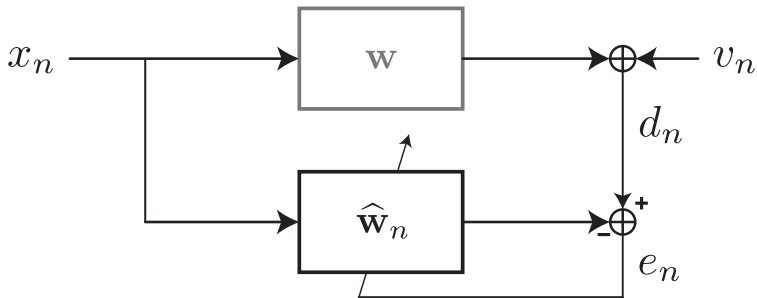


The Recursive Hessian Sketch for Adaptive Filtering

Robin Scheibler Martin Vetterli

School of Computer and Communication Sciences
École Polytechnique Fédérale de Lausanne, Switzerland

ICASSP
March 25, 2016



- Unknown filter \mathbf{w}
- Access to x_n, d_n
- Estimate of unknown filter $\hat{\mathbf{w}}_n$
- Estimation error $e_n = d_n - \hat{\mathbf{w}}_n^T \mathbf{x}$

1st order methods – Least mean squares (LMS)

- Stochastic gradient descent:

$$\hat{\mathbf{w}}_n = \arg \min_{\mathbf{w}} \mathbb{E}|e_n|^2, \quad \hat{\mathbf{w}}_n = \hat{\mathbf{w}}_{n-1} + \mu e_n \mathbf{x}_n$$

- Cheap, robust

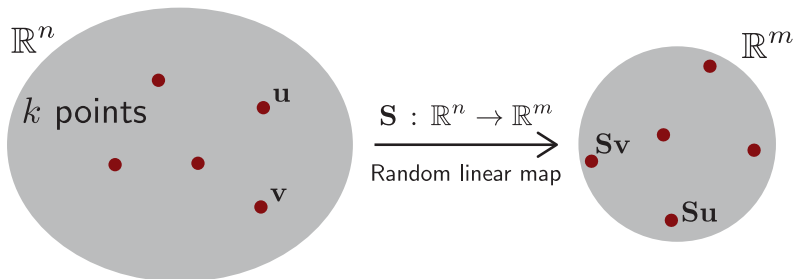
2nd order methods – Recursive Least Squares (RLS)

- Least squares problem:

$$\hat{\mathbf{w}}_n = \arg \min_{\mathbf{w}} \left\| \mathbf{\Lambda}_n^{1/2} (\mathbf{X}_n \mathbf{w} - \mathbf{d}_n) \right\|^2$$

- Complex, faster convergence, lower residual

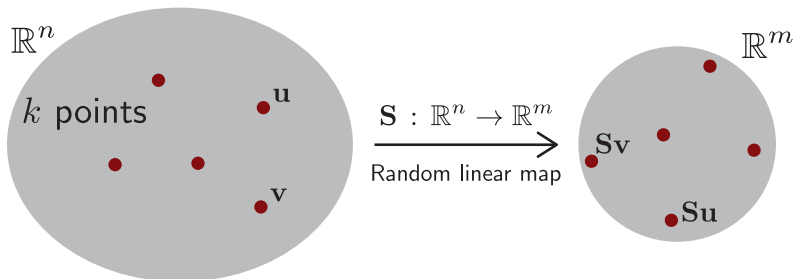
Theorem (Johnson-Lindenstrauss lemma)



Distances are preserved whp.

$$(1 - \epsilon)\|\mathbf{u} - \mathbf{v}\|^2 \leq \|\mathbf{S}\mathbf{u} - \mathbf{S}\mathbf{v}\|^2 \leq (1 + \epsilon)\|\mathbf{u} - \mathbf{v}\|^2$$

Theorem (Johnson-Lindenstrauss lemma)

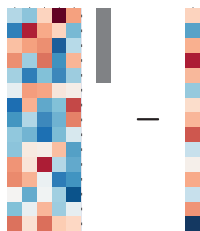


Distances are preserved whp.

$$(1 - \epsilon)\|\mathbf{u} - \mathbf{v}\|^2 \leq \|\mathbf{Su} - \mathbf{Sv}\|^2 \leq (1 + \epsilon)\|\mathbf{u} - \mathbf{v}\|^2$$

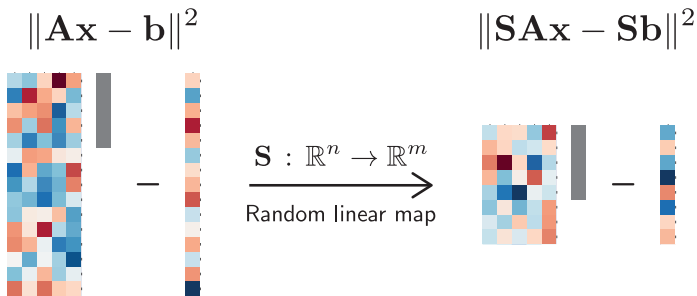
Application to solving least squares problem

$$\|\mathbf{Ax} - \mathbf{b}\|^2$$



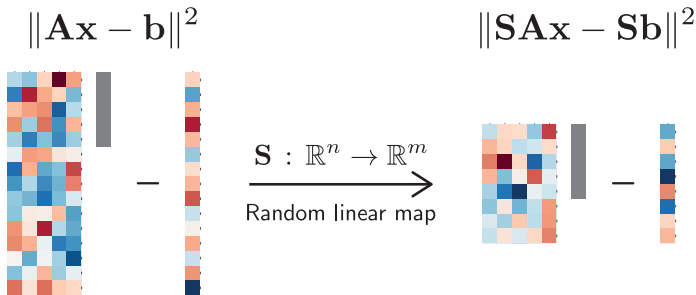
- Smaller system to solve!
- The J-L lemma implies $\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\|^2 \leq (1 + \epsilon)\|\mathbf{A}\mathbf{x}^{\text{LS}} - \mathbf{b}\|^2$
- But no good bound on solution error

