Randomized Short-Recurrence Iterative Methods for Approximate Low-Rank Factorizations

Workshop on
Computational and Mathematical Methods in Data Sciences
Technical University of Chemnitz

Nicolas Venkovic & Prof. Hartwig Anzt

Group of Computational Mathematics
School of Computation, Information and Technology
Technical University of Munich

October 17, 2025

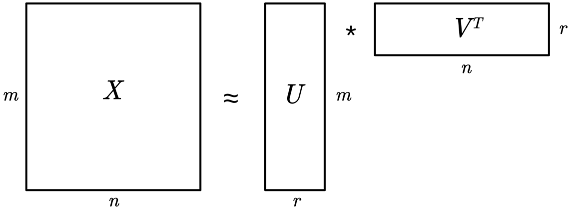
Outline I

Outline II

Introduction

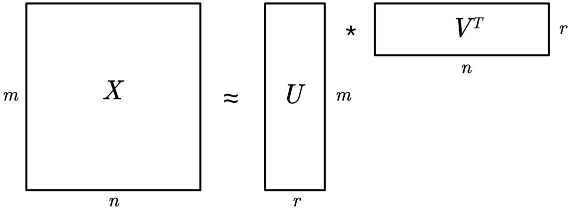
Low-rank matrix approximation — the "What?"

- Given a matrix $X \in \mathbb{R}^{m\times n}$, we seek some factor matrices $U \in \mathbb{R}^{m\times r}$ and $V \in \mathbb{R}^{n\times r}$, with $r \leq \min(m,n)$, s.t. $X - UV^T$ is small in some sense:



Low-rank matrix approximation — the "What?"

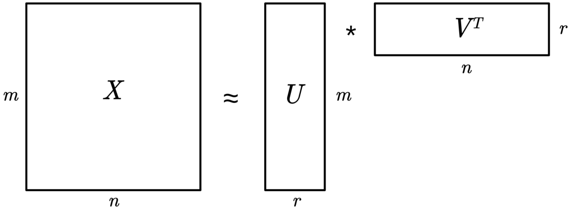
- Given a matrix $X \in \mathbb{R}^{m\times n}$, we seek some factor matrices $U \in \mathbb{R}^{m\times r}$ and $V \in \mathbb{R}^{n\times r}$, with $r \leq \min(m,n)$, s.t. $X - UV^T$ is small in some sense:



- In practice, we often have $r \ll \min(m,n)$.

Low-rank matrix approximation — the "What?"

- Given a matrix $X \in \mathbb{R}^{m \times n}$, we seek some factor matrices $U \in \mathbb{R}^{m \times r}$ and $V \in \mathbb{R}^{n \times r}$, with $r \leq \min(m, n)$, s.t. $X - UV^T$ is small in some sense:

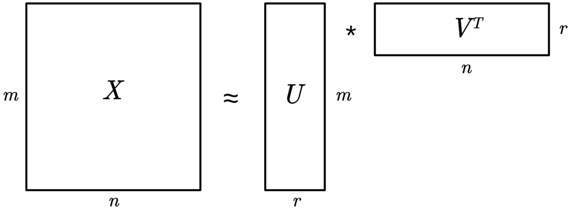


- In practice, we often have $r \ll \min(m, n)$.
- In this talk, we aim at minimizing the Frobenius residual norm:

$$\boxed{\text{Find } (U, V) \in \mathbb{R}^{m \times r} \times \mathbb{R}^{n \times r} \text{ s.t. } f(U, V) := \|X - UV^T\|_F^2 \text{ is minimized}}$$

Low-rank matrix approximation — the "What?"

- Given a matrix $X \in \mathbb{R}^{m \times n}$, we seek some factor matrices $U \in \mathbb{R}^{m \times r}$ and $V \in \mathbb{R}^{n \times r}$, with $r \leq \min(m, n)$, s.t. $X - UV^T$ is small in some sense:



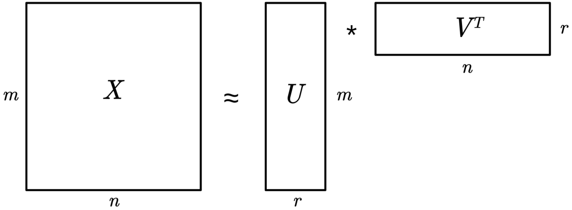
- In practice, we often have $r \ll \min(m, n)$.
- In this talk, we aim at minimizing the Frobenius residual norm:

$$\boxed{\text{Find } (U, V) \in \mathbb{R}^{m \times r} \times \mathbb{R}^{n \times r} \text{ s.t. } f(U, V) := \|X - UV^T\|_F^2 \text{ is minimized}}$$

- non-convex problem (global minima & saddle points).

Low-rank matrix approximation — the "What?"

- Given a matrix $X \in \mathbb{R}^{m\times n}$, we seek some factor matrices $U \in \mathbb{R}^{m\times r}$ and $V \in \mathbb{R}^{n\times r}$, with $r \leq \min(m,n)$, s.t. $X - UV^T$ is small in some sense:



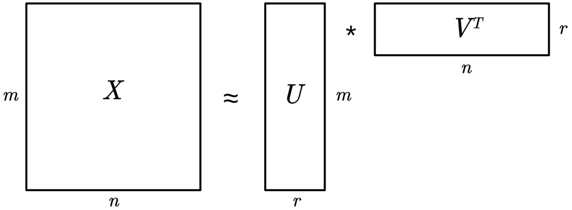
- In practice, we often have $r \ll \min(m,n)$.
- In this talk, we aim at minimizing the Frobenius residual norm:

$$\boxed{\text{Find } (U,V) \in \mathbb{R}^{m\times r} \times \mathbb{R}^{n\times r} \text{ s.t. } f(U,V) := \|X - UV^T\|_F^2 \text{ is minimized}}$$

- non-convex problem (global minima & saddle points).
- f is invariant under orthogonal transformation, i.e., $f(UQ,VQ) = f(U,V)$.

Low-rank matrix approximation — the "What?"

- Given a matrix $X \in \mathbb{R}^{m \times n}$, we seek some factor matrices $U \in \mathbb{R}^{m \times r}$ and $V \in \mathbb{R}^{n \times r}$, with $r \leq \min(m, n)$, s.t. $X - UV^T$ is small in some sense:



- In practice, we often have $r \ll \min(m, n)$.
- In this talk, we aim at minimizing the Frobenius residual norm:

$$\boxed{\text{Find } (U, V) \in \mathbb{R}^{m \times r} \times \mathbb{R}^{n \times r} \text{ s.t. } f(U, V) := \|X - UV^T\|_F^2 \text{ is minimized}}$$

- non-convex problem (global minima & saddle points).
- f is invariant under orthogonal transformation, i.e., $f(UQ, VQ) = f(U, V)$.
- regularization terms can be added to f to promote sparsity, orthogonality, balanced norms, or else, in the factors; making the problem convex, ...

Low-rank matrix approximation — the "Why?"

- Low-rank approximation allows:
  - Data compression:

$r(m+n)$ entries for (U,V) vs mn entries for X.

Low-rank matrix approximation — the "Why?"

- Low-rank approximation allows:
  - Data compression:

    $$r(m+n) \text{ entries for } (U,V) \text{ vs } mn \text{ entries for } X.$$

  - Fast approximation:

    E.g., $2r(m+n)$ FLOPs for $x \mapsto U(V^T x)$ vs $2mn$ FLOPs for $x \mapsto Xx$.

Low-rank matrix approximation — the "Why?"

- Low-rank approximation allows:
  - Data compression:

$$r(m+n) \text{ entries for } (U,V) \text{ vs } mn \text{ entries for } X.$$

  - Fast approximation:

    E.g., $2r(m+n)$ FLOPs for $x \mapsto U(V^T x)$ vs $2mn$ FLOPs for $x \mapsto Xx$.
- Low-rank approximation problems in disguise:
  - Matrix recovery:
    - Matrix completion ("Netflix problem"): retrieve X from partial knowledge of components.

