Jacobi Algorithm For Nonnegative Matrix Factorization With Transform Learning

Herwig WENDT, Dylan FAGOT, Cédric FEVOTTE

CNRS, IRIT, University of Toulouse, France, `firstname.lastname@irit.fr`

Abstract — Nonnegative matrix factorization (NMF) can be used to decompose a spectrogram $\mathbf{V} \in \mathbb{R}^{M\times N}$ into two nonnegative latent factors $\mathbf{W} \in \mathbb{R}^{M\times K}$ and $\mathbf{H} \in \mathbb{R}^{K\times N}$ which respectively encode spectral patterns (dictionary) and how these are mixed (activation). The results depend strongly on the time-frequency transform used for computing $\mathbf{V}$. **Can we *learn* a transform Φ so that $\mathbf{V}$ can be well approximated using NMF?**

NMF and transform learning

Baseline: IS-NMF

Audio data
$\mathbf{Y} \in \mathbb{R}^{M\times N}$: matrix that contains N adjacent and overlapping short-time frames of width M of the sound sample y

IS-NMF with sparsity

$$\text{Minimize} \quad D(|\Phi_{\text{DCT}}\mathbf{Y}|^{\circ 2}|\mathbf{W}\mathbf{H}) + \lambda\frac{M}{K}||\mathbf{H}||_1$$
$$\text{s.t.} \quad \mathbf{W} \geq 0, \mathbf{H} \geq 0, \ \forall k, ||\mathbf{w}_k||_1 = 1 \qquad (1)$$

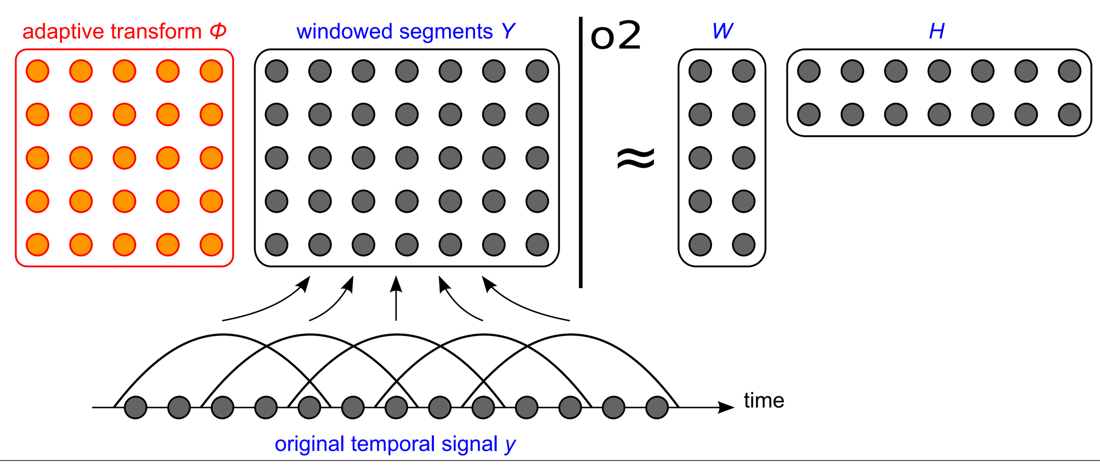
$D(\mathbf{A}|\mathbf{B}) = \sum_{ij}(a_{ij}/b_{ij} - \log(a_{ij}/b_{ij}) - 1)$ Itakura-Saito divergence
Factorization rank K

Transform learning: TL-NMF

$$\text{Minimize} \quad C(\Phi,\mathbf{W},\mathbf{H}) \overset{\text{def}}{=} D(|\Phi\mathbf{Y}|^{\circ 2}|\mathbf{W}\mathbf{H}) + \lambda\frac{M}{K}||\mathbf{H}||_1$$
$$\text{s.t.} \quad \mathbf{W} \geq 0, \mathbf{H} \geq 0, \ \forall k, ||\mathbf{w}_k||_1 = 1, \Phi^T\Phi = \mathbf{I}_M \qquad (2)$$