Application to solving least squares problem



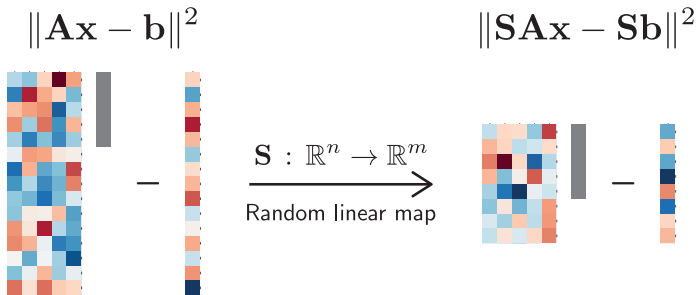
- Smaller system to solve!
- The J-L lemma implies $\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\|^2 \leq (1 + \epsilon)\|\mathbf{A}\mathbf{x}^{\text{LS}} - \mathbf{b}\|^2$
- But no good bound on solution error

Application to solving least squares problem



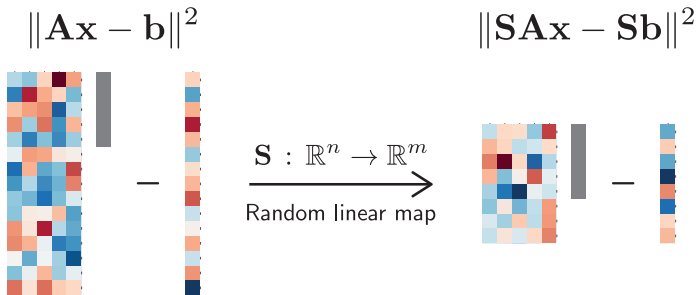
- Smaller system to solve!
- The J-L lemma implies $\|\mathbf{Ax} - \mathbf{b}\|^2 \leq (1 + \epsilon)\|\mathbf{Ax}^{\text{LS}} - \mathbf{b}\|^2$
- **But** no good bound on solution error

Application to solving least squares problem



- Smaller system to solve!
- The J-L lemma implies $\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\|^2 \leq (1 + \epsilon)\|\mathbf{Ax}^{\text{LS}} - \mathbf{b}\|^2$
- **But** no good bound on solution error

Application to solving least squares problem



- Smaller system to solve!
- The J-L lemma implies $\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\|^2 \leq (1 + \epsilon)\|\mathbf{Ax}^{\text{LS}} - \mathbf{b}\|^2$
- **But** no good bound on solution error

Apply sketching to the RLS algorithm.

$$\hat{\mathbf{w}}_n = \arg \min_{\mathbf{w}} \left\| \mathbf{\Lambda}_n^{1/2} (\mathbf{X}_n \mathbf{w} - \mathbf{d}_n) \right\|^2$$

Wish list

- As good as RLS
- With less computations
- Good convergence

Apply sketching to the RLS algorithm.

$$\hat{\mathbf{w}}_n = \arg \min_{\mathbf{w}} \left\| \mathbf{\Lambda}_n^{1/2} (\mathbf{X}_n \mathbf{w} - \mathbf{d}_n) \right\|^2$$

Wish list

- As good as RLS
- With less computations
- Good convergence

Apply sketching to the RLS algorithm.

$$\hat{\mathbf{w}}_n = \arg \min_{\mathbf{w}} \left\| \mathbf{\Lambda}_n^{1/2} (\mathbf{X}_n \mathbf{w} - \mathbf{d}_n) \right\|^2$$

Wish list

- As good as RLS
- With less computations
- Good convergence

Apply sketching to the RLS algorithm.

$$\hat{\mathbf{w}}_n = \arg \min_{\mathbf{w}} \left\| \mathbf{\Lambda}_n^{1/2} (\mathbf{X}_n \mathbf{w} - \mathbf{d}_n) \right\|^2$$

Wish list

- As good as RLS
- With less computations
- Good convergence

1. The iterative Hessian sketch
2. The recursive least squares
3. The recursive Hessian sketch

The iterative Hessian sketch

The Hessian sketch for least-squares

M. Pilanci, M. J. Wainwright, *Iterative Hessian sketch: Fast and accurate solution approximation for constrained least-squares*, 2014.

Goal

$$\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{Ax} - \mathbf{b}\|^2, \quad \begin{array}{l} \mathbf{A} : \text{ data matrix} \\ \mathbf{b} : \text{ response vector} \end{array}$$

The Hessian sketch

$$\tilde{\mathbf{x}} = \arg \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{SAx}\|^2 - (\mathbf{A}^\top \mathbf{b})^\top \mathbf{x}$$

Sketch **only** data matrix, then

$$\frac{\|\mathbf{x}^{\text{LS}} - \tilde{\mathbf{x}}\|_{\mathbf{A}}}{\|\mathbf{x}^{\text{LS}}\|_{\mathbf{A}}} \leq \delta$$

The Hessian sketch for least-squares

M. Pilanci, M. J. Wainwright, *Iterative Hessian sketch: Fast and accurate solution approximation for constrained least-squares*, 2014.

Goal

$$\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{Ax} - \mathbf{b}\|^2, \quad \begin{array}{l} \mathbf{A} : \text{ data matrix} \\ \mathbf{b} : \text{ response vector} \end{array}$$

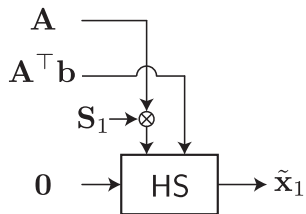
The Hessian sketch

$$\tilde{\mathbf{x}} = \arg \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{SAx}\|^2 - (\mathbf{A}^\top \mathbf{b})^\top \mathbf{x}$$

Sketch **only** data matrix, then

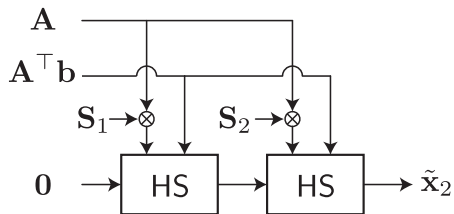
$$\frac{\|\mathbf{x}^{\text{LS}} - \tilde{\mathbf{x}}\|_{\mathbf{A}}}{\|\mathbf{x}^{\text{LS}}\|_{\mathbf{A}}} \leq \delta$$

The iterative Hessian sketch (IHS)



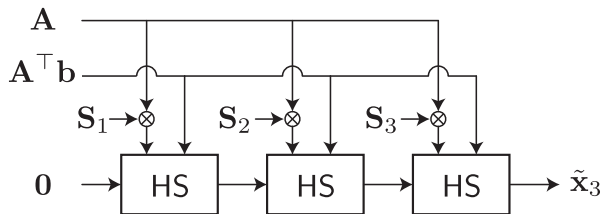
Relative error: δ

The iterative Hessian sketch (IHS)



