An Analysis of Sketched IRLS for Accelerated Sparse Residual Regression

Daichi Iwata, Michael Waechter, Wen-Yan Lin, and Yasuyuki Matsushita

Graduate School of Information Science and Technology, Osaka University
{iwata.daichi,waechter.michael,lin.daniel,yasumat}@ist.osaka-u.ac.jp

Abstract. This paper studies the problem of sparse residual regression, *i.e.*, learning a linear model using a norm that favors solutions in which the residuals are sparsely distributed. This is a common problem in a wide range of computer vision applications where a linear system has a lot more equations than unknowns and we wish to find the maximum feasible set of equations by discarding unreliable ones. We show that one of the most popular solution methods, iteratively reweighted least squares (IRLS), can be significantly accelerated by the use of matrix sketching. We analyze the convergence behavior of the proposed method and show its efficiency on a range of computer vision applications. The source code for this project can be found at https://github.com/Diwata0909/Sketched_IRLS.

Keywords: sparse residual regression, ℓ_1 minimization, randomized algorithm, matrix sketching

1 Introduction

We consider the problem of residual minimization, where we wish to learn a linear model that minimizes the residuals that deviate from the model in some distance metric. For a linear model we have a matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$ (we consider tall matrices, *i.e.*, the strongly over-determined case with $n \gg d$), a vector $\mathbf{b} \in \mathbb{R}^{n \times 1}$, $\mathbf{b} \notin \mathcal{R}(\mathbf{A})$ (the range of $\mathbf{A}$), and we seek to find

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \left\| \mathbf{A}\mathbf{x} - \mathbf{b} \right\|_p^p \tag{1}$$

for some p-norm. In this paper, we consider this linear model and call it an ℓ_p (residual) minimization problem. For the more general case including non-linear residual minimization we refer the reader to, *e.g.*, Aftab and Hartley [1] or Kiani and Drummond [36].

One of the most popular methods for ℓ_p residual minimization with $1 \leq p < 2$ is iteratively reweighted least squares (IRLS), in which weighted ℓ_2 minimization is repeatedly computed, eventually converging to the ℓ_p solution. It is generally understood that $p = 2$ is optimal for Gaussian distributed errors whereas the 1-norm leads to solutions with sparser residuals (or Laplacian distributed errors). For *sparse* residual regression, the ℓ_0 pseudo-norm is appropriate to consider; however, due to its computational complexity, a convex ℓ_1 relaxation is

often employed. Because of its capability of ignoring large outliers and reaching approximate solutions that yield sparse residuals, ℓ_1 residual minimization has become an important tool for various computer vision tasks. It can be speculated that the core reason of why ℓ_2 is still widely preferred over ℓ_1 despite ℓ_1's better suitability for many applications, is simply that ℓ_2 residual minimization can be solved efficiently in closed form while ℓ_1 residual minimization cannot. Therefore, acceleration of ℓ_1 residual minimization is wanted, especially when computing budget is limited, *e.g.*, in end-user applications on mobile devices.

In recent years, randomized algorithms are gaining attention because they offer immense speed up potential with small failure probability in various applications. In linear algebra, matrix operations can be made efficient with randomized *matrix sketching* [39,31,58]. It is based on the idea of approximating an input matrix by multiplying a randomly generated *sketching matrix* to obtain a much smaller matrix that still preserves important properties of the input matrix. In particular, certain sketching matrices $\mathbf{S}$ fulfill the *subspace embedding property* [11, Sec. 1.2]

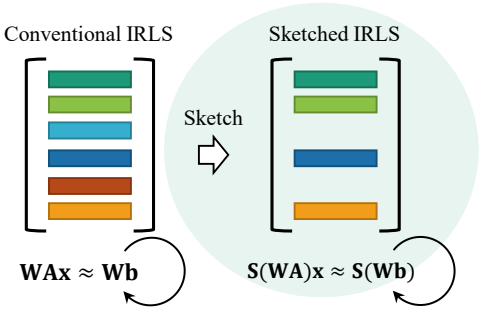


Fig. 1: Illustration of sketched IRLS: Unlike canonical IRLS, we suggest performing an approximate computation in sketched lower dimensions, yielding significant speed-up while retaining accuracy.

$$\frac{1}{\gamma}\|\mathbf{Ax}\|_2^2 \leq \|\mathbf{SAx}\|_2^2 \leq \gamma\|\mathbf{Ax}\|_2^2 \tag{2}$$

for some $\gamma > 1$ with high probability, meaning that the ℓ_2 subspace embedding $\mathbf{S}$ preserves the lengths of all vectors $\mathbf{Ax}$ within the bounds specified by γ, indicating that we can preserve $\mathbf{A}$'s range via sketching even though the sketched matrix $\mathbf{SA}$ lives in lower dimensions. This allows us to accelerate ℓ_2 regression.

In this paper, we show that ℓ_1 residual regression with IRLS can be significantly accelerated by matrix sketching, making it much more useful in practical applications. We call the new method *sketched IRLS*. The key idea is to speed up the internal computation block by projecting the linear system to lower dimensions using matrix sketching as illustrated in Fig. 1. Through comparisons with other robust techniques such as RANSAC, we also show that our method is versatile. This paper's contributions are summarized as follows:

- We propose accelerating IRLS for ℓ_1 minimization with matrix sketching,
- we analyze the error bound of sketched IRLS compared to canonical IRLS,
- and we provide an analysis of our proposed method's effectiveness on common computer vision problems using both synthetic and real-world data.

The proposed method yields the important benefit that the fundamental computer vision task of regression can be executed in an outlier-robust manner with

the flexible tool of IRLS without excessive computational burden, allowing it to be used in diverse applications with large datasets.

2 Related works

The problem we study in this paper is related to sparse regression and matrix sketching. Here we briefly review these two subject areas.

Sparse regression The field of sparse regression has recently developed rapidly, and its usefulness due to its outlier and noise robustness has been recognized in many tasks such as face recognition [59,38], image denoising [23], general signal processing [40], and background estimation [20]. Elad [22] and Zhang [61] give technical overviews on sparse regression. The goal of sparse regression is to obtain solutions where many of the unknowns become zero. A typical setting aims at minimizing the ℓ_0 norm of the solution with an under-determined constraint as

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{s.t.} \quad \mathbf{A}\mathbf{x} = \mathbf{b}. \tag{3}$$

Similar to this problem is the problem of minimizing the ℓ_0 norm of residuals (called robust sensing in [35]) with over-determined $\mathbf{A}\mathbf{x} \simeq \mathbf{b}$, expressed as

$$\min_{\mathbf{r}} \|\mathbf{r}\|_0 \quad \text{s.t.} \quad \mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}. \tag{4}$$

Both of these look for *sparse* unknowns via minimization of the ℓ_0-norm of a vector, which is NP-hard. To address this issue, various optimization strategies have been developed. One of them is based on greedy pursuit and tries to find an approximate solution for ℓ_0 minimization directly in an iterative manner [41,47,54,15,43]. Greedy methods are simple and easy to implement, but they are only guaranteed to work under very restricted conditions [52].

On the other hand, relaxation methods cast the NP-hard ℓ_0 minimization into, for example, ℓ_1 minimization, which is a convex surrogate for ℓ_0 minimization. Gorodnitsky and Rao proposed the FOCUSS algorithm [28], in which they approximate ℓ_1 minimization with the 2-norm using IRLS. It is also understood that ℓ_1 minimization can be solved by linear programming (LP) [26]. Several methods have been proposed to solve the ℓ_1 minimization problem, for example, one based on the interior point method [33], Least Angle Regression (LARS) [21] which is a homotopy algorithm, and the Iterative Shrinkage Thresholding Algorithm (ISTA) [13], to name a few. An excellent summary of recent ℓ_1 minimization methods is given by Yang *et al.* [60].

