Sketching for M-Estimators and Robust Numerical Linear Algebra

David Woodruff
CMU
Talk Outline

• Regression
  – Sketching for least squares regression
  – Sketching for fast robust regression

• Low Rank Approximation
  – Sketching for fast SVD
  – Sketching for fast robust low rank approximation

• Recent sketching/sampling work for robust problems
Linear Regression

Matrix form

Input: $n \times d$-matrix $A$ and a vector $b=(b_1, \ldots, b_n)$
$n$ is the number of examples; $d$ is the number of unknowns

Output: $x^*$ so that $Ax^*$ and $b$ are close

- Consider the over-constrained case, when $n \gg d$
Least Squares Regression

• Find $x^*$ that minimizes $|Ax-b|^2$

• $Ax^*$ is the projection of $b$ onto the column span of $A$

• Desirable statistical properties

• Closed form solution: $x^* = (A^TA)^{-1} A^T b$
Sketching to Solve Least Squares Regression

- How to find an approximate solution \( x \) to \( \min_x |Ax-b|_2^2 \)?

- **Goal:** output \( x' \) for which \( |Ax'-b|_2 \leq (1+\varepsilon) \min_x |Ax-b|_2 \) with high probability

- Draw \( S \) from a \( k \times n \) random family of matrices, for a value \( k \ll n \)

- Compute \( S^*A \) and \( S^*b \)

- Output the solution \( x' \) to \( \min_{x'} |(SA)x-(Sb)|_2 \)
How to Choose the Right Sketching Matrix?

- Recall: output the solution $x'$ to $\min_x |(SA)x-(Sb)|_2$
- Lots of matrices work
- $S$ is $d/\varepsilon^2 \times n$ matrix of i.i.d. Normal random variables
- Computing $SA$ may be slow…
- Can speed up to $O(nd \log n)$ time using Fast Johnson Lindenstrauss transforms [Sarlos]
  - Not sensitive to input sparsity
Faster Sketching Matrices [CW]

- CountSketch matrix

- Define $k \times n$ matrix $S$, for $k = O(d^2/\epsilon^2)$

- $S$ is really sparse: single randomly chosen non-zero entry per column

\[
\begin{bmatrix}
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 1 & 0 & -1 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\]

$S \ast A$ computable in $\text{nnz}(A)$ time (See also [MM,MP,NN])

Think of rows as hash buckets
Simple Proof [ANW]

- Replace A with [A, b], and then show $|SAX|_2 = (1 \pm \epsilon) |Ax|_2$ for all x
  - Can assume columns of A are orthonormal
  - Can assume x is a unit vector

- SA is a $6d^2/(\delta \epsilon^2) \times d$ matrix

- Suffices to show $|A^T S^T SA - I|_2 \leq |A^T S^T SA - I|_F \% \epsilon$

- Approximate matrix product for all matrices C and D
  \[ \Pr[|C S^T S D - CD|_F^2 \leq [6/(\delta (# rows of S))] \times |C|_F^2 |D|_F^2] \geq 1 - \delta \]

- Set C = $A^T$ and D = A

- Then $|A|_F^2 = d$ and (# rows of S) = $6d^2/(\delta \epsilon^2)$
Talk Outline

• Regression
  – Sketching for least squares regression
  – Sketching for fast robust regression

• Low Rank Approximation
  – Sketching for fast SVD
  – Sketching for fast robust low rank approximation

• Recent sketching/sampling work for robust problems
Other Fitness Measures

Example: Method of least absolute deviation ($l_1$ -regression)

- Find $x^*$ that minimizes $|Ax-b|_1 = \sum |b_i - \langle A_i^*, x \rangle|$

- Cost is less sensitive to outliers than least squares

- Can solve via linear programming

What about the many other fitness measures used in practice?
M-Estimators

• "Measure" function
  - $M: \mathbb{R} \rightarrow \mathbb{R}_{\geq 0}$
  - $M(x) = M(-x)$, $M(0) = 0$
  - $M$ is non-decreasing in $|x|$

