A Perturb Iterate Analysis of Frank Wolfe Type Methods

While our algorithms and the analysis can be applied to general θ, we focus on the case when $\theta = 1$ for notation simplicity. Note there could be multiple optimal $\boldsymbol{X}$, and hence we use $\boldsymbol{X}^*$ to denote one of the optimal $\boldsymbol{X}$ out of the whole optimal set, unless otherwise specified. Let's start with a general definition of Frank-Wolfe Method framework. The update iteration of Frank-Wolfe Method framework is

$$\begin{aligned} \boldsymbol{U}_k &= \arg\min_{\|\boldsymbol{U}\|_* \leq 1} \quad \langle \tilde{\nabla}_{k-1}, \boldsymbol{U} \rangle, \\ \boldsymbol{X}_k &= (1-\eta_k)\boldsymbol{X}_{k-1} + \eta_k \boldsymbol{U}_k, \end{aligned} \tag{12}$$

where $\tilde{\nabla}_{k-1}$ is the estimate of the gradient at point $\boldsymbol{X}_{k-1}$, and η_k is the step size at iterate k. For example, for SFW, $\tilde{\nabla}_{k-1}$ is $\frac{1}{|S|}\sum_{i\in S}\nabla f_i(\boldsymbol{X}_{k-1})$, for SVRF $\tilde{\nabla}_{k-1}$ is the variance reduced stochastic gradient (Hazan and Luo, 2016), and so on.

Note that the function F is convex and L-smooth.

$$\begin{aligned} F(\boldsymbol{X}_k) &\leq F(\boldsymbol{X}_{k-1}) + \langle \nabla F(\boldsymbol{X}_{k-1}), \boldsymbol{X}_k - \boldsymbol{X}_{k-1} \rangle + \frac{L}{2}\|\boldsymbol{X}_k - \boldsymbol{X}_{k-1}\|^2 && \text{(by smoothness)} \\ &\leq F(\boldsymbol{X}_{k-1}) + \eta_k \langle \nabla F(\boldsymbol{X}_{k-1}), \boldsymbol{U}_k - \boldsymbol{X}_{k-1} \rangle + \frac{L\eta_k^2}{2}\|\boldsymbol{U}_k - \boldsymbol{X}_{k-1}\|^2 && \text{(by Eq. (12))} \\ &\leq F(\boldsymbol{X}_{k-1}) + \eta_k \langle \nabla F(\boldsymbol{X}_{k-1}), \boldsymbol{U}_k - \boldsymbol{X}_{k-1} \rangle + \frac{L\eta_k^2 D^2}{2} && \text{(by Definition of D)} \end{aligned}$$

Let's look at the second term, which is the focus of our analysis.

$$\begin{aligned} &\langle \nabla F(\boldsymbol{X}_{k-1}), \boldsymbol{U}_k - \boldsymbol{X}_{k-1} \rangle \\ =&\langle \nabla F(\boldsymbol{X}_{k-1}) - \tilde{\nabla}_{k-1}, \boldsymbol{U}_k - \boldsymbol{X}_{k-1} \rangle + \langle \tilde{\nabla}_{k-1}, \boldsymbol{U}_k - \boldsymbol{X}_{k-1} \rangle \\ \leq&\langle \nabla F(\boldsymbol{X}_{k-1}) - \tilde{\nabla}_{k-1}, \boldsymbol{U}_k - \boldsymbol{X}_{k-1} \rangle + \langle \tilde{\nabla}_{k-1}, \boldsymbol{X}^* - \boldsymbol{X}_{k-1} \rangle && \text{(by Eq. (12))} \\ =&\langle \nabla F(\boldsymbol{X}_{k-1}) - \tilde{\nabla}_{k-1}, \boldsymbol{U}_k - \boldsymbol{X}^* \rangle + \langle \nabla F(\boldsymbol{X}_{k-1}), \boldsymbol{X}^* - \boldsymbol{X}_{k-1} \rangle \\ \leq&F(\boldsymbol{X}^*) - F(\boldsymbol{X}_{k-1}) + \langle \nabla F(\boldsymbol{X}_{k-1}) - \tilde{\nabla}_{k-1}, \boldsymbol{U}_k - \boldsymbol{X}^* \rangle. && \text{(by the convexity of } F) \end{aligned}$$

Let $h_k = F(\boldsymbol{X}_k) - F(\boldsymbol{X}^*)$ and by putting everything together we get,

$$h_k \leq (1-\eta_k)h_{k-1} + \eta_k \left[\langle \nabla F(\boldsymbol{X}_{k-1}) - \tilde{\nabla}_{k-1}, \boldsymbol{U}_k - \boldsymbol{X}^* \rangle + \frac{L\eta_k D^2}{2} \right]. \tag{13}$$

For the last term in the bracket, we call it residual. Note that $\frac{L\eta_k D^2}{2}$ can be controlled by diminishing step size η_k. The only remaining term to bound is $\langle \nabla F(\boldsymbol{X}_{k-1}) - \tilde{\nabla}_{k-1}, \boldsymbol{U}_k - \boldsymbol{X}^* \rangle$, **which captures the inexactness of the gradient estimation**. If the gradient is exact, that is, $\langle \nabla F(\boldsymbol{X}_{k-1}) - \tilde{\nabla}_{k-1}, \boldsymbol{U}_k - \boldsymbol{X}^* \rangle = 0$, the proof is then simple. One can plug in $\eta_k = 2/(k+1)$ and show by induction that $h(k) \leq 4LD^2/(k+2)$. Indeed for our analysis we would like to have this term decrease with the rate of $\mathcal{O}(1/k)$, in order to retain the $\mathcal{O}(1/k)$ convergence rate.

