Mathematics of Data: Homework Exercise-1

Hiroki Hayakawa

1/11/2019

1 Problem Statement

1-(a)

From the problem statement, $f(\mathbf{x})$ is defined as

$$f(\mathbf{x}) := \frac{1}{n}\sum_{i=1}^{n} g_i(\mathbf{x}) + \frac{\lambda}{2}\|x\|^2, \tag{1}$$

$$, \text{where } g_i(\mathbf{x}) := \begin{cases} \frac{1}{2} - b_i \mathbf{a_i}^T \mathbf{x} & b_i(\mathbf{a_i}^T \mathbf{x}) \leq 0, \\ \frac{1}{2}(1 - b_i \mathbf{a_i}^T \mathbf{x})^2 & 0 < b_i(\mathbf{a_i}^T \mathbf{x}) \leq 1, \\ 0, & 1 < b_i(\mathbf{a_i}^T \mathbf{x}). \end{cases} \tag{2}$$

By using $\mathbf{I}_L(i,i)$ and $\mathbf{I}_Q(i,i)$, $f(\mathbf{x})$ can be written as

$$f(\mathbf{x}) = \frac{\lambda}{2}\|x\|^2 + \frac{1}{n}\sum_{i=1}^{n} \mathbf{I}_L(i,i)(\frac{1}{2} - b_i \mathbf{a_i}^T \mathbf{x}) + \frac{1}{2n}\sum_{i=1}^{n} \mathbf{I}_Q(i,i)(1 - b_i \mathbf{a_i}^T \mathbf{x})^2. \tag{3}$$

By virtue of properties of gradient, we can express $\nabla f(\mathbf{x})$ as follows.

$$\nabla f(\mathbf{x}) = \nabla(\frac{\lambda}{2}\|x\|^2) + \nabla(\frac{1}{n}\sum_{i=1}^{n} \mathbf{I}_L(i,i)(\frac{1}{2} - b_i \mathbf{a_i}^T \mathbf{x})) + \nabla(\frac{1}{2n}\sum_{i=1}^{n} \mathbf{I}_Q(i,i)(1 - b_i \mathbf{a_i}^T \mathbf{x})^2). \tag{4}$$

We now calculate each term in the right-hand side of equation (4).

$$\nabla(\frac{\lambda}{2}\|x\|^2) = \lambda \mathbf{x}, \tag{5}$$

$$\begin{aligned} \nabla(\frac{1}{n}\sum_{i=1}^{n} \mathbf{I}_L(i,i)(\frac{1}{2} - b_i \mathbf{a_i}^T \mathbf{x})) =& \frac{1}{n}\sum_{i=1}^{n} \nabla(\mathbf{I}_L(i,i)(\frac{1}{2} - b_i \mathbf{a_i}^T \mathbf{x})) \\ =& -\frac{1}{n}\sum_{i=1}^{n} \mathbf{I}_L(i,i) b_i \mathbf{a_i} \\ =& -\frac{1}{n}\left[b_1 \mathbf{a}_1, \cdots, b_n \mathbf{a}_n\right] diag(\mathbf{I}_L(1,1), \cdots, \mathbf{I}_L(n,n))\mathbf{1} \\ =& -\frac{1}{n}\tilde{\mathbf{A}}^T \mathbf{I}_L \mathbf{1} \end{aligned} \tag{6}$$

$$\begin{aligned}
\nabla(\frac{1}{2n}\sum_{i=1}^{n}\mathbf{I}_Q(i,i)(1-b_i{\mathbf{a_i}}^T\mathbf{x})^2) =& \frac{1}{2n}\sum_{i=1}^{n}\mathbf{I}_Q(i,i)\nabla(1-b_i{\mathbf{a_i}}^T\mathbf{x})^2 \\
=& \frac{1}{n}\sum_{i=1}^{n}\mathbf{I}_Q(i,i)(1-b_i{\mathbf{a_i}}^T\mathbf{x})\nabla(1-b_i\mathbf{a}_i^T\mathbf{x}) \\
=& \frac{1}{n}\sum_{i=1}^{n}\mathbf{I}_Q(i,i)(b_i{\mathbf{a_i}}^T\mathbf{x}-1)b_i\mathbf{a}_i \\
=& \frac{1}{n}\left[b_1\mathbf{a}_1,\cdots,b_n\mathbf{a}_n\right] diag(\mathbf{I}_Q(1,1),\cdots,\mathbf{I}_Q(n,n)) \begin{bmatrix} b_1\mathbf{a}_1^T \\ \vdots \\ b_n\mathbf{a}_n^T \end{bmatrix}(\mathbf{x}-\mathbf{1}) \\
=& \frac{1}{n}\tilde{\mathbf{A}}^T\mathbf{I}_Q[\tilde{\mathbf{A}}\mathbf{x}-\mathbf{1}] \qquad (7)
\end{aligned}$$

Compiling the calculations above, we obtain the desired equation below.

$$\nabla f(\mathbf{x}) = \lambda\mathbf{x} + \frac{1}{n}\tilde{\mathbf{A}}^T\mathbf{I}_Q[\tilde{\mathbf{A}}\mathbf{x}-\mathbf{1}] - \frac{1}{n}\tilde{\mathbf{A}}^T\mathbf{I}_L\mathbf{1} \qquad (8)$$

The linear part of gradient$f(\mathbf{x})$ is $\frac{1}{n}\tilde{\mathbf{A}}^T\mathbf{I}_L\mathbf{1}$, so the Lipschitz constant for the part is zero. Thus, it is sufficient to compute the Lipchtz constant L for remaining parts with an assumption that all samples $\mathbf{a}_i$ lie in a region where $0 < b_i({\mathbf{a_i}}^T\mathbf{x}) \leq 1$. We define

$$\nabla f_Q(\mathbf{x}) = \lambda\mathbf{x} + \frac{1}{n}\tilde{\mathbf{A}}^T\mathbf{I}_Q[\tilde{\mathbf{A}}\mathbf{x}-\mathbf{1}]. \qquad (9)$$

By assuming that all samples $\mathbf{a}_i$ lie in a region where $0 < b_i({\mathbf{a_i}}^T\mathbf{x}) \leq 1$, the quation(8) can be expressed as

$$\nabla f_Q(\mathbf{x}) = \lambda\mathbf{x} + \frac{1}{n}\tilde{\mathbf{A}}^T[\tilde{\mathbf{A}}\mathbf{x}-\mathbf{1}], \qquad (10)$$

and from properties of norms and Cauchy Schwarz inequality, we obtain following inequality.

$$\begin{aligned}
\|\nabla f_Q(\mathbf{x}) - \nabla f_Q(\mathbf{y})\| =& \|\lambda(\mathbf{x}-\mathbf{y}) + \frac{1}{n}\tilde{\mathbf{A}}^T\tilde{\mathbf{A}}(\mathbf{x}-\mathbf{y})\| \\
\leq& \lambda\|\mathbf{x}-\mathbf{y}\| + \frac{1}{n}\|\tilde{\mathbf{A}}^T\|\|\tilde{\mathbf{A}}\|\|\mathbf{x}-\mathbf{y}\| \qquad (11)
\end{aligned}$$

