25S 01:Comparison of SVRG and SGD on Convex Optimization Problems

JunHyun Kim

May 2025

1 Introduction

This report implements Stochastic Variance Reduced Gradient (SVRG) algorithm to solve convex optimization problems, namely L2-regularized least squares and logistic regression.

[Ahmet: Hi JunHyun, thanks! Can you also include regular gradient descent in the comparisons in Fig 3 and Fig 6?]

2 Experiments

The SVRG algorithm was implemented and evaluated on two convex optimization problems: L2-regularized least squares and logistic regression. Training loss was plotted on a logarithmic scale versus the number of gradient evaluations normalized by dataset size.

To initialize the method, the initial point in all SVRG experiments was generated by performing a single iteration of SGD, following the approach in the original SVRG paper.

In addition to tracking loss, the variance of the gradient update directions was computed to assess stability during optimization. For SGD, the variance at iteration t was defined as:

$$\text{Var}_{\text{SGD}} = \frac{1}{n}\sum_{i=1}^{n} \|\nabla f_i(w_t) - \nabla f(w_t)\|^2$$

For SVRG, the variance of the variance-reduced estimator was computed as:

$$\text{Var}_{\text{SVRG}} = \frac{1}{n}\sum_{i=1}^{n} \|\nabla f_i(w_{t-1}) - \nabla f_i(\tilde{w}) + \nabla f(\tilde{w}) - \nabla f(w_{t-1})\|^2$$

These quantities were evaluated at each inner iteration and plotted to visualize the variance reduction effect of SVRG compared to standard SGD.

2.1 SVRG vs. SGD on Least Squares Regression

First tested SVRG on the L2-regularized least squares objective

$$f(w) = \frac{1}{2n}\|Aw - b\|^2 + \frac{\lambda}{2}\|w\|^2$$

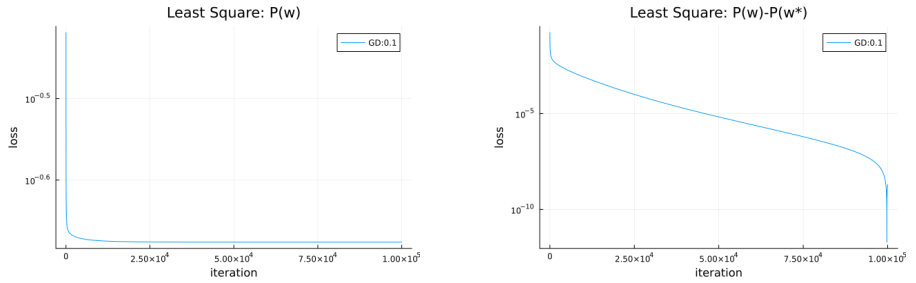
where $A \in \mathbb{R}^{1284\times123}$, $b \in \mathbb{R}^{1284}$, and $\lambda = 1e-4$.

Initially, by using `Convex.jl` with `SCS.Optimizer` in julia,

```
w1 = Variable(size(Atrain, 2))
n = size(Atrain, 1)
lsProblem = minimize((0.5 / n) * sumsquares((Atrain * w1 - btrain)) +
    0.5 *    * sumsquares(w1))
solve!(lsProblem, SCS.Optimizer)
```

The optimal value computed was 0.21079.

First, using gradient descent (GD) with step size 0.1 for 100,000 iterations yielded an optimal value of 0.21080, which closely matches the numerical solution obtained via Convex.jl using the SCS solver. As shown in Figure (b), the residual decreases steadily and reaches a minimum near iteration 100,000; however, it increases afterward due to cancellation error.



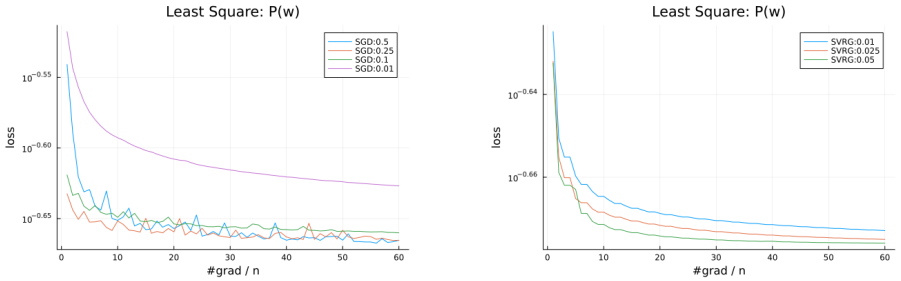
(a) Training loss

(b) Residual $P(w) - P(w^*)$

Figure 1: Gradient Descent on least squares. (a) Training loss. (b) Training loss residual $P(w) - P(w^*)$.