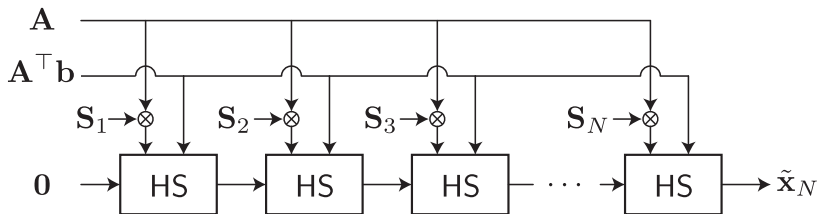
Relative error: δ^2

The iterative Hessian sketch (IHS)



Relative error: δ^3

The iterative Hessian sketch (IHS)



Relative error: ϵ in $N = \log(1/\epsilon)$ iterations

Iterative Hessian sketch : summary

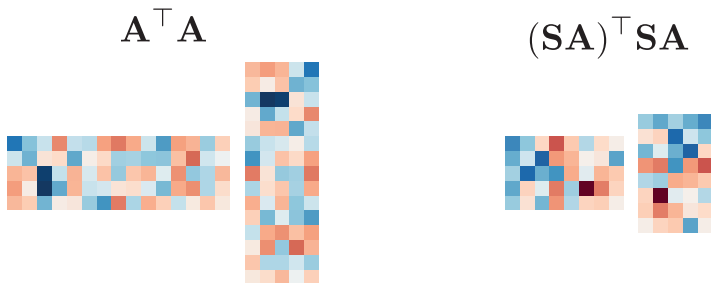
- Sketch **data matrix**, not the response vector
- ϵ -approx of LS in $\log(1/\epsilon)$ iterations
- Save computational cost of $\mathbf{A}^T \mathbf{A}$

Iterative Hessian sketch : summary

- Sketch **data matrix**, not the response vector
- ϵ -approx of LS in $\log(1/\epsilon)$ iterations
- Save computational cost of $\mathbf{A}^T \mathbf{A}$

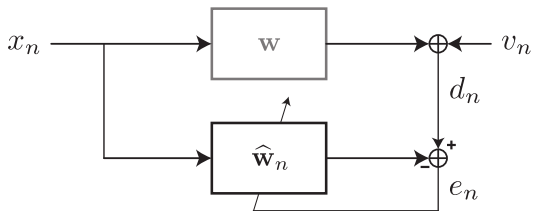
Iterative Hessian sketch : summary

- Sketch **data matrix**, not the response vector
- ϵ -approx of LS in $\log(1/\epsilon)$ iterations
- Save computational cost of $\mathbf{A}^T \mathbf{A}$

