Lecture 3: Low-rank matrix approximation

School of Mathematical Sciences, Xiamen University

1. Low-rank matrix approximation problem

- Suppose $\mathbf{B}$ is a rectangular matrix, accessible via matvecs $\mathbf{x} \mapsto \mathbf{Bx}$ and $\mathbf{y} \mapsto \mathbf{B}^*\mathbf{y}$. The task is to produce a low-rank approximation of $\mathbf{B}$ that is competitive with a best approximation of similar rank.
- The best rank-k approximation is unique if and only if $\sigma_k > \sigma_{k+1}$:

$$\min_{\operatorname{rank}(\mathbf{M})\le k} \|\mathbf{B} - \mathbf{M}\|_{\mathrm{F}}^2 = \|\mathbf{B} - \mathbf{U}_k\mathbf{U}_k^*\mathbf{B}\|_{\mathrm{F}}^2 = \sum\nolimits_{i>k} \sigma_i^2,$$

  where $\mathbf{U}_k$ is the matrix consisting of the leading k left singular vectors. Cost: $\mathcal{O}(mnp)$ where $p = \min(m, n)$.
- Let $s = k + l$ for a small natural number l. For a tolerance $\varepsilon > 0$, we seek a rank-s approximation $\widehat{\mathbf{B}}_s$ that competes with the best rank-k approximation:

$$\|\mathbf{B} - \widehat{\mathbf{B}}_s\|_{\mathrm{F}}^2 \le (1+\varepsilon)\|\mathbf{B} - \mathbf{U}_k\mathbf{U}_k^*\mathbf{B}\|_{\mathrm{F}}^2 = (1+\varepsilon)\sum\nolimits_{i>k} \sigma_i^2.$$

1.1 Randomized SVD: Intuition

- Draw a standard normal test vector $\mathbf{w} \in \mathbb{R}^n$. we have

$$\mathbf{B}\mathbf{w} = \sum_{i=1}^{p} \sigma_i \mathbf{u}_i (\mathbf{v}_i^* \mathbf{w}) := \sum_{i=1}^{p} \sigma_i \mathbf{u}_i w_i.$$

  The component $w_i := \mathbf{v}_i^* \mathbf{w}$ of the random vector along the ith right singular vector follows a standard normal distribution, and the components $(w_i, i = 1, \ldots, p)$ compose an independent family.

- On average, $\mathbb{E}(w_i^2) = 1$. Therefore, the image $\mathbf{B}\mathbf{w}$ tends to align with the left singular vectors associated with large singular values.
- By repeating this process with a statistically independent family $(\mathbf{w}^{(j)} : j = 1, \ldots, s)$ of random test vectors, we can obtain a family $(\mathbf{B}\mathbf{w}^{(j)} : j = 1, \ldots, s)$ of vectors whose span contains most of range$(\mathbf{U}_k)$. The number $s = k + l$ of test vectors needs to be a bit larger than the target rank k to obtain coverage of the subspace with high probability.

1.2 Randomized SVD: Algorithm (cost $\mathcal{O}(smn)$)

- For a rank parameter s, we draw a random test matrix:

$$\mathbf{\Omega} = \begin{bmatrix} \mathbf{w}^{(1)} & \cdots & \mathbf{w}^{(s)} \end{bmatrix} \quad \text{where} \quad \mathbf{w}^{(j)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n) \quad \text{i.i.d.}$$

- We obtain $\mathbf{Y} := \mathbf{B}\mathbf{\Omega}$. The orthogonal projector $\mathbf{P}_{\mathbf{Y}}$ onto range$(\mathbf{Y})$ serves as a proxy for the ideal projector $\mathbf{U}_k\mathbf{U}_k^*$. Computationally,

$$\mathbf{P}_{\mathbf{Y}} := \mathbf{Q}\mathbf{Q}^* \quad \text{where} \quad \mathbf{Q} := \text{orth}(\mathbf{Y}).$$

  The function orth returns an orthonormal basis and costs $\mathcal{O}(s^2m)$.

- Finally, we report the approximation $\widehat{\mathbf{B}}_s$ in factored form:

$$\widehat{\mathbf{B}}_s := \mathbf{P}_{\mathbf{Y}}\mathbf{B} = \mathbf{Q}(\mathbf{Q}^*\mathbf{B}).$$

- If desired, we can report the SVD of the approximation after a small amount of additional work ($\mathcal{O}(s^2n)$):

$$\widehat{\mathbf{B}}_s = (\mathbf{Q}\widehat{\mathbf{U}}_0)\widehat{\mathbf{\Sigma}}\widehat{\mathbf{V}}^* \quad \text{where} \quad (\widehat{\mathbf{U}}_0, \widehat{\mathbf{\Sigma}}, \widehat{\mathbf{V}}) = \text{svd}(\mathbf{Q}^*\mathbf{B}).$$

