Incrementally Updated Spectral Embeddings

Vasileios Charisopoulos*, Austin R. Benson, Anil Damle
Cornell University

INTRODUCTION

- **Spectral embeddings** in data science: low-dimensional subspaces aiming to capture significant behavior of data. Examples: PCA, spectral clustering, etc.

$$X := V\Lambda V^\top + V_\perp \Lambda_\perp V_\perp^\top, \quad (1)$$

with data matrix X (e.g. normalized adjacency matrix) and $V \in \mathbb{R}^{n\times r}$ subspace of interest.

- **Essential** to efficiently handle data of temporal / sequentially observed nature (e.g. evolving social networks, data streams from sensors, etc.). Observe sequence of data matrices with eigendecomposition

$$X_t := V_t \Lambda_t V_t^\top + \{\text{residual}\} \quad t = 1, 2, \dots, T, \quad (2)$$

with V_t subspace of interest at time t.

Prohibitive to compute V_t from scratch each time!

How to update V_t in $O(1)$ under minimal assumptions?

SETTING

In our approach, we assume that the data matrix is updated in an additive fashion:

$$X_t := X_{t-1} + E_t, \quad E_t \text{ is “small”} \quad (3)$$

The Davis-Kahan theorem implies (for E_t not too “big”):

$$\text{dist}(V_t, V_{t-1}) \leq \frac{\|E_t V_t\|_2}{\text{gap}_r - \|E_t\|_2} \approx \mathcal{O}(\|E_t\|_2) \quad (4)$$

→ **small changes** only slightly perturb V_{t-1}!

Heuristic: Use previously computed estimate V_{t-1} to “seed” some iterative method (subspace / block Krylov iteration).

- avoids restrictive assumptions of direct methods, accelerates under structured / sparse matrices
- similar idea to recycled Krylov methods (e.g. [2]) for sequences of linear systems, but no rigorous guarantees

ALGORITHM & CHALLENGES

The high-level procedure is summarized in Algorithm 1, where:

- ITERMETHOD is an iterative eigenvector method
- δ_t is an upper bound for the true subspace distance

Algorithm 1: Incremental updates

Input: X_0, V_0, Λ_0, update sequence $\{E_t\}_{t\in[T]}$
for $t = 1, \dots, T$ **do**
  $X_t := X_{t-1} + E_t$;
  compute $\delta_t \geq \text{dist}(V_t, V_{t-1})$ ▷ see (4)
  if $\delta_t > \varepsilon$ **then**
    $V_t, \Lambda_t := \text{ITERMETHOD}(V_{t-1}, \varepsilon)$ ▷ ε-accurate estimate
  end
end

Challenges & solutions in implementation and analysis:

1. V_{t-1} is only known *approximately* → bound for subspace distance under ε-approximate estimates
2. convergence of ITERMETHOD depends on $\frac{\lambda_r}{\lambda_{r+1}}$ → estimate λ_{r+1} using the “deflated” matrix $(I - V_t V_t^\top) X_t (I - V_t V_t^\top)$

THEORY

Proposition 1 (Informal - details in [1]). *When E_t are small, cost per update of Algorithm 1 is upper bounded by*

$$\mathcal{O}\left(r(\lambda_r/\lambda_{r+1}) \log\frac{\delta_t}{\varepsilon} + r(\gamma)\log\frac{1}{\varepsilon}\right) \quad \textit{eigensolver iterations}, \quad (5)$$

where $\gamma \in \mathbb{R}$ controls spectrum decay and $r(\cdot)$ is an eigensolver-specific convergence factor. Moreover, the bound can be **computed before** *each update.*

- Terms in red are bounded from above in real time.
- For certain applications (e.g. sparse adjacency matrices, random matrices), δ_t *simplifies.*

EXPERIMENTS

Methods applied to time-evolving **social network** dataset (Figures 1 and 2) and minute-by-minute household **power consumption** readings (Figure 3).

