

Revisiting Co-Occurring Directions: Sharper Analysis and Efficient Algorithm for Sparse Matrices

Luo Luo¹, Cheng Chen^{2*}, Guangzeng Xie³, Haishan Ye⁴

¹ Department of Mathematics, The Hong Kong University of Science and Technology

² Department of Computer Science and Engineering, Shanghai Jiao Tong University

³ Academy for Advanced Interdisciplinary Studies, Peking University

⁴ School of Management, Xi'an Jiaotong University

luoluo@ust.hk, jack_chen1990@sjtu.edu.cn, smsxgz@pku.edu.cn, hsy_e_cs@outlook.com

Abstract

We study the streaming model for approximate matrix multiplication (AMM). We are interested in the scenario that the algorithm can only take one pass over the data with limited memory. The state-of-the-art deterministic sketching algorithm for streaming AMM is the co-occurring directions (COD), which has much smaller approximation errors than randomized algorithms and outperforms other deterministic sketching methods empirically. In this paper, we provide a tighter error bound for COD whose leading term considers the potential approximate low-rank structure and the correlation of input matrices. We prove COD is space optimal with respect to our improved error bound. We also propose a variant of COD for sparse matrices with theoretical guarantees. The experiments on real-world sparse datasets show that the proposed algorithm is more efficient than baseline methods.

1 Introduction

A large scale machine learning system usually receives data sequentially and it is often impossible to exactly store the entire data set. Thus, the approximate matrix multiplication (AMM) in the streaming fashion is an important and fundamental task for scientific computation and big data analysis. For example, the product of matrices from multi-modal datasets captures the correlation between different modalities. In addition, many classical algorithms including canonical correlation analysis (Hotelling 1992), generalized eigenvector decomposition (Golub and Loan 1996), partial least squares (Wegelin 2000), spectral co-clustering (Dhillon 2001) require to perform approximate matrix multiplication when the data set is very large. On the other hand, data matrices from real-world are usually low-rank and sparse, which motivated us to design efficient and effective sparse algorithms.

This paper considers streaming AMM problem as follows. Give two large matrices $\mathbf{X} \in \mathbb{R}^{n \times d_x}$ and $\mathbf{Y} \in \mathbb{R}^{n \times d_y}$, we are interested in finding a low-rank estimator $\mathbf{A}^\top \mathbf{B}$ to approximate $\mathbf{X}^\top \mathbf{Y}$, where $\mathbf{A} \in \mathbb{R}^{m \times d_x}$, $\mathbf{B} \in \mathbb{R}^{m \times d_y}$ and m is much smaller than n , d_x and d_y . We focus on the row update model, that is, the algorithm receives rows of \mathbf{X} and \mathbf{Y} sequentially and it only takes one pass over input matrices

with limited memory. The key challenge for this problem is to reduce the space/time complexity while maintaining the approximation error.

Inspired by the idea of finding frequent items (Misra and Gries 1982), Liberty (2013) proposed frequent directions algorithm (FD), which considers the symmetric case of AMM such that $\mathbf{X} = \mathbf{Y}$ (a.k.a., the covariance sketching). FD achieves optimal tradeoffs between space cost and approximation error (Woodruff 2014a; Ghashami et al. 2016; Ghashami and Phillips 2014). Moreover, we can combine FD with subspace power iteration (Woodruff 2014b; Musco and Musco 2015) to design an algorithm which is efficient for sparse matrix multiplication (Ghashami, Liberty, and Phillips 2016), called sparse frequent directions (SFD). Recently, Huang (2019) integrated random sampling (Drineas, Kannan, and Mahoney 2006) into FD to reduce its time complexity. Luo et al. (2019) introduced a regularization term for FD, which makes the estimator is more friendly to inverse operation. FD technique can also be used to accelerate many popular machine learning models, such as convex online optimization (Luo et al. 2016, 2019), factorization machine (Luo et al. 2018), linear contextual bandits (Kuzborskij, Cella, and Cesa-Bianchi 2019; Chen et al. 2020) and ridge regression (Shi and Phillips 2020; Dickens 2020).

Mroueh, Marcheret, and Goel (2017) proposed a variant of FD called co-occurring directions (COD) for streaming AMM. COD shrinks the singular values of input matrices \mathbf{X} and \mathbf{Y} simultaneously at each iteration. It is shown that COD has significantly better performance than other sketching algorithms (Ye, Luo, and Zhang 2016; Drineas, Kannan, and Mahoney 2006; Clarkson and Woodruff 2017; Sarlos 2006) on AMM problem empirically. However, the existing spectral error bound of COD can not completely explain its high performance. It depends on the Frobenius norm of \mathbf{X} and \mathbf{Y} , which ignores the potential low-rank structure of the data matrix. Specifically, in the case of $\mathbf{X} = \mathbf{Y}$, the procedure of COD degrades to FD, but its error bound is worse than that of FD. Another deterministic sketching method for AMM, which we call FD-AMM (Ye, Luo, and Zhang 2016), directly adopts FD to sketch the concatenated matrix $\mathbf{Z} = [\mathbf{X}, \mathbf{Y}]$. The output of the algorithm is an approximation of $\mathbf{Z}^\top \mathbf{Z}$, whose sub-matrix corresponds to an estimator of $\mathbf{X}^\top \mathbf{Y}$.

In this paper, we provide a sharper analysis for co-

*Corresponding Author
Copyright © 2021, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

occurring directions (COD). We give a new spectral norm error bound which considers the potential low-rank structure of the target matrix. Our bound could be much tighter than Mroueh, Marcheret, and Goel’s (2017) results when the spectrum of the exact matrix product is dominated by its top singular values. In addition, we prove that the space complexity of COD is optimal to attain our improved error bound. Furthermore, in the case of $\mathbf{X} = \mathbf{Y}$, our result matches the error bound of FD.

We further propose sparse co-occurring directions (SCOD) and provide an error bound matches our improved analysis on standard COD while the running time of the algorithm mainly depends on the non-zero entries of input matrices. We conduct numerical experiments on cross-language datasets to show that SCOD has better performance than state-of-the-art algorithms empirically. Concurrent to our work, Wan and Zhang (2020) have also proposed a similar COD based algorithm to address streaming AMM with sparse inputs but their error bound does not consider the potential low-rankness.

The rest of the paper is organized as follows. In Section 2, define the notation used in this paper and introduce the background of related algorithms for streaming AMM. In Section 3, we provide our new error bound for COD algorithm and show the corresponding space lower bound. In Section 5, we propose SCOD and give its theoretical guarantees. In Section 6, we conduct the numerical experiments to show the superiority of SCOD. We defer detailed proof of some lemmas and theorems into supplementary materials. We conclude our work in Section 7.

2 Notations and Preliminaries

In this section, we first introduce the notation will be used in this paper. Then we give the backgrounds of frequent directions and related algorithms for AMM.

2.1 Notations

We let \mathbf{I}_p be the $p \times p$ identity matrix and $\mathbf{0}_{p \times q}$ be the $p \times q$ matrix of all zeros. For an $p \times q$ matrix $\mathbf{A} = [A_{ij}]$, we denote $(\mathbf{a}^{(i)})^\top$ be its i -th row, $\text{nnz}(\mathbf{A})$ be the number of non-zero entries of \mathbf{A} . The condensed singular value decomposition (SVD) of \mathbf{A} is defined as $\mathbf{U}\Sigma\mathbf{V}^\top$ where $\mathbf{U} \in \mathbb{R}^{m \times r}$ and $\mathbf{V} \in \mathbb{R}^{n \times r}$ are column orthogonal, $\Sigma = \text{diag}(\sigma_1(\mathbf{A}), \sigma_2(\mathbf{A}), \dots, \sigma_r(\mathbf{A}))$ with $\sigma_1(\mathbf{A}) \geq \sigma_2(\mathbf{A}) \geq \dots \geq \sigma_r(\mathbf{A}) > 0$ places the nonzero singular values on its diagonal entries and r is the rank of \mathbf{A} . We have $\sigma_i(\mathbf{A}) = 0$ for any $i > r$. Additionally, we let $\|\mathbf{A}\|_F = \sqrt{\sum_{i,j} A_{ij}^2} = \sqrt{\sum_{i=1}^r \sigma_i^2(\mathbf{A})}$ be the Frobenius norm, $\|\mathbf{A}\|_2 = \sigma_1(\mathbf{A})$ be the spectral norm, $\|\mathbf{A}\|_* = \sum_{i=1}^r \sigma_i(\mathbf{A})$ be the nuclear norm and $\|\mathbf{A}\|_k = \sum_{i=1}^k \sigma_i(\mathbf{A})$ be the Ky Fan k -norm. We also denote \mathbf{A}_k as the best rank- k approximation to \mathbf{A} for any unitary invariant norms, that is, $\mathbf{A}_k = \sum_{i=1}^k \sigma_i(\mathbf{A}) \mathbf{u}_i \mathbf{v}_i^\top$, where \mathbf{u}_i and \mathbf{v}_i are the i -th column of \mathbf{U} and \mathbf{V} respectively.