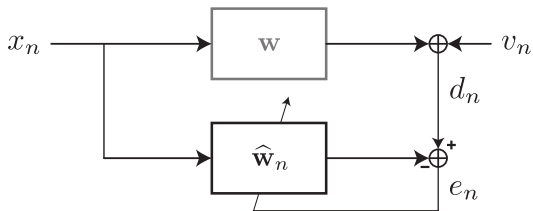


The recursive least squares

Exponentially weighted least squares

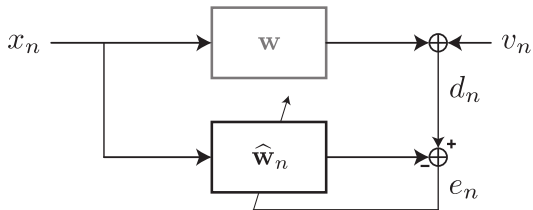


Exponentially weighted least squares



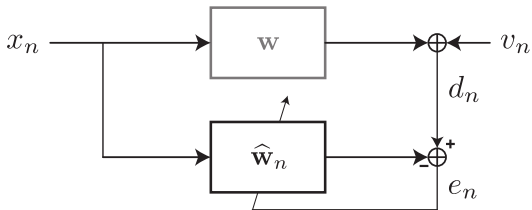
\hat{w}_n

Exponentially weighted least squares



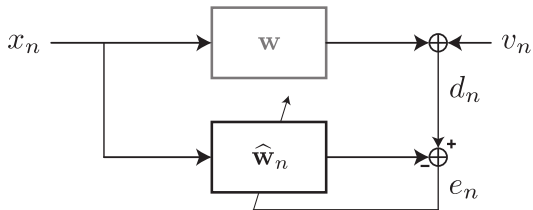
X_n \hat{w}_n

Exponentially weighted least squares



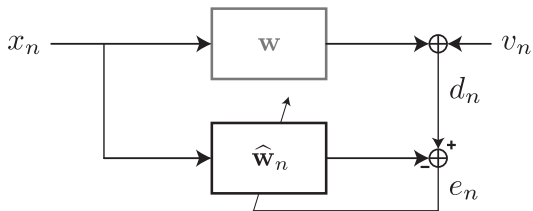
$$\left(\begin{array}{c|c} \begin{matrix} \color{red}\blacksquare & \color{blue}\blacksquare & \color{lightblue}\blacksquare & \color{orange}\blacksquare & \color{lightorange}\blacksquare & \color{lightgrey}\blacksquare & \color{grey}\blacksquare \\ \color{lightblue}\blacksquare & \color{orange}\blacksquare & \color{lightorange}\blacksquare & \color{lightgrey}\blacksquare & \color{grey}\blacksquare & \color{lightblue}\blacksquare & \color{orange}\blacksquare \\ \color{orange}\blacksquare & \color{lightorange}\blacksquare & \color{lightgrey}\blacksquare & \color{grey}\blacksquare & \color{lightblue}\blacksquare & \color{orange}\blacksquare & \color{lightorange}\blacksquare \\ \color{lightorange}\blacksquare & \color{lightgrey}\blacksquare & \color{grey}\blacksquare & \color{lightblue}\blacksquare & \color{orange}\blacksquare & \color{lightorange}\blacksquare & \color{lightgrey}\blacksquare \\ \color{grey}\blacksquare & \color{lightblue}\blacksquare & \color{orange}\blacksquare & \color{lightorange}\blacksquare & \color{lightgrey}\blacksquare & \color{grey}\blacksquare & \color{lightblue}\blacksquare \\ \color{lightblue}\blacksquare & \color{orange}\blacksquare & \color{lightorange}\blacksquare & \color{lightgrey}\blacksquare & \color{grey}\blacksquare & \color{lightblue}\blacksquare & \color{orange}\blacksquare \\ \color{orange}\blacksquare & \color{lightorange}\blacksquare & \color{lightgrey}\blacksquare & \color{grey}\blacksquare & \color{lightblue}\blacksquare & \color{orange}\blacksquare & \color{lightorange}\blacksquare \\ \color{lightorange}\blacksquare & \color{lightgrey}\blacksquare & \color{grey}\blacksquare & \color{lightblue}\blacksquare & \color{orange}\blacksquare & \color{lightorange}\blacksquare & \color{lightgrey}\blacksquare \\ \color{grey}\blacksquare & \color{lightblue}\blacksquare & \color{orange}\blacksquare & \color{lightorange}\blacksquare & \color{lightgrey}\blacksquare & \color{grey}\blacksquare & \color{lightblue}\blacksquare \\ \color{lightblue}\blacksquare & \color{orange}\blacksquare & \color{lightorange}\blacksquare & \color{lightgrey}\blacksquare & \color{grey}\blacksquare & \color{lightblue}\blacksquare & \color{orange}\blacksquare \end{matrix} & \color{grey}\blacksquare \\ \hline \begin{matrix} \color{lightblue}\blacksquare \\ \color{lightgrey}\blacksquare \\ \color{darkred}\blacksquare \\ \color{blue}\blacksquare \\ \color{orange}\blacksquare \\ \color{darkred}\blacksquare \\ \color{red}\blacksquare \\ \color{orange}\blacksquare \\ \color{blue}\blacksquare \\ \color{red}\blacksquare \end{matrix} \end{array} \right) \\ \mathbf{X}_n \quad \hat{\mathbf{w}}_n \quad \mathbf{y}_n$$

Exponentially weighted least squares



$$\begin{bmatrix} 1 \\ \lambda \\ \lambda^2 \\ \lambda^3 \\ \dots \\ \Lambda_n^{1/2} \end{bmatrix} \left(\begin{array}{c|c} \begin{matrix} \text{[Colorful matrix]} \\ \mathbf{X}_n \end{matrix} & \begin{matrix} \text{[Grey bar]} \\ \hat{\mathbf{w}}_n \end{matrix} \\ \hline & \begin{matrix} \text{[Colorful vector]} \\ \mathbf{y}_n \end{matrix} \end{array} \right)$$

Exponentially weighted least squares



$$\left\| \left[\begin{array}{c} 1 \\ \lambda \\ \lambda^2 \\ \lambda^3 \\ \vdots \end{array} \right] \left(\begin{array}{c} \mathbf{X}_n \hat{\mathbf{w}}_n - \mathbf{y}_n \end{array} \right) \right\|^2$$

The diagram shows the mathematical representation of the exponentially weighted least squares problem. The input x_n is represented by a vector of powers of λ , $\Lambda_n^{1/2}$. The output of the weight block w is represented by the product of the input vector and the weight vector $\hat{\mathbf{w}}_n$. The error signal e_n is represented by the difference between the output of the weight block and the desired signal \mathbf{y}_n . The error signal is squared to represent the cost function.

RLS filter update

$$\hat{\mathbf{w}}_n = \underbrace{\left(\mathbf{X}_n^\top \boldsymbol{\Lambda}_n \mathbf{X}_n \right)^{-1}}_{\mathbf{R}_n} \underbrace{\mathbf{X}_n^\top \boldsymbol{\Lambda}_n \mathbf{d}_n}_{\mathbf{y}_n}$$

Data update

$$\mathbf{X}_{n+1} = \begin{bmatrix} \mathbf{x}^\top \\ \mathbf{X}_n \end{bmatrix} \quad \mathbf{d}_{n+1} = \begin{bmatrix} d \\ \mathbf{d}_n \end{bmatrix}$$

RLS filter update

$$\hat{\mathbf{w}}_{n+1} = \underbrace{\left(\lambda \mathbf{R}_n + \mathbf{x} \mathbf{x}^\top \right)^{-1}}_{\text{rank-1 update!}} \underbrace{\left(\lambda \mathbf{y}_n + \mathbf{x} d \right)}_{\mathbf{y}_{n+1}}$$

RLS filter update

$$\hat{\mathbf{w}}_n = \underbrace{\left(\mathbf{X}_n^\top \Lambda_n \mathbf{X}_n \right)^{-1}}_{\mathbf{R}_n} \underbrace{\mathbf{X}_n^\top \Lambda_n \mathbf{d}_n}_{\mathbf{y}_n}$$

Data update

$$\mathbf{X}_{n+1} = \begin{bmatrix} \mathbf{x}^\top \\ \mathbf{X}_n \end{bmatrix} \quad \mathbf{d}_{n+1} = \begin{bmatrix} d \\ \mathbf{d}_n \end{bmatrix}$$

RLS filter update

$$\hat{\mathbf{w}}_{n+1} = \underbrace{\left(\lambda \mathbf{R}_n + \mathbf{x} \mathbf{x}^\top \right)^{-1}}_{\text{rank-1 update!}} \underbrace{\left(\lambda \mathbf{y}_n + \mathbf{x} d \right)}_{\mathbf{y}_{n+1}}$$

RLS filter update

$$\hat{\mathbf{w}}_n = \underbrace{\left(\mathbf{X}_n^\top \Lambda_n \mathbf{X}_n \right)^{-1}}_{\mathbf{R}_n} \underbrace{\mathbf{X}_n^\top \Lambda_n \mathbf{d}_n}_{\mathbf{y}_n}$$

Data update

$$\mathbf{X}_{n+1} = \begin{bmatrix} \mathbf{x}^\top \\ \mathbf{X}_n \end{bmatrix} \quad \mathbf{d}_{n+1} = \begin{bmatrix} d \\ \mathbf{d}_n \end{bmatrix}$$

RLS filter update

$$\hat{\mathbf{w}}_{n+1} = \underbrace{\left(\lambda \mathbf{R}_n + \mathbf{x}\mathbf{x}^\top \right)^{-1}}_{\text{rank-1 update!}} \underbrace{\left(\lambda \mathbf{y}_n + \mathbf{x}d \right)}_{\mathbf{y}_{n+1}}$$

