CS 15-859: Algorithms for Big Data Fall 2017

Lecture 11-1 — November 15

Prof. David Woodruff *Scribe: Paul Liang*

1 Weighted Low Rank Approximation

1.1 Problem Setup

In the Weighted Low Rank Approximation (WLRA) problem, we are given: $A \in \mathbb{R}^{n\times n}$, $W \in \mathbb{R}^{n\times n}$, $k \in \mathbb{N}$, $\epsilon > 0$. A is the input matrix and W is the weight matrix. Our goal is to output:

$$\text{rank-}k\ \hat{A} \in \mathbb{R}^{n\times n} \text{ such that}$$

$$\|W \circ (\hat{A} - A)\|_F^2 \leq (1+\epsilon) \min_{\text{rank-}k\ A'} \|W \circ (A' - A)\|_F^2$$

Let us define:

$$\text{OPT} = \min_{\text{rank-}k\ A'} \|W \circ (A' - A)\|_F^2 = \min_{\text{rank-}k\ A'} \sum_{i,j} W_{i,j}^2 (A'_{i,j} - A_{i,j})^2$$

Since $\hat{A}$ is rank k, we can alternatively write this as output:

$$U, V^T \in \mathbb{R}^{n\times k} \text{ such that}$$

$$\|W \circ (UV - A)\|_F^2 \leq (1+\epsilon)\ \text{OPT}$$

1.2 Motivation

Suppose there are several movies that can be grouped into several categories (eg. action, comedy, historical, cartoon, magical) and each movie is rated by several people. Each rater has his own distribution from which he chooses to assign scores to movies, so it seems natural to scale each rater's score by the inverse of his distribution's standard deviation. Therefore a weight matrix W can be constructed for this purpose for fair comparison across ratings.

Another motivation comes from a closely related problem, Matrix Completion. We are given $A \in \{\mathbb{R}, ?\}^{n\times n}$, $\Omega \subset [n] \times [n]$ and $k \in \mathbb{R}$. A is the entire matrix with some entries missing denoted as ?, Ω are the observed entries. We want to fill in the missing entries of A such that $\text{rank}(A) = k$. In other words output $A_{\bar{\Omega}}$ such that $\text{rank}(A) = k$. To solve this problem we can assign 0 weight to the missing entries and solve the Weighted Low Rank Approximation problem with the given k.

Finally many real-life datasets such as RateBeer, documents, amazon and biology all use Weighted Low Rank Approximations.

1.3 Results

The following results are from [5]. We will make one of the following assumptions on the weight matrix W to simplify the problem:

1. W has r distinct rows and columns.
2. W has r distinct columns.
3. W has rank at most r.

W has r distinct rows and columns:

Given: $A \in \mathbb{R}^{n\times n}$, $W \in \mathbb{R}^{n\times n}$ with r distinct rows and columns, $k \in \mathbb{N}$, $\epsilon > 0$. We can output:

$$\text{rank-}k\ \hat{A} \in \mathbb{R}^{n\times n} \text{ such that}$$

$$\|W \circ (\hat{A} - A)\|_F^2 \leq (1+\epsilon) \text{ OPT}$$

with probability 9/10 in:

$$O((\text{nnz}(A) + \text{nnz}(W)) \cdot n^\gamma) + n \cdot 2^{\tilde{O}(k^2 r/\epsilon)}$$

time, for an arbitrarily small constant $\gamma > 0$.

W has r distinct columns:

The time complexity is now:

$$O((\text{nnz}(A) + \text{nnz}(W)) \cdot n^\gamma) + n \cdot 2^{\tilde{O}(k^2 r^2/\epsilon)}$$

time, for an arbitrarily small constant $\gamma > 0$.

W has rank r:

The time complexity is now:

$$n^{O(k^2 r^2/\epsilon)}$$

time. Previously only $r = 1$ was known to be in polynomial time.

1.4 Hardness Results

For this we will use the the Random-4SAT Hypothesis [2, 3]. Given a random 4-SAT formula $\mathcal{S}$ on n variables, each clause with 4 literals. suppose each of the $\Theta(n^4)$ clauses are picked independently with probability $\Theta(1/n^3)$, $m = \Theta(n)$ is the number of clauses, any algorithm that outputs 1 with probability 1 when $\mathcal{S}$ is satisfiable and outputs 0 with probability $\geq 1/2$ requires $2^{\Omega(n)}$ time. Note that the probability is over the input instances.

