The Power-Oja Method for Decentralized Subspace Estimation/Tracking

Sissi Xiaoxiao Wu†, Hoi-To Wai†, Anna Scaglione†, Neil A. Jacklin⋆

†Arizona State University
⋆Northrop Grumman Mission Systems

March 1, 2017
Subspace Estimation

New Challenges

▶ Tracking and estimation accuracy.
▶ Distributed system: separate antennas.
▶ Large scale: massive array.
▶ Complexity: distribute computations to local processors.

Figure: Two examples: WiFi networks and Radar networks.
Literature Survey

- **The power method [Gv96]:**
  - A batch processing method with fast convergence.
  - Non-adaptive, high latency
  - Guarantee for a rank-\(p\) subspace.
  - Computations can be decentralized [LSM11].

- **The Oja’s method [OK85]**
  - A stochastic gradient decent (SGD) method adaptive for tracking varying statistics.
  - First order method: suffer from slow convergence.
  - unconstrained SGD, no guarantee for a rank-\(p\) subspace.
  - Computations can be decentralized [SPK08, SPZ16].

- **The key of the decentralization is average consensus [LSM11, SPK08, SPZ16, BDF13]**
  - Data are often measured distributively over large networks.
  - Gossip-based consensus algorithms solve multi-agent coordination and optimization problems in a decentralized manner.
  - Their key features are
    - ✓ built-in fault tolerance to intermittent computation/communication.
    - ✓ self reorganization to automatic failure correction.
Problem Statement

- We consider a non-stationary stochastic process $r(t) \in \mathbb{C}^N$ and let $\mathcal{T} \subset \{1, 2, \ldots\}$ be a sampling set. Define the sampled covariance:

$$
\hat{R}(\mathcal{T}) := |\mathcal{T}|^{-1} \sum_{s \in \mathcal{T}} r(s)r^H(s).
$$

(1)

- We track top $p$-D subspace by tackling the non-convex, stochastic optimization:

$$
\min_{U \in \mathbb{C}^{N \times p}} f_t(U) := \mathbb{E} \left[ \|r(t) -UU^H r(t)\|^2 \right], \forall \ t \geq 1.
$$

(2)

- We follow the stochastic approximation to the objective function $f(U)$:

$$
\hat{f}(U; \mathcal{T}_\tau) := \text{Tr} \left( (UU^HUU^H - 2UU^H)\hat{R}(\mathcal{T}_\tau) \right),
$$

(3)

where $\mathcal{T}_\tau \subset \{1, 2, \ldots\}$ is the set of observations made during the $\tau$th batch.

- If $r(t)$ is stationary for all $t \in \mathcal{T}_\tau$, then $\mathbb{E}[\hat{f}(U; \mathcal{T}_\tau)] = f_t(U)$.
- When $|\mathcal{T}_\tau|$ is large, $\hat{f}(U; \mathcal{T}_\tau)$ is a good approximation for $f_t(U)$.
- No unitary constraint on the subspace $U$, no guarantee for a rank-$p$ subspace.
Review the Power Method (PM)

The PM works with a whole batch of samples in $\mathcal{T}_\tau$.

- **Step 1:** Generate a random vector as an initial point $\tilde{u}^k(1, \tau)$
- **Step 2:** For $k = 1, \ldots, p$

$$
\tilde{u}^k(\ell + 1, \tau) = \hat{R}(\mathcal{T}_\tau) \tilde{u}^k(\ell, \tau) - \sum_{j=1}^{k-1} (\hat{u}^j(\tau))^H \left( \hat{R}(\mathcal{T}_\tau) \tilde{u}^k(\ell, \tau) \right) \hat{u}^j(\tau), \quad \forall \ell = 1, \ldots, L
$$

$$
\hat{u}^k(\tau) := \tilde{u}^k(L, \tau)/\|\tilde{u}^k(L, \tau)\|.
$$

- **Step 3:** Output the top-$p$ subspace: $\hat{U}_{PM}(\tau) := [\hat{u}^1(\tau) \hat{u}^2(\tau) \ldots \hat{u}^p(\tau)]$.

We use $\hat{U}_0$ to initialize the subspace and denote the above power process by

$$
\overline{U}_{PM}(\tau) = PM(\{r(s)\}_{s \in \mathcal{T}_\tau}; \hat{U}_0; L),
$$

(4)
Review the Oja’s Learning Rule

The Oja’s learning rule works with one sample of $r(t)$ at a time.

- Let $\hat{U}_{Oja}(t) \in C^{N \times p}$ be an estimate of $U(t)$ at iteration $t$, we perform the updates:

\[
\hat{U}_{Oja}(t + 1) = \hat{U}_{Oja}(t) - \gamma_t \nabla \hat{f}(\hat{U}_{Oja}(t); \{t\}),
\]

(5)

with $\nabla \hat{f}(\hat{U}(t), \{t\}) = -2r(t)r^H(t)\hat{U}(t) + r(t)r^H(t)\hat{U}(t)\hat{U}^H(t)\hat{U}(t) + \hat{U}(t)\hat{U}^H(t)r(t)r^H(t)\hat{U}(t)$.