Among the ℓ_1 minimization methods, IRLS is simple and easily implemented as it only iterates weighted ℓ_2 minimization, and it is known for its fast convergence, *i.e.*, only requiring a small number of iterations to achieve an accurate solution [46,14,9]. Despite the simplicity of the algorithm, ℓ_1 minimization with IRLS shows good performance in recent applications [20,38,42]. Also, IRLS can be naturally extended to more general ℓ_p minimization and minimization with other robust functions such as Huber and the Pseudo-Huber functions [1]. To

handle nonlinear problems, generalized IRLS has also been proposed and analyzed theoretically [49,44]. Due to this simplicity and versatility, IRLS has been one of the most popular methods for ℓ_1 minimization and related problems.

Matrix sketching Randomized algorithms are attracting attention recently as methods to speed up fundamental computation. In linear algebra, matrix sketching is a randomized algorithm for matrix operations. Woodruff [58], Mahoney [39], Halko *et al.* [31], and Drineas *et al.* [16] give technical overviews on matrix sketching and randomized algorithms. In recent years, randomized algorithms have been widely applied to some specialized linear and algebraic problems, *e.g.*, *consistent* linear systems [29], *symmetric diagonally dominant* linear systems [12], and matrix inversion [30]. Matrix sketching is also used for accelerating several types of matrix decomposition, *e.g.*, singular value decomposition [31], QR decomposition [19], and dynamic mode decomposition [24].

As mentioned, matrix sketching is based on the idea that the range of an input matrix can be well approximated with high probability by random projection. Such a sketched matrix can then for example be used in regression problems as they fulfill the subspace embedding property of Eq. (2). One of the earliest sketching matrices (projectors) were random Gaussian matrices [27]: Let $\mathbf{S} \in \mathbb{R}^{s \times n}$ be a matrix randomly sampled from a Gaussian distribution. Such an $\mathbf{S}$ fulfills the subspace embedding property with $\gamma = 1 + \varepsilon$ and a small $\varepsilon \geq 0$. This is a consequence of the Johnson-Lindenstrauss lemma [32]. However, with a dense $\mathbf{S}$ sketching takes $\mathcal{O}(nds)$ time which is very costly. To overcome this, various methods with other sketching matrices have been proposed. They can roughly be divided into sampling-based and projection-based methods.

Sampling-based methods extract rows of the matrix. The simple intuition is that, in strongly overdetermined systems, the row vectors are mostly linearly dependent and a subset of them is sufficient to maintain their span. Selecting rows of the matrix is equivalent to subsampling the data points. Row sampling could be achieved by premultiplying the input matrix with a binary matrix that picks the desired rows, but for efficiency this is in practice implemented with simply selecting those rows in a streaming fashion. The simplest sampling method is *uniform sampling* that selects rows with uniform probability. Drineas *et al.* [17] devised leverage score sampling, which samples based on precomputed leverage scores that unfortunately require computing the singular value decomposition (SVD), making leverage score sampling somewhat impractical.

Projection-based methods premultiply the input matrix with more general matrices so that the codomain bases are linear combinations of multiple of $\mathbf{A}$'s domain bases and not simply selections of $\mathbf{A}$'s domain bases. Ailon and Chazelle [3] proposed the Fast Johnson-Lindenstrauss Transform (FJLT), and Tropp [53] and Drineas *et al.* [18] proposed the Subsampled Randomized Hadamard Transform (SRHT) as an extension, which take $\mathcal{O}(nd \log s)$ time. Clarkson and Woodruff further proposed the much faster *CountSketch* method [11], which was originally used in data streaming [8]. CountSketch takes only $\mathcal{O}(\mathrm{nnz}(\mathbf{A}))$ time, where $\mathrm{nnz}(\mathbf{A})$ is the number of non-zero entries of $\mathbf{A}$.

3 Proposed method: sketched IRLS

For ℓ_p minimization ($1 \leq p < 2$) with linear models, IRLS converges to Eq. (1)'s solution by iteratively minimizing

$$\mathbf{x}^{(t+1)} = \underset{\mathbf{y}}{\operatorname{argmin}} \left\| \mathbf{W}^{(t)} \mathbf{A}\mathbf{y} - \mathbf{W}^{(t)} \mathbf{b} \right\|_2^2 \tag{5}$$

with a diagonal weight matrix $\mathbf{W}^{(t)}$ at the $(t)^{\text{th}}$ iteration that is initialized with $\mathbf{W}^{(0)} = \mathbf{I}_n$. This is called the ℓ_p Weiszfeld algorithm [56,57,2]. For ℓ_1 minimization, where $p = 1$, the weights are updated as

$$\mathbf{W}_{i,i}^{(t)} = \left| \mathbf{A}_{i,*} \mathbf{x}^{(t)} - b_i \right|^{-\frac{1}{2}},$$

where $\mathbf{A}_{i,*}$ is matrix $\mathbf{A}$'s i^{th} row, and $\mathbf{W}_{i,i}$ is weight matrix $\mathbf{W}$'s i^{th} diagonal element. Solving Eq. (5) requires $\mathcal{O}(nd^2 + d^3)$ arithmetic operations and is thus expensive for large matrices. Further, since it is repeatedly solved with updated $\mathbf{W}$ in IRLS, it is the computational bottleneck for IRLS-based ℓ_p minimization.

The key idea of the proposed method is to accelerate this computation block, the weighted ℓ_2 minimization of Eq. (5), with matrix sketching. Via matrix sketching, we reduce n-dimensional to much smaller s-dimensional vectors so that the computational complexity is significantly reduced. Specifically, Eq. (5)'s weighted ℓ_2 minimization is modified as

$$\min_{\mathbf{x}} \left\| \mathbf{W}\mathbf{A}\mathbf{x} - \mathbf{W}\mathbf{b} \right\|_2^2 \quad \xrightarrow{\text{sketch}} \quad \min_{\mathbf{x}} \left\| \widetilde{\mathbf{W}\mathbf{A}}\mathbf{x} - \widetilde{\mathbf{W}\mathbf{b}} \right\|_2^2,$$

where $\mathbf{W}\mathbf{A} \in \mathbb{R}^{n \times d}$ and $\widetilde{\mathbf{W}\mathbf{A}} \in \mathbb{R}^{s \times d}$, $s \ll n$, with retaining the solution's accuracy. For adopting matrix sketching in IRLS, there are two aspects to consider; (1) when to sketch in the algorithm, and (2) the choice of the sketching method.

3.1 When to sketch?

There are two possible points in time for applying matrix sketching in IRLS; (1) sketch only once before all IRLS iterations (named *sketch-once*), and (2) sketch in every iteration of the weighted ℓ_2 minimization (named *sketch-iteratively*). We analyze the behavior of these two strategies in this paper.

Sketch-once Not only the ℓ_2 minimization but also the sketching contributes to the algorithm's overall runtime. One way to reduce the sketching costs is to only perform it once at the beginning of IRLS, outside of IRLS's iteration loop. Algorithm 1 shows pseudo code for IRLS with sketch-once.

