Randomized Numerical Linear Algebra: Review and Progresses

Zhihua Zhang

Department of Computer Science and Engineering
Shanghai Jiao Tong University

The 12th China Workshop on Machine Learning and Applications
Xi'an, November 2014

- An interdisciplinary among Theoretical Computer Science (TCS), Numerical Linear Algebra (NLA), and Modern Data Analysis
- Many data mining and machine learning algorithms involve matrix decomposition, matrix inverse and matrix determinant; and some methods are based on low-rank matrix approximation.
- The Big Data phenomenon brings new challenges and opportunities to machine learning and data mining.

Singular Value Decomposition (SVD)

- Input: an $m \times n$ data matrix $\mathbf{A}$ of rank r and an integer k less than r.
- The (condensed) SVD: $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ where $\mathbf{U}^T\mathbf{U} = \mathbf{I}_r$, $\mathbf{V}^T\mathbf{V} = \mathbf{I}_r$, and $\mathbf{\Sigma} = \mathrm{diag}(\sigma_1, \ldots, \sigma_r)$ with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$.
  - time complexity: $\mathcal{O}(mn\min(m, n))$
- The truncated SVD: $\mathbf{A}_k = \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^T$ where $\mathbf{U}_k$ and $\mathbf{V}_k$ are the first k columns of $\mathbf{U}$ and $\mathbf{V}$, and $\mathbf{\Sigma}_k$ is the $k \times k$ top sub-block of $\mathbf{\Sigma}$.
  - $\mathbf{A}_k$ is the “closest” rank-k approximation to $\mathbf{A}$. That is,

$$\mathbf{A}_k = \operatorname*{argmin}_{\mathrm{rank}(\mathbf{X}) \leq k} \|\mathbf{A} - \mathbf{X}\|_\xi.$$

    where “$\xi = 2$" is the matrix spectral norm and “$\xi = F$" is the matrix Frobenius norm.
  - time complexity: $\mathcal{O}(mnk)$

Singular Value Decomposition (SVD)

- Input: an $m \times n$ data matrix $\mathbf{A}$ of rank r and an integer k less than r.
- The (condensed) SVD: $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ where $\mathbf{U}^T\mathbf{U} = \mathbf{I}_r$, $\mathbf{V}^T\mathbf{V} = \mathbf{I}_r$, and $\mathbf{\Sigma} = \mathrm{diag}(\sigma_1, \ldots, \sigma_r)$ with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$.
  - time complexity: $\mathcal{O}(mn\min(m,n))$
- The truncated SVD: $\mathbf{A}_k = \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^T$ where $\mathbf{U}_k$ and $\mathbf{V}_k$ are the first k columns of $\mathbf{U}$ and $\mathbf{V}$, and $\mathbf{\Sigma}_k$ is the $k \times k$ top sub-block of $\mathbf{\Sigma}$.
  - $\mathbf{A}_k$ is the "closest" rank-k approximation to $\mathbf{A}$. That is,

$$\mathbf{A}_k = \underset{\mathrm{rank}(\mathbf{X}) \leq k}{\mathrm{argmin}} \|\mathbf{A} - \mathbf{X}\|_\xi.$$

    where "$\xi = 2$" is the matrix spectral norm and "$\xi = F$" is the matrix Frobenius norm.
  - time complexity: $\mathcal{O}(mnk)$

Singular Value Decomposition (SVD)

- Input: an $m \times n$ data matrix $\mathbf{A}$ of rank r and an integer k less than r.
- The (condensed) SVD: $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ where $\mathbf{U}^T\mathbf{U} = \mathbf{I}_r$, $\mathbf{V}^T\mathbf{V} = \mathbf{I}_r$, and $\mathbf{\Sigma} = \mathrm{diag}(\sigma_1, \ldots, \sigma_r)$ with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$.
  - time complexity: $\mathcal{O}(mn\min(m,n))$
- The truncated SVD: $\mathbf{A}_k = \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^T$ where $\mathbf{U}_k$ and $\mathbf{V}_k$ are the first k columns of $\mathbf{U}$ and $\mathbf{V}$, and $\mathbf{\Sigma}_k$ is the $k \times k$ top sub-block of $\mathbf{\Sigma}$.
  - $\mathbf{A}_k$ is the "closest" rank-k approximation to $\mathbf{A}$. That is,

$$\mathbf{A}_k = \underset{\mathrm{rank}(\mathbf{X}) \leq k}{\mathrm{argmin}} \|\mathbf{A} - \mathbf{X}\|_\xi .$$

    where "$\xi = 2$" is the matrix spectral norm and "$\xi = F$" is the matrix Frobenius norm.
  - time complexity: $\mathcal{O}(mnk)$

The CUR Decomposition

A CUR decomposition algorithm seeks to find a subset of c columns of $\mathbf{A}$ to form a matrix $\mathbf{C} \in \mathbb{R}^{m \times c}$, a subset of r rows to form a matrix $\mathbf{R} \in \mathbb{R}^{r \times n}$, and an intersection matrix $\mathbf{U} \in \mathbb{R}^{c \times r}$ such that $\|\mathbf{A} - \mathbf{CUR}\|_\xi$ is minimized.

- The CUR decomposition results in an interpretable matrix approximation to $\mathbf{A}$.
- There are $\binom{n}{c}$ possible choices of constructing $\mathbf{C}$ and $\binom{m}{r}$ possible choices of constructing $\mathbf{R}$, so selecting the best subsets is a hard problem.

Kernel Methods

- $\mathbf{K}$: $n \times n$ kernel matrix.
- Matrix inverse $\mathbf{b} = (\mathbf{K} + \alpha \mathbf{I}_n)^{-1}\mathbf{y}$
  - time complexity: $\mathcal{O}(n^3)$
  - performed by Gaussian process regression, least square SVM, kernel ridge regression
- Partial eigenvalue decomposition of $\mathbf{K}$
  - time complexity: $\mathcal{O}(n^2 k)$
  - performed by kernel PCA and some manifold learning methods
- Space complexity: $\mathcal{O}(n^2)$
  - the iterative algorithms go many passes through the data
  - you had better put the entire kernel matrix in RAM
  - if the data does not fit in the RAM, one swap between RAM and disk in each pass.

Approaches for Large Scale Matrix Computations

Random Projection
The Johnson and Lindenstrauss Lemma
Randomized SVD
Subspace Embedding
Random Selection
Column Selection
CUR Decomposition
The Nyström Method
References

- Two typical approaches: incremental and distributed
- Randomized algorithms have been also used.

