Importance Sampling in Stochastic Gradient Descent and Randomized Kaczmarz Algorithm

Yinuo Zhao

December 4, 2024

1. Introduction: Dependence on $\mathbb{E}[L_i^2/\mu^2]$

Background

We consider the problem of minimizing a smooth convex function,

$$x_\star = \arg\min_x F(x)$$

where $F(x)$ is of the form $F(x) = \mathbb{E}_{i\sim\mathcal{D}} f_i(x)$ for smooth functionals f_i, and σ^2 represents the "residual" quantity at the minimum, $\sigma^2 = \mathbb{E}\|\nabla f_i(x_\star)\|_2^2$.

We will instate the following assumptions on the function F:

1. Each f_i is continuously differentiable and the gradient function ∇f_i has Lipschitz constant L_i ; that is,

$$\|\nabla f_i(x) - \nabla f_i(y)\|_2 \leq L_i\|x-y\|_2 \quad \text{for all vectors } x \text{ and } y.$$

2. F has a strong convexity parameter μ; that is,

$$\langle x-y, \nabla F(x) - \nabla F(y)\rangle \geq \mu\|x-y\|_2^2 \quad \text{for all vectors } x \text{ and } y.$$

1. Introduction: Dependence on $\mathbb{E}[L_i^2/\mu^2]$

Background

Stochastic Gradient Descent (SGD) is a widely used algorithm for optimization in machine learning and convex analysis. The foundational work by Bach and Moulines (2011) provided a non-asymptotic analysis of SGD, focusing on its convergence behavior in strongly convex settings. They derived an iteration complexity bound for achieving a specified accuracy ϵ, given by:

$$k = 2\log\left(\frac{\epsilon_0}{\epsilon}\right)\left(\frac{\mathbb{E}[L_i^2]}{\mu^2} + \frac{\sigma^2}{\mu^2\epsilon}\right),$$

- k is the number of iterations required to achieve the desired accuracy ϵ.
- $\mathbb{E}[L_i^2]$ is the average squared Lipschitz constant of the gradients.
- μ is the strong convexity parameter of the objective function.
- σ^2 represents the variance of the gradient noise.
- ϵ_0 is the initial error bound.

Key Observations

1. **Dependence on $\mathbb{E}[L_i^2/\mu^2]$:**
    - The first term in the complexity bound, $\frac{\mathbb{E}[L_i^2]}{\mu^2}$, reflects the **average squared condition number** of the problem.
    - This quadratic dependence on $\mathbb{E}[L_i^2]$ can significantly slow down convergence, particularly for problems where L_i (the Lipschitz constant for individual components) varies widely.
2. **Effect of Gradient Noise:**
    - The second term, $\frac{\sigma^2}{\mu^2\epsilon}$, dominates when ϵ (the desired accuracy) is very small or when σ^2 (noise variance) is large.
    - This highlights the sensitivity of SGD to noise, making the algorithm less efficient in high-noise scenarios.

2. Supremum Conditioning: Theorem 2.1 and Corollary 2.2

Theorem 2.1

Let each f_i be convex, where ∇f_i has Lipschitz constant L_i, with $L_i \leq \sup L$ almost surely. Let $F(x) = \mathbb{E}[f_i(x)]$ be μ-strongly convex. Set $\sigma^2 = \mathbb{E}[\|\nabla f_i(x_\star)\|^2]$, where $x_\star = \arg\min_x F(x)$. Suppose that the step size $\gamma < \frac{1}{\sup L}$. Then the SGD iterates satisfy:

$$\mathbb{E}[\|x_k - x_\star\|_2^2] \leq \left[1 - 2\gamma\mu(1 - \gamma \sup L)\right]^k \|x_0 - x_\star\|_2^2 + \frac{\gamma\sigma^2}{\mu(1 - \gamma \sup L)},$$

where the expectation is taken over the sampling of $\{i_k\}$.