A.1 Convergence of SFW-asyn

For Asyn-SFW, consider the worst case when a worker sends an update $\boldsymbol{u}\boldsymbol{v}^T$ based on $\boldsymbol{X}_\tau$. That is,

$$\tilde{\nabla}_{k-1} = \frac{1}{m_{k-\tau}} \sum_{i \in S_a} \nabla f_i(\boldsymbol{X}_{k-\tau}), \tag{14}$$

for some sample set S_a. We first setup an upper bound for $\langle \nabla F(\boldsymbol{X}_{k-1}) - \tilde{\nabla}_{k-1}, \boldsymbol{U}_k - \boldsymbol{X}^* \rangle$. Then we combine it with Eq. (13) to complete the proof for Theorem 1.

Lemma 1. *Under the same assumptions of Theorem 1, let $\tilde{\nabla}_{k-1}$ be the gradient estimate defined in Eq. (14), and $\boldsymbol{U}_k$ is defined in Eq. (12), then we can bound the inexactness of the gradient estimation for SFW-asyn:*

$$\mathbb{E}\left[\langle \nabla F(\boldsymbol{X}_{k-1}) - \tilde{\nabla}_{k-1}, \boldsymbol{U}_k - \boldsymbol{X}^* \rangle \right] \leq \frac{GD}{\sqrt{m_{k-\tau}}} + L\tau\eta_{k-\tau}D^2$$

Proof.

$$\begin{aligned}
&\langle \nabla F(\boldsymbol{X}_{k-1}) - \tilde{\nabla}_{k-1}, \boldsymbol{U}_k - \boldsymbol{X}^* \rangle \\
\leq &\|\nabla F(\boldsymbol{X}_{k-1}) - \tilde{\nabla}_{k-1}\|_F \|\boldsymbol{U}_k - \boldsymbol{X}^*\|_F \\
\leq &D\|\nabla F(\boldsymbol{X}_{k-1}) - \tilde{\nabla}_{k-1}\|_F \\
= &\left\| \frac{1}{m_{k-\tau}} \sum_{i \in S_a} \nabla f_i(\boldsymbol{X}_{k-\tau}) - \nabla F(\boldsymbol{X}_{k-\tau}) + \nabla F(\boldsymbol{X}_{k-\tau}) - \nabla F(\boldsymbol{X}_{k-1}) \right\|_F \cdot D \\
\leq &\left\| \frac{1}{m_{k-\tau}} \sum_{i \in S_a} \nabla f_i(\boldsymbol{X}_{k-\tau}) - \nabla F(\boldsymbol{X}_{k-\tau}) \right\|_F \cdot D + \|\nabla F(\boldsymbol{X}_{k-\tau}) - \nabla F(\boldsymbol{X}_{k-1})\|_F \cdot D
\end{aligned}$$

Hence the gradient inexactness is decomposed into two parts: the inexactness resulted from stochastic batch gradient (the first term), and the staleness (the second term). We increase the batch size and hence obtain a decrease rate of $1/k$ on the first part.

$$\mathbb{E} \left\| \frac{1}{m_{k-\tau}} \sum_{i \in S_a} \nabla f_i(\boldsymbol{X}_{k-\tau}) - \nabla F(\boldsymbol{X}_{k-\tau}) \right\|_F^2 \leq \frac{G^2}{m_{k-\tau}},$$

and therefore

$$\mathbb{E} \left\| \frac{1}{m_{k-\tau}} \sum_{i \in S_a} \nabla f_i(\boldsymbol{X}_{k-\tau}) - \nabla F(\boldsymbol{X}_{k-\tau}) \right\|_F \leq \frac{G}{\sqrt{m_{k-\tau}}},$$

by Jensen's inequality. For the stochastic inexactness, we have to control it by increasing batch size. Note that for some applications when the variance of the stochastic gradient shrinks as the algorithm approach the optimal point, the requirement of increasing batch size can be waived.

For the staleness:

$$\begin{aligned}
&\|\nabla F(\boldsymbol{X}_{k-1}) - \nabla F(\boldsymbol{X}_{k-\tau})\|_F \\
\leq &L\|\boldsymbol{X}_{k-1} - \boldsymbol{X}_{k-\tau}\|_F && \text{(by smoothness)} \\
= &L\|\boldsymbol{X}_{k-1} - \boldsymbol{X}_{k-2} + \boldsymbol{X}_{k-2} + ... - \boldsymbol{X}_{k-\tau}\|_F \\
\leq &L \sum_{a=1}^{\tau+1} \|\boldsymbol{X}_{k-a} - \boldsymbol{X}_{k-a-1}\|_F && \text{(by triangular inequality)} \\
\leq &L \sum_{a=1}^{\tau+1} \eta_{k-a} \|\boldsymbol{U}_{k-a} - \boldsymbol{X}_{k-a-1}\|_F && \text{(by Eq. (12))} \\
\leq &L\tau\eta_{k-\tau} D.
\end{aligned}$$

Combining the above two terms we finish the proof. □

Now back to the proof of Theorem 1.

