Topics in Randomized Numerical Linear Algebra

Alex Gittens

Applied and Computational Mathematics
California Institute of Technology
gittens@caltech.edu

May 31, 2013

Randomized Numerical Linear Algebra

We consider the performance of two classes of randomized low-rank approximants:

- *Projection-based schemes.* We approximate $\mathbf{A}$ by projecting it onto a random subspace of its range: $\mathbf{M} = \mathbf{P}_{\mathbf{AS}}\mathbf{A}$
- *SPSD sketches.* Here $\mathbf{A}$ is symmetric positive-semidefinite and we take $\mathbf{M} = \mathbf{C}\mathbf{W}^{\dagger}\mathbf{C}^T$, where

  $$\mathbf{C} = \mathbf{AS} \quad \text{and} \quad \mathbf{W} = \mathbf{S}^T\mathbf{AS}$$

  and $\mathbf{W}^{\dagger}$ denotes the Moore-Penrose pseudoinverse of $\mathbf{W}$.

Optimal rank-k approximation

Fix $\mathbf{A} \in \mathbb{R}^{m \times n}$ with $m \geq n$.
The most accurate rank-k approximation in the spectral and Frobenius norms

$$\mathbf{A}_k = \underset{\substack{\mathbf{M} \in \mathbb{R}^{m \times n} \\ \operatorname{rank}(\tilde{\mathbf{A}}) \leq k}}{\operatorname{argmin}} \|\mathbf{A} - \mathbf{M}\|_\xi, \quad \text{for } \xi \in \{2, \mathrm{F}\}.$$

can be computed in $\mathrm{O}(mn^2)$ time via the Singular Value Decomposition (SVD): if

$$\mathbf{A} = \mathbf{U\Sigma V}^T = \begin{array}{c} \begin{array}{cc} k & n-k \end{array} \\ \begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{bmatrix} \end{array} \begin{array}{c} \begin{array}{cc} k & n-k \end{array} \\ \begin{bmatrix} \mathbf{\Sigma}_1 & \\ & \mathbf{\Sigma}_2 \end{bmatrix} \end{array} \begin{bmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \end{bmatrix}$$

then $\mathbf{A}_k = \mathbf{U}_1 \mathbf{\Sigma}_1 \mathbf{V}_1^T$.

We measure the approximation accuracies relative to those of $\mathbf{A}_k$.

- An approximation $\mathbf{M}$ satisfies a *relative-error bound* if

$$\|\mathbf{A} - \mathbf{M}\|_\xi \leq (1 + \epsilon)\|\mathbf{A} - \mathbf{A}_k\|_\xi$$

  for an $\epsilon > 0$.

- It satisfies an *additive-error bound* if

$$\|\mathbf{A} - \mathbf{M}\|_\xi \leq \|\mathbf{A} - \mathbf{A}_k\|_\xi + \epsilon$$

  for an $\epsilon > 0$. In this case, ϵ is called the *additional error*.

Goal: Find relative- and additive-error bounds for randomized low-rank approximation schemes.

The target audience

Who is interested in these approximations, and what do they want?:

- The *numerical linear algebra community* wants high quality approximations with very low failure rates and low communication cost.
- The *machine learning community* wants approximations whose errors are on par with modeling inaccuracies and the imprecision of the data
- The *optimization community* is interested in varying levels of quality.
- The *theoretical computer science community* is interested in understanding the behavior of these algorithms, e.g. what is the optimal tradeoff between the error, failure rate, and the amount of arithmetic operations involved? How can communication cost be minimized?

Direct and iterative low-rank approximation methods

Classical *direct* methods (e.g. SVD or rank-revealing QR factorizations) are not appropriate for large matrices:

- Require forming an expensive ($\mathrm{O}(mn^2)$ time) factorization of $\mathbf{A}$ first.
- Can densify intermediate matrices, so do not take advantage of sparsity.

Classical *iterative* methods are more appropriate:

- They do not factorize $\mathbf{A}$.
- They build low-rank approximations iteratively, and check for convergence
- They can take advantage of sparsity.
- They focus on recovering the singular spaces of the matrix.

Iterative method 1: Subspace iteration

The top k-dimensional left invariant subspace of $\mathbf{A}$ is the top k-dimensional eigenspace of $\mathbf{A}\mathbf{A}^T$. Subspace iteration captures this space by repeatedly applying $\mathbf{A}\mathbf{A}^T$ to a random set of vectors.

$$\mathcal{R}((\mathbf{A}\mathbf{A}^T)^q\mathbf{S}) \to \mathcal{R}(\mathbf{U}_1)$$

<Anim rotation of space as B applied>

```
1: Let $\mathbf{Y} = \mathbf{A}\mathbf{S}$.
2: Compute the QR decomposition $\mathbf{Y} = \mathbf{Q}\mathbf{R}$.
3: Compute the SVD of $\mathbf{Q}^T\mathbf{A} = \mathbf{W}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^T$.
4: Set $\tilde{\mathbf{U}} = \mathbf{Q}\mathbf{W}$.
```