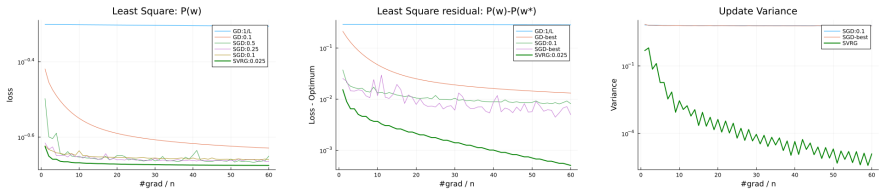
For SGD, decaying step sizes of the form $\alpha_k = \frac{c}{\sqrt{k}}$ were used, with $c \in [0.1, 0.5]$. Smaller values of c resulted in slower convergence, while larger values led to oscillations due to overshooting.

For SVRG, a fixed step size $\alpha \in [0.1, 0.5]$ was used with an inner loop length of $m = 2n$. SVRG demonstrated stable and fast convergence throughout the training process.



(a) SGD (b) SVRG

Figure 2: SGD and SVRG on the least squares problem. (a) Training loss from SGD. (b) Training loss from SVRG. Legends indicate the learning rates.



(a) Training loss (b) Loss residual $P(w) - P(w^*)$ (c) Gradient variance

Figure 3: Least squares problem with regularization parameter $\lambda =$ 1e-4, solved using `Convex.jl`. (a) Training loss comparison for SGD with decaying learning rates. The numbers in the legend indicate learning rate values. (b) Training loss residual $P(w) - P(w^*)$. (c) Variance of gradient updates.

Figure 3 shows that SVRG converges faster and more smoothly than both GD and SGD, despite using a smaller step size. In both training loss and loss residual, SVRG reaches lower values with fewer gradient computations. The gradient variance plot clearly shows that SVRG reduces variance over time, while SGD's variance stays high. This supports the effectiveness of SVRG in improving stability and convergence.

2.2 SVRG vs. SGD on Logistic Regression

The second experiment was conducted on L2-regularized logistic regression, with the objective defined as:

$$f(w) = \sum_{i=1}^{n} \log\left(1 + \exp(-b_i a_i^\top w)\right) + \frac{\lambda}{2}\|w\|^2$$

where $A \in \mathbb{R}^{1024\times 784}$, $b \in \mathbb{R}^{1024}$, and $\lambda =$ 1e-1.

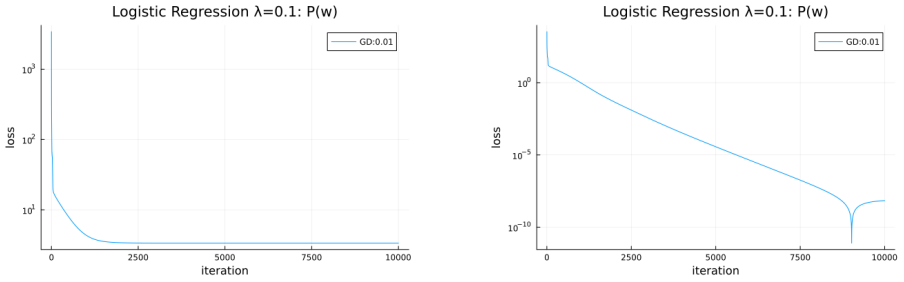
To obtain a numerical reference solution, the problem was solved using `Convex.jl` with the `SCS.Optimizer`:

```
w1 = Variable(size(Atrain, 2))
n = size(Atrain, 1)
lrProblem = minimize(logisticloss(-btrain .* (Atrain  * w1)) + 0.5 *
     * dot(w1, w1))

solve!(lrProblem, SCS.Optimizer)
```

The optimal value computed was 3.357001.

Gradient Descent (GD) was applied for 10,000 iterations with a fixed step size 0.01. The training loss and residual $P(w) - P(w^*)$ showed that GD converged steadily towards the optimum. However, in later iterations, the residual slightly increased due to cancellation error. The logistic regression loss was not normalized by $\frac{1}{n}$, unlike the least squares objective. This difference in scaling may justify the smaller step size used in practice.



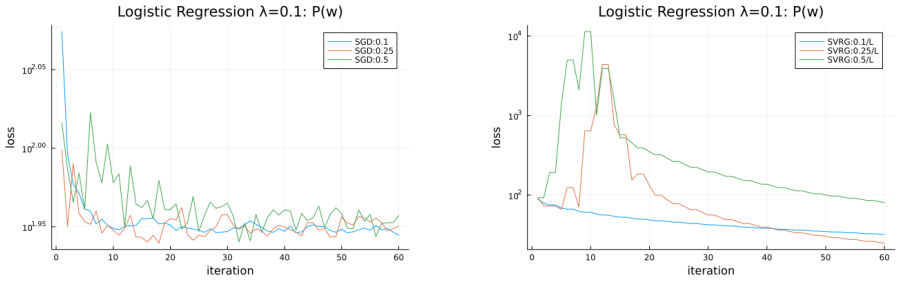
(a) Training loss

(b) Residual $P(w) - P(w^*)$

Figure 4: Gradient Descent on logistic regression. (a) Training loss. (b) Loss residual $P(w) - P(w^*)$.