- Convergence for stationary $r(t)$:
  - When $p = 1$ and $\gamma_t = c/t$, at a sub-linear rate of $O(1/t)$ [BDF13];
  - If $\sum_t \gamma_t = \infty$, $\sum_t \gamma_t^2 < \infty$, converges almost surely to the principal $p$-dimensional subspace, yet the convergence rate is not given.

- In practice, the Oja’s learning rule is often used for non-stationary $r(t)$ with $\gamma_t$. 

Motivations for the Power-Oja (P-Oja) Method

Observations for PM and Oja:

- For PM, when $r(t)$ is non-stationary and $|T_\tau| \ll \infty \rightarrow$ a poor approximation for $\hat{R}(T_\tau)$ to the true covariance $\rightarrow$ degraded performance.
- For Oja, the spectral gap,

$$\sigma_p(\hat{R}(T)) - \sigma_{p+1}(\hat{R}(T))$$

is an important factor in determining the convergence speed [BDF13].

Our motivations:

- We want both advantages of the two methods: tracking and estimation accuracy.
- Try to increase the spectral gap.
How to Increase the Spectral Gap

Our approach:

▶ Modify the stochastic approximation of the objective function:

\[
\hat{f}_{\text{POja}}(U; \mathcal{T}) = \text{Tr}\left( \left( UU^H UU^H - 2UU^H \right)(\hat{R}(\mathcal{T}))^L \right).
\]  

(6)

Apparently, \((\hat{R}(\mathcal{T}))^L\) has a better spectral gap than \(\hat{R}(\mathcal{T})\), i.e.,

\[
\sigma_p((\hat{R}(\mathcal{T}))^L) - \sigma_{p+1}((\hat{R}(\mathcal{T}))^L) > \sigma_p(\hat{R}(\mathcal{T})) - \sigma_{p+1}(\hat{R}(\mathcal{T})).
\]

▶ P-Oja tracks the subspace in a batch by batch manner:

▶ For the \(\tau\)th batch, we have

\[
\nabla \hat{f}_{\text{POja}}(\hat{U}_{\text{POja}}(\tau); \mathcal{T}_{\tau}) = -2(\hat{R}(\mathcal{T}_{\tau}))^L \hat{U}_{\text{POja}}(\tau) + (\hat{R}(\mathcal{T}_{\tau}))^L \hat{U}_{\text{POja}}(\tau) \hat{U}_{\text{POja}}^H(\tau) \hat{U}_{\text{POja}}(\tau)
\]

\[
+ \hat{U}_{\text{POja}}(\tau) \hat{U}_{\text{POja}}^H(\tau)(\hat{R}(\mathcal{T}_{\tau}))^L \hat{U}_{\text{POja}}(\tau).
\]

▶ \((\hat{R}(\mathcal{T}_{\tau}))^L \hat{U}_{\text{POja}}(\tau)\): performing \(L\) rounds of the power iterations on \(\hat{U}_{\text{POja}}(\tau)\) and we can approximately calculate it by \(\hat{U}_{\text{PM}}(\tau) \approx \text{PM}(\{r(s)\}_{s \in \mathcal{T}_{\tau}}; \hat{U}_{\text{POja}}(\tau); L)\).
The Power-Oja (P-Oja) Method

Finally, the P-Oja method is given by the following iterations:

\[
\hat{U}_{\text{P-Oja}}(\tau + 1) = \hat{U}_{\text{P-Oja}}(\tau) - \gamma_{\tau} \hat{\nabla} \hat{f}_{\text{P-Oja}}(\hat{U}_{\text{P-Oja}}(\tau); T_{\tau}) ,
\]

where \( \hat{\nabla} \hat{f}_{\text{P-Oja}} \) is the approximated gradient, evaluated as:

\[
\hat{\nabla} \hat{f}_{\text{P-Oja}}(\hat{U}_{\text{P-Oja}}(\tau); T_{\tau}) = \hat{U}_{PM}(\tau) \hat{U}_{\text{P-Oja}}^{H}(\tau) \hat{U}_{\text{P-Oja}}(\tau) \\
+ \hat{U}_{\text{P-Oja}}(\tau) \hat{U}_{\text{P-Oja}}^{H}(\tau) \hat{U}_{PM}(\tau) - 2 \hat{U}_{PM}(\tau) .
\]

where \( \hat{U}_{PM}(\tau) \approx \text{PM}(\{r(s)\}_{s \in T_{\tau}}; \hat{U}_{\text{P-Oja}}(\tau); L) .\)
Some Remarks for Power-Oja

- The P-Oja method is parametrized by $L$ and $T$
  - $T$: controls the variance in the sampled covariance $\hat{R}(T\tau)$;
  - $L$: the acceleration given by the power method subroutine.