• $|y|_M = \sum_{i=1}^{n} M(y_i)$

• Solve $\min_{x} |Ax-b|_M$

• Least squares and $L_1$-regression are special cases
Huber Loss Function

\[ M(x) = \frac{x^2}{2c} \text{ for } |x| \leq c \]

\[ M(x) = |x| - c/2 \text{ for } |x| > c \]

Enjoys smoothness properties of \( l_2 \) and robustness properties of \( l_1 \)
Other Examples

- $L_1$-$L_2$
  \[ M(x) = 2\left((1+x^2/2)^{1/2} - 1\right) \]

- Fair estimator
  \[ M(x) = c^2 \left[ |x|/c - \log(1+|x|/c) \right] \]

- Tukey estimator
  \[
  M(x) = \begin{cases} 
    c^2/6 \left(1-(x/c)^2\right)^3 & \text{if } |x| \leq c \\
    c^2/6 & \text{if } |x| > c 
  \end{cases}
  \]

Nice M-Estimators

- An M-Estimator is **nice** if it has at least linear growth and at most quadratic growth.

- There is $C_M > 0$ so that for all $a, a'$ with $|a| \geq |a'| > 0$,
  \[
  |a/a'|^2 \leq M(a)/M(a') \leq C_M |a/a'|
  \]

- Any **convex** $M$ satisfies the linear lower bound.

- Any **sketchable** $M$ satisfies the quadratic upper bound.
  - sketchable $\Rightarrow$ there is a distribution on $k \times n$ matrices $S$ for which $|Sx|_M = \mathcal{S}(|x|_M)$ with good probability and $k$ is slow-growing function of $n$. 
Nice M-Estimator Theorem

[Nice M-Estimators] $O(\text{nnz}(A)) + T(\text{poly}(d \log n))$ time algorithm for nice M to output $x'$ so that for any constant $C > 1$, with probability 99%:

$$|Ax' - b|_M \lesssim C \min_x |Ax - b|_M$$

Remarks:

- $T(\text{poly}(d \log n))$ is time to solve a weighted $\text{poly}(d \log n)$-sized version of M-regression
- For convex nice M-estimators can solve with convex programming, but slow – poly(nd) time
- Theorem also applies to non-convex M
- Our sketch is “universal”
- Can get $(1+\epsilon)$-approximation via sampling techniques
The same M-Sketch works for all nice M-estimators!

- many analyses of this data structure don’t work since they reduce the problem to a non-convex problem

\[ x' = \arg\min_x |TAx - Tb|_{w,M} \]

- Sketch used for estimating frequency moments [Indyk, W] and earthmover distance [Verbin, Zhang]

• \( S^i \) are independent CountSketch matrices with poly(d) rows

• \( D^i \) is \( n \times n \) diagonal and uniformly samples a \( \frac{1}{d \log n} \) fraction of the \( n \) rows
M-Sketch Intuition

• For a given $y = Ax - b$, consider $|Ty|_{w, M} = \Sigma_i w_i M((Ty)_i)$

• [Contraction] $|Ty|_{w, M} \geq 0.9 |y|_M$ with probability $1 - \exp(-d \log n)$

• [Dilation] $|Ty|_{w, M} \leq 1.1 |y|_M$ with probability 99%

• Contraction allows for a net argument (no scale-invariance!)

• Dilation implies the optimal $y^*$ does not dilate much

• Proof: try to estimate contribution to $|y|_M$ at all scales
  – E.g., if $y = (n, 1, 1, ..., 1)$ with a total of $n-1$ 1s, then $|y|_1 = n + (n-1)*1$
  – When estimating a given scale, use the fact that smaller stuff cancels each other out in a bucket and gives its 2-norm
Talk Outline

• Regression
  – Sketching for least squares regression
  – Sketching for fast robust regression

• Low Rank Approximation
  – Sketching for fast SVD
  – Sketching for fast robust low rank approximation

• Recent sketching/sampling work for robust problems
Low Rank Approximation

- $A$ is an $n \times d$ matrix
  - Think of $n$ points in $\mathbb{R}^d$

- **Goal:** find a low rank matrix approximating $A$
  - Easy to store, data more interpretable

