Sketching as a Tool for Numerical Linear Algebra

David P. Woodruff
IBM Research Almaden
dpwoodru@us.ibm.com
Editorial Scope

Topics

Foundations and Trends® in Theoretical Computer Science publishes surveys and tutorials on the foundations of computer science. The scope of the series is broad. Articles in this series focus on mathematical approaches to topics revolving around the theme of efficiency in computing. The list of topics below is meant to illustrate some of the coverage, and is not intended to be an exhaustive list.

- Algorithmic game theory
- Computational algebra
- Computational aspects of combinatorics and graph theory
- Computational aspects of communication
- Computational biology
- Computational complexity
- Computational geometry
- Computational learning
- Computational Models and Complexity
- Computational Number Theory
- Cryptography and information security
- Data structures
- Database theory
- Design and analysis of algorithms
- Distributed computing
- Information retrieval
- Operations research
- Parallel algorithms
- Quantum computation
- Randomness in computation

Information for Librarians

Foundations and Trends® in Theoretical Computer Science, 2014, Volume 10, 4 issues. ISSN paper version 1551-305X. ISSN online version 1551-3068. Also available as a combined paper and online subscription.
Sketching as a Tool for Numerical Linear Algebra

David P. Woodruff
IBM Research Almaden
dpwoodru@us.ibm.com
Contents

1 Introduction 2

2 Subspace Embeddings and Least Squares Regression 9
   2.1 Subspace embeddings . . . . . . . . . . . . . . . . . . . . . . 10
   2.2 Matrix multiplication . . . . . . . . . . . . . . . . . . . . . . . 22
   2.3 High probability . . . . . . . . . . . . . . . . . . . . . . . . . 28
   2.4 Leverage scores . . . . . . . . . . . . . . . . . . . . . . . . . . . 29
   2.5 Regression . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 34
   2.6 Machine precision regression . . . . . . . . . . . . . . . . . . . 37
   2.7 Polynomial fitting . . . . . . . . . . . . . . . . . . . . . . . . . . 40

3 Least Absolute Deviation Regression 42
   3.1 Sampling-Based solution . . . . . . . . . . . . . . . . . . . . . . 43
   3.2 The Role of subspace embeddings for L1-Regression . . . . . 48
   3.3 Gaussian sketching to speed up sampling . . . . . . . . . . . . 50
   3.4 Subspace embeddings using cauchy random variables . . . . 51
   3.5 Subspace embeddings using exponential random variables . . 57
   3.6 Application to hyperplane fitting . . . . . . . . . . . . . . . . . 65

4 Low Rank Approximation 68
   4.1 Frobenius norm error . . . . . . . . . . . . . . . . . . . . . . . . 70
   4.2 CUR decomposition . . . . . . . . . . . . . . . . . . . . . . . . . 75
Abstract

This survey highlights the recent advances in algorithms for numerical linear algebra that have come from the technique of linear sketching, whereby given a matrix, one first compresses it to a much smaller matrix by multiplying it by a (usually) random matrix with certain properties. Much of the expensive computation can then be performed on the smaller matrix, thereby accelerating the solution for the original problem. In this survey we consider least squares as well as robust regression problems, low rank approximation, and graph sparsification. We also discuss a number of variants of these problems. Finally, we discuss the limitations of sketching methods.

To give the reader a flavor of results in this survey, let us first consider the classical linear regression problem. In a special case of this problem one attempts to “fit” a line through a set of given points as best as possible.

For example, the familiar Ohm’s law states that the voltage $V$ is equal to the resistance $R$ times the electrical current $I$, or $V = R \cdot I$. Suppose one is given a set of $n$ example volate-current pairs $(v_j, i_j)$ but does not know the underlying resistance. In this case one is attempting to find the unknown slope of a line through the origin which best fits these examples, where best fits can take on a variety of different meanings.

More formally, in the standard setting there is one measured variable $b$, in the above example this would be the voltage, and a set of $d$ predictor variables $a_1, \ldots, a_d$. In the above example $d = 1$ and the single predictor variable is the electrical current. Further, it is assumed that the variables are linearly related up to a noise variable, that is $b = x_0 + a_1 x_1 + \cdots + a_d x_d + \gamma$, where $x_0, x_1, \ldots, x_d$ are the coefficients of a hyperplane we are trying to learn (which does not go through the origin if $x_0 \neq 0$), and $\gamma$ is a random variable which may be adversarially
chosen, or may come from a distribution which we may have limited or no information about. The $x_i$ are also known as the \textit{model parameters}. By introducing an additional predictor variable $a_0$ which is fixed to 1, we can in fact assume that the unknown hyperplane goes through the origin, that is, it is an unknown subspace of codimension 1. We will thus assume that $b = a_1 x_1 + \cdots + a_d x_d + \gamma$ and ignore the affine component throughout.