Low-rank matrix approximation — the "Why?"

- Low-rank approximation allows:
  - Data compression:

    $$r(m+n) \text{ entries for } (U,V) \text{ vs } mn \text{ entries for } X.$$

  - Fast approximation:

    E.g., $2r(m+n)$ FLOPs for $x \mapsto U(V^T x)$ vs $2mn$ FLOPs for $x \mapsto Xx$.
- Low-rank approximation problems in disguise:
  - Matrix recovery:
    - Matrix completion ("Netflix problem"): retrieve X from partial knowledge of components.
    - Matrix sensing: retrieve X from (possibly noisy) linear measurements.

Low-rank matrix approximation — the "Why?"

- Low-rank approximation allows:
  - Data compression:

    $$r(m+n) \text{ entries for } (U,V) \text{ vs } mn \text{ entries for } X.$$

  - Fast approximation:

    E.g., $2r(m+n)$ FLOPs for $x \mapsto U(V^T x)$ vs $2mn$ FLOPs for $x \mapsto Xx$.
- Low-rank approximation problems in disguise:
  - Matrix recovery:
    - Matrix completion ("Netflix problem"): retrieve X from partial knowledge of components.
    - Matrix sensing: retrieve X from (possibly noisy) linear measurements.
  - Model order reduction: PCA, truncated KL expansion, POD, ...

Low-rank matrix approximation — the "Why?"

- Low-rank approximation allows:
  - Data compression:

    $$r(m+n) \text{ entries for } (U,V) \text{ vs } mn \text{ entries for } X.$$

  - Fast approximation:

    E.g., $2r(m+n)$ FLOPs for $x \mapsto U(V^T x)$ vs $2mn$ FLOPs for $x \mapsto Xx$.
- Low-rank approximation problems in disguise:
  - Matrix recovery:
    - Matrix completion ("Netflix problem"): retrieve X from partial knowledge of components.
    - Matrix sensing: retrieve X from (possibly noisy) linear measurements.
  - Model order reduction: PCA, truncated KL expansion, POD, ...
  - Regularization of ill-posed problems: reduce high sensitivity of model output to small input variations.

Low-rank matrix approximation — the "Why?"

- Low-rank approximation allows:
  - Data compression:
    $$r(m+n) \text{ entries for } (U,V) \text{ vs } mn \text{ entries for } X.$$
  - Fast approximation:
    E.g., $2r(m+n)$ FLOPs for $x \mapsto U(V^T x)$ vs $2mn$ FLOPs for $x \mapsto Xx$.
- Low-rank approximation problems in disguise:
  - Matrix recovery:
    - Matrix completion ("Netflix problem"): retrieve X from partial knowledge of components.
    - Matrix sensing: retrieve X from (possibly noisy) linear measurements.
  - Model order reduction: PCA, truncated KL expansion, POD, ...
  - Regularization of ill-posed problems: reduce high sensitivity of model output to small input variations.
- Applications in multiple fields:
  - Pre-training of LLMs, recommendation systems, regularization of DNNs, pattern (e.g., face or signal) recognition, preconditioning of challenging numerical problems, approximating dynamical systems, ...

Low-rank matrix approximation — the "How?"

- The Eckart-Young theorem states that truncated singular value decompositions (SVDs) offer optimal low-rank matrix approximations.

Low-rank matrix approximation — the "How?"

- The Eckart-Young theorem states that truncated singular value decompositions (SVDs) offer optimal low-rank matrix approximations.
- Methods for computing SVDs are well-developed:
  - Krylov solvers, QR factorization with column pivoting (CPQR), interpolative decomposition (ID).

Low-rank matrix approximation — the "How?"

- The Eckart-Young theorem states that truncated singular value decompositions (SVDs) offer optimal low-rank matrix approximations.
- Methods for computing SVDs are well-developed:
  - Krylov solvers, QR factorization with column pivoting (CPQR), interpolative decomposition (ID).
- State-of-the-art SVD computing methods are not well-equipped to address the following cases:
  - X is only accessible through incomplete (possibly noisy) linear measurement (matrix recovery problems);

Low-rank matrix approximation — the "How?"

- The Eckart-Young theorem states that truncated singular value decompositions (SVDs) offer optimal low-rank matrix approximations.
- Methods for computing SVDs are well-developed:
  - Krylov solvers, QR factorization with column pivoting (CPQR), interpolative decomposition (ID).
- State-of-the-art SVD computing methods are not well-equipped to address the following cases:
  - X is only accessible through incomplete (possibly noisy) linear measurement (matrix recovery problems);
  - The low-rank factors need satisfy certain structural constraints, e.g., non-negativity (NMF matrix factorization), sparsity;

Low-rank matrix approximation — the "How?"

- The Eckart-Young theorem states that truncated singular value decompositions (SVDs) offer optimal low-rank matrix approximations.
- Methods for computing SVDs are well-developed:
  - Krylov solvers, QR factorization with column pivoting (CPQR), interpolative decomposition (ID).
- State-of-the-art SVD computing methods are not well-equipped to address the following cases:
  - X is only accessible through incomplete (possibly noisy) linear measurement (matrix recovery problems);
  - The low-rank factors need satisfy certain structural constraints, e.g., non-negativity (NMF matrix factorization), sparsity;
  - Non-standard problem regularization, e.g., convex relaxation, promoting factor sparsity, discouraging factor imbalance, enhancing stability, ...

Low-rank matrix approximation — the "How?"

- The Eckart-Young theorem states that truncated singular value decompositions (SVDs) offer optimal low-rank matrix approximations.
- Methods for computing SVDs are well-developed:
  - Krylov solvers, QR factorization with column pivoting (CPQR), interpolative decomposition (ID).
- State-of-the-art SVD computing methods are not well-equipped to address the following cases:
  - X is only accessible through incomplete (possibly noisy) linear measurement (matrix recovery problems);
  - The low-rank factors need satisfy certain structural constraints, e.g., non-negativity (NMF matrix factorization), sparsity;
  - Non-standard problem regularization, e.g., convex relaxation, promoting factor sparsity, discouraging factor imbalance, enhancing stability, ...
- In those cases, gradient descent algorithms and other first-order iterative methods are often the best alternative.

Low-rank matrix approximation — the "How?"

- The Eckart-Young theorem states that truncated singular value decompositions (SVDs) offer optimal low-rank matrix approximations.
- Methods for computing SVDs are well-developed:
  - Krylov solvers, QR factorization with column pivoting (CPQR), interpolative decomposition (ID).
- State-of-the-art SVD computing methods are not well-equipped to address the following cases:
  - X is only accessible through incomplete (possibly noisy) linear measurement (matrix recovery problems);
  - The low-rank factors need satisfy certain structural constraints, e.g., non-negativity (NMF matrix factorization), sparsity;
  - Non-standard problem regularization, e.g., convex relaxation, promoting factor sparsity, discouraging factor imbalance, enhancing stability, ...
- In those cases, gradient descent algorithms and other first-order iterative methods are often the best alternative.
- In this work, we propose enhanced gradient descent algorithms for the computation of approximate low-rank matrix factorizations.