By utilizing the fact that $\|\tilde{\mathbf{A}}^T\| = \|\mathbf{A}^T\|$, $\|\tilde{\mathbf{A}}\| = \|\mathbf{A}\|$, the inequality(11) can be described as follows.

$$\begin{aligned}
\|\nabla f_Q(\mathbf{x}) - \nabla f_Q(\mathbf{y})\| \leq& \lambda\|\mathbf{x}-\mathbf{y}\| + \frac{1}{n}\|\mathbf{A}^T\|\|\mathbf{A}\|\|\mathbf{x}-\mathbf{y}\| \\
=& (\lambda + \frac{1}{n}\|\mathbf{A}^T\|\|\mathbf{A}\|)\|\mathbf{x}-\mathbf{y}\| \qquad (12)
\end{aligned}$$

This shows that ∇f is L-Lipschitz continuous with $L = \lambda + \frac{1}{n}\|\mathbf{A}^T\|\|\mathbf{A}\|$.

1-(b)

For f, ∇f is L-Lipschitz continuous for any $\mathbf{x} \in \mathbb{R}^p$ and is a composition of affine vector functions. Thus, there is a Hessian for f and f is twice-differenciable at x. Here we denote $\mathbf{x} = [x_1, \ldots, x_p]^T$, $\tilde{\mathbf{A}}^T\tilde{\mathbf{A}} =$

$[\mathbf{c}_1, \ldots, \mathbf{c}_p]^T$, $\mathbf{c}_i = [c_{i1}, \ldots, c_{ip}]^T$, and $\nabla f = [\nabla f_1, \ldots, \nabla f_p]T$. We can express $\tilde{\mathbf{A}}^T\tilde{\mathbf{A}}\mathbf{x} = [\mathbf{c}_1^T\mathbf{x}, \ldots, \mathbf{c}_p^T\mathbf{x}]^T$. Given that $\mathbf{I}_Q = \mathbb{I}$, hessian of f is

$$
\begin{aligned}
\nabla^2 f(\mathbf{x}) &= \begin{bmatrix} \frac{\partial \nabla f_1}{\partial x_1} & \cdots & \frac{\partial \nabla f_p}{\partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial \nabla f_1}{\partial x_p} & \cdots & \frac{\partial \nabla f_p}{\partial x_p} \end{bmatrix} \\
&= \begin{bmatrix} \frac{\partial \lambda x_1}{\partial x_1} & \cdots & \frac{\partial \lambda x_p}{\partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial \lambda x_1}{\partial x_p} & \cdots & \frac{\partial \lambda x_p}{\partial x_p} \end{bmatrix} + \frac{1}{n} \begin{bmatrix} \frac{\partial \mathbf{c}_1^T\mathbf{x}}{\partial x_1} & \cdots & \frac{\partial \mathbf{c}_p^T\mathbf{x}}{\partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial \mathbf{c}_1^T\mathbf{x}}{\partial x_p} & \cdots & \frac{\partial \mathbf{c}_p^T\mathbf{x}}{\partial x_p} \end{bmatrix} + derivatives\ of\ consts(= zero) \\
&= \lambda \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix} + \frac{1}{n} \begin{bmatrix} c_{11} & \cdots & c_{p1} \\ \vdots & \ddots & \vdots \\ c_{1p} & \cdots & c_{pp} \end{bmatrix} \\
&= \lambda\mathbb{I} + \frac{1}{n}\tilde{\mathbf{A}}^T\tilde{\mathbf{A}}
\end{aligned}
\tag{13}
$$

In addition, the entries of $\tilde{\mathbf{A}}^T\tilde{\mathbf{A}}$ can be described as $\tilde{\mathbf{A}}^T\tilde{\mathbf{A}}_{ij} = \sum_{k=1}^{n} b_k^2 \mathbf{a}_{ki}\mathbf{a}_{kj}$. Given that $b_i^2 = 1$ from its definition, we thus obtain

$$\tilde{\mathbf{A}}^T\tilde{\mathbf{A}} = \mathbf{A}^T\mathbf{A} \tag{14}$$

and

$$\nabla^2 f(\mathbf{x}) = \lambda\mathbb{I} + \frac{1}{n}\mathbf{A}^T\mathbf{A} \tag{15}$$

1-(c)

The objective function f is quadratic, so its domain is the set of all real value $\mathbb{R}^p$. Thus, $dom(f)$ is convex. In addition, f is twice differentiable at $\mathbf{x}$ and $\nabla^2 f(\mathbf{x}) - \lambda\mathbb{I} = \frac{1}{n}\mathbf{A}^T\mathbf{A}$ is positive semidefinite since $\mathbf{A}^T\mathbf{A}$ is positive semidefinite. Therefore, the function f is λ-strongly convex.

2 Numerical Method for Linear Support Vector Machine

Codes, Graph.m and Graph _ stochastic.m, are made to create multipul plots in the following sections.

2.1-(a)

The simulation results of GD and GDstr method are shown in Figure 1 and Figure 2. The function f is λ strong convex. Therefore, its convergence rate is linear and can be described with following inequalities.

$$\|\mathbf{x}^k - \mathbf{x}^\star\| \leq (\frac{L-\lambda}{L+\mu})^{\frac{k}{2}} \|\mathbf{x}^0 - \mathbf{x}^\star\| \text{ if } \alpha = \frac{1}{L} \tag{16}$$

$$\|\mathbf{x}^k - \mathbf{x}^\star\| \leq (\frac{L-\lambda}{L+\mu})^{k} \|\mathbf{x}^0 - \mathbf{x}^\star\| \text{ if } \alpha = \frac{2}{L+\lambda} \tag{17}$$

As shown in the inequalities above, we found GDstr has faster convergence rate than GD method although both methods has linear convergence rate. Looling at the figure on the right in Figure 1, we can find linearity of the convergence rate, where the x-axis of the fighre on the right in Figure 1 means the number of iteration. Figure 3 is multipul plot of the simulation results of GD and GDstr method. This figure also shows that GDstr has faster convergence rate.