Sketch-iteratively Unlike sketch-once, sketch-iteratively performs sketching within the IRLS iteration loop. In each IRLS iteration, we generate an $\mathbf{S} \in \mathbb{R}^{s \times n}$ with $s \ll n$ and instead of Eq. (5) we solve

$$\mathbf{x}^{(t+1)} = \underset{\mathbf{x}}{\operatorname{argmin}} \left\| \underbrace{\mathbf{S}\mathbf{W}^{(t)}\mathbf{A}}_{=\widetilde{\mathbf{W}\mathbf{A}}^{(t)}} \mathbf{x} - \underbrace{\mathbf{S}\mathbf{W}^{(t)}\mathbf{b}}_{=\widetilde{\mathbf{W}\mathbf{b}}^{(t)}} \right\|_2^2.$$

Algorithm 1 IRLS with sketch-once

Input: $\mathbf{A} \in \mathbb{R}^{n \times d}$, $\mathbf{b} \in \mathbb{R}^n$, sketching size s, #iterations T, threshold δ for termination
Output: Approximate solution $\mathbf{x}$
$\widetilde{\mathbf{W}}^{(0)} \leftarrow$ Identity matrix $\mathbf{I}_s$
$\mathbf{x}^{(0)} \leftarrow \mathbf{0}$
$\widetilde{\mathbf{A}}, \widetilde{\mathbf{b}} \leftarrow \mathrm{sketch}(\mathbf{A}, \mathbf{b}, s)$
for $t = 0 : T$ **do**
  $\mathbf{x}^{(t+1)} \leftarrow \mathrm{solve\_linear\_least\_squares}(\widetilde{\mathbf{W}}^{(t)}\widetilde{\mathbf{A}}, \widetilde{\mathbf{W}}^{(t)}\widetilde{\mathbf{b}})$
  if $\left\|\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)}\right\|_2 < \delta$ **then break end if**
  $\widetilde{\mathbf{W}}_{i,i}^{(t+1)} \leftarrow \left(\max\{\epsilon, |\widetilde{\mathbf{A}}_{i,*}\mathbf{x}^{(t+1)} - \widetilde{b}_i|\}\right)^{-\frac{1}{2}}$
end for

Algorithm 2 IRLS with sketch-iteratively

Input: $\mathbf{A} \in \mathbb{R}^{n \times d}$, $\mathbf{b} \in \mathbb{R}^n$, sketching size s, #iterations T, threshold δ for termination
Output: Approximate solution $\mathbf{x}$
$\mathbf{W}^{(0)} \leftarrow$ Identity matrix $\mathbf{I}_n$
$\mathbf{x}^{(0)} \leftarrow \mathbf{0}$
for $t = 0 : T$ **do**
  $\widetilde{\mathbf{WA}}^{(t)}, \widetilde{\mathbf{Wb}}^{(t)} \leftarrow \mathrm{sketch}(\mathbf{W}^{(t)}\mathbf{A}, \mathbf{W}^{(t)}\mathbf{b}, s)$
  $\mathbf{x}^{(t+1)} \leftarrow \mathrm{solve\_linear\_least\_squares}(\widetilde{\mathbf{WA}}^{(t)}, \widetilde{\mathbf{Wb}}^{(t)})$
  if $\left\|\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)}\right\|_2 < \delta$ **then break end if**
  $\mathbf{W}_{i,i}^{(t+1)} \leftarrow \left(\max\{\epsilon, |\mathbf{A}_{i,*}\mathbf{x}^{(t+1)} - b_i|\}\right)^{-\frac{1}{2}}$
end for

While sketch-iteratively requires more computation time than sketch-once because of the sketching operation in the loop, it is expected to be more stable because the linear system is sketched differently in each iteration. Pseudo code for IRLS with sketch-iteratively is shown in Algorithm 2.

3.2 Sketching method choice

In Sec. 2, we discussed various matrix sketching methods. In this study, we mainly focus on *uniform sampling* and *CountSketch*, because of their computational efficiency. Both make the sketching matrix sparse, resulting in sketching computation costs of only $\mathcal{O}(\mathrm{nnz}(\mathbf{A}))$ time. In practice, they can be implemented in a streaming fashion without explicitly forming the sketching matrices $\mathbf{S}$.
Uniform sampling samples rows of a tall matrix $[\mathbf{A}|\mathbf{b}]$ or $\mathbf{W}[\mathbf{A}|\mathbf{b}]$ with uniform probability so that the row-dimension can be reduced. Uniform sampling of a massively over-determined system has been employed in the past in practical applications, in a similar manner to the sketch-once approach described in the previous section. In an explicit matrix form, the sketching matrix $\mathbf{S}$ is a sparse matrix in which each row has only one element that is 1 while the rest are zeros.
CountSketch [11] is similar to uniform sampling, but instead of subsampling rows of matrix $\mathbf{A}$, it uses a sketching matrix $\mathbf{S}$ in which each column has a single randomly chosen non-zero entry (typically, 1 or -1).

4 Theoretical analysis

We now analyze the errors in IRLS solutions obtained with and without sketching. To this end, we will show that the proposed sketched IRLS can reliably derive a solution close to the non-sketched case. As mentioned, sketching can approximate a matrix's range with a certain accuracy. Since "sketch-iteratively" uses sketching repeatedly, we consider the errors introduced in each iteration. The goal is to reveal the relationship between $\left\|\mathbf{A}\widetilde{\mathbf{x}}-\mathbf{b}\right\|_1$ and $\left\|\mathbf{A}\mathbf{x}^*-\mathbf{b}\right\|_1$ for IRLS, where $\mathbf{x}^*$ is the optimal solution, and $\widetilde{\mathbf{x}}$ is the approximate solution obtained with sketched IRLS. We derive it by combining the error bounds of sketching and IRLS's convergence rate. Let $\mathbf{x}^{*(t)}$ and $\widetilde{\mathbf{x}}^{(t)}$ be the solutions of canonical and sketched IRLS, resp., after t iterations, and $\widetilde{\mathbf{x}}^{*(t+1)}$ be the solution obtained by solving Eq. (5) without sketching and using $\mathbf{W}$ based on $\widetilde{\mathbf{x}}^{(t)}$.

4.1 Condition for the residual to decrease

Before considering error bounds, we first show what condition must be fulfilled so that the residual in sketched IRLS decreases monotonically, *i.e.*, $\left\|\mathbf{A}\widetilde{\mathbf{x}}^{(t+1)}-\mathbf{b}\right\|_1 \leq \left\|\mathbf{A}\widetilde{\mathbf{x}}^{(t)}-\mathbf{b}\right\|_1$. For matrix sketching with ℓ_2 regression problems, several error bounds are known [39,48,17]. In a general form, we can write them as

$$\left\|\mathbf{A}\widetilde{\mathbf{x}}-\mathbf{b}\right\|_2 \leq (1+\varepsilon)\left\|\mathbf{A}\mathbf{x}^*-\mathbf{b}\right\|_2, \tag{6}$$

$$\left\|\widetilde{\mathbf{x}}-\mathbf{x}^*\right\|_2 = \left\|\Delta\mathbf{x}\right\|_2 \leq \varepsilon_x, \tag{7}$$

using $\widetilde{\mathbf{x}}-\mathbf{x}^* := \Delta\mathbf{x}$ and small errors ε and ε_x, which depend on the sketching method and sketching size. The error decay rate is known for canonical IRLS, *e.g.*, linear and super-linear convergence rates [49,14]. According to Daubechies [14, Sec. 6.1], for sparse variable $\mathbf{x}$ problems (Eq. (3)) we have

$$\left\|\widetilde{\mathbf{x}}^{*(t+1)}-\mathbf{x}^*\right\|_1 \leq \mu\left\|\widetilde{\mathbf{x}}^{(t)}-\mathbf{x}^*\right\|_1 \tag{8}$$

with a constant $\mu \leq 1$. It is known that sparse variable problems (Eq. (3)) and sparse residual problems (Eq. (4)) can be treated as the same problem under $\mathbf{x}=\mathbf{r}$ [7], and for sparse residual problems we have

$$\left\|(\mathbf{A}\widetilde{\mathbf{x}}^{*(t+1)}-\mathbf{b})-(\mathbf{A}\mathbf{x}^*-\mathbf{b})\right\|_1 \leq \mu\left\|(\mathbf{A}\widetilde{\mathbf{x}}^{(t)}-\mathbf{b})-(\mathbf{A}\mathbf{x}^*-\mathbf{b})\right\|_1. \tag{9}$$

