Weighted SGD for ℓ_p Regression with Randomized Preconditioning

Jiyan Yang

Stanford University

SODA, Jan 2016

Joint work with Yin-Lam Chow (Stanford), Christopher Ré (Stanford) and Michael W. Mahoney (Berkeley)

Outline

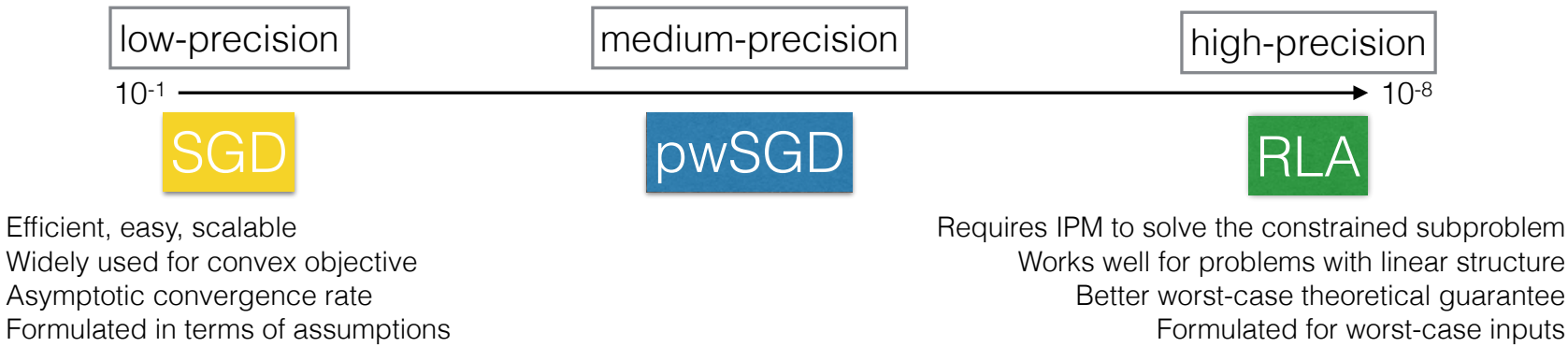
Problem formulation

Definition

Given a matrix $A \in \mathbb{R}^{n \times d}$, where $n \gg d$, a vector $b \in \mathbb{R}^n$, and a number $p \in [1, \infty]$, the *constrained overdetermined ℓ_p regression problem* is

$$\min_{x \in \mathcal{Z}} \|Ax - b\|_p.$$

Overview



pwSGD — Preconditioned weighted SGD

- It preserves the simplicity of SGD and the high quality theoretical guarantees of RLA.
- It is preferable when a medium-precision, e.g., 10^{-3} is desired.

Our main algorithm: pwSGD

Algorithm

1. Apply RLA techniques to construct a preconditioner and then construct an importance sampling distribution.
2. Apply an SGD-like iterative phase with weighted sampling on the preconditioned system.

Properties

- The preconditioner and the importance sampling distribution can be done as fast as in $\mathcal{O}(\log n \cdot \text{nnz}(A))$.
- The convergence rate of the SGD phase only depends on the low dimension d, independent of the high dimension n.

Complexity comparisons

solver	complexity (general)	complexity (sparse)
RLA sampling	$time(R) + \mathcal{O}(\text{nnz}(A)\log n + \bar{\kappa}_1^{\frac{3}{2}} d^{\frac{9}{2}}/\epsilon^3)$	$\mathcal{O}(\text{nnz}(A)\log n + d^{\frac{69}{8}} \log^{\frac{25}{8}} d/\epsilon^{\frac{5}{2}})$
randomized IPCPM	$time(R) + nd^2 + \mathcal{O}((nd + \text{poly}(d))\log(\bar{\kappa}_1 d/\epsilon))$	$\mathcal{O}(nd\log(d/\epsilon))$
pwSGD	$time(R) + \mathcal{O}(\text{nnz}(A)\log n + d^3\bar{\kappa}_1/\epsilon^2)$	$\mathcal{O}(\text{nnz}(A)\log n + d^{\frac{13}{2}} \log^{\frac{5}{2}} d/\epsilon^2)$

