Sketching for Geometric Problems

David P. Woodruff

Carnegie Mellon University, Pittsburgh, PA, USA
dpwoodru@gmail.com

Abstract

In this invited talk at the European Symposium on Algorithms (ESA), 2017, I will discuss a tool called sketching, which is a form of data dimensionality reduction, and its applications to several problems in high dimensional geometry. In particular, I will show how to obtain the fastest possible algorithms for fundamental problems such as projection onto a flat, and also study generalizations of projection onto more complicated objects such as the union of flats or subspaces. Some of these problems are just least squares regression problems, with many applications in machine learning, numerical linear algebra, and optimization. I will also discuss low rank approximation, with applications to clustering. Finally I will mention a number of other applications of sketching in machine learning, numerical linear algebra, and optimization.

1998 ACM Subject Classification F.2 Analysis of Algorithms and Problem Complexity

Keywords and phrases dimensionality reduction, low rank approximation, projection, regression, sketching

Digital Object Identifier 10.4230/LIPIcs.ESA.2017.1

Category Invited Talk

1 Projection

Formally, in the projection problem, we are given a point $b \in \mathbb{R}^n$ and a d-dimensional flat (affine subspace) H, and would like to compute the distance of b to H. In a typical setting, n is very large, and d, while much smaller than n, is also fairly large. Thus we cannot afford algorithms that say, are exponential in d. One way of being presented H is in its coordinate representation, so we can think of H as being the set of points y of the form $y = Ax + v$, where A is an $n \times d$ matrix and v is a point in $\mathbb{R}^n$, which we think of as an offset. Note that A is a tall and thin matrix. Letting $\text{dist}(b, H)$ denote the Euclidean distance of b to H, we have that $\text{dist}(b, H) = \text{dist}(b - v, H - v)$ by translation, where $H - v$ is the set of points y of the form $y = Ax$. Thus we can write $\text{dist}(b - v, H - v) = \min_{x \in \mathbb{R}^d} \|Ax - (b - v)\|_2$, which is just a regression problem. If A has linearly independent columns, i.e., represents a d-dimensional flat instead of a lower-dimensional flat, then the solution $x^* = (A^T A)^{-1} A^T (b - v)$. One can compute x^* in $O(nd^2)$ time, or faster by using fast matrix multiplication algorithms, but for large n and d this is too slow.

In the sketch and solve paradigm, one first relaxes the problem to a randomized approximation problem, instead allowing for one to output an $x' \in \mathbb{R}^d$ for which $\|Ax' - b\|_2 \leq (1 + \epsilon)\|Ax^* - b\|_2$ with large probability. We refer the reader to the survey [21] for more details and proofs of claims, but we describe the basic idea below. The crux of the sketch and solve paradigm is to first choose S from a random family of matrices, and many such families of matrices work, with the important property that S is wide and fat, that is, it has k rows and n columns for $k \ll n$. One then computes $S \cdot A$ and $S \cdot b$. Then one replaces the original regression problem with $\min_x \|(SA)x - (Sb)\|_2$. For small k, which we should

think of as being $\text{poly}(d/\epsilon)$, this problem does not even depend on the large dimension n. Therefore, one can now afford to compute the minimizer x' to this small regression problem using the closed form expression above, in only $\text{poly}(d/\epsilon)$ time. The goal is to choose S from an appropriate random family of matrices so that if one does this, then the minimizer x' is such that $\|Ax' - b\|_2 \leq (1+\epsilon)\|Ax^* - b\|_2$ with large probability.