In an experiment one is often given $n$ observations, or $n$ $(d + 1)$-tuples $(a_{i,1}, \ldots, a_{i,d}, b_i)$, for $i = 1, 2, \ldots, n$. It is more convenient now to think of the problem in matrix form, where one is given an $n \times d$ matrix $A$ whose rows are the values of the predictor variables in the $d$ examples, together with an $n \times 1$ column vector $b$ whose entries are the corresponding observations, and the goal is to output the coefficient vector $x$ so that $Ax$ and $b$ are close in whatever the desired sense of closeness may mean. Notice that as one ranges over all $x \in \mathbb{R}^d$, $Ax$ ranges over all linear combinations of the $d$ columns of $A$, and therefore defines a $d$-dimensional subspace of $\mathbb{R}^n$, which we refer to as the column space of $A$. Therefore the regression problem is equivalent to finding the vector $x$ for which $Ax$ is the closest point in the column space of $A$ to the observation vector $b$.

Much of the focus of this survey will be on the over-constrained case, in which the number $n$ of examples is much larger than the number $d$ of predictor variables. Note that in this case there are more constraints than unknowns, and there need not exist a solution $x$ to the equation $Ax = b$.

Regarding the measure of fit, or closeness of $Ax$ to $b$, one of the most common is the least squares method, which seeks to find the closest point in Euclidean distance, i.e.,

$$\arg\min_x \|Ax - b\|_2 = \sum_{i=1}^{n} (b_i - \langle A_{i,*}, x \rangle)^2,$$

where $A_{i,*}$ denotes the $i$-th row of $A$, and $b_i$ the $i$-th entry of the vector $b$. This error measure has a clear geometric interpretation, as the optimal $x$ satisfies that $Ax$ is the standard Euclidean projection of $b$ onto the column space of $A$. Because of this, it is possible to write the solution for this problem in a closed form. That is, necessarily one
has $A^T Ax^* = A^T b$ for the optimal solution $x^*$ by considering the gradient at a point $x$, and observing that in order for it to be 0, that is for $x$ to be a minimum, the above equation has to hold. The equation $A^T Ax^* = A^T b$ is known as the normal equation, which captures that the line connecting $Ax^*$ to $b$ should be perpendicular to the columns spanned by $A$. If the columns of $A$ are linearly independent, $A^T A$ is a full rank $d \times d$ matrix and the solution is therefore given by $x^* = (A^T A)^{-1} A^T b$. Otherwise, there are multiple solutions and a solution $x^*$ of minimum Euclidean norm is given by $x^* = A^\dagger b$, where $A^\dagger$ is the Moore-Penrose pseudoinverse of $A$. Recall that if $A = U \Sigma V^T$ is the singular value decomposition (SVD) of $A$, where $U$ is $n \times d$ with orthonormal columns, $\Sigma$ is a diagonal $d \times d$ matrix with non-negative non-increasing diagonal entries, and $V^T$ is a $d \times d$ matrix with orthonormal rows, then the Moore-Penrose pseudoinverse of $A$ is the $d \times n$ matrix $V \Sigma^\dagger U^T$, where $\Sigma^\dagger$ is a $d \times d$ diagonal matrix with $\Sigma^\dagger_{i,i} = 1/\Sigma_{i,i}$ if $\Sigma_{i,i} > 0$, and is 0 otherwise.

The least squares measure of closeness, although popular, is somewhat arbitrary and there may be better choices depending on the application at hand. Another popular choice is the method of least absolute deviation, or $\ell_1$-regression. Here the goal is to instead find $x^*$ so as to minimize

$$
\|Ax - b\|_1 = \sum_{i=1}^{n} |b_i - \langle A_{i,*}, x \rangle| .
$$

This measure is known to be less sensitive to outliers than the least squares measure. The reason for this is that one squares the value $b_i - \langle A_{i,*}, x \rangle$ in the least squares cost function, while one only takes its absolute value in the least absolute deviation cost function. Thus, if $b_i$ is significantly larger (or smaller) than $\langle A_{i,*}, x \rangle$ for the $i$-th observation, due, e.g., to large measurement noise on that observation, this requires the sought hyperplane $x$ to be closer to the $i$-th observation when using the least squares cost function than when using the least absolute deviation cost function. While there is no closed-form solution for least absolute deviation regression, one can solve the problem up to machine precision in polynomial time by casting it as a linear programming problem and using a generic linear programming algorithm.
The problem with the above solutions is that on massive data sets, they are often too slow to be of practical value. Using naïve matrix multiplication, solving the normal equations for least squares would take at least $n \cdot d^2$ time. For least absolute deviation regression, when casting the problem as a linear program one needs to introduce $O(n)$ variables (these are needed to enforce the absolute value constraints) and $O(n)$ constraints, and generic solvers would take $\text{poly}(n)$ time for an polynomial in $n$ which is at least cubic. While these solutions are polynomial time, they are prohibitive for large values of $n$.