Table: Summary of complexity of several unconstrained ℓ_1 solvers that use randomized linear algebra. Clearly, pwSGD has a uniformly better complexity than that of RLA sampling methods in terms of both d and ϵ, no matter which underlying preconditioning method is used.

solver	complexity (SRHT)	complexity (sparse)
RLA projection	$\mathcal{O}\left(nd\log(d/\epsilon) + d^3\log(nd)/\epsilon\right)$	$\mathcal{O}\left(\text{nnz}(A) + d^4/\epsilon^2\right)$
RLA sampling	$\mathcal{O}\left(nd\log n + d^3\log d + d^3\log d/\epsilon\right)$	$\mathcal{O}\left(\text{nnz}(A)\log n + d^4 + d^3\log d/\epsilon\right)$
RLA high-precision solvers	$\mathcal{O}\left(nd\log d + d^3\log d + nd\log(1/\epsilon)\right)$	$\mathcal{O}\left(\text{nnz}(A) + d^4 + nd\log(1/\epsilon)\right)$
pwSGD	$\mathcal{O}\left(nd\log n + d^3\log d + d^3\log(1/\epsilon)/\epsilon\right)$	$\mathcal{O}\left(\text{nnz}(A)\log n + d^4 + d^3\log(1/\epsilon)/\epsilon\right)$

Table: Summary of complexity of several unconstrained ℓ_2 solvers that use randomized linear algebra. When $d \geq 1/\epsilon$ and $n \geq d^2/\epsilon$, pwSGD is asymptotically better than the solvers listed above.

Outline

Viewing ℓ_p regression as stochastic optimization

$$\min_{x\in\mathcal{Z}} \|Ax-b\|_p^p \xrightarrow{(a)} \min_{y\in\mathcal{Y}} \|Uy-b\|_p^p \xrightarrow{(b)} \min_{y\in\mathcal{Y}} \mathbb{E}_{\xi\sim P}\left[H(y,\xi)\right].$$

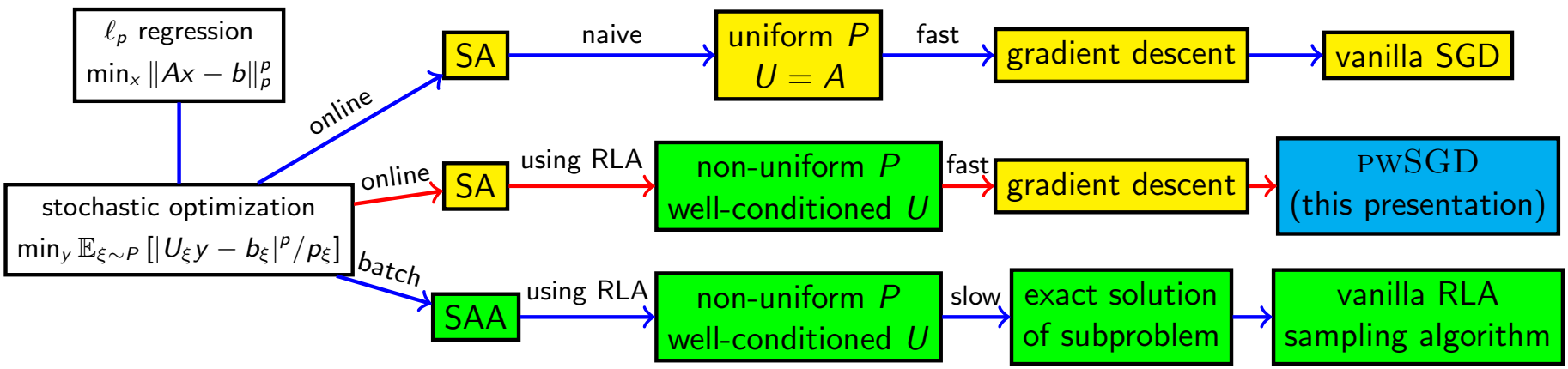
- (a) is done by using a different basis U.
- (b) is true for any sampling distribution $\{p_i\}_{i=1}^n$ over the rows by setting $H(y,\xi) = \frac{|U_\xi y - b_\xi|^p}{p_\xi}$.