Outline

1 Random Projection
- The Johnson and Lindenstrauss Lemma
- Randomized SVD

2 Subspace Embedding

3 Random Selection
- Column Selection
- CUR Decomposition
- The Nyström Method

4 References

Outline

1 Random Projection
- The Johnson and Lindenstrauss Lemma
- Randomized SVD

2 Subspace Embedding

3 Random Selection
- Column Selection
- CUR Decomposition
- The Nyström Method

4 References

The Johnson and Lindenstrauss Lemma

- This lemma has been given by Johnson and Lindenstrauss (1984), but the proof was not constructive.
- Indyk and Motwani (1998) and Dasgupta and Gupta (2003) constructed a result based on Gaussian random projection matrix $\mathbf{R} = [r_{ij}]$ where $r_{ij} \overset{iid}{\sim} N(0, 1)$.
- Matoušek (2008) generalized the result to the case that r_{ij}'s are any subgaussian random variables; that is,

$$r_{ij} \overset{iid}{\sim} \mathcal{G}(\nu^2) \ \text{ for } \ \nu \geq 1.$$

The Johnson and Lindenstrauss Lemma

Definition (ϵ-isometry)

Given $\epsilon \in (0, 1)$, a map $f : \mathbb{R}^p \to \mathbb{R}^q$ where $p > q$ is called an ϵ-isometry on set $\mathcal{X} \subset \mathbb{R}^p$ if for every pair $\mathbf{x}, \mathbf{y} \in \mathcal{X}$, we have

$$(1 - \epsilon)\|\mathbf{x} - \mathbf{y}\|_2^2 \leq \|f(\mathbf{x}) - f(\mathbf{y})\|_2^2 \leq (1 + \epsilon)\|\mathbf{x} - \mathbf{y}\|_2^2.$$

We consider the case that f is defined as a linear map $\mathbf{R} \in \mathbb{R}^{q \times p}$. The Basic idea is to construct a random projection $\mathbf{R} \in \mathbb{R}^{q \times p}$ that is an exact isometry "in expectation;" that is, for every $\mathbf{x} \in \mathbf{R}^p$,

$$\mathbb{E}\left[\|\mathbf{R}\mathbf{x}\|_2^2\right] = \|\mathbf{x}\|_2^2.$$

The Johnson and Lindenstrauss Lemma

Theorem (The Johnson and Lindenstrauss Lemma)

Let $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^p$*, and let* $\epsilon, \delta \in (0, 1)$*. Assume that* $\mathbf{R} \in \mathbb{R}^{q \times p}$ *(* $p > q$*) where* $r_{ij} \in \mathcal{G}(\nu^2)$ *for some* $\nu \geq 1$*. If* $q \geq 100\nu^2\epsilon^{-2}\log(n/\sqrt{\delta})$*, then with probability at least* $1-\delta$*,* $\mathbf{R}$ *is an* ϵ*-isometry on* $\mathcal{X}$

$$\Pr\left\{\sup_{\mathbf{y}\in\mathcal{Y}} \left|\|\mathbf{R}\mathbf{y}\|_2^2 - 1\right| \geq \epsilon\right\} \leq \delta.$$

where $\mathcal{Y} = \left\{\frac{\mathbf{x}_i - \mathbf{x}_j}{\|\mathbf{x}_i - \mathbf{x}_j\|_2} : \mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}, \mathbf{x}_i \neq \mathbf{x}_j\right\}.$

Prototype for Randomized SVD

Given an $m \times n$ matrix $\mathbf{A}$, a target number k of singular vectors, and an integer c such that $k < c < \min(m, n)$, a proto-algorithm based on random projection for Singular Value Decomposition (SVD) of $\mathbf{A}$ is as follows.

1. Construct an $m \times c$ column-orthonormal matrix $\mathbf{Q}$ and form $\mathbf{B} = \mathbf{Q}^T\mathbf{A}$;
2. Compute SVD of the small matrix: $\mathbf{B} = \mathbf{U}_B\mathbf{\Sigma}_B\mathbf{V}_B^T$;
3. Set $\tilde{\mathbf{U}} = \mathbf{Q}\mathbf{U}_B$;
4. Return $\tilde{\mathbf{U}}\mathbf{\Sigma}_B\mathbf{V}_B^T$ as an approximate SVD of $\mathbf{A}$, and $\mathbf{U}_{B,k}\mathbf{\Sigma}_{B,k}\mathbf{V}_{B,k}^T$ as a truncated SVD of $\mathbf{A}$.

A Proto-Algorithm for Construction of Random Projection Matrix $\mathbf{Q}$

Let $\mathbf{A}$ be an $m \times n$ matrix, and k be a target number of singular vectors.

1. Generate an $m \times 2k$ Gaussian test matrix $\mathbf{\Omega}$.
2. Form $\mathbf{Y} = (\mathbf{A}\mathbf{A}^T)^{\gamma}\mathbf{A}\mathbf{\Omega}$ where $\gamma = 1$ or $\gamma = 2$.
3. Construct a matrix $\mathbf{Q}$ whose columns form an orthonormal basis for the range of $\mathbf{Y}$.

Computational Complexity for the Randomized SVD

- The randomized SVD procedure requires only $2(\gamma + 1)$ passes over the matrix.
- The flop count is

$$(2\gamma + 2)kT_{mult} + O(k^2(m + n)),$$

where T_{mult} is the flop count of a matrix-vector multiply with $\mathbf{A}$ or $\mathbf{A}^T$.

Theoretical Analysis for the Randomized SVD

Theorem (Halko et al., 2011)

Let $\mathbf{A} \in \mathbb{R}^{m \times n}$. Give an exponent γ and a target number k of singular vectors, where $2 \le k \le \frac{1}{2}\min(m, n)$, running the Randomized SVD algorithm obtains a rank-2k factorization $\tilde{\mathbf{U}}_{2k}\tilde{\mathbf{\Sigma}}_{2k}\tilde{\mathbf{V}}_{2k}^T$. Then

$$\mathbb{E}\|\mathbf{A} - \tilde{\mathbf{U}}_{2k}\tilde{\mathbf{\Sigma}}_{2k}\tilde{\mathbf{V}}_{2k}^T\|_2 \le \left[1 + 4\sqrt{\frac{2\min(m, n)}{k-1}}\right]^{1/(2\gamma+1)} \sigma_{k+1}.$$

where $\mathbb{E}$ is taken w.r.t. the random test matrix and σ_{k+1} is the top $(k+1)$th singular value of $\mathbf{A}$.

Outline

1 Random Projection
- The Johnson and Lindenstrauss Lemma
- Randomized SVD

2 Subspace Embedding

3 Random Selection
- Column Selection
- CUR Decomposition
- The Nyström Method

4 References

The Subspace Embedding Problem

- For a fixed $m \times n$ matrix $\mathbf{A}$ of rank r and an error parameter $\epsilon \in (0, 1)$, we call $\mathbf{S} : \mathbb{R}^m \to \mathbb{R}^k$ a *subspace embedding matrix* for $\mathbf{A}$ if

$$(1 - \epsilon)\|\mathbf{Ax}\|_2 \leq \|\mathbf{SAx}\|_2 \leq (1 + \epsilon)\|\mathbf{Ax}\|_2$$

  for all $\mathbf{x} \in \mathbb{R}^n$.