The starting point of this survey is a beautiful work by Tamás Sarlós [105] which observed that one could use sketching techniques to improve upon the above time complexities, if one is willing to settle for a randomized approximation algorithm. Here, one relaxes the problem to finding a vector $x$ so that $\|Ax - b\|_p \leq (1 + \varepsilon)\|Ax^* - b\|_p$, where $x^*$ is the optimal hyperplane, with respect to the $p$-norm, for $p$ either 1 or 2 as in the discussion above. Moreover, one allows the algorithm to fail with some small probability $\delta$, which can be amplified by independent repetition and taking the best hyperplane found.

While sketching techniques will be described in great detail in the following chapters, we give a glimpse of what is to come below. Let $r \ll n$, and suppose one chooses a $r \times n$ random matrix $S$ from a certain distribution on matrices to be specified. Consider the following algorithm for least squares regression:

1. Sample a random matrix $S$.
2. Compute $S \cdot A$ and $S \cdot b$.
3. Output the exact solution $x$ to the regression problem $\min_x \|(SA)x - (Sb)\|_2$.

Let us highlight some key features of this algorithm. First, notice that it is a black box reduction, in the sense that after computing $S \cdot A$ and $S \cdot b$, we then solve a smaller instance of least squares regression, replacing the original number $n$ of observations with the smaller value of $r$. For $r$ sufficiently small, we can then afford to carry out step 3, e.g., by computing and solving the normal equations as described above.
The most glaring omission from the above algorithm is which random families of matrices $S$ will make this procedure work, and for what values of $r$. Perhaps one of the simplest arguments is the following. Suppose $r = \Theta(d/\varepsilon^2)$ and $S$ is a $r \times n$ matrix of i.i.d. normal random variables with mean zero and variance $1/r$, denoted $N(0, 1/r)$. Let $U$ be an $n \times (d + 1)$ matrix with orthonormal columns for which the column space of $U$ is equal to the column space of $[A, b]$, that is, the space spanned by the columns of $A$ together with the vector $b$.

Consider the product $S \cdot U$. By 2-stability of the normal distribution, i.e., if $A \sim N(0, \sigma_1^2)$ and $B \sim N(0, \sigma_2^2)$, then $A + B \sim N(0, \sigma_1^2 + \sigma_2^2)$, each of the entries of $S \cdot U$ is distributed as $N(0, 1/r)$ (recall that the column norms of $U$ are equal to 1). The entries in different rows of $S \cdot U$ are also independent since the rows of $S$ are independent. The entries in a row are also independent by rotational invariance of the normal distribution, that is, if $g \sim N(0, I_n/r)$ is an $n$-dimensional vector of normal random variables and $U_{*, 1}, \ldots, U_{*, d}$ are orthogonal vectors, then $\langle g, U_{*, 1} \rangle, \langle g, U_{*, 2} \rangle, \ldots, \langle g, U_{*, d+1} \rangle$ are independent. Here $I_n$ is the $n \times n$ identity matrix (to see this, by rotational invariance, these $d + 1$ random variables are equal in distribution to $\langle g, e_1 \rangle, \langle g, e_2 \rangle, \ldots, \langle g, e_{d+1} \rangle$, where $e_1, \ldots, e_{d+1}$ are the standard unit vectors, from which independence follows since the coordinates of $g$ are independent).

It follows that $S \cdot U$ is an $r \times (d+1)$ matrix of i.i.d. $N(0, 1/r)$ random variables. For $r = \Theta(d/\varepsilon^2)$, it is well-known that with probability $1 - \text{exp}(-d)$, all the singular values of $S \cdot U$ lie in the interval $[1 - \varepsilon, 1 + \varepsilon]$. This can be shown by arguing that for any fixed vector $x$, $\|S \cdot Ux\|_2^2 = (1 \pm \varepsilon)\|x\|_2^2$ with probability $1 - \text{exp}(-d)$, since, by rotational invariance of the normal distribution, $S \cdot Ux$ is a vector of $r$ i.i.d. $N(0, \|x\|_2^2)$ random variables, and so one can apply a tail bound for $\|S \cdot Ux\|_2^2$, which itself is a $\chi^2$-random variable with $r$ degrees of freedom. The fact that all singular values of $S \cdot U$ lie in $[1 - \varepsilon, 1 + \varepsilon]$ then follows by placing a sufficiently fine net on the unit sphere and applying a union bound to all net points; see, e.g., Theorem 2.1 of [104] for further details.