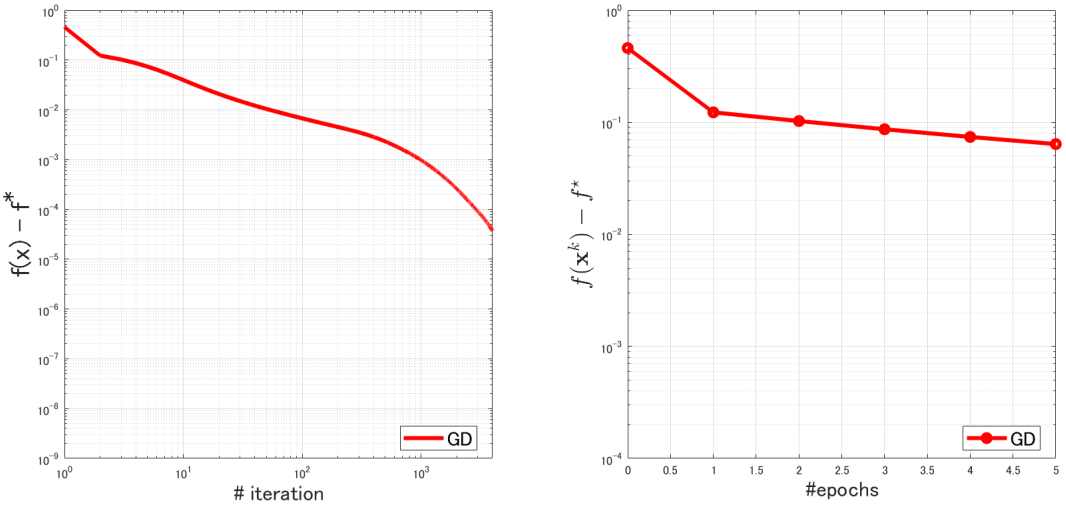

Figure 1: GD

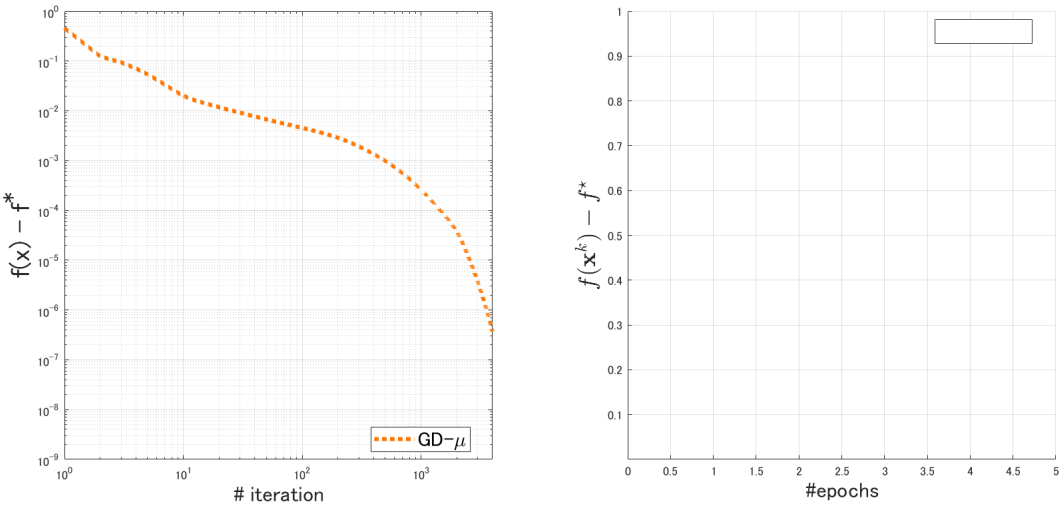

Figure 2: GDstr

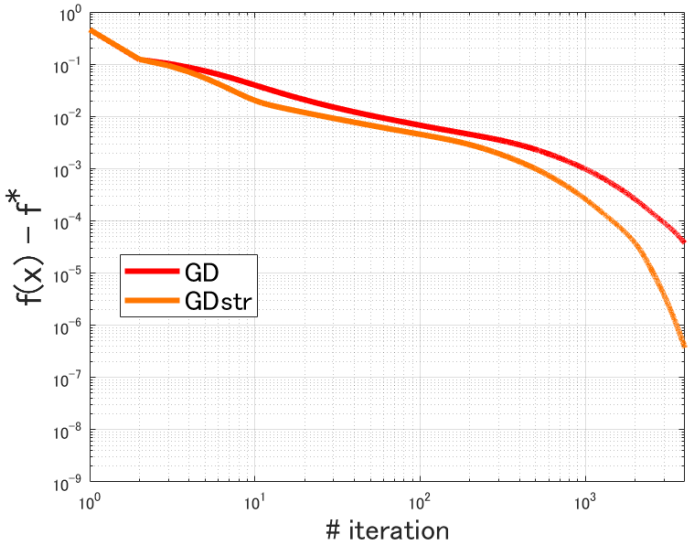

Figure 3: GD and GDstr

2.1-(b)

The simulation results of AGD and AGDstr methods are shown in Figure 4 and Figure 5. Non-monotonicity and ocillatory behavior can be seen in the both figures. Figure 6 is multipul plot of the simulation results of GD, AGD, and AGDstr method. This figure shows that Accelerated method has faster convergence rate than GD method and AGDstr has faster convergence rate than AGD after several iterations.

