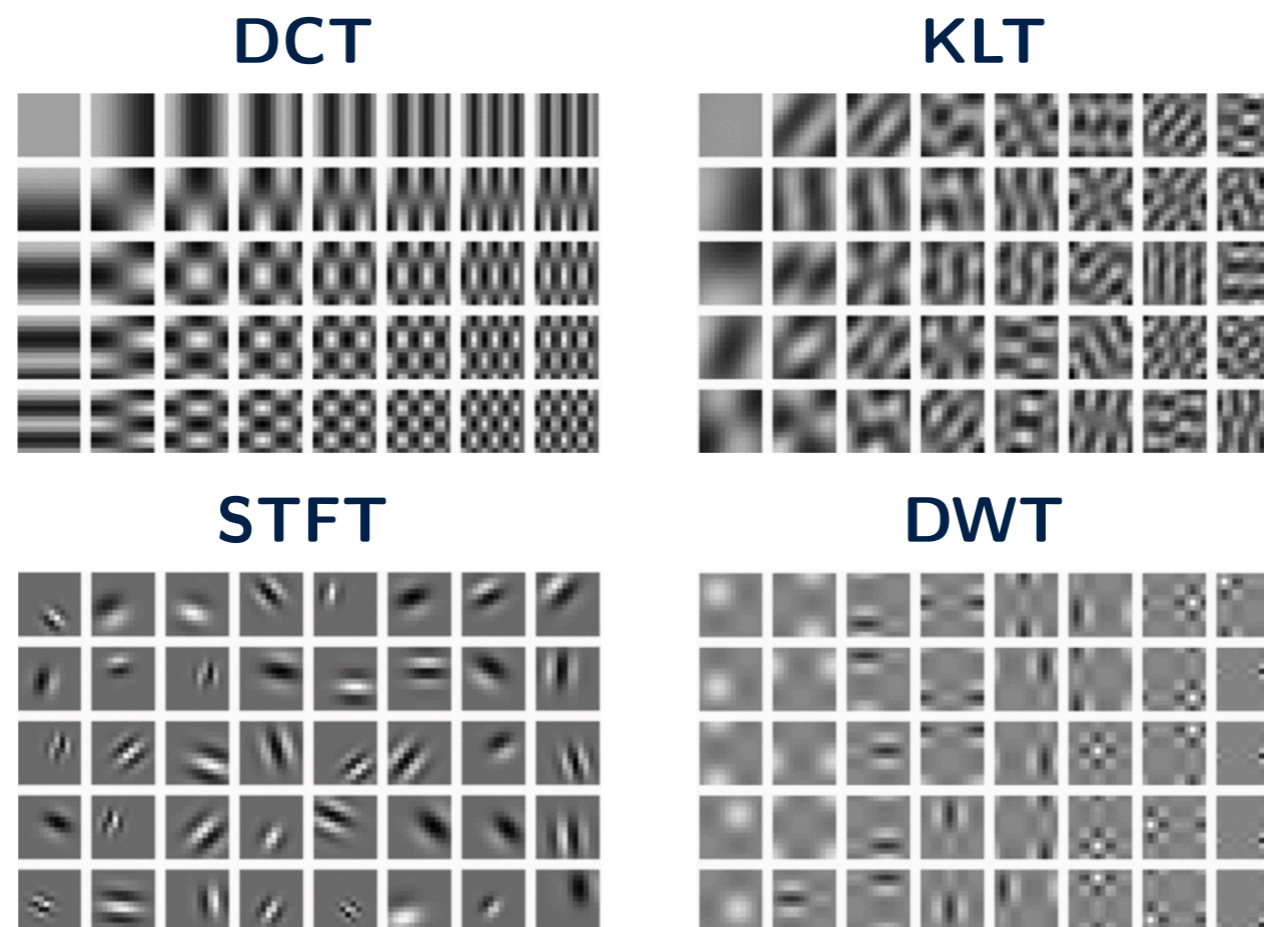


- DWT is behind the JPEG 2000 image compression standard

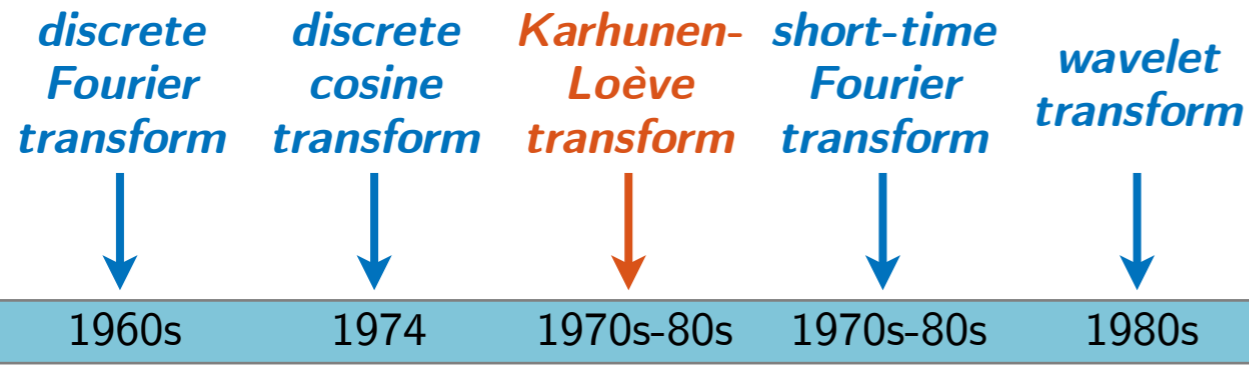
DCT vs KLT vs STFT vs DWT



- comparison of the dictionaries we looked at so far



Transform/analytic dictionary design



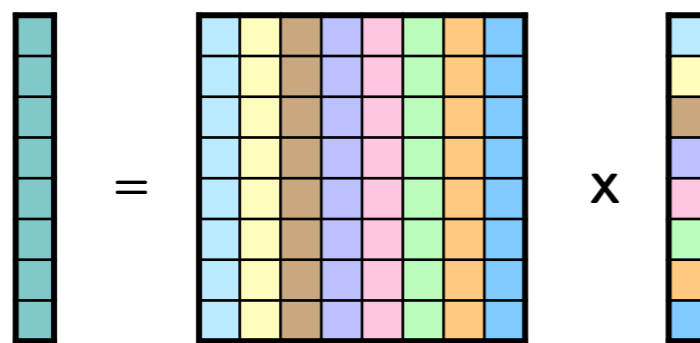
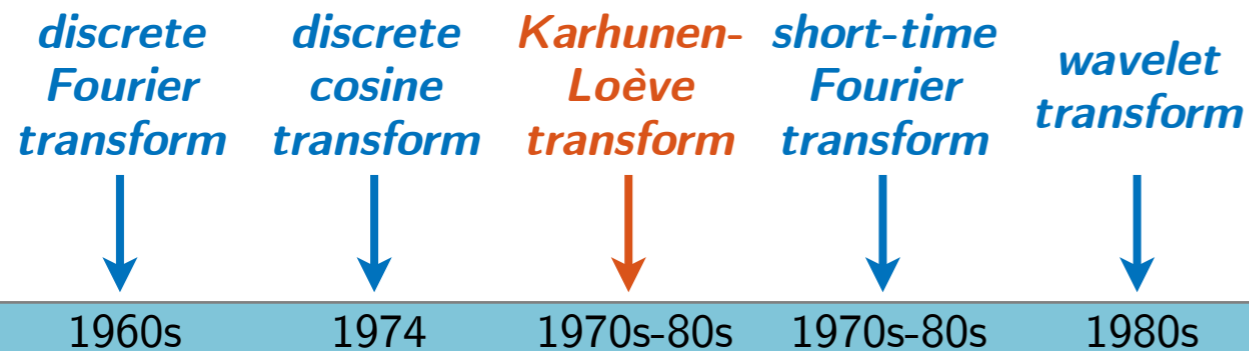
- summary