It turns out that a number of families of random matrices work, such as a $k \times n$ matrix S of i.i.d. normal random variables, where $k = O(d/\epsilon^2)$, and the entries in S are scaled by $1/\sqrt{k}$. The main difficulty with such matrices is that computing $S \cdot A$ is slow. That is, S is a dense matrix, and computing $S \cdot A$ naïvely takes at least nd^2/ϵ^2 time, which is even slower than the exact algorithm for computing x^*, which just took nd^2 time. Note that for the exact algorithm, the bottleneck was in the computation of $A^T A$, and note that both algorithms can be sped up with fast matrix multiplication. While this is too slow for our purposes, in a very nice paper of Sárlos [19], he showed that one could choose S from a much more structured random family of matrices called Fast Johnson Lindenstrauss transforms. This reduces the time for computing $S \cdot A$ to $nd \log n$, and using the connection to regression described above, gives an overall algorithm in $nd \log n + \text{poly}(d/\epsilon)$ time for least squares regression. While this is optimal in the matrix dimensions, often A is itself a sparse matrix and one would like algorithms which run in time proportional to the number $\text{nnz}(A)$ of non-zero entries of A. In work with Clarkson [7] we show this is in fact possible by using the so-called CountSketch matrices from the data stream literature, where we achieve an overall running time of $O(\text{nnz}(A)) + \text{poly}(d/\epsilon)$ for regression. The key property of CountSketch matrices is that they are extremely sparse, having only a single non-zero entry per column. This enables the matrix-matrix product $S \cdot A$ to be computed in only $\text{nnz}(A)$ time. This is easily shown to be optimal, as any algorithm achieving relative error for general matrices A needs to read a constant fraction of the non-zero entries, as otherwise it might miss a very large entry. A number of interesting tradeoffs between the number of rows of S and its sparsity are possible, see also the followup works [15, 17].

In many settings one does not only want to project a point to a flat, but rather to a much more complicated object, such as the union of flats. A natural question is what properties of the object allow for sparse, low-dimensional sketching matrices S. A natural concept that arises is the sphereical mean width, or equivalently, the Gaussian mean width of the object. Intuitively this measures the average fatness of an object, over all directions on the unit sphere. While the sphere is very fat, a line is not. The less fat the object, the fewer dimensions one needs to preserve the norms of points in the object by a sketching matrix. In recent work of Bourgain, Dirksen, and Nelson, sparse sketching matrices for projecting onto general objects were developed [4]. One application of this is to tensor regression [14].

2 Low Rank Approximation

I will also discuss the low rank approximation problem, where the goal is to approximate a high rank matrix by a matrix of much lower rank. Low rank matrices have fewer parameters, and consequently can be stored much more efficiently in factored form and applied to vectors very quickly. Also, in many instances one has an underlying matrix which is of low rank, which then becomes high rank because of noise that was added. Hence in some settings, low rank approximation can also be viewed as a tool for noise removal.

Formally, one is given an $n \times d$ matrix A, and think of the n rows of A as being points in $\mathbb{R}^d$. The goal is to find a rank k matrix A' such that $\|A - A'\|_F \leq (1+\epsilon)\|A - A_k\|_F$, where for a matrix B, $\|B\|_F = \left(\sum_{i \in [n], j \in [d]} B_{i,j}^2\right)^{1/2}$ is the Frobenius norm, and A_k is the

best rank-k approximation to A under Frobenius norm. A natural way of solving low rank approximation is via the truncated singular value decomposition (SVD). Recalling that any matrix A can be expressed as $U\Sigma V^T$, where U and V have orthonormal columns, and Σ is a diagonal matrix with non-negative non-increasing values as one moves down the diagonal, we have that A_k is given by zero-ing out all but the top k diagonal entries of Σ, obtaining Σ_k. This effectively selects the k leftmost vectors of U and k uppermost vectors of V^T, which are also known as the principal components.

While the SVD gives an exact solution, it runs in time $\min(nd^2, dn^2)$, which can be sped up using fast matrix multiplication, but is still much slower than what we would like. As in the case of least squares regression, we can use sketching to obtain significantly faster algorithms if we allow randomization and approximation. Namely, if we allow for outputting a rank-k matrix A' for which $\|A - A'\|_F \leq (1+\epsilon)\|A - A_k\|_F$, then we can solve this problem in $\text{nnz}(A) + (n+d)\text{poly}(k/\epsilon)$ time [7]. To get some perspective on this, even when A is dense, the time, up to $\text{poly}(k/\epsilon)$ factors, is nd, which is significantly faster than what is achievable by the SVD. For sparse matrices, we obtain even larger speedups.