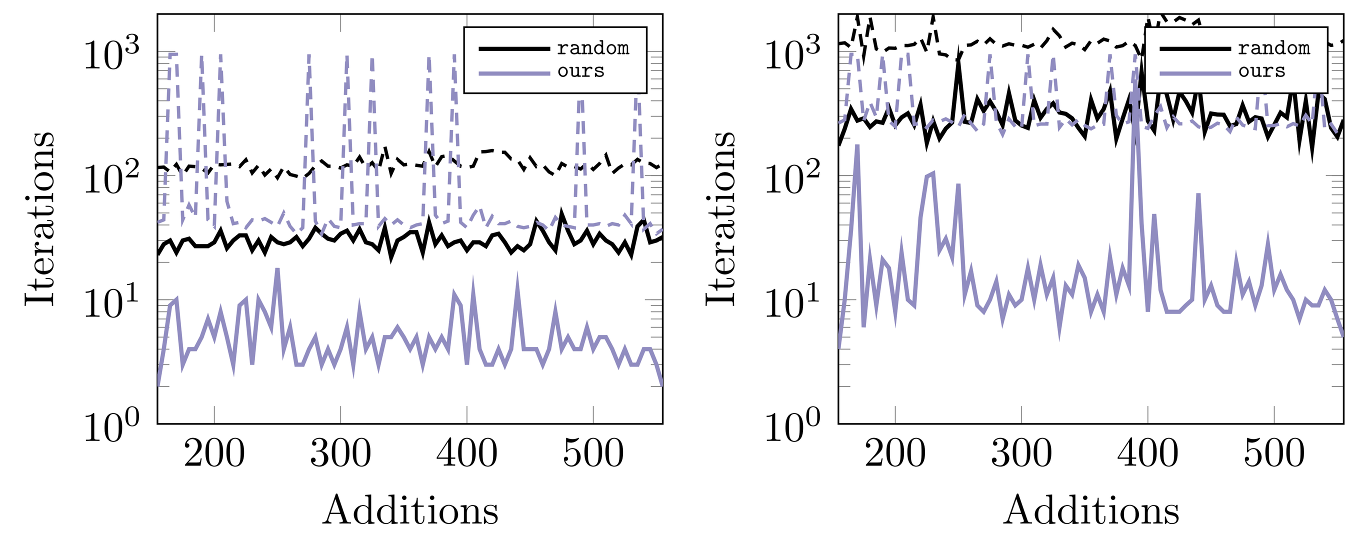

Figure 1: Maintaining the spectral embedding of a graph dataset. Benchmarking against random initialization of V_t (`random`). Dashed lines are upper bounds. Warm-starting clearly outperforms naive initialization.

