Homework 5 for "Convex Optimization" Part. 4

1500010611 汪祎非

We consider the $l1$-regularized problem

$$\min_x \frac{1}{2}\|Ax-b\|_2^2 + \mu\|x\|_1 \tag{1}$$

where $A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m$ and $\mu > 0$ are given. We denote $f(x) = \frac{1}{2}\|Ax-b\|_2^2 + \mu\|x\|_1$, $g(x) = \frac{1}{2}\|Ax-b\|_2^2$, $h(x) = \|x\|_1$.

1 Adagrad

We consider to fix the step size to be s and apply continuation strategy. We have three parameters α, M_1, M_2 for continuation strategy. We have one parameter δ for Adagrad. We set $\mu_0 = \max\{\mu, \alpha\|A^T b\|_\infty\}$ and set $i = 0$. We then set $r_0 = 0$. For each μ_i, because $f(x)$ is not smooth, we consider to take the subgradient of $f_i(x) = g(x) + \mu_i h(x)$ to optimize. We denote the subgradient by $p_i(x) = A^T(Ax-b) + \mu_i \text{sign(x)}$. We update x_{k+1} in the following way:

$$\begin{cases} g_k = p_i(x_k) \\ r_{k+1} = r_k + g_k \odot g_k \\ x_{k+1} = x_k - \dfrac{s}{\sqrt{r_{k+1}} + \delta} \odot g_k \end{cases} \tag{Adagrad-upd}$$

If $\mu_i > \mu$, after M_1 iterations, we update $\mu_{i+1} = \max\{\mu, \alpha\mu_i\}$ and $i = i+1$. We then reset $x_0 = x_k$, $r_0 = r_k$ and $k = 0$.

If $\mu_i = \mu$, after M_2 iterations, we stop our algorithm. The algorithm of Adagrad is given below. In practice, we take $s = 1, \alpha = 0.1, M_1 = 280, M_2 = 280, \delta = 10^{-8}$.

Algorithm 1 Adagrad with continuation strategy

Input: t, continuation parameter α, M_1, M_2.
Calculate $\mu_0 = \max\{\alpha\|A^T b\|_\infty, \mu\}$. Let $i = 0$, $r^0 = 0$, $k = 0$.
while $\mu_i > \mu$ **do**
  while $k < M_1$ **do**
    Update (x_{k+1}, r_{k+1}) by (Adagrad-upd), $k = k + 1$
  end while
  $\mu_{i+1} = \max\{\mu, \alpha\mu_i\}$, $i = i + 1$
  Set $x_0 = x_k$, $r_0 = r_k$, $k = 0$.
end while
while $k < M_2$ **do**
  Update (x_{k+1}, r_{k+1}) by (Adagrad-upd), $k = k + 1$
end while
return x_k

2 Adam

We consider to fix the step size to be s and apply continuation strategy. We have three parameters α, M_1, M_2 for continuation strategy. We have three parameters ρ_1, ρ_2, δ for Adam. We set $\mu_0 = \max\{\mu, \alpha\|A^T b\|_\infty\}$ and set $i = 0$. We then set $r_0 = 0, u_0 = 0$. For each μ_i, because $f(x)$ is not smooth, we consider to take the subgradient $p_i(x)$ to optimize. We update x_{k+1} in the following way:

$$\begin{cases} g_k = p_i(x_k) \\ r_{k+1} = \rho_1 r_k + (1-\rho_1) g_k \\ u_{k+1} = \rho_2 u_k + (1-\rho_2) g_k \odot g_k \\ x_{k+1} = x_k - \dfrac{s\sqrt{1-\rho_2^k}}{1-\rho_1^k} \dfrac{r_{k+1}}{\sqrt{u_{k+1}} + \delta} \end{cases} \tag{Adam-upd}$$

Note the operations are applied element-wise.