- modelling data by a simpler class of mathematical functions
 - ◆ smooth functions (DFT, DCT)
 - ◆ piecewise-smooth functions (wavelets)
- desired properties
 - ◆ localisation (STFT, wavelets)
 - ◆ multi-resolution (wavelets)
 - ◆ adaptivity (KLT, wavelet packets)
- fast implementation is usually available
- limited expressiveness

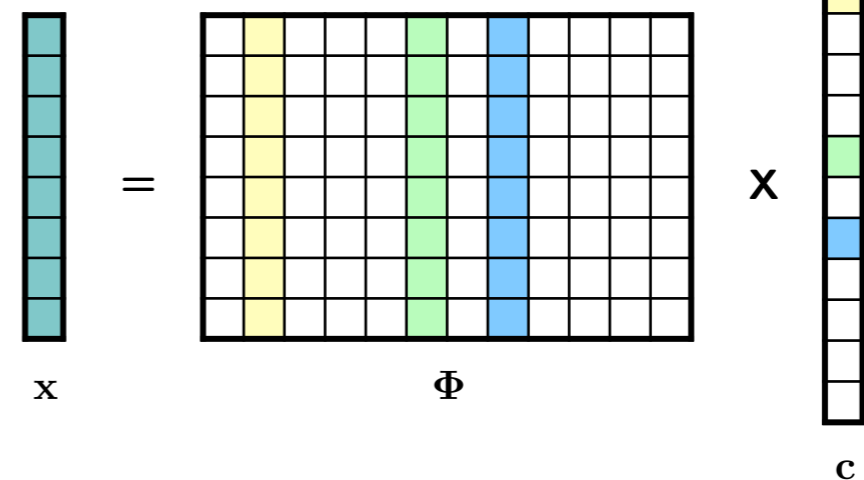
Outline

- A historical overview of dictionary design techniques
 - signal representation via stochastic models
 - transforms & analytic dictionaries
 - trained dictionaries (dictionary learning)
- Discussion
 - applications
 - connection with deep learning

A paradigm shift in dictionary design



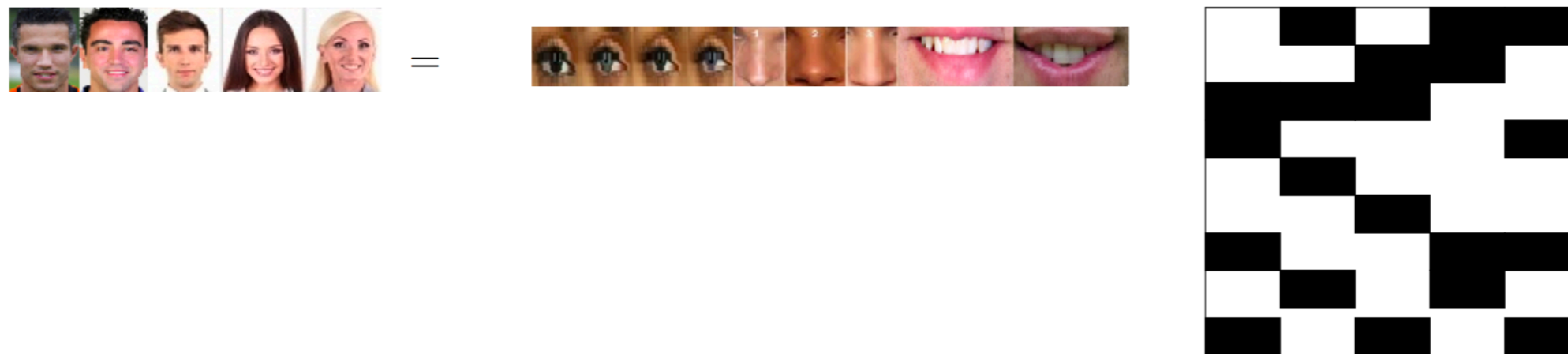
orthogonal atoms
complete dictionary
all signals use all atoms
dense coefficients
mathematical modelling



non-orthogonal atoms
over-complete dictionary
different signals use different atoms
sparse coefficients
adaptation to data realisations

Illustrative example

- **Modelling assumption:** Each data point is a combination of only a few (sparse) fundamental elements, i.e., dictionary atoms

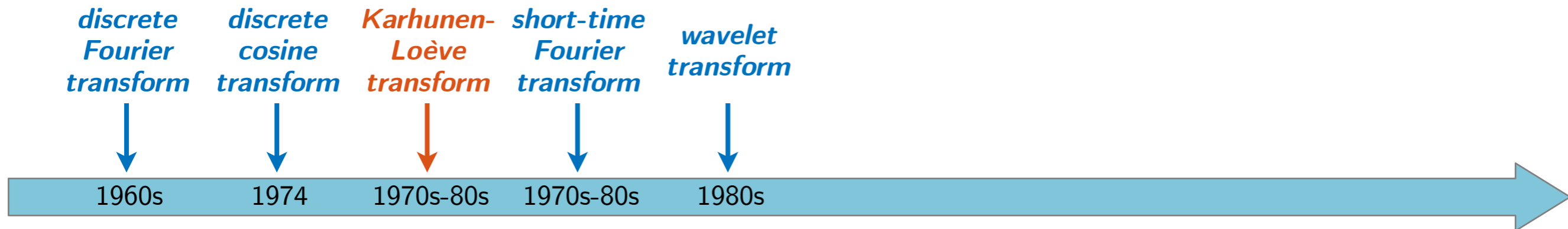


Signals

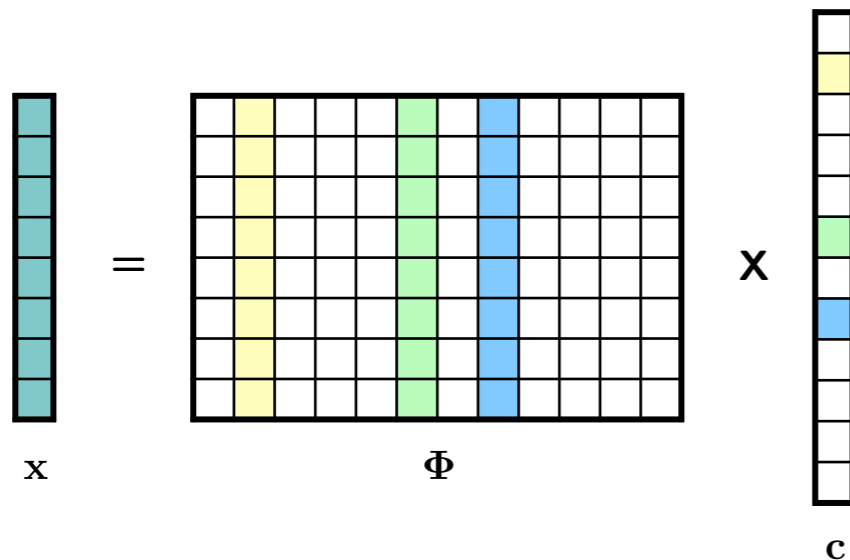
Dictionary

Coefficients

Sparse representations



- given dictionary, express signal as linear combination of a small number of atoms