- $A_k = \text{argmin}_{\text{rank } k \text{ matrices } B} \|A - B\|_F$ can be found via the SVD

- Computing $A_k$ exactly is expensive
Approximate Low Rank Approximation

- **Goal**: output a rank $k$ matrix $A'$, so that
  \[ |A-A'|_F \leq (1+\epsilon) |A-A_k|_F \]

- Can do this in $\text{nnz}(A) + (n+d)\cdot\text{poly}(k/\epsilon)$ time [CW]
Solution to Low-Rank Approximation [S]

- Given $n \times d$ input matrix $A$
- Compute $S \cdot A$ using a sketching matrix $S$ with $k/\varepsilon << n$ rows. $S \cdot A$ takes random linear combinations of rows of $A$

- Project rows of $A$ onto $SA$, then find best rank-$k$ approximation to points inside of $SA$. 
What is the Matrix S?

• S can be a $k/\varepsilon \times n$ matrix of i.i.d. normal random variables

• [S] S can be an $O\sim(k/\varepsilon) \times n$ Fast Johnson Lindenstrauss Matrix

• [CW] S can be a poly$(k/\varepsilon) \times n$ CountSketch matrix
Caveat: Projecting the Points onto SA is Slow

- Current algorithm:
  1. Compute $S^*A$
  2. Project each of the rows onto $S^*A$
  3. Find best rank-$k$ approximation of projected points inside of rowspace of $S^*A$

- Bottleneck is step 2

- [CW] Approximate the projection
  - Fast algorithm for approximate constrained regression
  \[
  \min_{\text{rank}-k} \|X(SA) - A\|_F^2
  \]
  - $\text{nnz}(A) + (n+d)\text{poly}(k/\epsilon)$ time
Talk Outline

• Regression
  – Sketching for least squares regression
  – Sketching for fast robust regression

• Low Rank Approximation
  – Sketching for fast SVD
  – Sketching for fast robust low rank approximation

• Recent sketching/sampling work for robust problems
Robust Low Rank Approximation

• Given n x d matrix A, think of its rows as points $a_1, a_2, ..., a_n$ in $\mathbb{R}^d$

• (Rotational invariance) if you rotate $\mathbb{R}^d$ by rotation $W$, obtaining points $a_1 W, a_2 W, ..., a_n W$, cost is preserved

• Cost function studied in [DZHZ06, SV07, DV07, FL11, VX12]:

$$\min_{k-\text{dim } V} \sum_i d(a_i, V)^p$$

• For $p$ in [1,2), cost function is more robust than the SVD ($p = 2$)
Prior Work on this Cost Function

• A k-dimensional space $V'$ is a $(1+\epsilon)$-approximation if
  \[
  \sum_{i} d(a_i, V')^p \leq (1 + \epsilon) \min_{k\text{-dim } V} \sum_{i} d(a_i, V)^p
  \]

• For constant $1 \leq p < \infty$,
  • $(1+\epsilon)$-approximation in $n \cdot d \cdot \text{poly}(k/\epsilon) + \exp(\text{poly}(k/\epsilon))$ time [SV07]
  • (Weak Coreset) $\text{poly}(k/\epsilon)$-dimensional space $V'$ containing a k-dim space $V''$ which is a $(1+\epsilon)$-approximation in $n \cdot d \cdot \text{poly}(k/\epsilon)$ time [DV07, FL11]

• For $p > 2$,
  • NP-hard to approximate up to a constant factor $\gamma_p$ [DTV10, GRSW12].
  • there is a $\text{poly}(nd)$ time $\sqrt{2\gamma_p}$-approximation algorithm [DTV10]
Questions from Prior Work

1. **(Exponential Term)** Is $\exp(\text{poly}(k/\epsilon))$ time for $1 \leq p < 2$ necessary?

2. **(Input Sparsity)** Can one achieve a leading order term in the time complexity of $\text{nnz}(A)$, as in the case of $p = 2$?