Proof of Thm 1. We finish the proof by induction. The case of $k = 1$ is obviously true:

$$\begin{aligned}
F(\boldsymbol{X}_1) - F^* &\leq \langle \nabla F(\boldsymbol{X}^*), \boldsymbol{X}_1 - \boldsymbol{X}^* \rangle + \frac{L}{2}\|\boldsymbol{X}_1 - \boldsymbol{X}^*\|^2 && \text{(by smoothness)} \\
&\leq 0 + \frac{LD^2}{2}.
\end{aligned}$$

Suppose $\mathbb{E}[h_k] \le \frac{(4\tau+3)2LD^2}{k+2}$ for $k \le T-1$. Then

$$\begin{aligned}
\mathbb{E}[h_T] \le & \mathbb{E}\left[(1-\eta_T)h_{T-1} + \eta_T\left(\langle \nabla F(\boldsymbol{X}_{T-1}) - \tilde{\nabla}_{T-1}, \boldsymbol{U}_T - \boldsymbol{X}^*\rangle + \frac{L\eta_T D^2}{2}\right)\right] \\
\le & \mathbb{E}[(1-\eta_T)h_{T-1}] + \eta_T\left(\frac{GD}{\sqrt{m_{T-\tau}}} + L\tau\eta_{T-\tau}D^2 + \frac{L\eta_T D^2}{2}\right) && \text{(by Lemma 1)} \\
= & \mathbb{E}\left[\frac{T-1}{T+1}h_{T-1}\right] + \frac{2}{T+1}\left[\frac{\tau LD^2}{T+1-\tau} + \frac{2\tau LD^2}{T+1-\tau} + \frac{LD^2}{T+1}\right] && \text{(plug in } m_T \text{ and } \eta_T\text{)} \\
\le & \frac{T-1}{T+1}\frac{(3\tau+1)4LD^2}{T+1} + \frac{2}{T+1}\left[\frac{\tau LD^2}{T+1-\tau} + \frac{2\tau LD^2}{T+1-\tau} + \frac{LD^2}{T+1}\right] && \text{(recursion condition)} \\
= & \frac{4LD^2}{(T+1)^2}\left[(3\tau+1)(T-1) + \frac{T+1}{T+1-\tau}\left(\tau + \frac{\tau}{2}\right) + \frac{1}{2}\right] \\
\le & \frac{4LD^2}{(T+1)^2}\left[(3\tau+1)(T-1) + 3\tau + 1\right] && (\tau < T/2) \\
= & \frac{4LD^2}{(T+1)^2}\left[(3\tau+1)T\right] \\
\le & \frac{(3\tau+1)4LD^2}{(T+2)} && \text{(since } \frac{T}{(T+1)^2} < \frac{1}{T+2}\text{)}
\end{aligned}$$

□

A.2 Convergence of SVRF-asyn

For SVRF-asyn, consider the worst case when a worker sends an update $\boldsymbol{u}\boldsymbol{v}^T$ based on $\boldsymbol{X}_\tau$. That is,

$$\tilde{\nabla}_{k-1} = \frac{1}{m_{k-\tau}}\sum_{i\in S_a}[\nabla f_i(\boldsymbol{X}_{k-\tau}) - \nabla f_i(\boldsymbol{W})] + \nabla F(\boldsymbol{W}), \tag{15}$$

for some sample set S_a.

Before we get started, let's get prepared with some standard lemma on variance reduced algorithms.

Lemma 2. ***(Lemma 1 restated in (Hazan and Luo, 2016))*** *For $\boldsymbol{X}, \boldsymbol{W}$ such that $\|\boldsymbol{X}\|_* = \|\boldsymbol{W}\|_* = 1$*

$$\begin{aligned}
&\mathbb{E}\left[\|\nabla f_i(\boldsymbol{X}) - \nabla f_i(\boldsymbol{W}) + \nabla F(\boldsymbol{W}) - \nabla F(\boldsymbol{X})\|^2\right] \\
\le & 6L\left(2\,\mathbb{E}[F(\boldsymbol{X}) - F(\boldsymbol{W}^*)] + \mathbb{E}[F(\boldsymbol{W}) - F(\boldsymbol{W}^*)]\right)
\end{aligned}$$

Lemma 3. *Under the same assumptions as in Theorem 2, let $\tilde{\nabla}_{k-1}$ be the gradient estimate defined in Eq. (15), $\boldsymbol{U}_k$ defined as in Eq. (12). If*

$$\mathbb{E}\|\nabla F(\boldsymbol{X}_{k-1}) - \tilde{\nabla}_{k-1}\|_F < \frac{15\tau LD}{k+1-\tau}$$

for $k \le T$. Then for $k \le T$

$$\mathbb{E}[F(\boldsymbol{X}_k) - F^*] \le \frac{(15\tau+1)\cdot 4LD^2}{k+1}$$

Proof. The proof is similar to the proof for Theorem 1. Let $h_k = F(\boldsymbol{X}_k) - F^*$. We prove by induction.