From these, the residuals $\left\|\mathbf{A}\widetilde{\mathbf{x}}^{(t+1)}-\mathbf{b}\right\|_1$ and $\left\|\mathbf{A}\widetilde{\mathbf{x}}^{(t)}-\mathbf{b}\right\|_1$ satisfy

$$\begin{aligned}
\left\|(\mathbf{A}\widetilde{\mathbf{x}}^{(t+1)}-\mathbf{b})\right\|_1 - \left\|(\mathbf{A}\mathbf{x}^*-\mathbf{b})\right\|_1 &\leq \left\|(\mathbf{A}\widetilde{\mathbf{x}}^{(t+1)}-\mathbf{b})-(\mathbf{A}\mathbf{x}^*-\mathbf{b})\right\|_1 \\
&= \left\|(\mathbf{A}\widetilde{\mathbf{x}}^{*(t+1)}-\mathbf{b})+\mathbf{A}\Delta\mathbf{x}-(\mathbf{A}\mathbf{x}^*-\mathbf{b})\right\|_1 \\
&\leq \left\|(\mathbf{A}\widetilde{\mathbf{x}}^{*(t+1)}-\mathbf{b})-(\mathbf{A}\mathbf{x}^*-\mathbf{b})\right\|_1 + \left\|\mathbf{A}\Delta\mathbf{x}\right\|_1 \\
&\overset{\text{Eq. (9)}}{\leq} \mu\left\|(\mathbf{A}\widetilde{\mathbf{x}}^{(t)}-\mathbf{b})-(\mathbf{A}\mathbf{x}^*-\mathbf{b})\right\|_1 + \left\|\mathbf{A}\Delta\mathbf{x}\right\|_1 \\
&\leq \mu\left\|(\mathbf{A}\widetilde{\mathbf{x}}^{(t)}-\mathbf{b})\right\|_1 + \mu\left\|(\mathbf{A}\mathbf{x}^*-\mathbf{b})\right\|_1 + \left\|\mathbf{A}\Delta\mathbf{x}\right\|_1.
\end{aligned} \tag{10}$$

From Eq. (10), we finally have

$$\left\|(\mathbf{A}\widetilde{\mathbf{x}}^{(t+1)}-\mathbf{b})\right\|_1 \leq \mu\left\|(\mathbf{A}\widetilde{\mathbf{x}}^{(t)}-\mathbf{b})\right\|_1 + (1+\mu)\left\|(\mathbf{A}\mathbf{x}^{*}-\mathbf{b})\right\|_1 + \left\|\mathbf{A}\Delta\mathbf{x}\right\|_1, \quad (11)$$

and know that when $\left\|\mathbf{A}\Delta\mathbf{x}\right\|_1 \leq (1-\mu)\left\|(\mathbf{A}\widetilde{\mathbf{x}}^{(t)}-\mathbf{b})\right\|_1 - (1+\mu)\left\|(\mathbf{A}\mathbf{x}^{*}-\mathbf{b})\right\|_1$ holds, then $\left\|\mathbf{A}\widetilde{\mathbf{x}}^{(t+1)}-\mathbf{b}\right\|_1 \leq \left\|\mathbf{A}\widetilde{\mathbf{x}}^{(t)}-\mathbf{b}\right\|_1$. We remark that $\Delta\mathbf{x}$ satisfies Eq. (7), and when the error of sketching $\Delta\mathbf{x}$ is sufficiently close to $\mathbf{0}$, then sketched IRLS lets the objective decrease monotonically.

4.2 Worst-case residual bound

In this section, we show the relationship between the residuals of canonical IRLS $\left\|\mathbf{A}\mathbf{x}^{*(t)}-\mathbf{b}\right\|_1$ and sketched IRLS $\left\|\mathbf{A}\widetilde{\mathbf{x}}^{(t)}-\mathbf{b}\right\|_1$ after t IRLS iterations using norm relations. For canonical IRLS, after t iterations we have

$$\eta\left\|\mathbf{A}\mathbf{x}^{*(0)}-\mathbf{b}\right\|_1 = \left\|\mathbf{A}\mathbf{x}^{*(t)}-\mathbf{b}\right\|_1, \quad (12)$$

where, assuming that t is big enough for $\mathbf{x}^{*(t)}$ to converge to $\mathbf{x}^{*}$, $\eta \in [0,1]$ is the ratio between the ℓ_1 residuals of the solutions for Eq. (1) with $p=1$ and $p=2$. For sketched IRLS, under Section 4.1's condition, after t iterations we have

$$\left\|\mathbf{A}\widetilde{\mathbf{x}}^{(t)}-\mathbf{b}\right\|_1 \leq \left\|\mathbf{A}\widetilde{\mathbf{x}}^{(0)}-\mathbf{b}\right\|_1. \quad (13)$$

Further, we have the norm relations

$$\|\mathbf{x}\|_2 \leq \|\mathbf{x}\|_1 \leq \sqrt{n}\,\|\mathbf{x}\|_2, \quad (14)$$

where n is the number of dimensions of $\mathbf{x}$. We can now derive a bound on the solution error due to the sketching after t IRLS iterations:

$$\underline{\left\|\mathbf{A}\widetilde{\mathbf{x}}^{(t)}-\mathbf{b}\right\|_1} \overset{\text{Eq. (13)}}{\leq} \left\|\mathbf{A}\widetilde{\mathbf{x}}^{(0)}-\mathbf{b}\right\|_1 \overset{\text{Eq. (14)}}{\leq} \sqrt{n}\left\|\mathbf{A}\widetilde{\mathbf{x}}^{(0)}-\mathbf{b}\right\|_2 \overset{\text{Eq. (6)}}{\leq} \sqrt{n}\,(1+\varepsilon)\left\|\mathbf{A}\mathbf{x}^{*(0)}-\mathbf{b}\right\|_2$$
$$\overset{\text{Eq. (14)}}{\leq} \sqrt{n}\,(1+\varepsilon)\left\|\mathbf{A}\mathbf{x}^{*(0)}-\mathbf{b}\right\|_1 \overset{\text{Eq. (12)}}{=} \underline{\sqrt{n}\,(1+\varepsilon)\eta^{-1}\left\|\mathbf{A}\mathbf{x}^{*(t)}-\mathbf{b}\right\|_1}. \quad (15)$$

This bound may not be very tight if n is large; however, we next show that a tighter bound can be expected in practice.

4.3 Expected residual bound

The reason why the bound shown in Eq. (15) looks somewhat loose is mainly due to the right side expression of Eq. (14). The bound expresses the worst case and if that case rarely occurs, the bound does not have strong implications. Therefore, it is reasonable to consider the expected value for the ℓ_1-norm. Here, let $\mathbf{x} \in \mathbb{R}^n$ be an arbitrary vector with $\|\mathbf{x}\|_2 = r$ and the expected value for the ℓ_1-norm be $\mathbb{E}[\|\mathbf{x}\|_1]$. The expectation of the ℓ_1-norm of a vector $\mathbf{x}$ becomes $\mathbb{E}[\|\mathbf{x}\|_1] = \mathbb{E}[|x_1|] + \ldots + \mathbb{E}[|x_n|]$. In polar coordinates, $x_1, \ldots, x_n$ are

$$\begin{cases} x_1 &= r\cos\theta_1, \\ x_2 &= r\sin\theta_1\cos\theta_2, \\ & \cdots \\ x_{n-1} &= r\sin\theta_1\sin\theta_2\ldots\sin\theta_{n-2}\cos\theta_{n-1}, \\ x_n &= r\sin\theta_1\sin\theta_2\ldots\sin\theta_{n-2}\sin\theta_{n-1}. \end{cases}$$