(inspired from [1])

Orthogonality constraint: $\Phi \in \mathbb{O}^M$
- Mimics commonly used Fourier or DCT transform Φ_{DCT}
- Avoids blow-up & trivial solutions such as $(\Phi,\mathbf{W},\mathbf{H}) = (\mathbf{0},\mathbf{0},\mathbf{0})$



Block-descent algorithm

```
Algorithm 1: TL-NMF-Jacobi
  Input : Y, τ, K, λ
  Output: Φ, W, H
  Initialize Φ = Φ^(0), W = W^(0), H = H^(0) and set l = 1
  while ε > τ do
      H^(l) ← Update H as in [2]                      (U1)
      W^(l) ← Update W as in [2]                      (U2)
      Φ^(l) ← Update Φ using Algorithm 2              (U3)
      Normalize Φ to remove sign ambiguity
      ε = [C(Φ^(l−1),W^(l−1),H^(l−1)) − C(Φ^(l),W^(l),H^(l))] / |C(Φ^(l),W^(l),H^(l))|
      l ← l + 1
  end
```

- In [2]: projected Gradient Descent for (U3)
- **Here: new Jacobi-like iterative approach for update of Φ**

Jacobi algorithm

Principle

Orthogonal matrix in $\mathbb{O}^M$ $\longrightarrow$ product of Givens matrices $\mathbf{R}_{pq}(\theta) \in \mathbb{O}^M$

$$\mathbf{R}_{pq}(\theta) = \begin{array}{cc} & \begin{array}{ccccc} & p & & q & \end{array} \\ \begin{array}{c} \\ p \\ \\ q \\ \\ \end{array} & \begin{bmatrix} \mathbf{I} & 0 & \dots & \dots & 0 \\ 0 & \cos\theta & 0 & -\sin\theta & \vdots \\ \vdots & 0 & \mathbf{I} & 0 & \vdots \\ \vdots & \sin\theta & 0 & \cos\theta & 0 \\ 0 & \dots & \dots & 0 & \mathbf{I} \end{bmatrix} \end{array}. \qquad (3)$$

$M(M-1)/2$ distinct axes of rotation $(p,q) \in \{1,\dots,M\}\times\{1,\dots,M\}$, rotation angle $\theta \in [0, 2\pi[$

Jacobi update: given $\Phi^{(i)}$ and (p,q), with $\mathbf{X}^{(i)} = \Phi^{(i)}\mathbf{Y}$,

$$\Phi^{(i+1)} = \mathbf{R}_{pq}(\hat{\theta})\Phi^{(i)} \qquad (4)$$
$$\hat{\theta} = \arg\min_\theta J_{pq}(\theta) \qquad (5)$$
$$J_{pq}(\theta) \overset{\text{def}}{=} D(|\mathbf{R}_{pq}(\theta)\mathbf{X}^{(i)}|^{\circ 2}|\mathbf{W}\mathbf{H}) \qquad (6)$$

References

[1] S. Ravishankar and Y. Bresler, "Learning sparsifying transforms," *IEEE T. Signal Process.*, 2013.

[2] D. Fagot, H. Wendt, and C. Févotte, "Nonnegative matrix factorization with transform learning," ICASSP 2018.

[3] C. Févotte and J. Idier, "Algorithms for nonnegative matrix factorization with the β-divergence," *Neural Comput.*, 2011.