- The *Subspace Embedding Problem* is to find such an embedding matrix obliviously. More specifically, one designs a distribution π over linear maps from $\mathbb{R}^m$ to $\mathbb{R}^k$ such that for any fixed $m \times n$ matrix $\mathbf{A}$, if we choose $\mathbf{S} \sim \pi$, then with high probability $\mathbf{S}$ is an embedding matrix for $\mathbf{A}$.

The Subspace Embedding Problem

- For a fixed $m \times n$ matrix $\mathbf{A}$ of rank r and an error parameter $\epsilon \in (0, 1)$, we call $\mathbf{S} : \mathbb{R}^m \to \mathbb{R}^k$ a *subspace embedding matrix* for $\mathbf{A}$ if

$$(1-\epsilon)\|\mathbf{Ax}\|_2 \leq \|\mathbf{SAx}\|_2 \leq (1+\epsilon)\|\mathbf{Ax}\|_2$$

  for all $\mathbf{x} \in \mathbb{R}^n$.

- The *Subspace Embedding Problem* is to find such an embedding matrix obliviously. More specifically, one designs a distribution π over linear maps from $\mathbb{R}^m$ to $\mathbb{R}^k$ such that for any fixed $m \times n$ matrix $\mathbf{A}$, if we choose $\mathbf{S} \sim \pi$, then with high probability $\mathbf{S}$ is an embedding matrix for $\mathbf{A}$.

Sparse Embedding Matrices

For a fixed $m \times n$ matrix $\mathbf{A}$ with $m > n$, let $\mathrm{nnz}(\mathbf{A})$ denote the number of non-zero entries of $\mathbf{A}$. Assume that $\mathrm{nnz}(\mathbf{A}) \geq m$ and that there are no all-zero rows or columns in $\mathbf{A}$. Let $[m] = \{1, 2, \ldots, m\}$. For a parameter k, define a random linear map $\mathbf{\Phi D} : \mathbb{R}^m \to \mathbb{R}^k$ as follows

- $h : [m] \to [k]$ is a random map so that for each $i \in [m]$, $h(i) = t$ where $t \in [k]$ with probability $1/k$.
- $\mathbf{\Phi} \in \{0, 1\}^{k \times m}$ is a $k \times m$ binary matrix, with $\phi_{h(i),i} = 1$ and all remaining entries 0.
- $\mathbf{D}$ is an $m \times m$ random diagonal matrix, with each diagonal entry independently chosen to be $+1$ or -1 with equal probability.

A matrix of the form $\mathbf{S} = \mathbf{\Phi D}$ is referred to as a *sparse embedding matrix* (Dasgupta *et al.*, 2010; Clarkson and Woodruff, 2013).

Sparse Embedding Matrices

For a fixed $m \times n$ matrix $\mathbf{A}$ with $m > n$, let $\mathrm{nnz}(\mathbf{A})$ denote the number of non-zero entries of $\mathbf{A}$. Assume that $\mathrm{nnz}(\mathbf{A}) \geq m$ and that there are no all-zero rows or columns in $\mathbf{A}$. Let $[m] = \{1, 2, \ldots, m\}$. For a parameter k, define a random linear map $\mathbf{\Phi D} : \mathbb{R}^m \to \mathbb{R}^k$ as follows

- $h : [m] \to [k]$ is a random map so that for each $i \in [m]$, $h(i) = t$ where $t \in [k]$ with probability $1/k$.
- $\mathbf{\Phi} \in \{0, 1\}^{k \times m}$ is a $k \times m$ binary matrix, with $\phi_{h(i),i} = 1$ and all remaining entries 0.
- $\mathbf{D}$ is an $m \times m$ random diagonal matrix, with each diagonal entry independently chosen to be $+1$ or -1 with equal probability.

A matrix of the form $\mathbf{S} = \mathbf{\Phi D}$ is referred to as a *sparse embedding matrix* (Dasgupta *et al.*, 2010; Clarkson and Woodruff, 2013).

Sparse Embedding Matrices

For a fixed $m \times n$ matrix $\mathbf{A}$ with $m > n$, let $\mathrm{nnz}(\mathbf{A})$ denote the number of non-zero entries of $\mathbf{A}$. Assume that $\mathrm{nnz}(\mathbf{A}) \geq m$ and that there are no all-zero rows or columns in $\mathbf{A}$. Let $[m] = \{1, 2, \ldots, m\}$. For a parameter k, define a random linear map $\mathbf{\Phi D} : \mathbb{R}^m \to \mathbb{R}^k$ as follows

- $h : [m] \to [k]$ is a random map so that for each $i \in [m]$, $h(i) = t$ where $t \in [k]$ with probability $1/k$.
- $\mathbf{\Phi} \in \{0, 1\}^{k \times m}$ is a $k \times m$ binary matrix, with $\phi_{h(i),i} = 1$ and all remaining entries 0.
- $\mathbf{D}$ is an $m \times m$ random diagonal matrix, with each diagonal entry independently chosen to be $+1$ or -1 with equal probability.

A matrix of the form $\mathbf{S} = \mathbf{\Phi D}$ is referred to as a *sparse embedding matrix* (Dasgupta *et al.*, 2010; Clarkson and Woodruff, 2013).

Sparse Embedding Matrices

For a fixed $m \times n$ matrix $\mathbf{A}$ with $m > n$, let $\mathrm{nnz}(\mathbf{A})$ denote the number of non-zero entries of $\mathbf{A}$. Assume that $\mathrm{nnz}(\mathbf{A}) \geq m$ and that there are no all-zero rows or columns in $\mathbf{A}$. Let $[m] = \{1, 2, \ldots, m\}$. For a parameter k, define a random linear map $\mathbf{\Phi D} : \mathbb{R}^m \to \mathbb{R}^k$ as follows

- $h : [m] \to [k]$ is a random map so that for each $i \in [m]$, $h(i) = t$ where $t \in [k]$ with probability $1/k$.
- $\mathbf{\Phi} \in \{0, 1\}^{k \times m}$ is a $k \times m$ binary matrix, with $\phi_{h(i),i} = 1$ and all remaining entries 0.
- $\mathbf{D}$ is an $m \times m$ random diagonal matrix, with each diagonal entry independently chosen to be $+1$ or -1 with equal probability.

