Projection Free Online Matrix Completion

Qianqian Ma
maqq@bu.edu

Artin Spiridonoff
artin@bu.edu

1 Introduction

In this project, we aim to propose a new efficient online algorithm to solve low-rank matrix completion problem. Matrix completion refers to the problem of recovering a low rank matrix from an incomplete subset of its entries. This problem arises in a number of applications that involve *collaborative filtering*, where one tries to predict an unknown preference of a user based on collective known preference of a large number of users [1].

This problem can be solved by online convex optimization algorithms like Online Gradient Descent and Stochastic Gradient Descent. While these algorithms are usually efficient, there exists a computational bottleneck when involves matrix completion problem, i.e., the projection step. Since the domain of interest in matrix completion problem is the set of matrices with bounded trace norm (which is a convex surrogate of rank), projecting into this set amounts to computing the full singular value decomposition (SVD) of the matrix projected, which requires a lot of computational effort.

To address this issue, Online Frank-Wolfe (OFW) algorithm was proposed in [2], which employed a linear optimization step to replace the projection step. Therefore, instead of conducting the full SVD, we just need compute the top singular vectors. Thus, the computational efficiency can be improved greatly. [2] showed a regret bound of $\mathcal{O}(T^{3/4})$ for OFW.

[3] proposed a Frank-Wolfe type algorithm called blockFW which replaces the linear programming of FW with a quadratic one and proposes an efficient method to solve it. This method requires calculating the top-k singular vectors, when $k \ll \min\{m, n\}$, this will be much more efficient than computing the full SVD. They also showed linear convergence of blockFW.

Despite the fast convergence of blockFW, this algorithm is not suited for online convex optimization. As a consequence, when dealing with matrix completion problem, we can not reveal the entries one by one, which is important in practical applications. To make accommodate for this, we plan to combine the blockFW algorithm and OFW model together and propose a new algorithm to solve low-rank matrix completion problem in this project. We will also provide detailed analysis and experiments to prove the effectiveness of the proposed algorithm.

The rest of this report is organized as follows. In Section 2 we review two projection-free algorithms. In Section 3 we present a technical analysis of constrained FTRL with time-varying regularizers. The results of this section will be used in Section 4 which studies Online Matrix Completion problem and potential methods to use FTRL to solve it efficiently. In Section 5 we present some numerical experiments followed by conclusion remarks in Section 6.

1.1 Notation

We use $\|X\|_*$ and $\|X\|_F$ to denote the nuclear (Trace) norm and Frobenius norm of a matrix $X \in \mathbb{R}^{m\times n}$, respectively. For two matrices A, B of the same size, their inner product is defined as $\langle A, B\rangle = AB^\top$ and $A \bullet B$ denotes their element-wise product.

2 Projection-Free Algorithms

2.1 Online Frank-Wolfe Algorithm

[2] proposes an efficient online learning algorithm (OFW) that doesn't require projection, using the Frank-Wolfe technique. In general, projection step amounts to solving a convex quadratic program over the domain. This paper gives online learning algorithms that replaces the projection step with a linear optimization step. In the application of bounded trace norm matrices of size $m \times n$, computing the projection of a matrix requires $O(nm^2)$ time, assuming $m \leq n$. Linear optimization over this domain requires calculating the top singular vectors of the matrix which can be done typically linear time in the number of non-zero entries of the matrix.

Algorithm 1 Online Frank-Wolfe (OFW)

```
Input parameter: constant $a \geq 0$.
Initialize $\mathbf{x}_1$ arbitrarily.
for $t = 1, 2, \ldots, T$ do
    Play $\mathbf{x}_t$ and observe $f_t$
    Define $F_t = \frac{1}{t}\sum_{\tau=1}^{t} f_\tau$
    Compute $\mathbf{v}_t \leftarrow \arg\min_{\mathbf{x}\in\mathcal{V}}\{\langle \nabla F_t(\mathbf{x}_t), \mathbf{x}\rangle\}$
    Set $\mathbf{x}_{t+1} = (1 - t^{-a})\mathbf{x}_t + t^{-a}\mathbf{v}_t$
end for
```

2.2 blockFW Algorithm

In [3], the authors considers the following optimization problem:

$$\min_{X\in\mathbb{R}^{m\times n}} \{f(X) : \|X\|_* \leq \theta\}, \tag{1}$$

where $f(X)$ is a strongly convex and smooth function. To solve this problem, they developed a rank-k variant Frank-Wolfe algorithm, also they proved this algorithm can converge in $O(\ln(1/\epsilon))$ rate which is much faster than Frank-Wolfe algorithm whose rate is $O(1/\epsilon)$. Besides, they showed that the per-iteration complexity of this algorithm scales with $\text{poly}\ln(1/\epsilon)$. The pseudo code of this algorithm is as following:

