A Fast Approximate Algorithm for Large-Scale Latent Semantic Indexing

Dell Zhang
SCSIS
Birkbeck, University of London
London WC1E 7HX, UK
dell.z@ieee.org

Zheng Zhu
SCSIS
Birkbeck, University of London
London WC1E 7HX, UK
zheng@dcs.bbk.ac.uk

Abstract

Latent Semantic Indexing (LSI) is an effective method to discover the underlying semantic structure of data. It has numerous applications in information retrieval and data mining. However, the computational complexity of LSI may be prohibitively high when applied to very large datasets. In this paper, we present a fast approximate algorithm for large-scale LSI that is conceptually simple and theoretically justified. Our main contribution is to show that the proposed algorithm has provable error bound and linear computational complexity.

1 Introduction

In many problem domains, the data can be naturally modelled as a matrix $\mathbf{A}$ such that each column corresponds to a feature and each row describes an instances as a point or vector in the feature space. For example, a collection of m documents that contain n distinctive terms can be represented as an $m \times n$ document-term matrix [17].

The technique of Latent Semantic Indexing (LSI) [6, 3, 20, 14] employs truncated Singular Value Decomposition (SVD) [13, 11] (see Section 2) to find the best low-rank description of $\mathbf{A} \in \mathbb{R}^{m \times n}$, i.e., the matrix $\mathbf{D} \in \mathbb{R}^{m \times n}$ of rank k ($k \ll m, n$) with minimum error $\|\mathbf{A} - \mathbf{D}\|_F$ where $\|\cdot\|_F$ denotes the Frobenius norm. Since the intrinsic dimensionality of data is usually much smaller than n, the low-rank matrix $\mathbf{D}$ actually reveals the underlying semantic structure of the original data matrix $\mathbf{A}$. For example, it has been shown that using $\mathbf{D}$ instead of $\mathbf{A}$ for information retrieval can effectively deal with the tough problem of *synonymy* and *polysemy* [6].

There are numerous applications of LSI in information retrieval and data mining, including ad hoc text retrieval [6, 3, 20, 14], cross-language retrieval [10], distributed retrieval [23], text categorisation [5], Web search [15, 26], face or object recognition [25, 19], and DNA microarray data analysis [2, 21, 24].

However, the computational complexity of LSI is super-linear (in m and n) [13, 11], which may be prohibitive when $\mathbf{A}$ is very large.

In this paper, we present a fast approximate algorithm for large-scale LSI that is conceptually simple and theoretically justified. The central idea is to perform truncated SVD computation not directly on $\mathbf{A}$ but on its *sketch* sub-matrix $\mathbf{S}$ that consists of the s ($k \leq s \ll m, n$) columns of $\mathbf{A}$ with largest lengths (l^2 norms) $|\cdot|$. We prove that the rank k description $\mathbf{D}^*$ of $\mathbf{A}$ obtained in this way is close to the optimal rank k description of $\mathbf{A}$:

$$\|\mathbf{A}-\mathbf{D}^*\|_F^2 \leq \min_{\substack{\mathbf{D}\in\mathbb{R}^{m\times n} \\ rank(\mathbf{D})\leq k}} \|\mathbf{A}-\mathbf{D}\|_F^2 + 2\sqrt{k}(1-p)\|\mathbf{A}\|_F^2 \ , \tag{1}$$

where $p = \|\mathbf{S}\|_F^2 / \|\mathbf{A}\|_F^2$. Furthermore, we show that the proposed algorithm only requires $O(m+n)$ additional time and space.

The rest of this paper is organised as follows. In Section 2, we review the background knowledge of linear algebra. In Section 3, we describe our fast approximate algorithm for large-scale LSI. In Section 4, we give an analysis on the algorithm's error bound and computational complexity. In Section 5, we present the preliminary experimental results. In Section 6, we discuss related work. In Section 7, we make conclusions.

2 Preliminary

This section contains a brief review of linear algebra [13, 11] that is relevant to our work.