If $\mu_i > \mu$, after M_1 iterations, we update $\mu_{i+1} = \max\{\mu, \alpha\mu_i\}$ and $i = i + 1$. We then reset $x_0 = x_k, r_0 = r_k, u_0 = u_k$ and $k = 0$.

If $\mu_i = \mu$, after M_2 iterations, we stop our algorithm. The algorithm of Adam is given below.

In practice, we take $s = 0.1, \alpha = 0.5, M_1 = 50, M_2 = 300, \rho_1 = 0.9, \rho_2 = 0.999, \delta = 10^{-8}$.

Algorithm 2 Adam with continuation strategy

Input: t, continuation parameter α, M_1, M_2.

Calculate $\mu_0 = \max\{\alpha\|A^T b\|_\infty, \mu\}$. Let $i = 0$, $r^0 = 0$, $u_0 = 0$, $k = 0$.
while $\mu_i > \mu$ **do**
  while $k < M_1$ **do**
    Update $(x_{k+1}, r_{k+1}, u_{k+1})$ by (Adam-upd), $k = k + 1$
  end while
  $\mu_{i+1} = \max\{\mu, \alpha\mu_i\}$, $i = i + 1$
  Set $x_0 = x_k$, $r_0 = r_k$, $u_0 = u$, $k = 0$.
end while
while $k < M_2$ **do**
  Update $(x_{k+1}, r_{k+1}, u_{k+1})$ by (Adam-upd), $k = k + 1$
end while
return x_k

3 RMSProp

We consider to apply continuation strategy. We have three parameters α, M_1, M_2 for continuation strategy. We have two parameters ρ, δ for RMSProp. We also have two parameters s, β for step size. Here we choose to gradually decay step size. We set $\mu_0 = \max\{\mu, \alpha\|A^T b\|_\infty\}$ and set $i = 0$. We then set $r_0 = 0$, $s_0 = s$. For each μ_i, because $f(x)$ is not smooth, we consider to take the subgradient $p_i(x)$ to optimize. We update x_{k+1} in the following way:

$$\begin{cases} g_k = p_i(x_k) \\ r_{k+1} = \rho r_k + (1-\rho) g_k \odot g_k \\ x_{k+1} = x_k - \dfrac{s_i}{\sqrt{r_{k+1}} + \delta} \odot g_k \end{cases} \tag{RMSProp-upd}$$

Note the operations are applied element-wise.

If $\mu_i > \mu$, after M_1 iterations, we update $\mu_{i+1} = \max\{\mu, \alpha\mu_i\}$, step size $s_{i+1} = \beta s_i$ and $i = i+1$. We then reset $x_0 = x_k$, $r_0 = r_k$, $u_0 = u_k$ and $k = 0$.

If $\mu_i = \mu$, after M_2 iterations, we stop our algorithm. The algorithm of RMSProp is given below.

In practice, we take $s = 0.04, \beta = 0.14, \alpha = 0.1, M_1 = 280, M_2 = 280, \rho = 0.9$.

4 Momentum

We consider to apply continuation strategy. We have three parameters α, M_1, M_2 for continuation strategy. We have two parameters ρ, δ for Momentum. We also have two parameters s, β

Algorithm 3 RMSProp with continuation strategy

Input: t, continuation parameter α, M_1, M_2.

Calculate $\mu_0 = \max\{\alpha\|A^T b\|_\infty, \mu\}$. Let $i = 0$, $r^0 = 0$, $k = 0$.
while $\mu_i > \mu$ **do**
  while $k < M_1$ **do**
    Update (x_{k+1}, r_{k+1}) by (RMSProp-upd), $k = k + 1$
  end while
  $\mu_{i+1} = \max\{\mu, \alpha\mu_i\}$, $s_{i+1} = \beta s_i$, $i = i + 1$
  Set $x_0 = x_k$, $r_0 = r_k$, $k = 0$.
end while
while $k < M_2$ **do**
  Update (x_{k+1}, r_{k+1}) by (RMSProp-upd), $k = k + 1$
end while
return x_k