2.2 Frequent Directions

Frequent directions (Liberty 2013; Ghashami et al. 2016) is a deterministic algorithm for covariance sketching. Given any

matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ and sketch size m that is much smaller than n and d , FD processes the rows of \mathbf{X} one by one and produces a sketch matrix $\mathbf{A} \in \mathbb{R}^{2m \times d}$ to approximate $\mathbf{X}^\top \mathbf{X}$ by $\mathbf{A}^\top \mathbf{A}$. We present the details of FD in Algorithm 1, which requires $\mathcal{O}(md)$ space and $\mathcal{O}(mnd)$ time complexity. The algorithm has the following theoretical guarantees.

Lemma 1 (Ghashami and Phillips 2014; Ghashami et al. 2016). *The output \mathbf{A} of Algorithm 1 satisfies*

$$\|\mathbf{X}^\top \mathbf{X} - \mathbf{A}^\top \mathbf{A}\|_2 \leq \frac{1}{m-k} \left(\|\mathbf{X}\|_F^2 - \|\mathbf{X}_k\|_F^2 \right) \quad (1)$$

for any $k < m$.

Ghashami et al. (2016) also prove FD is space optimal with respect to the guaranteed accuracy in Lemma 1. Note that the shrinking step in line 8 of the algorithm is necessary because the output could be extremely worse without this operation (Desai, Ghashami, and Phillips 2016; Luo et al. 2019).

Algorithm 1 Frequent Directions (FD)

- 1: **Input:** $\mathbf{X} \in \mathbb{R}^{n \times d}$ and sketch size m
 - 2: $\mathbf{A} \leftarrow \mathbf{0}_{2m \times d}$
 - 3: **for** $t = 1, 2, \dots, n$
 - 4: insert $(\mathbf{x}^{(t)})^\top$ into a zero valued row of \mathbf{A}
 - 5: **if** \mathbf{A} has no zero valued rows **then**
 - 6: $[\mathbf{U}, \Sigma, \mathbf{V}] \leftarrow \text{SVD}(\mathbf{A})$
 - 7: $\delta \leftarrow \sigma_m^2(\mathbf{A})$
 - 8: $\widehat{\Sigma} \leftarrow \sqrt{\max(\Sigma^2 - \delta \mathbf{I}_{2m}, \mathbf{0}_{2m \times 2m})}$
 - 9: $\mathbf{A} \leftarrow \widehat{\Sigma} \mathbf{V}^\top$
 - 10: **end if**
 - 11: **end for**
 - 12: **Output:** \mathbf{A}
-

2.3 Sketching Algorithms for AMM

It is natural to exploit the idea of FD to solve general AMM problem (Ye, Luo, and Zhang 2016). We can concatenate the input matrix $\mathbf{X} \in \mathbb{R}^{n \times d_x}$ and $\mathbf{Y} \in \mathbb{R}^{n \times d_y}$ to construct a larger matrix $\mathbf{Z} = [\mathbf{X}, \mathbf{Y}] \in \mathbb{R}^{n \times (d_x + d_y)}$, and then apply FD on \mathbf{Z} to approximate $\mathbf{Z}^\top \mathbf{Z}$ by $\mathbf{C}^\top \mathbf{C}$, where $\mathbf{C} = [\mathbf{A}, \mathbf{B}]$, $\mathbf{A} \in \mathbb{R}^{n \times d_x}$ and $\mathbf{B} \in \mathbb{R}^{n \times d_y}$. The top right sub-matrix of the $\mathbf{C}^\top \mathbf{C}$, i.e., the matrix $\mathbf{A}^\top \mathbf{B}$ is an approximation of $\mathbf{X}^\top \mathbf{Y}$. Intuitively, this algorithm wastes a large proportion of cost to approximate $\mathbf{X}^\top \mathbf{X}$ and $\mathbf{Y}^\top \mathbf{Y}$ (the other sub-matrices of $\mathbf{Z}^\top \mathbf{Z}$), which is unnecessary for the AMM task.

Mroueh, Marcheret, and Goel (2017) proposed the co-occurring directions (COD) for AMM. We present its detailed procedure in Algorithm 2. Each iteration of COD constructs the column basis of \mathbf{A} and \mathbf{B} by QR factorization independently and executes the shrinkage step on the small interaction matrix $\mathbf{R}_x \mathbf{R}_y^\top$. We point out that both COD and FD-AMM requires $\mathcal{O}(m(d_x + d_y))$ space and $\mathcal{O}(mn(d_x + d_y))$ time complexity. However, COD looks more reasonable than

FD-AMM since all of its operations surrounds approximating $\mathbf{X}^\top \mathbf{Y}$. The numerical experiments (Mroueh, Marcheret, and Goel 2017) show that COD performs significantly better than FD-AMM (Ye, Luo, and Zhang 2016) and other AMM algorithms (Drineas, Kannan, and Mahoney 2006; Clarkson and Woodruff 2017; Sarlos 2006) when input matrices is dense. We can prove that COD holds the guaranteed accuracy as follows.

Lemma 2 (Mroueh, Marcheret, and Goel 2017). *The output \mathbf{A} and \mathbf{B} of Algorithm 2 satisfies*

$$\|\mathbf{X}^\top \mathbf{Y} - \mathbf{A}^\top \mathbf{B}\|_2 \leq \frac{\|\mathbf{X}\|_F \|\mathbf{Y}\|_F}{m}. \quad (2)$$

Unfortunately, the result of Lemma 2 does not reveal the advantage of COD entirely. Consider that case of $\mathbf{X} = \mathbf{Y}$, the procedure of COD will reduce to FD, but the error bound of (2) becomes a special case of (1) in Lemma 1 with $k = 0$. The real-world dataset typically enjoys some approximate low-rank structure, which leads to the right-hand side of bound (1) could be much smaller than the one of (2). Hence although COD has better empirical performance, the existing error bounds are not tight enough.