3. **(M-Estimators)** What about algorithms for M-estimator loss functions:

$$\min_{k-\text{dim } V} \sum_{i} M(d(a_i, V))$$
Results for Robust Low Rank Approximation [CW]

• **(Hardness)** For $p$ in $[1,2)$ it’s NP-hard to get a $(1+1/d)$-approximation
  • Since $p > 2$ is also hard, there is a “singularity” at $p = 2$

• **(Input Sparsity Time Algorithm)** For $p$ in $[1,2)$ we get an algorithm in time
  $$\text{nnz}(A) + (n+d)\text{poly}(k/\epsilon) + \exp(\text{poly}(k/\epsilon))$$

• **(Weak Coreset)** Get $\text{nnz}(A) + (n+d)\text{poly}(k/\epsilon)$ time and dimension $\text{poly}(k/\epsilon)$

• **(Nice M-Estimators)** For $L = (\log n)^{O(\log k)}$, in $O(\text{nnz}(A)) + (n+d)\text{ poly}(L/\epsilon)$
  time, get weak coreset of dimension $\text{poly}(L/\epsilon)$
Template Algorithm

1. (Create Probabilities) Find probabilities $p_1, p_2, \ldots, p_n$, $\sum p_i = \text{poly}(k)$

2. (Sample) Include the $i$-th row of $A$ in a sample set $S$ independently with probability $p_i$

3. (Adaptively Sample) Sample a set $T$ of $\text{poly}(k/\epsilon)$ rows of $A$ proportional to their "residual" $M(|A_i - A_i P_S|_2)$

4. (Brute Force) Find the best $k$-dimensional subspace in span($S \cup T$)

What are $p_1, \ldots, p_n$? For $p = 1$:
- Compute $AR$ for a CountSketch matrix $R$ with $c = \text{poly}(k)$ columns
- Let $U \in \mathbb{R}^{n \times c}$, $\text{colspan}(U) = \text{colspan}(AR)$, and for all vectors $x$,
  $|x|_1 \leq |Ux|_1 \leq \text{poly}(k)|x|_1$
- $p_i = |e_i U|_1$
Talk Outline

• Regression
  – Sketching for least squares regression
  – Sketching for fast robust regression

• Low Rank Approximation
  – Sketching for fast SVD
  – Sketching for fast robust low rank approximation

• Recent sketching/sampling work for robust problems
Recent Work

• Low rank approximation with entrywise $\ell_p$-norm loss
  • [SWZ17]: for $p$ in $[1,2)$, get a $\text{poly}(k \log n)$-approximation in $\text{nnz}(A) + n\text{poly}(k \log n)$ time
  • [CGKMW17]: for any $p \geq 1$, get a $\text{poly}(k \log n)$-approximation in $\text{poly}(n)$ time
  • [BKW17]: for $p = 0$, i.e., robust PCA, get $\text{poly}(k \log n)$-approximation with a weak coreset of size $\text{poly}(k \log n)$
  • [BBKMW18]: for $p$ in $(0,2)$, get a $(1+\epsilon)$-approximation in $n^{\text{poly}(\frac{k}{\epsilon})}$ time
General Robust Loss Functions

• Find rank-k \( \hat{A} \) with \( |\hat{A} - A|_g \leq \alpha \cdot \min_{\text{rank-k} B} |A - B|_g \) for approximation factor \( \alpha \geq 1 \)

• For a matrix \( C \), \( |C|_g = \sum_{i,j} g(C_{i,j}) \) where \( g: \mathbb{R} \rightarrow \mathbb{R}^{\geq 0} \)

• [SWZ18]: in poly time, get a poly\((k \log n)\)-approximation with a weak coreset of size poly\((k \log n)\), for any \( g \) which
  
  • has approximate triangle inequality
  
  • and is monotone and approximately symmetric
  
  • and has an efficient regression algorithm

• Includes, e.g., Huber loss function
Tukey Regression [CWW]

Regression algorithms for loss functions which “plateau”

• For Tukey Biweight loss $M$, and regression $\min_{x} |Ax-b|_{M}$, in $\text{nnz}(A) \log n$ time can reduce to a small $\text{poly}(d/\epsilon)$-sized problem

• NP-hard to approximate $|Ax-b|_{M}$ up to a constant factor