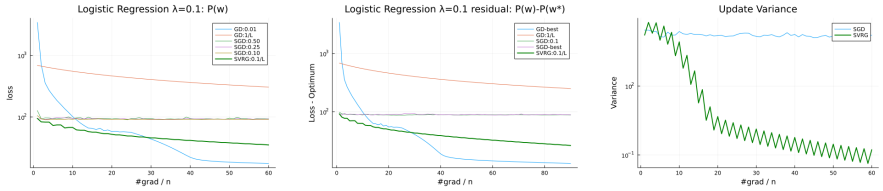
For SGD, similar with least square problem decaying step sizes of the form $\alpha_k = \frac{c}{\sqrt{k}}$ were used with $c \in [0.1, 0.5]$. Smaller values of c resulted in slower convergence, while larger values led to instability and oscillation during early iterations.

For SVRG, step sizes were selected based on the theoretical bound $\alpha < \frac{1}{4L}$, where L is the Lipschitz constant of the gradient; $\frac{\|A\|^2}{4} + \lambda = 174421$ in this practice. While both $1/4L$ and $c = 1/2L$ showed some initial oscillation, the step size $\alpha = \frac{1}{4L}$ ultimately achieved the fastest and most stable convergence. The inner loop length was fixed at $m = 2n$ as in the least squares case.



(a) SGD (b) SVRG

Figure 5: SGD and SVRG on logistic regression. (a) Training loss from SGD with decaying step sizes. (b) Training loss from SVRG with fixed step sizes $\alpha = \frac{c}{L}$.



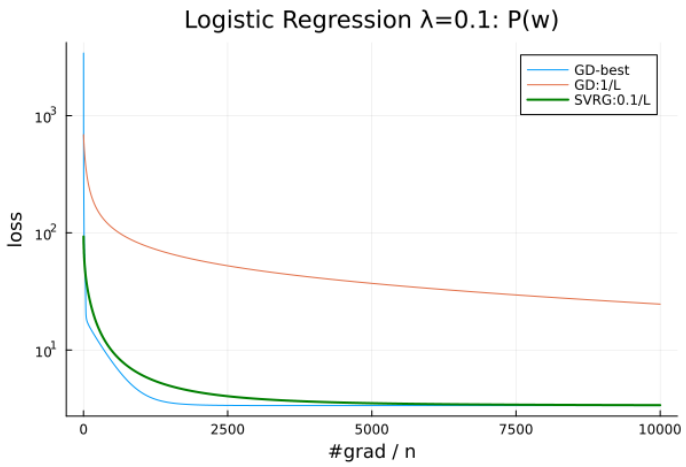
(a) Training loss (b) Loss residual $P(w) - P(w^*)$ (c) Gradient variance

Figure 6: Logistic regression with regularization parameter $\lambda = 1e-1$, solved using `Convex.jl`. (a) Training loss comparison for GD, SGD and SVRG. (b) Loss residual $P(w) - P(w^*)$. (c) Variance of gradient updates. SVRG shows significantly lower variance compared to SGD.

Figure 6 demonstrates that SVRG achieves superior convergence behavior over SGD. However, due to the small step size used in SVRG, specifically $\alpha = 0.1/L = 5.73 \times 10^{-7}$, gradient descent achieves lower loss and residual values after approximately 30 iterations.

Nevertheless, when GD is run with its theoretical step size $1/L$, SVRG converges faster and more smoothly, highlighting the advantage of variance reduction under comparable conditions.

In both training loss and residual plots, SVRG reaches the optimal value with fewer gradient evaluations than SGD. The gradient variance plot clearly indicates that SVRG effectively reduces the variance of its update direction over time, contributing to its stability and fast convergence.



(a) Training loss

Figure 7: Training loss comparison between GD and SVRG on logistic regression $\lambda = 1e-1$ over 10,000 iterations.

Figure 7 shows even after 10,000 iterations, SVRG's final loss remained about 0.036064 (about 1%) higher than that of GD-best, reflecting the impact of its extremely small step size on practical convergence speed. Nevertheless, compared to gradient descent using the theoretical step size $1/L$, SVRG converges significantly faster and achieves lower loss values with fewer gradient evaluations.

In both experiments, SVRG achieved the better objective value with substantially fewer gradient evaluations than SGD. Moreover, compared to gradient descent with the theoretical step size $1/L$, SVRG demonstrated much faster convergence, underscoring its practical efficiency under limited computational budgets.

3 Conclusion

This report presented a comparison of SGD and SVRG on two convex optimization problems: least squares and logistic regression. The results confirm that SVRG achieves faster and more stable convergence without requiring a decaying step size, primarily due to its variance reduction and ability to maintain a fixed step size.

While gradient descent with well-tuned parameters can sometimes achieve lower final losses, SVRG demonstrated superior convergence efficiency, particularly when compared to SGD and gradient descent using theoretical step size $1/L$. These findings support the theoretical advantages of SVRG and demonstrate its practical effectiveness for convex optimization under limited computational budgets.