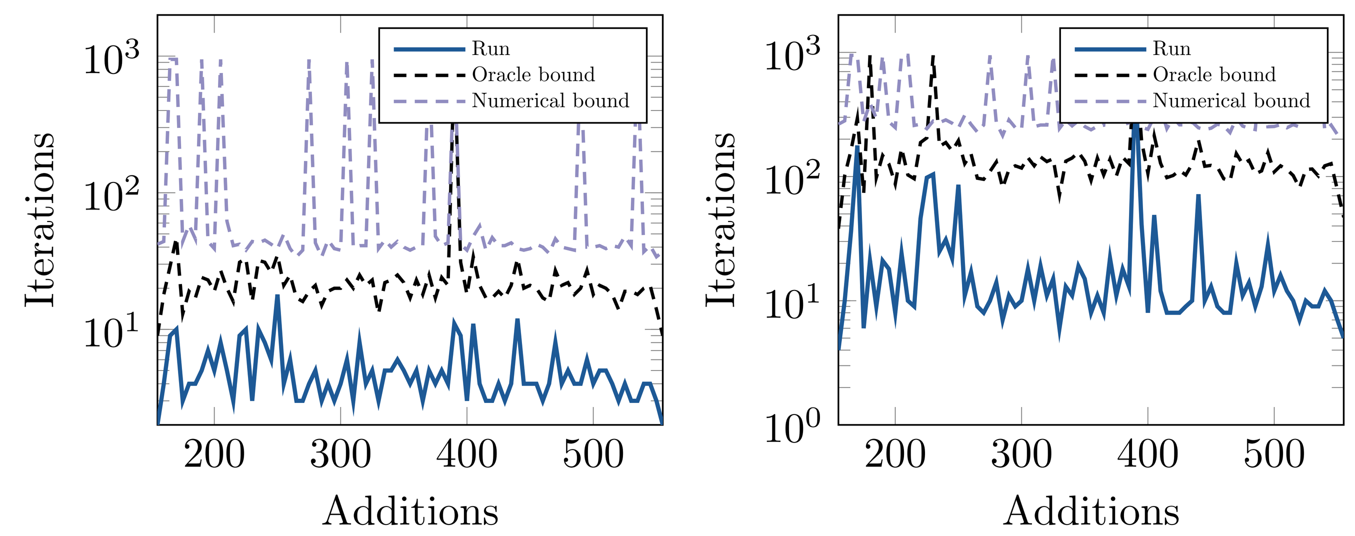

Figure 2: Bounds on # iterations using the oracle subspace distance vs. estimate from (5).

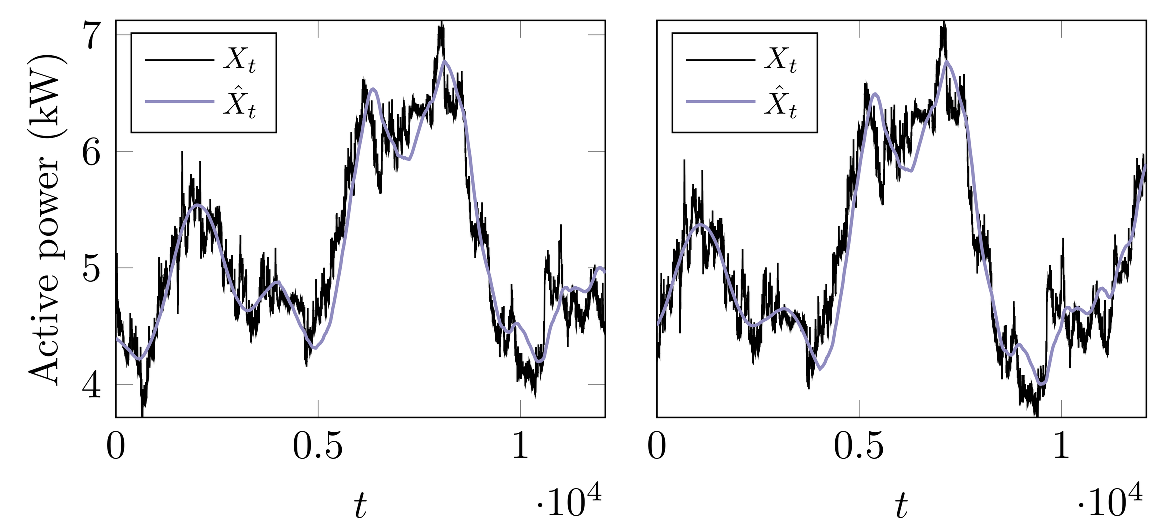

Figure 3: Low-rank reconstruction of time-series. Cost is 2 – 3 iterations per update.

All code made available under

`https://github.com/VHarisop/inc-spectral-embeddings/`

Acknowledgements. Supported by NSF DMS-1830274.

References

[1] Vasileios Charisopoulos, Austin R. Benson, and Anil Damle. Incrementally updated spectral embeddings, 2019, arXiv:1909.01188.

[2] Michael L Parks, Eric De Sturler, Greg Mackey, Duane D Johnson, and Spandan Maiti. Recycling krylov subspaces for sequences of linear systems. *SIAM Journal on Scientific Computing*, 28(5):1651–1674, 2006.