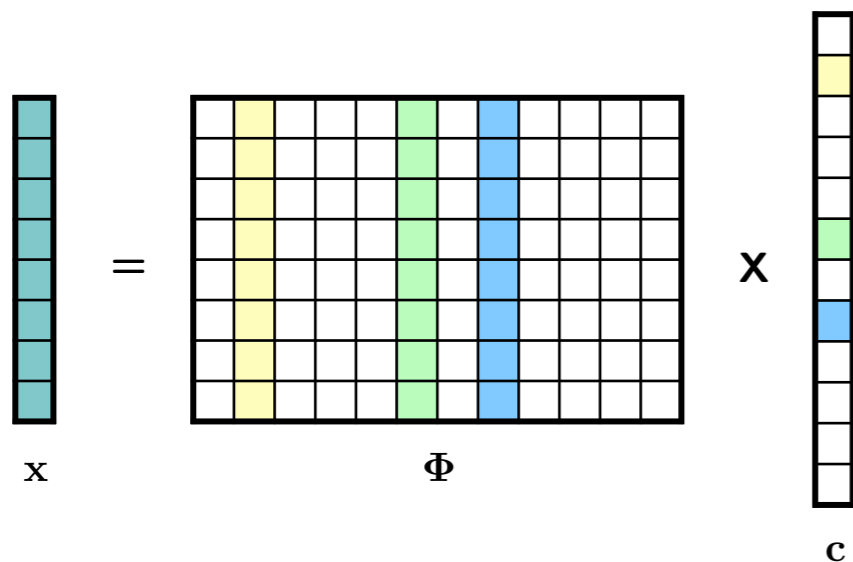
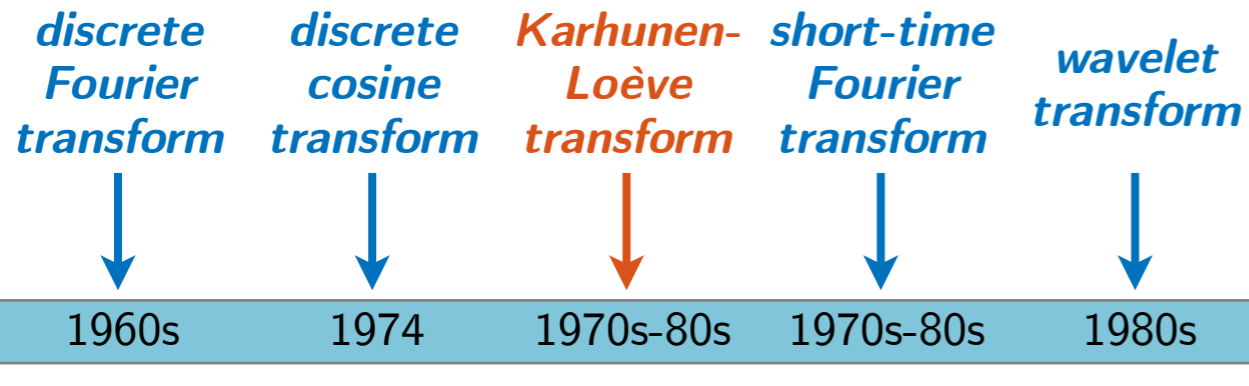
$$\min_c \|c\|_0 \text{ subject to } x = \Phi c + \eta \text{ and } \|\eta\|_2^2 \leq \epsilon$$

- the problem is NP-hard
- two approximation algorithms
 - matching pursuit (MP)
 - least absolute shrinkage and selection operator (Lasso)

Mallat and Zhang, "Matching pursuits with time-frequency dictionaries," IEEE TSP, 1993.

Tibshirani, "Regression Shrinkage and Selection via the Lasso," Journal of the Royal Statistical Society: Series B, 1996.

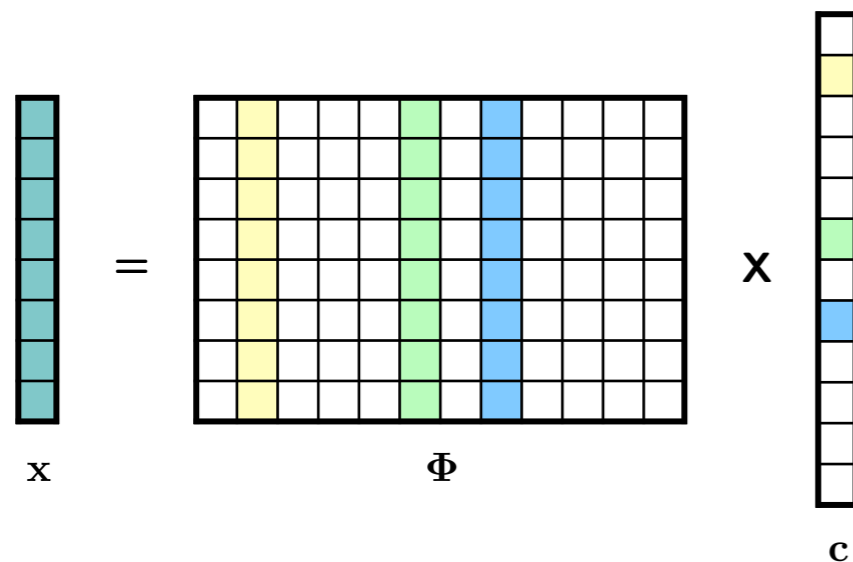
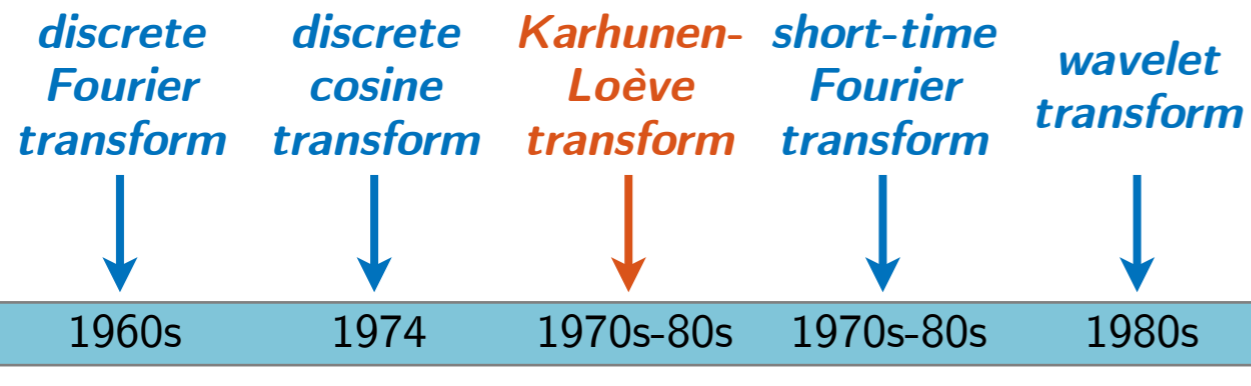
Sparse representations



- MP
 - choose a subset of atoms from Φ
 - one atom at a time to maximally (greedily) reduce approximation error
- Lasso
 - solve a convex relaxation by replacing the 0-norm with 1-norm on \mathbf{c}