- Solve LS at each step
- Update solution with matrix inversion lemma
- Cost quadratic in filter length

- Solve LS at each step
- Update solution with matrix inversion lemma
- Cost quadratic in filter length

- Solve LS at each step
- Update solution with matrix inversion lemma
- Cost quadratic in filter length

The recursive Hessian sketch

- Recall the Hessian sketch ($\mathbf{A} = \Lambda_n^{1/2} \mathbf{X}_n$, $\mathbf{b} = \Lambda_n^{1/2} \mathbf{d}_n$)

$$\tilde{\mathbf{w}}_n = \arg \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{S}_n \mathbf{A} \mathbf{x}\|^2 - (\mathbf{A}^\top \mathbf{b})^\top \mathbf{x}$$

- Random row sampling : \mathbf{S}_n , fixed aspect ratio $q = \frac{m}{n}$

$$\mathbf{S}_n = \begin{bmatrix} b_n & 0 \\ 0 & \mathbf{S}_{n-1} \end{bmatrix}, \quad b_n = \begin{cases} 1 & \text{w.p. } q \\ 0 & \text{w.p. } 1 - q \end{cases}$$

Caveat: IHS proof does not cover this sketch (yet)

- Recall the Hessian sketch ($\mathbf{A} = \mathbf{\Lambda}_n^{1/2} \mathbf{X}_n$, $\mathbf{b} = \mathbf{\Lambda}_n^{1/2} \mathbf{d}_n$)

$$\tilde{\mathbf{w}}_n = \arg \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{S}_n \mathbf{A} \mathbf{x}\|^2 - (\mathbf{A}^\top \mathbf{b})^\top \mathbf{x}$$

- Random row sampling : \mathbf{S}_n , fixed aspect ratio $q = \frac{m}{n}$

$$\mathbf{S}_n = \begin{bmatrix} b_n & 0 \\ 0 & \mathbf{S}_{n-1} \end{bmatrix}, \quad b_n = \begin{cases} 1 & \text{w.p. } q \\ 0 & \text{w.p. } 1 - q \end{cases}$$

Caveat: IHS proof does not cover this sketch (yet)

- Recall the Hessian sketch ($\mathbf{A} = \mathbf{\Lambda}_n^{1/2} \mathbf{X}_n$, $\mathbf{b} = \mathbf{\Lambda}_n^{1/2} \mathbf{d}_n$)

$$\tilde{\mathbf{w}}_n = \arg \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{S}_n \mathbf{A} \mathbf{x}\|^2 - (\mathbf{A}^\top \mathbf{b})^\top \mathbf{x}$$

- Random row sampling : \mathbf{S}_n , fixed aspect ratio $q = \frac{m}{n}$

$$\mathbf{S}_n = \begin{bmatrix} b_n & 0 \\ 0 & \mathbf{S}_{n-1} \end{bmatrix}, \quad b_n = \begin{cases} 1 & \text{w.p. } q \\ 0 & \text{w.p. } 1 - q \end{cases}$$

Caveat: IHS proof does not cover this sketch (yet)

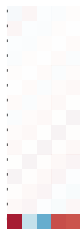
- Recall the Hessian sketch ($\mathbf{A} = \mathbf{\Lambda}_n^{1/2} \mathbf{X}_n$, $\mathbf{b} = \mathbf{\Lambda}_n^{1/2} \mathbf{d}_n$)

$$\tilde{\mathbf{w}}_n = \arg \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{S}_n \mathbf{A} \mathbf{x}\|^2 - (\mathbf{A}^\top \mathbf{b})^\top \mathbf{x}$$

- Random row sampling : \mathbf{S}_n , fixed aspect ratio $q = \frac{m}{n}$

$$\mathbf{S}_n = \begin{bmatrix} b_n & 0 \\ 0 & \mathbf{S}_{n-1} \end{bmatrix}, \quad b_n = \begin{cases} 1 & \text{w.p. } q \\ 0 & \text{w.p. } 1 - q \end{cases}$$

$$b_1 = 1$$



Caveat: IHS proof does not cover this sketch (yet)

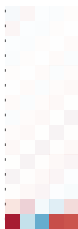
- Recall the Hessian sketch ($\mathbf{A} = \mathbf{\Lambda}_n^{1/2} \mathbf{X}_n$, $\mathbf{b} = \mathbf{\Lambda}_n^{1/2} \mathbf{d}_n$)

$$\tilde{\mathbf{w}}_n = \arg \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{S}_n \mathbf{A} \mathbf{x}\|^2 - (\mathbf{A}^\top \mathbf{b})^\top \mathbf{x}$$

- Random row sampling : \mathbf{S}_n , fixed aspect ratio $q = \frac{m}{n}$

$$\mathbf{S}_n = \begin{bmatrix} b_n & 0 \\ 0 & \mathbf{S}_{n-1} \end{bmatrix}, \quad b_n = \begin{cases} 1 & \text{w.p. } q \\ 0 & \text{w.p. } 1 - q \end{cases}$$

$$b_2 = 0$$



Caveat: IHS proof does not cover this sketch (yet)

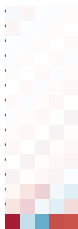
- Recall the Hessian sketch ($\mathbf{A} = \mathbf{\Lambda}_n^{1/2} \mathbf{X}_n$, $\mathbf{b} = \mathbf{\Lambda}_n^{1/2} \mathbf{d}_n$)

$$\tilde{\mathbf{w}}_n = \arg \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{S}_n \mathbf{A} \mathbf{x}\|^2 - (\mathbf{A}^\top \mathbf{b})^\top \mathbf{x}$$

- Random row sampling : \mathbf{S}_n , fixed aspect ratio $q = \frac{m}{n}$

$$\mathbf{S}_n = \begin{bmatrix} b_n & 0 \\ 0 & \mathbf{S}_{n-1} \end{bmatrix}, \quad b_n = \begin{cases} 1 & \text{w.p. } q \\ 0 & \text{w.p. } 1 - q \end{cases}$$

$$b_3 = 0$$



Caveat: IHS proof does not cover this sketch (yet)

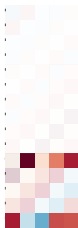
- Recall the Hessian sketch ($\mathbf{A} = \mathbf{\Lambda}_n^{1/2} \mathbf{X}_n$, $\mathbf{b} = \mathbf{\Lambda}_n^{1/2} \mathbf{d}_n$)

$$\tilde{\mathbf{w}}_n = \arg \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{S}_n \mathbf{A} \mathbf{x}\|^2 - (\mathbf{A}^\top \mathbf{b})^\top \mathbf{x}$$