Gradient descent algorithms

- Gradient descent algorithms are defined upon setting search directions of a block coordinate descent along the gradients of the objective function:

$$\nabla_U f(U,V) = -2RV \quad \text{and} \quad \nabla_V f(U,V) = -2R^T U \quad \text{where} \quad R := X - UV.$$

Gradient descent algorithms

- Gradient descent algorithms are defined upon setting search directions of a block coordinate descent along the gradients of the objective function:

$$\nabla_U f(U,V) = -2RV \quad \text{and} \quad \nabla_V f(U,V) = -2R^T U \quad \text{where} \quad R := X - UV.$$

Given a pair of initial approximate low-rank factors $U_0 \in \mathbb{R}^{m\times r}$ and $V_0 \in \mathbb{R}^{n\times r}$, 2-block gradient descent (2BGD) iterates are defined as follows:

Algorithm 2 2BGD(X, U_0, V_0)

$R_0 := X - U_0 V_0^T$
for $t = 0, 1, \dots$ **do**
    $P_t := -R_t V_t$
    $Q_t := -R_t^T U_t$
    $U_{t+1} := U_t + \eta_t P_t$ $\triangleright$ $\eta_t \in (0, \infty)$ is learning rate
    $V_{t+1} := V_t + \eta_t Q_t$
    $R_{t+1} := X - U_{t+1} V_{t+1}^T$

Gradient descent algorithms

- Gradient descent algorithms are defined upon setting search directions of a block coordinate descent along the gradients of the objective function:

$$\nabla_U f(U,V) = -2RV \text{ and } \nabla_V f(U,V) = -2R^T U \text{ where } R := X - UV.$$

Given a pair of initial approximate low-rank factors $U_0 \in \mathbb{R}^{m\times r}$ and $V_0 \in \mathbb{R}^{n\times r}$, 2-block gradient descent (2BGD) iterates are defined as follows:

Algorithm 3 2BGD(X, U_0, V_0)

1: $R_0 := X - U_0 V_0^T$
2: **for** $t = 0, 1, \dots$ **do**
3: $\quad P_t := -R_t V_t$
4: $\quad Q_t := -R_t^T U_t$
5: $\quad U_{t+1} := U_t + \eta_t P_t$ $\qquad \triangleright$ $\eta_t \in (0, \infty)$ is learning rate
6: $\quad V_{t+1} := V_t + \eta_t Q_t$
7: $\quad R_{t+1} := X - U_{t+1} V_{t+1}^T$

Properly decreasing learning rates guarantee convergence to stationary points of f, but we want to try and achieve faster convergence.

Alternating 2-block subspace coordinate descent methods

Alternating 2-block subspace coordinate descent methods

- Consider the Frobenius inner product $(X, Y)_F := \text{tr}(X^T Y)$ with induced norm $\|X\|_F := (X, X)_F^{1/2}$.

Alternating 2-block subspace coordinate descent methods

- Consider the Frobenius inner product $(X, Y)_F := \mathrm{tr}(X^T Y)$ with induced norm $\|X\|_F := (X, X)_F^{1/2}$.
- In this work:
  - Subspace block coordinate descent methods introduced as projections with carefully defined orthogonality constraints.

Alternating 2-block subspace coordinate descent methods

- Consider the Frobenius inner product $(X, Y)_F := \mathrm{tr}(X^T Y)$ with induced norm $\|X\|_F := (X, X)_F^{1/2}$.
- In this work:
  - Subspace block coordinate descent methods introduced as projections with carefully defined orthogonality constraints.
  - Orthogonality: $\boxed{\mathbb{R}^{m\times n} \ni X \perp \mathcal{S} \subset \mathbb{R}^{m\times n} \iff (X, Y)_F = 0 \; \forall \, Y \in \mathcal{S}}$.

Alternating 2-block subspace coordinate descent methods

- Consider the Frobenius inner product $(X,Y)_F := \mathrm{tr}(X^TY)$ with induced norm $\|X\|_F := (X,X)_F^{1/2}$.
- In this work:
  - Subspace block coordinate descent methods introduced as projections with carefully defined orthogonality constraints.
  - Orthogonality: $\boxed{\mathbb{R}^{m\times n} \ni X \perp \mathcal{S} \subset \mathbb{R}^{m\times n} \iff (X,Y)_F = 0 \;\forall\, Y \in \mathcal{S}}$.

Definition (A2BSCD methods)

- Given $X \in \mathbb{R}^{m\times n}$, $U_0 \in \mathbb{R}^{m\times r}$, $V_0 \in \mathbb{R}^{n\times r}$ and $r \leq \min(m,n)$, a sequence of A2BSCD with search directions $P_t \in \mathbb{R}^{m\times r}$ and $Q_t \in \mathbb{R}^{n\times r}$ is defined by:

$$\begin{cases} U_{t+1} \in U_t + \mathrm{span}\{P_t\} \;\text{ s.t. }\; \nabla_U f(U,V_t)|_{U_{t+1}} \propto \tilde{R}_t V_t \perp \mathrm{span}\{P_t\} \\ V_{t+1} \in V_t + \mathrm{span}\{Q_t\} \;\text{ s.t. }\; \nabla_V f(U_{t+1},V)|_{V_{t+1}} \propto R_{t+1}^T U_{t+1} \perp \mathrm{span}\{Q_t\} \end{cases}$$

where $\tilde{R}_t := X - U_{t+1}V_t^T$ and $R_{t+1} := X - U_{t+1}V_{t+1}^T$.
- Proper descent algorithms are s.t. $f(U_{t+1},V_{t+1}) \leq f(U_{t+1},V_t) \leq f(U_t,V_t)$.

Optimality of A2BSCD iterates

- Particular A2BSCD methods are instantiated by the definition of update formulae for the search directions $(P_t, Q_t) \in \mathbb{R}^{m\times r} \times \mathbb{R}^{n\times r}$ for the iterates U_{t+1} and V_{t+1}, respectively, for $t = 0, 1, \ldots$

Optimality of A2BSCD iterates

- Particular A2BSCD methods are instantiated by the definition of update formulae for the search directions $(P_t, Q_t) \in \mathbb{R}^{m\times r} \times \mathbb{R}^{n\times r}$ for the iterates U_{t+1} and V_{t+1}, respectively, for $t = 0, 1, \ldots$
- All A2BSCD iterates are characterized as follows:

Theorem (Optimality of A2BSCD iterates)

Irrespective of the choice of search directions $(P_t, Q_t) \in \mathbb{R}^{m\times r} \times \mathbb{R}^{n\times r}$:

- *all A2BSCD iterates are optimal in the sense that:*

$$U_{t+1} = \arg \min_{U \in U_t + \textit{span}\{P_t\}} f(U, V_t) \;\; \textit{and} \;\; V_{t+1} = \arg \min_{V \in V_t + \textit{span}\{Q_t\}} f(U_{t+1}, V).$$

Optimality of A2BSCD iterates

- Particular A2BSCD methods are instantiated by the definition of update formulae for the search directions $(P_t, Q_t) \in \mathbb{R}^{m\times r} \times \mathbb{R}^{n\times r}$ for the iterates U_{t+1} and V_{t+1}, respectively, for $t = 0, 1, \dots$
- All A2BSCD iterates are characterized as follows:

Theorem (Optimality of A2BSCD iterates)

Irrespective of the choice of search directions $(P_t, Q_t) \in \mathbb{R}^{m\times r} \times \mathbb{R}^{n\times r}$:

- *all A2BSCD iterates are optimal in the sense that:*

$$U_{t+1} = \arg \min_{U \in U_t + \textit{span}\{P_t\}} f(U, V_t) \;\; \textit{and} \;\; V_{t+1} = \arg \min_{V \in V_t + \textit{span}\{Q_t\}} f(U_{t+1}, V).$$

- *all non-trivial A2BSCD methods are proper block descent algorithms, i.e., s.t. $f(U_{t+1}, V_{t+1}) \leq f(U_t, V_t)$, and converge to stationary points of f.*

Optimality of A2BSCD iterates

- Particular A2BSCD methods are instantiated by the definition of update formulae for the search directions $(P_t, Q_t) \in \mathbb{R}^{m\times r} \times \mathbb{R}^{n\times r}$ for the iterates U_{t+1} and V_{t+1}, respectively, for $t = 0, 1, \ldots$
- All A2BSCD iterates are characterized as follows:

Theorem (Optimality of A2BSCD iterates)

Irrespective of the choice of search directions $(P_t, Q_t) \in \mathbb{R}^{m\times r} \times \mathbb{R}^{n\times r}$:

- *all A2BSCD iterates are optimal in the sense that:*

$$U_{t+1} = \arg \min_{U \in U_t + \mathit{span}\{P_t\}} f(U, V_t) \quad \textit{and} \quad V_{t+1} = \arg \min_{V \in V_t + \mathit{span}\{Q_t\}} f(U_{t+1}, V).$$

- *all non-trivial A2BSCD methods are proper block descent algorithms, i.e., s.t. $f(U_{t+1}, V_{t+1}) \le f(U_t, V_t)$, and converge to stationary points of f.*

Proof.