For a vector $\mathbf{x} \in \mathbb{R}^n$, let x_j, $j = 1, \ldots, n$ denote the j-th element of $\mathbf{x}$. The length or l^2 norm of $\mathbf{x}$ is

$$|\mathbf{x}| = \sqrt{\sum_{j=1}^{n} x_j^2} \ . \tag{2}$$

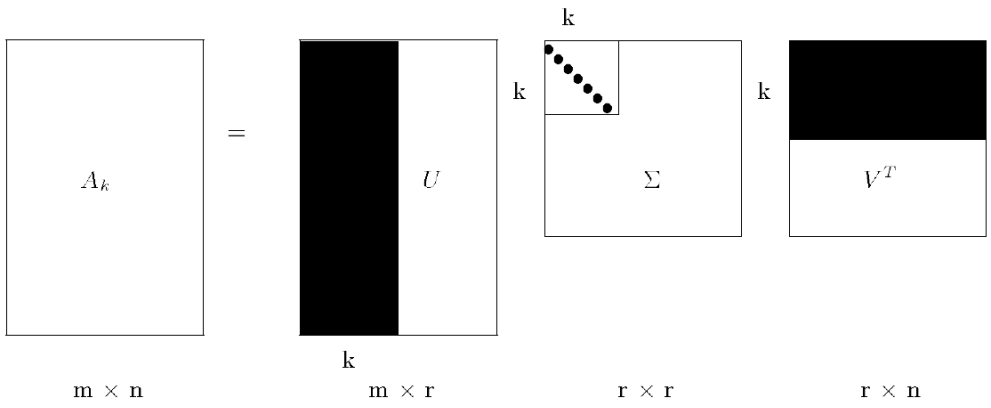

Figure 1. Truncated Singular Valude Decomposition (SVD) [3].

For a matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, let A_{ij} denote the (i,j)-th element of $\mathbf{A}$, and also let $\mathbf{A}^{(i)}$, $i = 1,\ldots,m$ denote the i-th row of $\mathbf{A}$ as a row vector and $\mathbf{A}_{(j)}$, $j = 1,\ldots,n$ denote the j-th column of $\mathbf{A}$ as a column vector. The Frobenius norm of $\mathbf{A}$ that provides a measure of $\mathbf{A}$'s size is defined by

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij}^2} \ . \tag{3}$$

If $\mathrm{Tr}(\mathbf{A})$ represents the matrix trace, i.e., the sum of the diagonal elements, of $\mathbf{A}$, then

$$\|\mathbf{A}\|_F^2 = \mathrm{Tr}\left(\mathbf{A}\mathbf{A}^T\right) = \mathrm{Tr}\left(\mathbf{A}^T\mathbf{A}\right) \ . \tag{4}$$

Furthermore, if $\{\mathbf{x}_1,\ldots,\mathbf{x}_n\}$ constitute a basis of $\mathbb{R}^n$, then

$$\|\mathbf{A}\|_F^2 = \sum_{j=1}^{n} |\mathbf{A}\mathbf{x}_j|^2 \ . \tag{5}$$

The rank of $\mathbf{A}$, $\mathrm{rank}(\mathbf{A})$, is the number of linearly independent columns (or rows) of $\mathbf{A}$.

Given a matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, there exist orthogonal matrices $\mathbf{U} = [\mathbf{u}_1,\ldots,\mathbf{u}_m] \in \mathbb{R}^{m\times m}$ with $\{\mathbf{u}_t\}_{t=1}^m \in \mathbb{R}^m$ and $\mathbf{V} = [\mathbf{v}_1,\ldots,\mathbf{v}_n] \in \mathbb{R}^{n\times n}$ with $\{\mathbf{v}_t\}_{t=1}^n \in \mathbb{R}^n$ such that

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \ , \tag{6}$$

where $\mathbf{\Sigma} = \mathrm{diag}(\sigma_1,\ldots,\sigma_\rho) \in \mathbb{R}^{m\times n}$, $\rho = \min\{m,n\}$ and $\sigma_1 \geq \ldots \sigma_\rho \geq 0$. The three matrices $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{\Sigma}$ constitute the Singular Value Decomposition (SVD) of $\mathbf{A}$. The σ_t are the singular values of $\mathbf{A}$, and the vectors $\mathbf{u}_t$, $\mathbf{v}_t$ are the t-th left and and right singular vectors of $\mathbf{A}$ respectively.