Algorithm 2 blockFW

```
$\eta \leftarrow \frac{1}{2\kappa}$
for $t = 1, 2, \ldots, T$ do
    $A_t \leftarrow \beta\eta X_t - \nabla f(X_t)$
    $(u_1, v_1, \ldots, u_k, v_k) \leftarrow k - \mathrm{SVD(A_t)}$
                    $\diamond(u_i, v_i)$ is the $i$-th largest pair of left/right singular vectors of $A_t$, $\sigma := (u_i^\top A_t v_i)_{i=1}^k$.
    $a \leftarrow \arg\min_{a\in\mathbb{R}^k, a\geq 0, \|a\|_1\leq 1}\left\|a - \frac{1}{\beta\eta}\sigma\right\|_2$
    $V_t \leftarrow \sum_{i=1}^k a_i u_i v_i^\top$
    $X_{t+1} \leftarrow X_t + \eta(V_t - X_t)$
end for
```

3 Constrained FTRL with Time-Varying Regularizers

In this section we analyze constrained FTRL with time-varying regularizers and prove a regret bound under mild assumptions. Consider the following algorithm.

Algorithm 3 Constrained FTRL with Time-Varying Regularizers

```
Input: Closed and convex set $\mathcal{V}$, a sequence of regularizers $\psi_1, \ldots, \psi_T : \mathcal{V} \to \mathbb{R}$.
for $t = 1, \ldots, T$ do
    Let $F_t(\mathbf{x}) = \sum_{i=1}^{t-1} \ell_i(\mathbf{x}) + \psi_t(\mathbf{x})$
    Compute $\mathbf{x}_t = \arg\min_{\mathbf{x}\in\mathcal{V}} F_t(\mathbf{x})$.
    Play $\mathbf{x}_t$ and pay $\ell_t(\mathbf{x}_t)$.
end for
```

Theorem 1. *Let ℓ_t be convex functions over $\mathcal{V}$ and regularizers $\psi_t, t = 1, \ldots, T$ satisfy $\psi_{t+1}(\mathbf{x}) \geq \psi_t(\mathbf{x}), \forall \mathbf{x} \in \mathcal{V}$. Moreover, assume F_t is λ_t-strongly-convex with respect to norm $\|.\|_t$ over $\mathcal{V}$. Then using Algorithm 3 we have,*

$$Regret_T(\mathbf{u}) \leq \psi_{T+1}(\mathbf{u}) - \min_{\mathbf{x}\in\mathcal{V}} \psi_1(\mathbf{x}) + \sum_{t=1}^{T} \frac{\|\mathbf{g}_t\|_{*t}^2}{2\lambda_t}, \qquad \forall \mathbf{g}_t \in \partial\ell_t(\mathbf{x}_t), t = 1, \ldots, T. \tag{2}$$

Proof. We have,

$$\sum_{t=1}^{T} \ell_t(\mathbf{x}_t) - \ell_t(\mathbf{u}) = \psi_{T+1}(\mathbf{u}) - \min_{\mathbf{x}\in\mathcal{V}} \psi_1(\mathbf{x}) + \sum_{t=1}^{T} [F_t(\mathbf{x}_t) - F_{t+1}(\mathbf{x}_{t+1}) + \ell_t(\mathbf{x}_t)] + F_{T+1}(\mathbf{x}_{T+1}) - F_{T+1}(\mathbf{u}). \tag{3}$$

Next, we try to bound each of the terms $F_t(\mathbf{x}_t) - F_{t+1}(\mathbf{x}_{t+1}) + \ell_t(\mathbf{x}_t)$ in (3). Note that by definition,

$$F_t(\mathbf{x}_t) - F_t(\mathbf{y}) \leq \langle \mathbf{G}_t, \mathbf{x}_t - \mathbf{y} \rangle - \frac{\lambda_t}{2}\|\mathbf{x}_t - \mathbf{y}\|_t^2 \leq -\frac{\lambda_t}{2}\|\mathbf{x}_t - \mathbf{y}\|_t^2, \qquad \forall \mathbf{y} \in \mathcal{V}, \mathbf{G}_t \in \partial F_t(\mathbf{x}_t).$$