Solving ℓ_p regression via stochastic optimization

To solve this stochastic optimization problem, typically one needs to answer the following three questions.

- *(C1): How to sample: SAA (i.e., draw samples in a batch mode and deal with the subproblem) or SA (i.e., draw a mini-batch of samples in an online fashion and update the weight after extracting useful information)?*
- *(C2): Which probability distribution P (uniform distribution or not) and which basis U (preconditioning or not) to use?*
- *(C3): Which solver to use (e.g., how to solve the subproblem in SAA or how to update the weight in SA)?*

A unified framework for RLA and SGD



Outline

ℓ_p Well-conditioned basis

Definition (ℓ_p-norm condition number (Clarkson et al., 2013))

Given a matrix $A \in \mathbb{R}^{m \times n}$ and $p \in [1, \infty]$, let

$$\sigma_p^{\max}(A) = \max_{\|x\|_2 = 1} \|Ax\|_p \text{ and } \sigma_p^{\min}(A) = \min_{\|x\|_2 = 1} \|Ax\|_p.$$

Then, we denote by $\kappa_p(A)$ the *ℓ_p-norm condition number of A*, defined to be:

$$\kappa_p(A) = \sigma_p^{\max}(A) / \sigma_p^{\min}(A).$$

Motivation

- For ℓ_2, a perfect preconditioner is the one that transforms A into an orthonormal basis.
- However, doing such requires factorizations like QR and SVD of A which is expensive.
- Can we do QR on a similar but much smaller matrix?
- Idea: we use randomized linear algebra to compute a sketch and perform QR on it.

An important tool: sketch

- Given a matrix $A \in \mathbb{R}^{n \times d}$, a sketch can be viewed as a compressed representation of A, denoted by ΦA.
- The matrix $\Phi \in \mathbb{R}^{r \times n}$ preserves the norm of vectors in the range space of A up to small constants. That is,

$$(1 - \epsilon)\|Ax\| \leq \|\Phi Ax\| \leq (1 + \epsilon)\|Ax\|, \quad \forall x \in \mathbb{R}^d.$$

- $r \ll n$.

Type of ℓ_2 sketches

- Sub-Gaussian sketch
  e.g., Gaussian transform: $\Phi A = GA$
  time: $\mathcal{O}(nd^2)$, $r = \mathcal{O}(d/\epsilon^2)$
- Sketch based on randomized orthonormal systems [Tropp, 2011]
  e.g., Subsampled randomized Hadamard transform (SRHT): $\Phi A = SDHA$
  time: $\mathcal{O}(nd \log n)$, $r = \mathcal{O}(d \log(nd) \log(d/\epsilon^2)/\epsilon^2)$
- Sketch based on sparse transform [Clarkson and Woodruff, 2013]
  e.g., count-sketch like transform (CW): $\Phi A = RDA$
  time: $\mathcal{O}(\texttt{nnz}(A))$, $r = (d^2 + d)/\epsilon^2$
- Sampling with approximate leverage scores [Drineas et al., 2012]
  Leverage scores can be viewed as a measurement of the importance of the rows in the LS fit.
  e.g., using CW transform to estimate the leverage scores
  time: $\texttt{t}_{\texttt{proj}} + \mathcal{O}(\texttt{nnz}(A)) \log n$, $r = \mathcal{O}(d \log d/\epsilon^2)$

Normally, when ϵ is fixed, the required sketching size r only depends on d, independent of n.