The singular values of $\mathbf{A}$ are the non-negative square roots of the eigenvalues of $\mathbf{A}\mathbf{A}^T$ or $\mathbf{A}^T\mathbf{A}$. The left singular vectors of $\mathbf{A}$ are the eigenvectors of $\mathbf{A}\mathbf{A}^T$ and the right singular vectors of $\mathbf{A}$ are the eigenvectors of $\mathbf{A}^T\mathbf{A}$.

The number of positive singular values $r = \mathrm{rank}(\mathbf{A})$, so $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > \sigma_{r+1} = \ldots = \sigma_\rho = 0$. If $\mathbf{U}_r \in \mathbb{R}^{m\times r}$ denotes the matrix consist of the first r columns of $\mathbf{U}$, $\mathbf{V}_r \in \mathbb{R}^{r\times n}$ denotes the matrix consist of the first r columns of $\mathbf{V}$, and $\mathbf{\Sigma}_r \in \mathbb{R}^{r\times r}$ denotes the principal $r \times r$ sub-matrix of $\mathbf{\Sigma}$, then

$$\mathbf{A} = \mathbf{U}_r\mathbf{\Sigma}_r\mathbf{V}_r^T = \sum_{t=1}^{r} \sigma_t \mathbf{u}_t\mathbf{v}_t^T \ . \tag{7}$$

For $1 \leq k < r$, the *truncated* SVD of $\mathbf{A}$ (see Figure 1) is given by

$$\mathbf{A}_k = \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^T = \sum_{t=1}^{k} \sigma_t \mathbf{u}_t\mathbf{v}_t^T \ . \tag{8}$$

The matrix $\mathbf{A}_k \in \mathbb{R}^{m\times n}$ is the projection of $\mathbf{A}$ on to the space spanned by the top k singular vectors of $\mathbf{A}$, i.e.,

$$\mathbf{A}_k = \mathbf{U}_k\mathbf{U}_k^T\mathbf{A} = \left(\sum_{t=1}^{k} \mathbf{u}_t\mathbf{u}_t^T\right)\mathbf{A} \ , \tag{9}$$

and

$$\mathbf{A}_k = \mathbf{A}\mathbf{V}_k\mathbf{V}_k^T = \mathbf{A}\left(\sum_{t=1}^{k} \mathbf{v}_t\mathbf{v}_t^T\right) \ . \tag{10}$$

Furthermore, the distance (as measured by $\|\cdot\|_F$) between $\mathbf{A}$ and any rank k approximation to $\mathbf{A}$ is minimised by $\mathbf{A}_k$, i.e.,

$$\begin{aligned} \min_{\substack{\mathbf{D}\in\mathbb{R}^{m\times n} \\ rank(\mathbf{D})\leq k}} \|\mathbf{A}-\mathbf{D}\|_F^2 &= \|\mathbf{A}-\mathbf{A}_k\|_F^2 \\ &= \sum_{t=k+1}^{r} \sigma_t^2(\mathbf{A}) \ . \end{aligned} \tag{11}$$

In other words, the matrix $\mathbf{A}_k$ constructed from the k largest singular triplets of $\mathbf{A}$ is the optimal rank k approximation to $\mathbf{A}$ with respect to $\|\cdot\|_F$. It can be shown that

$$\|\mathbf{A}\|_F^2 = \sum_{t=1}^{r} \sigma_t^2(\mathbf{A}) \ . \tag{12}$$

According to the matrix perturbation theory [22], the size of the difference between two matrices can be used to bound the difference between the singular value spectrum of the two matrices. In particular, the Hoffman-Wielandt inequality states that if $\mathbf{A}, \mathbf{E} \in \mathbb{R}^{m \times n}$, $m \geq n$, then

$$\sum_{t=1}^{n} \left(\sigma_t(\mathbf{A}+\mathbf{E}) - \sigma_t(\mathbf{A})\right)^2 \leq \|\mathbf{E}\|_F^2 \ . \tag{13}$$

3 Algorithm

Consider a very large data matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, and without loss of generality assume that $m \geq n$. The LSI of $\mathbf{A}$ can be considered as an optimization process that finds $\mathbf{A}_k$, i.e., the optimal rank k approximation to $\mathbf{A}$ in terms of the Frobenius norm $\|\cdot\|_F$ of their difference. Since the computational complexity of this process depends on the dimension of $\mathbf{A}$, our idea for accelerating the process is to divide it into two stages:

1. reduce the high dimensional matrix $\mathbf{A}$ to a low dimensional matrix $\mathbf{S} \in \mathbb{R}^{m \times s}$ that is close to $\mathbf{A}$;
2. perform the truncated SVD of $\mathbf{S}$ that provides approximations to the singular values and singular vectors of the original matrix $\mathbf{A}$.