Algorithm 2 Co-Occurring Directions (COD)

```

1: Input:  $\mathbf{X} \in \mathbb{R}^{n \times d_x}$ ,  $\mathbf{Y} \in \mathbb{R}^{n \times d_y}$  and sketch size  $m$ 
2:  $\mathbf{A} \leftarrow \mathbf{0}_{2m \times d_x}$ 
3:  $\mathbf{B} \leftarrow \mathbf{0}_{2m \times d_y}$ 
4: for  $t = 1, 2, \dots, n$ 
5:   insert  $(\mathbf{x}^{(t)})^\top$  into a zero valued row of  $\mathbf{A}$ 
6:   insert  $(\mathbf{y}^{(t)})^\top$  into a zero valued row of  $\mathbf{B}$ 
7:   if  $\mathbf{A}$  or  $\mathbf{B}$  has no zero valued rows then
8:      $(\mathbf{Q}_x, \mathbf{R}_x) \leftarrow \text{QR}(\mathbf{A}^\top)$ 
9:      $(\mathbf{Q}_y, \mathbf{R}_y) \leftarrow \text{QR}(\mathbf{B}^\top)$ 
10:     $[\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}] \leftarrow \text{SVD}(\mathbf{R}_x \mathbf{R}_y^\top)$ 
11:     $\delta \leftarrow \sigma_m(\mathbf{R}_x \mathbf{R}_y^\top)$ 
12:     $\widehat{\mathbf{\Sigma}} \leftarrow \max(\mathbf{\Sigma} - \delta \mathbf{I}_{2m}, \mathbf{0}_{2m \times 2m})$ 
13:     $\mathbf{A} \leftarrow \widehat{\mathbf{\Sigma}}^{1/2} \mathbf{U}^\top \mathbf{Q}_x^\top$ 
14:     $\mathbf{B} \leftarrow \widehat{\mathbf{\Sigma}}^{1/2} \mathbf{V}^\top \mathbf{Q}_y^\top$ 
15:   end if
16: end for
17: Output:  $\mathbf{A}$  and  $\mathbf{B}$ 

```

3 Sharper Analysis for COD

In this section, we provide a tighter error bound for COD. We let $\delta^{(t)}$ be the value of δ at time step t . If the algorithm does not enter the “then” section in the t -th step, then we have $\delta^{(t)} = 0$. Similarly, let $\mathbf{A}^{(t)}$, $\mathbf{B}^{(t)}$, $\mathbf{Q}_x^{(t)}$, $\mathbf{Q}_y^{(t)}$, $\mathbf{U}^{(t)}$, $\mathbf{\Sigma}^{(t)}$, $\mathbf{V}^{(t)}$ and $\widehat{\mathbf{\Sigma}}^{(t)}$ be the corresponding variables after the main loop has been executed for t times. Additionally, we use $\widehat{\mathbf{A}}^{(t)}$ and $\widehat{\mathbf{B}}^{(t)}$ to represent the matrices after insert operations (line

5-6) have been executed at the t -th iteration. We need the following two lemmas for proving our main results.

Lemma 3 (Mroueh, Marcheret, and Goel 2017). *The output matrices \mathbf{A} and \mathbf{B} of Algorithm 2 satisfy*

$$\|\mathbf{X}^\top \mathbf{Y} - \mathbf{A}^\top \mathbf{B}\|_2 \leq \sum_{t=1}^n \delta^{(t)} \quad (3)$$

and

$$\|\mathbf{A}^\top \mathbf{B}\|_* \leq \|\mathbf{X}\|_F \|\mathbf{Y}\|_F - m \sum_{t=1}^n \delta^{(t)}. \quad (4)$$

Lemma 4. *The output of Algorithm 2 holds that*

$$\|\mathbf{X}^\top \mathbf{Y}\|_* - \|\mathbf{A}^\top \mathbf{B}\|_* \leq \sum_{i=k+1}^d \sigma_i(\mathbf{X}^\top \mathbf{Y}) + k \sum_{t=1}^n \delta^{(t)}. \quad (5)$$

Lemma 4 is the key lemma of our proof. It improves the result in analysis of COD (Mroueh, Marcheret, and Goel 2017). The term $\sum_{i=k+1}^d \sigma_i(\mathbf{X}^\top \mathbf{Y})$ on the right-hand side of (5) considers the potential approximate low-rank structure of $\mathbf{X}^\top \mathbf{Y}$, which leads to a tighter error bound of COD as follows.

Theorem 1. *The output of Algorithm 2 holds that*

$$\|\mathbf{X}^\top \mathbf{Y} - \mathbf{A}^\top \mathbf{B}\|_2 \leq \frac{1}{m-k} \left(\|\mathbf{X}\|_F \|\mathbf{Y}\|_F - \|\mathbf{X}^\top \mathbf{Y}\|_k \right).$$

for any $k < m$.

Proof. Let $\Delta = \sum_{t=1}^n \delta^{(t)}$. Connecting inequality (4) in Lemma 3 and the result of Lemma 4, we have

$$\begin{aligned} m\Delta + \|\mathbf{X}^\top \mathbf{Y}\|_* - \|\mathbf{X}\|_F \|\mathbf{Y}\|_F \\ \leq \|\mathbf{X}^\top \mathbf{Y}\|_* - \|\mathbf{A}^\top \mathbf{B}\|_* \leq \sum_{i=k+1}^d \sigma_i(\mathbf{X}^\top \mathbf{Y}) + k\Delta, \end{aligned}$$

that is $\Delta \leq \frac{1}{m-k} (\|\mathbf{X}\|_F \|\mathbf{Y}\|_F - \|\mathbf{X}^\top \mathbf{Y}\|_k)$. Substituting above bound of Δ into inequality (4) of Lemma 3, we finish the proof of this theorem. \square

To achieve the accuracy that $\|\mathbf{X}^\top \mathbf{Y} - \mathbf{A}^\top \mathbf{B}\|_2 \leq \varepsilon$, the previous error bound (Lemma 2) requires the sketch size to be at least $m_1 = \frac{1}{\varepsilon} \|\mathbf{X}\|_F \|\mathbf{Y}\|_F$, while Theorem 1 only requires the sketch size $m_2 = k + \frac{1}{\varepsilon} (\|\mathbf{X}\|_F \|\mathbf{Y}\|_F - \|\mathbf{X}^\top \mathbf{Y}\|_k)$. When input matrices \mathbf{X} and \mathbf{Y} have strong correlation and approximate low-rank structure, m_2 could be much smaller than m_1 .

In addition, the error bound of Theorem 1 matches that of FD (Lemma 1) when $\mathbf{X} = \mathbf{Y}$:

$$\begin{aligned} \|\mathbf{X}^\top \mathbf{X} - \mathbf{A}^\top \mathbf{A}\|_2 &\leq \frac{1}{m-k} (\|\mathbf{X}\|_F \|\mathbf{X}\|_F - \sum_{i=1}^k \sigma_i(\mathbf{X}^\top \mathbf{X})) \\ &= \frac{1}{m-k} (\|\mathbf{X}\|_F^2 - \sum_{i=1}^k \sigma_i^2(\mathbf{X})) = \frac{1}{m-k} (\|\mathbf{X}\|_F^2 - \|\mathbf{X}_k\|_F^2). \end{aligned}$$

On the other hand, the previous error bound (Lemma 2) is worse than that of FD (Lemma 1) in the symmetric case of $\mathbf{X} = \mathbf{Y}$.

4 Space Lower Bounds Analysis

In this section, we show that COD is space optimal with respect to our new error bound in Theorem 1. We first introduce the following lemma for low-rank matrices.

Lemma 5 (Kapralov and Talwar 2013). *For each $\delta > 0$ there exists a set of matrices $\mathcal{Q} = \{\mathbf{Q}_1, \dots, \mathbf{Q}_N\}$ and $N = 2^{\Omega(\ell(d-\ell)\log(1/\delta))}$, where $\mathbf{Q}_i \in \mathbb{R}^{\ell \times d}$ with $\mathbf{Q}_i \mathbf{Q}_i^\top = \mathbf{I}_\ell$, such that $\|\mathbf{Q}_i \mathbf{Q}_j^\top\|_2 < 1 - \delta$.*

By using Lemma 5, we can construct a sets contains exponential number of matrices that each pair of them are not “too close”. The formalized result is shown in Lemma 6.

Lemma 6. *For each $\delta > 0$ and $d_x \leq d_y$ there exists a set of matrices $\widehat{\mathcal{Z}}_\ell = \{(\widehat{\mathbf{X}}^{(1)}, \widehat{\mathbf{Y}}^{(1)}), \dots, (\widehat{\mathbf{X}}^{(N)}, \widehat{\mathbf{Y}}^{(N)})\}$, where $N = 2^{\Omega(\ell(d_y-\ell)\log(1/\delta))}$ and $\widehat{\mathbf{X}}^{(i)} \in \mathbb{R}^{\ell \times d_x}$, $\widehat{\mathbf{Y}}^{(i)} \in \mathbb{R}^{\ell \times d_y}$ satisfy $\widehat{\mathbf{X}}^{(i)} \widehat{\mathbf{X}}^{(i)\top} = \mathbf{I}_\ell$ and $\widehat{\mathbf{Y}}^{(i)} \widehat{\mathbf{Y}}^{(i)\top} = \mathbf{I}_\ell$ for any $i = 1, \dots, n$ and*

$$\|\widehat{\mathbf{X}}^{(i)\top} \widehat{\mathbf{Y}}^{(i)\top} - \widehat{\mathbf{X}}^{(j)\top} \widehat{\mathbf{Y}}^{(j)\top}\|_2 > \sqrt{2\delta}$$

for any $j \neq i$.