A matrix of the form $\mathbf{S} = \mathbf{\Phi D}$ is referred to as a *sparse embedding matrix* (Dasgupta *et al.*, 2010; Clarkson and Woodruff, 2013).

Sparse Embedding Matrices

For a fixed $m \times n$ matrix $\mathbf{A}$ with $m > n$, let $\mathrm{nnz}(\mathbf{A})$ denote the number of non-zero entries of $\mathbf{A}$. Assume that $\mathrm{nnz}(\mathbf{A}) \geq m$ and that there are no all-zero rows or columns in $\mathbf{A}$. Let $[m] = \{1, 2, \ldots, m\}$. For a parameter k, define a random linear map $\mathbf{\Phi D} : \mathbb{R}^m \to \mathbb{R}^k$ as follows

- $h : [m] \to [k]$ is a random map so that for each $i \in [m]$, $h(i) = t$ where $t \in [k]$ with probability $1/k$.
- $\mathbf{\Phi} \in \{0, 1\}^{k \times m}$ is a $k \times m$ binary matrix, with $\phi_{h(i),i} = 1$ and all remaining entries 0.
- $\mathbf{D}$ is an $m \times m$ random diagonal matrix, with each diagonal entry independently chosen to be $+1$ or -1 with equal probability.

A matrix of the form $\mathbf{S} = \mathbf{\Phi D}$ is referred to as a *sparse embedding matrix* (Dasgupta *et al.*, 2010; Clarkson and Woodruff, 2013).

Subspace Embedding in Input-Sparsity Time

Randomized Numerical Linear Algebra

Zhang

Random Projection
The Johnson and Lindenstrauss Lemma
Randomized SVD

Subspace Embedding

Random Selection
Column Selection
CUR Decomposition
The Nyström Method

References

Theorem (Meng and Mahoney, 2013)

Let $\mathbf{S} = \mathbf{\Phi D} \in \mathbb{R}^{k \times m}$ *with* $k = \frac{n^2+n}{\epsilon^2 \delta}$. *Then with probability at least* $1 - \delta$,

$$(1-\epsilon)\|\mathbf{Ax}\|_2 \leq \|\mathbf{SAx}\|_2 \leq (1+\epsilon)\|\mathbf{Ax}\|_2$$

for all $\mathbf{x} \in \mathbb{R}^n$. *In addition,* **SA** *can be computed in* $\mathcal{O}(\mathrm{nnz}(\mathbf{A}))$.

Spectral Sparsifiers

Theorem (Batson, Spielman and Srivastava, 2014)

Suppose $\rho > 1$ *and* $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m\} \subseteq \mathbb{R}^n$ *with*

$$\sum_{i \le m} \mathbf{v}_i \mathbf{v}_i^T = \mathbf{I}_n.$$

Then there exist scalars $d_i \ge 0$ *with* $|\{i : d_i \ne 0\}| \le \lceil \rho n \rceil$ *such that*

$$\left(1 - \frac{1}{\sqrt{\rho}}\right)^2 \mathbf{I}_n \preceq \sum_{i \le m} d_i \mathbf{v}_i \mathbf{v}_i^T \preceq \left(1 + \frac{1}{\sqrt{\rho}}\right)^2 \mathbf{I}_n.$$

This theorem shows that

$$\frac{\lambda_1(\sum_{i \le m} d_i \mathbf{v}_i \mathbf{v}_i^T)}{\lambda_n(\sum_{i \le m} d_i \mathbf{v}_i \mathbf{v}_i^T)} \le \frac{\rho + 1 + 2\sqrt{\rho}}{\rho + 1 - 2\sqrt{\rho}}.$$

Outline

1 Random Projection
- The Johnson and Lindenstrauss Lemma
- Randomized SVD

2 Subspace Embedding

3 Random Selection
- Column Selection
- CUR Decomposition
- The Nyström Method

4 References

Column Selection and The CX Decomposition

- Given an $m \times n$ matrix $\mathbf{A}$, column selection algorithms aim to find a matrix with c columns of $\mathbf{A}$ such that $\|\mathbf{A} - \mathbf{C}\mathbf{C}^{+}\mathbf{A}\|_{\xi} = \|(\mathbf{I}_m - \mathbf{C}\mathbf{C}^{+})\mathbf{A}\|_{\xi}$ achieves the minimum. Here "$\xi = 2$," "$\xi = F$," and "$\xi = *$" respectively represent the matrix spectral norm, the matrix Frobenius norm, and the matrix nuclear norm, and $\mathbf{C}^{+}$ is the Moore-Penrose inverse of $\mathbf{C}$.
- Let $\mathbf{X}$ be the best rank k approximation to $\mathbf{A}$ in the column span of $\mathbf{C}$. Then $\mathbf{C}\mathbf{X}$ is called the CX Decomposition of $\mathbf{A}$.
- Since there are $\binom{n}{c}$ possible choices of constructing $\mathbf{C}$, selecting the best subset is a hard problem.

A Randomized Algorithm for Column Selection

Given an $m \times n$ matrix $\mathbf{A}$ and a rank parameter k, a random sampling based on the statistical leverage score is:

- Compute the importance sampling probabilities $\{\pi_i\}_{i=1}^{n}$. Here $\pi_i = \frac{1}{k}\|{\mathbf{V}_k}^{(i)}\|$, where $\mathbf{V}_k$ is an $n \times k$ orthonormal matrix spanning the top-k right singular subspace of $\mathbf{A}$.
- Randomly select $c = O(k \log(k/\epsilon^2))$ columns of $\mathbf{A}$ according to these probabilities to form the matrix $\mathbf{C}$.

Theoretical Result for the Random Column Selection (Drineas et al., 2008)

Let $\mathbf{C}_k$ be the best rank-k approximation to the matrix $\mathbf{C}$, and define the projection matrix $P_{C_k} = \mathbf{C}_k \mathbf{C}_k^{+}$. Then

$$\|\mathbf{A} - P_{C_k}\mathbf{A}\|_F \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F,$$

where $\mathbf{A}_k = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^T$ is the best rank k approximation to $\mathbf{A}$.