We will also consider the Maximum Edge Biclique (MEB) problem. Given a bipartite graph $G = (U, V, E)$ with $|U| = |V| = n$, we wish to output a k_1-by-k_2 complete bipartite subgraph of G such that $k_1 \cdot k_2$ is maximized.

The following result is from [3]. Assume that the Random-4SAT Hypothesis is true, then there exists 2 constants $\epsilon_1 > \epsilon_2 > 0$ such that any algorithm that distinguishes between bipartite graphs $G = (U, V, E)$ with $|U| = |V| = n$ in 2 cases:

1. there is a bipartite clique of size $\geq (n/16)^2(1 + \epsilon_1)$
2. all bipartite cliques are of size $\leq (n/16)^2(1 + \epsilon_2)$

requires $2^{\Omega(n)}$ time.

Additionally, we have a reduction from MEB to WLRA: Given an instance of the MEB problem, a bipartite graph $G = (U, V, E)$ with $|U| = |V| = n$, we wish to output a k_1-by-k_2 complete bipartite subgraph of G such that $k_1 \cdot k_2$ is maximized. Note that this problem is equivalently hard to its complement: when we wish to output a k_1-by-k_2 complete bipartite subgraph of G such that $|E| - k_1 \cdot k_2$ is minimized. Now we can transform this instance of the MEB problem to an instance of the WLRA problem: set matrix A as follows:

$$A_{i,j} = \begin{cases} 1 & \text{if edge } (U_i, V_j) \text{ exists} \\ 0 & \text{else} \end{cases}$$

and set the weight matrix W as follows:

$$W_{i,j} = \begin{cases} 1 & \text{if edge } (U_i, V_j) \text{ exists} \\ n^6 & \text{else} \end{cases}$$

This implies that if you can get a $(1 + \epsilon)$ close approximation to OPT in WLRA, even with $r = 1$ and with an arbitrary W matrix (not necessarily low rank or with distinct rows and/or columns), then I can obtain a $(1 + \epsilon)$ close approximation to OPT in MEB complement. But we know that the MEB complement problem and MEB problem takes $2^{\Omega(n)}$ time if we assume the Random-4SAT Hypothesis. Stating this formally, given $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$ with r distinct columns, $k \in \mathbb{N}$, $\epsilon > 0$, $W_{i,j} \in \{0, 1, 2, ..., \text{poly}(n)\}$, to output:

$$\text{rank-}k \ \hat{A} \in \mathbb{R}^{n \times n} \text{ such that}$$

$$\|W \circ (\hat{A} - A)\|_F^2 \leq (1 + \epsilon) \text{ OPT}$$

with probability 9/10 and assuming the Random-4SAT Hypothesis, $\exists \epsilon_0$ such that for any algorithm with $\epsilon < \epsilon_0$ and $k \geq 1$ takes $2^{\Omega(r)}$ time.

1.5 Algorithmic Techniques

Despite this hardness result, we can improve the time complexity using several algorithmic techniques. We will use the following:

Polynomial System Verifier

Recall the Polynomial System Verifier from [1, 6]. Given a real polynomial system $P(x)$ with v variables, $x = (x_1, x_2, ..., x_v)$, m polynomial constraints $f_i(x) \geq 0$, $\forall i \in [m]$, d the maximum

maximum degree of all polynomial constraints, H the bitsizes of the coefficients of the polynomials, it takes:

$$(md)^{O(v)}\text{poly}(H)$$

time to decide if there exists a solution to polynomial system P.

Lower Bound on the Cost

This result is from [4]. Define $T = \{x \in \mathbb{R}^v | f_1(x) \geq 0, ..., f_m(x) \geq 0\}$, the feasible set. Suppose we have a non-negative polynomial $G(x)$ and we evaluate $G(x)$ for all $x \in T$. Suppose I tell you that $\min_{x \in T} G(x) > 0$. How small can $\min_{x \in T} G(x)$ be? It can get arbitrary small! For example, consider $x_1, x_2 \in \mathbb{R}$, no f_i constraints, and let $G(x_1, x_2) = (x_1 x_2 - 1)^2 + x_2^2$. $G(x_1, x_2) > 0$ since $G(x_1, x_2) = 0$ must imply that $x_2 = 0$ which causes $(x_1 x_2 - 1)^2 = 1$, a contradiction. But we can make $G(x_1, x_2)$ arbitrary small by setting $x_1 \to \infty$ and $x_2 = 1/x_1 \to 0$.