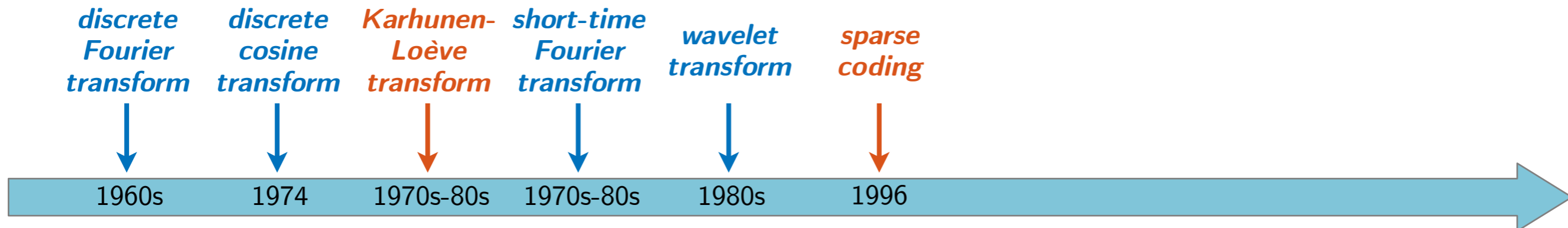
$$\min_{\mathbf{c}} \|\mathbf{x} - \Phi \mathbf{c}\|_2^2 + \lambda \|\mathbf{c}\|_1$$

Sparse representations



- given dictionary, MP and Lasso can find a sparse approximation of the data
- the sparsity depends on not only data but also the **dictionary**
- finding optimised dictionaries is the goal of **dictionary learning**

Dictionary learning: Probabilistic approach



- probabilistic approach: maximum likelihood

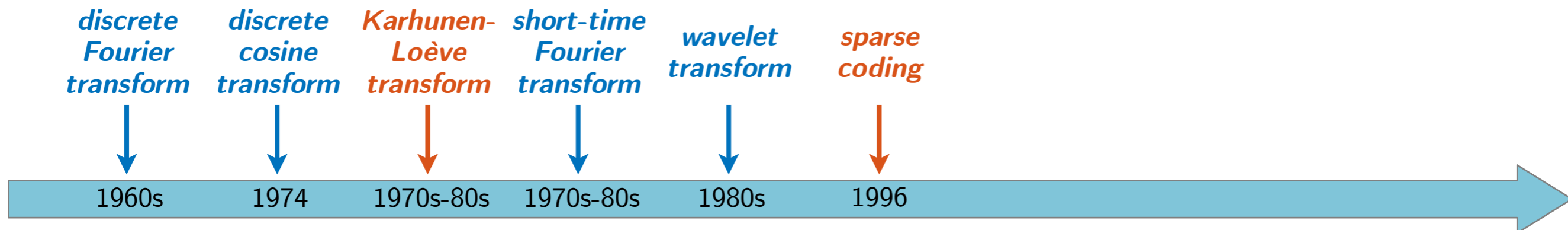
$$\begin{aligned}\Phi^* &= \arg \max_{\Phi} [\log P(\mathbf{x}|\Phi)] \\ &= \arg \max_{\Phi} [\log \int_{\mathbf{c}} P(\mathbf{x}|\mathbf{c}, \Phi) P(\mathbf{c}) d\mathbf{c}]\end{aligned}$$



- assumption 1: Laplace distribution of coefficients \mathbf{c}_i
- assumption 2: Gaussian distribution of error η

$$\begin{aligned}\Phi^* &= \arg \min_{\Phi, \mathbf{c}} -\log [P(\mathbf{x}|\mathbf{c}, \Phi) P(\mathbf{c})] \\ &= \arg \min_{\Phi, \mathbf{c}} \|\mathbf{x} - \Phi \mathbf{c}\|_2^2 + \lambda \|\mathbf{c}\|_1\end{aligned}$$

Dictionary learning: Probabilistic approach

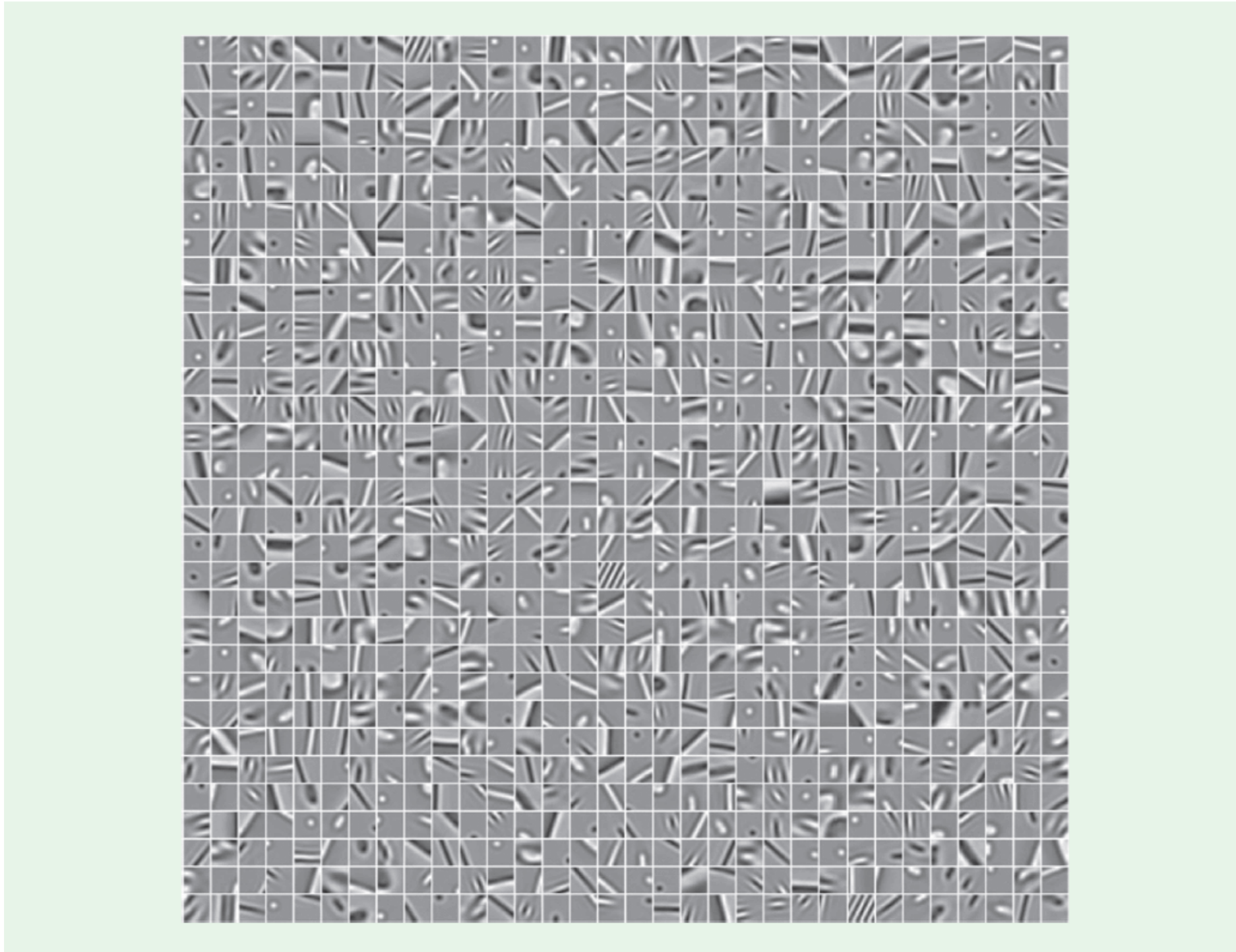


- the problem is solved by iterating between two steps

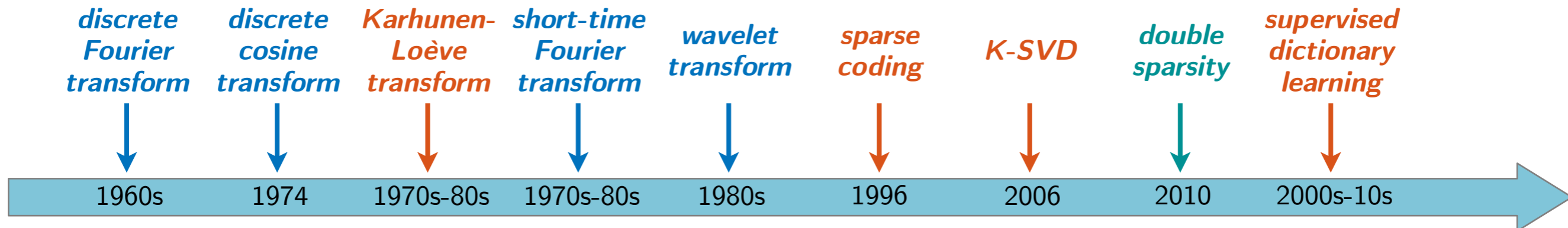
$$\min_{\Phi, \mathbf{c}} \|\mathbf{x} - \Phi \mathbf{c}\|_2^2 + \lambda \|\mathbf{c}\|_1$$