Based on the theorem of orthogonal projections. $\square$

Alternating 2-block subspace gradient descent

- A natural instance of A2BSCD method is obtained by setting the search directions P_t and Q_t along gradient directions of f:

Definition (A2BSGD method and iterates)

- Given $X \in \mathbb{R}^{m\times n}$, $U_0 \in \mathbb{R}^{m\times r}$, $V_0 \in \mathbb{R}^{n\times r}$ and $r \leq \min(m,n)$, a sequence of alternating 2-block subspace gradient descent (A2BSGD) iterates is defined by setting the search directions of the A2BSCD algorithm to:

$$P_t := R_t V_t \propto \nabla_U f(U, V_t)|_{U_t} \quad \text{and} \quad Q_t := \tilde{R}_t^T U_{t+1} \propto \nabla_V f(U_{t+1}, V)|_{V_t}\,.$$

Alternating 2-block subspace gradient descent

- A natural instance of A2BSCD method is obtained by setting the search directions P_t and Q_t along gradient directions of f:

Definition (A2BSGD method and iterates)

- Given $X \in \mathbb{R}^{m\times n}$, $U_0 \in \mathbb{R}^{m\times r}$, $V_0 \in \mathbb{R}^{n\times r}$ and $r \leq \min(m, n)$, a sequence of alternating 2-block subspace gradient descent (A2BSGD) iterates is defined by setting the search directions of the A2BSCD algorithm to:

$$P_t := R_t V_t \propto \nabla_U f(U, V_t)|_{U_t} \quad \text{and} \quad Q_t := \tilde{R}_t^T U_{t+1} \propto \nabla_V f(U_{t+1}, V)|_{V_t}\,.$$

- The main iterates of the A2BSGD method are given by:

$$U_{t+1} := U_t + \alpha_t P_t \quad \text{and} \quad V_{t+1} := V_t + \gamma_t Q_t$$

for $t = 0, 1, \ldots,$

Alternating 2-block subspace gradient descent

- A natural instance of A2BSCD method is obtained by setting the search directions P_t and Q_t along gradient directions of f:

Definition (A2BSGD method and iterates)

- Given $X \in \mathbb{R}^{m\times n}$, $U_0 \in \mathbb{R}^{m\times r}$, $V_0 \in \mathbb{R}^{n\times r}$ and $r \leq \min(m,n)$, a sequence of alternating 2-block subspace gradient descent (A2BSGD) iterates is defined by setting the search directions of the A2BSCD algorithm to:

$$P_t := R_t V_t \propto \nabla_U f(U, V_t)|_{U_t} \quad \text{and} \quad Q_t := \tilde{R}_t^T U_{t+1} \propto \nabla_V f(U_{t+1}, V)|_{V_t}\,.$$

- The main iterates of the A2BSGD method are given by:

$$U_{t+1} := U_t + \alpha_t P_t \quad \text{and} \quad V_{t+1} := V_t + \gamma_t Q_t$$

for $t = 0, 1, \ldots,$ in which the optimal step sizes $\alpha_t, \gamma_t \in \mathbb{R}$ are:

$$\alpha_t := \frac{\|P_t\|_F^2}{(P_t V_t^T V_t, P_t)_F} \quad \text{and} \quad \gamma_t := \frac{\|Q_t\|_F^2}{(Q_t U_{t+1}^T U_{t+1}, Q_t)_F}.$$

Alternating local optimality

- The term "locally optimal" was coined by Andrew Knyazev (2001) to refer to the enrichment with previous search directions of the search space used for the subspace optimization of Rayleigh quotients of symmetric matrices.

Alternating local optimality

- The term "locally optimal" was coined by Andrew Knyazev (2001) to refer to the enrichment with previous search directions of the search space used for the subspace optimization of Rayleigh quotients of symmetric matrices.
- To apply local optimality to the subspace minimization of $f(U, V)$, we stress that the A2BSGD search directions constitute a trivial case of:

$$P_t \in \mathrm{span}\{P_{t-1}, R_t V_t\} \quad \text{and} \quad Q_t \in \mathrm{span}\{Q_{t-1}, \tilde{R}_t^T U_{t+1}\}. \tag{1}$$

Alternating local optimality

- The term "locally optimal" was coined by Andrew Knyazev (2001) to refer to the enrichment with previous search directions of the search space used for the subspace optimization of Rayleigh quotients of symmetric matrices.
- To apply local optimality to the subspace minimization of $f(U,V)$, we stress that the A2BSGD search directions constitute a trivial case of:

$$P_t \in \text{span}\{P_{t-1}, R_tV_t\} \quad \text{and} \quad Q_t \in \text{span}\{Q_{t-1}, \tilde{R}_t^T U_{t+1}\}. \tag{1}$$

Theorem (Alternating local optimality)

Given $X \in \mathbb{R}^{m\times n}$, $U_0 \in \mathbb{R}^{m\times r}$, $V_0 \in \mathbb{R}^{n\times r}$, *and* $r \leq \min(m,n)$, *with search directions satisfying Eq.* (1), $P_{-1} := 0_{m\times r}$ *and* $Q_{-1} := 0_{n\times r}$, *we have:*

$$\begin{cases} \displaystyle\min_{U\in U_t+\textit{span}\{P_{t-1},R_tV_t\}} f(U,V_t) \leq \min_{U\in U_t+\textit{span}\{R_tV_t\}} f(U,V_t) \\ \displaystyle\min_{V\in V_t+\textit{span}\{Q_{t-1},\tilde{R}_t^TU_{t+1}\}} f(U_{t+1},V) \leq \min_{V\in V_t+\textit{span}\{\tilde{R}_t^TU_{t+1}\}} f(U_{t+1},V) \end{cases}$$

for $t = 0, 1, \dots$.

Andrew Knyazev (2001). Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method. SIAM journal on scientific computing, 23(2):517–541.

Alternating 2-block subspace locally optimal gradient descent

Definition (A2BSLOGD method and iterates)

- Given $X \in \mathbb{R}^{m\times n}$, $U_0 \in \mathbb{R}^{m\times r}$, $V_0 \in \mathbb{R}^{n\times r}$ and $r \leq \min(m,n)$, a sequence of A2BSLOGD iterates is defined by:

$$\begin{cases} U_{t+1} & := \arg \min\limits_{U \in U_t + \mathrm{span}\{R_t V_t, P_{t-1}\}} \|X - UV_t^T\|_F \\ V_{t+1} & := \arg \min\limits_{V \in V_t + \mathrm{span}\{\tilde{R}_t^T U_{t+1}, Q_{t-1}\}} \|X - U_{t+1}V^T\|_F \end{cases} \quad \text{for} \quad t = 0, 1, \ldots$$

Alternating 2-block subspace locally optimal gradient descent

Definition (A2BSLOGD method and iterates)

- Given $X \in \mathbb{R}^{m\times n}$, $U_0 \in \mathbb{R}^{m\times r}$, $V_0 \in \mathbb{R}^{n\times r}$ and $r \leq \min(m,n)$, a sequence of A2BSLOGD iterates is defined by:

$$\begin{cases} U_{t+1} & := \arg \min_{U\in U_t + \mathsf{span}\{R_tV_t, P_{t-1}\}} \|X - UV_t^T\|_F \\ V_{t+1} & := \arg \min_{V\in V_t + \mathsf{span}\{\tilde{R}_t^T U_{t+1}, Q_{t-1}\}} \|X - U_{t+1}V^T\|_F \end{cases} \text{for } \ t = 0, 1, \dots$$