Here, we assume that arbitrary random vectors are generated by uniformly distributed θ, which means that a probability density function $p(\theta)$ is a constant. In this case, the probability density function for x_1 depends on the arcsine distribution. Namely, for $-r \leq x_1 \leq r$, the probability density function is $p(x_1) = \frac{1}{\pi\sqrt{r^2-x_1^2}}$. Therefore, the expectation $\mathbb{E}[|x_1|]$ becomes

$$\begin{aligned}\mathbb{E}[|x_1|] &= \int_{-r}^{r} |x_1|\, p(x_1)\, \mathrm{d}x_1 = \int_{-r}^{r} |x_1| \frac{1}{\pi\sqrt{r^2-x_1^2}}\, \mathrm{d}x_1 = \frac{2}{\pi}\int_{0}^{r} \frac{x_1}{\sqrt{r^2-x_1^2}}\, \mathrm{d}x_1 \\ &= \frac{2}{\pi}\left[-\sqrt{r^2-x_1^2}\right]_0^r = \frac{2}{\pi} r. \end{aligned} \tag{16}$$

We can write x_2 as $x_2 = r_1 \cos\theta_2$ with $r_1 = r\sin\theta_1$, and we can obtain $\mathbb{E}[|x_2|]$ by using Eq. (16) recursively, *i.e.*, $\mathbb{E}[|x_2|] = \int_{-\mathbb{E}[r_1]}^{\mathbb{E}[r_1]} |x_2| p(x_2)\, \mathrm{d}x_2 = \frac{2}{\pi}\mathbb{E}[r_1] = (\frac{2}{\pi})^2 r$. Finally, the expected value $\mathbb{E}[\|\mathbf{x}\|_1]$ becomes

$$\begin{aligned}\mathbb{E}[\|\mathbf{x}\|_1] &= \mathbb{E}[|x_1|] + \mathbb{E}[|x_2|] + \ldots + \mathbb{E}[|x_{n-1}|] + \mathbb{E}[|x_n|] \\ &= \tfrac{2}{\pi} r + \left(\tfrac{2}{\pi}\right)^2 r + \ldots + \left(\tfrac{2}{\pi}\right)^{n-1} r + \left(\tfrac{2}{\pi}\right)^{n-1} r \\ &= \sum_{i=1}^{n-1} \left(\tfrac{2}{\pi}\right)^i r + \left(\tfrac{2}{\pi}\right)^{n-1} r = \tfrac{\frac{2}{\pi}+(\frac{2}{\pi})^{n-1}-2(\frac{2}{\pi})^n}{1-\frac{2}{\pi}} r. \end{aligned}$$

We thus have an expected bound of $\left\|\mathbf{A}\widetilde{\mathbf{x}}^{(t)} - \mathbf{b}\right\|_1 \leq \frac{\frac{2}{\pi}+(\frac{2}{\pi})^{n-1}-2(\frac{2}{\pi})^n}{1-\frac{2}{\pi}} (1+\varepsilon)\, \eta^{-1} \left\|\mathbf{A}\mathbf{x}^{*(t)} - \mathbf{b}\right\|_1$. With $n \to \infty$, the fraction expression converges to $\frac{2}{\pi-2} \simeq 1.75$. The expected value is considerably smaller than the worst case value $\sqrt{n}$, indicating that when n is large, Sec. 4.2's worst case bound is hardly relevant. Although it is still far from a strict bound due to the dependency on η, as we will see in the following sections, sketched IRLS yields highly accurate approximations in practice in various settings.

5 Performance evaluation

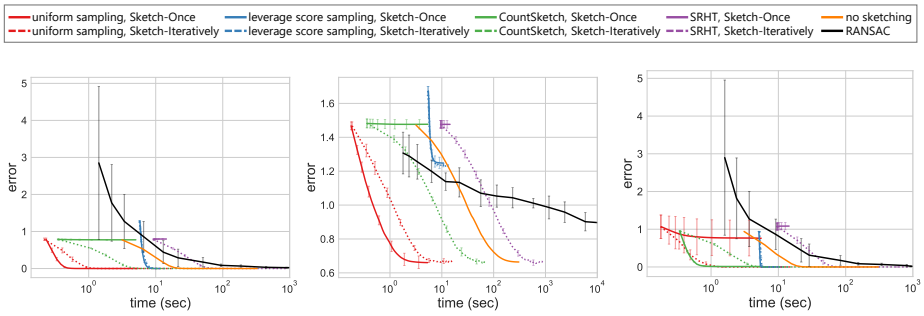
We first evaluate the proposed method's performance in comparison to canonical IRLS using synthetic data in common problems, namely, in residual minimization and low-rank approximation.

5.1 Residual minimization

In ℓ_1 residual minimization, given $\mathbf{A} \in \mathbb{R}^{n\times d}$, $n > d$ and $\mathbf{b} \in \mathbb{R}^n$, we wish to solve

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_1$$

for $\mathbf{x} \in \mathbb{R}^d$. To form a highly over-determined linear system, we set the size of matrix $\mathbf{A}$ to $(n, d) = (10^6, 40)$. For assessing the performance variations with respect to the outlier distribution, we formed matrix $\mathbf{A}$ in three distinct ways: (a) $\mathbf{A}$

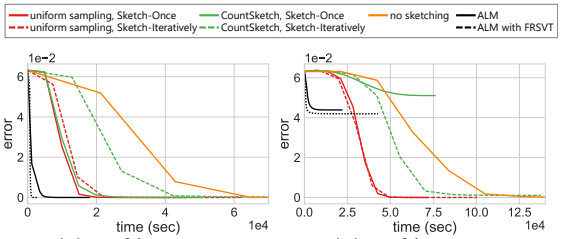


(a) *Uniform data 20%* (b) *Uniform data 60%* (c) *Biased data*

Fig. 2: Averages of the error $\frac{1}{d}\|\mathbf{x}^* - \mathbf{x}^{(t)}\|_2$ and the standard deviations over time in residual minimization with various matrix sketching methods and RANSAC on uniform synthetic data with a lower outlier rate (*left*), with a higher outlier rate (*center*), and with biased synthetic data (*right*).

drawn from a uniform distribution in $[0, 10]$ and flipped signs of 20 % elements to create outliers (called *uniform data 20%*, hereafter), (b) $\mathbf{A}$ created like (a) but with 60 % outliers (called *uniform data 60%*, hereafter), and (c) $\mathbf{A}$ created like (a) but further corrupted by adding a large value ($= 10^3$) to randomly selected 0.1 % of rows (called *biased data*, hereafter). The ground truth $\mathbf{x}^*$ is created, and based on $\mathbf{x}^*$, $\mathbf{b}$ is pre-computed before adding outliers. We evaluate the *error* defined as $\frac{1}{d}\|\mathbf{x}^* - \mathbf{x}^{(t)}\|_2$, where $\mathbf{x}^{(t)}$ is the solution after the t^{th} iteration. We also compare the accuracy with RANSAC [25] to assess the robustness of ℓ_1 minimization against outliers. Unlike ℓ_1 minimization, RANSAC requires adjusting a few parameters. We changed the number of samplings, RANSAC's most important parameter, and chose the best result from $\{d+1, d\times 10, d\times 10^2\}$.