2. Supremum Conditioning: Theorem 2.1 and Corollary 2.2

Corollary 2.2

For any desired tolerance $\|x - x_\star\|_2^2 \leq \epsilon$, if the Lipschitz constants and strong convexity parameters are known, the step size γ can be optimized as:

$$\gamma = \frac{\mu\epsilon}{2\epsilon\mu \sup L + 2\sigma^2}.$$

With this step size, the number of iterations k required for SGD to reach the tolerance ϵ is given by:

$$k = 2\log\left(\frac{2\epsilon_0}{\epsilon}\right)\left(\frac{\sup L}{\mu} + \frac{\sigma^2}{\mu^2\epsilon}\right),$$

where $\epsilon_0 = \|x_0 - x_\star\|_2^2$.

3. Motivation: Transition to Importance Sampling

Linear Dependence on $\sup L/\mu$: Improvements and Limitations

The convergence bound in **Theorem 2.1** and **Corollary 2.2** replaces the earlier quadratic dependence on $\mathbb{E}[L_i^2/\mu^2]$ with a linear dependence on the **uniform conditioning** $\sup L/\mu$:

$$k = 2\log\left(\frac{2\epsilon_0}{\epsilon}\right)\left(\frac{\sup L}{\mu} + \frac{\sigma^2}{\mu^2\epsilon}\right).$$

This is a **quadratic improvement** in the number of required iterations when all Lipschitz constants L_i are of similar magnitude. However, when the L_i's have large variability (i.e., when the components f_i scale very differently), the supremum $\sup L$ can dominate the average conditioning $\mathbb{E}[L_i^2/\mu^2]$, making the bound overly conservative.

involves $N+1$ quadratic functions:

1. The first function $f_1(x)$:

$$f_1(x) = \frac{N}{2}(x[1] - b)^2, \quad b = \pm 1.$$

2. The remaining N functions $f_i(x)$ ($i = 2, \ldots, N+1$):

$$f_i(x) = \frac{1}{2}x[2]^2.$$

Lipschitz Constants:

- $L_1 = N$, and $L_i = 1$ for $i = 2, \ldots, N+1$.
- $\sup L = N$.

Conditioning Parameters:

- Strong convexity parameter $\mu = \frac{N}{N+1}$.
- Average Lipschitz constant:

$$\bar{L} = \frac{2N}{N+1}.$$

- Average quadratic Lipschitz constant:

$$\mathbb{E}[L_i^2] = N.$$

involves $N+1$ quadratic functions:

1. The first function $f_1(x)$:

$$f_1(x) = \frac{N}{2}(x[1]-b)^2, \quad b = \pm 1.$$

2. The remaining N functions $f_i(x)$ ($i = 2, \ldots, N+1$):

$$f_i(x) = \frac{1}{2}x[2]^2.$$

Iteration Complexity:

- Using uniform sampling:

$$\frac{\sup L}{\mu} = N+1, \quad \mathbb{E}\left[\frac{L_i^2}{\mu^2}\right] \approx N+1.$$

- Linear dependence on $\bar{L}/\mu = 2$ would significantly reduce the iteration count. However, this cannot be achieved with uniform sampling.

Interpretation of the Example

- To achieve meaningful progress in reducing the error, $f_1(x)$ must be sampled frequently because it dominates the optimization problem.
- Uniform sampling, however, treats all components equally. As a result, the algorithm wastes many iterations on less influential components ($f_2, \ldots, f_{N+1}$), requiring $N + 1$ iterations in expectation to adequately sample $f_1(x)$.

Key Insight

- While $\sup L/\mu$ improves upon $\mathbb{E}[L_i^2/\mu^2]$, it remains inefficient in scenarios with high variability in L_i. For such problems, a better strategy is needed to prioritize components like $f_1(x)$ without fully disregarding others.

4. Pure Importance Sampling: Transition to $\bar{L}/\mu$

Weighted SGD Framework

The paper introduces a re-weighting strategy for SGD, where the Lipschitz constant $L_i^{(w)}$ of each component $f_i^{(w)}$ is scaled by a weight $w(i)$:

$$L_i^{(w)} = \frac{L_i}{w(i)}.$$

The supremum Lipschitz constant $\sup L^{(w)}$ is now given by:

$$\sup L^{(w)} = \sup_i \frac{L_i}{w(i)}.$$

4. Pure Importance Sampling: Transition to $\bar{L}/\mu$

Optimal Weights

The supremum $\sup L^{(w)}$ is minimized by choosing the weights:

$$w(i) = \frac{L_i}{\bar{L}}, \quad \text{where} \quad \bar{L} = \mathbb{E}[L_i].$$

Substituting this choice of weights:

$$\sup L^{(w)} = \sup_i \frac{L_i}{L_i/\bar{L}} = \bar{L}.$$

Thus, by setting $w(i) \propto L_i$, the re-weighted Lipschitz constant $\sup L^{(w)}$ equals the **average Lipschitz constant** $\bar{L}$, eliminating dependence on $\sup L$.

