Fast and Space Efficient NLA, Compressed Sensing

Lecture 24

Dec 1, 2020

Some topics today

We have seen fast "approximation" algorithms for matrix multiplication

- random sampling
- Using JL

Today:

- Subspace embeddings for faster linear least squares and low-rank approximation
- Frequent directions algorithms for one/two pass approximate SVD
- Compressed Sensing

Subspace Embedding

Question: Suppose we have linear subspace E of $\mathbb{R}^n$ of dimension d. Can we find a projection $\Pi : \mathbb{R}^d \to \mathbb{R}^k$ such that for *every* $x \in E$, $\|\Pi x\|_2 = (1 \pm \epsilon)\|x\|_2$?

- Not possible if $k < d$.
- Possible if $k = \ell$. Pick Π to be an orthonormal basis for E.
  Disadvantage: This requires knowing E and computing orthonormal basis which is slow.

What we really want: *Oblivious* subspace embedding ala JL based on random projections

Oblivious Supspace Embedding

Theorem

Suppose E *is a linear subspace of* $\mathbb{R}^n$ *of dimension* d*. Let* Π *be a DJL matrix* $\Pi \in \mathbb{R}^{k \times d}$ *with* $k = O(\frac{d}{\epsilon^2} \log(1/\delta))$ *rows. Then with probability* $(1 - \delta)$ *for every* $x \in E$,

$$\|\frac{1}{\sqrt{k}}\Pi x\|_2 = (1 \pm \epsilon)\|x\|_2.$$

In other words JL Lemma extends from one dimension to arbitrary number of dimensions in a graceful way.

Part I

Faster algorithms via subspace embeddings

Linear least squares/Regression

Linear least squares: Given $A \in \mathbb{R}^{n \times d}$ and $b \in \mathbb{R}^d$ find x to minimize $\|Ax - b\|_2$.

Interesting when $n \gg d$ the over constrained case when there is no solution to $Ax = b$ and want to find best fit.

Geometrically Ax is a linear combination of columns of A. Hence we are asking what is the vector z in the column space of A that is closest to vector b in ℓ_2 norm.

Closest vector to b is the projection of b into the column space of A so it is "obvious" geometrically. How do we find it?

Linear least squares/Regression

Linear least squares: Given $A \in \mathbb{R}^{n \times d}$ and $b \in \mathbb{R}^d$ find x to minimize $\|Ax - b\|_2$.

Interesting when $n \gg d$ the over constrained case when there is no solution to $Ax = b$ and want to find best fit.

Geometrically Ax is a linear combination of columns of A. Hence we are asking what is the vector z in the column space of A that is closest to vector b in ℓ_2 norm.

Closest vector to b is the projection of b into the column space of A so it is "obvious" geometrically. How do we find it? Find an orthonormal basis $z_1, z_2, \ldots, z_r$ for the columns of A. Compute projection c as $c = \sum_{j=1}^{r} \langle b, z_j \rangle z_j$ and output answer as $\|b - c\|_2$.

Linear least squares via Subspace embeddings

Let $a_1, a_2, \ldots, a_d$ be the columns of A and let E be the subspace spanned by $\{a_1, a_2, \ldots, a_d, b\}$

E has dimension at most $d + 1$.

Use subspace embedding on E. Applying JL matrix Π with $k = O(\frac{d}{\epsilon^2})$ rows we reduce $a_1, a_2, \ldots, a_d, b$ to $a'_1, a'_2, \ldots, a'_d, b'$ which are vectors in $\mathbb{R}^k$.

Solve $\min_{x' \in \mathbb{R}^d} \|A'x' - b'\|_2$

Low-rank approximation

Recall: Given $A \in \mathbb{R}^{n \times d}$ and integer k want to find best rank matrix B to minimize $\|A - B\|_F$

- SVD gives optimum for all k. If $A = UDV^T = \sum_{i=1}^{d} \sigma_i u_i v_i^T$ then $A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T$ is optimum for every k.
- $\|A - A_k\|_F^2 = \sum_{i>k} \sigma_i^2$.
- $v_1, v_2, \ldots, v_k$ are k orthogonal unit vectors from $\mathbb{R}^d$ and maximize the sum of squares of the projection of the **rows** of A onto the space spanned by them
- $u_1, u_2, \ldots, u_k$ are k orthogonal unit vectors from $\mathbb{R}^n$ that maximize the sum of squares of the projections of the **columns** of A onto the space spanned

Low-rank approximation via subspace embeddings

Column view of SVD: $u_1, u_2, \ldots, u_k$ are k orthogonal unit vectors from $\mathbb{R}^n$ that maximize the sum of squares of the projections of the columns of A onto the space spanned

Let $a_1, a_2, \ldots, a_d$ be the columns of A and let E be subspace spanned by them. $\dim(E) \leq d$ obviously.