$$
\begin{aligned}
\mathbb{E}\left[h_T\right] \leq & \mathbb{E}\left[(1-\eta_T)h_{T-1} + \eta_T\left(\langle \nabla F(\boldsymbol{X}_{T-1}) - \tilde{\nabla}_{T-1}, \boldsymbol{U}_T - \boldsymbol{X}^*\rangle + \frac{L\eta_T D^2}{2}\right)\right] \\
= & \mathbb{E}\left[\frac{T-1}{T+1}h_{T-1}\right] + \frac{2}{T+1}\left[\frac{LD^2(15\tau)}{T+1-\tau} + \frac{LD^2}{T+1}\right] && \text{(plug in } m_T \text{ and } \eta_T\text{)} \\
\leq & \frac{T-1}{T+1}\frac{(15\tau+1)4LD^2}{T+1} + \frac{2}{T+1}\left[\frac{LD^2(15\tau)}{T+1-\tau} + \frac{LD^2}{T+1}\right] && \text{(plug in } h_{T-1}\text{)} \\
= & \frac{4LD^2}{(T+1)^2}\left[(15\tau+1)(T-1) + \frac{1}{2} + \frac{T+1}{T+1-\tau}\frac{15}{2}\tau\right] \\
\leq & \frac{4LD^2}{(T+1)^2}\left[(15\tau+1)(T-1) + \frac{1}{2} + 15\tau\right] && (\tau < T/2) \\
\leq & \frac{4LD^2}{(T+1)^2}\left[(15\tau+1)T\right] \\
\leq & \frac{(15\tau+1)4LD^2}{(T+2)} && \text{(since } \frac{T}{(T+1)^2} < \frac{1}{T+2}\text{)}
\end{aligned}
$$

□

The remain task is to setup an upper bound for $\|\nabla F(\boldsymbol{X}_{k-1}) - \tilde{\nabla}_{k-1}\|_F$.

Lemma 4. *Under the same assumptions as in Theorem 2, let $\tilde{\nabla}_{k-1}$ be the gradient estimate defined in Eq.*(15)*, $\boldsymbol{U}_k$ defined as in Eq.*(12)*, then we can bound the inexactness of the gradient estimation for SVRF-asyn:*

$$\mathbb{E}\,\|\nabla F(\boldsymbol{X}_{k-1}) - \tilde{\nabla}_{k-1}\|_F < \frac{LD(15\tau)}{k+1-\tau}$$

Proof. For notation simplicity, we denote ∇_{k-1} as the variance reduced gradient. That is,

$$\nabla_{k-1} = \frac{1}{m_{k-1}}\sum_{i\in S_b}\left(\nabla f_i(\boldsymbol{X}_{k-1}) - \nabla f_i(\boldsymbol{W})\right) + \nabla F(\boldsymbol{W})$$

We show this lemma by induction. The $k=1$ case is obvious. Suppose for $k < T-1$,

$$\mathbb{E}\,\|\nabla F(\boldsymbol{X}_{k-1}) - \tilde{\nabla}_{k-1}\|_F < \frac{LD(15\tau)}{k+1-\tau}.$$

By above lemma we have

$$\mathbb{E}[F(\boldsymbol{X}_k) - F(\boldsymbol{W}^*)] \leq \frac{(15\tau+1)\cdot 4LD^2}{k+1}.$$

Then

$$\|\nabla F(\boldsymbol{X}_{T-1}) - \tilde{\nabla}_{T-1}\|_F \leq \|\nabla F(\boldsymbol{X}_{T-1}) - \nabla_{T-1}\|_F + \|\nabla_{T-1} - \tilde{\nabla}_{T-1}\|_F.$$

Bounding $\|\nabla F(\boldsymbol{X}_{T-1}) - \nabla_{T-1}\|_F$:

$$
\begin{aligned}
& \mathbb{E}\,\|\nabla F(\boldsymbol{X}_{T-1}) - \nabla_{T-1}\|_F^2 \\
\leq & \frac{6L}{m_{T-1}}\left(2\,\mathbb{E}[F(\boldsymbol{X}_{T-1}) - F(\boldsymbol{W}^*)] + \mathbb{E}[F(\boldsymbol{W}) - F(\boldsymbol{W}^*)]\right) \\
\leq & \frac{6L}{m_{T-1}}\left(\frac{8LD^2}{T+1} + \frac{LD^2}{2^t}\right)\cdot(15\tau+1) \\
\leq & \frac{6L}{m_{T-1}}\left(\frac{8LD^2}{T+1} + \frac{8LD^2}{T+1}\right)\cdot(15\tau+1) \\
= & \frac{L^2D^2}{(T+1)^2}\cdot(15\tau^2+\tau) \\
\leq & \frac{L^2D^2}{(T+1)^2}\cdot 16\tau^2
\end{aligned}
$$