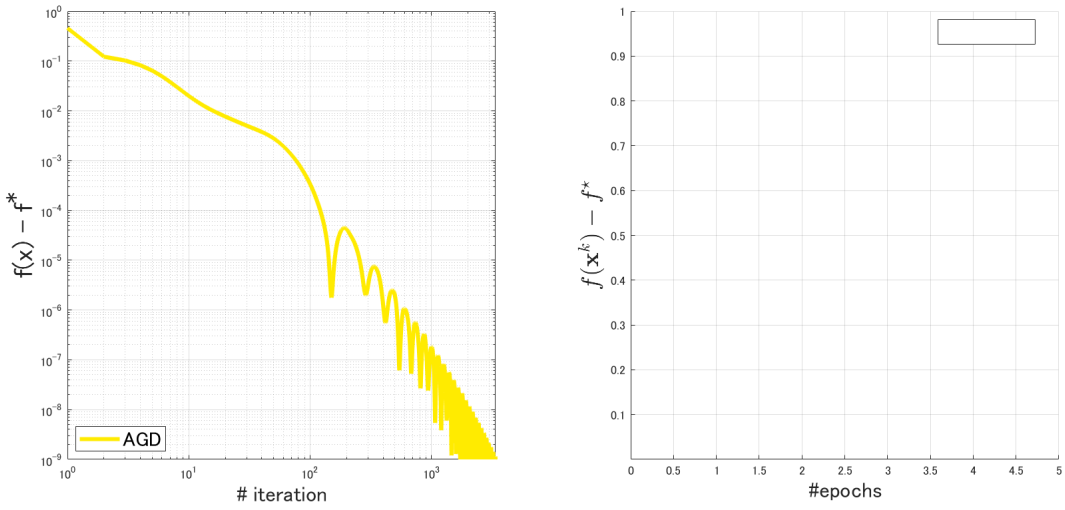

Figure 4: AGD

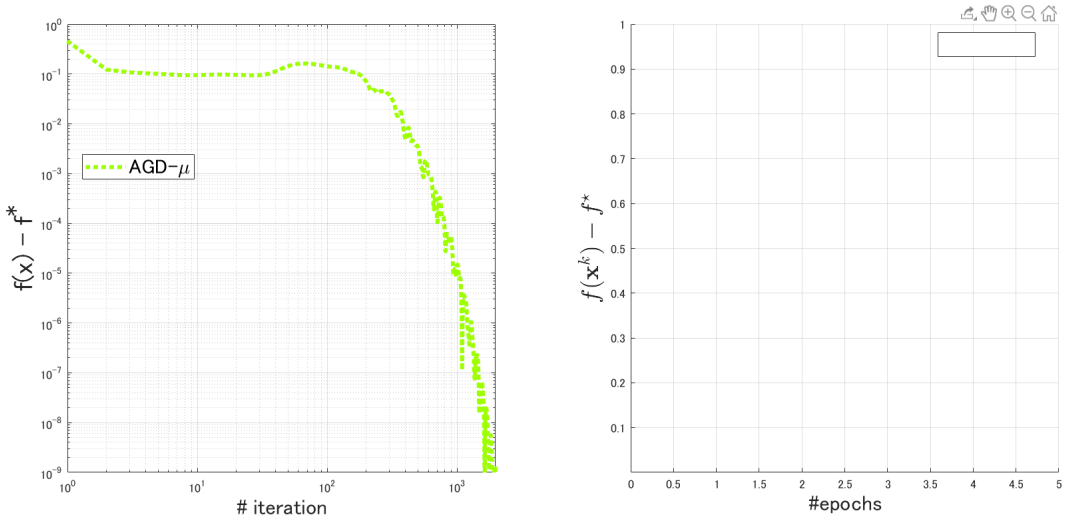

Figure 5: AGDstr

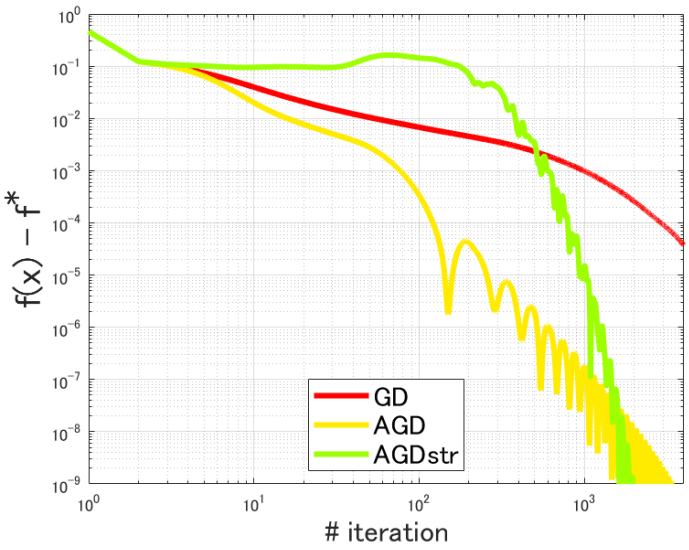

Figure 6: GD, AGD, and AGDstr

2.1-(c)

The simulation results of LSGD and LSAGD methods are shown in Figure 7 and Figure 8. Figure 9 is multipul plot of the simulation results of GD, AGD, LSGD, and LSAGD method. This figure shows that Line-search method finds better constant L at each iteration and improve convergence rate of GD and AGD.