Proof. Based on Lemma 5, there exist a set of matrices $\mathcal{Y} = \{\widehat{\mathbf{Y}}^{(1)}, \dots, \widehat{\mathbf{Y}}^{(N)}\}$, where $N = 2^{\Omega(\ell(d-\ell)\log(1/\delta))}$; and $\widehat{\mathbf{Y}}^{(i)} \in \mathbb{R}^{\ell \times d}$ satisfies $\widehat{\mathbf{Y}}^{(i)} \widehat{\mathbf{Y}}^{(i)\top} = \mathbf{I}_\ell$ and $\|\widehat{\mathbf{Y}}^{(i)} \widehat{\mathbf{Y}}^{(j)\top}\| < 1 - \delta$. We further set $\widehat{\mathbf{X}}^{(i)} = [\mathbf{I}_\ell, \mathbf{0}_{\ell \times (d_x-\ell)}]$. We have

$$\begin{aligned} & \|\widehat{\mathbf{X}}^{(i)\top} \widehat{\mathbf{Y}}^{(i)} - \widehat{\mathbf{X}}^{(j)\top} \widehat{\mathbf{Y}}^{(j)}\|_2^2 = \|\widehat{\mathbf{Y}}^{(i)} - \widehat{\mathbf{Y}}^{(j)}\|_2^2 \\ & = \|(\widehat{\mathbf{Y}}^{(i)} - \widehat{\mathbf{Y}}^{(j)})(\widehat{\mathbf{Y}}^{(i)\top} - \widehat{\mathbf{Y}}^{(j)\top})\|_2 \\ & \geq 2 - \|\widehat{\mathbf{Y}}^{(j)} \widehat{\mathbf{Y}}^{(i)\top} + \widehat{\mathbf{Y}}^{(i)} \widehat{\mathbf{Y}}^{(j)\top}\|_2 \geq 2\delta, \end{aligned}$$

where we use the definition of $\widehat{\mathbf{X}}^{(i)}$, $\widehat{\mathbf{Y}}^{(i)}$ and the fact $\|\mathbf{A}^\top \mathbf{A}\|_2 = \|\mathbf{A}\|_2^2$. \square

Then we present a lower bound of space complexity for approximate matrix multiplication, which matches the memory cost of COD. Hence, we can conclude that COD is space optimal with respect to the guaranteed accuracy in Theorem 1.

Theorem 2. *We consider any matrix sketching algorithm with inputs as $\mathbf{X} \in \mathbb{R}^{n \times d_x}$ and $\mathbf{Y} \in \mathbb{R}^{n \times d_y}$ and outputs $\mathbf{A} \in \mathbb{R}^{m \times d_x}$ and $\mathbf{B} \in \mathbb{R}^{m \times d_y}$ with guarantee*

$$\|\mathbf{X}^\top \mathbf{Y} - \mathbf{A}^\top \mathbf{B}\|_2 \leq \frac{1}{m-k} \left(\|\mathbf{X}\|_F \|\mathbf{Y}\|_F - \|\mathbf{X}^\top \mathbf{Y}\|_k \right)$$

for any $k < m$. Assuming that a constant number of bits is required to describe a word (i.e., a unit of memory), then the algorithm requires at least $\Omega(m(d_x + d_y))$ bits of space.

Proof. Without loss of generality, we suppose that $d_y \geq d_x$. Let $\widehat{\mathcal{Z}}_\ell = \{(\widehat{\mathbf{X}}^{(1)}, \widehat{\mathbf{Y}}^{(1)}), \dots, (\widehat{\mathbf{X}}^{(N)}, \widehat{\mathbf{Y}}^{(N)})\}$ be the set of matrices defined in Lemma 6 with $\ell = m/4$, $\delta = 1/8$ and $N = 2^{\Omega(\frac{m}{4} \cdot (d_y - m/4) \log(8))}$. We construct matrices $\mathbf{X}^{(i)} = [\widehat{\mathbf{X}}^{(i)}; \mathbf{0}_{(n-m/4) \times d_x}] \in \mathbb{R}^{n \times d_x}$ and $\mathbf{Y}^{(i)} = [\widehat{\mathbf{Y}}^{(i)}; \mathbf{0}_{(n-m/4) \times d_y}] \in \mathbb{R}^{n \times d_y}$ for $i = 1, \dots, N$. Then we have $\mathcal{Z}_\ell = \{(\mathbf{X}^{(i)}, \mathbf{Y}^{(i)})\}_{i=1}^N$ which satisfies $\|\mathbf{X}^{(i)\top} \mathbf{Y}^{(i)} - \mathbf{X}^{(j)\top} \mathbf{Y}^{(j)}\|_2 > 1/2$ for each $i \neq j$. Let

(\mathbf{A}, \mathbf{B}) be the output of the matrix sketching algorithm with input $(\mathbf{X}^{(i)}, \mathbf{Y}^{(i)})$. The guarantee of the algorithm indicates

$$\|\mathbf{X}^{(i)\top} \mathbf{Y}^{(i)} - \mathbf{A}^\top \mathbf{B}\|_2 \leq \frac{1}{m-k} \left(\frac{m}{4} - k \right) \leq \frac{1}{4}.$$

Hence, each (\mathbf{A}, \mathbf{B}) only encodes one matrix pencil in \mathcal{Z}_ℓ (the product of the matrices), which means that the lower bound of space complexity to attach the desired accuracy is $\log_2 N = \Omega(md_y) = \Omega(m(d_x + d_y))$ bits. \square

5 Sparse Co-Occurring Directions

In this section, we proposed a variant of COD for sparse AMM. We also prove its error bound is similar to our improved result of COD.

5.1 The Algorithm

We describe details of our sparse co-occurring directions (SCOD) in Algorithm 4. The procedure of SCOD maintains the sparse data in two buffer matrices \mathbf{X}' and \mathbf{Y}' . The algorithm restricts the non-zero entries in buffers to be less than $m(d_x + d_y)$ and the number of rows of each buffer is at most $d_x + d_y$. When the buffers are full, we perform subspace power method (SPM) (Woodruff 2014b; Musco and Musco 2015) to approximate the data in the buffers by low-rank matrices $\widetilde{\mathbf{X}} \in \mathbb{R}^{m \times d_x}$ and $\widetilde{\mathbf{Y}} \in \mathbb{R}^{m \times d_y}$ such that $\mathbf{X}'^\top \mathbf{Y}' \approx \widetilde{\mathbf{X}}^\top \widetilde{\mathbf{Y}}$. We present the procedure of SPM in Algorithm 3.

Let $\widetilde{\mathbf{X}}^{(i)}$ and $\widetilde{\mathbf{Y}}^{(i)}$ be the results of $\widetilde{\mathbf{X}}$ and $\widetilde{\mathbf{Y}}$ after Algorithm 4 has executed “then” section for i -times. Define $\mathbf{C} = [\widetilde{\mathbf{X}}^{(1)}; \dots; \widetilde{\mathbf{X}}^{(T)}]$ and $\mathbf{D} = [\widetilde{\mathbf{Y}}^{(1)}; \dots; \widetilde{\mathbf{Y}}^{(T)}]$ where T is the number of total times we enter “then” section of the algorithm. Then \mathbf{C} and \mathbf{D} are the estimators of \mathbf{X} and \mathbf{Y} respectively and the procedure of SCOD can be regarded as as running standard COD on input matrices \mathbf{C} and \mathbf{D} in streaming fashion. Since the row numbers of buffers \mathbf{X}' and \mathbf{Y}' could be much larger than m , the operations on dense matrices (line 14-20) will not be executed frequently. Hence, SCOD is much more efficient than COD for sparse inputs.

