REAL-TIME IMPLEMENTATIONS OF SPARSE LINEAR PREDICTION FOR SPEECH PROCESSING

AALBORG UNIVERSITY
DENMARK

The
Danish Council for
Strategic Research

T.L. Jensen[1], D. Giacobello[2], M.G. Christensen[3], S.H. Jensen[1] and M. Moonen[4]

[1]Dept. of Electronic Systems, Aalborg University, Denmark
[2]Beats Electronics LLC, Santa Monica, CA, USA
[3]Audio Analysis Lab, Dept. of Architecture, Design and & Technology, Aalborg Universitet, Denmark
[4]Dept. of Electrical Engineering (ESAT-SCD) and iMinds Future Health Dept., KU Leuven, Belgium

`{tlj,shj}@es.aau.dk`, `giacobello@ieee.com`, `mgc@create.aau.dk`, `marc.moonen@esat.kuleuven.be`

Introduction

- Linear prediction is one of the most successful tools for the analysis and coding of speech.
- 2-norm minimization is amenable of producing an optimization problem that is attractive both theoretically and computationally (Yule-Walker equations).
- The 1-norm criterion can give a sparse approximations of the prediction error which allow for a simple coding strategy and/or sparse approximation of high-order predictor. However, computationally can not be solved as efficient as the 2-norm approach.
- **Objective:** hand-tailor an algorithm for solving the sparse linear prediction problem suitable for real-time processing.

Sparse Linear Prediction

- Speech production model with samples $x[t]$

$$x[t] = \sum_{k=1}^{n} \alpha_k x[t-k] + r[t], \qquad (1)$$

- $\{\alpha_k\}$ are the prediction coefficients.
- $r[t]$ is the prediction error.
- Matrix model for a segment of $T = T_2 - T_1 + 1$ samples, $t = T_1, T_1 + 1, \ldots, T_2$

$$x = \begin{bmatrix} x[T_1] \\ \vdots \\ x[T_2] \end{bmatrix} = X\alpha + r, \qquad (2)$$

$$X = \begin{bmatrix} x[T_1-1] & \cdots & x[T_1-n] \\ \vdots & & \vdots \\ x[T_2-1] & \cdots & x[T_2-n] \end{bmatrix} \in \mathbb{R}^{m\times n}. \qquad (3)$$

- The general LPC problem is then written as

$$\underset{\alpha\in\mathbb{R}^n}{\text{minimize}} \quad \|x - X\alpha\|_p^p + \gamma\|\alpha\|_q^q . \qquad (4)$$

- The regularization term γ in (4) can be seen as being related to the prior knowledge of the distribution of the prediction coefficients vector α.
- We will use the 1-norm as a computationally tractable approximation of the cardinality measure.
- The problem then becomes [1]

$$\underset{\alpha\in\mathbb{R}^n}{\text{minimize}} \quad \|x - X\alpha\|_1 + \gamma\|\alpha\|_1 . \qquad (5)$$

Methods

- Interior-point methods because: 1) used by state-of-the-art general-purpose software 2) and real-time signal processing [2, 3].
- Key ingredient: fast and stable procedure for solving a linear system of equations in each iteration [4].
- Different "algorithm recipes": primal method [5] and dual method [4].

Primal method

- We then need to solve the system

$$(X^T D_1 X + \gamma^2 D_2)\Delta\alpha = X^T g_1 - \gamma g_2 . \qquad (6)$$

where $g_1 \in \mathbb{R}^m$, $g_2 \in \mathbb{R}^n$ and $D_1 \in \mathbb{R}^{m\times m}$, $D_2 \in \mathbb{R}^{n\times n}$ are positive definite diagonal matrices.
- Formed and solved in $\mathcal{O}(n^2 m + n^3)$ operations via Cholesky factorization.

Implementation

- The proposed algorithms are implemented in M (Matlab) and C++.
- The C++ and M implementation uses the LAPACK and BLAS library from the Intel Math Kernel Library (MKL).
- Mixed precision: first single precision operations then double precision.

Dual method

- We then need to solve the system

$$(D_1 + X D_2 X^T)\Delta\lambda = -r . \qquad (7)$$

- D_1, D_2 are positive definite matrices, $r \in \mathbb{R}^m$.
- Formed and solved in $\mathcal{O}(m^2 n + m^3)$ operations via Cholesky factorization.