Randomized preconditioners

Algorithm

1. Compute a sketch ΦA.
2. Compute the economy QR factorization of $\Phi A = QR$.
3. Return R^{-1}.

Randomized preconditioners (cont')

Analysis

- Since A and ΦA are "similar", $AR^{-1} \approx \Phi AR^{-1} = Q$.
- Using norm preservation property of the sketch and norm equivalence, we have

$$\begin{aligned}\|AR^{-1}x\|_p &\le \|\Phi AR^{-1}x\|_p/\sigma_\Phi \le r^{\max\{0,1/p-1/2\}} \cdot \|\Phi AR^{-1}\|_2 \cdot \|x\|_2/\sigma_\Phi \\ &= r^{\max\{0,1/p-1/2\}} \cdot \|x\|_2/\sigma_\Phi, \quad \forall x \in \mathbb{R}^d,\end{aligned}$$

and

$$\begin{aligned}\|AR^{-1}x\|_p &\ge \|\Phi AR^{-1}\|_p/(\sigma_\Phi\kappa_\Phi) \ge r^{\min\{0,1/p-1/2\}} \cdot \|\Phi AR^{-1}x\|_2/(\sigma_\Phi\kappa_\Phi) \\ &= \sigma_\Phi r^{\min\{0,1/p-1/2\}} \cdot \|x\|_2/(\sigma_\Phi\kappa_\Phi), \quad \forall x \in \mathbb{R}^d.\end{aligned}$$

Qualities of preconditioners

name	running time	$\bar{\kappa}_p(U)$
Dense Cauchy [SW11]	$\mathcal{O}(nd^2 \log d + d^3 \log d)$	$\mathcal{O}(d^{5/2} \log^{3/2} d)$
Fast Cauchy [CDM+12]	$\mathcal{O}(nd \log d + d^3 \log d)$	$\mathcal{O}(d^{11/2} \log^{9/2} d)$
Sparse Cauchy [MM12]	$\mathcal{O}(\text{nnz}(A) + d^7 \log^5 d)$	$\mathcal{O}(d^{\frac{13}{2}} \log^{\frac{11}{2}} d)$
Reciprocal Exponential [WZ13]	$\mathcal{O}(\text{nnz}(A) + d^3 \log d)$	$\mathcal{O}(d^{\frac{7}{2}} \log^{\frac{5}{2}} d)$

Table: Summary of running time and condition number, for several different ℓ_1 conditioning methods.

name	running time	$\kappa_p(U)$	$\bar{\kappa}_p(U)$
sub-Gaussian	$\mathcal{O}(nd^2)$	$\mathcal{O}(1)$	$\mathcal{O}(\sqrt{d})$
SRHT [Tropp11]	$\mathcal{O}(nd \log n + d^3 \log d)$	$\mathcal{O}(1)$	$\mathcal{O}(\sqrt{d})$
Sparse ℓ_2 Embedding [CW12]	$\mathcal{O}(\text{nnz}(A) + d^4)$	$\mathcal{O}(1)$	$\mathcal{O}(\sqrt{d})$

Table: Summary of running time and condition number, for several different ℓ_2 conditioning methods.

Why preconditioning is useful?

We can actually show that the convergence rate of using SGD for ℓ_p regression problem relies on the ℓ_p condition number of the linear system. Using such a randomized preconditioner will drastically reduce the number of iterations needed.

Outline

Main algorithm – pwSGD

1. Use RLA to compute $R \in \mathbb{R}^{d\times d}$ such that $U = AR^{-1}$ is well-conditioned.
2. Compute or estimate $\|U_i\|_p^p$ as λ_i, for $i \in [n]$.
3. Let $q_i = \frac{\lambda_i}{\sum_{j=1}^n \lambda_j}$, for $i \in [n]$.
4. For $t = 1, \ldots, T$

   Pick ξ_t from $[n]$ based on distribution $\{q_i\}_{i=1}^n$.