Bounding $\|\nabla_{T-1} - \tilde{\nabla}_{T-1}\|_F$:

$$\begin{aligned}
&\mathbb{E}\,\|\nabla_{T-1} - \tilde{\nabla}_{T-1}\|_F \\
=&\mathbb{E}\,\|\nabla_{T-1} - \nabla F(\boldsymbol{X}_{T-1}) + \nabla F(\boldsymbol{X}_{T-\tau}) - \tilde{\nabla}_{T-1} + \nabla F(\boldsymbol{X}_{T-1}) - \nabla F(\boldsymbol{X}_{T-\tau})\|_F \\
\leq&\mathbb{E}\,\|\nabla_{T-1} - \nabla F(\boldsymbol{X}_{T-1})\|_F + \mathbb{E}\,\|\nabla F(\boldsymbol{X}_{T-\tau}) - \tilde{\nabla}_{T-1}\|_F + \mathbb{E}\,\|\nabla F(\boldsymbol{X}_{T-1}) - \nabla F(\boldsymbol{X}_{T-\tau})\|_F \\
\leq&\frac{LD}{(T+1)} \cdot \sqrt{16\tau^2} + \frac{LD}{(T+1-\tau)} \cdot \sqrt{16\tau^2} + \frac{2LD\tau}{(T+1-\tau)} \\
\leq&\frac{LD}{(T+1-\tau)} \cdot (10\tau)
\end{aligned}$$

where the first inequality follows from the triangular inequality and the second inequality follows from Jensen's inequality, the bound of $\|\nabla F(\boldsymbol{X}_{T-1}) - \nabla_{T-1}\|_F$, and the intermediate result in Lemma 1.

Therefore

$$\begin{aligned}
&\mathbb{E}\,\|\nabla F(\boldsymbol{X}_{T-1}) - \tilde{\nabla}_{T-1}\|_F \\
\leq&\mathbb{E}\,\|\nabla F(\boldsymbol{X}_{T-1}) - \nabla_{T-1}\|_F + \mathbb{E}\,\|\nabla_{T-1} - \tilde{\nabla}_{T-1}\|_F \\
\leq&\frac{LD}{(T+1)} \cdot 4\tau + \frac{LD}{(T+1-\tau)} \cdot 10\tau \\
<&\frac{LD(15\tau)}{T+1-\tau}
\end{aligned}$$

where the first inequality follows from the triangular inequality and the second inequality follows from Jensen's inequality. □

By combining the above two Lemma, Theorem 2 is easily established.

A.3 Convergence with constant batch size

Proof of Theorem 3. We prove by induction. The case when $k = 1$ is obvious. Suppose the theorem is true for $i \leq k-1$, then for $i = k$,

$$\begin{aligned}
\mathbb{E}\,[h_k] \leq&\mathbb{E}\left[(1-\eta_k)h_{k-1} + \eta_k\left(\langle \nabla F(\boldsymbol{X}_{k-1}) - \tilde{\nabla}_{k-1}, \boldsymbol{U}_k - \boldsymbol{X}^*\rangle + \frac{L\eta_k D^2}{2}\right)\right] \\
\leq&\mathbb{E}\left[(1-\eta_k)h_{k-1} + \eta_k\left(\frac{GD}{\sqrt{m}} + \frac{L\eta_k D^2}{2}\right)\right] \\
\leq&\frac{LD^2}{k+1}\left[(k-1)\frac{\mathbb{E}\,[h_{k-1}]}{LD^2} + \frac{2}{c} + \frac{2}{k+1}\right] && \text{(plug in } m \text{ and } \eta_k\text{)} \\
\leq&\frac{LD^2}{k+1}\left[\frac{4(k-1)}{k+1} + \frac{k-1}{c} + \frac{2}{c} + \frac{2}{k+1}\right] \\
=&\frac{LD^2}{k+1}\left[\frac{4k-2}{k+1} + \frac{k+1}{c}\right] \\
\leq&\frac{4LD^2}{k+2} + \frac{1}{c}LD^2 && \text{(since } \frac{k}{(k+1)^2} < \frac{1}{k+2}\text{)}
\end{aligned}$$

□

Proof of Theorem 4. We prove by induction. The case when $k = 1$ is obvious. Suppose the theorem is true