- **sparse approximation:** given Φ , solve for \mathbf{c} via Lasso
- **dictionary update:** given \mathbf{c} , update Φ via gradient descent
- works at patch level for efficiency
- does not necessarily find global optimum
- trained atoms are remarkably similar to mammalian simple-cell receptive fields

Dictionary learned with sparse coding



Dictionary learning



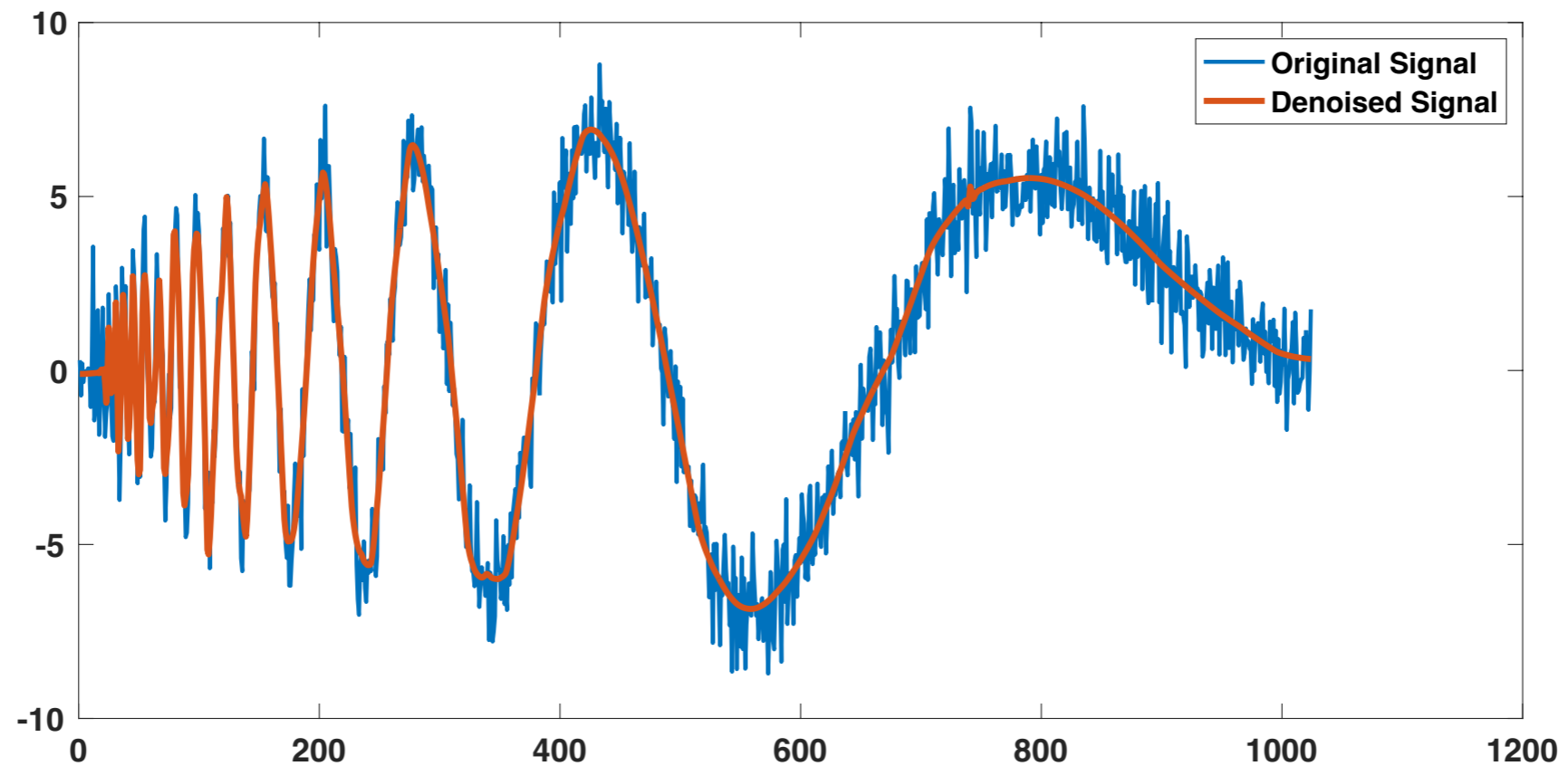
- summary

- learning representations directly from data realisations
- desired properties
 - ◆ over-completeness
 - ◆ sparse representations
 - ◆ efficiency in training
- may be combined with analytical dictionary design
 - ◆ trained dictionary with structures (e.g., parametric dictionary learning)

Outline

- A historical overview of dictionary design techniques
 - signal representation via stochastic models
 - transforms & analytic dictionaries
 - trained dictionaries (dictionary learning)
- Discussion
 - applications
 - connection with deep learning

Signal denoising



denoising using the order 4 symlets wavelets

Image compression

original



JPEG 2000 (10% in size)



JPEG 2000 (1% in size)