- The main left iterates are given by:

$$U_{t+1} := U_t + \alpha_t R_t V_t + \beta_t P_{t-1} \ \text{ for } \ t = 0, 1, \dots$$

Alternating 2-block subspace locally optimal gradient descent

Definition (A2BSLOGD method and iterates)

- Given $X \in \mathbb{R}^{m\times n}$, $U_0 \in \mathbb{R}^{m\times r}$, $V_0 \in \mathbb{R}^{n\times r}$ and $r \leq \min(m,n)$, a sequence of A2BSLOGD iterates is defined by:

$$\begin{cases} U_{t+1} & := \arg \min_{U \in U_t + \text{span}\{R_t V_t, P_{t-1}\}} \|X - U V_t^T\|_F \\ V_{t+1} & := \arg \min_{V \in V_t + \text{span}\{\tilde{R}_t^T U_{t+1}, Q_{t-1}\}} \|X - U_{t+1} V^T\|_F \end{cases} \quad \text{for} \quad t = 0, 1, \ldots$$

- The main left iterates are given by:

$$U_{t+1} := U_t + \alpha_t R_t V_t + \beta_t P_{t-1} \quad \text{for} \quad t = 0, 1, \ldots$$

where the optimal step sizes α_t and β_t depend on:

$$\|R_t V_t\|_F^2,\ (R_t V_t V_t^T V_t, R_t V_t)_F,$$
$$(P_{t-1} V_t^T V_t, P_{t-1})_F,\ (R_t V_t V_t^T V_t, P_{t-1})_F,\ (R_t V_t, P_{t-1})_F$$

Alternating 2-block subspace locally optimal gradient descent

Definition (A2BSLOGD method and iterates)

- Given $X \in \mathbb{R}^{m\times n}$, $U_0 \in \mathbb{R}^{m\times r}$, $V_0 \in \mathbb{R}^{n\times r}$ and $r \leq \min(m,n)$, a sequence of A2BSLOGD iterates is defined by:

$$\begin{cases} U_{t+1} & := \arg \min\limits_{U \in U_t + \text{span}\{R_t V_t, P_{t-1}\}} \|X - UV_t^T\|_F \\ V_{t+1} & := \arg \min\limits_{V \in V_t + \text{span}\{\tilde{R}_t^T U_{t+1}, Q_{t-1}\}} \|X - U_{t+1}V^T\|_F \end{cases} \text{for} \;\; t = 0, 1, \ldots$$

- The main left iterates are given by:

$$U_{t+1} := U_t + \alpha_t R_t V_t + \beta_t P_{t-1} \;\; \text{for} \;\; t = 0, 1, \ldots$$

where the optimal step sizes α_t and β_t depend on:

$$\|R_t V_t\|_F^2,\ (R_t V_t V_t^T V_t, R_t V_t)_F,$$
$$(P_{t-1} V_t^T V_t, P_{t-1})_F,\ (R_t V_t V_t^T V_t, P_{t-1})_F,\ (R_t V_t, P_{t-1})_F$$

and the corresponding search direction is updated by:

$$P_t := R_t V_t + (\beta_t / \alpha_t) P_{t-1} \;\; \text{for} \;\; t = 0, 1, \ldots$$

Alternating 2-block subspace locally optimal gradient descent

Definition (A2BSLOGD method and iterates, cont'd)

- The main right iterates of the A2BSLOGD method are given by:

$$V_{t+1} := V_t + \gamma_t \tilde{R}_t^T U_{t+1} + \omega_t Q_{t-1} \quad \text{for} \quad t = 0, 1, \ldots$$

Alternating 2-block subspace locally optimal gradient descent

Definition (A2BSLOGD method and iterates, cont'd)

- The main right iterates of the A2BSLOGD method are given by:

$$V_{t+1} := V_t + \gamma_t \tilde{R}_t^T U_{t+1} + \omega_t Q_{t-1} \;\text{ for }\; t = 0, 1, \ldots$$

where the optimal step sizes γ_t and ω_t depend on:

$$\|\tilde{R}_t^T U_{t+1}\|_F^2,\ (\tilde{R}_t^T U_{t+1} U_{t+1}^T U_{t+1}, \tilde{R}_t^T U_{t+1})_F$$

$$(Q_{t-1} U_{t+1}^T U_{t+1}, Q_{t-1})_F,\ (\tilde{R}_t^T U_{t+1} U_{t+1}^T U_{t+1}, Q_{t-1})_F,\ (\tilde{R}_t^T U_{t+1}, Q_{t-1})_F$$

Alternating 2-block subspace locally optimal gradient descent

Definition (A2BSLOGD method and iterates, cont'd)

- The main right iterates of the A2BSLOGD method are given by:

$$V_{t+1} := V_t + \gamma_t \tilde{R}_t^T U_{t+1} + \omega_t Q_{t-1} \quad \text{for} \quad t = 0, 1, \dots$$

where the optimal step sizes γ_t and ω_t depend on:

$$\|\tilde{R}_t^T U_{t+1}\|_F^2,\ (\tilde{R}_t^T U_{t+1} U_{t+1}^T U_{t+1}, \tilde{R}_t^T U_{t+1})_F$$

$$(Q_{t-1} U_{t+1}^T U_{t+1}, Q_{t-1})_F,\ (\tilde{R}_t^T U_{t+1} U_{t+1}^T U_{t+1}, Q_{t-1})_F,\ (\tilde{R}_t^T U_{t+1}, Q_{t-1})_F$$

and the corresponding search direction is updated by:

$$Q_t := \tilde{R}_t^T U_{t+1} + (\omega_t / \gamma_t) Q_{t-1} \quad \text{for} \quad t = 0, 1, \dots$$

Simultaneous 2-block subspace coordinate descent methods

Simultaneous 2-block subspace coordinate descent methods

- Until now, we attempted to find stationary points of $f(U, V)$ by alternating between fixing one factor and minimizing the (convex) function in terms of the other factor.

Simultaneous 2-block subspace coordinate descent methods

- Until now, we attempted to find stationary points of $f(U, V)$ by alternating between fixing one factor and minimizing the (convex) function in terms of the other factor.
- Now, we intend to simultaneously produce pairs of new left and right iterates. We refer to those approaches as simultaneous 2-block subspace coordinate descent (S2BSCD) methods:

Simultaneous 2-block subspace coordinate descent methods

- Until now, we attempted to find stationary points of $f(U, V)$ by alternating between fixing one factor and minimizing the (convex) function in terms of the other factor.
- Now, we intend to simultaneously produce pairs of new left and right iterates. We refer to those approaches as simultaneous 2-block subspace coordinate descent (S2BSCD) methods:

Definition (S2BSCD methods)

- Given $X \in \mathbb{R}^{m\times n}$, $U_0 \in \mathbb{R}^{m\times r}$, $V_0 \in \mathbb{R}^{n\times r}$ and $r \leq \min(m, n)$, a sequence of simultaneous 2-block subspace coordinate descent iterates is defined by:

$$\begin{cases} U_{t+1} \in U_t + \text{span}\{P_t\} \\ V_{t+1} \in V_t + \text{span}\{Q_t\} \end{cases} \quad \text{s.t.} \quad \begin{cases} R_{t+1} V_t \perp \text{span}\{P_t\} \\ R_{t+1}^T U_t \perp \text{span}\{Q_t\} \end{cases}$$

where $P_t \in \mathbb{R}^{m\times r}$ and $Q_t \in \mathbb{R}^{n\times r}$ are given search directions.
- Proper descent algorithms are s.t. $f(U_{t+1}, V_{t+1}) \leq f(U_t, V_t)$.