for $k \leq T-1$, then for $k = T$,

$$
\begin{aligned}
h_T \leq & (1-\eta_T) h_{T-1} + \eta_T \left[\langle \nabla F(\boldsymbol{X}_{T-1}) - \tilde{\nabla}_{T-1}, \boldsymbol{U}_T - \boldsymbol{X}^* \rangle + \frac{L\eta_T D^2}{2} \right] \\
\leq & (1-\eta_T) h_{T-1} + \eta_T \left[\frac{GD}{\sqrt{m}} + L\tau\eta_{T-\tau} D^2 + \frac{L\eta_T D^2}{2} \right] && \text{(similar to Lemma 1)} \\
= & \frac{T-1}{T+1} h_{T-1} + \frac{2}{T+1} \left[\frac{\tau L D^2}{c} + \frac{2\tau L D^2}{T+1-\tau} + \frac{L D^2}{T+1} \right] && \text{(plug in } m = \frac{c^2 G^2}{L^2 D^2} \text{ and } \eta_T\text{)} \\
\leq & \frac{T-1}{T+1} \frac{(4\tau+1) 2 L D^2}{T+1} + \frac{T-1}{T+1} \frac{\tau L D^2}{c} + \frac{2}{T+1} \left[\frac{\tau L D^2}{c} + \frac{2\tau L D^2}{T+1-\tau} + \frac{L D^2}{T+1} \right] \\
= & \frac{4LD^2}{(T+1)^2} \left[\left(2\tau + \frac{1}{2} \right) (T-1) + \frac{1}{2} + \frac{T+1}{T+1-\tau} \tau \right] + \frac{\tau L D^2}{c} \\
\leq & \frac{4LD^2}{(T+1)^2} \left[\left(2\tau + \frac{1}{2} \right) (T-1) + \frac{1}{2} + 2\tau \right] + \frac{\tau L D^2}{c} && (\tau < T/2) \\
= & \frac{4LD^2}{(T+1)^2} \left[\left(2\tau + \frac{1}{2} \right) T \right] + \frac{\tau L D^2}{c} \\
\leq & \frac{(4\tau+1) 2 L D^2}{(T+2)} + \frac{\tau L D^2}{c} && \left(\text{since } \frac{T}{(T+1)^2} < \frac{1}{T+2}\right)
\end{aligned}
$$

□

Proof of Corollary 1. By Eq. (11), let

$$\frac{(4\tau+1) \cdot 2LD^2}{k+2} + \frac{\tau}{c} L D^2 \leq \epsilon$$

, solve for k, and one can obtain the iteration bounds. For each iteration, we need

$$\mathcal{O}(c^2/\tau^2)$$

stochastic gradient iteration. Multiply the iteration bound by

$$\mathcal{O}(c^2/\tau^2)$$

and we can see how much stochastic gradient evaluations that we need. The number of linear optimization equals the number of total iterations. □

Algorithm 4 Naive Asynchronous Stochastic Variance Reduced Frank-Wolfe Method (SVRF-asyn) (Only for analysis, not implementation)

// The Master Node
Input: Max delay tolerance τ; Max iteration count T; max inner-iteration counts N_k; Step size η_t and batch size m_t
Initialization: Randomly initialize with $\|\boldsymbol{X}_0\|_* = 1$ and broadcast $\boldsymbol{X}_0$.
for iteration $k = 0, 1, \cdots, T$ **do**
  $\boldsymbol{X}_0 = \boldsymbol{W}_k$, $t_m = 1$
  while $t_m \leq N_k$ **do**
    Wait until received $\{\boldsymbol{U}_w, t_w\}$ from a worker w.
    if $t_m - t_w > \tau$, abandon $\boldsymbol{U}_w$ and continue.
    $t_m = t_m + 1$
    $\boldsymbol{X}_{t_m} \leftarrow \eta_{t_m} \boldsymbol{U}_w + (1 - \eta_{t_m}) \boldsymbol{X}_{t_m - 1}$
  end while
  $\boldsymbol{W}_{k+1} = \boldsymbol{X}_{N_k}$
  Broadcast $\boldsymbol{W}_{k+1}$ and the update-W-signal
end for
// For each worker $w = 1, 2, \cdots, W$
while No Stop Signal **do**
  if Update-W-signal **then**
    Update the local copy of $\boldsymbol{W}$ and Compute $\nabla F(\boldsymbol{W})$
  end if
  Receive from the Master $\boldsymbol{X}_{t_m}$.
  // t_m is the inner iteration count at the master node.
  $t_w = t_m$, $\boldsymbol{X}_w = \boldsymbol{X}_m$.
  // Update the local copy of $\boldsymbol{X}$ and iteration count.
  Randomly sample an index set S where $|S| = m_{t_w}$
  $\nabla_w = \frac{1}{m_{t_w}} \sum_{i \in S} \left(\nabla f_i(\boldsymbol{X}_{t_w}) - \nabla f_i(\boldsymbol{W}) \right) + \nabla F(\boldsymbol{W})$ // the variance reduced minibatch gradient
  $\boldsymbol{U}_w \leftarrow \operatorname{argmin}_{\|\boldsymbol{U}\|_* \leq \theta} \langle \nabla_w, \boldsymbol{U} \rangle$
  send $\{\boldsymbol{U}_w, t_w\}$ to the Master node.
end while

B Asynchronous Stochastic Variance Reduced Frank Wolfe

In this section we describe how to run SVRF asynchronously, and in a communication efficient way. Similar to Section 3, we begin with a naive asynchronous SVRF, as in Algorithm 4. The core idea is to run the inner iteration of SVRF asynchronously.

And then we describe how to make the naive asynchronous SVRF communication efficient. We reduce the per-iteration communication cost of inner iterations to $\mathcal{O}(n)$, as what we achieved in SFW-asyn.

C Distributed Computational Cluster Setup

In this section we describe how to perform the experiments in Section 5 on Amazon AWS.

We use MIT StarCluster software as the heavy-lifting tool for AWS cluster management. We launch 15 AWS `M1.SMALL` instances as the worker nodes, and 1 AWS `M1.LARGE`, and connect these machines with a virtual private network.