compression using the Cohen-Daubechies-Feauveau wavelets

Image reconstruction

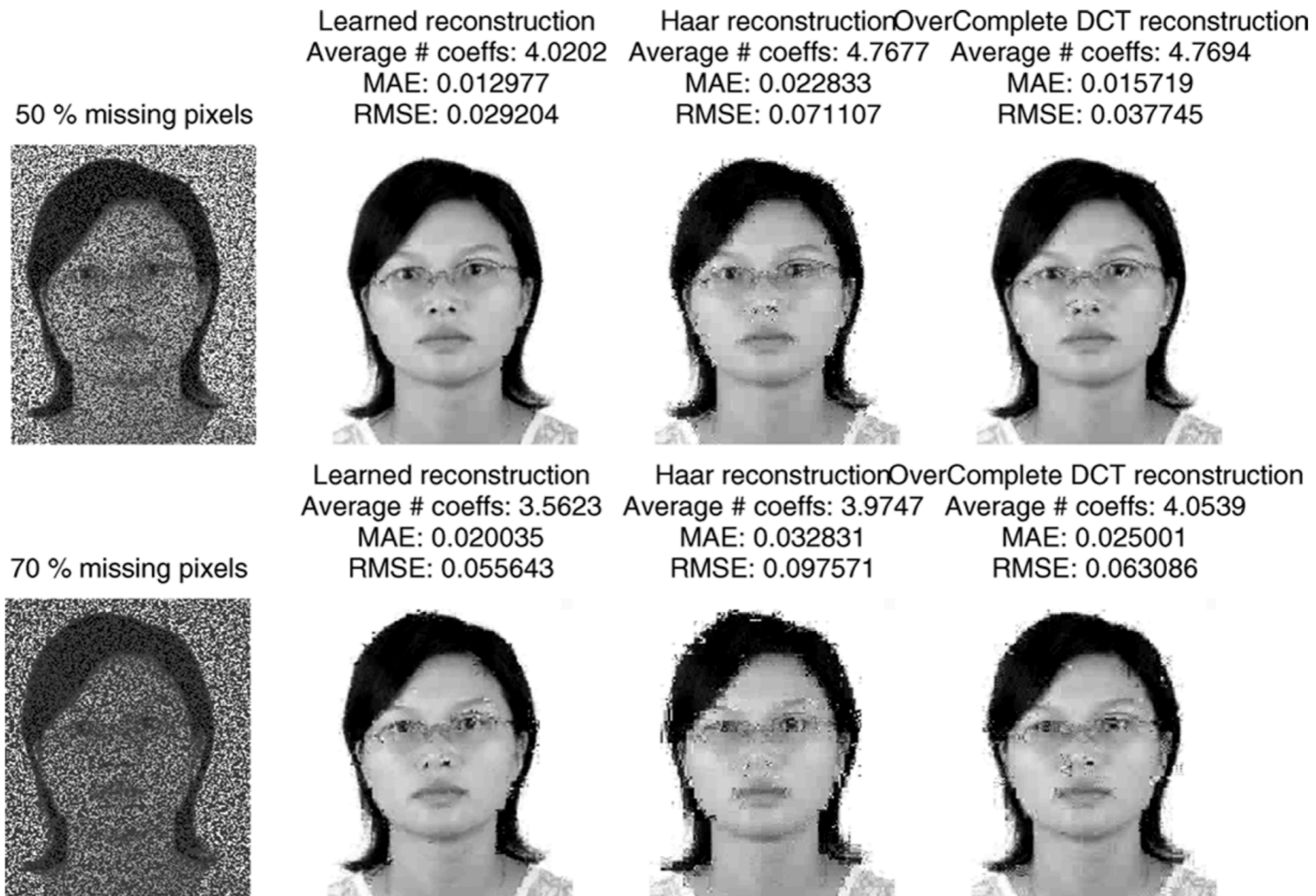
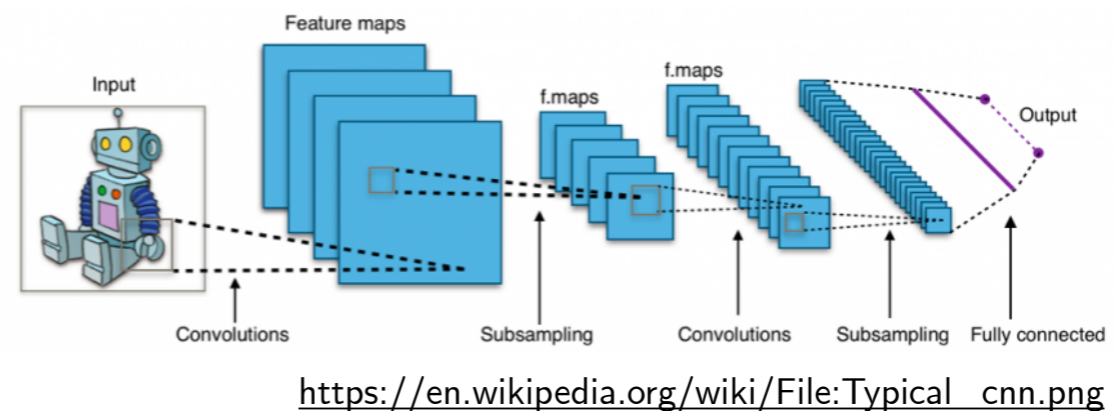
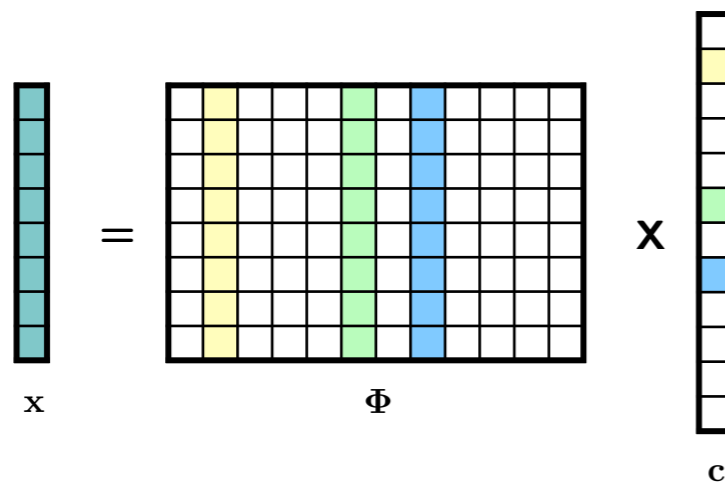


