Report: Accelerating Stochastic Gradient Descent using Predictive Variance Reduction

Sitao Min

Basic concept about stochastic gradient descent

In machine learning problem, we have feature vector x_i = $(a_{i1}, a_{i2}, a_{i3}, \ldots, a_{id})$, and label vector y_i , and in training dataset we have , $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$,n training examples. We want to find a mapping from x to y, or say find a function $y = f(x, w)$ for prediction and other use, w is parameter vector used for training to find a perfect function to fit x to y. For supervised learning we have a loss function $\psi(w) = f(x, w) - y$, adjusting parameter w to minimize loss function ψ we can find a optimized parameter w and find a perfect fitting $f(x, w)$. So in machine learning, we usually minimize the following problem:

$$minP(w), \qquad P(w) := \frac{1}{n}\sum_{i=1}^{n}\psi_i(w) \qquad (1)$$

The standard method is **gradient descent**, which requires following updat rule for t = 1,2, ...

$$w^{(t)} = w^{(t-1)} - \eta_t \nabla P(w^{(t-1)}) \qquad (2)$$

because gradient descent requires evalution of n derivatives each update step, which is expensive for large data machine learning, so we often use the other method calls **stochastic gradient descent(SGD)**, in which we only randomly select 1 item from $\{\psi_1, \psi_2, \ldots, \psi_n\}$ in each updating step t, we denote this item ψ_{i_t}

$$w^{(t)} = w^{(t-1)} - \eta_t \nabla \psi_{i_t}(w^{(t-1)}) \qquad (3)$$

And SGD, we can maintain the expectation $\mathbb{E}[w^{(t)}|w^{(t-1)}]$ equals to Gradient Descent.

Notice that we can use a random variable ξ_t related to $w^{(t-1)}$to denote the $\nabla\psi_{i_t}(w^{(t-1)})$ as a function $g_t(w^{(t-1)}, \xi_t)$, (**Question2: I can not understand this, why can use random variable to denote this?**)so the update formula can be written as:

$$w^{(t)} = w^{(t-1)} - \eta_t g_t(w^{(t-1)}, \xi_t) \qquad (4)$$

However, the disadvantage of SGD is that it introduce large variance due to the randomness of choosing gradient item, so its convergence rate is much slower and we usually need to adjust learning rate with the iteration of training process to make it converge to local minima(**Question3:in paper this part I don't understand well about how to compute convergence rate and why that SGD convergence in sub-linear rate, especially formula (5) and (6) in this paper**).

Stochastic Variance Reduced Gradient(SVRG)

Main idea of SVRG is to reduce the variance of randomness by adding another item in SGD update rule. The step is to keep a snapshot of $\tilde{w}$ after every m iteration of SGD, ($\tilde{w}$ can be chosen randomly from m iteration's w or last ieration's w from each m iteration, which means keep $\tilde{w}$ close to w), and then maintain the average gradient of $\tilde{w}$, denoted as $\tilde{\mu} = \nabla P(\tilde{w}) = \frac{1}{n}\sum_{i=1}^{n} \nabla\psi_i(\tilde{w})$, and add $(\tilde{\mu} - \nabla\psi_i(\tilde{w}))$ to SGD update rules as follow:

$$w^{(t)} = w^{(t-1)} - \eta_t(\nabla\psi_{i_t}(w^{(t-1)}) - \nabla\psi_{i_t}(\tilde{w}) + \tilde{\mu}) \quad (4)$$

(**Question4: actually, I don't know how they come up with this idea and why it can be feasible?**) Notice that the expectation $\mathbb{E}[w^{(t)}|w^{(t-1)}] = w^{t-1} - \eta_t\nabla P(w^{t-1})$ which is the same as SGD, so this method reduce the variance but maintain the close training result with SGD.

The total process of SVRG is as follow

```
SVRG

Parameters update frequency $m$ and learning rate $\eta$

Initialize $\tilde{w_0}$

Iterate:for $s = 1, 2, 3, \dots$

$\tilde{w} = w\tilde{}_{s-1}$

$\tilde{\mu} = \frac{1}{n}\sum_{i=1}^{n} \nabla\psi_i(\tilde{w})$

$w_0 = w$

Iterate:for $t = 1, 2, \dots, m$

Randomly picj $i_t \in \{1, \dots, n\}$ and update weight

$w^{(t)} = w^{(t-1)} - \eta_t(\nabla\psi_{i_t}(w^{(t-1)}) - \nabla\psi_{i_t}(\tilde{w}) + \tilde{\mu})$

end

option I set $\tilde{w_s} = w_m$

option II set $\tilde{w_s} = w_t$ for randomly chosen $t \in \{0, \dots, m-1\}$
```

SVRG compute at each stage $2m + n$ gradient, which is slightly more than the standard SGD which compute m gradient each stage. Suppose that total iteration of SGD is M, the running time of SGD is $O(M)$ and running time of SVRG is $O(2M + \frac{nM}{m})$, if we choose m as same order of n ($\frac{m}{n} = const$), then SVRG is $O((2 + const)M) = O(M)$, so it is not slower than SGD.

Analysis of SVRG

They prove that
$the\ geometric\ convergence\ in\ expectation\ for\ SVRG\ is\ \mathbb{E}\ P(\tilde{w_s}) \leq \mathbb{E}\ P(w_*) + \alpha^s[P(\tilde{w_0} - P(w_*))]$

and $\alpha \leq 1$ (**Question5 cannot understand what the meaning of their conclusion especially they say that their time comlexity is $nln(1/\epsilon)$ and same as SGD in the middle of page 5**)

Conclusion

This paper :

1.Introduces an explicit variance reduction method called SVRG, which does not require the storage of gradients.

2.They prove that for smooth and strongly convex functions, SVRG enjoys the same fast convergence rate as those of SDCA and SAG.

3. They do some computer simulation to prove their result.

My questons:

Cannot understand their proof and result of time bound of SVRG

Cannot understand what this SVRG has done, how they come up with this idea.