- P-Oja reduces into the Oja’s learning rule when $p = 1$, $T = 1$, $L = 1$.

- If the samples in the batch is sufficient, we are very likely to obtain a rank-$p$ subspace. Recall that no guarantee for a rank-$p$ subspace for Oja.
Decentralization

Preliminaries:

- We denote the communication network between $M$ processor units as an undirected graph $G = (V, E)$ such that $V = \{1, \ldots, M\}$ and $E \subseteq V \times V$.
- The graph is assumed to be sparse and connected.
- A doubly stochastic matrix $W$ associated with $G$, s.t. $[W]_{ij} = 0$ iff $(i, j) \notin E$.
- Each processor unit locally processes its subarray’s sampling data, and meanwhile exchanges information with its neighbors in $G$.

Figure: Two examples: WiFi networks and Radar networks.
Average Consensus

- Stored and computed in processor unit $i$:
  - $r_i(t) \in \mathbb{C}^N = [r_1(t); ...; r_M(t)]$
  - $\hat{u}_i(\ell, \tau) \in \mathbb{C}^N = [\hat{u}_1(\ell, \tau); ...; \hat{u}_M(\ell, \tau)]$
  - $z^0_i := r_i^H(t)\hat{u}_i(\ell, \tau)$ and $z^0_i$

- The power iteration can be expressed as:
  $$\sum_{s \in T_\tau} r(s) r^H(s) \hat{u}(\ell, \tau) = \sum_{s \in T_\tau} r(s) AC^i(\{z^0_j\}_{j=1}^M; K)$$

  with $\|z^K - \left(\sum_{i=1}^M z^0_i / M\right) 1\| \leq \lambda_2(W) |^K \|z_0 - \left(\sum_{i=1}^M z^0_i / M\right) 1\|$, where

  - The convergence rate depends on $\lambda_2(W)$ [DKM+10].
  - $\lim_{K \to \infty} \sum_{j=1}^M z^0_j = M \cdot AC^i(\{z^0_j\}_{j=1}^M; K)$ at a geometric rate in $K$ [DKM+10].
An Illustration of Decentralized Power-Oja

**Key technique in Oja:** 
\[ A^H B = \sum_{i=1}^{M} V_i = \sum_{i=1}^{M} (A_i)^H B_i. \]
Some Remarks for Decentralized Power-Oja

- The message exchanged in the approximate gradient is a $p \times p$ matrix.

- It is crucial for us to choose a proper network topology.

$$\|z^K - \bar{z}1\| \leq |\lambda_2(W)|^K \|z_0 - \bar{z}1\|.$$

- It is more economical to connect the nearby units with a higher probability while the far-apart units with a lower probability.

- Example: small-world graph with optimal constant weights [XB04]:

$$W = I - \frac{2}{\lambda_1(L) + \lambda_{N-1}(L)} L,$$

where $L$ is the Laplacian matrix.
Numerical Simulations: parameter settings

- A massive array with $N = 256$ antennas grouped to $M = 64$ subarrays, each equipped with four antennas.
- $T = 1500$, SNR = 20dB, the power iteration is $L = 20$.
- Degree-6 small-world graph with rewiring probability 0.2.
- $\gamma_t = 5 \times 10^{-4}$ for the Oja’s learning rule, $\gamma_t = 0.01t$ with P-Oja for stationary signals and $\gamma_t = 0.04t$ for non-stationary signals.

**Figure:** Grouping antennas into subarrays with distributed processors for spectrum sensing.
Numerical Simulations: for stationary signals

Figure: The normalized objective value for a constant 2-D signal space.

- The convergence rate increases as $T$ increases.
- The decentralized performance will approach the centralized one as $K$ increases.
- The P-Oja method converges much faster than the Oja's method, and the decentralized algorithms work well under the chosen graph.
Numerical Simulations: for non-stationary signals

Figure: The normalized objective value for a variant 1-D signal space. The legend is the same as that in Fig. 1, except for the number of gossip iterations is now $K = 10$. The diamond-marked curve is the NOV for the conventional power method.

- As $T$ increases, the convergence rate increases.
- The decentralized and centralized methods coincide with each other when $K = 10$.
- The power method cannot track the change of the covariance.
Conclusions

- We propose P-Oja method to integrate the Oja’s learning rule and power method.

- It exhibits both tracking ability and estimation accuracy.

- All the computations are distributed into individual processor units.

- Our simulation results demonstrate that the proposed P-Oja can both track the change of statistic, but converges much faster than the conventional Oja method.
References


Thank You

&&

Question Welcomed!