# Fast Algorithms for High-Order Sparse Linear Prediction with Applications to Speech Processing

February 26, 2015

T. L. Jensen
Joint work with D. Giacobello, T. van Waterschoot and M. G. Christensen

Dept. of Electronic Systems

Aalborg University

AALBORG UNIVERSITY
DENMARK

# Real-time optimization in signal processing



- Hard real-time (a solution is required at a certain time).
- General optimization in signal processing: as fast as possible.
- Current well-known methods: NLMS, RLS, LPC analysis/synthesis, Kalman filtering, Viterbi (decoding)....
- Real-time optimization for more complicated problems:
  - More complicated constraints
  - General convex problems or possible non-convex problems.
  - Non-smooth problems.

# Principles of linear prediction



- A stationary set of samples of speech $x[t]$, for $t = 1, \ldots, T$, are written as a linear combination of $N$ past samples

$$x[t] = \sum_{n=1}^{N} \alpha_n x[t-n] + r[t], \tag{1}$$

- $\{\alpha_n\}$ are the prediction coefficients and $r[t]$ is the prediction error.
- Matrix formulation (certain boundary conditions)

$$x = X\alpha + r \tag{2}$$

- Find the prediction coefficients via

$$\underset{\alpha}{\text{minimize}} \quad \|x - X\alpha\|_p^p \tag{3}$$

# Conventional linear prediction



- Select $p = 2$

$$\underset{\alpha}{\text{minimize}} \quad \|x - X\alpha\|_2^2 \tag{4}$$

- Solution satisfying the normal equation

$$X^T X\alpha = X^T x \tag{5}$$

- The autocorrelation matrix $R = X^T X$, is Toeplitz and with the special right-hand side, $X^T x$ the system can be solved using the Levinson–Durbin algorithm in $\mathcal{O}(N^2)$.

# Long-term prediction



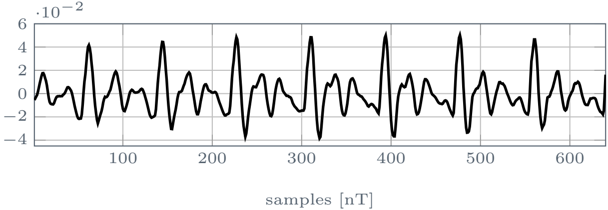
- Generally, linear prediction models only short-term redundancies of speech, thus is often used in combination with a single-tap or multi-tap long-term predictor[1]. The speech model for the long-term predictor is

$$d[t] = \sum_{k=0}^{K} \phi_k d[t - T_\mathrm{p} - k] + r[t], \qquad (6)$$

- $\{\phi_k\}$ are the (long term) prediction coefficients and $r[t]$ is the prediction error, and pitch period $T_\mathrm{p}$ [in samples].

[1] P. Kabal and R.P. Ramachandran. "Joint optimization of linear predictors in speech". In: *Acoustics, Speech and Signal Processing, IEEE Transactions on* 37.5 (1989), pp. 642–650. ISSN: 0096-3518.

- The combination of short term and long prediction filter can be seen as a sparse high order filter:



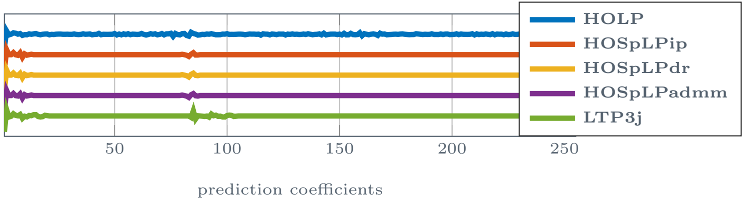




Figure : A 640 samples segment of the voiced speech (vowel /a/ uttered by a female speaker) and some predictors.

# High-order sparse linear prediction



- Imposing sparsity via the 1-norm convex relaxation:

$$\underset{\alpha}{\text{minimize}} \quad \|x - X\alpha\|_2^2 + \gamma\|\alpha\|_1 \,. \tag{7}$$

- However, when imposing sparsity on both the residual vector and high-order predictor, gains can been obtained both in terms of modeling and coding performance[2]

$$\underset{\alpha}{\text{minimize}} \quad \|x - X\alpha\|_1 + \gamma\|\alpha\|_1 \,. \tag{8}$$

- In general, check out[3].