Algorithm: Randomized SVD.

```
$\mathbf{\Omega} = \mathrm{randn}(n, s)$
$\mathbf{Y} = \mathbf{B}\mathbf{\Omega}$
$\mathbf{Q} = \mathrm{orth}(\mathbf{Y})$
$\mathbf{C} = \mathbf{Q}^*\mathbf{B}$
$(\widehat{\mathbf{U}}_0, \widehat{\mathbf{\Sigma}}, \widehat{\mathbf{V}}) = \mathrm{svd}(\mathbf{C})$
$\widehat{\mathbf{U}} = \mathbf{Q}\widehat{\mathbf{U}}_0$
```

Theorem 1

Consider a matrix $\mathbf{B} \in \mathbb{R}^{m\times n}$, and fix the target rank $k \le p$. When $s \ge k+2$, the randomized SVD method produces a random rank-s approximation $\widehat{\mathbf{B}}_s$ that satisfies

$$\mathbb{E}\|\mathbf{B} - \widehat{\mathbf{B}}_s\|_{\mathrm{F}}^2 \le \left(1 + \frac{k}{s-k-1}\right)\sum_{i>k} \sigma_i^2(\mathbf{B}).$$

Proof. See A. Kireeva and J. A. Tropp, arXiv:2402.17873, 2024. □

1.3 Randomized subspace iteration

Algorithm: Randomized subspace iteration.

```
$\mathbf{X}_0 = \mathrm{randn}(n, s)$
for $t = 1, 2, \ldots, T$
    $\mathbf{Q}_t := \mathrm{orth}(\mathbf{B}\mathbf{X}_{t-1})$
    $\mathbf{X}_t := \mathbf{B}^*\mathbf{Q}_t$
end
$\widehat{\mathbf{B}}_s := \mathbf{Q}_T\mathbf{X}_T^*$
```

- Randomized SVD is the special case of this algorithm with $T = 1$.
- Randomized subspace iteration produces approximations

$$\widehat{\mathbf{B}}_s = \mathbf{Q}_t(\mathbf{Q}_t^*\mathbf{B}) \ \text{ where } \ \mathbf{Q}_t = \mathrm{orth}((\mathbf{B}\mathbf{B}^*)^{t-1}\mathbf{B}\boldsymbol{\Omega}) \ \text{ for } \ t = 1, 2, \ldots$$

- Much as the block power method drives its iterates toward the leading eigenspace, subspace iteration drives range($\mathbf{Q}_t$) so that it aligns with range($\mathbf{U}_r$), the leading left singular subspace of $\mathbf{B}$.

2. Low-rank spsd approximation from entries

Consider an spsd matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ that we access via entry evaluations: $(j,k) \mapsto a_{jk}$. The task is to produce a low-rank spsd approximation of $\mathbf{A}$ using as few as entry evaluations.

2.1 Column Nyström approximation

- Given a list $S \subseteq \{1,2,3,...,n\}$ of column indices, the column Nyström approximation:

$$\mathbf{A}_{\langle S\rangle} := \mathbf{A}(:,S)\mathbf{A}(S,S)^{\dagger}\mathbf{A}(S,:).$$

- The column Nyström approximation has several remarkable properties: (1) $\mathrm{range}(\mathbf{A}_{\langle S\rangle}) = \mathrm{range}(\mathbf{A}(:,S))$; (2) $\mathbf{0} \preccurlyeq \mathbf{A}_{\langle S\rangle} \preccurlyeq \mathbf{A}$.
- Our goal is to find a set S of s columns that make the error

$$\|\mathbf{A} - \mathbf{A}_{\langle S\rangle}\|_{*} = \mathrm{tr}(\mathbf{A} - \mathbf{A}_{\langle S\rangle})$$

as small as possible.

2.2 Pivoted partial Cholesky

- Set $\mathbf{A}_0 := \mathbf{A}$ and $\widehat{\mathbf{A}}_0 := \mathbf{0}$.

  At each step $t = 1, 2, \ldots, s$, select $i_t \in \{1, 2, \ldots, n\}$, and update

$$\mathbf{A}_t := \mathbf{A}_{t-1} - \frac{\mathbf{A}_{t-1}(:, i_t)\mathbf{A}_{t-1}(i_t, :)}{\mathbf{A}_{t-1}(i_t, i_t)};$$

$$\widehat{\mathbf{A}}_t := \widehat{\mathbf{A}}_{t-1} + \frac{\mathbf{A}_{t-1}(:, i_t)\mathbf{A}_{t-1}(i_t, :)}{\mathbf{A}_{t-1}(i_t, i_t)}.$$

  Exercise: Prove the following results: (i) $\mathbf{A}_t + \widehat{\mathbf{A}}_t = \mathbf{A}$;

$$\text{(ii) } \mathrm{diag}(\mathbf{A}_t) = \mathrm{diag}(\mathbf{A}_{t-1}) - \frac{1}{\mathbf{A}_{t-1}(i_t, i_t)} |\mathbf{A}_{t-1}(:, i_t)|^2.$$

Proposition 2

Suppose that we apply the pivoted partial Cholesky algorithm to an spsd matrix $\mathbf{A}$, and we select i_t from S in any order. Then $\widehat{\mathbf{A}}_{|S|} = \mathbf{A}_{\langle S \rangle}$, where $|S|$ denotes the number of elements of the set S.

The proof is left as an exercise.