$$c_t = \begin{cases} \operatorname{sgn}\left(A_{\xi_t} x_t - b_{\xi_t}\right)/q_{\xi_t} & \text{if } p = 1; \\ 2\left(A_{\xi_t} x_t - b_{\xi_t}\right)/q_{\xi_t} & \text{if } p = 2. \end{cases}$$

   Update x by

$$x_{t+1} = \begin{cases} x_t - \eta c_t H^{-1} A_{\xi_t} & \text{if } \mathcal{Z} = \mathbb{R}^d; \\ \arg\min_{x\in\mathcal{Z}} \ \eta c_t A_{\xi_t} x + \frac{1}{2}\|x_t - x\|_H^2 & \text{otherwise.} \end{cases}$$

   where $H = R^\top R$.
5. $\bar{x} \leftarrow \frac{1}{T}\sum_{t=1}^T x_t$.
6. **Return** $\bar{x}$ for $p = 1$ or x_T for $p = 2$.

Main theoretical bound (ℓ_1 Regression)

Let $f(x) = \|Ax - b\|_1$ and suppose $f(x^*) > 0$. Then there exists a step-size η such that after

$$T = d\bar{\kappa}_1^2(U)\frac{c}{\epsilon^2}$$

iterations, pwSGD returns a solution vector estimate $\bar{x}$ that satisfies the expected relative error bound

$$\frac{\mathbb{E}\left[f(\bar{x})\right] - f(x^*)}{f(x^*)} \leq \epsilon.$$

Recall, $\bar{\kappa}_1^2(U)$ only depends on d and thus so does T.

If the sparse sketch is used, the overall complexity becomes $\mathcal{O}(\log n \cdot \text{nnz}(A) + \text{poly}(d)/\epsilon^2)$.

Main theoretical bound (ℓ_2 Regression)

Let $f(x) = \|Ax - b\|_2$ and suppose $f(x^*) > 0$. Then there exists a step-size η such that after

$$T = c_1 \bar{\kappa}_2^2(U) \cdot \log\left(\frac{2c_2\kappa(U)}{\epsilon}\right) \cdot \left(1 + \frac{\kappa^2(U)}{c_3\epsilon}\right)$$

iterations, pwSGD returns a solution vector estimate x_T that satisfies the expected relative error bound

$$\frac{\mathbb{E}\left[\|A(x_T - x^*)\|_2^2\right]}{\|Ax^*\|_2^2} \leq \epsilon.$$

Furthermore, when $\mathcal{Z} = \mathbb{R}^d$, there exists a step-size η such that after

$$T = c_1 \bar{\kappa}_2^2(U) \cdot \log\left(\frac{c_2\kappa(U)}{\epsilon}\right) \cdot \left(1 + \frac{2\kappa^2(U)}{\epsilon}\right)$$

iterations, pwSGD returns a solution vector estimate x_T that satisfies the expected relative error bound

$$\frac{\mathbb{E}\left[f(x_T)\right] - f(x^*)}{f(x^*)} \leq \epsilon.$$

Connection with weighted randomized Kaczmarz algorithm

- Our algorithm pwSGD for least-squares regression is related to the weighted randomized Kaczmarz (RK) algorithm [Strohmer and Vershynin, 2009].
- Roughly speaking, pwSGD is equivalent to applying the weighted randomized Karczmarz algorithm on a well-conditioned basis U.
- Theoretical results indicate that weighted RK algorithm inherits a convergence rate that depends on condition number $\kappa(A)$ times the scaled condition number $\bar{\kappa}_2(A)$.
- The advantage of preconditioning in pwSGD is reflected here since $\kappa(U) \approx 1$ and $\hat{\kappa}_2(U) \approx \sqrt{d}$.