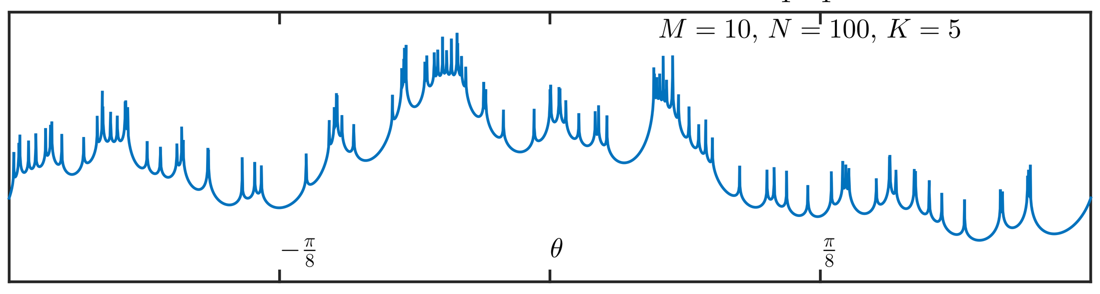
[4] J. H. Manton, "Optimization algorithms exploiting unitary constraints," *IEEE T. Signal Process.*, 2002.

Jacobi update for one axis (p,q)

$$J_{pq}(\theta) = \sum_n \left[\frac{(\cos\theta x_{pn}^{(i)} - \sin\theta x_{qn}^{(i)})^2}{\hat{v}_{pn}} + \frac{(\sin\theta x_{pn}^{(i)} + \cos\theta x_{qn}^{(i)})^2}{\hat{v}_{qn}}\right.$$
$$\left. - 2\log\left[(\cos\theta x_{pn}^{(i)} - \sin\theta x_{qn}^{(i)})(\sin\theta x_{pn}^{(i)} + \cos\theta x_{qn}^{(i)})\right]\right] + cst \qquad (7)$$

where $\hat{v}_{mn} \overset{\text{def}}{=} [\mathbf{W}\mathbf{H}]_{mn}$

$\longrightarrow$ non-convex, non-smooth, N poles for $\theta \in (-\frac{\pi}{4}, \frac{\pi}{4}]$



Randomized grid search
- draw N_{prop} random proposals $\tilde{\theta} \in (-\frac{\alpha\pi}{4}, \frac{\alpha\pi}{4}]$ with $\alpha \in (0,1]$
- approximate (5) as $\hat{\theta} \approx \arg\min_{\theta\in\{\tilde{\theta}_i\}_{i=1}^{N_{\text{prop}}}} J_{pq}(\theta)$

$\rightarrow$ no decrease of (2): move on to a different axis (p,q)
$\rightarrow$ decrease of (2): repeat for smaller value of α

Update for Φ (U3)

- Can find $\frac{M}{2}$ mutually independent random couples (p,q) (rotation set) $\longrightarrow$ update in parallel
- Repeat $K(=\lfloor 2R/M \rfloor)$ times for each update (U3)
- α sequentially updated as

$$\alpha = \alpha(k,l) = l^{-a_1}k^{-a_2}$$

where l is outer loop iteration (Algo. 1) and $k = 1,\dots,K$
$\rightarrow$ plays role similar to a step size parameter