Algorithm 3 Subspace Power Method (SPM)

- 1: **Input:** $\mathbf{M} \in \mathbb{R}^{d_1 \times d_2}$, target rank m and integer $q > 0$
 - 2: $\mathbf{G} = [G_{ij}] \in \mathbb{R}^{d_2 \times m}$, where $G_{ij} \sim \mathcal{N}(0, 1)$ i.i.d
 - 3: $\mathbf{K} = (\mathbf{M}\mathbf{M}^\top)^q \mathbf{M}\mathbf{G} \in \mathbb{R}^{d_1 \times m}$
 - 4: $\mathbf{Z} \leftarrow$ orthonormal column basis of \mathbf{K}
 - 5: **Output:** \mathbf{Z}
-

5.2 Analysis of Error Bound

The analysis of SCOD is more challenging than sparse frequent directions (SFD) (Ghashami, Liberty, and Phillips 2016) which only addresses the case of $\mathbf{X} = \mathbf{Y}$. The reason is the “mergeability property” of FD (Ghashami et al. 2016; Desai, Ghashami, and Phillips 2016; Ghashami, Liberty, and Phillips 2016) only works for Frobenius norm and it is not applicable to COD.

Algorithm 4 Sparse Co-Occurring Directions (SCOD)

```
1: Input:  $\mathbf{X} \in \mathbb{R}^{n \times d_x}$ ,  $\mathbf{Y} \in \mathbb{R}^{n \times d_y}$ , sketch size  $m$ , failure
   probability  $\delta$  and sequence  $\{q_i\}_{i=1,2,\dots}$ 
2:  $i = 0$ 
3:  $\mathbf{A} \leftarrow \mathbf{0}_{m \times d_x}$ ,  $\mathbf{B} \leftarrow \mathbf{0}_{m \times d_y}$ 
4:  $\mathbf{X}' \leftarrow$  empty,  $\mathbf{Y}' \leftarrow$  empty
5: for  $t = 1, 2, \dots, n$ 
6:    $\mathbf{X}' \leftarrow [\mathbf{X}'; (\mathbf{x}^{(t)})^\top]$ ,  $\mathbf{Y}' \leftarrow [\mathbf{Y}'; (\mathbf{y}^{(t)})^\top]$ 
7:   if  $\text{nnz}(\mathbf{X}') + \text{nnz}(\mathbf{Y}') > m(d_x + d_y)$  or  $t = n$ 
     or  $\text{rows}(\mathbf{X}') = d_x + d_y$  or  $\text{rows}(\mathbf{Y}') = d_x + d_y$ 
     then
8:      $\mathbf{Z} = \text{SubspacePowerMethod}(\mathbf{X}'^\top \mathbf{Y}', m, q_i)$ 
9:      $[\tilde{\mathbf{U}}, \tilde{\Sigma}, \tilde{\mathbf{V}}] = \text{SVD}(\mathbf{Z}^\top \mathbf{X}'^\top \mathbf{Y}')$ 
10:     $\tilde{\mathbf{X}} \leftarrow \tilde{\Sigma}^{1/2} \tilde{\mathbf{U}}^\top \mathbf{Z}^\top$ 
11:     $\tilde{\mathbf{Y}} \leftarrow \tilde{\Sigma}^{1/2} \tilde{\mathbf{V}}^\top$ 
12:     $\mathbf{A} \leftarrow [\mathbf{A}; \tilde{\mathbf{X}}]$ 
13:     $\mathbf{B} \leftarrow [\mathbf{B}; \tilde{\mathbf{Y}}]$ 
14:     $(\mathbf{Q}_x, \mathbf{R}_x) \leftarrow \text{QR}(\mathbf{A}^\top)$ 
15:     $(\mathbf{Q}_y, \mathbf{R}_y) \leftarrow \text{QR}(\mathbf{B}^\top)$ 
16:     $[\mathbf{U}, \Sigma, \mathbf{V}] \leftarrow \text{SVD}(\mathbf{R}_x \mathbf{R}_y^\top)$ 
17:     $\delta \leftarrow \sigma_m(\mathbf{R}_x \mathbf{R}_y^\top)$ 
18:     $\hat{\Sigma} \leftarrow \max(\Sigma - \delta \mathbf{I}_m, \mathbf{0}_{m \times m})$ 
19:     $\mathbf{A} \leftarrow \hat{\Sigma}^{1/2} \mathbf{U}^\top \mathbf{Q}_x^\top$ 
20:     $\mathbf{B} \leftarrow \hat{\Sigma}^{1/2} \mathbf{V}^\top \mathbf{Q}_y^\top$ 
21:     $\mathbf{X}' \leftarrow$  empty,  $\mathbf{Y}' \leftarrow$  empty
22:     $i \leftarrow i + 1$ 
23:   end if
24: end for
25: Output:  $\mathbf{A}$  and  $\mathbf{B}$ 
```

The approximation error of SCOD comes from two parts: the compressing error from sub-routine SPM and the merge error from estimating $\mathbf{C}^\top \mathbf{D}$ by $\mathbf{A}^\top \mathbf{B}$. We first consider a single call of SPM, which approximation error can be bounded as follows.

Lemma 7. *Let $q = \tilde{\Theta}(\log(md_1/p)/\varepsilon)$ for Algorithm 3, then the output \mathbf{Z} satisfies $\|\mathbf{M} - \mathbf{Z}\mathbf{Z}^\top \mathbf{M}\|_2 \leq (1 + \varepsilon)\sigma_{m+1}(\mathbf{M})$ with probability at least $1 - p$.*

Based on Lemma 7, we can bound the total compressing error of SCOD by the following lemma.

Lemma 8. *Setting $q_i = \tilde{\Theta}(\log(md_1/p_i)/\varepsilon)$ and $p_i = \delta/2i^2$, then we have then Algorithm 4 holds that*

$$\|\mathbf{X}^\top \mathbf{Y} - \mathbf{C}^\top \mathbf{D}\|_2 \leq \frac{1 + \varepsilon}{m - k} (\|\mathbf{X}\|_F \|\mathbf{Y}\|_F - \|\mathbf{X}^\top \mathbf{Y}\|_k),$$

for any $k < m$ and $\varepsilon > 0$ with probability $1 - \delta$.

Proof. Let $\mathbf{X}^{(i)}$ and $\mathbf{Y}^{(i)}$ be the value of \mathbf{X}' and \mathbf{Y}' when we execute subspace power methods at i -th time in line 8 of Algorithm 4, then we have

$$\mathbf{X} = [\mathbf{X}^{(1)}; \dots; \mathbf{X}^{(T)}] \quad \text{and} \quad \mathbf{Y} = [\mathbf{Y}^{(1)}; \dots; \mathbf{Y}^{(T)}].$$

Using Lemma 7 with $\mathbf{M} = \tilde{\mathbf{X}}^{(i)\top} \tilde{\mathbf{Y}}^{(i)}$ and $q = q_i$, then with probability $1 - p_i$, we have

$$\begin{aligned} & \left\| \tilde{\mathbf{X}}^{(i)\top} \tilde{\mathbf{Y}}^{(i)} - \mathbf{X}'^{(i)\top} \mathbf{Y}'^{(i)} \right\|_2 \leq (1 + \varepsilon)\sigma_{m+1}(\mathbf{X}'^{(i)\top} \mathbf{Y}'^{(i)}) \\ & \leq \frac{1 + \varepsilon}{m - k} \left(\|\mathbf{X}'^{(i)}\|_F \|\mathbf{Y}'^{(i)}\|_F - \|\mathbf{X}'^{(i)\top} \mathbf{Y}'^{(i)}\|_k \right), \end{aligned} \quad (6)$$

where the last step use Srebro, Rennie, and Jaakkola's (2005) Lemma 1 such that $\|\mathbf{X}'^{(i)\top} \mathbf{Y}'^{(i)}\|_* \leq \|\mathbf{X}'^{(i)}\|_F \|\mathbf{Y}'^{(i)}\|_F$. Summing over inequality (6) with $i = 1, \dots, T$, we have