To solve this problem, we can intersect T with a Ball $= \{x : \|x\|_\infty \leq 2^H\}$. In the case above this would prevent x_1 from increasing towards ∞, effectively setting a lower bound on $G(x_1, x_2)$. Then the minimum value that nonnegative G takes over $T \cap$ Ball is either 0 or $\geq (2^H + m)^{-d^v}$. This will be our lower bound on the cost of the polynomial system, so that we can perform binary search over the cost and know when to stop. Similar to the ellipsoid algorithm for linear programming.

Multiple Regression Sketch

Given: $A^{(1)}, A^{(2)}, ..., A^{(m)} \in \mathbb{R}^{n \times k}$ and $b^{(1)}, b^{(2)}, ..., b^{(m)} \in \mathbb{R}^{n \times 1}$, let:

$$x^{(j)} = \underset{x \in \mathbb{R}^{k \times 1}}{\arg\min} \|A^{(j)}x - b^{(j)}\|_2^2 \quad \forall j \in [m]$$

Choose S to be a random Gaussian matrix of size $t \times n$, and denote the sketched solution:

$$y^{(j)} = \underset{y \in \mathbb{R}^{k \times 1}}{\arg\min} \|SA^{(j)}x - Sb^{(j)}\|_2^2 \quad \forall j \in [m]$$

We have the following guarantee: for all $\epsilon \in (0, 1/2)$, one can set $t = O(k/\epsilon)$ such that:

$$\sum_{j=1}^{m} \|A^{(j)}y^{(j)} - b^{(j)}\|_2^2 \leq (1+\epsilon) \sum_{j=1}^{m} \|A^{(j)}x^{(j)} - b^{(j)}\|_2^2$$

with probability 9/10.

1.6 Warmup: Inefficient WLRA Algorithm

Given: $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $k \in \mathbb{N}$, $\epsilon > 0$, suppose $A_{i,j} \in \{0, \pm 1, \pm 2, ..., \pm \Delta\}$, $W_{i,j} \in \{0, 1, 2, ..., \Delta\}$. Output

$$\text{rank-}k\ \hat{A} \in \mathbb{R}^{n \times n} \text{ such that}$$

$$\|W \circ (\hat{A} - A)\|_F^2 \leq (1+\epsilon)\ \text{OPT}$$

with probability 9/10. The naive algorithm is as follows:

1. Create $2nk$ variables for $U, V^T \in \mathbb{R}^{n \times k}$.
2. Write polynomial $g(x_1, ..., x_{2nk}) = \|W \circ (A - UV)\|_F^2$.

3. Pick $C \in [L^-, L^+]$, run polynomial verifier $g(x) \leq C$.
4. Optimize C by binary search over $[L^-, L^+]$.

The runtime is $2^{\Omega(nk)}$. How can we do better? The step which incurs too much time is step 1, where we create $2nk$ variables for U and V. Recall that polynomial verifier runs in $(\#constraints \cdot degree)^{O(\#variables)}$ and the lower bound on the cost is $(\#constraints)^{-degree^{O(\#variables)}}$. We would like to write a polynomial with a few number of variables ($\text{poly}(kr/\epsilon)$) without without blowing up degree and number of constraints too much.

1.7 Main Idea: Guess a Sketch

To reduce the number of variables to $\text{poly}(kr/\epsilon)$, given that we can do Multiple regression sketch with $O(k/\epsilon)$ rows and the Weight matrix W has rank at most r. Given the same inputs A, W, r, k, ϵ, the algorithm is as follows. Let W_j be the j-th column of W and set D_{W_j} to be a diagonal matrix with vector W_j. The objective function is:

$$\|W \circ (UV - A)\|_F^2 \leq (1+\epsilon) \text{ OPT}$$

which we can rewrite as:

$$\sum_{j=1}^{n} \|D_{W_j} U V_j - D_{W_j} A_j\|_F^2 \leq (1+\epsilon) \text{ OPT} \qquad \sum_{i=1}^{n} \|U^i V D_{W_i} - A^i D_{W_i}\|_F^2 \leq (1+\epsilon) \text{ OPT}$$

which is in the form of a Multiple Regression problem, so we can sketch by Gaussian Matrix $S, T^T \in \mathbb{R}^{t \times n}$:

$$\sum_{j=1}^{n} \|S D_{W_j} U V_j - S D_{W_j} A_j\|_F^2 \qquad \sum_{i=1}^{n} \|U^i V D_{W_i} T - A^i D_{W_i} T\|_F^2$$

and we can guess $SD_{W_j}U \in \mathbb{R}^{(t \times k)}$ (and $VD_{W_i}T \in \mathbb{R}^{(k \times t)}$) by creating $t \times k$ variables for each of n $SD_{W_j}U$s. But this would take $n \times t \times k$ variables! The key is to notice that W has rank r so certain columns are linear combinations of others so we actually do not have to create variables for all n $SD_{W_j}U$s. More formally, let W_j be the j-th column of W, and consider the column span of W which only has r column vectors. So we only create variables for $SD_{W_j}U \in \mathbb{R}^{(t \times k)}$, $\forall j \in [r]$. For those W_j not in the column span of W, we can write it as a linear combination of the vectors in the column span and express $SD_{W_j}U$ as a linear combination of the existing variables in the column span (eg. write $SD_{W_4}U = SD_{W_1}U + SD_{W_2}U$ using existing variables $SD_{W_1}U, SD_{W_2}U$ if $W_1, W_2 \in \text{col}(W)$ but $W_4 \in \text{col}(W)$). We create $t \times k$ variables for each $SD_{W_j}U \in \mathbb{R}^{(t \times k)}$, giving a total of $r \times t \times k$ variables.

1.8 Wrapping Up

We can use an explicit formula for the regression solution in terms of the variables created. The regression solution is a rational function, but can use tricks to clear the denominator. Multiple Regression Sketch + Bounded Rank Weight Matrix imply a small number of variables, and runs in time $n^{O(rk^2/\epsilon)}$.

1.9 Open Problems

For a rank-r weight matrix W, the upper bound is $n^{O(k^2 r/\epsilon)}$ but the lower bound is only $2^{\Omega(r)}$, can we close this gap? Can we prove a hardness result with respect to the parameter k, e.g., a $2^{\Omega(k)}$ lower bound for WLRA problem?

1.10 Conclusion

Overall, we studied intractable matrix factorization problems through the lens of parameterized complexity (nonnegative matrix factorization, weighted low rank approximation). Parameterized Complexity gives a way of coping with intractability for emerging machine learning problems.

References

[1] Saugata Basu, Richard Pollack, and Marie-Françoise Roy. On the combinatorial and algebraic complexity of quantifier elimination. *J. ACM*, 43(6):1002–1045, November 1996. URL: `http://doi.acm.org/10.1145/235809.235813`, `doi:10.1145/235809.235813`.

[2] Uriel Feige. Relations between average case complexity and approximation complexity. In *Proceedings of the Thiry-fourth Annual ACM Symposium on Theory of Computing*, STOC '02, pages 534–543, New York, NY, USA, 2002. ACM. URL: `http://doi.acm.org/10.1145/509907.509985`, `doi:10.1145/509907.509985`.

[3] Andreas Goerdt and André Lanka. On the hardness and easiness of random 4-sat formulas. In *ISAAC*, volume 3341 of *Lecture Notes in Computer Science*, pages 470–483. Springer, 2004.

[4] Gabriela Jeronimo, Daniel Perrucci, and Elias P. Tsigaridas. On the minimum of a polynomial function on a basic closed semialgebraic set and applications. *SIAM Journal on Optimization*, 23(1):241–255, 2013.

[5] Ilya Razenshteyn, Zhao Song, and David P. Woodruff. Weighted low rank approximations with provable guarantees. In *Proceedings of the Forty-eighth Annual ACM Symposium on Theory of Computing*, STOC '16, pages 250–263, New York, NY, USA, 2016. ACM. URL: `http://doi.acm.org/10.1145/2897518.2897639`, `doi:10.1145/2897518.2897639`.

[6] James Renegar. On the computational complexity and geometry of the first-order theory of the reals. part i: Introduction. preliminaries. the geometry of semi-algebraic sets. the decision problem for the existential theory of the reals. *Journal of Symbolic Computation*, 13(3):255 – 299, 1992. URL: `http://www.sciencedirect.com/science/article/pii/S0747717110800033`, `doi:https://doi.org/10.1016/S0747-7171(10)80003-3`.