The Adaptive Sampling Algorithm

Lemma (Deshpande et al., 2006)

Given a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, *let* $\mathbf{C}_1 \in \mathbb{R}^{m \times c_1}$ *consist of* c_1 *columns of* $\mathbf{A}$, *and define the residual* $\mathbf{B} = \mathbf{A} - \mathbf{C}_1\mathbf{C}_1^+\mathbf{A}$. *Additionally, for* $i = 1, \cdots, n$, *define*

$$\pi_i \; = \; \|\mathbf{b}_i\|_2^2 / \|\mathbf{B}\|_F^2.$$

We further sample c_2 *columns i.i.d. from* $\mathbf{A}$, *in each trial of which the* i*-th column is chosen with probability* π_i. *Let* $\mathbf{C}_2 \in \mathbb{R}^{m \times c_2}$ *contain the* c_2 *sampled columns and let* $\mathbf{C} = [\mathbf{C}_1, \mathbf{C}_2] \in \mathbb{R}^{m \times (c_1 + c_2)}$. *Then, for any integer* $k > 0$, *the following inequality holds:*

$$\mathbb{E}\|\mathbf{A} - \mathbf{C}\mathbf{C}^+\mathbf{A}\|_F^2 \; \leq \; \|\mathbf{A} - \mathbf{A}_k\|_F^2 + \frac{k}{c_2}\|\mathbf{A} - \mathbf{C}_1\mathbf{C}_1^+\mathbf{A}\|_F^2,$$

where the expectation is taken w.r.t. $\mathbf{C}_2$.

The Near-Optimal Column Selection Algorithm

Boutsidis et al. (2013) derived a near-optimal algorithm, which consists of three steps:

- the approximate SVD via random projection (Halko et al. 2011)
- a dual set sparsification algorithm—an extension of spectral sparsifier (BSS)
- the adaptive sampling algorithm (Deshpande et al., 2006)

The Near-Optimal Column Selection Algorithm

Theorem (Boutsidis et al., 2013)

Given a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ *of rank* ρ*, a target rank* k $(2 \leq k < \rho)$*, and* $0 < \epsilon < 1$*, the algorithm selects*

$$c \;=\; \frac{2k}{\epsilon}\Big(1 + o(1)\Big)$$

columns of $\mathbf{A}$ *to form a matrix* $\mathbf{C} \in \mathbb{R}^{m \times c}$*. Then the following inequality holds:*

$$\mathbb{E}\|\mathbf{A} - \mathbf{C}\mathbf{C}^{+}\mathbf{A}\|_F^2 \;\leq\; (1+\epsilon)\,\|\mathbf{A} - \mathbf{A}_k\|_F^2,$$

where the expectation is taken w.r.t. $\mathbf{C}$*. Furthermore, the matrix* $\mathbf{C}$ *can be obtained in time:*

$$O\big(mk^2\epsilon^{-4/3} + nk^3\epsilon^{-2/3}\big) + T_{Multiply}\big(mnk\epsilon^{-2/3}\big).$$

The CUR Decomposition (Drineas et al., 2008; Mahoney and Drineas, 2009)

Randomized Numerical Linear Algebra

Zhang

Random Projection
The Johnson and Lindenstrauss Lemma
Randomized SVD

Subspace Embedding

Random Selection
Column Selection
CUR Decomposition
The Nyström Method

References

Given an $m \times n$ matrix $\mathbf{A}$, and integers $c < n$ and $r < m$, the CUR decomposition of $\mathbf{A}$ finds $\mathbf{C} \in \mathbb{R}^{m \times c}$ with c columns from $\mathbf{A}$, $\mathbf{R} \in \mathbb{R}^{r \times n}$ with r rows from $\mathbf{A}$, and $\mathbf{U} \in \mathbb{R}^{c \times r}$ such that $\mathbf{A} = \mathbf{CUR} + \mathbf{E}$. Here $\mathbf{E} = \mathbf{A} - \mathbf{CUR}$ is the residual error matrix.

The CUR Problem

Definition (The CUR Decomposition)

Given an $m \times n$ matrix $\mathbf{A}$ of rank ρ, a rank parameter k, and accuracy parameter $\epsilon \in (0, 1)$, construct a matrix $\mathbf{C} \in \mathbb{R}^{m \times c}$ with c columns from $\mathbf{A}$, $\mathbf{R} \in \mathbb{R}^{r \times n}$ with rows from $\mathbf{A}$, and $\mathbf{U} \in \mathbb{R}^{c \times r}$, with c, r, and $\mathrm{rank}(\mathbf{U})$ being as small as possible, such that

$$\|\mathbf{A} - \mathbf{CUR}\|_F^2 \leq (1 + \epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2.$$

Here $\mathbf{A}_k = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^T \in \mathbb{R}^{m \times n}$ is the best rank k matrix obtained via the SVD of $\mathbf{A}$: $\mathbf{A} = \mathbf{U\Sigma V}^T$.

The Subspace Sampling CUR Algorithm

Drineas et al., (2008) proposed a two-stage randomized CUR algorithm that called *Subspace Sampling*.

- The first stage samples c columns of $\mathbf{A}$ to construct $\mathbf{C}$ according to the sampling probabilities proportional to the squared ℓ_2-norm of the rows of $\mathbf{V}_k$;
- The second stage samples r rows from $\mathbf{A}$ and $\mathbf{C}$ simultaneously to construct $\mathbf{R}$ and $\mathbf{W}$ and let $\mathbf{U} = \mathbf{W}^{\dagger}$. The sampling probabilities in this stages are proportional to the leverage scores of $\mathbf{A}$ and $\mathbf{C}$, respectively.

The Subspace Sampling CUR Algorithm

Lemma (Drineas et al., 2008)

Given an $m \times n$ matrix $\mathbf{A}$ and a target rank $k \ll \min\{m, n\}$, the subspace sampling algorithm selects $c = \mathcal{O}(k\epsilon^{-2} \log k \log(1/\delta))$ columns and $r = \mathcal{O}\big(c\epsilon^{-2} \log c \log(1/\delta)\big)$ rows without replacement. Then

$$\|\mathbf{A} - \mathbf{CUR}\|_F \;=\; \big\|\mathbf{A} - \mathbf{CW}^{+}\mathbf{R}\big\|_F \;\leq\; (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F,$$

holds with probability at least $1 - \delta$, where $\mathbf{W}$ contains the rows of $\mathbf{C}$ with scaling. The running time is dominated by the truncated SVD of $\mathbf{A}$, that is, $\mathcal{O}(mnk)$.

The Adaptive Sampling CUR Algorithm

Wang and Zhang (2013) proposed an *Adaptive Sampling CUR Algorithm*.

- Select $c = \frac{2k}{\epsilon}(1 + o(1))$ columns of $\mathbf{A}$ to construct $\mathbf{C} \in \mathbb{R}^{m \times c}$ using Algorithm of Boutsidis et al. (2013);
- Select $r_1 = c$ rows of $\mathbf{A}$ to construct $\mathbf{R}_1 \in \mathbb{R}^{r_1 \times n}$ using Algorithm of Boutsidis et al. (2013);
- Adaptively sample $r_2 = c/\epsilon$ rows from $\mathbf{A}$ according to the residual $\mathbf{A} - \mathbf{A}\mathbf{R}_1^{\dagger}\mathbf{R}_1$;
- Return $\mathbf{C}$, $\mathbf{R} = [\mathbf{R}_1^T, \mathbf{R}_2^T]^T$, and $\mathbf{U} = \mathbf{C}^{\dagger}\mathbf{A}\mathbf{R}^{\dagger}$.