4. Pure Importance Sampling: Transition to $\bar{L}/\mu$

Iteration Complexity with Importance Sampling

Applying Corollary 2.2 to the re-weighted SGD iterations with weights $w(i) = \frac{L_i}{\bar{L}}$, the iteration complexity becomes:

$$k = 2\log(2\varepsilon_0/\varepsilon)\left(\frac{\sup L_{(w)}}{\mu} + \frac{\sigma^2_{(w)}}{\mu^2\varepsilon}\right) \leq 2\log(2\varepsilon_0/\varepsilon)\left(\frac{\bar{L}}{\mu} + \frac{\bar{L}}{\inf L}\cdot\frac{\sigma^2}{\mu^2\varepsilon}\right)$$

Here:

- The first term $\frac{\bar{L}}{\mu}$ reflects the dependence on the **average Lipschitz constant**.
- However, the inequality might be tight in the presence of components with very small L_i that contribute towards the residual error. When $\sigma^2 > 0$, we therefore get a dissatisfying scaling of the second term by a factor of $\bar{L}/\inf L$.

5. Partially Biased Sampling: Mixture Importance Sampling to Address Residual Problems

Motivation

While pure importance sampling achieves a dependence on $\bar{L}/\mu$, it amplifies sensitivity to residual noise, especially in the presence of components with very small L_i. This issue becomes evident when the noise variance σ^2 is non-zero, leading to unsatisfactory scaling of the second term in the complexity bound:

$$k = 2\log\left(\frac{2\epsilon_0}{\epsilon}\right)\left(\frac{\bar{L}}{\mu} + \frac{\sigma^2}{\mu^2\epsilon}\right).$$

To mitigate this, the paper introduces **partially biased sampling**, which balances the benefits of uniform sampling (stability) and importance sampling (speed).

Mixture Importance Sampling

The partially biased sampling distribution is a **mixture** of uniform sampling and importance sampling. The weights are defined as:

$$w(i) = \frac{1}{2} + \frac{1}{2} \cdot \frac{L_i}{\bar{L}},$$

where:

- $\frac{1}{2}$ represents the uniform sampling component.
- $\frac{1}{2} \cdot \frac{L_i}{\bar{L}}$ represents the importance sampling component.

This weighting ensures that all components are sampled, while giving more emphasis to components with larger L_i.

Key Results with Partially Biased Sampling

1. **Supremum Lipschitz Constant**: The re-weighted supremum Lipschitz constant is bounded as:

$$\sup L^{(w)} = \sup_i \frac{L_i}{\frac{1}{2} + \frac{1}{2} \cdot \frac{L_i}{\bar{L}}} \leq 2\bar{L}.$$

   This shows that the supremum Lipschitz constant is now effectively controlled by $\bar{L}$, reducing worst-case dependence.

2. **Noise Variance**: The variance term $\sigma^{(w)2}$ under the partially biased distribution is bounded as:

$$\sigma^{(w)2} \leq 2\sigma^2.$$

   This ensures that the noise amplification seen in pure importance sampling is mitigated.

Key Results with Partially Biased Sampling

3. **Iteration Complexity**: Substituting these bounds into Corollary 2.2, the iteration complexity for partially biased sampling becomes:

$$k = 4 \log \left(\frac{2\epsilon_0}{\epsilon} \right) \left(\frac{\bar{L}}{\mu} + \frac{\sigma^2}{\mu^2 \epsilon} \right).$$

   Compared to pure importance sampling, the partially biased approach maintains the dependence on $\bar{L}/\mu$ while controlling the residual noise term.