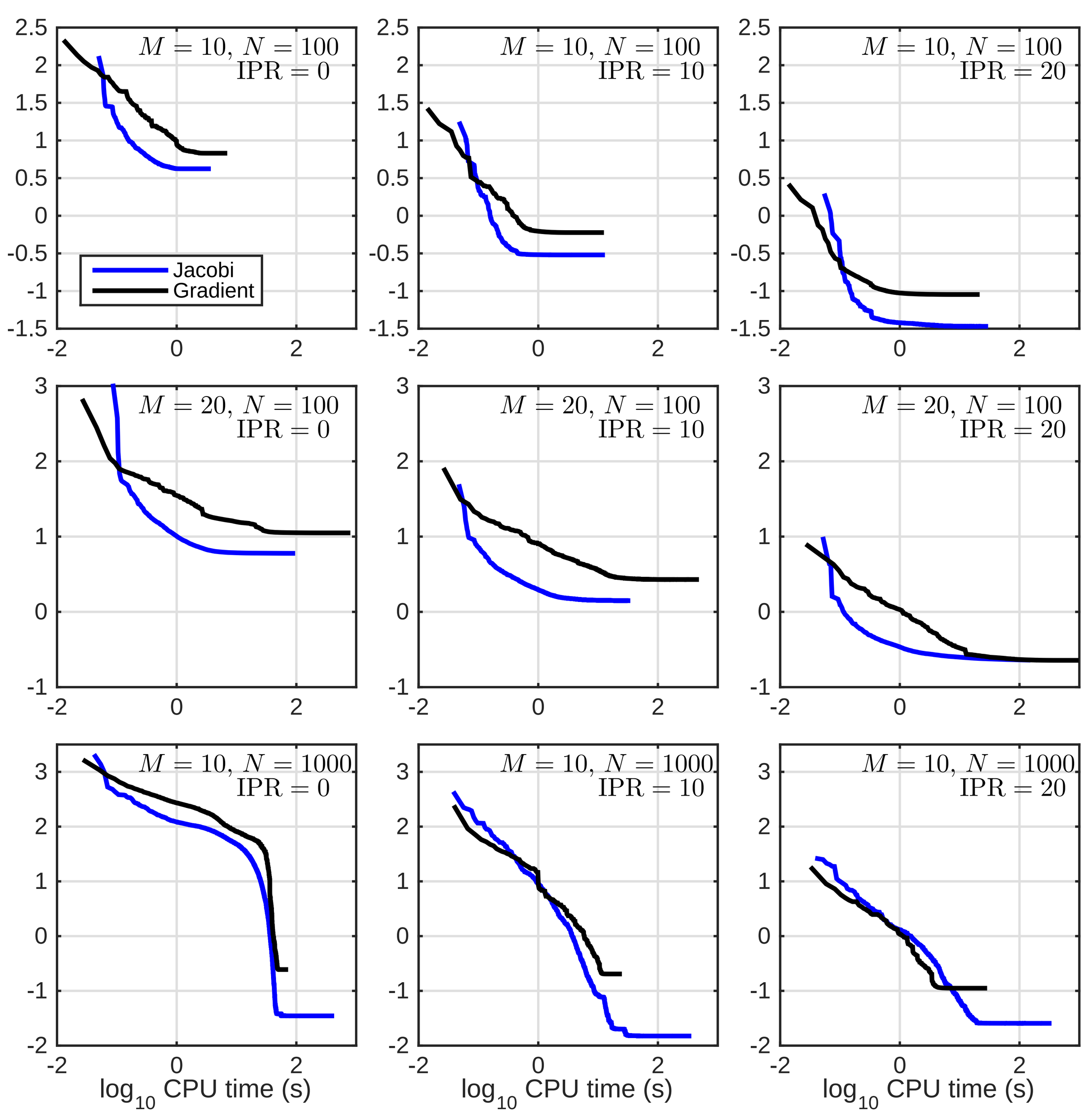
```
Algorithm 2: Jacobi update of Φ at iteration l
  Input : Φ, X = ΦY, V̂ = WH, N_prop, R, l
  Output: Φ
  for k = 1, ..., ⌊2R/M⌋ do
      Generate a random permutation of (1, ..., M) in u
      for j = 1, ..., M/2 do
          (p, q) = (u_j, u_{j+M/2})
          for s = 1, ..., N_prop do
              Draw at random θ̃ ∈ (−α(k,l)π/4, α(k,l)π/4]
              Evaluate J_pq(θ̃) = D(|R_pq(θ̃)X|^∘2 | V̂)
          end
          θ̂ ← arg min_θ̃ J_pq(θ̃)
          if D(|R_pq(θ̂)X|^∘2 | V̂) < D(|X|^∘2 | V̂) then
              Update transform Φ ← R_pq(θ̂)Φ
          end
      end
  end
```