The Adaptive Sampling CUR Algorithm

Lemma (Wang and Zhang, 2013)

Given a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ *and a matrix* $\mathbf{C} \in \mathbb{R}^{m \times c}$ *such that* $\mathrm{rank}(\mathbf{C}) = \mathrm{rank}(\mathbf{C}\mathbf{C}^{\dagger}\mathbf{A}) = \rho$ $(\rho \leq c \leq n)$, *let* $\mathbf{R}_1 \in \mathbb{R}^{r_1 \times n}$ *consist of* r_1 *rows of* $\mathbf{A}$ *and define the residual* $\mathbf{B} = \mathbf{A} - \mathbf{A}\mathbf{R}_1^{\dagger}\mathbf{R}_1$. *Additionally, for* $i = 1, \cdots, m$, *we define*

$$\pi_i \;=\; \|\mathbf{b}^{(i)}\|_2^2 / \|\mathbf{B}\|_F^2.$$

We further sample r_2 *rows i.i.d. from* $\mathbf{A}$, *in each trial of which the* i*-th row is chosen with probability* p_i. *Let* $\mathbf{R}_2 \in \mathbb{R}^{r_2 \times n}$ *contain the* r_2 *sampled rows and let* $\mathbf{R} = [\mathbf{R}_1^T, \mathbf{R}_2^T]^T \in \mathbb{R}^{(r_1+r_2) \times n}$. *Then we have*

$$\mathbb{E}\|\mathbf{A} - \mathbf{C}\mathbf{C}^{\dagger}\mathbf{A}\mathbf{R}^{\dagger}\mathbf{R}\|_F^2 \;\leq\; \|\mathbf{A} - \mathbf{C}\mathbf{C}^{\dagger}\mathbf{A}\|_F^2 + \frac{\rho}{r_2}\|\mathbf{A} - \mathbf{A}\mathbf{R}_1^{\dagger}\mathbf{R}_1\|_F^2,$$

where the expectation is taken w.r.t. $\mathbf{R}_2$.

The Adaptive Sampling CUR Algorithm

Theorem (Wang and Zhang, 2013)

Given a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ *and a positive integer* $k \ll \min\{m, n\}$, *the Adaptive Sampling CUR algorithm randomly selects* $c = \frac{2k}{\epsilon}(1+o(1))$ *columns of* $\mathbf{A}$ *to construct* $\mathbf{C} \in \mathbb{R}^{m \times c}$, *and then selects* $r = \frac{c}{\epsilon}(1+\epsilon)$ *rows of* $\mathbf{A}$ *to construct* $\mathbf{R} \in \mathbb{R}^{r \times n}$. *Then we have*

$$\mathbb{E}\|\mathbf{A} - \mathbf{CUR}\|_F \; = \; \mathbb{E}\|\mathbf{A} - \mathbf{C}(\mathbf{C}^\dagger \mathbf{A} \mathbf{R}^\dagger)\mathbf{R}\|_F \; \leq \; (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F.$$

The algorithm costs time $\mathcal{O}\big((m+n)k^3\epsilon^{-2/3} + mk^2\epsilon^{-2} + nk^2\epsilon^{-4}\big) + T_{Multiply}\big(mnk\epsilon^{-1}\big)$ *to compute matrices* $\mathbf{C}$, $\mathbf{U}$ *and* $\mathbf{R}$.

Optimal CUR Algorithm

Boutsidis and Woodruff (2014) proposed *Optimal CUR Algorithm*.

- Construction **C** with $O(k + \frac{k}{\epsilon})$ columns:
  - Compute the top k singular vectors of **A**: $\mathbf{Z}_1$
  - Sample $O(k \log k)$ columns from $\mathbf{Z}_1^T$ with the leverage scores
  - Down-sample columns to $c_1 = O(k)$ columns with the sampling algorithm of Boutsidis et al. (2013)
  - Adaptively sample $c_2 = O(\frac{k}{\epsilon})$ columns of **A**
- Construction **R** with $O(k + \frac{k}{\epsilon})$ rows:
  - Find $\mathbf{Z}_2$ in the span of **C** such that: $\|\mathbf{A} - \mathbf{Z}_2\mathbf{Z}_2^T\mathbf{A}\|_F^2 \leq (1+\epsilon) \cdot \|\mathbf{A} - \mathbf{A}_k\|_F^2$
  - Sample $O(k \log k)$ rows from $\mathbf{Z}_2$ with the leverage scores
  - Down-sample rows to $r_1 = O(k)$ rows with the sampling algorithm of Boutsidis et al. (2013)
  - Sample $r_2 = O(\frac{k}{\epsilon})$ rows with adaptive sampling

Optimal CUR Algorithm

Lemma (Boutsidis and Woodruff, 2014)

Given a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{V} \in \mathbb{R}^{m \times c}$ *and an integer* k, *let* $\mathbf{V} = \mathbf{Y}\Psi$ *be a QR decomposition of* $\mathbf{V}$, $\Gamma = \mathbf{Y}^T\mathbf{A}$, $\Gamma_k = \Delta\Sigma_k\mathbf{V}_k^T$ *be a rank* k *SVD of* Γ, $\Delta \in \mathbb{R}^{c \times k}$. *Then* $\mathbf{Y}\Delta\Delta^T\mathbf{Y}^T$ *satisfies:*

$$\|\mathbf{A} - \mathbf{Y}\Delta\Delta^T\mathbf{Y}^T\mathbf{A}\|_F^2 \leq \|\mathbf{A} - \mathbf{Y}\Delta\Sigma_k\mathbf{V}_k^T\|_F^2 = \|\mathbf{A} - \Pi_{V,k}^F(\mathbf{A})\|_F^2.$$

Optimal CUR Algorithm

Theorem (Boutsidis and Woodruff, 2014)

Given a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ *of rank* ρ*, a target rank* $1 \leq k \leq \rho$*, and* $0 < \epsilon < 1$*, the optimal CUR algorithm selects at most* $c = O(k/\epsilon)$ *columns and at most* $r = O(k/\epsilon)$ *rows from* $\mathbf{A}$ *form matrices* $\mathbf{C} \in \mathbb{R}^{m \times c}$*,* $\mathbf{R} \in \mathbb{R}^{r \times n}$*, and* $\mathbf{U} \in \mathbb{R}^{c \times r}$ *with* $\mathrm{rank}(\mathbf{U}) = k$ *such that, with some probability,*

$$\|\mathbf{A} - \mathbf{CUR}\|_F^2 \leq \|(1 + O(\epsilon))\|\mathbf{A} - \mathbf{A}_k\|_F^2.$$