A Family of Partially Biased Schemes

The choice of weights corresponds to an equal mix of uniform and fully biased sampling. More generally, we could consider sampling according to any one of a family of weights which interpolate between uniform and fully biased sampling:

$$w^{\lambda}(i) = \lambda + (1-\lambda)\frac{L_i}{\bar{L}}, \quad \lambda \in [0,1].$$

- **When $\lambda = 0$:** This corresponds to **fully biased sampling**, where the weights are proportional to L_i. Larger L_i gets higher sampling probability, favoring samples with higher Lipschitz constants.
- **When $\lambda = 1$:** This corresponds to **uniform sampling**, where all samples have equal weights, regardless of L_i.

Algorithm 3.1: Stochastic Gradient Descent with Partially Biased Sampling

Input:

- Initial estimate $\boldsymbol{x}_0 \in \mathbb{R}^d$
- Bias parameter $\lambda \in [0, 1]$
- Step size $\gamma > 0$
- Tolerance parameter $\delta > 0$
- Access to the source distribution $\mathcal{D}$
- If $\lambda < 1$: bounds on the Lipschitz constants L_i; the weights $w^\lambda(i)$ derived from them (see eq. 3.16); and access to the weighted distribution $\mathcal{D}^{(\lambda)}$.

$$w^\lambda(i) = \lambda + (1 - \lambda)\frac{L_i}{\bar{L}}, \quad \lambda \in [0, 1]$$

Output: Estimated solution $\hat{\boldsymbol{x}}$ to the problem $\min_{\boldsymbol{x}} F(\boldsymbol{x})$

$k \leftarrow 0$

repeat

$k \leftarrow k + 1$

Draw an index $i \sim \mathcal{D}^{(\lambda)}$.

$\boldsymbol{x}_k \leftarrow \boldsymbol{x}_{k-1} - \frac{\gamma}{w^\lambda(i)} \nabla f_i(\boldsymbol{x}_{k-1})$

until $\nabla F(\boldsymbol{x}) \leq \delta$

$\hat{\boldsymbol{x}} \leftarrow \boldsymbol{x}_k$

Advantages of Partially Biased Sampling

1. **Balanced Sampling**:
    - Combines the stability of uniform sampling with the efficiency of importance sampling.
    - Ensures that small L_i components are not completely ignored, reducing noise sensitivity.
2. **Improved Robustness**:
    - Unlike pure importance sampling, partially biased sampling avoids the excessive scaling of the noise variance σ^2, making it more robust in noisy scenarios.
3. **General Applicability**:
    - Achieves a practical balance, making it suitable for both smooth and non-smooth objectives where residual error or noise is significant.

6. Connection to the Kaczmarz Method

Introduction

The **Randomized Kaczmarz (RK) method** is a classical algorithm for solving linear systems $Ax = b$, and it has been shown to be an instance of **Stochastic Gradient Descent (SGD)** when applied to the least squares formulation. This connection allows RK to leverage the techniques and insights developed for SGD, such as importance sampling and partially biased sampling, to improve convergence rates and robustness.

6. Connection to the Kaczmarz Method

1. RK as an Instance of SGD

The least squares problem minimizes the quadratic objective:

$$F(x) = \frac{1}{2n}\|Ax - b\|_2^2 = \frac{1}{2n}\sum_{i=1}^{n}\left((a_i, x) - b_i\right)^2,$$

where a_i are rows of A and b is the observation vector. The gradient of $F(x)$ is:

$$\nabla F(x) = \frac{1}{n}\sum_{i=1}^{n}\nabla f_i(x), \quad \text{with } \nabla f_i(x) = \left((a_i, x) - b_i\right) a_i.$$

6. Connection to the Kaczmarz Method

1. RK as an Instance of SGD

The RK update rule:

$$x_{k+1} = x_k + \frac{b_i - (a_i, x_k)}{\|a_i\|_2^2} a_i,$$

can be rewritten as:

$$x_{k+1} = x_k - \gamma \nabla f_i(x_k),$$

where $\gamma = \frac{1}{\|a_i\|_2^2}$. This is equivalent to SGD with row a_i sampled based on the probability $p_i \propto \|a_i\|_2^2$, making RK a specific instance of weighted SGD.