A simple method to construct the low dimensional matrix $\mathbf{S}$ is to pick $s \ll n$ columns from $\mathbf{A}$. Since

$$\begin{aligned} \|\mathbf{A}\|_F^2 &= \sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij}^2 \\ &= \sum_{j=1}^{n}\left(\sum_{i=1}^{m} A_{ij}^2\right) \\ &= \sum_{j=1}^{n} |\mathbf{A}_{(j)}|^2 \ , \end{aligned} \tag{14}$$

the best strategy for minimizing the Frobenius norm loss is obviously to select the s columns with largest lengths $|\cdot|$. Furthermore, we can get the truncated SVD of $\mathbf{S}$ through performing eigen-decomposition of the symmetric matrix $\mathbf{S}^T\mathbf{S} \in \mathbb{R}^{s \times s}$ which can be computed efficiently if s is small.

Our fast approximate algorithm for large-scale LSI is presented in Figure 2. Given a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, the algorithm returns the approximations to $\mathbf{A}$'s left singular vectors $\widehat{\mathbf{U}}_k = [\hat{\mathbf{u}}_1, \ldots, \hat{\mathbf{u}}_k]$ as well as the approximations to $\mathbf{A}$'s singular values $\widehat{\mathbf{\Sigma}}_k = \mathrm{diag}\,(\hat{\sigma}_1, \ldots, \hat{\sigma}_k)$. Then a rank k approximation to $\mathbf{A}$ can be given by

$$\mathbf{D}^* = \widehat{\mathbf{U}}_k \widehat{\mathbf{U}}_k^T \mathbf{A} \ , \tag{15}$$

which (as we will show later in Section 4) is close to $\mathbf{A}_k$ the optimal rank k approximation to $\mathbf{A}$.

4 Analysis

4.1 Error Bound

The matrix $\mathbf{D}^* = \widehat{\mathbf{U}}_k \widehat{\mathbf{U}}_k^T \mathbf{A}$ is a good (though not optimal) approximation to the original matrix $\mathbf{A}$, in the sense that the incurred error $\|\mathbf{A} - \mathbf{D}^*\|_F^2$ is bounded by the smallest possible error $\min_{\mathbf{D} \in \mathbb{R}^{m \times n}; rank(\mathbf{D}) \leq k} \|\mathbf{A} - \mathbf{D}\|_F^2$ plus an additional error term depending on a portion of $\|\mathbf{A}\|_F^2$.

Theorem 1. *If using the fast approximate algorithm for large-scale LSI (as described in Figure 2) on $\mathbf{A} \in \mathbb{R}^{m \times n}$ we get the result $\widehat{\mathbf{U}}_k$, then $\mathbf{D}^* = \widehat{\mathbf{U}}_k \widehat{\mathbf{U}}_k^T \mathbf{A}$ satisfies*

$$\|\mathbf{A} - \mathbf{D}^*\|_F^2 \leq \min_{\substack{\mathbf{D} \in \mathbb{R}^{m \times n} \\ rank(\mathbf{D}) \leq k}} \|\mathbf{A} - \mathbf{D}\|_F^2 + 2\sqrt{k}(1-p)\|\mathbf{A}\|_F^2 \ , \tag{16}$$

where $p = \|\mathbf{S}\|_F^2 / \|\mathbf{A}\|_F^2$.