Simultaneous 2-block subspace gradient descent

- A natural instance of S2BSCD method is obtained by setting the search directions P_t and Q_t along gradient directions of f:

Definition (S2BSGD method and iterates)

- Given $X \in \mathbb{R}^{m\times n}$, $U_0 \in \mathbb{R}^{m\times r}$, $V_0 \in \mathbb{R}^{n\times r}$ and $r \leq \min(m, n)$, a sequence of simultaneous 2-block subspace gradient descent (S2BSGD) iterates is defined by setting the search directions of the S2BSCD algorithm to:

$$P_t := R_t V_t \propto \nabla_U f(U, V_t)|_{U_t} \quad \text{and} \quad Q_t := R_t^T U_t \propto \nabla_V f(U_t, V)|_{V_t} .$$

Simultaneous 2-block subspace gradient descent

- A natural instance of S2BSCD method is obtained by setting the search directions P_t and Q_t along gradient directions of f:

Definition (S2BSGD method and iterates)

- Given $X \in \mathbb{R}^{m\times n}$, $U_0 \in \mathbb{R}^{m\times r}$, $V_0 \in \mathbb{R}^{n\times r}$ and $r \leq \min(m,n)$, a sequence of simultaneous 2-block subspace gradient descent (S2BSGD) iterates is defined by setting the search directions of the S2BSCD algorithm to:

$$P_t := R_t V_t \propto \nabla_U f(U, V_t)|_{U_t} \quad \text{and} \quad Q_t := R_t^T U_t \propto \nabla_V f(U_t, V)|_{V_t} \,.$$

- The main iterates of the S2BSGD method are given by:

$$U_{t+1} := U_t + \alpha_t P_t \quad \text{and} \quad V_{t+1} := V_t + \gamma_t Q_t$$

for $t = 0, 1, \ldots,$

Simultaneous 2-block subspace gradient descent

- A natural instance of S2BSCD method is obtained by setting the search directions P_t and Q_t along gradient directions of f:

Definition (S2BSGD method and iterates)

- Given $X \in \mathbb{R}^{m\times n}$, $U_0 \in \mathbb{R}^{m\times r}$, $V_0 \in \mathbb{R}^{n\times r}$ and $r \leq \min(m,n)$, a sequence of simultaneous 2-block subspace gradient descent (S2BSGD) iterates is defined by setting the search directions of the S2BSCD algorithm to:

$$P_t := R_t V_t \propto \nabla_U f(U, V_t)|_{U_t} \quad \text{and} \quad Q_t := R_t^T U_t \propto \nabla_V f(U_t, V)|_{V_t}.$$

- The main iterates of the S2BSGD method are given by:

$$U_{t+1} := U_t + \alpha_t P_t \quad \text{and} \quad V_{t+1} := V_t + \gamma_t Q_t$$

for $t = 0, 1, \ldots$, in which the optimal step sizes $\alpha_t, \gamma_t \in \mathbb{R}$ depend on:

$$\|P_t\|_F^2, (P_t V_t^T V_t, P_t)_F, (U_t Q_t^T V_t, P_t)_F, (P_t Q_t^T V_t, P_t)_F,$$
$$\|Q_t\|_F^2, (V_t P_t^T U_t, Q_t)_F, (Q_t U_t^T U_t, Q_t)_F, (Q_t P_t^T U_t, Q_t)_F.$$

Simultaneous 2-block subspace locally optimal gradient descent

Definition (S2BSLOGD method and iterates)

- Given $X \in \mathbb{R}^{m\times n}$, $U_0 \in \mathbb{R}^{m\times r}$, $V_0 \in \mathbb{R}^{n\times r}$ and $r \leq \min(m, n)$, a sequence of S2BSLOGD iterates is defined by:

$$\begin{cases} U_{t+1} \in U_t + \mathsf{span}\{R_t V_t, P_{t-1}\} \\ V_{t+1} \in V_t + \mathsf{span}\{R_t^T U_t, Q_{t-1}\} \end{cases} \quad \text{s.t.} \quad \begin{cases} R_{t+1} V_t \perp \mathsf{span}\{R_t V_t, P_{t-1}\} \\ R_{t+1}^T U_t \perp \mathsf{span}\{R_t^T U_t, Q_{t-1}\} \end{cases}$$

for $t = 0, 1, \ldots$

Simultaneous 2-block subspace locally optimal gradient descent

Definition (S2BSLOGD method and iterates)

- Given $X \in \mathbb{R}^{m\times n}$, $U_0 \in \mathbb{R}^{m\times r}$, $V_0 \in \mathbb{R}^{n\times r}$ and $r \leq \min(m,n)$, a sequence of S2BSLOGD iterates is defined by:

$$\begin{cases} U_{t+1} \in U_t + \text{span}\{R_tV_t, P_{t-1}\} \\ V_{t+1} \in V_t + \text{span}\{R_t^TU_t, Q_{t-1}\} \end{cases} \quad \text{s.t.} \quad \begin{cases} R_{t+1}V_t \perp \text{span}\{R_tV_t, P_{t-1}\} \\ R_{t+1}^TU_t \perp \text{span}\{R_t^TU_t, Q_{t-1}\} \end{cases}$$

for $t = 0, 1, \ldots$

- The main left iterates are given by:

$$\begin{cases} U_{t+1} := U_t + \alpha_tR_tV_t + \beta_tP_{t-1} \\ V_{t+1} := V_t + \gamma_t\tilde{R}_t^TU_{t+1} + \omega_tQ_{t-1} \end{cases} \quad \text{for} \;\; t = 0, 1, \ldots$$

Simultaneous 2-block subspace locally optimal gradient descent

Definition (S2BSLOGD method and iterates)

- Given $X \in \mathbb{R}^{m\times n}$, $U_0 \in \mathbb{R}^{m\times r}$, $V_0 \in \mathbb{R}^{n\times r}$ and $r \leq \min(m,n)$, a sequence of S2BSLOGD iterates is defined by:

$$\begin{cases} U_{t+1} \in U_t + \text{span}\{R_tV_t, P_{t-1}\} \\ V_{t+1} \in V_t + \text{span}\{R_t^TU_t, Q_{t-1}\} \end{cases} \quad \text{s.t.} \quad \begin{cases} R_{t+1}V_t \perp \text{span}\{R_tV_t, P_{t-1}\} \\ R_{t+1}^TU_t \perp \text{span}\{R_t^TU_t, Q_{t-1}\} \end{cases}$$

  for $t = 0, 1, \ldots$

- The main left iterates are given by:

$$\begin{cases} U_{t+1} := U_t + \alpha_t R_t V_t + \beta_t P_{t-1} \\ V_{t+1} := V_t + \gamma_t \tilde{R}_t^T U_{t+1} + \omega_t Q_{t-1} \end{cases} \quad \text{for} \;\; t = 0, 1, \ldots$$

  and the corresponding search directions are updated by:

$$\begin{cases} P_t := R_tV_t + (\beta_t/\alpha_t)P_{t-1} \\ Q_t := \tilde{R}_t^TU_{t+1} + (\omega_t/\gamma_t)Q_{t-1} \end{cases} \quad \text{for} \;\; t = 0, 1, \ldots$$

Simultaneous 2-block subspace locally optimal gradient descent

Definition (S2BSLOGD method and iterates, cont'd)