$$\begin{aligned} \|\mathbf{X}^\top \mathbf{Y} - \mathbf{C}^\top \mathbf{D}\|_2 & \leq \sum_{t=1}^T \|\mathbf{X}'^{(i)} \mathbf{Y}'^{(i)} - \tilde{\mathbf{X}}^{(i)\top} \tilde{\mathbf{Y}}^{(i)}\|_2 \\ & \leq \frac{1 + \varepsilon}{m - k} \sum_{t=1}^T \left(\|\mathbf{X}'^{(i)}\|_F \|\mathbf{Y}'^{(i)}\|_F - \|\mathbf{X}'^{(i)\top} \mathbf{Y}'^{(i)}\|_k \right) \\ & \leq \frac{1 + \varepsilon}{m - k} \left(\|\mathbf{X}\|_F \|\mathbf{Y}\|_F - \|\mathbf{X}^\top \mathbf{Y}\|_k \right) \end{aligned}$$

with probability $1 - \delta$. The last inequality is based on the Cauchy-Schwarz inequality and the triangle inequality of Ky Fan k -norm. Note that the failure probability is no more than $p_1 + \dots + p_T = \frac{\delta}{2} \sum_{i=1}^T 1/i^2 \leq \delta$. \square

Unlike SFD (Ghashami et al. 2016) which introduces a verifying step to boost the success probability, our method instead requires q_i to be increased logarithmically to ensure the error bound of SCOD holds with probability at least $1 - \delta$ for given $\delta \in (0, 1)$. Another important property of SCOD is that the compression step shrink the magnitude of the product of input matrices. The steps in line 10-11 of Algorithm 4 balance the singular values of $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{Y}}$, which leads to the following lemma:

Lemma 9. *Algorithm 4 holds that*

$$\|\tilde{\mathbf{X}}^{(i)}\|_F \|\tilde{\mathbf{Y}}^{(i)}\|_F \leq \|\mathbf{X}'^{(i)}\|_F \|\mathbf{Y}'^{(i)}\|_F.$$

Since the analysis of merging error is similar to standard COD, we can establish the error bound of SCOD by using above lemmas.

Theorem 3. *Setting $q_i = \tilde{\Theta}(\log(md_1/p_i)/\varepsilon)$ with constant $\varepsilon > 0$ and $p_i = \delta/2i^2$, with probability $1 - \delta$, the outputs \mathbf{A} and \mathbf{B} of Algorithm 4 hold that*

$$\begin{aligned} & \|\mathbf{X}^\top \mathbf{Y} - \mathbf{A}^\top \mathbf{B}\|_2 \\ & \leq \left(\frac{2 + \varepsilon}{m - k} + \frac{(1 + \varepsilon)k}{(m - k)^2} \right) (\|\mathbf{X}\|_F \|\mathbf{Y}\|_F - \|\mathbf{X}^\top \mathbf{Y}\|_k) \end{aligned}$$

for all $k < m$.

Proof. Consider that \mathbf{A} and \mathbf{B} can be viewed as the output of running Algorithm 2 with input matrices

$$\mathbf{C} = [\tilde{\mathbf{X}}^{(1)}; \dots; \tilde{\mathbf{X}}^{(T)}] \quad \text{and} \quad \mathbf{D} = [\tilde{\mathbf{Y}}^{(1)}; \dots; \tilde{\mathbf{Y}}^{(T)}].$$

Following the proof of Theorem 1, we have

$$\begin{aligned}
& \left\| \mathbf{C}^\top \mathbf{D} - \mathbf{A}^\top \mathbf{B} \right\|_2 \\
& \leq \frac{1}{m-k} \left(\sum_{i=1}^T \|\tilde{\mathbf{X}}^{(i)}\|_F \|\tilde{\mathbf{Y}}^{(i)}\|_F - \|\mathbf{C}^\top \mathbf{D}\|_k \right) \\
& \leq \frac{\sum_{i=1}^T \|\tilde{\mathbf{X}}^{(i)}\|_F \|\tilde{\mathbf{Y}}^{(i)}\|_F - \|\mathbf{X}^\top \mathbf{Y}\|_k + \|\mathbf{X}^\top \mathbf{Y} - \mathbf{C}^\top \mathbf{D}\|_k}{m-k} \\
& \leq \frac{\sum_{i=1}^T \|\tilde{\mathbf{X}}^{(i)}\|_F \|\tilde{\mathbf{Y}}^{(i)}\|_F - \|\mathbf{X}^\top \mathbf{Y}\|_k + k \|\mathbf{X}^\top \mathbf{Y} - \mathbf{C}^\top \mathbf{D}\|_2}{m-k} \\
& \leq \frac{1}{m-k} \left(\sum_{i=1}^T \|\mathbf{X}'^{(i)}\|_F \|\mathbf{Y}'^{(i)}\|_F - \|\mathbf{X}^\top \mathbf{Y}\|_k \right) \\
& \quad + \frac{(1+\varepsilon)k}{(m-k)^2} \left(\|\mathbf{X}\|_F \|\mathbf{Y}\|_F - \|\mathbf{X}^\top \mathbf{Y}\|_k \right) \\
& \leq \left(\frac{1}{m-k} + \frac{(1+\varepsilon)k}{(m-k)^2} \right) \left(\|\mathbf{X}\|_F \|\mathbf{Y}\|_F - \|\mathbf{X}^\top \mathbf{Y}\|_k \right)
\end{aligned}$$

where we use Lemma 8, 9 and triangle inequality.

Combing above results and Lemma 8, we have

$$\begin{aligned}
& \left\| \mathbf{X}^\top \mathbf{Y} - \mathbf{A}^\top \mathbf{B} \right\|_2 \\
& \leq \left\| \mathbf{X}^\top \mathbf{Y} - \mathbf{C}^\top \mathbf{D} \right\|_2 + \left\| \mathbf{C}^\top \mathbf{D} - \mathbf{A}^\top \mathbf{B} \right\|_2 \\
& \leq \left(\frac{2+\varepsilon}{m-k} + \frac{(1+\varepsilon)k}{(m-k)^2} \right) \left(\|\mathbf{X}\|_F \|\mathbf{Y}\|_F - \|\mathbf{X}^\top \mathbf{Y}\|_k \right),
\end{aligned}$$

with probability at least $1 - \delta$. \square

5.3 Complexity Analysis

We use the constant-word-size model for our analysis like that of sparse FD (Ghashami et al. 2016). We suppose floating point numbers are represented by a constant number of bits, random access into memory requires $\mathcal{O}(1)$ time and multiplying a sparse matrix \mathbf{M} by a dense vector requires $\mathcal{O}(\text{nnz}(\mathbf{M}))$ time and storing \mathbf{M} requires $\mathcal{O}(\text{nnz}(\mathbf{M}))$ space.

The procedure of SCOD (Algorithm 4) implies the buffer \mathbf{X}' and \mathbf{Y}' is sparse and contains at most $m(d_x + d_y)$ non-zero entries and it is not difficult to verify that all dense matrices in the algorithm cost no more than $\mathcal{O}(m(d_x + d_y))$ space. Hence, the space complexity of SCOD is $\mathcal{O}(m(d_x + d_y))$ in total which is the same as COD (Algorithm 2).