Image restoration



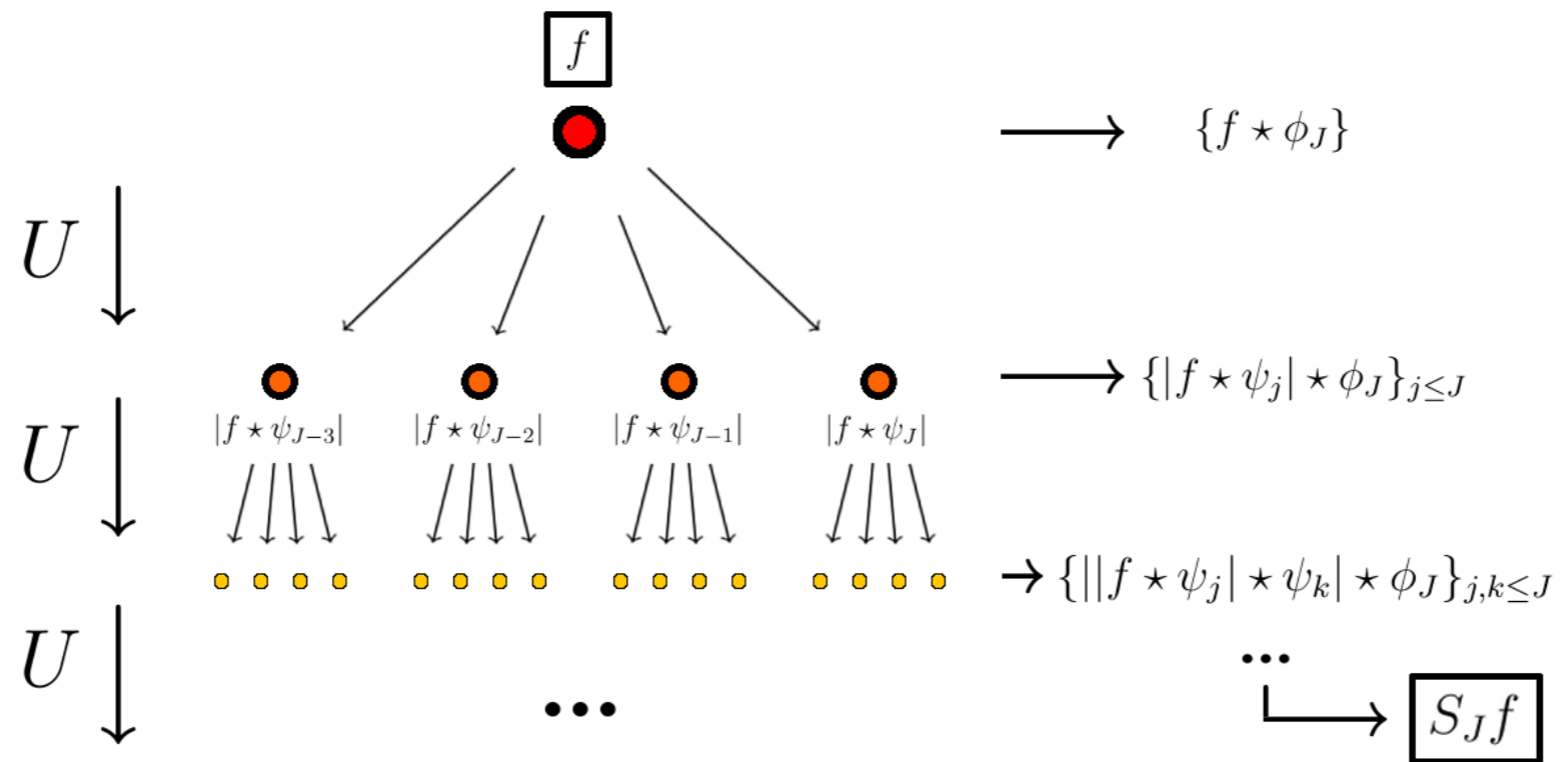
Connection with deep learning

- Dictionary learning vs. Deep learning
 - both extract **feature representations** from data realisations
 - both apply **sparsifying operations** such as shrinkage or rectified linear units
 - the former leads to representations that are not necessarily **hierarchical** (shallow model, no convolution operator, no pooling)
 - the former is normally for **reconstruction/approximation** (similar to autoencoders) while the latter is mainly for **classification**



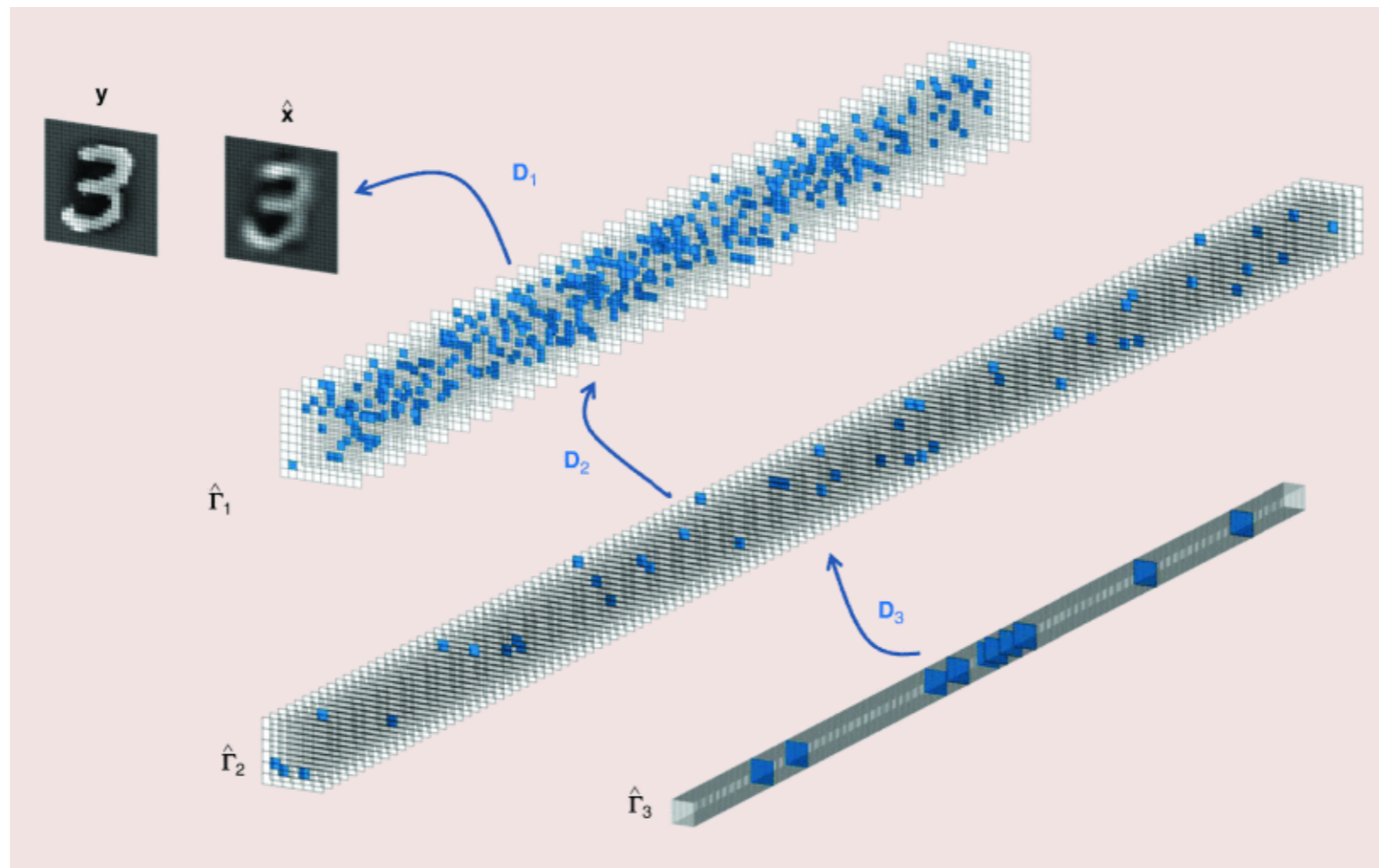
Dictionary-inspired deep architectures

- Scattering transform



Dictionary-inspired deep architectures

- Multi-layer convolutional sparse coding



References



Dictionaries for Sparse Representation Modeling

Digital sampling can display signals, and it should be possible to expose a large part of the desired signal information with only a limited signal sample.

By RON RUBINSTEIN, Student Member IEEE, ALFRED M. BRUCKSTEIN, Member IEEE, AND MICHAEL ELAD, Senior Member IEEE

ABSTRACT Sparse and redundant representation modeling of data assumes an ability to describe signals as linear combinations of a few atoms from a pre-specified dictionary. As such, the choice of the dictionary that sparsifies the signals is crucial for the success of this model. In general, the choice of a proper dictionary can be done using one of two ways: i) building a sparsifying dictionary based on a mathematical model of the data, or ii) learning a dictionary to perform best on a training set. In this paper we describe the evolution of these two paradigms. As manifestations of the first approach, we cover topics such as wavelets, wavelet packets, contourlets, and curvelets, all aiming to exploit 1-D and 2-D mathematical models for constructing effective dictionaries for signals and images. Dictionary learning takes a different route, attaching the dictionary to a set of examples it is supposed to serve. From the seminal work of Field and Olshausen, through the MOD, the K-SVD, the Generalized PCA and others, this paper surveys the various options such training has to offer, up to the most recent contributions and structures.

KEYWORDS Dictionary learning; harmonic analysis; signal approximation; signal representation; sparse coding; sparse representation

1. INTRODUCTION

The process of digitally sampling a natural signal leads to its representation as the sum of Delta functions in space or

Manuscript received April 5, 2009; accepted November 21, 2009. Date of publication April 22, 2010; date of current version May 19, 2010. This research was partly supported by the European Community's FP7-FET program, SMALL project, under grant agreement 229113, and by the ISF grant 599/08. The authors are with the Department of Computer Science, The Technion-Israel Institute of Technology, Haifa 32000, Israel (e-mail: ronrubin@cs.technion.ac.il; fredy@cs.technion.ac.il; elad@cs.technion.ac.il; http://www.cs.technion.ac.il). Digital Object Identifier 10.1109/PROC.2010.2040851.