[2] D. Giacobello et al. "Speech coding based on sparse linear prediction". In: *Proc. of the European Signal Processing Conference (EUSIPCO)*. 2009, pp. 2524–2528.

[3] Daniele Giacobello et al. "Sparse linear prediction and its applications to speech processing". In: *Audio, Speech, and Language Processing, IEEE Transactions on* 20.5 (2012), pp. 1644–1657.

# Solving the sparse linear prediction problem



- The objective:

$$f(\alpha) = \|x - X\alpha\|_1 + \gamma\|\alpha\|_1 \tag{9}$$

- is convex but not differentiable, neither is any of the terms $\rightarrow$ proximal gradient methods are not applicable[4][5][6].

---

[4] Yu. Nesterov. *Gradient methods for minimizing composite objective function.* Université catholique de Louvain, Center for Operations Research and Econometrics (CORE). No 2007076, CORE Discussion Papers. 2007.

[5] A. Beck and M. Teboulle. "A fast iterative shrinkage-thresholding algorithm for linear inverse problems". In: *SIAM Journal of Imaging Sciences* 2 (1 2009), pp. 183–202.

[6] S. J. Wright, R.D. Nowak, and M. A. T. Figueiredo. "Sparse reconstruction by separable approximation". In: *Signal Processing, IEEE Transaction on* 57 (2009), pp. 2479–2493.

# Solving the sparse linear prediction problem I



- The problem can be solved as a general linear programming problem using interior-point methods[7][8].
- Work-load concentrated on solving linear systems with coefficient matrix

$$C = X^T D_1 X + D_2, \quad D_1, D_2 \text{ diagonal, change at each iteration} \tag{10}$$

- With $X \in \mathbb{R}^{260\times 100}$: solved in $\simeq 10\,\mathrm{ms}$ on a standard laptop computer using C++ and MKL BLAS.

[7] Ghasem Alipoor and Mohammad Hasan Savoji. "Wide-band speech coding based on bandwidth extension and sparse linear prediction". In: *Telecommunications and Signal Processing (TSP), 2012 35th International Conference on*. IEEE. 2012, pp. 454–459.

[8] T.L. Jensen et al. "Real-time implementations of sparse linear prediction for speech processing". In: *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE. 2013, pp. 8184–8188.

# Solving the sparse linear prediction problem II



- Can it be done faster/more efficient?
- Is it possible to exploit the structure of $X$ and $R = X^T X$?
- Is the high accuracy of the IP methods necessary?

# Solving the sparse linear prediction problem III



- Investigate Douglas-Rachford and Alternating Directional Method of Multipliers (ADMM)
- Can be understood as dual methods of each other.
- Long history but have recently gained interested, also in signal processing[9][10].

---

[9] M.V. Afonso, J.M. Bioucas-Dias, and M.A.T. Figueiredo. "Fast Image Recovery Using Variable Splitting and Constrained Optimization". In: *Image Processing, IEEE Transactions on* 19.9 (2010), pp. 2345–2356. ISSN: 1057-7149.

[10] J. Yang and Y. Zhang. "Alternating Direction Algorithms for $\ell_1$-Problems in Compressive Sensing". In: *SIAM Journal of Scientific Computing* 33.1 (2011), pp. 250–278.

# Douglas-Rachford I



- Write the problem as

$$\underset{\alpha}{\text{minimize}} \quad f_1(\alpha) + f_2(X\alpha) \tag{11}$$

- $f_1(u) = \gamma\|u\|_1$ and $f_2(u) = \|x - u\|_1$.
- Let $h(u_1, u_2) = f_1(u_1) + f_2(u_2)$, then the problem can be written as

$$\begin{aligned} \underset{u_1,u_2}{\text{minimize}} \quad & h(u_1, u_2) \\ \text{subject to} \quad & u_2 = Xu_1 \, . \end{aligned} \tag{12}$$

# Douglas-Rachford I



- One form of the Douglas-Rachford algorithm is then

$$u^{(k+1)} = \mathbf{prox}_{th}(z^{(k)}) \tag{13}$$

$$y^{(k+1)} = \mathcal{P}_{\mathbb{Q}}(2u^{(k+1)} - z^{(k)}) \tag{14}$$

$$z^{(k+1)} = z^{(k)} + \eta(y^{(k+1)} - u^{(k+1)}) \tag{15}$$

- Relaxation parameter $\rho \in (0, 2)$, step-size parameter $t > 0$, set $\mathbb{Q} = \{ [u_1, u_2]^T \mid u_2 = Xu_1 \}$
- For smooth and strongly convex problems there are optimal choices for $t, \rho$. For non-smooth it is more heuristics[11].
- In this form, also known as Spingarns method[12].