Then we analyze the time complexity of SCOD. The constraints on buffer size means we have

$$T \leq \frac{\text{nnz}(\mathbf{X}) + \text{nnz}(\mathbf{Y})}{m(d_x + d_y)} + \frac{n}{d_x + d_y}.$$

Since each QR factorization or SVD on $m \times d$ matrix cost $\mathcal{O}(m^2 d)$ time, the operation on dense matrices of Algorithm 4 from line 9-20 requires at most

$$\mathcal{O}(m^2(d_x + d_y)T) = \mathcal{O}(m(\text{nnz}(\mathbf{X}) + \text{nnz}(\mathbf{Y})) + m^2 n).$$

Note that SCOD calls SPM with input $\mathbf{M} = \mathbf{X}'^\top \mathbf{Y}'$. Since both \mathbf{X}' and \mathbf{Y}' are sparse, it is unnecessary to construct \mathbf{M} explicitly and we can multiply \mathbf{X}' and \mathbf{Y}' on \mathbf{G} separately in line 3 of Algorithm 3. Then the time complexity of executing SPM needs $\mathcal{O}(mq_i(\text{nnz}(\mathbf{X}'^{(i)}) + \text{nnz}(\mathbf{Y}'^{(i)})) + m^2 d_x)$ when the algorithm enters “then” section at the i -th time. Following

the upper bound of T and the setting of q_i in Theorem 3, the calls of SPM in Algorithm 3 entirely takes at most

$$\begin{aligned}
& \mathcal{O} \left(\sum_{i=1}^T \left(mq_i(\text{nnz}(\mathbf{X}'^{(i)}) + \text{nnz}(\mathbf{Y}'^{(i)})) + m^2 d_x \right) \right) \\
& \leq \mathcal{O}(mq_T(\text{nnz}(\mathbf{X}) + \text{nnz}(\mathbf{Y})) + Tm^2 d_x) \\
& = \tilde{\mathcal{O}}(m(\text{nnz}(\mathbf{X}) + \text{nnz}(\mathbf{Y})) + m^2 n).
\end{aligned}$$

Hence, the total time complexity of proposed SCOD is $\tilde{\mathcal{O}}(m(\text{nnz}(\mathbf{X}) + \text{nnz}(\mathbf{Y})) + m^2 n)$.

6 Numerical Experiments

In this section, we empirically compare the proposed sparse co-occurring directions (SCOD) with frequent direction based AMM (FD-AMM) (Ye, Luo, and Zhang 2016), co-occurring directions (COD) (Mroueh, Marcheret, and Goel 2017) and sparse frequent direction based AMM algorithm (SFD-AMM)¹. Instead of increasing q_i logarithmically as the analysis of Theorem 2, we fix $q_i = 5$ in our experiment since the empirical error arise from subspace power method is very small in practice.

We evaluate performance of all algorithms on cross-language datasets: Amazon Product Reviews (APR), PAN-PC-11 (PAN), JRC Acquis (JRC) and Europarl (EURO) which contain millions of English (EN), French (FR) and Spanish (ES) sentences (Prettenhofer and Stein 2010; Potthast et al. 2010, 2011; Koehn 2005). We use bag-of-words feature for our experiments. All of input matrices are large but very sparse and we summary the parameters in Table 1.

We demonstrate sketch-error and time-error comparisons in Figure 1 and 2 respectively. It is apparently that SCOD always performs better than all baseline algorithms. We do not include the curve of FD-AMM and COD in time-error comparison because these two algorithms take much more time than others. Due to the limit of space, we defer the result of sketch-time comparison and detailed computing infrastructure in appendix.

7 Conclusion

In this paper, we first improved the error bound of a deterministic sketching algorithm COD for streaming AMM problem. In symmetric case, our result matches the error bound of classical algorithm FD. We also proved COD matches the space lower bound complexity to achieve our error bound. In addition, we proposed a sparse variant of COD with a reasonable error bound. The experimental results show that the proposed algorithm has better performance than baseline methods in practice.

It would be interesting to borrow the idea of this paper to establish better theoretical guarantees and streaming algorithms for more classical machine learning and statistical models such as canonical correlation analysis (Hotelling 1992; Avron et al. 2013; Ye, Luo, and Zhang 2016), generalized eigenvector decomposition (Bhatia et al. 2018; Golub and Loan 1996) and spectral co-clustering (Dhillon 2001).

¹SFD-AMM refers to the method simply replacing FD step in FD-AMM with sparse frequent directions (Ghashami et al. 2016). We provide more detailed discussion about SFD-AMM in appendix.

Dataset	n	d_x	d_y	density(\mathbf{X})	density(\mathbf{Y})
APR (EN-FR)	2.32×10^4	2.80×10^4	4.28×10^4	6.31×10^{-4}	4.53×10^{-4}
PAN (EN-FR)	8.90×10^4	5.12×10^4	9.96×10^4	4.38×10^{-4}	2.43×10^{-4}
JRC (EN-FR)	1.50×10^5	1.72×10^5	1.87×10^5	1.65×10^{-4}	1.64×10^{-4}
JRC (EN-ES)	1.50×10^5	1.72×10^5	1.92×10^5	1.65×10^{-4}	1.60×10^{-4}
JRC (FR-ES)	1.50×10^5	1.87×10^5	1.92×10^5	1.64×10^{-4}	1.60×10^{-4}
EURO (EN-FR)	4.76×10^5	7.25×10^4	8.77×10^4	3.46×10^{-4}	3.65×10^{-4}
EURO (EN-ES)	4.76×10^5	7.25×10^4	8.80×10^4	3.46×10^{-4}	3.47×10^{-4}
EURO (FR-ES)	4.76×10^5	8.77×10^4	8.80×10^4	3.65×10^{-4}	3.47×10^{-4}

Table 1: We present the size and density of datasets used in our experiments, where $\text{density}(\mathbf{X}) = \text{nnz}(\mathbf{X})/nd_x$ and $\text{density}(\mathbf{Y}) = \text{nnz}(\mathbf{Y})/nd_y$. All of these datasets are publicly available (Ferrero et al. 2016).