Hence, for all vectors $y$, $\|SUy\|_2 = (1 \pm \varepsilon)\|Uy\|_2$. But now consider the regression problem $\min_x \|(SA)x - (Sb)\|_2 = \min_x \|S(Ax - b)\|_2$. For each vector $x$, $Ax - b$ is in the column space of $U$, and therefore
by the previous paragraph, \( \| S(Ax - b) \|_2 = (1 \pm \varepsilon) \| Ax - b \|_2 \). It follows that by solving the regression problem \( \min_x \| (SA)x - (Sb) \|_2 \), we obtain a \((1 \pm \varepsilon)\)-approximation to the original regression problem with probability \( 1 - \exp(-d) \).

The above technique of replacing \( A \) by \( S \cdot A \) is known as a sketching technique and \( S \cdot A \) is referred to as a (linear) sketch of \( A \). While the above is perhaps the simplest instantiation of sketching, notice that it does not in fact give us a faster solution to the least squares regression problem. This is because, while solving the regression problem \( \min_x \| (SA)x - (Sb) \|_2 \) can now be done naïvely in only \( O(rd^2) \) time, which no longer depends on the large dimension \( n \), the problem is that \( S \) is a dense matrix and computing \( S \cdot A \) may now be too slow, taking \( \Theta(nrd) \) time.

Thus, the bottleneck in the above algorithm is the time for matrix-matrix multiplication. Tamás Sarlós observed \cite{105} that one can in fact choose \( S \) to come from a much more structured random family of matrices, called fast Johnson-Lindenstrauss transforms \cite{2}. These led to roughly \( O(nd \log d) + \text{poly}(d/\varepsilon) \) time algorithms for the least squares regression problem. Recently, Clarkson and Woodruff \cite{27} improved upon the time complexity of this algorithm to obtain optimal algorithms for approximate least squares regression, obtaining \( O(\text{nnz}(A)) + \text{poly}(d/\varepsilon) \) time, where \( \text{nnz}(A) \) denotes the number of non-zero entries of the matrix \( A \). We call such algorithms input-sparsity algorithms, as they exploit the number of non-zero entries of \( A \). The \( \text{poly}(d/\varepsilon) \) factors were subsequently optimized in a number of papers \cite{92, 97, 18}, leading to optimal algorithms even when \( \text{nnz}(A) \) is not too much larger than \( d \).

In parallel, work was done on reducing the dependence on \( \varepsilon \) in these algorithms from polynomial to polylogarithmic. This started with work of Rokhlin and Tygert \cite{103} (see also the Blendenpik algorithm \cite{8}), and combined with the recent input sparsity algorithms give a running time of \( O(\text{nnz}(A) \log(1/\varepsilon)) + \text{poly}(d) \) for least squares regression \cite{27}. This is significant for high precision applications of least squares regression, for example, for solving an equation of the form \( A^T Ax = A^T b \). Such equations frequently arise in interior point methods for linear programming, as well as iteratively reweighted least squares regression, which

Full text available at: http://dx.doi.org/10.1561/0400000060
is a subroutine for many important problems, such as logistic regression; see [94] for a survey of such techniques for logistic regression. In these examples \( A \) is often formed from the Hessian of a Newton step in an iteration. It is clear that such an equation is just a regression problem in disguise (in the form of the normal equations), and the (exact) solution of \( \text{argmin}_x \|Ax - b\|_2 \) provides such a solution. By using high precision approximate regression one can speed up the iterations in such algorithms.

Besides least squares regression, related sketching techniques have also been instrumental in providing better robust \( \ell_1 \)-regression, low rank approximation, and graph sparsifiers, as well as a number of variants of these problems. We will cover these applications each in more detail.

**Roadmap:** In the next chapter we will discuss least squares regression in full detail, which includes applications to constrained and structured regression. In Chapter 3, we will then discuss \( \ell_p \)-regression, including least absolute deviation regression. In Chapter 4 we will discuss low rank approximation, while in Chapter 5, we will discuss graph sparsification. In Chapter 6, we will discuss the limitations of sketching techniques. In Chapter 7, we will conclude and briefly discuss a number of other directions in this area.
References


References


References


References


References


References


References


References