We can write,

$$\begin{aligned} F_t(\mathbf{x}_t) - F_{t+1}(\mathbf{x}_{t+1}) + \ell_t(\mathbf{x}_t) &= (F_t(\mathbf{x}_t) + \ell_t(\mathbf{x}_t)) - (F_t(\mathbf{x}_{t+1}) + \ell_t(\mathbf{x}_{t+1})) + \psi_t(\mathbf{x}_{t+1}) - \psi_{t+1}(\mathbf{x}_{t+1}) \\ &\leq (F_t(\mathbf{x}_t) - F_t(\mathbf{x}_{t+1})) + \langle \mathbf{g}_t, \mathbf{x}_t - \mathbf{x}_{t+1} \rangle + \psi_t(\mathbf{x}_{t+1}) - \psi_{t+1}(\mathbf{x}_{t+1}) \\ &\leq -\frac{\lambda_t}{2}\|\mathbf{x}_t - \mathbf{x}_{t+1}\|_t^2 + \frac{1}{2\lambda_t}\|\mathbf{g}_t\|_{*t}^2 + \frac{\lambda_t}{2}\|\mathbf{x}_t - \mathbf{x}_{t+1}\|_t^2 + \psi_t(\mathbf{x}_{t+1}) - \psi_{t+1}(\mathbf{x}_{t+1}) \\ &= \frac{\|\mathbf{g}_t\|_{*t}^2}{2\lambda_t} + \psi_t(\mathbf{x}_{t+1}) - \psi_{t+1}(\mathbf{x}_{t+1}). \end{aligned} \tag{4}$$

where $\mathbf{g}_t \in \partial \ell_t(\mathbf{x}_t)$. We used convexity of ℓ_t in the first inequality, λ_t-strong-convexity of F_t and Cauchy-Schwarz in the second inequality. Combining relations above and noting $F_{T+1}(\mathbf{x}_{T+1}) - F_{T+1}(\mathbf{u}) \leq 0$ and $\psi_t(\mathbf{x}_{t+1}) - \psi_{t+1}(\mathbf{x}_{t+1}) \leq 0$ we obtain (2).

□

Corrollary 1. *Suppose the losses ℓ_t be such that exists $A_t \succeq 0$,*

$$\ell_t(\mathbf{y}) \geq \ell_t(\mathbf{x}) + \langle \mathbf{g}, \mathbf{y} - \mathbf{x} \rangle + \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|_{A_t}^2, \qquad \forall \mathbf{x}, \mathbf{y} \in \mathcal{V}, \mathbf{g} \in \partial \ell_t(\mathbf{x}).$$

Then using Algoritm 3 with linearized losses $\tilde{\ell}_t(\mathbf{x}) = \langle \mathbf{g}_t, \mathbf{x} \rangle + \frac{1}{2}\|\mathbf{x} - \mathbf{x}_t\|_{A_t}^2$ and regularizers $\psi_t = \frac{\lambda}{2}\|\mathbf{x}\|^2$ we obtain,

$$\sum_{t=1}^{T} \ell_t(\mathbf{x}_t) - \ell_t(\mathbf{u}) \leq \frac{\lambda}{2}\|\mathbf{u}\|^2 + \frac{1}{2}\sum_{t=1}^{T}\|\mathbf{g}_t\|_{S_t^{-1}}^2,$$

where $S_t = \lambda I + \sum_{i=1}^{t} A_i$.

Remark 1. *This is not a direct result of Theorem 3. However, modifying the proof by approximating $\ell_t(\mathbf{x}_t) - \ell_t(\mathbf{x}_{t+1}) \leq \langle \mathbf{g}_t, \mathbf{x}_t - \mathbf{x}_{t+1} \rangle - \frac{1}{2}\|\mathbf{x}_t - \mathbf{x}_{t+1}\|_{A_t}^2$ in (4), implies the result above.*

4 Online Matrix Completion

In this section we analyze the Online Matrix Completion problem. Let us define the losses as,

$$\ell_t(X) = \frac{1}{2}\|X - M\|_{E_t}^2 := \frac{1}{2}\sum_{(i,j) \in E_t} (X_{(i,j)} - [M_t]_{i,j})^2$$

where $E_t \subseteq [m] \times [n]$ is the set of observed entries at time t and $X, M_t \in \mathbb{R}^{m \times n}$. Note that $\|.\|_{E_t}$ is not a norm. We have,

$$[\nabla \ell_t(X)]_{i,j} = \begin{cases} X_{(i,j)} - [M_t]_{i,j}, & \text{if } (i,j) \in E_t, \\ 0, & \text{otherwise.} \end{cases}$$

Moreover,

$$\ell_t(Y) = \ell_t(X) + \langle \nabla \ell_t(X), Y - X \rangle + \frac{1}{2}\|X - Y\|_{E_t}^2,$$

The constrained set $\mathcal{V}$ is defined as, $\mathcal{V} = \{X \in \mathbb{R}^{m \times n} | \|X\|_* \leq \theta\}$.