Experimental Setup

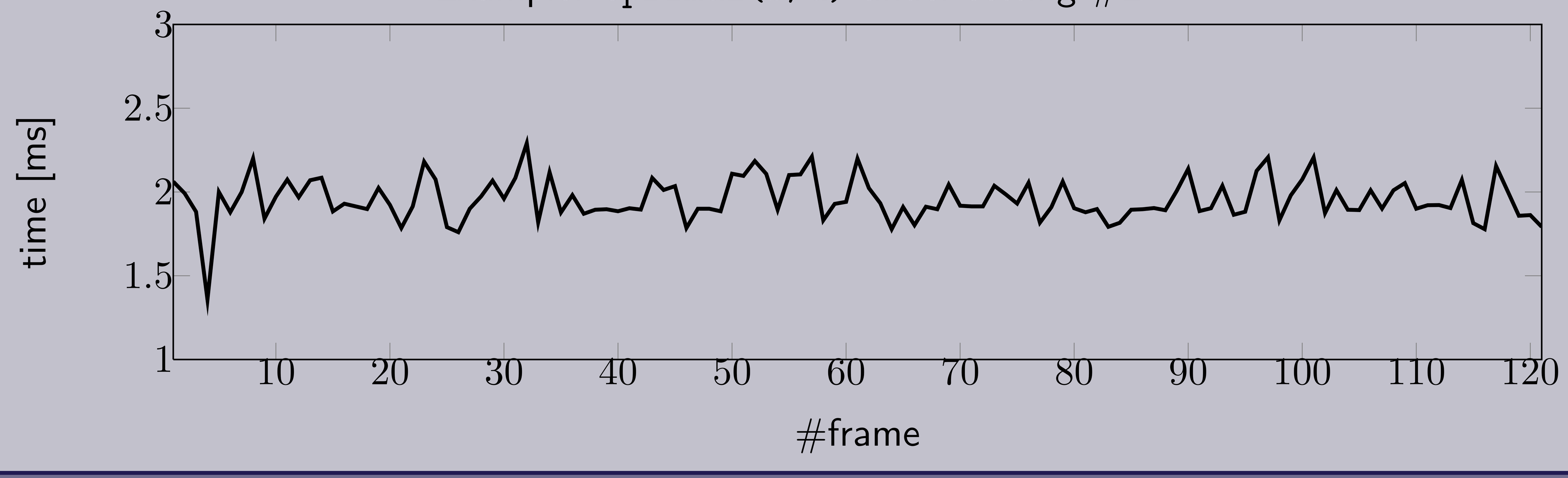
- Benchmarking is performed using a $\approx 2.5\,\mathrm{s}$ long vocalized speech signal sampled at $8\,\mathrm{kHz}$.
- Settings:
  - #1 frame length is $20\,\mathrm{ms}$ ($T = 160$ samples) with order $n = 100$.
  - #2 frame length is $20\,\mathrm{ms}$ ($T = 160$ samples) with order $n = 40$.
  - #3 frame length is $5\,\mathrm{ms}$ ($T = 40$ samples) with order $n = 10$.

Experimental Results

Average (min/max) timings in milliseconds, averaged over 100 runs for each frame.

Methods	#1	#2	#3
`CVX+SeDuMi`	**416.4** (279.2/520.1)	**344.7** (246.3/428.5)	**172.3** (148.1/200.0)
`Mosek`	**38.40** (28.05/44.00)	**17.12** (14.15/41.06)	**4.56** (3.60/4.82)
`Mprimal`	**25.24** (14.41/35.48)	**11.47** (6.32/14.54)	**4.27** (2.26/6.08)
`Mdual`	**23.49** (13.09/30.19)	**13.55** (7.78/19.67)	**3.15** (2.14/4.84)
`CVXGEN`	N/A	N/A	**0.56** (0.38/0.72)
`Cprimal`	**10.63** (6.70/13.58)	**2.30** (1.51/2.75)	**0.24** (0.14/0.41)
`Cdual`	**13.79** (7.36/17.70)	**5.52** (3.07/8.61)	**0.41** (0.28/0.64)
`Cprimal(s/d)`	**8.02** (5.29/10.64)	**1.96** (1.36/2.29)	**0.23** (0.15/0.30)
`Cdual(s/d)`	**10.22** (5.08/14.69)	**4.60** (2.23/6.96)	**0.39** (0.24/0.63)

Example: `Cprimal(s/d)` under setting #2



Analysis and Conclusion

- `Mprimal` vs `Cprimal`: speed-up of #1: 2.4, #2: 5.0 and #3 17.8. Increasing for smaller problems.
- `CVX+SeDuMi` is a highly used optimization software for prototyping and is only used here to highlight the potential speed-up that a hand-tailored algorithm can achieve.
- **Conclusion:** non-trivial real-time signal processing using hand-tailored convex optimization is possible.

Implementations: `sparsesampling.com/sparse_lp`.

References

[1] D. Giacobello, M. G. Christensen, M. N. Murthi, S. H. Jensen, and M. Moonen, "Sparse linear prediction and its applications to speech processing," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 5, pp. 1644–1657, Jul. 2012.

[2] J. Mattingley and S. Boyd, "CVXGEN: A code generator for embedded convex optimization," *Optim. Eng.*, vol. 13, no. 1, pp. 1–27, Mar. 2012.

[3] G. Alipoor and M. H. Savoji, "Wide-band speech coding based on bandwidth extension and sparse linear prediction," in *Int. Conf. Telecommun. Signal Process. (TSP)*, Jul. 2012, pp. 454–459.

[4] S. J. Wright, *Primal-Dual Interior-Point Methods*, SIAM, 1997.

[5] L. Vandenberghe, "The CVXOPT linear and quadratic cone program solvers," 2010, Documentation.