Outline

On datasets with increasing condition number

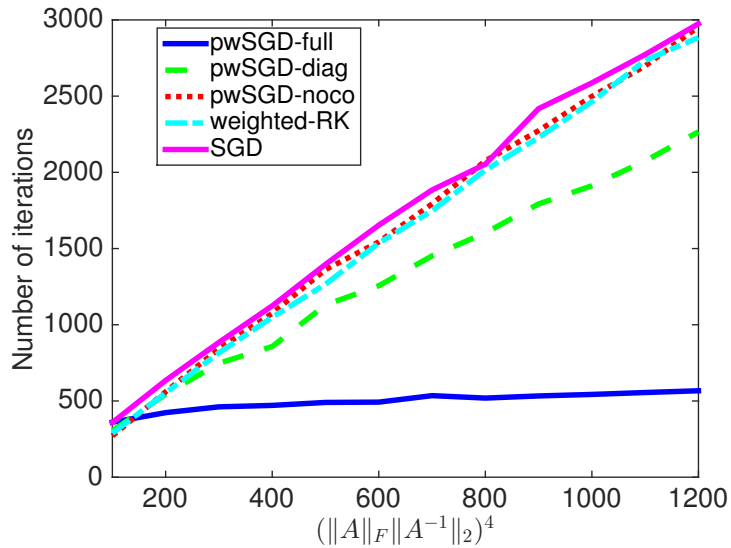
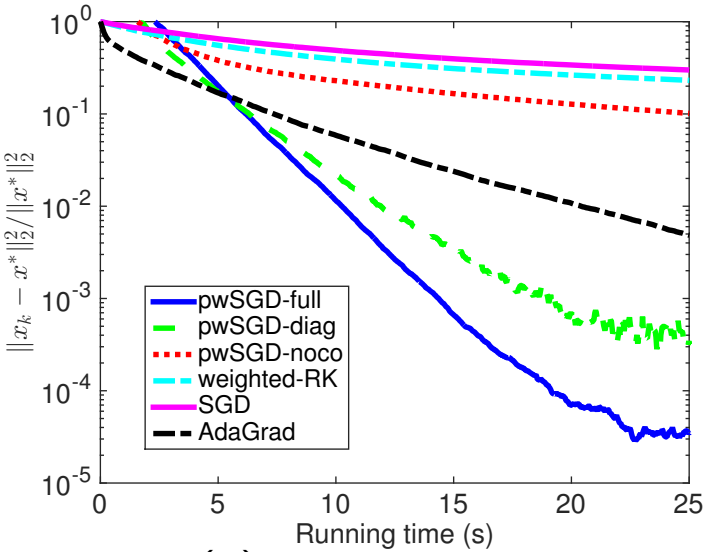
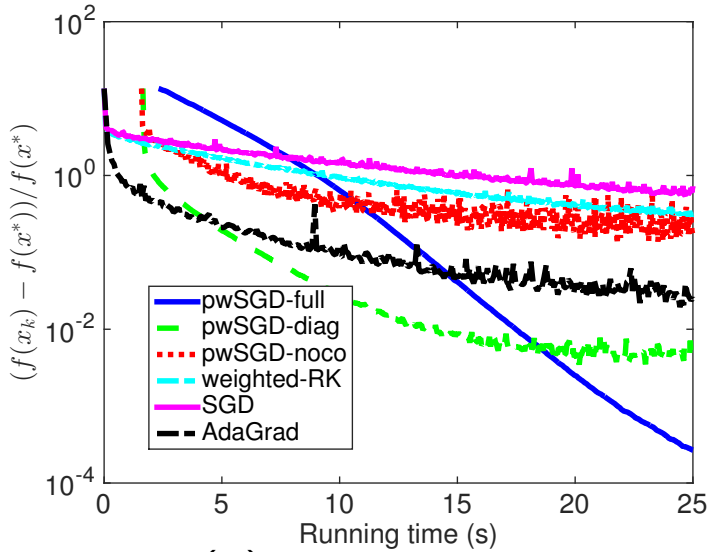


Figure: Convergence rate comparison of several SGD-type algorithms for solving ℓ_2 regression on synthetic datasets with increasing condition number. For each method, the optimal step-size is set according to the theory with target accuracy $|f(\hat{x}) - f(x^*)|/f(x^*) = 0.1$. The y-axis is showing the minimum number of iterations for each method to find a solution with the target accuracy.