Wlog $u_1, u_2, \ldots, u_k \in E$. Why?

Low-rank approximation via subspace embeddings

Column view of SVD: $u_1, u_2, \ldots, u_k$ are k orthogonal unit vectors from $\mathbb{R}^n$ that maximize the sum of squares of the projections of the columns of A onto the space spanned

Let $a_1, a_2, \ldots, a_d$ be the columns of A and let E be subspace spanned by them. $\dim(E) \leq d$ obviously.

Wlog $u_1, u_2, \ldots, u_k \in E$. Why?
If $u_1, u_2, \ldots, u_k$ fixed then $v_1, v_2, \ldots, v_k$ are determined. Why?

Low-rank approximation via subspace embeddings

Column view of SVD: $u_1, u_2, \ldots, u_k$ are k orthogonal unit vectors from $\mathbb{R}^n$ that maximize the sum of squares of the projections of the columns of A onto the space spanned

Let $a_1, a_2, \ldots, a_d$ be the columns of A and let E be subspace spanned by them. $\dim(E) \leq d$ obviously.

Wlog $u_1, u_2, \ldots, u_k \in E$. Why?
If $u_1, u_2, \ldots, u_k$ fixed then $v_1, v_2, \ldots, v_k$ are determined. Why?

Low-rank approximation via subspace embeddings

Column view of SVD: $u_1, u_2, \ldots, u_k$ are k orthogonal unit vectors from $\mathbb{R}^n$ that maximize the sum of squares of the projections of the columns of A onto the space spanned

Let $a_1, a_2, \ldots, a_d$ be the columns of A and let E be subspace spanned by them. $\dim(E) \leq d$ obviously.

Wlog $u_1, u_2, \ldots, u_k \in E$. Why?
If $u_1, u_2, \ldots, u_k$ fixed then $v_1, v_2, \ldots, v_k$ are determined. Why?
Let Π be an ϵ-approximate subspace preserving embedding for E

Claim: $\|(\Pi A) - (\Pi A)_k\|_F \leq (1 + \epsilon)\|A - A_k\|_F$

Analysis

Claim: $\|(\Pi A) - (\Pi A)_k\|_F \leq (1+\epsilon)\|A - A_k\|_F$

Analysis

Claim: $\|(\Pi A) - (\Pi A)_k\|_F \leq (1+\epsilon)\|A - A_k\|_F$

Proof sketch: Let $a'_1, \ldots, a'_d$ be columns of ΠA and let $u'_1, \ldots, u'_k$ be $\Pi u_1, \ldots, \Pi u_k$.

Analysis

Claim: $\|(\Pi A) - (\Pi A)_k\|_F \leq (1+\epsilon)\|A - A_k\|_F$

Proof sketch: Let $a'_1, \ldots, a'_d$ be columns of ΠA and let $u'_1, \ldots, u'_k$ be $\Pi u_1, \ldots, \Pi u_k$.

$\|A - A_k\|_F^2 = \sum_{i=1}^{d} \|a_i - \sum_{j=1}^{k} v_j(i) u_j\|_2^2$

Analysis

Claim: $\|(\Pi A) - (\Pi A)_k\|_F \leq (1+\epsilon)\|A - A_k\|_F$

Proof sketch: Let $a'_1, \ldots, a'_d$ be columns of ΠA and let $u'_1, \ldots, u'_k$ be $\Pi u_1, \ldots, \Pi u_k$.

$\|A - A_k\|_F^2 = \sum_{i=1}^d \|a_i - \sum_{j=1}^k v_j(i) u_j\|_2^2$

From subspace embedding property of Π,
$\|\Pi(a_i - \sum_{j=1}^k v_j(i) u_j)\|_2 \leq (1+\epsilon)\|a_i - \sum_{j=1}^k v_j(i) u_j\|_2$

Since $u'_1, u'_2, \ldots, u'_k$ is a feasible solution for k-rank approximation to ΠA.

Analysis

Claim: $\|(\Pi A) - (\Pi A)_k\|_F \leq (1+\epsilon)\|A - A_k\|_F$

Proof sketch: Let $a'_1, \ldots, a'_d$ be columns of ΠA and let $u'_1, \ldots, u'_k$ be $\Pi u_1, \ldots, \Pi u_k$.