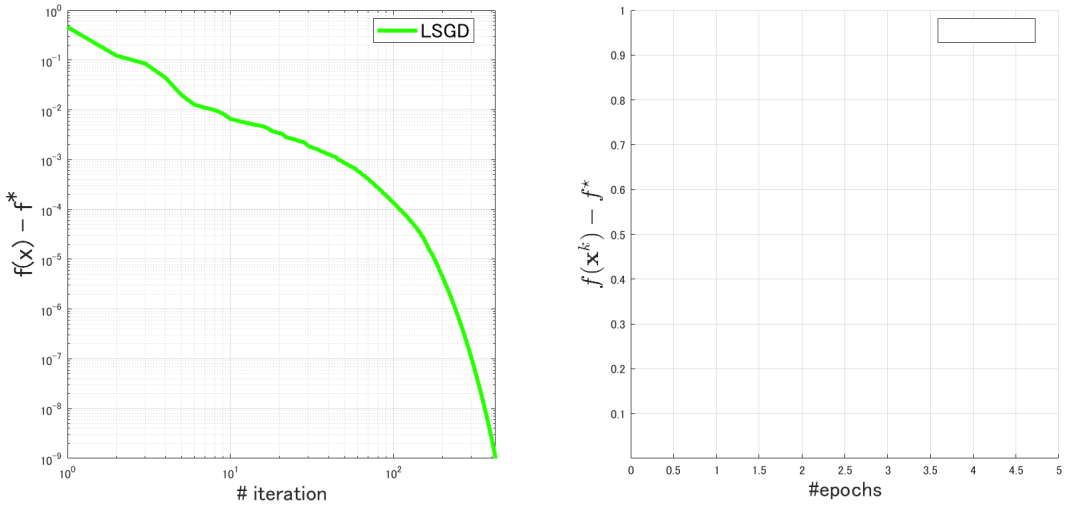

Figure 7: LSGD

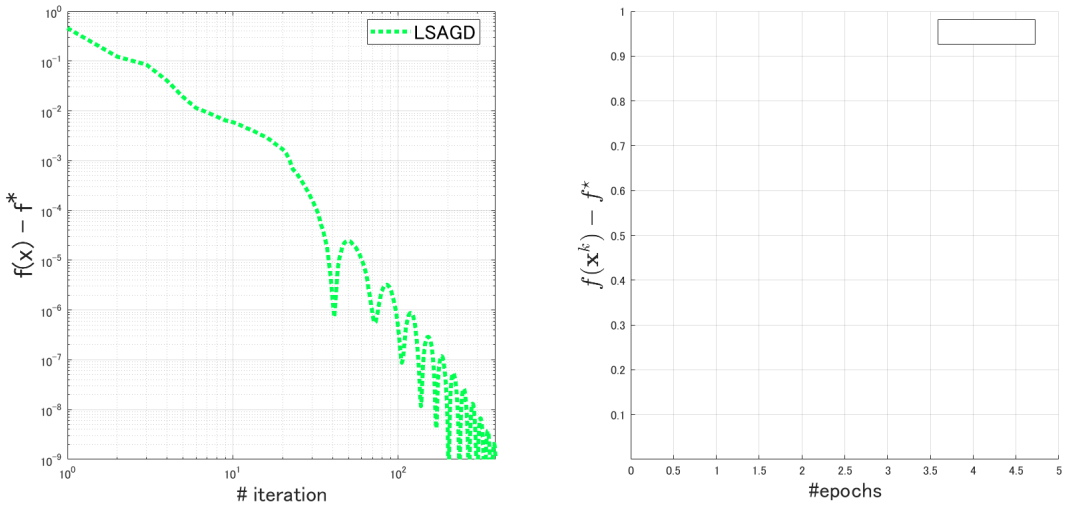

Figure 8: LSAGD

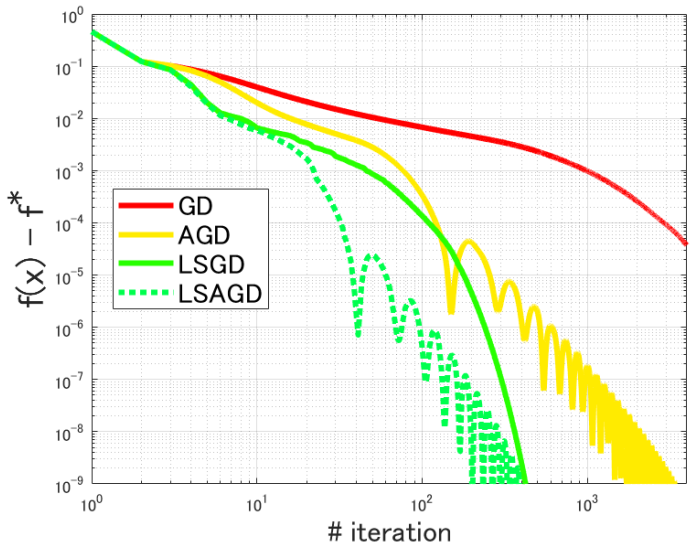

Figure 9: GD, AGD, LSGD, and LSAGD

2.1-(d)

The simulation results of AGDR and LSAGDR methods are shown in Figure 4 and Figure 11. Figure 12 is multipul plot of the simulation results of AGD, AGDR, and LSAGDR method. This figure shows that restart strategy prevent periodic behavior from AGD method and faster convergence rate can be attained by compositing Line-search with restart strategy.

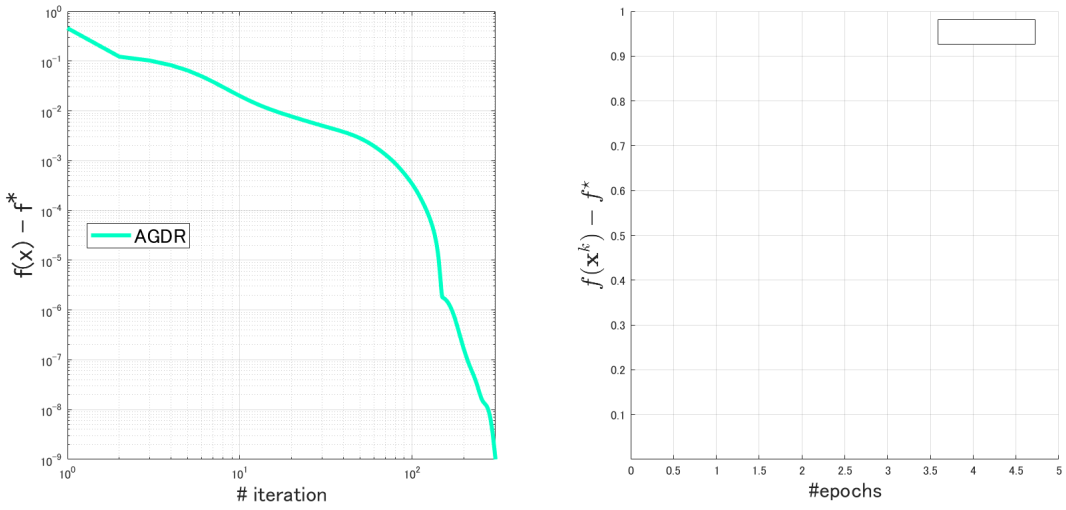

Figure 10: AGDR

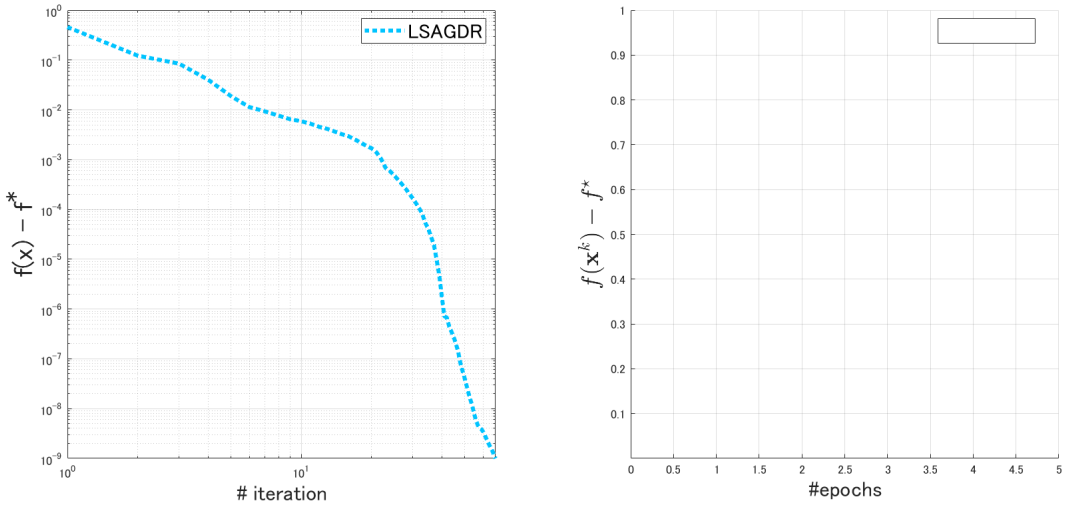

Figure 11: LSAGDR

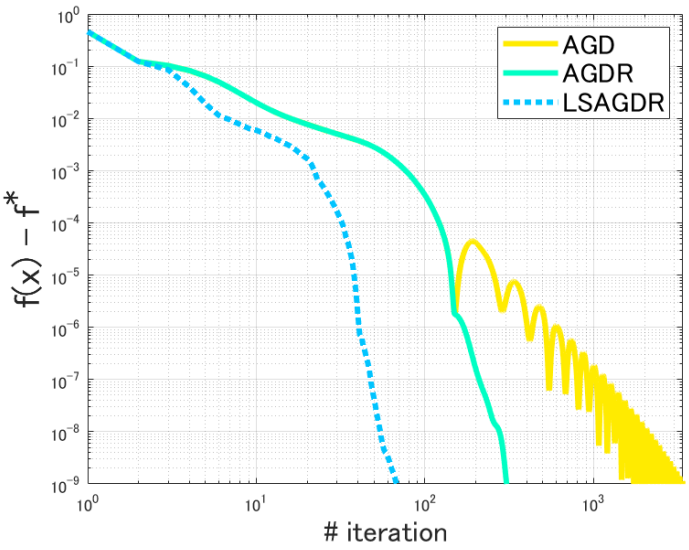

Figure 12: AGD, AGDR, and LSAGDR

2.1-(e)

The simulation result of AdaGrad is shown in Figure 13. Figure 14 is multipul plot of the simulation results of GD and AdaGrad method. This figure indicates the effect of step size adaptation in AdaGrad method. When $f(x)$ is close to $f^\star$ AdaGrad has better convergence rate.