The matrices $\mathbf{C}$*,* $\mathbf{U}$*, and* $\mathbf{R}$ *can be computed in time*

$$\mathcal{O}\left[\mathrm{nnz}(\mathbf{A}) \log n + (m+n) \times \mathrm{poly}(\log n, k, 1/\epsilon)\right].$$

The Nyström Method

Random Projection
The Johnson and Lindenstrauss Lemma
Randomized SVD
Subspace Embedding
Random Selection
Column Selection
CUR Decomposition
The Nyström Method
References

- **Random Selection**:
  selects c ($\ll n$) columns of $\mathbf{K}$ to construct $\mathbf{C}$ using some randomized algorithms. After permutation we have

$$\mathbf{K} = \begin{bmatrix} \mathbf{W} & \mathbf{K}_{21}^T \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix}, \qquad \mathbf{C} = \begin{bmatrix} \mathbf{W} \\ \mathbf{K}_{21} \end{bmatrix}.$$

- **The Nyström Approximation**: $\tilde{\mathbf{K}}_c^{\mathrm{nys}} \approx \mathbf{K}$

$$\underbrace{\tilde{\mathbf{K}}_c^{\mathrm{nys}}}_{n\times n} = \underbrace{\mathbf{C}}_{n\times c} \underbrace{\mathbf{W}^\dagger}_{c\times c} \underbrace{\mathbf{C}^T}_{c\times n}.$$

The Nyström Method

- **Random Selection**:
  selects c ($\ll n$) columns of $\mathbf{K}$ to construct $\mathbf{C}$ using some randomized algorithms. After permutation we have

$$\mathbf{K} = \begin{bmatrix} \mathbf{W} & \mathbf{K}_{21}^T \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix}, \qquad \mathbf{C} = \begin{bmatrix} \mathbf{W} \\ \mathbf{K}_{21} \end{bmatrix}.$$

- **The Nyström Approximation**: $\tilde{\mathbf{K}}_c^{\mathrm{nys}} \approx \mathbf{K}$

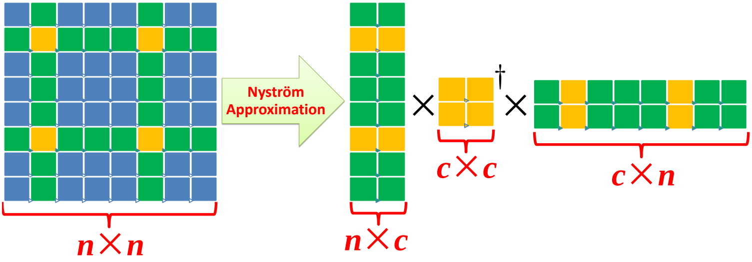
$$\underbrace{\tilde{\mathbf{K}}_c^{\mathrm{nys}}}_{n\times n} = \underbrace{\mathbf{C}}_{n\times c} \underbrace{\mathbf{W}^\dagger}_{c\times c} \underbrace{\mathbf{C}^T}_{c\times n}.$$

The Nyström Approximation

- **The Nyström Approximation**:

$$\mathbf{K} \quad \approx \quad \tilde{\mathbf{K}}_c^{\text{nys}} \quad = \quad \mathbf{C}\mathbf{W}^{\dagger}\mathbf{C}^T$$

(A low-rank factorization).



Problem Formulation

Problem:

- How to select informative columns of $\mathbf{K} \in \mathbb{R}^{n \times n}$ to construct $\mathbf{C} \in \mathbb{R}^{n \times c}$?
- The approximation error $\left\|\mathbf{K} - \mathbf{C}\mathbf{U}\mathbf{C}^T\right\|_F$ or $\left\|\mathbf{K} - \mathbf{C}\mathbf{U}\mathbf{C}^T\right\|_2$ should be as small as possible.

Criterion: Upper Error Bounds

- Using approximation algorithms to find c *good* columns (not necessarily the *best*)
- Hope that $\frac{\|\mathbf{K}-\mathbf{C}\mathbf{U}\mathbf{C}^T\|_F}{\|\mathbf{K}-\mathbf{K}_k\|_F}$ has upper bound, which is the smaller the better.

Uniform Sampling: The Simplest Algorithm

- Sample c columns of $\mathbf{K}$ uniformly at random to construct $\mathbf{C}$.
- The simplest, but the most widely used.

Adaptive Sampling

The adaptive sampling algorithm [Deshpande *et al.* , 2006]:

1. Sample c_1 columns of $\mathbf{K}$ to construct $\mathbf{C}_1$ using some algorithm;
2. Compute the residual $\mathbf{B} = \mathbf{K} - \mathbf{C}_1\mathbf{C}_1^{\dagger}\mathbf{K}$;
3. Compute sampling probabilities $p_i = \frac{\|\mathbf{b}_i\|_2^2}{\|\mathbf{B}\|_F^2}$, for $i = 1$ to n;
4. Sample further c_2 columns of $\mathbf{K}$ in c_2 i.i.d. trials, in each trial the i-th column is chosen with probability p_i; Denote the selected columns by $\mathbf{C}_2$;
5. Return $\mathbf{C} = [\mathbf{C}_1 \, , \, \mathbf{C}_2]$.

Adaptive Sampling

- The error term $\|\mathbf{K} - \mathbf{C}\mathbf{C}^{\dagger}\mathbf{K}\|_F$ is bounded theoretically, but $\|\mathbf{K} - \mathbf{C}\mathbf{W}^{\dagger}\mathbf{C}^{T}\|_F$ is not.
- Empirically, the adaptive sampling algorithm works very well.

Better Sampling Algorithms?

- We hope $\frac{\|\mathbf{K}-\mathbf{C}\mathbf{W}^{\dagger}\mathbf{C}^T\|_F}{\|\mathbf{K}-\mathbf{K}_k\|_F}$ will be very small if the column sampling algorithm is good enough.
- But it cannot be arbitrarily small.
- Lower Error Bound

Theorem (Wang & Zhang, JMLR 2013)

Whatever column sampling is used to select c columns, there exists a bad case $\mathbf{K}$ *such that*

$$\frac{\|\mathbf{K}-\mathbf{C}\mathbf{W}^{\dagger}\mathbf{C}^T\|_F^2}{\|\mathbf{K}-\mathbf{K}_k\|_F^2} \geq \Omega\left(1+\frac{nk}{c^2}\right).$$

Different Types of Low-Rank Approximation?