[11] P. Patrinos, L. Stella, and A. Bemporad. *Douglas-Rachford splitting: complexity estimates and accelerated variants.* Proc. 53rd IEEE Conference on Decision and Control (CDC). 2014.

[12] J.E. Spingarn. "Applications of the method of partial inverses to convex programming: Decomposition". In: *Mathematical Programming* 32.2 (1985), pp. 199–223.

# Douglas-Rachford II



- Step one and three is simply soft-thresholding and level 1 BLAS.
- The projection in step 2 is

$$\mathcal{P}_{\mathbb{Q}}(v) = \begin{bmatrix} I \\ X \end{bmatrix} (I + X^T X)^{-1} (v_1 + X^T v_2)\,. \qquad (16)$$

- To compute (16) we need to solve a linear system of equations with (constant) coefficient matrix $I + X^T X$ and varying right-hand sides $(v_1 + X^T v_2)$.
- Recall $R = X^T X$ (symmetric and Toeplitz).

# ADMM I



- Reformulate as a basis pursuit problem

$$\begin{aligned} \underset{\tilde{z}}{\text{minimize}} \quad & \|\tilde{z}\|_1 \\ \text{subject to} \quad & \tilde{X}\tilde{z} = \tilde{x} \end{aligned} \tag{17}$$

- with

$$\tilde{X} = \begin{bmatrix} X & \gamma I \end{bmatrix} \tag{18}$$

$$\tilde{x} = \gamma x \,. \tag{19}$$

# ADMM II



- This problem formulation readily brings us to an ADMM algorithm defined by the iterations:

$$\tilde{z}^{(k+1)} = \mathcal{P}_{\mathbb{U}}(\tilde{y}^{(k)} - \tilde{u}^{(k)}) \tag{20}$$

$$\tilde{y}^{(k+1)} = \mathcal{S}_{1/\rho}(\tilde{z}^{(k+1)} + \tilde{u}^{(k)}) \tag{21}$$

$$\tilde{u}^{(k+1)} = \tilde{u}^{(k)} + \tilde{z}^{(k+1)} - \tilde{y}^{(k+1)}\,. \tag{22}$$

- where $\mathbb{U} = \{\tilde{z} \in \mathbb{R}^{m+n} \,|\, \tilde{X}\tilde{z} = \tilde{x}\}$

# ADMM II



- We find it instructive to write the algorithm in the form:

$$\alpha^{(k+1)} = \alpha_{\gamma,2} - \begin{bmatrix} -\gamma I \\ X \end{bmatrix}^{+} \left(y^{(k)} - u^{(k)}\right) \tag{23}$$

$$e^{(k+1)} = x - X\alpha^{(k+1)} \tag{24}$$

$$y^{(k+1)} = \mathcal{S}_{1/\rho}\left(\begin{bmatrix} \gamma\alpha^{(k+1)} \\ e^{(k+1)} \end{bmatrix} + u^{(k)}\right) \tag{25}$$

$$u^{(k+1)} = u^{(k)} + \begin{bmatrix} \gamma\alpha^{(k+1)} \\ e^{(k+1)} \end{bmatrix} - y^{(k+1)}. \tag{26}$$

- where $\alpha_{2,\gamma} = (X^T X + \gamma I)^{-1} X^T x$ and $(\cdot)^+$ denotes the Moore-Penrose pseudo-inverse.
- Note that with $\tilde{y}^{(0)} - \tilde{u}^{(0)} = 0$, we have $\alpha^{(1)} = \alpha_{\gamma,2}$, and the ADMM algorithm can then be interpreted as iterative "sparsification" of the $\ell_2$-regularized "classical" linear prediction solution.