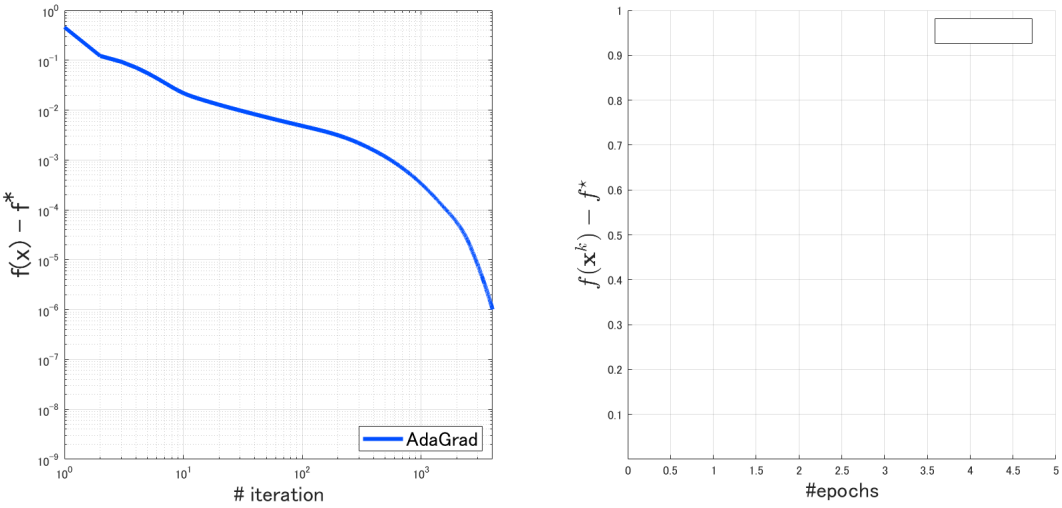

Figure 13: AdaGrad

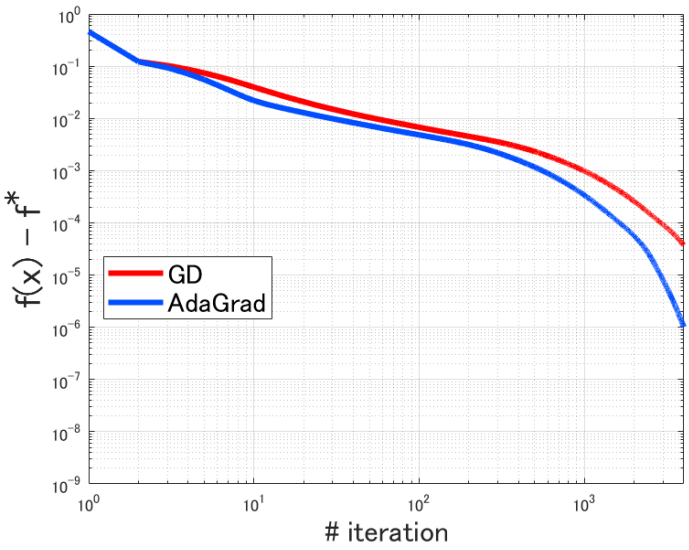

Figure 14: GD and AdaGrad

2.1-(f)

The simulation result of ADAM is shown in Figure 15. Figure 16 is multipul plot of the simulation results of GD, AdaGrad, and ADAM method. This figure indicates ADAM has better convergence rate than AdaGrad When $f(x)$ is close to $f^{\star}$.

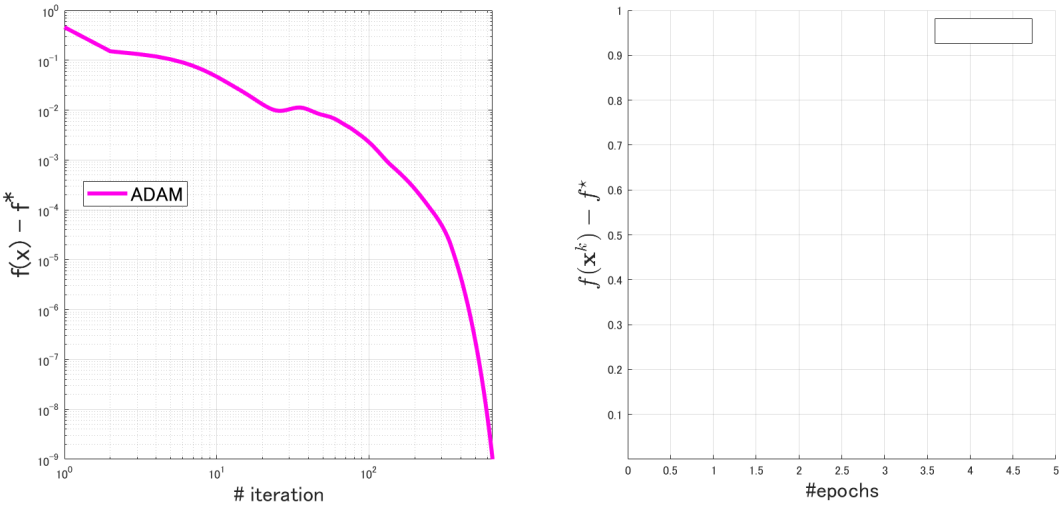

Figure 15: ADAM

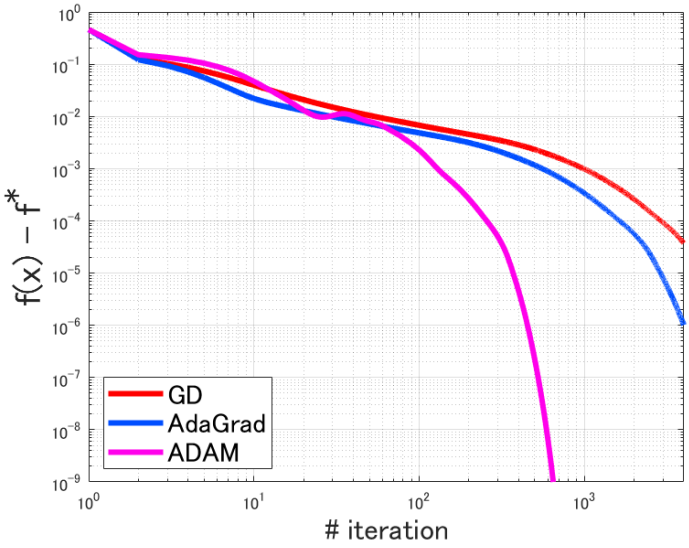

Figure 16: GD, AdaGrad, and ADAM

2.2-(a)

We confirm that the estimation of $\nabla f_{i_k}(\mathbf{x})$ is equal to $\nabla f(\mathbf{x})$ in order to show $\nabla f_{i_k}(\mathbf{x})$ is an unbiased estimation of $\nabla f(\mathbf{x})$. Since we pick $i_k \in \{1, \ldots, n\}$ uniformly at random, the probability we pick i_k is $1/n$ and the estimation of $\nabla f_{i_k}(\mathbf{x})$ is as follows.