Numerical experiments

Synthetic data — Update (U3) only

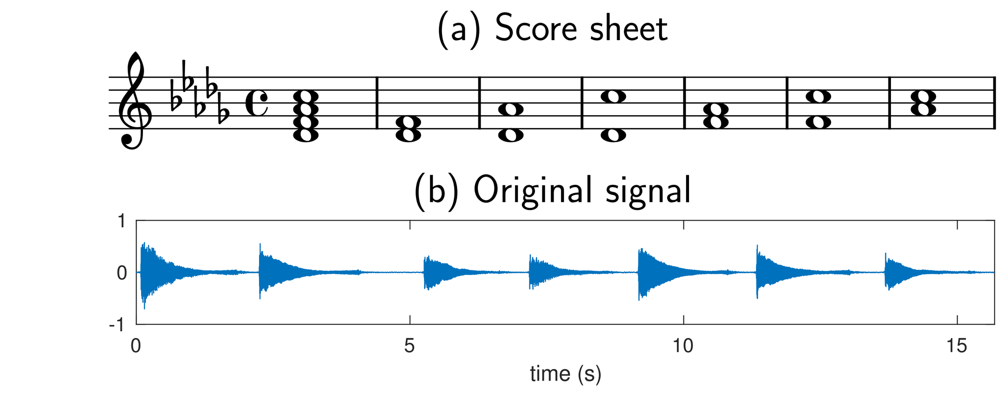
- Minimize $\quad F(\Phi) = D(|\Phi\mathbf{Y}|^{\circ 2}|\mathbf{V}^*) \quad$ s.t. $\quad \Phi \in \mathbb{O}^M$
  where $\mathbf{V}^* = |\Phi^*\mathbf{Y}|^{\circ 2}$

Random data $\mathbf{Y} \in \mathbb{R}^{M\times N}$ and random transform $\Phi^* \in \mathbb{O}^M$,
Random initialization $\Phi = \Phi^{(0)}$ in vicinity of ground-truth Φ^*:
$$\text{IPR} = 10\log_{10}\frac{||\Phi^*||_F^2}{||\Phi^{(0)} - \Phi^*||_F^2}$$



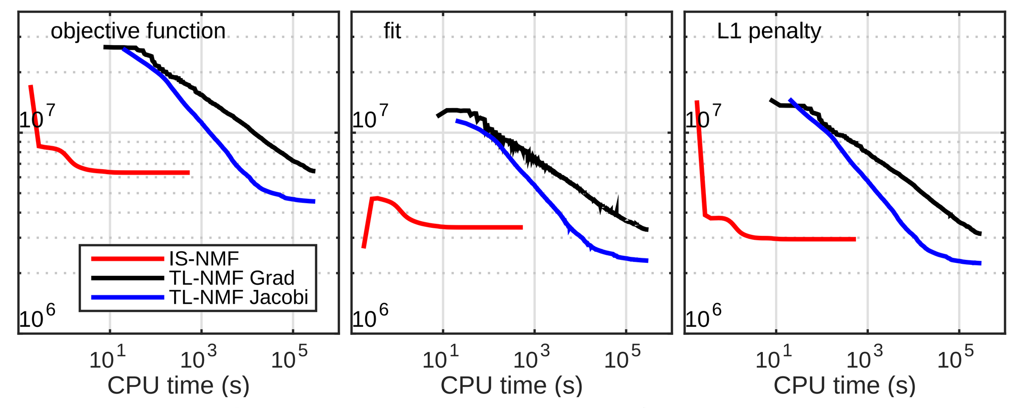
$\rightarrow$ smaller objective function value than gradient descent
$\rightarrow$ faster decrease of objective function value