Figures 2a, 2b and 2c show the results of *uniform data 20%*, *uniform data 60%*, and *biased data*, resp. All results are averages of 10 trials with different random seeds. The plots show averages of the *error*, error bars indicate standard deviations. From Figs. 2a and 2b we can observe that ℓ_1 minimization works well for problems with relatively few outliers, and RANSAC shows slow convergences. Both methods need to solve least squares minimization many times, IRLS can be expected to improve the solution for each loop, but RANSAC does not necessarily do so. RANSAC is further known to fail at high-dimensional problems. Regarding sketching methods, while sampling-based methods with sketch-once work well for *uniform data* as shown in Fig. 2a, their accuracies become lower and variances become larger on the biased data as shown in Fig. 2c. Projection-based methods work well for either data with low variances, especially CountSketch with sketch-iteratively. For leverage score sampling, the comparison was done using the known leverage scores while it is usually unknown a priori. From here on out, we will focus only on uniform sampling and CountSketch as the chosen sketching methods because of their efficiency as discussed in Sec. 3.2.



(a) 10 % outliers (b) 40 % outliers

Fig. 3: Error over time in ℓ_1 singular value decomposition on synthetic data with lower outlier rate, (*left*), with higher outlier rate (*right*)

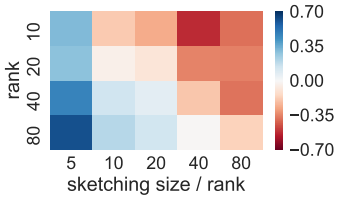


Fig. 4: Comparison between sketch-iteratively and sketch-once. Colors indicate faster convergence for **sketch-iteratively** or **sketch-once**.

5.2 Low-rank approximation

Next, we evaluate the proposed method on low-rank approximation with ℓ_1 singular value decomposition. Given a matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$, its rank-$r$ approximation ($r < \min(m, n)$) with ℓ_1-norm can be written as

$$\min_{\mathbf{U},\mathbf{V}} \|\mathbf{M} - \mathbf{U}\mathbf{V}^\top\|_1,$$

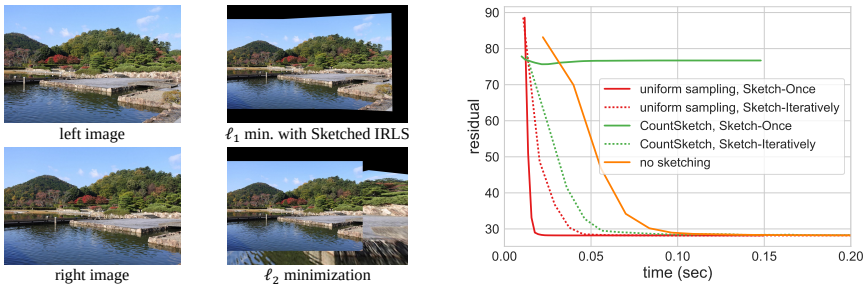
where $\mathbf{U}$ and $\mathbf{V}^\top$ are $m \times r$ and $r \times n$ matrices, respectively. The solution method involves ℓ_1 minimization subproblems that are computed iteratively [34] as

$$\mathbf{U} \leftarrow \operatorname*{argmin}_{\mathbf{U}} \|\mathbf{M} - \mathbf{U}\mathbf{V}^\top\|_1, \quad \mathbf{V} \leftarrow \operatorname*{argmin}_{\mathbf{V}} \|\mathbf{M} - \mathbf{U}\mathbf{V}^\top\|_1,$$

starting from a random initialization of $\mathbf{V}$.

We generated an $\mathbf{M} \in \mathbb{R}^{10^4 \times 10^4}$ with $\operatorname{rank}(\mathbf{M}) = 40$, flipped signs of 10 % and 40 % of the elements. We set the sketching size s as $1,600$ and $3,200$. We also compare the accuracy with robust principal component analysis (R-PCA) [6]. R-PCA decomposes a corrupted matrix $\mathbf{M}$ into a low-rank matrix $\mathbf{A}$ and a sparse corruption matrix $\mathbf{E}$, *i.e.*, $\mathbf{M} = \mathbf{A} + \mathbf{E}$. To solve R-PCA, the Augmented Lagrange Multiplier method (ALM) [37] is a known effective approach. Also a previous study accelerated ALM by fast randomized singular value thresholding (FRSVT) [45]. We used native ALM and ALM with FRSVT as comparison. These methods also require tuning hyper-parameter λ, and we chose the best result from $\lambda \in \{10^{-1}, 10^{-2}, 10^{-3}\}$.

To assess the accuracy, we define the *error* as $\frac{1}{mn}\|\mathbf{M}^* - \mathbf{M}^{(t)}\|_F$, where $\mathbf{M}^*$ is the uncorrupted original matrix and $\mathbf{M}^{(t)}$ is the estimated low-rank matrix after the t^{th} iteration. Figures 3a and 3b show the results. In this setting, ℓ_1 minimization achieves high accuracy in both datasets. R-PCA converges quickly but does not work well for the dataset with many outliers. The sketch-once strategy shows about 3 times faster convergence compared to canonical IRLS in Fig. 3a, and also uniform sampling strategy shows fast convergence in Fig. 3b.



(a) *Left:* Input images. *Right:* ℓ_1 and ℓ_2 stitching results.

(b) ℓ_1 residual variation over time

Fig. 5: Performance of image stitching with ℓ_1 and ℓ_2 homography estimation

5.3 When to sketch?

In the following, we show an experiment that serves to give a first intuition for when a user should pick the sketch-once or the sketch-iteratively sketching regime. We generated $\mathbb{R}^{10^4 \times 10^4}$ matrices with 10 % noise, we picked the ranks from $\{10, 20, 40, 80\}$ and sketching sizes from $\{\text{rank} \times 5, \times 10, \times 20, \times 40, \times 80\}$, and conducted a low-rank approximation experiment. We checked the times t_{SO} and t_{SI} until sketch-once and sketch-iteratively achieved an accuracy of 10^{-5}. Figure 4 shows the values of $\ln(t_{SO}/t_{SI})$. For each rank, the larger the sketching size, the faster sketch-once converges. As the matrix rank increases, sketch-iteratively shows faster convergence at larger sketching sizes. The sketching size for ideal approximations increases faster than $O(r)$. When the sketching size is not big enough, the risk of not making ideal approximations becomes higher. Especially, at small sketching sizes, sketch-once is susceptible to making bad approximations, whereas sketch-iteratively alleviates this effect by repeated sketching and shows faster convergence for smaller sketching sizes.

6 Applications

Countless computer vision tasks are built upon robust regression. In practical scenarios, the input is typically quite large and noisy, either due to a noisy capturing process or a noisy precomputation step, *e.g.*, incorrect feature matches. In this section, we demonstrate the proposed sketched IRLS's effectiveness on such real-world applications. We adapted our method to problems with over-determined system: homography estimation and point cloud alignment.

6.1 Homography estimation

Homography estimation is an important building block for image stitching and plane-to-plane alignment. Given correspondences in two images, the image trans-

formation can be written by a 3×3 homography $\mathbf{H}$ with 8 DoF as

$$\lambda[x', y', 1]^\top = \mathbf{H}[x, y, 1]^\top,$$

where λ is a scaling factor, and (x, y) and (x', y') are corresponding points in the left and right images, respectively [50]. Given a set of correspondences, one can estimate $\mathbf{H}$ by solving a homogeneous system [51, Chap. 6.1]. The set of correspondences may contain many wrong correspondences from erroneous feature matching. Therefore, conventional methods use robust estimation techniques, such as RANSAC or ℓ_1 regression.

Result We obtained point correspondences by matching AKAZE [4] features in the two images shown in the left column of Fig. 5a. From $17,076$ obtained point correspondences we estimated the homography. The second column of Fig. 5a shows the stitching results of ℓ_1 and ℓ_2 minimization, respectively. ℓ_1 minimization successfully joined the two images whereas ℓ_2 minimization produced a strongly erroneous result due to feature mismatches. In this experiment, we sketched the matrix to $17,076\times2\times1\,\% \simeq 341$ equations (note that each point pair gives two equations), and confirmed that the solution did not differ from the direct ℓ_1 minimization without sketching. In Fig. 5b, we can see that the proposed method (uniform sampling + sketch-once) converges about $5\times$ faster than canonical IRLS while maintaining the same accuracy.