- The *Ensemble Nyström Method* [Kumar *et al.*, JMLR 2012]:

$$\mathbf{K} \approx \sum_{i=1}^{t} \frac{1}{t} \mathbf{C}^{(i)} \mathbf{W}^{(i)\dagger} \mathbf{C}^{(i)T}$$

- It does not improve the lower error bound.
- Lower Error Bound

Theorem (Wang & Zhang, JMLR 2013)

Whatever column sampling is used to select c columns, there exists a bad case $\mathbf{K}$ *such that*

$$\frac{\left\|\mathbf{K} - \sum_{i=1}^{t} \frac{1}{t} \mathbf{C}^{(i)} \mathbf{W}^{(i)\dagger} \mathbf{C}^{(i)T}\right\|_F^2}{\|\mathbf{K} - \mathbf{K}_k\|_F^2} \geq \Omega\left(1 + \frac{nk}{c^2}\right).$$

The Modified Nyström Method

- The *Modified Nyström Method* [Wang & Zhang, JMLR 2013]:

$$\mathbf{K} \approx \mathbf{C}\Big(\underbrace{\mathbf{C}^{\dagger}\mathbf{K}(\mathbf{C}^{\dagger})^{T}}_{c\times c}\Big)\mathbf{C}^{T}.$$

Theorem (Wang & Zhang, JMLR 2013)

Using a column sampling algorithm, the error incurred by the modified Nyström method satisfies

$$\mathbb{E}\frac{\|\mathbf{K} - \mathbf{C}(\mathbf{C}^{\dagger}\mathbf{K}(\mathbf{C}^{\dagger})^{T})\mathbf{C}^{T}\|_F^2}{\|\mathbf{K} - \mathbf{K}_k\|_F^2} \leq 1 + \sqrt{\frac{k}{c}}.$$

Comparisons between the Two Methods

- **The Standard Nyström Method: fast.**
  It costs only $T_{\mathrm{SVD}}(c^3)$ time to compute the intersection matrix $\mathbf{U}^{\mathrm{nys}} = \mathbf{W}^{\dagger}$.

- **The Modified Nyström Method: slow.**
  It costs $T_{\mathrm{SVD}}(nc^2) + T_{\mathrm{Multiply}}(n^2c)$ time to compute the intersection matrix $\mathbf{U}^{\mathrm{mod}} = \mathbf{C}^{\dagger}\mathbf{K}(\mathbf{C}^{\dagger})^T$ naively.

Comparisons between the Two Methods

- **The Standard Nyström Method: inaccurate.**
  It cannot attain $1 + \epsilon$ Frobenius relative-error bound unless

$$c \geq \sqrt{nk/\epsilon}$$

  columns are selected, whatever column selection algorithm is used. (Due to its lower error bound.)

- **The Modified Nyström Method: accurate.**
  Some adaptive sampling based algorithms attain $1 + \epsilon$ Frobenius relative-error bound when

$$c = \mathcal{O}(k/\epsilon^2).$$

  (c is the smaller the better.)

Comparisons between the Two Methods

Theorem (Exact Recovery)

For the symmetric matrix $\mathbf{K}$ *defined previously, the following three statements are equivalent:*

1. $\mathrm{rank}(\mathbf{W}) = \mathrm{rank}(\mathbf{K})$,
2. $\mathbf{K} = \mathbf{C}\mathbf{W}^{\dagger}\mathbf{C}^{T}$,
   (i.e., the standard Nyström method is exact)
3. $\mathbf{K} = \mathbf{C}\big(\mathbf{C}^{\dagger}\mathbf{K}(\mathbf{C}^{\dagger})^{T}\big)\mathbf{C}^{T}$,
   (i.e., the modified Nyström method is exact)

Outline

1 Random Projection
- The Johnson and Lindenstrauss Lemma
- Randomized SVD

2 Subspace Embedding

3 Random Selection
- Column Selection
- CUR Decomposition
- The Nyström Method

4 References

References

Santosh S. Vempala. *The Random Projection Method*. American Mathematical Society, 2000.

Michael W. Mahoney. *Randomized Algorithms for Matrices and Data*. Foundations and Trends in Machine Learning, 3(2): 123-224, 2011.

N. Halko, P. G. Martinsson, and J. A. Tropp. *Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions*. SIAM Review, 53(2): 217-288, 2011

W. B. Johnson and J. Lindenstrauss. *Extensions of Lipschitz mapping into a Hilbert space*. Contemporary Mathematics, 1984.

S. Dasgupta and A. Gupta. *An elementary proof of a theorem of Johnson and Lindenstrauss*. Random Structure & Algorithms, 2003.

J. Matoušek. *On variants of the Johnson and Lindenstrauss Leamma*. Random Structure & Algorithms, 2008.

References

- A. Dasgupta, R. Kumar, and T. Sarlós: *A sparse Johnson-Lindenstrauss Transform*. In STOC, 2010.
- K. L. Clarkson and D. P. Woodruff: *Low Rank Approximation and Regression in Sparsity Time*. In STOC, 2013.
- X. Meng and M. W. Mahoney. *Low-distortion subspace embeddings in input-sparsity time and applications to robust linear regression.* STOC, 2013.
- J. Nelson and H. L. Nguyên. *OSNAP: Faster numerical linear algebra algorithms via sparser subspace embeddings* In FOCS, 2013.
- J. Batson, D. Spielman, and N. Srivastave: *Twice-Ramanujan Sparsifiers*. SIAM Review, 2014.

References

A. Frieze, K. Kannan, and Rademacher, S. Vempala: *Fast Monte-Carlo algorithms for finding low-rank approximation*. In FOCS, 1998. Journal of the ACM, 2004.

A. Deshpande, L. Rademacher, S. Vempala, and G.Wang: *Matrix approximation and projective clustering via volume sampling.* Theory of Computing, 2006.

C. Boutsidis, P. Drineas, and M. Magdon-Ismail: *Near optimal column-based matrix reconstruction*. SIAM Journal on Computing, 2013.

V. Guruswami and A. K. Sinop: *Optimal column based low-rank matrix reconstruction*. In SODA, 2012.

References

P. Drineas, M. W. Mahoney, and S. Muthukrishnan. *Relative-error CUR matrix decompositions*. SIAM Journal on Matrix Analysis and Applications, 2008.

M. W. Mahoney and P. Drineas. *CUR matrix decompositions for improved data analysis*. Proceedings of the National Academy of Sciences, 2009.

Sshuse Wang and Zhihua Zhang: *Improving CUR matrix decomposition and the Nyström approximation via adaptive sampling*. JMLR, 2013.

C. Boutsidis and D. P. Woodruff: *Optimal CUR matrix decompositions*. In STOC, 2014.

S. Kumar, M. Mohri, and A. Talwalkar: *Sampling methods for the Nyström method*. JMLR, 2012.

K. L. Clarkson and D. P. Woodruff: *Low Rank Approximation and Regression in Sparsity Time*. In STOC, 2013.