- The optimal step sizes γ_t, β_t, γ_t and ω_t depend on:

$$\|R_tV_t\|_F^2, (R_tV_t, P_{t-1})_F, (R_tV_tV_t^TV_t, R_tV_t)_F, (R_tV_tV_t^TV_t, P_{t-1})_F,$$
$$(P_{t-1}V_t^TV_t, R_tV_t)_F, (P_{t-1}V_t^TV_t, P_{t-1})_F, (U_tU_t^TR_tV_t, R_tV_t)_F,$$
$$(U_tU_t^TR_tV_t, P_{t-1})_F, (U_tQ_{t-1}^TV_t, R_tV_t)_F, (U_tQ_{t-1}^TV_t, P_{t-1})_F,$$
$$(R_tV_tU_t^TR_tV_t, R_tV_t)_F, (R_tV_tU_t^TR_tV_t, P_{t-1})_F, (R_tV_tQ_{t-1}^TV_t, R_tV_t)_F,$$
$$(R_tV_tQ_{t-1}^TV_t, P_{t-1})_F, (P_{t-1}U_t^TR_tV_t, R_tV_t)_F, (P_{t-1}U_t^TR_tV_t, P_{t-1})_F,$$
$$(P_{t-1}Q_{t-1}^TV_t, R_tV_t)_F, (P_{t-1}Q_{t-1}^TV_t, P_{t-1})_F, \|R_t^TU_t\|_F^2, (R_t^TU_t, Q_{t-1})_F,$$
$$(V_tV_t^TR_t^TU_t, R_t^TU_t)_F, (V_tV_t^TR_t^TU_t, Q_{t-1})_F(V_tP_{t-1}^TU_t, R_t^TU_t)_F,$$
$$(V_tP_{t-1}^TU_t, Q_{t-1})_F, (R_t^TU_tU_t^TU_t, R_t^TU_t)_F, (R_t^TU_tU_t^TU_t, Q_{t-1})_F,$$
$$(Q_{t-1}U_t^TU_t, R_t^TU_t)_F, (Q_{t-1}U_t^TU_t, Q_{t-1})_F, (R_t^TU_tV_t^TR_t^TU_t, R_t^TU_t)_F,$$
$$(R_t^TU_tV_t^TR_t^TU_t, Q_{t-1})_F, (Q_{t-1}V_t^TR_t^TU_t)_F, (Q_{t-1}V_t^TR_t^TU_t, Q_{t-1})_F,$$
$$(R_t^TU_tP_{t-1}^TU_t, R_t^TU_t)_F, (R_t^TU_tP_{t-1}^TU_t, Q_{t-1})_F, (Q_{t-1}P_{t-1}^TU_t, R_t^TU_t)_F,$$
$$(Q_{t-1}P_{t-1}^TU_t, Q_{t-1})_F.$$

Randomization

Random subspace embeddings

- The main building block of randomization is subspace embedding:

Definition ((ε, δ, d)-oblivious embedding)

A random linear map $x \in \mathbb{R}^n \mapsto \Theta x \in \mathbb{R}^k$ is an (ε, δ, d)-oblivious subspace embedding (OSE) of dimension $k < n$ with some $\delta \in (0, 1)$ if, for any d-dimensional subspace $\mathcal{S} \subset \mathbb{R}^n$, we have:

$$\text{Pr}\left\{(1-\varepsilon)\|x\|_2 \leq \|\Theta x\|_2 \leq (1+\varepsilon)\|x\|_2\right\} \geq 1-\delta \;\; \forall \;\; x \in \mathcal{S}.$$

Random subspace embeddings

- The main building block of randomization is subspace embedding:

Definition ((ε, δ, d)-oblivious embedding)

A random linear map $x \in \mathbb{R}^n \mapsto \Theta x \in \mathbb{R}^k$ is an (ε, δ, d)-oblivious subspace embedding (OSE) of dimension $k < n$ with some $\delta \in (0, 1)$ if, for any d-dimensional subspace $\mathcal{S} \subset \mathbb{R}^n$, we have:

$$\mathsf{Pr}\left\{(1-\varepsilon)\|x\|_2 \leq \|\Theta x\|_2 \leq (1+\varepsilon)\|x\|_2\right\} \geq 1-\delta \;\; \forall \;\; x \in \mathcal{S}.$$

Random subspace embeddings can be used to reduce problem dimension, FLOP counts and data movement.

Random subspace embeddings

- The main building block of randomization is subspace embedding:

Definition ((ε, δ, d)-oblivious embedding)

A random linear map $x \in \mathbb{R}^n \mapsto \Theta x \in \mathbb{R}^k$ is an (ε, δ, d)-oblivious subspace embedding (OSE) of dimension $k < n$ with some $\delta \in (0, 1)$ if, for any d-dimensional subspace $\mathcal{S} \subset \mathbb{R}^n$, we have:

$$\Pr\left\{(1-\varepsilon)\|x\|_2 \leq \|\Theta x\|_2 \leq (1+\varepsilon)\|x\|_2\right\} \geq 1-\delta \;\; \forall \;\; x \in \mathcal{S}.$$

Random subspace embeddings can be used to reduce problem dimension, FLOP counts and data movement.

- The practicality of randomization lies in finding subspace embeddings s.t.:
  - The embedding dimension k is sufficiently smaller than n, and
  - Sketching, i.e., the linear map $x \mapsto \Theta x$, can be applied efficiently.

Random subspace embeddings

- The main building block of randomization is subspace embedding:

Definition ((ε, δ, d)-oblivious embedding)

A random linear map $x \in \mathbb{R}^n \mapsto \Theta x \in \mathbb{R}^k$ is an (ε, δ, d)-oblivious subspace embedding (OSE) of dimension $k < n$ with some $\delta \in (0, 1)$ if, for any d-dimensional subspace $\mathcal{S} \subset \mathbb{R}^n$, we have:

$$\Pr\{(1-\varepsilon)\|x\|_2 \leq \|\Theta x\|_2 \leq (1+\varepsilon)\|x\|_2\} \geq 1 - \delta \ \ \forall \ \ x \in \mathcal{S}.$$

Random subspace embeddings can be used to reduce problem dimension, FLOP counts and data movement.

- The practicality of randomization lies in finding subspace embeddings s.t.:
  - The embedding dimension k is sufficiently smaller than n, and
  - Sketching, i.e., the linear map $x \mapsto \Theta x$, can be applied efficiently.
- Common sketching strategies include:
  - CountSketch, scaled random Gaussian matrices, sub-sampled fast transforms, random sign matrices.

Randomization of simultaneous 2-block coordinate descent

- All the block subspace coordinate descent algorithms presented thus far may be recast to make use of randomization in order to reduce FLOP counts and data movement while minimally impacting convergence.

Randomization of simultaneous 2-block coordinate descent

- All the block subspace coordinate descent algorithms presented thus far may be recast to make use of randomization in order to reduce FLOP counts and data movement while minimally impacting convergence.
- For example, we have:

Definition (RS2BSCD methods)

Given a matrix $X \in \mathbb{R}^{m \times n}$ with approximate factors $U_0 \in \mathbb{R}^{m \times r}$, $V_0 \in \mathbb{R}^{n \times r}$ of rank $r \leq \min(m, n)$, and two random subspace embeddings:

$$x \in \mathbb{R}^m \mapsto \Theta_1 x \in \mathbb{R}^k \quad \text{and} \quad y \in \mathbb{R}^n \mapsto \Theta_2 y \in \mathbb{R}^\ell,$$

a sequence of randomized simultaneous 2-block subspace coordinate descent (RS2BSCD) iterates is defined by:

$$\begin{cases} U_{t+1} \in U_t + \mathrm{span}\{P_t\} \\ V_{t+1} \in V_t + \mathrm{span}\{Q_t\} \end{cases} \quad \text{s.t.} \quad \begin{cases} \Theta_1(R_{t+1}V_t) \perp \mathrm{span}\{\Theta_1 P_t\} \\ \Theta_2(R_{t+1}^T U_t) \perp \mathrm{span}\{\Theta_2 Q_t\} \end{cases}$$

where $P_t \in \mathbb{R}^{m \times r}$ and $Q_t \in \mathbb{R}^{n \times r}$ are given search directions.