Remark 2. *Note that the losses in matrix completion are not strognly convex, which is one of the assumptions made in references [2, 3].*

4.1 Naiive Idea

The main idea that propels the linear convergence of blockFW is an efficient method for solving the quadratic programming with respect to $\|.\|_F$. Thus we propose the following regularizers. Let E_t' be the complement of E_t. Define,

$$\psi_t(X) = \frac{\lambda}{2}\|X\|_F^2 + \frac{1}{2}\sum_{i=1}^{t-1}\|X - X_i\|_{E_t'}^2.$$

Using Algorithm 3 with $\tilde{\ell}_t(X) = \langle G_t, X\rangle + \frac{1}{2}\|X - X_t\|_{E_t}^2$ and ψ_t defined as above, we have,

$$\begin{aligned} F_t(X) &= \sum_{i=1}^{t-1}\langle G_i, X\rangle + \frac{\lambda}{2}\|X\|_F^2 + \frac{1}{2}\sum_{i=1}^{t-1}\left(\|X - X_i\|_{E_t}^2 + \|X - X_i\|_{E_t'}^2\right) \\ &= \sum_{i=1}^{t-1}\langle G_i, X\rangle + \frac{\lambda}{2}\|X\|_F^2 + \frac{1}{2}\sum_{i=1}^{t-1}\|X - X_i\|_F^2. \end{aligned}$$

Thus, F_t is $(\lambda + t - 1)$-strongly-convex with respect to $\|.\|_2$. Moreover,

$$\arg\min_{X\in\mathcal{V}} F_t = \arg\min_{X\in\mathcal{V}}\left\{\sum_{i=1}^{t-1}\langle G_i, X\rangle + \frac{\lambda + t - 1}{2}\left\|X - \tilde{X}_t\right\|_F^2\right\}, \qquad \tilde{X}_t := \frac{\sum_{i=1}^{t-1} X_i}{\lambda + t - 1}.$$

Hence, the following holds.

$$\sum_{t=1}^{T}\ell_t(X_t) - \ell_t(U) \le \frac{\lambda}{2}\|U\|_F^2 + \frac{1}{2}\sum_{t=1}^{T-1}\|U - X_t\|_{E_t'}^2 + \sum_{t=1}^{T}\frac{\|G_t\|_F^2}{2(\lambda + t)}.$$

The regret bound above is not ideal and in the worst case can grow linearly in time.

4.2 Fixed Regularizers

If we set the regularizers to $\psi_t(X) = \frac{\lambda}{2}\|X\|_F^2$, using Algorithm 3 we need to compute the following,

$$X_t = \arg\min_{X\in\mathcal{V}}\sum_{i=1}^{t-1}\left(\langle G_i, X\rangle + \frac{1}{2}\|X - X_i\|_{E_i}^2\right) + \frac{\lambda}{2}\|X\|_F^2. \tag{5}$$

If (5) can be done efficiently, assuming gradients are bounded, we can obtain $\mathcal{O}(\ln T)$ regret bound.

Define $R_t \in \mathbb{R}^{m\times n}$ with

$$[R_t]_{i,j} = \left(\lambda + \sum_{i=1}^{t-1} 1[(i,j) \in E_i]\right)^{\frac{1}{2}}.$$

Define $Y = R_t \bullet X$, then (5) becomes equivalent to solving the following minimization:

$$Y_t = \arg\min_{R_t\bullet Y\in\mathcal{V}}\langle R_t\sum_{i=1}^{t-1}(G_i - \mathbf{x}_i), Y\rangle + \frac{1}{2}\|Y\|_F^2, \qquad X_t = R_t \bullet Y_t. \tag{6}$$

Even though this minimization is similar to the one in blockFW, it's not over $\mathcal{V}$ anymore.