for step size. Here we choose to gradually decay step size. We set $\mu_0 = \max\{\mu, \alpha\|A^T b\|_\infty\}$ and set $i = 0$. We then set $r_0 = 0$, $s_0 = s$. For each μ_i, because $f(x)$ is not smooth, we consider to take the subgradient $p_i(x)$ to optimize. We update x_{k+1} in the following way:

$$\begin{cases} g_k = p_i(x_k) \\ r_{k+1} = \rho r_k + g_k \\ x_{k+1} = x_k - s_i r_k \end{cases} \tag{Momentum-upd}$$

If $\mu_i > \mu$, after M_1 iterations, we update $\mu_{i+1} = \max\{\mu, \alpha\mu_i\}$, step size $s_{i+1} = \beta s_i$ and $i = i+1$. We then reset $x_0 = x_k$, $r_0 = r_k$, $u_0 = u_k$ and $k = 0$.

If $\mu_i = \mu$, after M_2 iterations, we stop our algorithm. The algorithm of Momentum is given below.

In practice, we take $s = 5 \times 10^{-4}$, $\beta = 0.9$, $\alpha = 0.5$, $M_1 = 50$, $M_2 = 300$, $\rho = 0.9$.

5 Numerical result

The whole test program is named *Test_hw05_02.m*. This time we run all algorithms mentioned before with same A and b. Suppose the value of objective function from cvx mosek is f_c, the value from proposed method is f_p, then the value of *objval to cvx mosek* is $\frac{f_p - f_c}{f_c}$. The numerical result is given in the following tables:

Algorithm 4 Momentum with continuation strategy

Input: t, continuation parameter α, M_1, M_2.

Calculate $\mu_0 = \max\{\alpha\|A^T b\|_\infty, \mu\}$. Let $i = 0$, $r^0 = 0$, $k = 0$.
while $\mu_i > \mu$ **do**
    while $k < M_1$ **do**
        Update (x_{k+1}, r_{k+1}) by (Momentum-upd), $k = k + 1$
    end while
    $\mu_{i+1} = \max\{\mu, \alpha\mu_i\}$, $s_{i+1} = \beta s_i$, $i = i + 1$
    Set $x_0 = x_k$, $r_0 = r_k$, $k = 0$.
end while
while $k < M_2$ **do**
    Update (x_{k+1}, r_{k+1}) by (Momentum-upd), $k = k + 1$
end while
return x_k

Table 1 Random seed is 2. The cpu time of cvx mosek is 1.08

Method	cpu time	objval to cvx mosek	error to cvx mosek
Adagrad	0.97	-2.98×10^{-7}	2.67×10^{-6}
Adam	0.68	-8.30×10^{-7}	2.76×10^{-6}
RMSProp	0.97	-4.05×10^{-7}	2.63×10^{-6}
Momentum	0.61	-1.10×10^{-6}	2.92×10^{-6}

Table 2 Random seed is 7. The cpu time of cvx mosek is 1.06

Method	cpu time	objval to cvx mosek	error to cvx mosek
Adagrad	0.99	-1.85×10^{-6}	2.75×10^{-6}
Adam	0.64	-2.33×10^{-6}	2.87×10^{-6}
RMSProp	0.96	-1.65×10^{-6}	2.72×10^{-6}
Momentum	0.56	-2.45×10^{-6}	2.92×10^{-6}

Table 3 Random seed is 9. The cpu time of cvx mosek is 1.02

Method	cpu time	objval to cvx mosek	error to cvx mosek
Adagrad	0.94	-6.78×10^{-7}	2.19×10^{-6}
Adam	0.70	-1.15×10^{-6}	2.29×10^{-6}
RMSProp	0.94	-4.88×10^{-7}	2.11×10^{-6}
Momentum	0.56	-1.37×10^{-6}	2.35×10^{-6}