Audio data



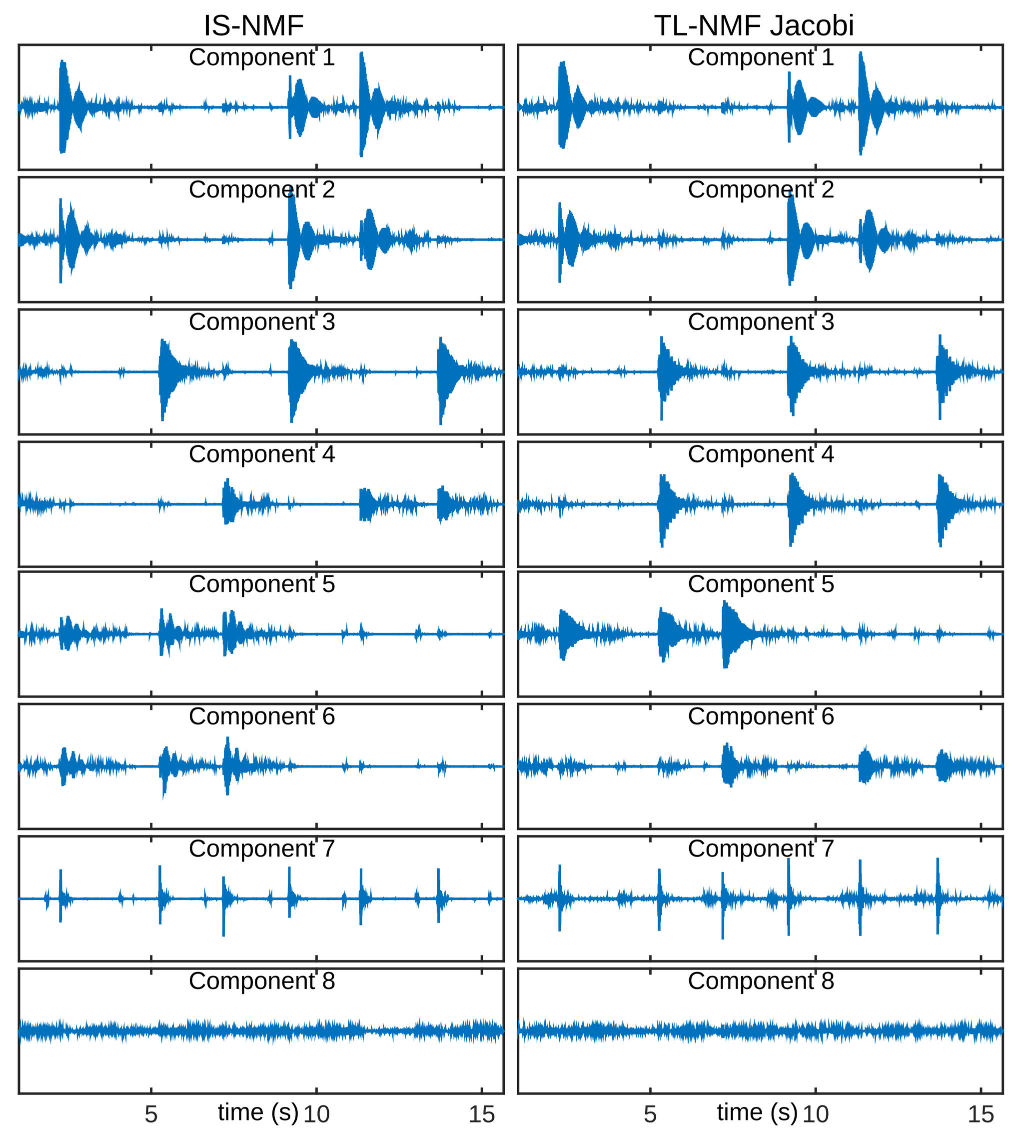
$f_s = 16$ kHz, $M = 640$ (40 ms, 50 % overlap, sine-bell), $N = 785$