# Solving positive-definite symmetric Toeplitz systems I



- Fast algorithms[13] $\rightarrow \mathcal{O}(N^2)$.
- Superfast algorithms[14] $\rightarrow \mathcal{O}(N \log^2 N)$ subsequent solves: $\mathcal{O}(N \log N)$.
- "Intermediate"[15] $\rightarrow \mathcal{O}(N^2)$ subsequent solves: $\mathcal{O}(N \log N)$.
- Break-even point in the number of operations at approximately $N = 256$ for $N$ as a radix 2 number. We will use $N = 250$, so go for the intermediate.

[13] N. Levinson. "The Weiner RMS Error Criterion in Filter Design and Prediction". In: *Journal of Mathematics and Physics* 25 (1947), pp. 261–278.

[14] R.R. Bitmead and B.D.O Anderson. "Asymptotically fast solution of Toeplitz and related systems of linear equations". In: *Linear Algebra and its Applications* 34 (1980), pp. 103–116; G.S. Ammar and W.B Gragg. "Superfast solution of real positive definite Toeplitz systems". In: *SIAM Journal on Matrix Analysis and Applications* 9.1 (1988), pp. 61–76.

[15] J. R. Jain. "An efficient algorithm for a large Toeplitz set of linear equations". In: *Acoustics, Speech and Signal Processing, IEEE Transaction on* 27.6 (1979).

# Solving positive-definite symmetric Toeplitz systems I



The inverse of a Toeplitz matrix can be described by the Gohberg-Semencul formula

$$\delta_N T^{-1} = T_1 T_1^T - T_0^T T_0 \tag{27}$$

where

$$T_0 = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \rho_0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \rho_{N-1} & \cdots & \rho_0 & 0 \end{bmatrix}, \quad T_1 = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ \rho_{N-1} & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \rho_0 & \cdots & \rho_{N-1} & 1 \end{bmatrix}. \tag{28}$$

The variables $\delta_N$ and $\rho_0, \ldots, \rho_{N-1}$ is a by-product of the Durbin algorithm (or Szegő recursions).

# Solving positive-definite symmetric Toeplitz systems II



- The solution to the system $Tx = b$ is then given by

$$x = T^{-1}b = \frac{1}{\delta_N}\left(T_1 T_1^T b - T_0^T T_0 b\right) . \qquad (29)$$

- Evaluation of matrix-vector products with $T_0, T_1$ is possible via FFTs/IFFTs.

# Timings

Results in ms on a standard desktop, single sentence, 131 frames of 20ms.

| Methods | Timings |
|---|---|
| **CVX+SeDuMi** | 1327.29/2467.80/3619.74 |
| **Mosek** | 145.54/224.71/307.60 |
| **Cprimal** | 55.24/92.70/180.46 |
| **Cprimal(s/d)** | 33.59/63.66/112.09 |
| **DR-L** | 0.65/6.62/10.11 |
| **DR-GS** | 0.61/2.28/3.26 |
| **ADMM-L** | 0.65/2.99/5.14 |
| **ADMM-GS** | 0.61/1.29/1.92 |

Table : Timing in milliseconds. Format: min/average/max. The settings are $T = 320$, $N = 250$ ($M = 570$).

# Convergence behaviour example

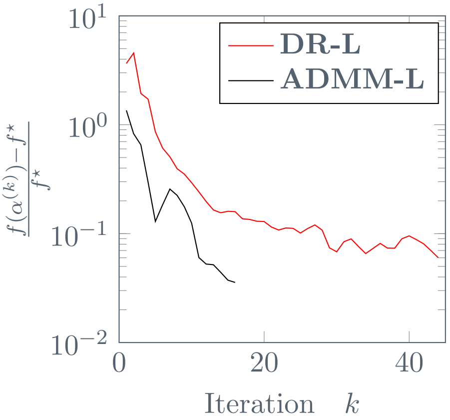



Figure : The endpoints of the graphs illustrates where the stopping criteria has become active and stopped the iterative algorithm.

# Convergence behaviour



- The splitting methods solved the problem to a low accuracy. Define the metrics

$$m_{\mathbf{DR}} = \frac{f_{\mathbf{DR}} - f^\star}{f^\star}, \qquad m_{\mathbf{ADMM}} = \frac{f_{\mathbf{ADMM}} - f^\star}{f^\star} \tag{30}$$

- On average $m_{\mathbf{DR}}$ and $m_{\mathbf{ADMM}}$ is 0.14 and 0.12, respectively.
- ADMM uses 13.5 iterations on average, while the DR based algorithms uses 35.3 iterations on average.
- Sub-optimal solutions can still provide exactly sparse solutions due to the soft-thresholding function.
- Do we only need a sparse and “small” solution with “small” residual?