Time-accuracy tradeoffs for ℓ_2 regression



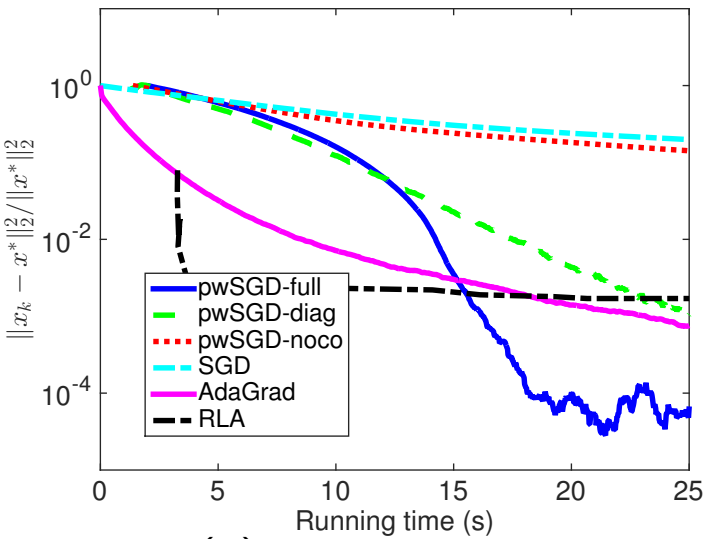
(a) ℓ_2 regression



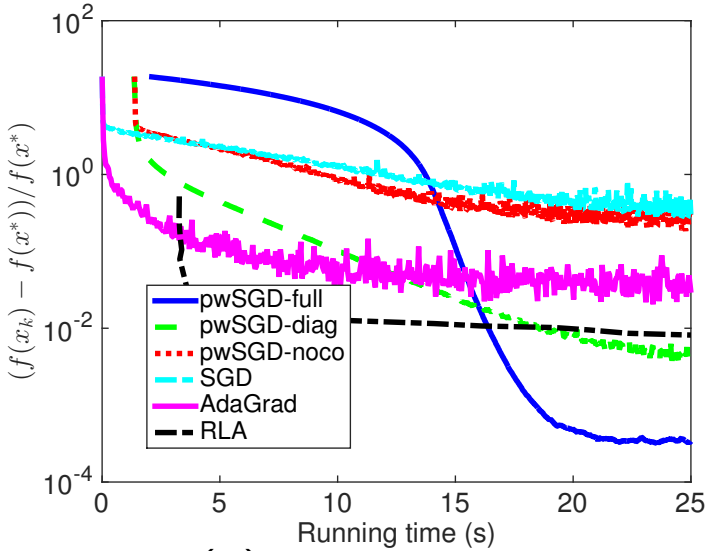
(b) ℓ_2 regression

Figure: Time-accuracy tradeoffs of several algorithms including pwSGD with three different choices of preconditioners on `year` dataset.

Time-accuracy tradeoffs for ℓ_1 regression



(a) ℓ_1 regression



(b) ℓ_1 regression

Figure: Time-accuracy tradeoffs of several algorithms including pwSGD with three different choices of preconditioners on `year` dataset.

Conclusion

- We propose a novel RLA-SGD hybrid algorithm called pwSGD.
- After a preconditioning step and constructing a non-uniform sampling distribution with RLA, its SGD phase inherits fast convergence rates that only depend on the lower dimension of the input matrix.
- We show that for unconstrained ℓ_2 regression, this complexity is asymptotically better than several state-of-the-art solvers in the regime where $d \geq 1/\epsilon$ and $n \geq d^2/\epsilon$.
- For unconstrained ℓ_1 regression. This complexity is *uniformly* better than that of RLA methods in terms of both ϵ and d.
- Empirically, pwSGD is preferable when a medium-precision solution is desired.