0018-9219/\$26.00 © 2010 IEEE

time. This representation, while convenient for the purposes of display or playback, is mostly inefficient for analysis tasks. Signal processing techniques commonly require more meaningful representations which capture the useful characteristics of the signal—for recognition, the representation should highlight salient features; for denoising, the representation should efficiently separate signal and noise; and for compression, the representation should capture a large part of the signal with only a few coefficients. Interestingly, in many cases these seemingly different goals align, sharing a core desire for simplification.

Representing a signal involves the choice of a dictionary, which is the set of elementary signals—or atoms—used to decompose the signal. When the dictionary forms a basis, every signal is uniquely represented as the linear combination of the dictionary atoms. In the simplest case the dictionary is orthogonal, and the representation coefficients can be computed as inner products of the signal and the atoms; in the non-orthogonal case, the coefficients are the inner products of the signal and the dictionary inverse, also referred to as the bi-orthogonal dictionary.

For years, orthogonal and bi-orthogonal dictionaries were dominant due to their mathematical simplicity. However, the weakness of these dictionaries—namely their limited expressiveness—eventually outweighed their simplicity. This led to the development of newer overcomplete dictionaries, having more atoms than the dimensions of the signal, which promised to represent a wider range of signal phenomena.

The move to overcomplete dictionaries was done cautiously, in an attempt to minimize the loss of favorable properties offered by orthogonal transforms. Many dictionaries formed tight frames, which ensured that the representation of the signal as a linear combination of the atoms could still be identified with the inner products of the signal and the dictionary. Another approach, manifested by

Ivana Tošić and Pascal Frossard

Dictionary Learning

What is the right representation for my signal?



© DIGITAL STOCK & LUDWIG

Huge amounts of high-dimensional information are captured every second by diverse natural sensors such as the eyes or ears, as well as artificial sensors like cameras or microphones. This information is largely redundant in two main aspects: it often contains multiple correlated versions of the same physical world and each version is usually densely sampled by generic sensors. The relevant information about the underlying processes that cause our observations is generally of much reduced dimensionality compared to such recorded data sets.

The extraction of this relevant information by identifying the generating causes within classes of signals is the central topic of this article. We present methods for determining the proper representation of data sets by means of reduced dimensionality subspaces, which are adaptive to both the characteristics of the signals and the processing task at hand. These representations are based on the principle that our observations can be described by a sparse subset of atoms taken from a redundant dictionary, which represents the causes of our observations of the world. We describe methods for learning dictionaries that are appropriate for the representation of given classes of signals and multisensor data. We further show that dimensionality reduction based on dictionary representation can be extended to address specific tasks such as data analysis or classification when the learning includes a class separability criteria in the objective function. The benefits of dictionary learning clearly show that a proper understanding of causes underlying the sensed world is key to task-specific representation of relevant information in high-dimensional data sets.

WHAT IS THE GOAL OF DIMENSIONALITY REDUCTION?

Natural and artificial sensors are the only tools we have for sensing the world and gathering information about physical processes and their causes. These sensors are usually not aware of the physical process underlying the phenomena they "see," hence they often sample the information with a higher rate than the effective dimension of the process. However, to store, transmit or analyze the processes we observe, we do not need such abundant data: we only need the information that is relevant to understand the causes, to reproduce the physical processes, or to make decisions. In other words, we can reduce the

Digital Object Identifier 10.1109/PROC.2010.2040852
Date of publication: 17 February 2010

IEEE SIGNAL PROCESSING MAGAZINE | 27 | MARCH 2011

1053-5888/11/26-0062011IEEE

1798

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 35, NO. 8, AUGUST 2013

Representation Learning: A Review and New Perspectives

Yoshua Bengio, Aaron Courville, and Pascal Vincent

Abstract—The success of machine learning algorithms generally depends on data representation, and we hypothesize that this is because different representations can entangle and hide more or less the different explanatory factors of variation behind the data. Although specific domain knowledge can be used to help design representations, learning with generic priors can also be used, and the quest for AI is motivating the design of more powerful representation-learning algorithms implementing such priors. This paper reviews recent work in the area of unsupervised feature learning and deep learning, covering advances in probabilistic models, autoencoders, manifold learning, and deep networks. This motivates longer term unanswered questions about the appropriate objectives for learning good representations, for computing representations (i.e., inference), and the geometrical connections between representation learning, density estimation, and manifold learning.

Index Terms—Deep learning, representation learning, feature learning, unsupervised learning, Boltzmann machine, autoencoder, neural nets

1 INTRODUCTION

THE performance of machine learning methods is heavily dependent on the choice of data representation (or features) on which they are applied. For that reason, much of the actual effort in deploying machine learning algorithms goes into the design of preprocessing pipelines and data transformations that result in a representation of the data that can support effective machine learning. Such feature engineering is important but labor intensive and highlights the weakness of current learning algorithms: Their inability to extract and organize the discriminative information from the data. Feature engineering is a way to take advantage of human ingenuity and prior knowledge to compensate for that weakness. To expand the scope and ease of applicability of machine learning, it would be highly desirable to make learning algorithms less dependent on feature engineering so that novel applications could be constructed faster, and more importantly, to make progress toward artificial intelligence (AI). An AI must fundamentally understand the world around us, and we argue that this can only be achieved if it can learn to identify and disentangle the underlying explanatory factors hidden in the observed milieu of low-level sensory data.

This paper is about representation learning, i.e., learning representations of the data that make it easier to extract useful information when building classifiers or other predictors. In the case of probabilistic models, a good representation is often one that captures the posterior

• The authors are with the Department of Computer Science and Operations Research, Université de Montréal, PO Box 6128, Succ. Centre-Ville, Montréal, Québec H3C 3J7, Canada.

Manuscript received 9 Apr. 2012; revised 17 Oct. 2012; accepted 24 Feb. 2013; published online 28 Feb. 2013.

Recommended for acceptance by S. Bengio, L. Deng, H. Larochelle, H. Lee, and R. Salakhutdinov.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2012-04-0260.
Digital Object Identifier no. 10.1109/TPAMI.2013.50.

distribution of the underlying explanatory factors for the observed input. A good representation is also one that is useful as input to a supervised predictor. Among the various ways of learning representations, this paper focuses on deep learning methods: those that are formed by the composition of multiple nonlinear transformations with the goal of yielding more abstract—and ultimately more useful—representations. Here, we survey this rapidly developing area with special emphasis on recent progress. We consider some of the fundamental questions that have been driving research in this area. Specifically, what makes one representation better than another? Given an example, how should we compute its representation, i.e., perform feature extraction? Also, what are appropriate objectives for learning good representations?

2 WHY SHOULD WE CARE ABOUT LEARNING REPRESENTATIONS?

Representation learning has become a field in itself in the machine learning community, with regular workshops at the leading conferences such as NIPS and ICML, and a new conference dedicated to it, ICLR, sometimes under the header of *Deep Learning or Feature Learning*. Although depth is an important part of the story, many other priors are interesting and can be conveniently captured when the problem is cast as one of learning a representation, as discussed in the next section. The rapid increase in scientific activity on representation learning has been accompanied and nourished by a remarkable string of empirical successes both in academia and in industry. Below, we briefly highlight some of these high points.

2.1 Speech Recognition and Signal Processing

Speech was one of the early applications of neural networks, in particular convolutional (or time-delay) neural

1. International Conference on Learning Representations.

0162-8828/13/331-00 © 2013 IEEE
Published by the IEEE Computer Society