TL-NMF Objective function



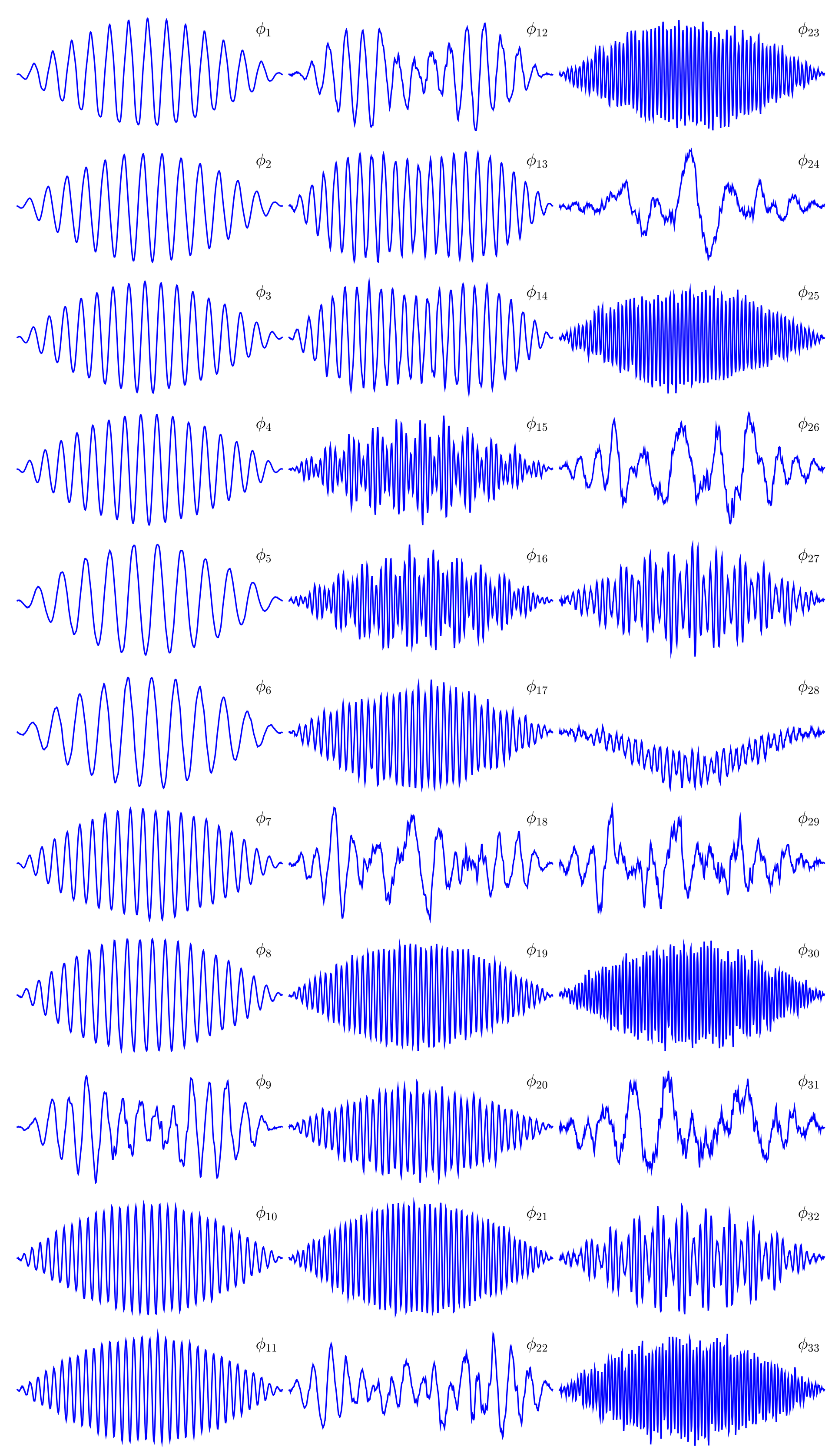
$K = 8$, $\tau = 10^{-10}$ (IS-NMF) & $\tau \approx 4\cdot 10^{-7}$ (TL-NMF)
$N_{\text{prop}} = 100$, $R = 6$ and $(a_1, a_2) = (0.3, 0.7)$, identical random init's
$\longrightarrow$ IS-NMF fastest (since Φ fixed), TL-NMF-Jacobi smallest objective

Decomposition



reconstruction using standard Wiener filtering approach (cf., [1])
$\rightarrow$ set of components for IS-NMF & TL-NMF comparable

Learnt atoms



$\rightarrow$ quadrature pairs
$\rightarrow$ variety of oscillatory patterns