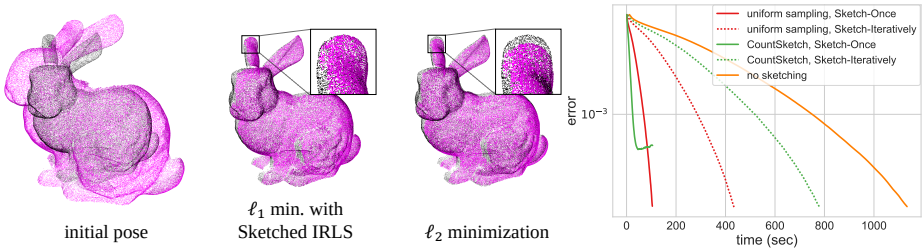
6.2 Point cloud alignment

Consider the two point sets $\mathbf{P} = [\mathbf{p}_1, \dots, \mathbf{p}_l]$ and $\mathbf{Q} = [\mathbf{q}_1, \dots, \mathbf{q}_l]$ captured from different viewpoints around an object. For each i, the points $\mathbf{p}_i \in \mathbb{R}^3$ and $\mathbf{q}_i \in \mathbb{R}^3$ approximately correspond to the same surface point on the 3D object. Since $\mathbf{P}$ may be positioned and scaled differently in space than $\mathbf{Q}$, we search for a similarity transform $\mathbf{T}$ that, applied to all points in $\mathbf{P}$, makes $\mathbf{P}$ and $\mathbf{Q}$ roughly coincide. We therefore optimize

$$\min_{\mathbf{T}} \sum_i \left\| \mathbf{T}\tilde{\mathbf{p}}_i - \tilde{\mathbf{q}}_i \right\|, \tag{17}$$

with the tilde denoting homogeneous representations. This problem is commonly solved with ℓ_2 minimization but with large outliers, *e.g.*, due to wrong point correspondences, ℓ_1 minimization may be superior. In the veteran but still frequently used Iterative Closest Point (ICP) algorithm [10,5], the point sets are first manually pre-aligned, then it searches for the nearest neighbor in $\mathbf{Q}$ for each point in $\mathbf{P}$, optimize Eq. (17), and iterate.

Result In this task, we used the Stanford Bunny [55]. The two input point clouds are shown in Fig. 6a (left). Each set consists of 10^5 points, viewed from different viewpoints in different scalings. We applied the proposed sketched IRLS to perform ICP with ℓ_1 residual minimization. The second and third figures of Fig. 6a show the results of ℓ_1 minimization with sketched IRLS and conventional ℓ_2 minimization. It is observed that ℓ_1 minimization results in an accurate alignment of the two point clouds, unaffected by inaccurate correspondences.



(a) Input point clouds and alignment results (b) Error variation over time

Fig. 6: Performance of point cloud alignment via ICP with ℓ_1 and ℓ_2 similarity transform estimation

For this experiment we set the sketching size s as 10 % of the original problem, and gradually transformed the point set $\mathbf{Q}$ with fixed $\mathbf{P}$. Since we have the ground truth here, we evaluate the error by $\frac{1}{l}\|\mathbf{Q}^* - \mathbf{Q}^{(t)}\|_F$, where $\mathbf{Q}^{(t)}$ is the result after the t^{th} iteration. The evaluation of the error variation *w.r.t.* time only for computing the transformation (excluding matchings) is summarized in Fig. 6b. While CountSketch + sketch-once did not show good convergence, the other sketching methods find a good alignment with significant speed up compared to the conventional method.

7 Discussion

Our experiments showed that sketching is effective in reducing IRLS's runtime and ℓ_1 minimization with IRLS works well on a wide range of computer vision problems. Other robust methods such as RANSAC and RPCA are certainly good at specific problems, but IRLS ℓ_1 minimization is versatile without requiring parameter tuning and its convergence is demonstrably superior in some tasks.

Regarding when to sketch, sketch-once is superior if the rank r of the design matrix $\mathbf{A}$ is very small and the sketching is not aggressive, *i.e.*, $s \gg r$. However, if the rank is high or we sketch aggressively, *e.g.*, $s < 5r$, then it is likely that $\mathbf{A}$'s range will not be preserved, and we need to perform sketch-iteratively to be able to recover from badly chosen samples.

If one naïvely performs subsampling on the input data, this would be equal to sketch-once sketching, basically treating IRLS as a black box that can never be touched again after the data has initially been subsampled. The experiments showed that in applications where sketch-iteratively performs better, we want to open that black box and perform subsampling in every iteration.

Acknowledgments: This work is supported by JSPS CREST Grant Number JPMJCR1764, Japan. Michael Waechter was supported through a postdoctoral fellowship by the Japan Society for the Promotion of Science (JP17F17350).

References

1. Aftab, K., Hartley, R.: Convergence of iteratively re-weighted least squares to robust M-estimators. In: Winter Conference on Applications of Computer Vision (WACV) (2015) 1, 3
2. Aftab, K., Hartley, R., Trumpf, J.: Generalized Weiszfeld algorithms for Lq optimization. Transactions on Pattern Analysis and Machine Intelligence (PAMI) **37**(4), 728–745 (2015) 5
3. Ailon, N., Chazelle, B.: Approximate nearest neighbors and the fast Johnson-Lindenstrauss transform. In: Symposium on Theory of Computing (2006) 4
4. Alcantarilla, P.F., Nuevo, J., Bartoli, A.: Fast explicit diffusion for accelerated features in nonlinear scale spaces. In: British Machine Vision Conference (BMVC) (2013) 13
5. Besl, P.J., McKay, N.D.: A method for registration of 3-D shapes. Transactions on Pattern Analysis and Machine Intelligence (PAMI) **14**(2), 239–256 (1992) 13
6. Candès, E.J., Li, X., Ma, Y., Wright, J.: Robust principal component analysis? Journal of ACM **58**(3), 11:1–11:37 (2011) 11
7. Candes, E.J., Tao, T.: Decoding by linear programming. IEEE Transactions on Information Theory **51**(12), 4203–4215 (2005) 7
8. Charikar, M., Chen, K., Farach-Colton, M.: Finding frequent items in data streams. In: International Colloquium on Automata, Languages and Programming (2002) 4
9. Chen, C., He, L., Li, H., Huang, J.: Fast iteratively reweighted least squares algorithms for analysis-based sparse reconstruction. Medical Image Analysis **49**, 141 – 152 (2018) 3
10. Chen, Y., Medioni, G.: Object modeling by registration of multiple range images. In: International Conference on Robotics and Automation (ICRA) (1991) 13
11. Clarkson, K.L., Woodruff, D.P.: Low rank approximation and regression in input sparsity time. In: Symposium on Theory of Computing (2013) 2, 4, 6
12. Cohen, M.B., Kyng, R., Miller, G.L., Pachocki, J.W., Peng, R., Rao, A.B., Xu, S.C.: Solving SDD linear systems in nearly $m \log^{1/2} n$ time. In: Symposium on Theory of Computing (2014) 4
13. Daubechies, I., Defrise, M., De Mol, C.: An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. Communications on Pure and Applied Mathematics **57**(11), 1413–1457 (2004) 3
14. Daubechies, I., DeVore, R., Fornasier, M., Güntürk, C.S.: Iteratively reweighted least squares minimization for sparse recovery. Communications on Pure and Applied Mathematics **63**(1), 1–38 (2010) 3, 7
15. Donoho, D.L., Tsaig, Y., Drori, I., Starck, J.L.: Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit. Transactions on Information Theory **58**(2), 1094–1121 (2012) 3
16. Drineas, P., Mahoney, M.W.: Randnla: Randomized numerical linear algebra. Communications of the ACM **59**(6), 8090 (2016) 4
17. Drineas, P., Mahoney, M.W., Muthukrishnan, S.: Sampling algorithms for ℓ_2 regression and applications. In: ACM-SIAM Symposium on Discrete Algorithm (2006) 4, 7
18. Drineas, P., Mahoney, M.W., Muthukrishnan, S., Sarlós, T.: Faster least squares approximation. Numerische Mathematik **117**(2), 219–249 (2011) 4
19. Duersch, J., Gu, M.: Randomized QR with column pivoting. SIAM Journal on Scientific Computing **39**(4), 263–291 (2017) 4