We use `MPI4PY` (Dalcín et al., 2008) as the fundamental APIs to implement distributed algorithms. `MPI4PY` is built on-top-of the Message Passing Standard, and is capable of implementing asynchronus distributed algorithms

Algorithm 5 Asynchronous Stochastic Variance Reduced Frank-Wolfe Method (SVRF-asyn)

// The Master Node
Input: Max delay tolerance τ; Max iteration count T; max inner-iteration counts N_k; Step size η_t and batch size m_t
Initialization: Randomly initialize $\boldsymbol{X}_0 = \boldsymbol{u}_0\boldsymbol{v}_0^T$ s.t. $\|\boldsymbol{X}_0\|_* = 1$ and broadcast $\{\boldsymbol{u}_0, \boldsymbol{v}_0\}$ to all the workers; The iteration count at the master node $t_m = 0$.
for iteration $k = 0, 1, \cdots, T$ **do**
  $\boldsymbol{X}_0 = \boldsymbol{W}_k$, $t_m = 1$ // Maintain a local copy for output
  while $t_m \leq N_k$ **do**
    Wait until received from a worker $\{\boldsymbol{u}_w, \boldsymbol{v}_w, t_w\}$.
    if $t_m - t_w > \tau$ **then**
      Send $(\boldsymbol{u}_{t_m}, \boldsymbol{v}_{t_m}), ...(\boldsymbol{u}_{t_w+1}, \boldsymbol{v}_{t_w+1})$ to node w.
      continue.
    end if
    $t_m = t_m + 1$ and store $\{\boldsymbol{u}_w, \boldsymbol{v}_w\}$ as $\boldsymbol{u}_{t_m}$ and $\boldsymbol{v}_{t_m}$
    Send $(\boldsymbol{u}_{t_m}, \boldsymbol{v}_{t_m}), ...(\boldsymbol{u}_{t_w+1}, \boldsymbol{v}_{t_w+1})$ to node w.
    $\boldsymbol{X}_{t_m} \leftarrow \eta_{t_m}\boldsymbol{u}_{t_m}\boldsymbol{v}_{t_m}^T + (1 - \eta_{t_m})\boldsymbol{X}_{t_m-1}$
    // Not run in real time
    // Maintain a local copy for output
  end while
  $\boldsymbol{W}_{k+1} = \boldsymbol{X}_{N_k}$ // Maintain a local copy for output
  Send update-W-signal to all workers.
  Send $(\boldsymbol{u}_N, \boldsymbol{v}_N), ...$ to all workers.
end for
// For each worker $w = 1, 2, \cdots, W$
while No Stop Signal **do**
  if Update-W-signal **then**
    Obtain $(\boldsymbol{u}_N, \boldsymbol{v}_N), ...$ from the master
    Update the local copy of $\boldsymbol{X}$ to $\boldsymbol{X}_N$ as in Eqn 6
    $\boldsymbol{W} \leftarrow \boldsymbol{X}_N$ and Compute $\nabla F(\boldsymbol{W})$
  else
    Obtain $(\boldsymbol{u}_{t_m}, \boldsymbol{v}_{t_m}), ...(\boldsymbol{u}_{t_w+1}, \boldsymbol{v}_{t_w+1})$ from the master node.
    Update the local copy of $\boldsymbol{X}_{t_w}$ to $\boldsymbol{X}_{t_m}$ according to Eqn 6
  end if
  Randomly sample an index set S where $|S| = m_{t_w}$
  $\nabla_w = \frac{1}{m_{t_w}} \sum_{i \in S} (\nabla f_i(\boldsymbol{X}_{t_w}) - \nabla f_i(\boldsymbol{W})) + \nabla F(\boldsymbol{W})$
  $\boldsymbol{u}_w\boldsymbol{v}_w^T \leftarrow \text{argmin}_{\|\boldsymbol{U}\|_* \leq \theta} \langle \nabla_w, \boldsymbol{U} \rangle$
  send $\{\boldsymbol{u}_w, \boldsymbol{v}_w, t_w\}$ to the Master node.
end while

D More simulation results

In order to better understand how much speedup asynchrony offers, and the conditions that the speedup occur, we test SFW-asyn against SFW-dist for matrix sensing and PNN under a distributed computational modeled by queuing theory.

Queuing model is frequently used to model the staleness of each workers in distributed computational setting Mitliagkas et al. (2016). We consider each D_1D_2 operation takes one unit of time in expectation. Therefore, each stochastic gradient evaluation of matrix sensing and PNN takes one unit of time in expectation; we solve the 1-SVD up to a practical precision Allen-Zhu et al. (2017), and we consider 1-SVD takes ten units of time in expectation. In our simulation we find that setting the expected time of 1-SVD as ten or twenty or five has marginal impact on the results. We further assume that the computation time follows a geometric distribution:

Assumption 3. *Denote random variable t as the computation time required for each worker to finish a computation task that takes C units of time in expectation. Then for $x = C, 2C, ...nC$, $\mathbb{P}(t = x) = p(1 - p)^{x/C-1}$ for a distribution parameter p.*

The intuition behind the staleness parameter p is that, when p is set to 1, there is no randomness - each worker finish the work in the exactly same amount of time. When the staleness parameter is small, say, 0.1, then the computational time of each worker differs a lot - some may finish their jobs faster, while some are

slower.