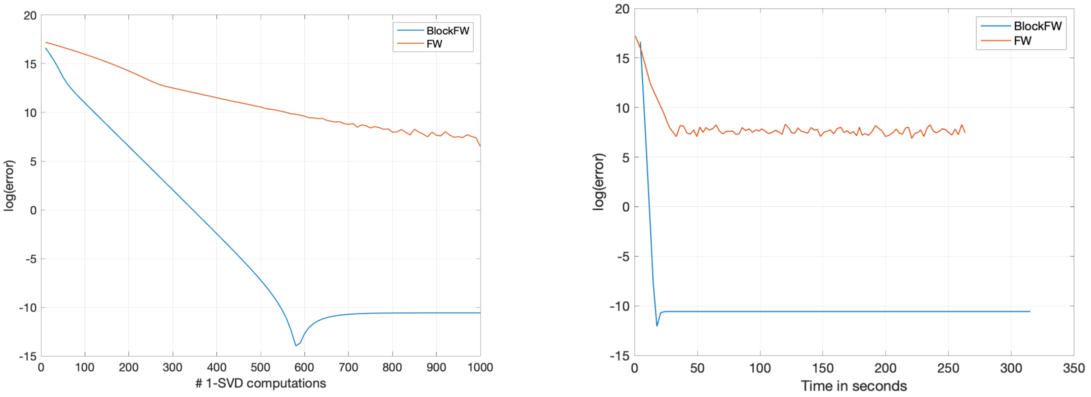
5 Implementations

5.1 Traditional algorithms for matrix completion

To verify the effectiveness of the algorithms we have discussed and compare the performance of these algorithms, we conducted the following synthetic experiment for matrix completion. We generate a random rank-10 matrix in dimension 1000×1000 with some small additional noises, we also set the trace norm bound be 10000, i.e., $\theta = 10000$.

First, we implememted the offline algorithms FW and blockFW on this synthetic dataset. When implementing FW, we use $\eta = 0.05$, when implementing blockFW, we use $\eta = 0.2$ and $k = 10$. From Figure 1, we can see blockFW converges to the optimal solution while FW converges to a range of the optimal solution. This is because we use constant stepsizes for both algorithms. Besides, we can see that blockFW converges faster both in the number of 1-SVD computations and in wall clock-time, which proves that the blockFW is a more efficient algorithm for matrix completion problem compared to FW.

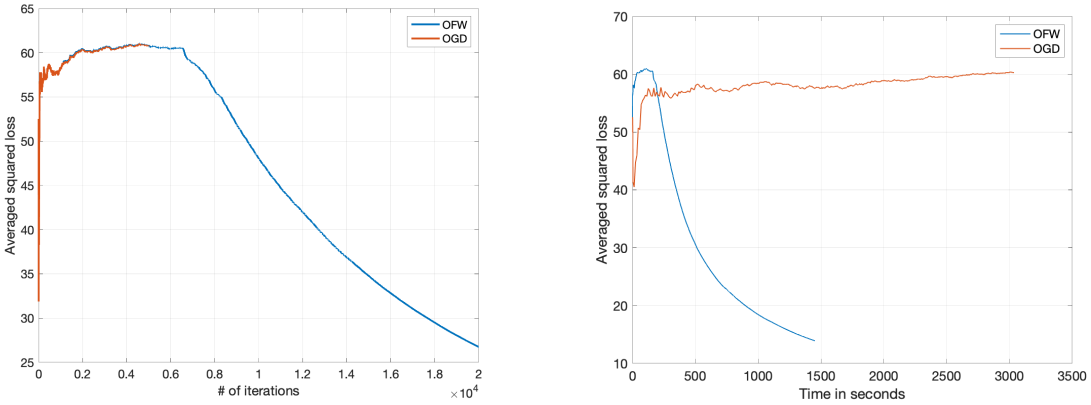
Next, we implemented the online algorithms OFW and OGD on the synthetic dataset. For OGD, we use time-varying stepsize $\eta = 1/\sqrt{t}$. For OFW, we let $a = \frac{1}{2}$. From Figure 2, we can see that OFW converges much faster than OGD, this is because OGD needs conduct full-SVD to project the update to the trace norm ball in each step, while OFW just needs a linear programming step which is equivalent to 1-SVD. In addition, in plot 2(b), the error of OGD is increasing since the size of our synthetic matrix is 1000×1000 while in this plot, OGD are just given 2000 revealed entries which can lead to this phenomenon.



(a)

(b)

Figure 1: Comparison of Frank-Wolfe and blockFW on synthetic dataset. Plot (a) shows log error vs. the number of the 1-SVD computations for FW and blockFW respectively. Plot (b) shows log error vs. running time for 1000 FW iterations and 100 blockFW iterations.