20. Dutta, A., Richtárik, P.: Online and batch supervised background estimation via l1 regression. In: Winter Conference on Applications of Computer Vision (WACV) (2019) 3
21. Efron, B., Hastie, T., Johnstone, I., Tibshirani, R.: Least angle regression. Annals of Statistics **32**(2), 407–499 (2004) 3
22. Elad, M.: Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing. Springer, 1st edn. (2010) 3
23. Elad, M., Aharon, M.: Image denoising via sparse and redundant representations over learned dictionaries. Transactions on Image Processing **15**(12), 3736–3745 (2006) 3
24. Erichson, N., Mathelin, L., Brunton, S., Kutz, J.: Randomized dynamic mode decomposition. SIAM Journal on Applied Dynamical Systems **18**(4), 1867–1891 (2017) 4
25. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24**(6), 381–395 (1981) 10
26. Gentle, J.E.: Matrix Algebra: Theory, Computations, and Applications in Statistics. Springer (2007) 3
27. Goldstine, H.H., von Neumann, J.: Numerical inverting of matrices of high order. ii. In: American Mathematical Society (1951) 4
28. Gorodnitsky, I.F., Rao, B.D.: Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm. Transactions on Signal Processing **45**(3), 600–616 (1997) 3
29. Gower, R., Richtárik, P.: Randomized iterative methods for linear systems. SIAM Journal on Matrix Analysis and Applications **36**(4), 1660–1690 (2015) 4
30. Gower, R., Richtrik, P.: Randomized quasi-Newton updates are linearly convergent matrix inversion algorithms. SIAM Journal on Matrix Analysis and Applications **38**(4), 1380–1409 (2016) 4
31. Halko, N., Martinsson, P.G., Tropp, J.A.: Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. SIAM Rev. **53**(2), 217–288 (2011) 2, 4
32. Johnson, W.B., Lindenstrauss, J.: Extensions of Lipschitz mappings into a Hilbert space. Contemporary Mathematics **26**, 189–206 (1984) 4
33. Karmarkar, N.: A new polynomial-time algorithm for linear programming. Combinatorica **4**(4), 373–395 (1984) 3
34. Ke, Q., Kanade, T.: Robust L_1 norm factorization in the presence of outliers and missing data by alternative convex programming. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2005) 11
35. Kekatos, V., Giannakis, G.B.: From sparse signals to sparse residuals for robust sensing. IEEE Transactions on Signal Processing **59**(7), 3355–3368 (2011) 3
36. Kiani, K.A., Drummond, T.: Solving robust regularization problems using iteratively re-weighted least squares. In: Winter Conference on Applications of Computer Vision (WACV) (2017) 1
37. Lin, Z., Chen, M., Wu, L., Ma, Y.: The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices. Tech. Rep. UILU-ENG-09-2215, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign (2010) 11
38. Lu, C., Lin, Z., Yan, S.: Smoothed low rank and sparse matrix recovery by iteratively reweighted least squares minimization. Transactions on Image Processing **24**(2), 646–654 (2015) 3

39. Mahoney, M.W.: Randomized algorithms for matrices and data. Foundations and Trends® in Machine Learning **3**(2), 123–224 (2011) 2, 4, 7
40. Mallat, S.G.: A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way. Academic Press, Inc., 3rd edn. (2008) 3
41. Mallat, S.G., Zhang, Z.: Matching pursuits with time-frequency dictionaries. Transactions on Signal Processing **41**(12), 3397–3415 (1993) 3
42. Millikan, B., Dutta, A., Rahnavard, N., Sun, Q., Foroosh, H.: Initialized iterative reweighted least squares for automatic target recognition. In: Military Communications Conference (2015) 3
43. Needell, D., Tropp, J.A.: CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. Applied and Computational Harmonic Analysis **26**(3), 301–321 (2009) 3
44. Ochs, P., Dosovitskiy, A., Brox, T., Pock, T.: On iteratively reweighted algorithms for non-smooth non-convex optimization in computer vision. SIAM Journal on Imaging Sciences **8**(1), 331–372 (2015) 4
45. Oh, T.H., Matsushita, Y., Tai, Y.W., Kweon, I.S.: Fast randomized singular value thresholding for low-rank optimization. Transactions on Pattern Analysis and Machine Intelligence (PAMI) **40**(2), 376–391 (2018) 11
46. Osborne, M.R.: Finite Algorithms in Optimization and Data Analysis. John Wiley & Sons, Inc. (1985) 3
47. Pati, Y.C., Rezaiifar, R., Krishnaprasad, P.S.: Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In: Asilomar Conference on Signals, Systems, and Computers (1993) 3
48. Sarlós, T.: Improved approximation algorithms for large matrices via random projections. In: Symposium on Foundations of Computer Science. pp. 143–152 (2006) 7
49. Sigl, J.: Nonlinear residual minimization by iteratively reweighted least squares. Computational Optimization and Applications **64**(3), 755–792 (2016) 4, 7
50. Szeliski, R.: Video mosaics for virtual environments. Computer Graphics and Applications **16**(2), 22–30 (1996) 13
51. Szeliski, R.: Computer Vision - Algorithms and Applications. Texts in Computer Science, Springer (2011) 13
52. Tropp, J.A.: Greed is good: Algorithmic results for sparse approximation. Transactions on Information Theory **50**(10), 2231–2242 (2004) 3
53. Tropp, J.A.: Improved analysis of the subsampled randomized Hadamard transform. Advances in Adaptive Data Analysis **3**, 115–126 (2011) 4
54. Tropp, J.A., Gilbert, A.C., Strauss, M.J.: Algorithms for simultaneous sparse approximation. Part I. Signal Process. **86**(3), 572–588 (2006) 3
55. Turk, G., Levoy, M.: Zippered polygon meshes from range images. In: SIGGRAPH (1994) 13
56. Weiszfeld, E.: Sur le point pour lequel la somme des distances de n points donnés est minimum. Tohoku Mathematical Journal (1937) 5
57. Weiszfeld, E., Plastria, F.: On the point for which the sum of the distances to n given points is minimum. Annals of Operations Research (2009) 5
58. Woodruff, D.P.: Sketching as a tool for numerical linear algebra. Foundations and Trends® in Theoretical Computer Science **10**(1-2), 1–157 (2014) 2, 4
59. Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y.: Robust face recognition via sparse representation. Transactions on Pattern Analysis and Machine Intelligence (PAMI) **31**(2), 210–227 (2009) 3

60. Yang, A., Zhou, Z., Ganesh Balasubramanian, A., Sastry, S., Ma, Y.: Fast L1-minimization algorithms for robust face recognition. Transactions on Image Processing **22**(8), 3234–3246 (2013) 3
61. Zhang, Z., Xu, Y., Yang, J., Li, X., Zhang, D.: A survey of sparse representation: Algorithms and applications. IEEE Access **3**, 490–530 (2015) 3