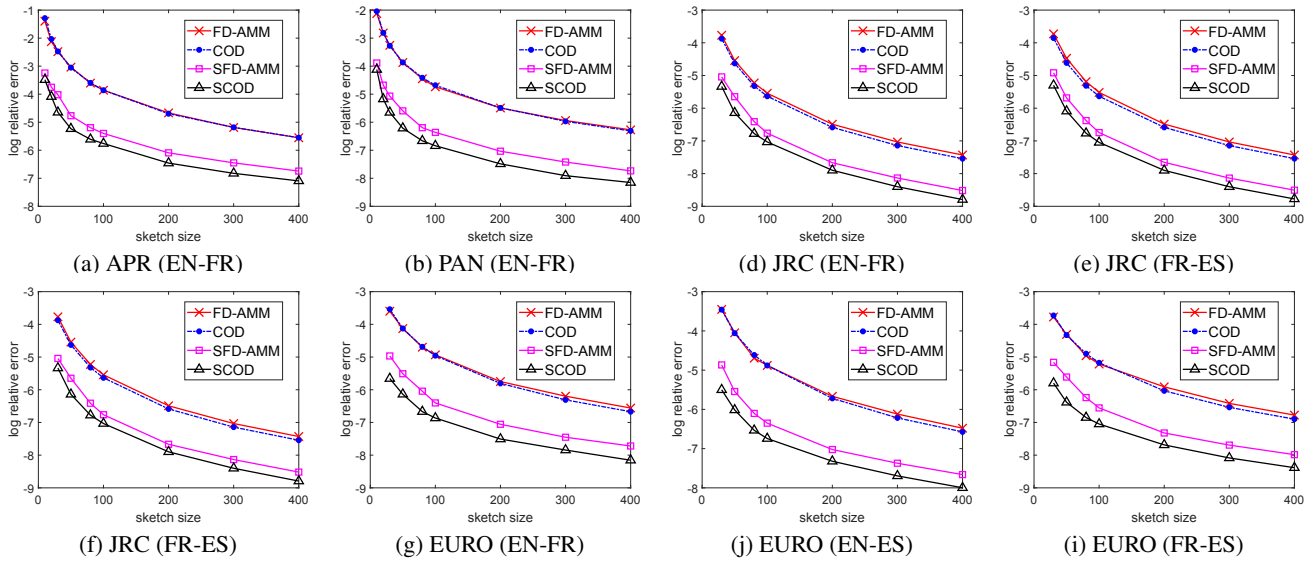


Figure 1: The plot of sketch size against relative spectral norm error

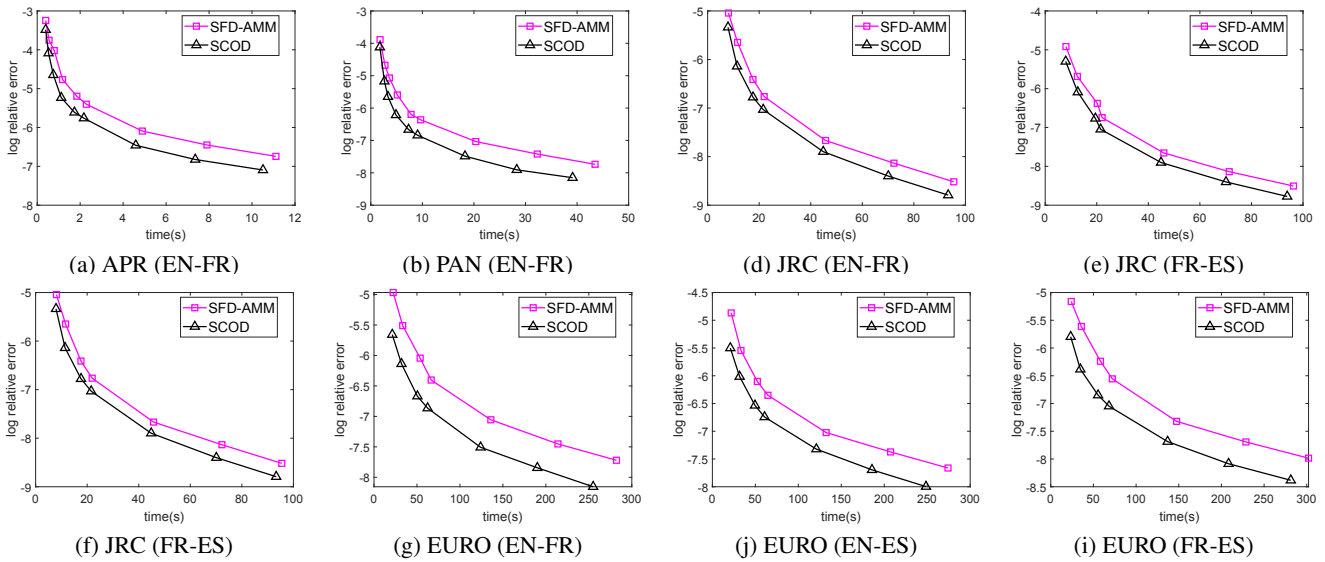


Figure 2: The plot of time (s) against relative spectral norm error

Acknowledgements

Luo Luo is supported by GRF 16201320.

References

- Avron, H.; Boutsidis, C.; Toledo, S.; and Zouzias, A. 2013. Efficient dimensionality reduction for canonical correlation analysis. In *ICML*.
- Bhatia, K.; Pacchiano, A.; Flammarion, N.; Bartlett, P. L.; and Jordan, M. I. 2018. Gen-Oja: Simple & efficient algorithm for streaming generalized eigenvector computation. In *NIPS*.
- Chen, C.; Luo, L.; Zhang, W.; Yu, Y.; and Lian, Y. 2020. Efficient and Robust High-Dimensional Linear Contextual Bandits. In *IJCAI*.
- Clarkson, K. L.; and Woodruff, D. P. 2017. Low-rank approximation and regression in input sparsity time. *Journal of the ACM* 63(6): 1–45.
- Desai, A.; Ghashami, M.; and Phillips, J. M. 2016. Improved practical matrix sketching with guarantees. *IEEE Transactions on Knowledge and Data Engineering* 28(7): 1678–1690.
- Dhillon, I. S. 2001. Co-clustering documents and words using bipartite spectral graph partitioning. In *SIGKDD*.
- Dickens, C. 2020. Ridge Regression with Frequent Directions: Statistical and Optimization Perspectives. *arXiv preprint:2011.03607*.
- Drineas, P.; Kannan, R.; and Mahoney, M. W. 2006. Fast Monte Carlo algorithms for matrices I: Approximating matrix multiplication. *SIAM Journal on Computing* 36(1): 132–157.
- Ferrero, J.; Agnes, F.; Besacier, L.; and Schwab, D. 2016. A multilingual, multi-style and multi-granularity dataset for cross-language textual similarity detection. In *LREC*.
- Ghashami, M.; Liberty, E.; and Phillips, J. M. 2016. Efficient frequent directions algorithm for sparse matrices. In *SIGKDD*.
- Ghashami, M.; Liberty, E.; Phillips, J. M.; and Woodruff, D. P. 2016. Frequent directions: Simple and deterministic matrix sketching. *SIAM Journal on Computing* 45(5): 1762–1792.
- Ghashami, M.; and Phillips, J. M. 2014. Relative errors for deterministic low-rank matrix approximations. In *SODA*.
- Golub, G. H.; and Loan, C. F. V. 1996. Matrix computations. *Johns Hopkins University Press, 3rd edition*.
- Hotelling, H. 1992. Relations between two sets of variates. In *Breakthroughs in statistics*, 162–190. Springer.
- Huang, Z. 2019. Near optimal frequent directions for sketching dense and sparse matrices. *Journal of Machine Learning Research* 20(56): 1–23.
- Kapralov, M.; and Talwar, K. 2013. On differentially private low rank approximation. In *SODA*.
- Koehn, P. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, 79–86. Cite-seer.
- Kuzborskij, I.; Cella, L.; and Cesa-Bianchi, N. 2019. Efficient linear bandits through matrix sketching. In *AISTATS*.
- Liberty, E. 2013. Simple and deterministic matrix sketching. In *SIGKDD*.
- Luo, H.; Agarwal, A.; Cesa-Bianchi, N.; and Langford, J. 2016. Efficient second order online learning by sketching. In *NIPS*.
- Luo, L.; Chen, C.; Zhang, Z.; Li, W.-J.; and Zhang, T. 2019. Robust Frequent Directions with Application in Online Learning. *Journal of Machine Learning Research* 20(45): 1–41.
- Luo, L.; Zhang, W.; Zhang, Z.; Zhu, W.; Zhang, T.; and Pei, J. 2018. Sketched follow-the-regularized-leader for online factorization machine. In *SIGKDD*.
- Misra, J.; and Gries, D. 1982. Finding repeated elements. *Science of computer programming* 2(2): 143–152.
- Mroueh, Y.; Marcheret, E.; and Goel, V. 2017. Co-Occurring directions sketching for approximate matrix multiply. In *AISTATS*.
- Musco, C.; and Musco, C. 2015. Randomized block Krylov methods for stronger and faster approximate singular value decomposition. In *NIPS*.
- Potthast, M.; Barrón-Cedeño, A.; Stein, B.; and Rosso, P. 2011. Cross-language plagiarism detection. *Language Resources and Evaluation* 45(1): 45–62.
- Potthast, M.; Stein, B.; Barrón-Cedeño, A.; and Rosso, P. 2010. An evaluation framework for plagiarism detection. In *COLING*.
- Prettenhofer, P.; and Stein, B. 2010. Cross-language text classification using structural correspondence learning. In *ACL*.
- Sarlos, T. 2006. Improved approximation algorithms for large matrices via random projections. In *FOCS*.
- Shi, B.; and Phillips, J. M. 2020. A deterministic streaming sketch for ridge regression. *arXiv preprint:2002.02013*.
- Srebro, N.; Rennie, J.; and Jaakkola, T. S. 2005. Maximum-margin matrix factorization. In *NIPS*.
- Wan, Y.; and Zhang, L. 2020. Approximate Multiplication of Sparse Matrices with Limited Space. *arXiv preprint:2009.03527*.
- Wegelin, J. A. 2000. A survey of Partial Least Squares (PLS) methods, with emphasis on the two-block case. *University of Washington, Technical Report*.
- Woodruff, D. P. 2014a. Low rank approximation lower bounds in row-update streams. In *NIPS*.
- Woodruff, D. P. 2014b. Sketching as a Tool for Numerical Linear Algebra. *Foundations and Trends in Theoretical Computer Science* 10(1-2): 1–157.
- Ye, Q.; Luo, L.; and Zhang, Z. 2016. Frequent direction algorithms for approximate matrix multiplication with applications in CCA. In *IJCAI*.