6. Connection to the Kaczmarz Method

2. Sampling by Row Norm

The RK method selects rows a_i with probabilities proportional to their squared norm:

$$p_i = \frac{\|a_i\|_2^2}{\|A\|_F^2},$$

where $\|A\|_F^2$ is the Frobenius norm of A. This importance sampling strategy ensures faster error reduction for rows with larger norms and aligns with the partially biased and hybrid sampling strategies seen in SGD.

Theoretical guarantees for RK show exponential convergence:

$$\mathbb{E}[\|x_k - x_\star\|_2^2] \leq \left(1 - \frac{\lambda_{\min}^+(A^T A)}{\|A\|_F^2}\right)^k \|x_0 - x_\star\|_2^2,$$

where $\lambda_{\min}^+(A^T A)$ is the smallest non-zero eigenvalue of $A^T A$. This highlights the role of the condition number $K(A) = \|A\|_F^2 / \lambda_{\min}^+(A^T A)$ in determining the convergence rate.

6. Connection to the Kaczmarz Method

3. Introducing Mixture Sampling

Building on the partially biased sampling strategy discussed in SGD, the RK method incorporates **mixture sampling** to balance the benefits of uniform sampling (stability) and importance sampling (efficiency). The modified RK update is:

$$x_{k+1} = x_k + 2c \cdot \frac{b_i - (a_i, x_k)}{\|A\|_F^2/n + \|a_i\|_2^2} a_i,$$

with the sampling probability:

$$p_i = \frac{1}{2} \cdot \frac{\|a_i\|_2^2}{\|A\|_F^2} + \frac{1}{2} \cdot \frac{1}{n}.$$

6. Connection to the Kaczmarz Method

3. Introducing Mixture Sampling

This mixture distribution ensures:

- Faster convergence by prioritizing rows with larger norms,
- Reduced noise amplification compared to fully biased sampling,
- Stability by ensuring smaller rows are not completely ignored.

The partially biased RK method achieves a convergence bound of:

$$\mathbb{E}[\|x_k - x_\star\|_2^2] \leq \left[1 - \frac{2c(1-2c)}{K(A)}\right]^k \|x_0 - x_\star\|_2^2 + \frac{cK(A)}{1-2c} \cdot \frac{2\sigma^2}{n\|A\|_F^2}.$$

6. Connection to the Kaczmarz Method

In this section, we present some numerical results for the randomized Kaczmarz algorithm with partially biased sampling, that is, applying Algorithm to the least squares problem $F(x) = \frac{1}{2}\|Ax - b\|_2^2$ (so $f_i(x) = \frac{n}{2}(\langle a_i, x\rangle - b_i)^2$) and considering $\lambda \in [0, 1]$.

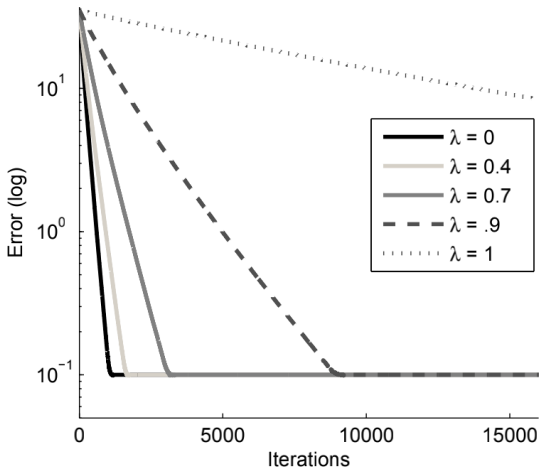
Experimental Setup:

- The least squares problem is given by $F(x) = \frac{1}{2}\|Ax - b\|_2^2$, where A is a 1000×10 matrix, x is the variable vector, and b includes noise e.