The basic idea behind using sketching for low rank approximation is to first compute $S \cdot A$, where S is one of the random matrices discussed above with a small number of rows, on the order of $\text{poly}(k/\epsilon)$. One then argues that there is a $(1+\epsilon)$-approximate rank-k solution in the span of the rows of SA. It follows that by projecting each of the rows of A onto the rowspan of SA, and then working in the coordinate representation of SA, one effectively reduces the dimension from d to $\text{poly}(k/\epsilon)$. Since the running time of the SVD is $O(nd^2)$, this smaller value of d allows one to now compute the SVD in only $n \cdot \text{poly}(k/\epsilon)$ time. One argues by the Pythagorean theorem that by first projecting the rows of A onto the rowspan of SA, and then performing an SVD, that one still obtains a $(1+\epsilon)$-approximation. Choosing S to be a CountSketch matrix, this whole procedure, except for the projection of the rows of A onto the rowspan of SA, can be executed in $\text{nnz}(A) + (n+d)\text{poly}(k/\epsilon)$ time. The bottleneck is the projection of the rows of A onto the rowspan of SA, but this can be done in $\text{nnz}(A) + (n+d)\text{poly}(k/\epsilon)$ time by using the approximate projection algorithms discussed above.

I will also discuss applications of low rank approximation to k-means clustering. Here the general idea is, if given n points in $\mathbb{R}^d$, to form an $n \times d$ matrix A and then compute a so-called projection-cost preserving sketch of A, which can then be used to prove a low rank approximation with certain strong properties [10, 11, 12]. One then replaces the original dimension d with a much smaller dimension depending on only k and $1/\epsilon$. Given such a small dimension, one then runs standard algorithms from the coreset literature to reduce the number n of points to $\text{poly}(k/\epsilon)$.

3 Additional Applications

Finally, I will conclude by mentioning a number of other problems sketching has been applied to, such as special kinds of low rank approximations callled CUR decompositions, in which the goal is to approximate a matrix A by a low rank matrix in which the factors of the low rank matrix consist of actual rows and columns of A. Thus, if A has sparse rows or columns, then so do its factors. Sketching has been applied successfully to obtain $\text{nnz}(A)$ time algorithms for CUR decompositions [5, 20].

Another interesting use of sketching is to high precision regression. One might complain that the natural sketch and solve algorithm producing a vector $x' \in \mathbb{R}^d$ for which $\|Ax' - b\|_2 \leq (1+\epsilon)\|Ax^* - b\|_2$ has running time $\text{nnz}(A) + \text{poly}(d/\epsilon)$ and is undesirable if ϵ is very small.

By using sketching it is possible to obtain algorithms running in roughly $\text{nnz}(A)\log(1/\epsilon)$ time [7]. The main idea is to use sketching to obtain an $O(1)$-approximate initialization to gradient descent as well as an $O(1)$-approximate preconditioner.

Other applications include robust low rank approximation [8, 20], kernelized problems [1], distributed and streaming computation [2, 3, 6, 13], tensor low rank approximation [20], weighted low rank approximation [18], structure-preseving low rank approximation [9, 16], etc. I refer the reader to my recent monograph for many of the details and additional applications of sketching [21]. While this accompanying article to my ESA talk is primarily focused on my own work, this is just due to the nature of the talk, and please see the above monograph for many other references on these and related topics.

References

1 Haim Avron, Kenneth L. Clarkson, and David P. Woodruff. Sharper bounds for regression and low-rank approximation with regularization. In *RANDOM*, 2017.

2 Maria-Florina Balcan, Vandana Kanchanapally, Yingyu Liang, and David Woodruff. Improved distributed principal component analysis. In *Advances in Neural Information Processing Systems (NIPS)*, 2014. URL: `https://arxiv.org/pdf/1408.5823`.

3 Maria-Florina Balcan, Yingyu Liang, Le Song, David Woodruff, and Bo Xie. Communication efficient distributed kernel principal component analysis. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 725–734. ACM, 2016. URL: `https://arxiv.org/pdf/1503.06858`.