- Random row sampling : \mathbf{S}_n , fixed aspect ratio $q = \frac{m}{n}$

$$\mathbf{S}_n = \begin{bmatrix} b_n & 0 \\ 0 & \mathbf{S}_{n-1} \end{bmatrix}, \quad b_n = \begin{cases} 1 & \text{w.p. } q \\ 0 & \text{w.p. } 1 - q \end{cases}$$

$$b_4 = 1$$



Caveat: IHS proof does not cover this sketch (yet)

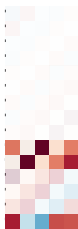
- Recall the Hessian sketch ($\mathbf{A} = \mathbf{\Lambda}_n^{1/2} \mathbf{X}_n$, $\mathbf{b} = \mathbf{\Lambda}_n^{1/2} \mathbf{d}_n$)

$$\tilde{\mathbf{w}}_n = \arg \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{S}_n \mathbf{A} \mathbf{x}\|^2 - (\mathbf{A}^\top \mathbf{b})^\top \mathbf{x}$$

- Random row sampling : \mathbf{S}_n , fixed aspect ratio $q = \frac{m}{n}$

$$\mathbf{S}_n = \begin{bmatrix} b_n & 0 \\ 0 & \mathbf{S}_{n-1} \end{bmatrix}, \quad b_n = \begin{cases} 1 & \text{w.p. } q \\ 0 & \text{w.p. } 1 - q \end{cases}$$

$$b_5 = 1$$



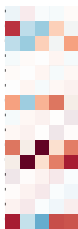
Caveat: IHS proof does not cover this sketch (yet)

- Recall the Hessian sketch ($\mathbf{A} = \mathbf{\Lambda}_n^{1/2} \mathbf{X}_n$, $\mathbf{b} = \mathbf{\Lambda}_n^{1/2} \mathbf{d}_n$)

$$\tilde{\mathbf{w}}_n = \arg \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{S}_n \mathbf{A} \mathbf{x}\|^2 - (\mathbf{A}^\top \mathbf{b})^\top \mathbf{x}$$

- Random row sampling : \mathbf{S}_n , fixed aspect ratio $q = \frac{m}{n}$

$$\mathbf{S}_n = \begin{bmatrix} b_n & 0 \\ 0 & \mathbf{S}_{n-1} \end{bmatrix}, \quad b_n = \begin{cases} 1 & \text{w.p. } q \\ 0 & \text{w.p. } 1 - q \end{cases}$$



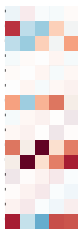
Caveat: IHS proof does not cover this sketch (yet)

- Recall the Hessian sketch ($\mathbf{A} = \mathbf{\Lambda}_n^{1/2} \mathbf{X}_n$, $\mathbf{b} = \mathbf{\Lambda}_n^{1/2} \mathbf{d}_n$)

$$\tilde{\mathbf{w}}_n = \arg \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{S}_n \mathbf{A} \mathbf{x}\|^2 - (\mathbf{A}^\top \mathbf{b})^\top \mathbf{x}$$

- Random row sampling : \mathbf{S}_n , fixed aspect ratio $q = \frac{m}{n}$

$$\mathbf{S}_n = \begin{bmatrix} b_n & 0 \\ 0 & \mathbf{S}_{n-1} \end{bmatrix}, \quad b_n = \begin{cases} 1 & \text{w.p. } q \\ 0 & \text{w.p. } 1 - q \end{cases}$$



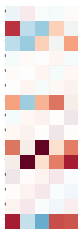
Caveat: IHS proof does not cover this sketch (yet)

- Recall the Hessian sketch ($\mathbf{A} = \mathbf{\Lambda}_n^{1/2} \mathbf{X}_n$, $\mathbf{b} = \mathbf{\Lambda}_n^{1/2} \mathbf{d}_n$)

$$\tilde{\mathbf{w}}_n = \arg \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{S}_n \mathbf{A} \mathbf{x}\|^2 - (\mathbf{A}^\top \mathbf{b})^\top \mathbf{x}$$

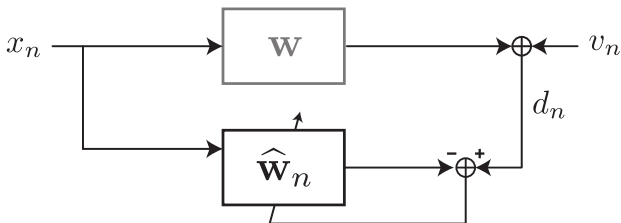
- Random row sampling : \mathbf{S}_n , fixed aspect ratio $q = \frac{m}{n}$

$$\mathbf{S}_n = \begin{bmatrix} b_n & 0 \\ 0 & \mathbf{S}_{n-1} \end{bmatrix}, \quad b_n = \begin{cases} 1 & \text{w.p. } q \\ 0 & \text{w.p. } 1 - q \end{cases}$$

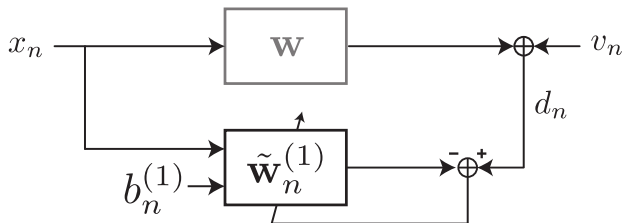


Caveat: IHS proof does not cover this sketch (yet)

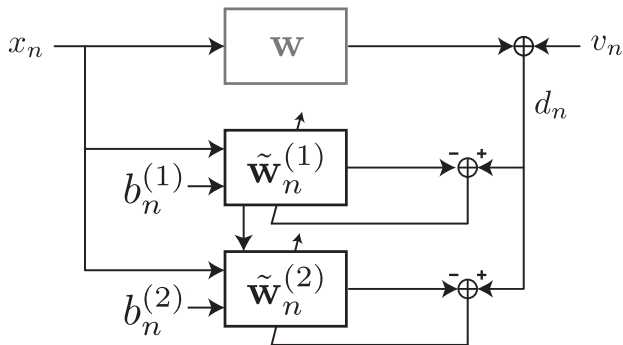
The recursive Hessian sketch (RHS)