$\|A - A_k\|_F^2 = \sum_{i=1}^d \|a_i - \sum_{j=1}^k v_j(i) u_j\|_2^2$

From subspace embedding property of Π,
$\|\Pi(a_i - \sum_{j=1}^k v_j(i) u_j)\|_2 \leq (1+\epsilon)\|a_i - \sum_{j=1}^k v_j(i) u_j\|_2$

Since $u'_1, u'_2, \ldots, u'_k$ is a feasible solution for k-rank approximation to ΠA.

Claim: $\|(\Pi A) - (\Pi A)_k\|_F \geq (1-\epsilon)\|A - A_k\|_F$.

Analysis

Claim: $\|(\Pi A) - (\Pi A)_k\|_F \leq (1+\epsilon)\|A - A_k\|_F$

Proof sketch: Let $a'_1, \ldots, a'_d$ be columns of ΠA and let $u'_1, \ldots, u'_k$ be $\Pi u_1, \ldots, \Pi u_k$.

$\|A - A_k\|_F^2 = \sum_{i=1}^{d} \|a_i - \sum_{j=1}^{k} v_j(i) u_j\|_2^2$

From subspace embedding property of Π,
$\|\Pi(a_i - \sum_{j=1}^{k} v_j(i) u_j)\|_2 \leq (1+\epsilon)\|a_i - \sum_{j=1}^{k} v_j(i) u_j\|_2$

Since $u'_1, u'_2, \ldots, u'_k$ is a feasible solution for k-rank approximation to ΠA.

Claim: $\|(\Pi A) - (\Pi A)_k\|_F \geq (1-\epsilon)\|A - A_k\|_F$. Prove it!

Running Time

- A has d columns in $\mathbb{R}^n$ and ΠA has d columns in $\mathbb{R}^k$ where $k = O(\frac{d}{\epsilon^2} \ln(1/\delta))$. Hence dimensionality reduction from n to k and one can run SVD on ΠA.
- ΠA can be computed fast in time roughly proportional to $nnz(A)$ (number of non-zeroes of A).

Part II

Frequent Directions Algorithm

Low-rank approximation

Faster low-rank approximation algorithms based on randomized algorithm: sampling and subspace embeddings

- Can we find a deterministic algorithm?
- Streaming algorithm?

Low-rank approximation and SVD

Given matrix $A \in \mathbb{R}^{n \times d}$ and (small) integer k

Row view of SVD: $v_1, v_2, \ldots, v_k$ are k orthogonal unit vectors from $\mathbb{R}^d$ that maximize the sum of squares of the projections of the rows A onto the space spanned

Let $a_1, a_2, \ldots, a_n$ be the rows of A (treated as vectors in $\mathbb{R}^d$)

$\sigma_j^2 = \sum_{i=1}^{n} \langle a_i, v_j \rangle^2$ and $\|A - A_k\|_F^2 = \sum_{j>k} \sigma_j^2$

Low-rank approximation and SVD

Given matrix $A \in \mathbb{R}^{n \times d}$ and (small) integer k

Row view of SVD: $v_1, v_2, \ldots, v_k$ are k orthogonal unit vectors from $\mathbb{R}^d$ that maximize the sum of squares of the projections of the rows A onto the space spanned

Let $a_1, a_2, \ldots, a_n$ be the rows of A (treated as vectors in $\mathbb{R}^d$)

$\sigma_j^2 = \sum_{i=1}^n \langle a_i, v_j \rangle^2$ and $\|A - A_k\|_F^2 = \sum_{j>k} \sigma_j^2$

Consider matrix $D_k V_k^T$ whose rows are $\sigma_1 v_1, \sigma_2 v_2, \ldots, \sigma_k v_k$.
$\|D_k V_k^T\|_F^2 = \sum_{j=1}^k \sigma_j^2 = \|A_k\|_F^2$

Frequent Directions Algorithm

[Liberty] and analyzed for relative error guarantee by [Ghashami-Phillips]
Liberty inspired by Misra-Greis frequent items algorithm.

Rows of A come one by one

Algorithm maintains a matrix $Q \in \mathbb{R}^{\ell \times d}$ where $\ell = k(1 + 1/\epsilon)$.
Hence memory is $O(kd/\epsilon)$

At end of algorithm let Q_k be best rank k-approximation for Q.
Then $\|A - \text{Proj}_{Q_k}(A)\|_F \leq (1 + \epsilon)\|A - A_k\|_F$.

Thus a $(1 + \epsilon)$-approximate k-dimensional subspace for rows of A be identified by storing $O(k/\epsilon)$ rows.