2.3 Pivoted partial Cholesky: evaluating fewer entries

- Set $\mathbf{F}_0 := \mathbf{0}$.

  At each step $t = 1, 2, \ldots, s$, select $i_t \in \{1, 2, \ldots, n\}$ and set

  $$\mathbf{c}_t := \mathbf{A}(:, i_t) - \mathbf{F}_{t-1}(\mathbf{F}_{t-1}(i_t, :))^*.$$

  Update $\mathbf{F}_t := \begin{bmatrix} \mathbf{F}_{t-1} & \mathbf{c}_t/\sqrt{\mathbf{c}_t(i_t)} \end{bmatrix}$.

  Exercise: Prove that $\widehat{\mathbf{A}}_t = \mathbf{F}_t\mathbf{F}_t^*$ for $t = 0, 1, 2, \ldots, s$.

2.4 Pivot selection rules

- Uniform random pivoting: $i_t \sim \text{uniform}\{1, 2, \ldots, n\}$.

  Assumption: data points represent an i.i.d. sample from a population, so one is just as good as another.

- Greedy pivoting: $i_t \in \text{argmax}\{\mathbf{A}_{t-1}(i, i) : i = 1, 2, \ldots, n\}$.

  Note that $\mathbf{A}_t(i_t, i_t) = 0$.

- Importance sampling pivoting: $\mathbb{P}\{i_t = j\} = \mathbf{A}_{t-1}(j, j)/\text{tr}(\mathbf{A}_{t-1})$.

  Balance between uniform random and greedy.

2.5 Randomly pivoted partial Cholesky

Algorithm: Randomly pivoted partial Cholesky.

```
$\mathbf{F} = \mathrm{zeros}(n, s)$                                   (Preallocation)
$\mathbf{d} = \mathrm{diag}(\mathbf{A})$
for $t = 1, 2, \ldots, s$
    Sample $i_t \sim \mathbf{d} / \sum_{j=1}^{n} \mathbf{d}(j)$
    $\mathbf{c} = \mathbf{A}(:, i_t) - \mathbf{F}(\mathbf{F}(i_t, :))^*$
    $\mathbf{F}(:, t) = \mathbf{c} / \sqrt{\mathbf{c}(i_t)}$
    $\mathbf{d} = \mathbf{d} - |\mathbf{F}(:, t)|^2$
    $\mathbf{d} = \max\{\mathbf{d}, \mathbf{0}\}$              (Improve numerical stability)
    Stop when $\sum_{j=1}^{n} \mathbf{d}(j) < \eta \cdot \mathrm{tr}(\mathbf{A})$               (Optional)
end
```

- To produce a rank-s approximation, the algorithm only requires $(s+1)n - s$ entries of $\mathbf{A}$: its diagonal and the s pivot columns.

- Define the expected residual map: $\Phi(\mathbf{A}) := \mathbb{E}(\mathbf{A}_1)$. This function measures the average progress that we make after one step of the algorithm. A quick calculation yields a formula for the expected residual map:

$$\begin{aligned}\Phi(\mathbf{A}) &= \sum_{j=1}^{n} \left[\mathbf{A} - \frac{\mathbf{A}(:,j)\mathbf{A}(j,:)}{\mathbf{A}(j,j)}\right] \frac{\mathbf{A}(j,j)}{\mathrm{tr}(\mathbf{A})} \\ &= \mathbf{A} - \frac{1}{\mathrm{tr}(\mathbf{A})} \sum_{j=1}^{n} \mathbf{A}(:,j)\mathbf{A}(j,:) = \mathbf{A} - \frac{\mathbf{A}^2}{\mathrm{tr}(\mathbf{A})}.\end{aligned}$$

As a result, we have

$$\mathbb{E}(\mathrm{tr}(\mathbf{A}_1)) = \mathrm{tr}(\mathbb{E}(\mathbf{A}_1)) = \left(1 - \frac{\mathrm{tr}(\mathbf{A}^2)}{(\mathrm{tr}(\mathbf{A}))^2}\right) \mathrm{tr}(\mathbf{A}) \le \frac{n-1}{n}\mathrm{tr}(\mathbf{A}).$$

In each iteration, we decrease the expected trace of the residual on average.

- The best rank-k approximation in $\|\cdot\|_*$:

$$\min_{\operatorname{rank}(\mathbf{M})\leq k} \|\mathbf{A}-\mathbf{M}\|_* = \sum_{j>k} \sigma_j(\mathbf{A}).$$

- Fix a comparison rank k and a tolerance $\varepsilon > 0$. Randomly pivoted Cholesky produces an approximation $\widehat{\mathbf{A}}_s$ that attains the error bound

$$\mathbb{E}(\|\mathbf{A}-\widehat{\mathbf{A}}_s\|_\star) \leq (1+\varepsilon)\sum_{j>k}\sigma_j(\mathbf{A})$$

  after selecting s columns where

$$s \geq \frac{k}{\varepsilon} + k\log\left(\frac{1}{\varepsilon\eta}\right) \quad \text{and} \quad \eta := \frac{1}{\operatorname{tr}(\mathbf{A})}\sum_{j>k}\sigma_j(\mathbf{A}).$$