We want to emphasize that

(1) this assumption is **not required** for our convergence results (Theorem 1 and 2) to hold.

(2) the cost of communication is not taken into consideration, and hence we are implicitly favoring sfw-dist.

We show the convergence VS simulated time in Fig 6, and the speedup over single worker in in Fig 7. As in Fig 7, the speedup of SFW-asyn is almost linear, while SFW-dist compromises as the number of workers get larger.

The improvement over SFW-dist is less significant, as we increase the staleness parameter from $p = 0.1$ to $p = 0.8$. Obviously, SFW-dist performs better on large staleness parameter. When the staleness parameter is closer to one, the machines performs more uniformly (i.e., they finish the mini-batch computation in the similar amout of time), and therefore the slow-dowm of SFW-dist due to slowest worker is less significant. However, the performance of SFW-asyn also compromises slightly, as we go from $p = 0.1$ to $p = 0.8$. That is, SFW-asyn slightly prefer random delay, rather than consistent delay. This means that our analysis based on worst case scenario is loose, and can be improved.

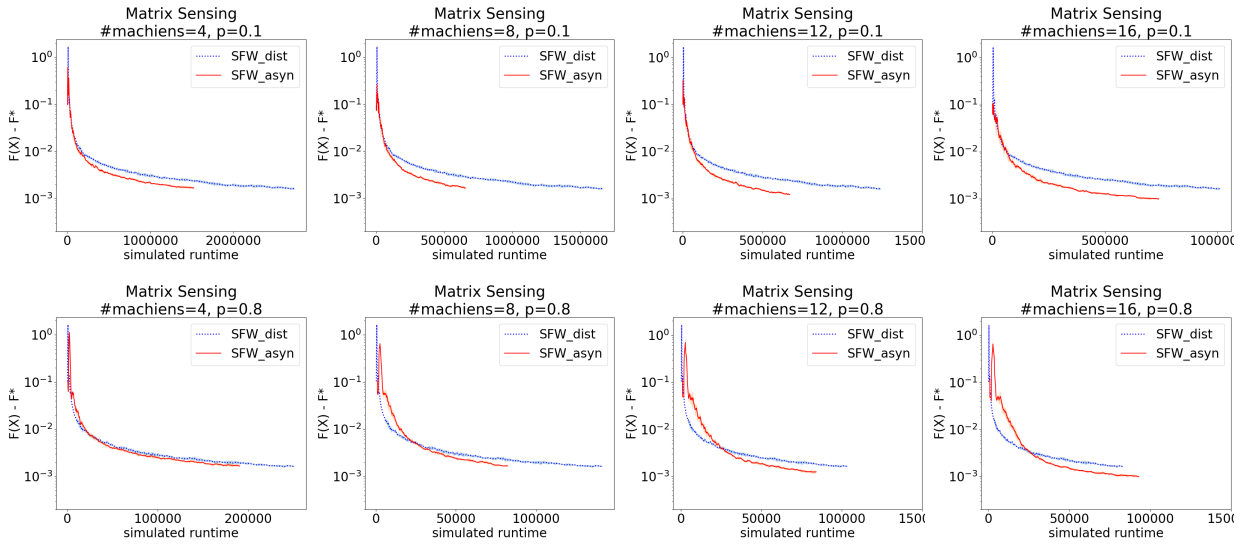

Figure 6: Convergence of the relative loss of Matrix Sensing problem VS the time simulated by queuing model. Simulation are repeated 5 times, and 1 standard deviation is shown in the colored shadow.

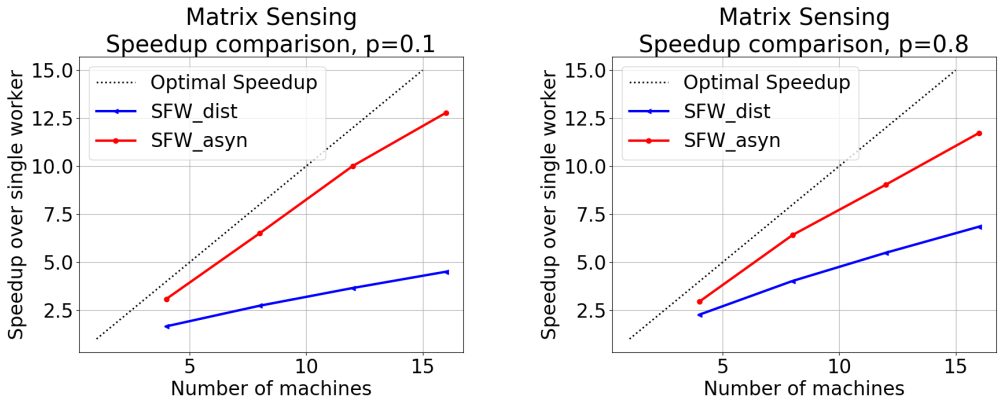

Figure 7: Comparing the time needed to achieve the same relative error (0.002) against single worker.