4 Jean Bourgain, Sjoerd Dirksen, and Jelani Nelson. Toward a unified theory of sparse dimensionality reduction in euclidean space. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 499–508, 2015.

5 Christos Boutsidis and David P. Woodruff. Optimal CUR matrix decompositions. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC)*, pages 353–362. ACM, `https://arxiv.org/pdf/1405.7910`, 2014.

6 Christos Boutsidis, David P. Woodruff, and Peilin Zhong. Optimal principal component analysis in distributed and streaming models. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 236–249. ACM, 2016. URL: `https://arxiv.org/pdf/1504.06729`.

7 Kenneth L. Clarkson and David P. Woodruff. Low rank approximation and regression in input sparsity time. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 81–90, 2013. URL: `https://arxiv.org/pdf/1207.6365`.

8 Kenneth .L Clarkson and David P. Woodruff. Input sparsity and hardness for robust subspace approximation. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 310–329. IEEE, 2015. URL: `https://arxiv.org/pdf/1510.06073`.

9 Kenneth L. Clarkson and David P. Woodruff. Low-rank PSD approximation in input-sparsity time. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 2061–2072, 2017.

10 Michael B. Cohen, Sam Elder, Cameron Musco, Christopher Musco, and Madalina Persu. Dimensionality reduction for k-means clustering and low rank approximation. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing (STOC)*, pages 163–172. ACM, 2015. URL: `https://arxiv.org/pdf/1410.6801`.

11 Michael B. Cohen, Jelani Nelson, and David P. Woodruff. Optimal approximate matrix product in terms of stable rank. In *Proceedings of the 43rd International Colloquium*

on Automata, Languages and Programming (ICALP), Rome, Italy, July 12-15, 2016, volume 55 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 11:1–11:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2015. URL: `https://arxiv.org/pdf/1507.02268`, `doi:10.4230/LIPIcs.ICALP.2016.11`.

12 Dan Feldman, Melanie Schmidt, and Christian Sohler. Turning big data into tiny data: Constant-size coresets for k-means, PCA and projective clustering. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 1434–1453, 2013.

13 Ravindran Kannan, Santosh S Vempala, and David P. Woodruff. Principal component analysis and higher correlations for distributed data. In *Proceedings of The 27th Conference on Learning Theory (COLT)*, pages 1040–1057, 2014.

14 Xingguo Li and David P. Woodruff. Near optimal sketching of low-rank tensor regression, 2017. Manuscript.

15 Xiangrui Meng and Michael W. Mahoney. Low-distortion subspace embeddings in input-sparsity time and applications to robust linear regression. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 91–100. ACM, 2013. URL: `https://arxiv.org/pdf/1210.3135`.

16 Cameron Musco and David P. Woodruff. Sublinear time low-rank approximation of positive semidefinite matrices. *CoRR*, abs/1704.03371, 2017.

17 Jelani Nelson and Huy L. Nguyên. OSNAP: Faster numerical linear algebra algorithms via sparser subspace embeddings. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 117–126. IEEE, 2013. URL: `https://arxiv.org/pdf/1211.1002`.

18 Ilya Razenshteyn, Zhao Song, and David P. Woodruff. Weighted low rank approximations with provable guarantees. In *Proceedings of the 48th Annual Symposium on the Theory of Computing (STOC)*, 2016.

19 Tamás Sarlós. Improved approximation algorithms for large matrices via random projections. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS), 21-24 October 2006, Berkeley, California, USA, Proceedings*, pages 143–152, 2006.

20 Zhao Song, David P. Woodruff, and Peilin Zhong. Low rank approximation with entrywise ℓ_1-norm error. In *Proceedings of the 49th Annual Symposium on the Theory of Computing (STOC)*. ACM, 2017. URL: `https://arxiv.org/pdf/1611.00898`.

21 David P. Woodruff. Sketching as a tool for numerical linear algebra. *Foundations and Trends in Theoretical Computer Science*, 10(1-2):1–157, 2014.