$$\begin{aligned} E[\nabla f_{i_k}(\mathbf{x})] =& \frac{1}{n}\sum_{i_k=1}^{n} \nabla f_{i_k}(\mathbf{x}) \\ =& \frac{1}{n}\sum_{i_k=1}^{n} \{\lambda\mathbf{x} + \mathbf{1}_{\{0<b_{i_k}(\mathbf{a}_{i_k}^T\mathbf{x}-b_{i_k})\leq 1\}}\mathbf{a}_{i_k}(\mathbf{a}_{i_k}^T\mathbf{x} - b_{i_k}) - \mathbf{1}_{\{b_{i_k}(\mathbf{a}_i^T\mathbf{x}-b_{i_k})\leq 0\}}b_{i_k}\mathbf{a}_{i_k}\} \\ =& \lambda\mathbf{x} + \frac{1}{n}\sum_{i_k=1}^{n}\mathbf{I}_Q(i_k,i_k)\mathbf{a}_{i_k}({\mathbf{a}_{\mathbf{i_k}}}^T\mathbf{x} - b_{i_k}) - \frac{1}{n}\sum_{i_k=1}^{n}\mathbf{I}_L(i_k,i_k)b_{i_k}\mathbf{a}_{\mathbf{i_k}}, \\ & \text{where the defnition of } \mathbf{I}_Q(i_k,i_k) \text{ and } \mathbf{I}_L(i_k,i_k) \text{ are as in Problem 1.} \end{aligned} \tag{18}$$

From calculation process of equation(6) and equation(7), we obtain

$$\begin{aligned} \nabla f(\mathbf{x}) =& \lambda\mathbf{x} + \frac{1}{n}\sum_{i=1}^{n} b_i\mathbf{a}_i\mathbf{I}_Q(i,i)(b_i\mathbf{a}_i^T\mathbf{x} - 1) - \frac{1}{n}\sum_{i=1}^{n}\mathbf{I}_L(i,i)b_i\mathbf{a}_{\mathbf{i}} \\ =& \lambda\mathbf{x} + \frac{1}{n}\sum_{i=1}^{n}\mathbf{a}_{i_k}\mathbf{I}_Q(i,i)(\mathbf{a}_i^T\mathbf{x} - b_i) - \frac{1}{n}\sum_{i=1}^{n}\mathbf{I}_L(i,i)b_i\mathbf{a}_{\mathbf{i}} \quad (\because b_i^2 = 1). \end{aligned} \tag{19}$$

This means that $E[\nabla f_{i_k}(\mathbf{x})] = \nabla f(\mathbf{x})$ and $\nabla f_{i_k}(\mathbf{x})$ is an unbiased estimation of $\nabla f(\mathbf{x})$.
Here we define $\mathcal{X}_Q = \{\mathbf{x} \in \mathbb{R}^p : 0 < b_{i_k}(\mathbf{a}_{i_k}^T\mathbf{x} - b_{i_k}) \leq 1\}$, $\mathcal{X}_L = \{\mathbf{x} \in \mathbb{R}^p : b_{i_k}(\mathbf{a}_{i_k}^T\mathbf{x} - b_{i_k}) \leq 0\}$, and $\mathcal{X}_O = \{\mathbf{x} \in \mathbb{R}^p : 1 < b_{i_k}(\mathbf{a}_{i_k}^T\mathbf{x} - b_{i_k})\}$. If $\mathbf{x} \in \mathcal{X}_L \cup \mathcal{X}_O$, then ∇f_{i_k} is obviously Lipschitz continuous with $L(f_{i_k}) = \|\mathbf{a}_{i_k}\|^2 + \lambda$. In the following discussion we ensure the Lipschitz continuity of ∇f_{i_k} with the assumption that $\mathbf{x} \in \mathcal{X}_Q$. From problem statement,

$$\nabla f_{i_k} = \lambda\mathbf{x} + \mathbf{a}_{i_k}(\mathbf{a}_{i_k}^T\mathbf{x} - b_{i_k}) \text{ if } \mathbf{x} \in \mathcal{X}_Q. \tag{20}$$

For all $\mathbf{x}, \mathbf{y} \in \mathcal{X}_Q$, we derive

$$\|\nabla f_{i_k}(\mathbf{x}) - \nabla f_{i_k}(\mathbf{y})\| = \|\lambda(\mathbf{x}-\mathbf{y}) + \mathbf{a}_{i_k}\mathbf{a}_{i_k}^T(\mathbf{x}-\mathbf{y})\|. \tag{21}$$

By vietue of properties of norm and Cauchy Schwarz inequality, we obtain following inequality.

$$\begin{aligned}\|\nabla f_{i_k}(\mathbf{x}) - \nabla f_{i_k}(\mathbf{y})\| \leq& \lambda\|\mathbf{x}-\mathbf{y}\| + \|\mathbf{a}_{i_k}\|\|\mathbf{a}_{i_k}^T\|\|\mathbf{x}-\mathbf{y}\| \\ =& \lambda\|\mathbf{x}-\mathbf{y}\| + \|\mathbf{a}_{i_k}\|^2\|\mathbf{x}-\mathbf{y}\|\end{aligned} \tag{22}$$

Thus, ∇f_{i_k} is Lipschitz continuous with $L(f_{i_k}) = \|\mathbf{a_{i_k}}\|^2 + \lambda$.

2.2-(b)

The simulation result of SGD is shown in Figure 17. Figure 18 is multipul plot of the simulation results of GD and SGD method. This figure indicates that converge rate of SGD method decreases faster than GD.