(a)

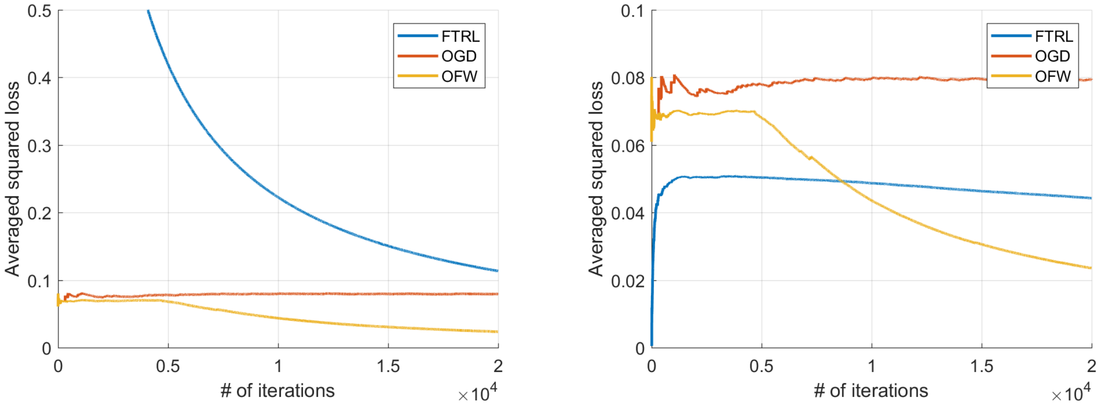
(b)

Figure 2: Comparison of OGD and OFW on synthetic dataset. The left plot shows the average squared loss vs. 20000 iterations for OFW and 5000 iterations for OGD (We didn't run the same number of iterations for OFW and OGD since OGD is very slow and 5000 iterations of OGD are enough to compare the performance of the two algorithms). The right plot shows the average squared loss vs. the running time for 40000 OFW iterations and 2000 OGD iterations.

5.2 FTRL for Online Matrix completion

For FTRL we tried two methods. In the first method, at every iteration we calculate the global minimum of (5) and then project it to the set $\{\|X\|_* \leq \theta\}$. This method requires a full-SVD. For the second method, we perform the minimization of (6), over the constrained set $Y \in \mathcal{V}$ instead of $R_t \bullet Y \in \mathcal{V}$, where $\mathcal{V} = \{X | \|X\|_* \leq \theta, \text{rank}(X) \leq k\}$. This method is faster than the first method, since it does not require the full singular value decomposition at every iteration. These simulations are performed on a synthetic matrix of size 300×300 with $\lambda = 0.1$ and $\theta = 600$ and $k = 10$.

It can be seen in Figure 3 that using the first method, the error in FTRL decays very slowly. On the other hand, using the second method, FTRL performs slightly better than the OGD, however, it is still slower than OFW. Thus, it's not showing the $\mathcal{O}(\ln(T))$ regeret bound that we expected from our analysis. This can be explained by the fact that we don't solve (6) accurately, since we relaxed the constrained set.



(a)

(b)

Figure 3: Performance of FTRL using two different methods to solve the Regularized Leader. The left figure uses the first method projection, while the right figure uses change of variables and a relaxation on the constrained set. Performances of OGD and OFW on the same dataset are plotted for comparison.

6 Conclusion

In this project, we studied the problem of online projection free matrix completion. We studied two projection-free algorithms, OFW and blockFW. The former has a large regret bound while the latter has a linear convergence rate, however, it is offline. Then, we tried to apply the ideas of blockFW to develop a fast online matrix completion algorithm. We used FTRL to utilize the unique properties of the matrix completion problem. We faced technical obstacles to efficiently solve the minimizations required in FTRL. Finally, we performed a number of numerical simulations to compare the performance of the mentioned algorithms.

References

[1] David Gamarnik and Sidhant Misra. A note on alternating minimization algorithm for the matrix completion problem. *IEEE Signal Processing Letters*, 23(10):1340–1343, 2016.

[2] Elad Hazan and Satyen Kale. Projection-free online learning. *arXiv preprint arXiv:1206.4657*, 2012.

[3] Zeyuan Allen-Zhu, Elad Hazan, Wei Hu, and Yuanzhi Li. Linear convergence of a frank-wolfe type algorithm over trace-norm balls. In *Advances in Neural Information Processing Systems*, pages 6191–6200, 2017.