Summary of methods

Summary of FLOPs per iteration

- Methods for general matrices:

Method	FLOPs per iteration (dense X)
2BGD	$(6r+3)mn + 2r\cdot(m+n)$
A2BSGD	$(8r+4)mn + (4r^2+6r)(m+n)$
RA2BSGD	$(8r+4)mn + (2r^2+2r)(m+n) + (2r^2+4r)(k+\ell)$
A2BSLOGD	$(8r+4)mn + (6r^2+16r)(m+n)$
RA2BSLOGD	$(8r+4)mn + (2r^2+6r)(m+n) + (4r^2+10r)(k+\ell)$
S2BSGD	$(6r+3)mn + (10r^2+10r)(m+n)$
RS2BSGD	$(6r+3)mn + (4r^2+2r)(m+n) + (6r^2+8r)(k+\ell)$
S2BSLOGD	$(6r+3)mn + (22r^2+42r)(m+n)$
RS2BSLOGD	$(6r+3)mn + (6r^2+6r)(m+n) + (16r^2+36r)(k+\ell)$

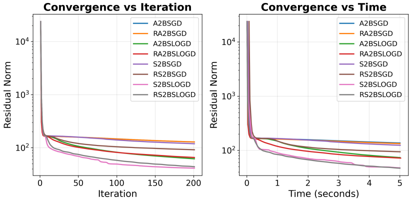
- Methods for symmetric positive semi-definite matrices:

Method	FLOPs per iteration (dense X)
1BGD	$(4r+3)n^2 + 2r\cdot n$
S1BSGD	$(4r+3)n^2 + (10r^2+9r)n$
RS1BSGD	$(4r+3)n^2 + (4r^2+2r)n + (6r^2+7r)k$
S1BSLOGD	$(4r+3)n^2 + (22r^2+33r)n$
RS1BSLOGD	$(4r+3)n^2 + (6r^2+6r)n + (16r^2+27r)k$

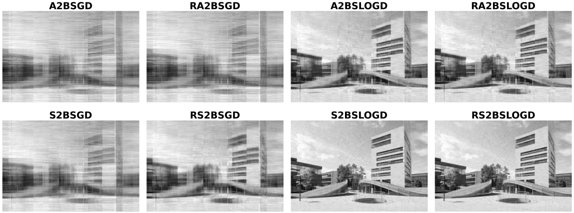
Numerical experiments

Toy example

- Rank-100 approximation of a 800-by-1,200 grayscale image:



- Low-rank approximations achieved in 5 seconds:



- We observe that:
  - Simultaneous schemes converge faster than alternating ones,
  - Local optimality accelerates convergence,
  - Randomization minimally impacts convergence.

Closing remarks

Conclusion

- Findings:
  First-order short-recurrence iterations are introduced for the approximation of low-rank matrix factorizations based on Galerkin projections:

Conclusion

- Findings:
  First-order short-recurrence iterations are introduced for the approximation of low-rank matrix factorizations based on Galerkin projections:
  - Simultaneous methods converge slightly faster than alternating schemes.

Conclusion

- Findings:
  First-order short-recurrence iterations are introduced for the approximation of low-rank matrix factorizations based on Galerkin projections:
  - Simultaneous methods converge slightly faster than alternating schemes.
  - Local optimality, i.e., enriching the search and orthogonality spaces with previous search directions, achieves the fastest convergence behaviors.

Conclusion

- Findings:
  First-order short-recurrence iterations are introduced for the approximation of low-rank matrix factorizations based on Galerkin projections:
  - Simultaneous methods converge slightly faster than alternating schemes.
  - Local optimality, i.e., enriching the search and orthogonality spaces with previous search directions, achieves the fastest convergence behaviors.
  - Extra FLOPs/iteration due to the evaluation of optimal step sizes, though already moderate, are reduced using random embeddings.

Conclusion

- Findings:
  First-order short-recurrence iterations are introduced for the approximation of low-rank matrix factorizations based on Galerkin projections:
  - Simultaneous methods converge slightly faster than alternating schemes.
  - Local optimality, i.e., enriching the search and orthogonality spaces with previous search directions, achieves the fastest convergence behaviors.
  - Extra FLOPs/iteration due to the evaluation of optimal step sizes, though already moderate, are reduced using random embeddings.
  - Convergence behaviors are minimally impacted by randomization.

Conclusion

- Findings:
  First-order short-recurrence iterations are introduced for the approximation of low-rank matrix factorizations based on Galerkin projections:
  - Simultaneous methods converge slightly faster than alternating schemes.
  - Local optimality, i.e., enriching the search and orthogonality spaces with previous search directions, achieves the fastest convergence behaviors.
  - Extra FLOPs/iteration due to the evaluation of optimal step sizes, though already moderate, are reduced using random embeddings.
  - Convergence behaviors are minimally impacted by randomization.
- Dissemination:
  - Preprint to be submitted at the Journal of Machine Learning Research:

    Venkovic & Anzt (2025). Randomized first-order short-recurrence subspace iterative methods for approximate low-rank matrix factorizations.

Conclusion

- Findings:
  First-order short-recurrence iterations are introduced for the approximation of low-rank matrix factorizations based on Galerkin projections:
  - Simultaneous methods converge slightly faster than alternating schemes.
  - Local optimality, i.e., enriching the search and orthogonality spaces with previous search directions, achieves the fastest convergence behaviors.
  - Extra FLOPs/iteration due to the evaluation of optimal step sizes, though already moderate, are reduced using random embeddings.
  - Convergence behaviors are minimally impacted by randomization.
- Dissemination:
  - Preprint to be submitted at the Journal of Machine Learning Research:
    Venkovic & Anzt (2025). Randomized first-order short-recurrence subspace iterative methods for approximate low-rank matrix factorizations.
  - Repository allowing reproducible experiments:
    github.com/venkovic/julia-iterative-low-rank

Conclusion

- Findings:
  First-order short-recurrence iterations are introduced for the approximation of low-rank matrix factorizations based on Galerkin projections:
  - Simultaneous methods converge slightly faster than alternating schemes.
  - Local optimality, i.e., enriching the search and orthogonality spaces with previous search directions, achieves the fastest convergence behaviors.
  - Extra FLOPs/iteration due to the evaluation of optimal step sizes, though already moderate, are reduced using random embeddings.
  - Convergence behaviors are minimally impacted by randomization.
- Dissemination:
  - Preprint to be submitted at the Journal of Machine Learning Research:

    Venkovic & Anzt (2025). Randomized first-order short-recurrence subspace iterative methods for approximate low-rank matrix factorizations.
  - Repository allowing reproducible experiments:

    github.com/venkovic/julia-iterative-low-rank
  - Find this presentation at:

    venkovic.github.io/research

Related ongoing and future work

- Locally optimal sort-recurrence iterative methods for sparse approximate for sparse approximate inverses (SPAIs) of SPD matrices:

  Venkovic & Anzt (2025). Global iterative methods for sparse approximate inverses of symmetric positive-definite matrices.

  github.com/venkovic/julia-global-spd-spai

Related ongoing and future work

- Locally optimal sort-recurrence iterative methods for sparse approximate for sparse approximate inverses (SPAIs) of SPD matrices:

  Venkovic & Anzt (2025). Global iterative methods for sparse approximate inverses of symmetric positive-definite matrices.

  github.com/venkovic/julia-global-spd-spai

- Related ongoing work:
  - Locally optimal short-recurrence iterative methods for SPAIs of general matrices.

- Related future works:
  - SPAIs:
    - Parallelization.

Related ongoing and future work

- Locally optimal sort-recurrence iterative methods for sparse approximate for sparse approximate inverses (SPAIs) of SPD matrices:

  Venkovic & Anzt (2025). Global iterative methods for sparse approximate inverses of symmetric positive-definite matrices.

  github.com/venkovic/julia-global-spd-spai

- Related ongoing work:
  - Locally optimal short-recurrence iterative methods for SPAIs of general matrices.

- Related future works:
  - SPAIs:
    - Parallelization.
  - Low-rank approximation:
    - Application to matrix recovery (completion and sensing problems).
    - Non-negative matrix factorizations.
    - Tensor factorizations.