FD Algorithm

Frequent-Directions

```
Initialize Q^0 as an all zeroes ℓ × d matrix
For each row a_i ∈ A do
    Set Q_+ ← Q^{i−1} with last row replaced by a_i
    Compute SVD of Q_+ as UDV^T
    C^i = DV^T (for analysis)
    δ_i = σ_ℓ^2 (for analysis)
    D' = diag(sqrt(σ_1^2 − δ_i), sqrt(σ_2^2 − δ_i), ..., sqrt(σ_{ℓ−1}^2 − δ_i), 0)
    Q^i = D'V^T
EndFor
Return Q = Q^n
```

If $\ell = \lceil k(1 + 1/\epsilon) \rceil$ and Q_k is the rank k approximation to output Q then

$$\|A - \mathrm{Proj}_{Q_k}(A)\|_F \leq (1 + \epsilon)\|A - A_k\|_F$$

Running time

- One pass algorithm but requires second pass to compute actual singular values etc
- Space $O(kd/\epsilon)$
- Run time: n computations of SVD on $k/\epsilon \times d$ matrix. Can be improved (see home work problem).

Interesting even when $k = 1$. Alternative to power method to find top singular value/vector. Deterministic.

Part III

Compressed Sensing

Sparse recovery

Recall:

- Vector $x \in \mathbb{R}^n$ and integer k
- x updated in streaming setting one coordinate at a time (can be positive or negative changes)
- Want to find best k-sparse vector $\tilde{x}$ that approximates x. $\min_{y, \|y\|_0 \leq k} \|y - x\|_2$. Optimum solution is clear: take y to be the largest k coordinates of x in absolute value.
- Using Count-Sketch: $O(\frac{k}{\epsilon^2}\text{polylog}(n))$ space one can find k-sparse z such that $\|z - x\|_2 \leq (1 + \epsilon)\|y^* - x\|_2$ with high probability.
- Count-Sketch can be seen as Πx for some $\Pi \in \mathbb{R}^{m \times n}$ where $m = O(\frac{k}{\epsilon^2}\text{polylog}(n))$.

Compressed Sensing

Compressed sensing: we want to create projection matrix Π such that for *any* x we can create from Πx a good k-sparse approximation to x

Doable! With Π that has $O(k \log(n/k))$ rows. Creating Π requires randomization but once found it can be used. Called RIP matrices. First due to Candes, Romberg, Tao and Donoho. Lot of work in signal processing and algorithms.

Compressed Sensing

Theorem (Candes-Romberg-Tao, Donoho)

For every n, k *there is a matrix* $\Pi \in \mathbb{R}^{m \times n}$ *with* $m = O(k \log(n/k))$ *and a polytime algorithm such that for any* $x \in \mathbb{R}^n$, *the algorithm given* Πx *outputs a* k*-sparse vector* $\tilde{x}$ *such that* $\|\tilde{x} - x\|_2 \leq O(\frac{1}{\sqrt{k}})\|x_{tail(k)}\|_1$. *In particular it recovers* x *exactly if it is* k*-sparse.*

Matrix that satisfies above property are called RIP matrices (restricted isometry property)

Closely connected to JL matrices

Understanding RIP matrices

Suppose x, x' are two distinct k-sparse vectors in $\mathbb{R}^n$

Basic requirement: $\Pi x \neq \Pi x'$ otherwise cannot recover exactly

Let $S, S' \subset [n]$ be the indices in the support of x, x' respectively. Πx is in the span of columns of Π_S and $\Pi x'$ is in the span of columns of $\Pi_{S'}$

Thus we need columns of $\Pi_{S \cup S'}$ to be linearly independent for any S, S' with $S \neq S'$ and $|S| \leq k$ and $|S'| \leq k$. Any $2k$ columns of Π should be linearly independent.

Understanding RIP matrices

Suppose x, x' are two distinct k-sparse vectors in $\mathbb{R}^n$

Basic requirement: $\Pi x \neq \Pi x'$ otherwise cannot recover exactly

Let $S, S' \subset [n]$ be the indices in the support of x, x' respectively. Πx is in the span of columns of Π_S and $\Pi x'$ is in the span of columns of $\Pi_{S'}$

Thus we need columns of $\Pi_{S \cup S'}$ to be linearly independent for any S, S' with $S \neq S'$ and $|S| \leq k$ and $|S'| \leq k$. Any $2k$ columns of Π should be linearly independent.

Sufficient information theoretically. Computationally?

Recovery

Suppose we have Π such that any $2k$ columns are linearly independent.