# Prediction gain



| METHOD | $N$ = 320 | $N$ = 640 |
|---|---|---|
| **LTP1** | 17.3±0.8 | 14.2±1.0 |
| **LTP3** | 22.3±0.8 | 19.9±0.9 |
| **LTP3j** | 24.2±0.6 | 22.6±0.8 |
| **HOLP** | 32.4±0.6 | 31.3±0.7 |
| **HOSpLPip** | 28.6±1.1 | 27.8±1.4 |
| **HOSpLPdr** | 28.5±1.4 | 27.6±1.6 |
| **HOSpLPadmm** | 28.3±1.7 | 27.2±1.6 |

Table : Average prediction gains [dB] for segments of different length $N$, TIMIT database, only voiced speech frames. A 95% confidence interval is shown. The number of nonzero elements, $\mathbf{card}(\cdot)$, is shown for comparison. Fixed $\gamma = 0.12$.

# AR Interpolation I



- A segment of known and unknown samples

$$x = Kx_{\mathrm{k}} + Ux_{\mathrm{u}}, \tag{31}$$

- where $U$ and $K$ are $T \times T$ "rearrangements"
- If the AR coefficients are known, the residual is

$$r = A(Kx_{\mathrm{k}} + Ux_{\mathrm{u}}) \tag{32}$$

- with $A$ the so-called analysis matrix obtained from $\alpha$.
- The least-squares solution is

$$x_{\mathrm{u}} = -\left(A_{\mathrm{u}}^T A_{\mathrm{u}}\right)^{-1} A_{\mathrm{u}}^T A_{\mathrm{k}} x_{\mathrm{k}} \tag{33}$$

with $A_{\mathrm{u}} = AU$ and $A_{\mathrm{k}} = AK$.

# AR Interpolation II



| METHOD | $T_{\mathrm{GAP}}$ 4 | 6 | 8 | 10 | 20 |
|---|---|---|---|---|---|
| **sLP** | 3.92±0.09 | 3.15±0.15 | 2.96±0.16 | 2.30±0.18 | 1.71±0.22 |
| **LTP1** | 4.13±0.07 | 3.44±0.14 | 3.17±0.12 | 2.71±0.09 | 2.45±0.13 |
| **LTP3** | 4.17±0.07 | 3.53±0.09 | 3.22±0.13 | 2.92±0.12 | 2.63±0.09 |
| **LTPj** | 4.12±0.05 | 3.63±0.12 | 3.31±0.12 | 3.00±0.11 | 2.75±0.16 |
| **HOLP** | 4.27±0.04 | 3.55±0.06 | 3.34±0.08 | 2.91±0.09 | 2.61±0.11 |
| **HOSpLPip** | 4.34±0.03 | 3.75±0.05 | 3.56±0.08 | 3.27±0.09 | 3.12±0.15 |
| **HOSpLPdr** | 4.34±0.02 | 3.74±0.08 | 3.55±0.07 | 3.27±0.11 | 3.12±0.12 |
| **HOSpLPadmm** | 4.31±0.04 | 3.69±0.07 | 3.54±0.07 | 3.24±0.08 | 3.11±0.11 |

- Use $\alpha$ and $x_{\mathrm{k}}$ from previously known frame of size 40 ms.
- 1000 sentences from the TIMIT database (both voiced and unvoiced).
- $T_{\mathrm{GAP}}$ is the length of the unknown vector measured in ms.
- Average MOS for speech reconstruction with different gap size losses. A 95% confidence interval is shown.

# Conclusion



- Propose fast algorithms for sparse linear prediction.
- Usage of $\mathcal{O}(N \log N)$ algorithms for repeated solve of positive definite symmetric Toeplitz systems.
- The low accuracy solution provided by the fast algorithms allows to be implemented in real-time systems, particularly in wideband speech processing.
- Experimental evidence obtained through perceptually objective measures shows that the low accuracy solution performs as good as the high accuracy solution when applied in a autoregressive model-based speech reconstruction framework.