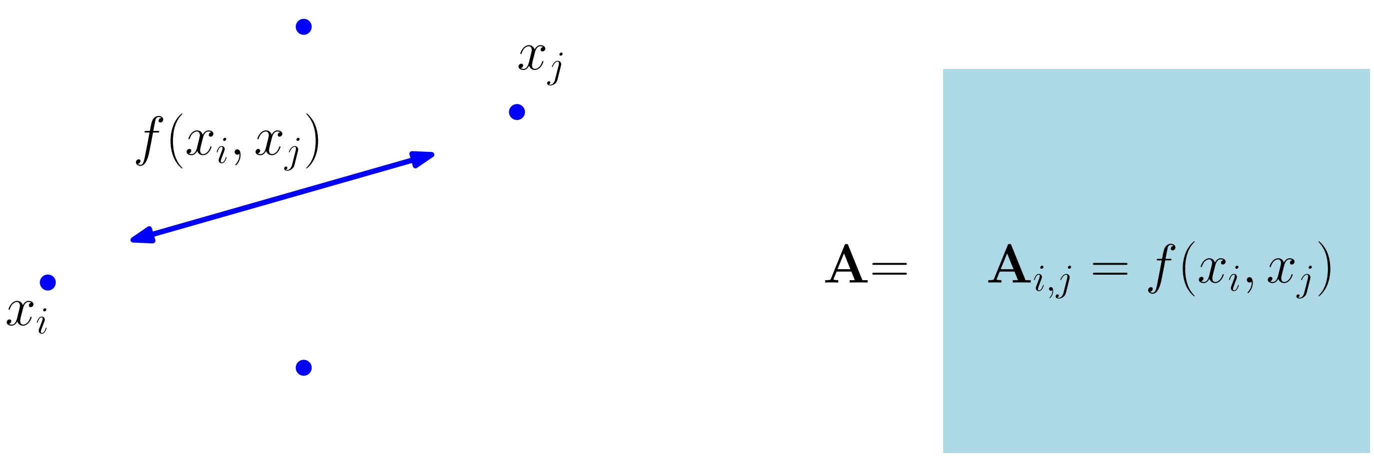
Faster Linear Algebra for Distance Matrices

Piotr Indyk, Sandeep Silwal
MIT **(Oral Designated Paper)**

Distance Matrices

Given a dataset $X \subset \mathbf{R}^d$ of n points, the $n \times n$ distance matrix $\mathbf{A}$ records the pairwise distances, under a distance function f.

We study the cases where $f = \ell_1, \ell_2, \ell_\infty$ as well as $f = \ell_p^p$ and other functions.



Distance matrices are ubiquitous in ML but require $\Omega(n^2)$ time and space to use.

Goal: Scalable and efficient algorithms for distance matrices.

Three Gems of the Paper

Consider the ℓ_1 case: $\mathbf{A}_{i,j} = \|x_i - x_j\|_1$.

Theorem 1: For any input vector y, we can compute $\mathbf{A}y$ exactly in $O(nd)$ time after $O(nd \log n)$ preprocessing.

Not all distance functions admit fast matrix vector products. Consider the case $\mathbf{A}_{i,j} = \|x_i - x_j\|_\infty$.

Theorem 2: For any $\alpha > 0$ and $d = \omega(\log n)$, any algorithm for exactly computing Az for any input z, where A is the ℓ_∞ distance matrix, requires $\Omega(n^{2-\alpha})$ time (assuming the Strong Exponential Time Hypothesis).

We can also initialize (approximate) distance matrices in time faster than previously known results.

For the ℓ_2 case, the standard way to create an approximate distance matrix is to use dimensionality reduction onto $O(\log n)$ dimensions (Johnson Lindenstrauss Lemma) and compute the distance matrix in the projected space, which takes time $O(n^2 \log n)$.

Theorem 3: For any $\varepsilon \in (0, 1)$, we can calculate $\mathbf{B}$ such that each entry of $\mathbf{B}$ satisfies $(1-\varepsilon)\|x_i - x_j\|_2 \le \mathbf{B}_{ij} \le (1+\varepsilon)\|x_i - x_j\|_2$ in time $O(\varepsilon^{-2} n^2 \log^2(\varepsilon^{-1} \log n))$.

This result requires tools **beyond** dimensionality reduction as the Jonhson Lindenstrauss Lemma is tight!

See paper for additional theoretical results and full proofs!

Applications of our Results

Matrix vector products imply faster algorithms for many **downstream applications** including:

1)Iterative Methods 2) Matrix Multiplication 3) Low-rank Approximation 4) Eigenvector Approximation 5) Linear Systems Solving

Sample of Applications for the ℓ_1 Function

Problem	Runtime	Prior Work
$(1+\varepsilon)$ Relative error rank k low-rank approximation	$\tilde{O}\left(\frac{ndk}{\varepsilon^{1/3}} + \frac{nk^{w-1}}{\varepsilon^{(w-1)/3}}\right)$	$O\left(\frac{ndk}{\varepsilon} + \frac{nk^{w-1}}{\varepsilon^{w-1}}\right)$ [BCW20]
$(1 \pm \varepsilon)$ Approximation to top k singular values	$\tilde{O}\left(\frac{ndk}{\varepsilon^{1/2}} + \frac{nk^2}{\varepsilon} + \frac{k^3}{\varepsilon^{3/2}}\right)$	$\tilde{O}\left(\frac{n^2dk}{\varepsilon^{1/2}} + \frac{nk^2}{\varepsilon} + \frac{k^3}{\varepsilon}^{3/2}\right)$ [MM15]
Multiply distance matrix A with any other $C \in \mathbb{R}^{n\times n}$	$\tilde{O}(nd)$	$O(n^w)$
Any iterative method using T matrix vector products	$\tilde{O}(ndT)$	$O(n^2d + n^2T)$

$\omega \approx 2.37$ denotes the matrix multiplication constant.

See full paper for further applications for other functions.

Experiments

We perform empirical evaluations for our ℓ_1 matrix vector product upper bound. Similar results apply for upper bound results for other functions.

As matrix-vector queries are the dominating subroutine in many key practical linear algebra algorithms such as the power method for eigenvalue estimation or iterative methods for linear regression, a fast matrix-vector query runtime automatically translates to faster algorithms for downstream applications.

Dataset	(n, d)	Algo.	Preprocessing	Query Time
Gaussians	$(5 \cdot 10^4, 50)$	Naive	453.7 s	43.3 s
		Ours	0.55 s	0.09 s
MNIST	$(5 \cdot 10^4, 784)$	Naive	2672.5 s	38.6 s
		Ours	5.5 s	1.9 s
Glove	$(1.2 \cdot 10^6, 50)$	Naive	-	$\approx$ 2.6 days
		Ours	16.8 s	3.4 s

(n, d) denotes the number of points and dimension of the dataset, respectively. Query times are averaged over 10 trials with Gaussian vectors as queries.

We observe $>$ **3 orders of magnitude** speedup over naive methods!

See paper for full details.