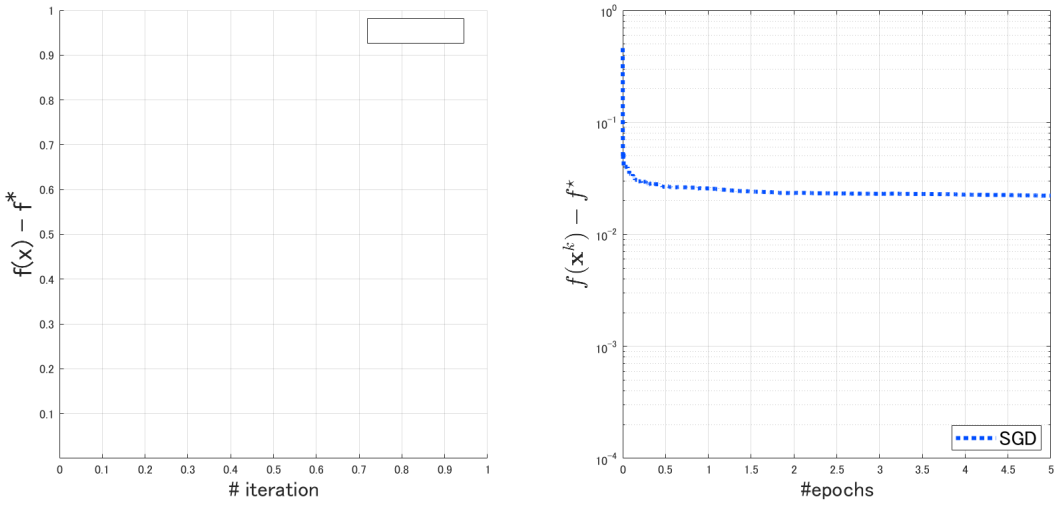

Figure 17: SGD

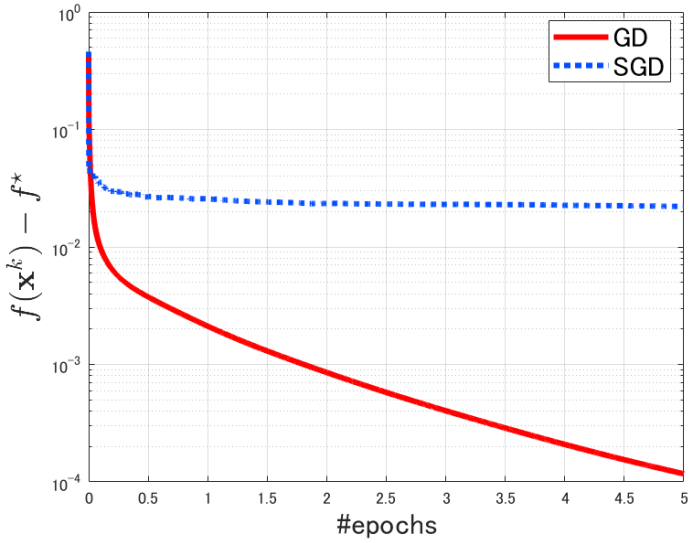

Figure 18: GD and SGD

2.2-(c)

The simulation result of SGD is shown in Figure 19. Figure 20 is multipul plot of the simulation results of SGD and SAG method. This figure indicates that SAG method maintain its converge rate even after a few epochs.

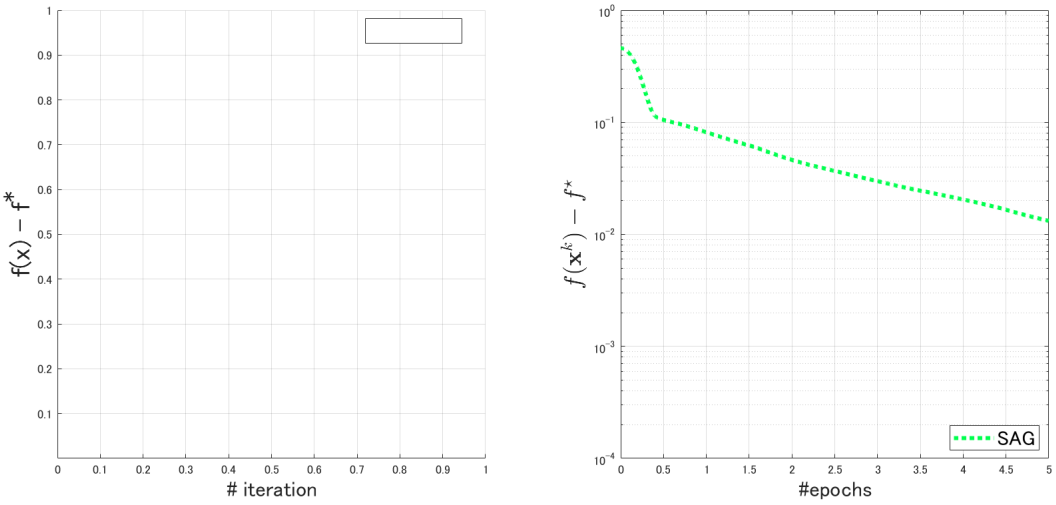

Figure 19: SAG

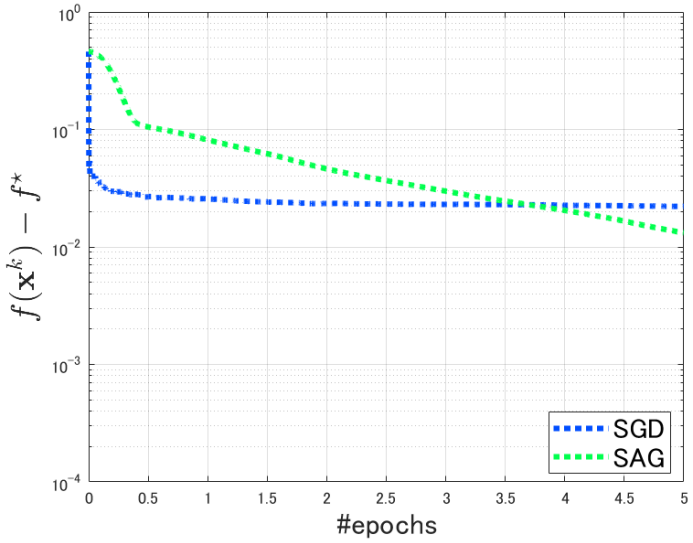

Figure 20: SGD and SAG

2.2-(d)

The simulation result of SVR is shown in Figure 21. With less than half epochs iterations, $f(\mathbf{x}^k)$ quickly converges to the optimal value. However, the computational cost per iteration on this method is high, and SVR method, therefore, takes longer time to finish its simulations than other methods even if the number of iteration is low.

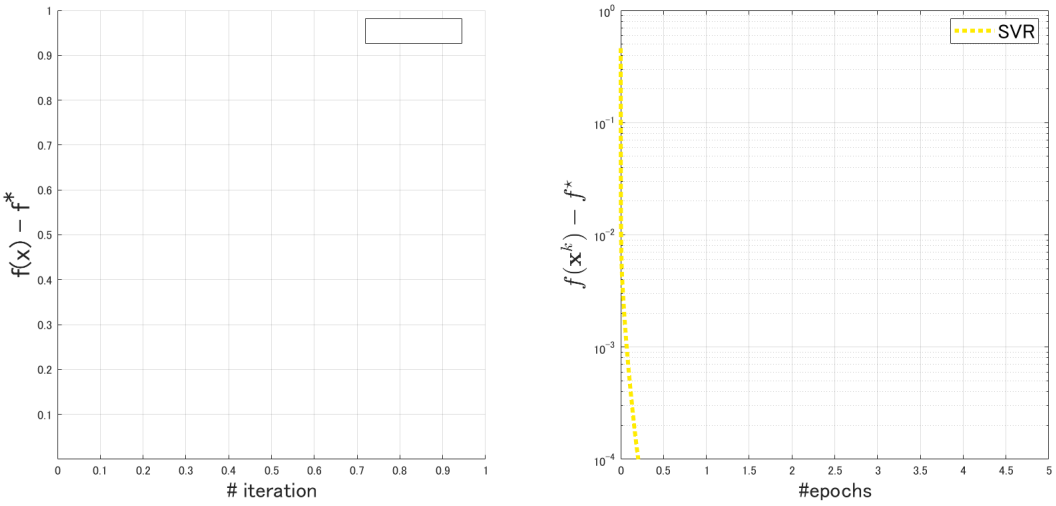

Figure 21: SVR