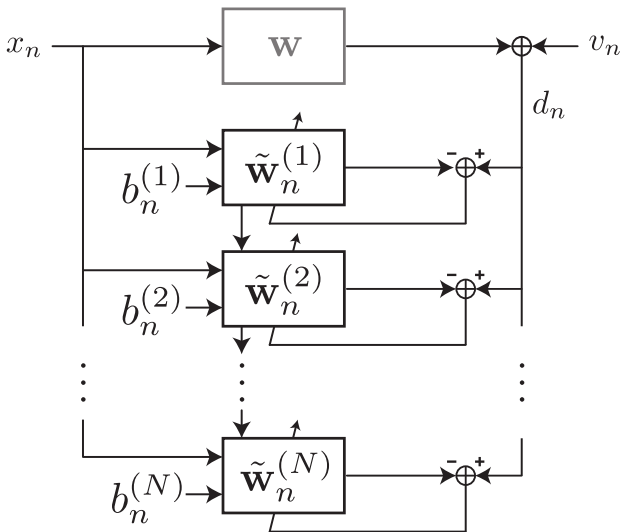
The recursive Hessian sketch (RHS)



The recursive Hessian sketch (RHS)



The recursive Hessian sketch (RHS)



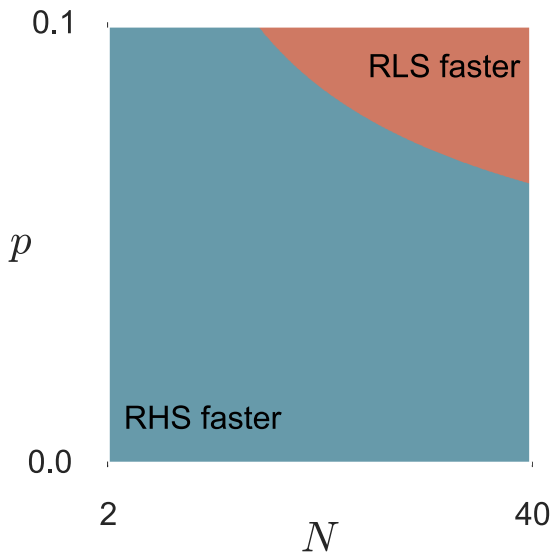
- Apply Hessian sketch to RLS
- Update inverse matrix w.p. q
- Cascade N sketched RLS

- Apply Hessian sketch to RLS
- Update inverse matrix w.p. q
- Cascade N sketched RLS

- Apply Hessian sketch to RLS
- Update inverse matrix w.p. q
- Cascade N sketched RLS