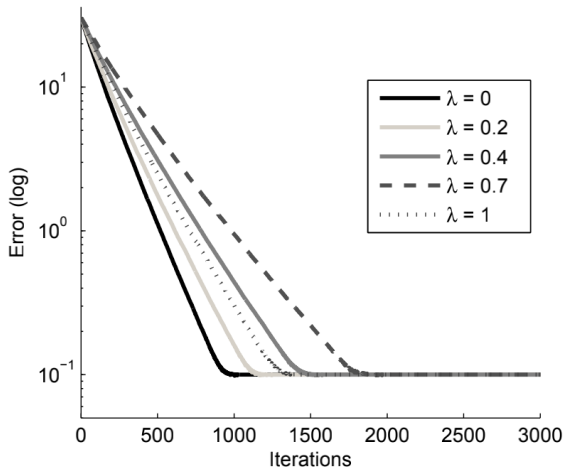
  1. **Case 1**: Rows of A are standard normal except for the last row with a variance of 10^2; noise e has variance 0.1^2.
  2. **Case 2**: All rows have standard normal entries, and e has variance 0.1^2.
  3. **Case 3-5**: Rows are altered with increasing variance j, with cases categorized as "high," "medium," and "low" noise regimes.

6. Connection to the Kaczmarz Method

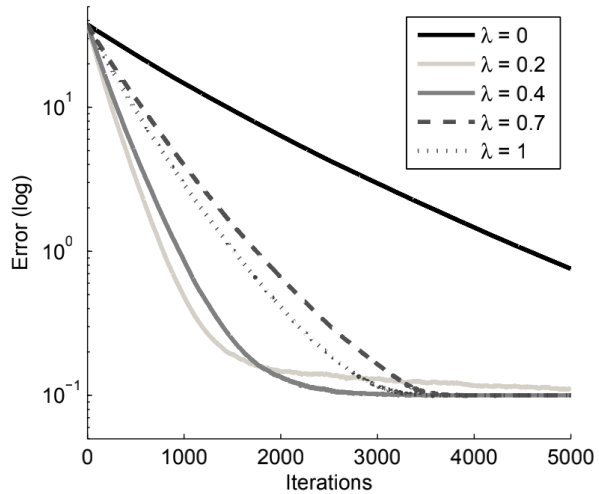
Case 1:



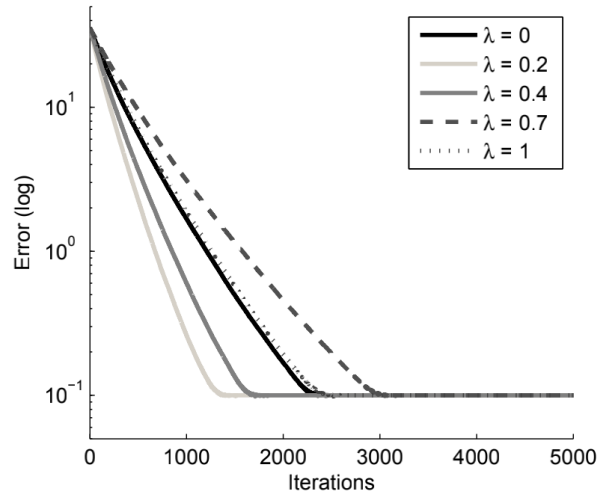
Case 2:



Case 3:

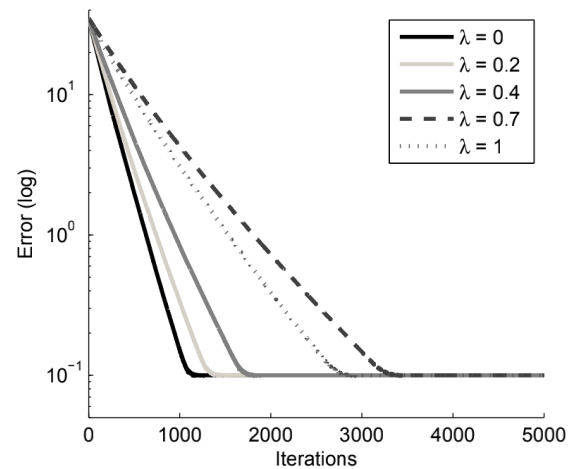


Case 4:



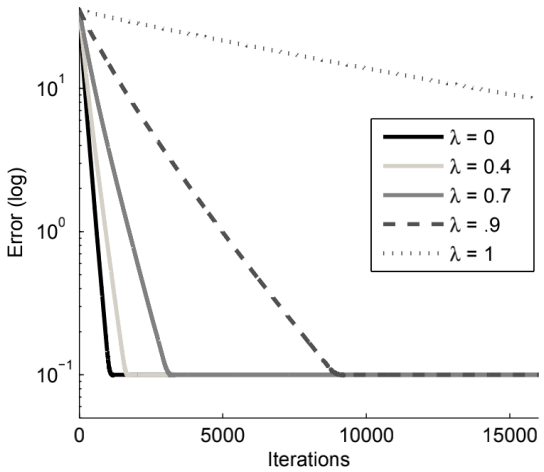
1. **Case 1**: Rows of A are standard normal except for the last row with a variance of 10^2; noise e has variance 0.1^2.
2. **Case 2**: All rows have standard normal entries, and e has variance 0.1^2.
3. **Case 3-5**: Rows are altered with increasing variance j, with cases categorized as "high," "medium," and "low" noise regimes.

Case 5:

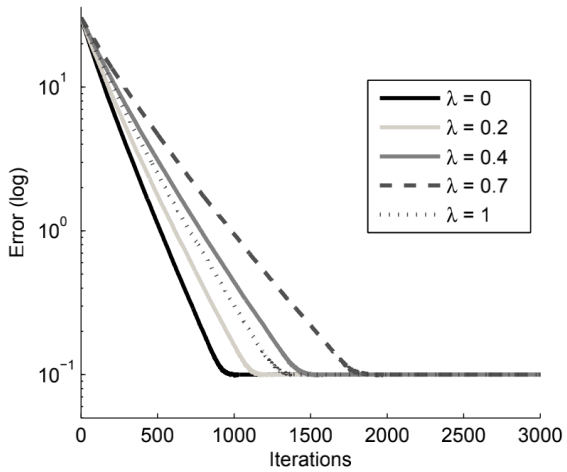


6. Connection to the Kaczmarz Method

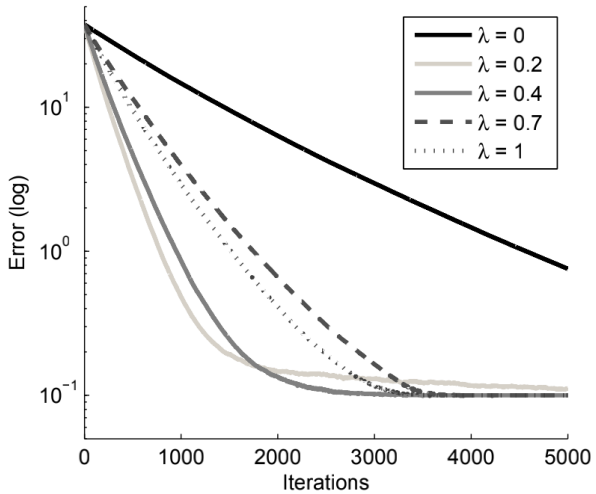
Case 1:



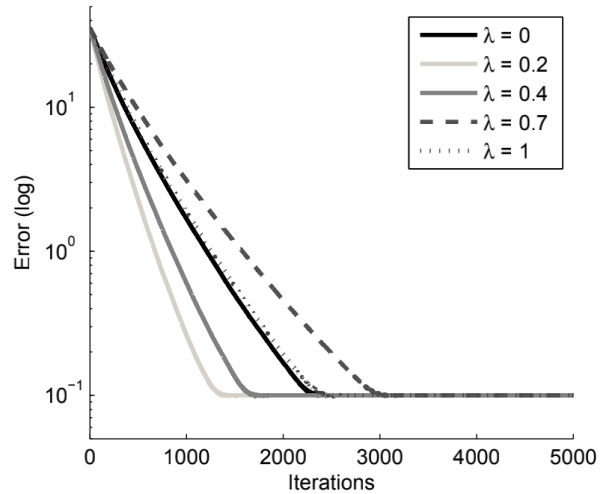
Case 2:



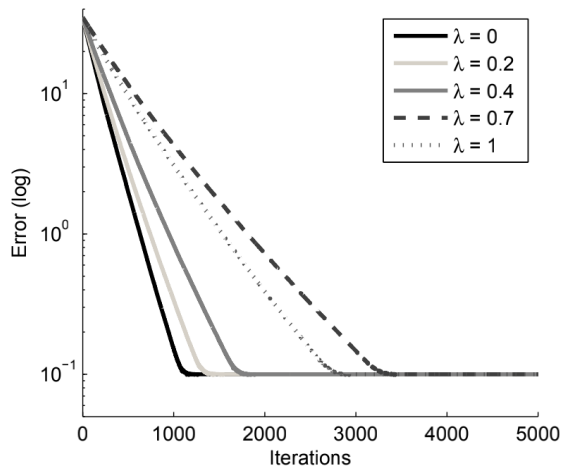
Case 3:



Case 4:



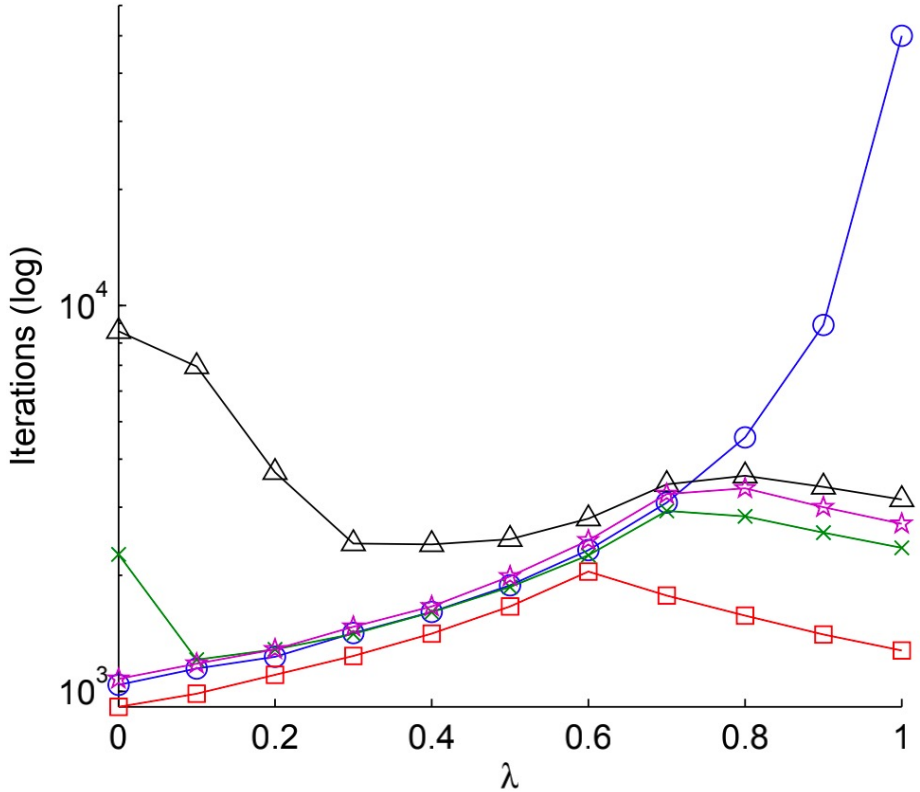
Case 5:



Convergence Behavior:

- Figure 1 illustrates the log-scale approximation error across iterations for different λ values. Cases with hybrid sampling consistently outperform the extremes ($\lambda = 0$ or $\lambda = 1$) in noisy conditions.

6. Connection to the Kaczmarz Method



Case 1: Rows of A are standard normal except for the last row with a variance of 10^2, e has variance 0.12

Case 3: Rows are altered with increasing variance j, high noise regimes
Case 5: Rows are altered with increasing variance j, low noise regimes
Case 4: Rows are altered with increasing variance j, medium noise regimes

Case 2: All rows have standard normal entries, and e has variance 0.1^2

- Figure 2 shows the iteration count needed to achieve a fixed error $\|x_k - x_*\|_2^2 \leq \epsilon$. The trends reaffirm that hybrid sampling optimally balances convergence speed and residual minimization.

Summary

- The improved dependence on the conditioning for smooth and strongly convex SGD.
- The discussion of importance sampling for SGD.
- The connection between SGD and the randomized Kaczmarz method.