Suppose x is k-sparse and we have Πx. How do we recover x?

Solve the following:

$$\min \|z\|_0 \quad \text{such that} \quad \Pi z = \Pi x$$

Recovery

Suppose we have $\mathbf{\Pi}$ such that any $2k$ columns are linearly independent.

Suppose x is k-sparse and we have $\mathbf{\Pi}x$. How do we recover x?

Solve the following:

$$\min \|z\|_0 \quad \text{such that} \quad \mathbf{\Pi}z = \mathbf{\Pi}x$$

Guaranteed to recover x by uniqueness but NP-Hard!

Recovery

Instead of solving

$$\min \|z\|_0 \quad \text{such that} \quad \Pi z = \Pi x$$

solve

$$\min \|z\|_1 \quad \text{such that} \quad \Pi z = \Pi x$$

which is a linear/convex programming problem and hence can be solved in polynomial-time.

If Π satisfies additional properties then one can show that above recovers x.

RIP Property

Definition

A $m \times n$ matrix Π has the (ϵ, k)-RIP property if for every k-sparse $x \in \mathbb{R}^n$,

$$(1-\epsilon)\|x\|_2^2 \leq \|\Pi x\|_2^2 \leq (1+\epsilon)\|x\|_2^2$$

.

Equivalent, whenever $|S| \leq k$ we have

$$\|\Pi_S^T \Pi_S - I_k\|_2 \leq \epsilon$$

which is equivalent to saying that if σ_1 and σ_k are the largest and smallest singular value of Π_S then $\frac{\sigma_1^2}{\sigma_k^2} \leq (1+\epsilon)$

Every k columns of Π are approximately orthonormal.

Recovery theorem

Suppose Π is $(\epsilon, 2k)$-RIP with $\epsilon < \sqrt{2} - 1$ and let $\tilde{x}$ be optimum solution to the following LP

$$\min \|z\|_1 \quad \text{such that} \quad \Pi z = \Pi x$$

Then $\|\tilde{x} - x\|_2 \leq O(\frac{1}{\sqrt{k}}) \|x_{\text{tail}(k)}\|_1$.

Called ℓ_2/ℓ_1 guarantee. Proof is somewhat similar to the one for sparse recovery with Count-Sketch.

More efficient "combinatorial" algorithms that avoid solving LP.

RIP matrices and subspace embeddings

Definition

A $m \times n$ matrix Π has the (ϵ, k)-RIP property if for every k-sparse $x \in \mathbb{R}^n$,

$$(1 - \epsilon)\|x\|_2^2 \leq \|\Pi x\|_2^2 \leq (1 + \epsilon)\|x\|_2^2$$

.

Fix $S \subset [n]$ with $|S| = k$. S defines a subspace of k-sparse vectors.

Total of $\binom{n}{k}$ different subspaces. Want to preserve the length of vectors in all of these subspaces.

Fix $S \subset [n]$ with $|S| = k$. S defines a subspace of k-sparse vectors. Total of $\binom{n}{k}$ different subspaces. Want to preserve the length of vectors in all of these subspaces.

Given a subspace W of dimension d we saw that if Π is JL matrix with $m = O(d/\epsilon^2)$ rows we have the property that for every $x \in W$: $\|\Pi x\|_2^2 \simeq (1 \pm \epsilon)\|x\|_2^2$. Via a net argument where net size is $e^{O(k)}$.

If we want to preserve $\binom{n}{k}$ different subspaces need to preserve nets of all subspaces

Hence via union bound we get $m = O(\frac{1}{\epsilon^2} \log(e^{O(k)} \binom{n}{k}))$ which is $O(\frac{k}{\epsilon^2} \log n)$.

Fix $S \subset [n]$ with $|S| = k$. S defines a subspace of k-sparse vectors. Total of $\binom{n}{k}$ different subspaces. Want to preserve the length of vectors in all of these subspaces.

Given a subspace W of dimension d we saw that if Π is JL matrix with $m = O(d/\epsilon^2)$ rows we have the property that for every $x \in W$: $\|\Pi x\|_2^2 \simeq (1 \pm \epsilon)\|x\|_2^2$. Via a net argument where net size is $e^{O(k)}$.

If we want to preserve $\binom{n}{k}$ different subspaces need to preserve nets of all subspaces

Hence via union bound we get $m = O(\frac{1}{\epsilon^2} \log(e^{O(k)} \binom{n}{k}))$ which is $O(\frac{k}{\epsilon^2} \log n)$.

Other techniques give $m = O(k^2/\epsilon^2)$.