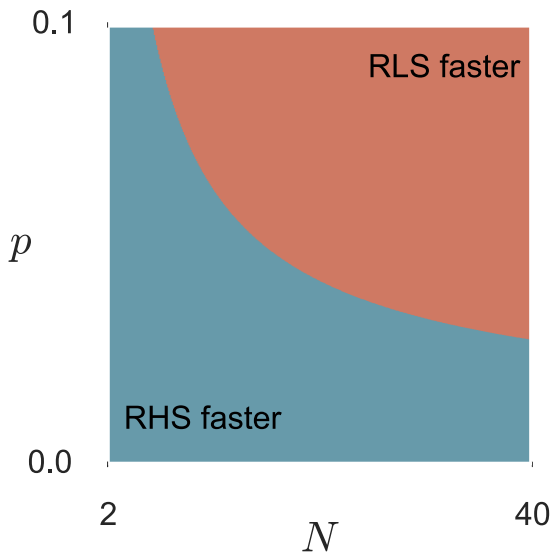
Complexity RLS vs RHS

Filter length : 10



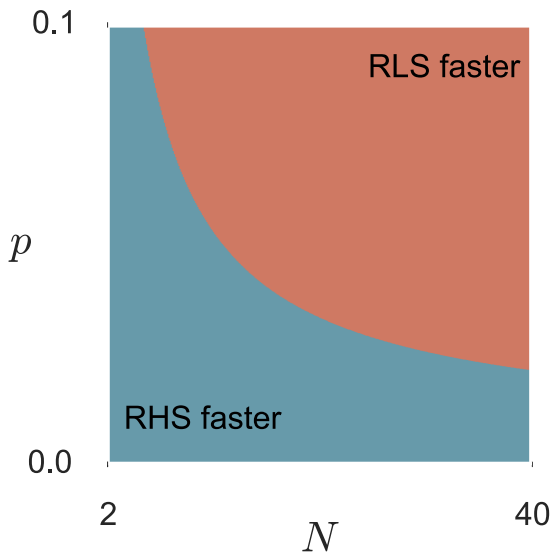
Complexity RLS vs RHS

Filter length : 50



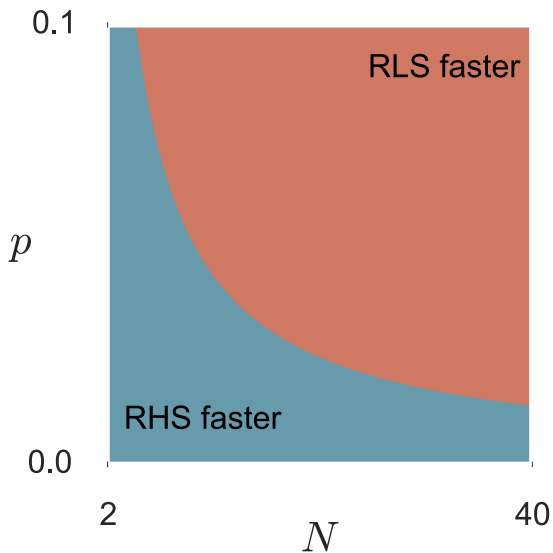
Complexity RLS vs RHS

Filter length : 100



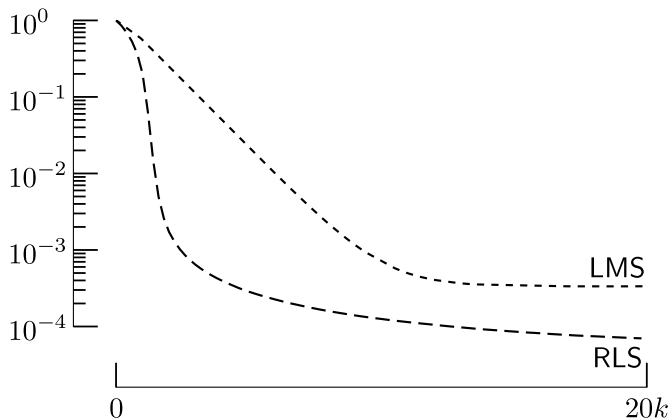
Complexity RLS vs RHS

Filter length : 1000



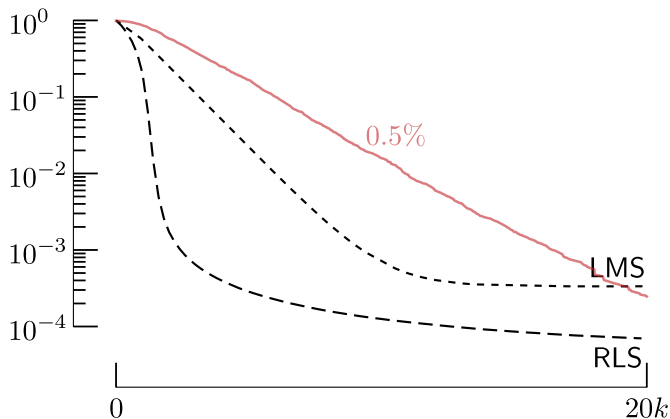
Simulation results — MSE, SNR 30dB

Filter length 1000, $N = 5$, 300 realizations



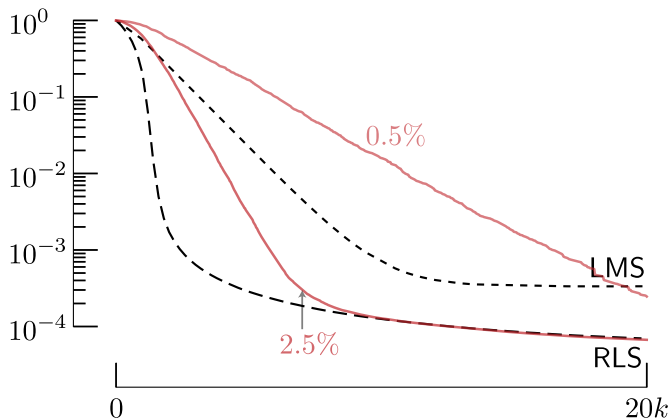
Simulation results — MSE, SNR 30dB

Filter length 1000, $N = 5$, 300 realizations



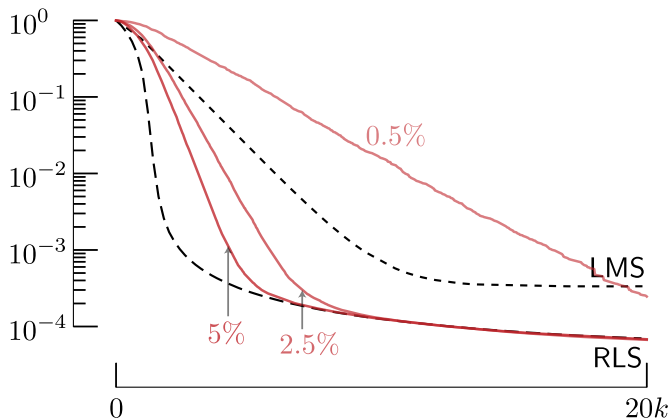
Simulation results — MSE, SNR 30dB

Filter length 1000, $N = 5$, 300 realizations



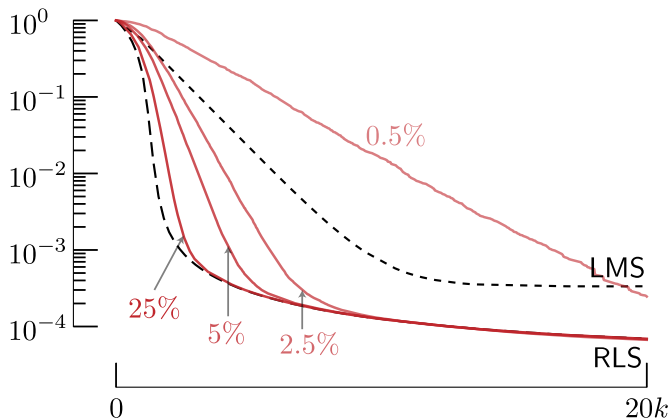
Simulation results — MSE, SNR 30dB

Filter length 1000, $N = 5$, 300 realizations



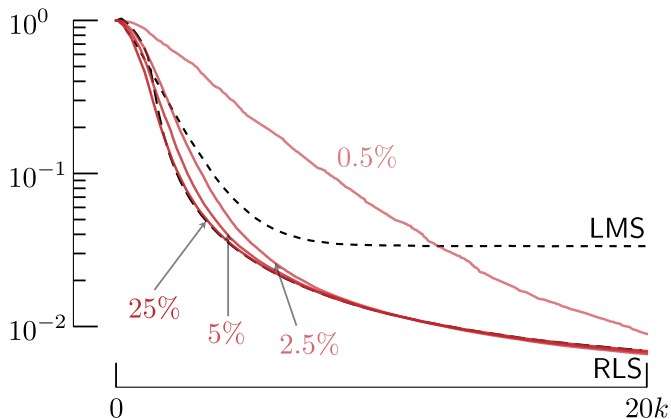
Simulation results — MSE, SNR 30dB

Filter length 1000, $N = 5$, 300 realizations



Simulation results — MSE, SNR 10dB

Filter length 1000, $N = 5$, 300 realizations



Contributions

- A sketched adaptive filter converging to RLS solution
- Lower computational complexity
- Extensive simulation

What's next ?

- Proof of IHS for random row sampling
- Experiments with non-stationary input (e.g. audio, speech)
- Investigate tracking behavior

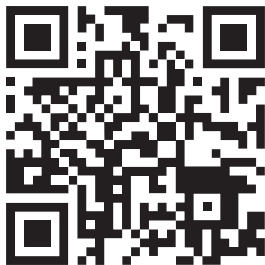
Contributions

- A sketched adaptive filter converging to RLS solution
- Lower computational complexity
- Extensive simulation

What's next ?

- Proof of IHS for random row sampling
- Experiments with non-stationary input (e.g. audio, speech)
- Investigate tracking behavior

Thanks for your attention!



Code and figures available at
<http://github.com/LCAV/SketchRLS/>