Iterative method 2: Lanczos iteration

Krylov subspace (Lanczos) methodology: capture top singular space, then use to approx singular values have guarantees on quality of singular space as function of mult. eigengap and iterations. In practice, test for convergence, have numerical issues requiring complicated restarting/deflating etc. processes, efficiency depends on properties of the matrix (e.g. degenerate singular values cause issues) Separate issues of singular space from sing values

Projection-based low-rank approximations

<Illustrate> Capture top singular space of matrix, then project onto it
Unlike the classical algorithms, the goal is to d

Input: an $m \times n$ matrix $\mathbf{A}$ and an $n \times \ell$ matrix $\mathbf{S}$, where ℓ is an integer in $[1, n]$.

Output: matrices $\tilde{\mathbf{U}}, \tilde{\mathbf{\Sigma}}, \tilde{\mathbf{V}}$ constituting the SVD of $\mathbf{P}_{\mathbf{AS}}\mathbf{A} = \tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^T$.

```
Let $\mathbf{Y} = \mathbf{AS}$.
Compute the QR decomposition $\mathbf{Y} = \mathbf{QR}$.
Compute the SVD of $\mathbf{Q}^T\mathbf{A} = \mathbf{W}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^T$.
Set $\tilde{\mathbf{U}} = \mathbf{QW}$.
```

Comparison with classical iterative methods

Comparison with classical methods: how it's like Lanczos/Krylov-space w/o many iterations, or orthog iteration running time comparison (same-ish, but Lanczos complicated implementation and number of iterations variable on matrix), communication cost comparison (better, 2ish passes) importance of error bounds to guiding understanding (b/c can't check error iteratively)

History: , origi by Papadimitriou, popularized by Sarlos, Wolfe et al. use randomized projections

Importance of oversampling, spectral decay, power method, type of randomness used

The advantage of fast transformations

Advantage of subsampled fast orthogonal transforms when not using power method. E.g. SRHT (define) introduce SRFT etc. Wolfe et al. realized fast transformation known results

Goal: to reduce the amount of oversampling needed to get good bounds Known results vs our results for SRHT (when one better than other) mention did work with Christos

Structural result at the basis of our result Idea: take advantage of spectral decay in the tangent main ideas of proof: spread out energy evenly across columns of matrix, connection w/ spectral norm preserve singular values via concentration argument preserve spectral norm matrix multiplication result outline spectral norm result

outline Frobenius norm result

problem of preserving positivity in addition to getting low-rank approximation could use above projection-based schemes alternative schemes which eliminate projection step (give spsd approx scheme)

How the two compare: stable, low-error w/ two passes unstable in general, low-error w/ one-pass still higher than other when take as many passes as stable, even lower error <Illust>

Note sketching allows for both column-sampling and mixture-based schemes

In particular, SPSD sketch scheme incorporates Nyström extensions, which use no information on matrix other than size, requires only column sampling <Illust>

introduce more particular examples of sketching schemes: gaussian, levscore, SRFT explain leverage scores (better measure of nonuniformity) explain coherence (cruder)

Previous results on SPSD sketching (non-adaptive!): Nystrom, CD11 give table our results asymptotically better comment not many other SPSD sketching schemes considered give table w/ error information argue all perform well in practice

Approach to analyzing SPSD sketch performance: First deterministic results Plug in stochastic for tangent

Key point: realize the factorization of CWC in terms of $A^{p-1/2}$ Note can replace $A^{1/2}$ with generalized Cholesky factorization Also explains why SPSD sketches with same number of iterations better than projection based approxs derive spectral bound – can relate directly to earlier deterministic bound derive Frobenius bound – need to do some more estimations derive trace norm bound – simple

Given established framework, note which quantities need bounding For leverage-based scheme, blah blah For gaussian-based scheme, blah blah state leverage based result

state gaussian based result compare to leverage-based state SRFT-based result compare to gaussian/lev

To get Nyström result, need bound blah, observe can be written as sum of rank one mats, use matrix Chernoff bound state and explain matrix Chernoff bound

recall coherence state Nyström result point out dependence on μ

illustrate dependence on μ and sparsity of U1 illustrate optimality dimensional dependence in rel err bound for Nystrom prove optimality of rel err bound for Nystrom introduce stabilization algorithms: one use Tikhonov regularization, one truncates W one which I thought was due to me but actually due to WS01, one due to CD11

Illust errors of regularized SPSD sketches argue that CD11 better in terms of norm reconstruction, but keep in mind down stream apps, is more violent, so might make sense to look at WS01 alg

State result in CD11 for truncated W which applies only to Nyström

State theorem for WS01 algorithm, and about condition numbers (what is a reasonable condition number for solving linear systems? this would motivate choice of regularization parameter, and say whether this bound is useful)

Combine with some particular SPSD sketch to get guarantees for the regularized version