Proof. Using the rules that $\|\mathbf{X}\|_F^2 = \mathrm{Tr}\left(\mathbf{X}^T\mathbf{X}\right)$ and $\widehat{\mathbf{U}}_k^T \widehat{\mathbf{U}}_k = \mathbf{I} \in \mathbb{R}^{k \times k}$, we get

$$\begin{aligned} & \|\mathbf{A} - \mathbf{D}^*\|_F^2 \\ = \ & \|\mathbf{A} - \widehat{\mathbf{U}}_k \widehat{\mathbf{U}}_k^T \mathbf{A}\|_F^2 \\ = \ & \mathrm{Tr}\left(\left(\mathbf{A} - \widehat{\mathbf{U}}_k \widehat{\mathbf{U}}_k^T \mathbf{A}\right)^T \left(\mathbf{A} - \widehat{\mathbf{U}}_k \widehat{\mathbf{U}}_k^T \mathbf{A}\right)\right) \\ = \ & \mathrm{Tr}\left(\mathbf{A}^T\mathbf{A} - 2\mathbf{A}^T \widehat{\mathbf{U}}_k \widehat{\mathbf{U}}_k^T \mathbf{A} + \mathbf{A}^T \widehat{\mathbf{U}}_k \widehat{\mathbf{U}}_k^T \widehat{\mathbf{U}}_k \widehat{\mathbf{U}}_k^T \mathbf{A}\right) \\ = \ & \mathrm{Tr}\left(\mathbf{A}^T\mathbf{A} - \mathbf{A}^T \widehat{\mathbf{U}}_k \widehat{\mathbf{U}}_k^T \mathbf{A}\right) \\ = \ & \mathrm{Tr}\left(\mathbf{A}^T\mathbf{A}\right) - \mathrm{Tr}\left(\mathbf{A}^T \widehat{\mathbf{U}}_k \widehat{\mathbf{U}}_k^T \mathbf{A}\right) \\ = \ & \|\mathbf{A}\|_F^2 - \|\mathbf{A}^T \widehat{\mathbf{U}}_k\|_F^2 \ . \end{aligned} \tag{17}$$

Applying the Cauchy-Schwartz inequality and noting that $\hat{\mathbf{u}}_1, \ldots, \hat{\mathbf{u}}_k$ provides a basis of $\mathbb{R}^k$, we get

$$\begin{aligned} & \left| \|\mathbf{A}^T \widehat{\mathbf{U}}_k\|_F^2 - \sum_{t=1}^{k} \hat{\sigma}_t^2 \right| \\ = \ & \left| \sum_{t=1}^{k} |\mathbf{A}^T \hat{\mathbf{u}}_t|^2 - \sum_{t=1}^{k} \hat{\sigma}_t^2 \right| \\ = \ & \left| \sum_{t=1}^{k} 1 \cdot \left(|\mathbf{A}^T \hat{\mathbf{u}}_t|^2 - \hat{\sigma}_t^2\right) \right| \\ \leq \ & \sqrt{k} \sqrt{\sum_{t=1}^{k} \left(|\mathbf{A}^T \hat{\mathbf{u}}_t|^2 - \hat{\sigma}_t^2\right)^2} \\ = \ & \sqrt{k} \sqrt{\sum_{t=1}^{k} \left(|\mathbf{A}^T \hat{\mathbf{u}}_t|^2 - |\mathbf{S}^T \hat{\mathbf{u}}_t|^2\right)^2} \\ = \ & \sqrt{k} \sqrt{\sum_{t=1}^{k} \left(\hat{\mathbf{u}}_t^T (\mathbf{A}\mathbf{A}^T - \mathbf{S}\mathbf{S}^T) \hat{\mathbf{u}}_t\right)^2} \end{aligned}$$

Input: $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $k, s \in \mathbb{Z}^+$ s.t. $1 \leq k \leq s \leq n$.

Output: $\widehat{\mathbf{U}}_k$ and $\widehat{\mathbf{\Sigma}}_k$.

- Calculate $|\mathbf{A}_{(j)}| = \sqrt{\sum_{i=1}^{m} A_{ij}^2}$ for $j = 1, \ldots, n$.
- Construct $\mathbf{S} = \left[\mathbf{A}_{(j_1)}, \ldots, \mathbf{A}_{(j_s)}\right]$ where $\mathbf{A}_{(j_t)}$, $t = 1, \ldots, s$ are the s columns of $\mathbf{A}$ with largest lengths $|\mathbf{A}_{(j_t)}|$.
- Compute $\mathbf{S}^T\mathbf{S}$.
- Perform the eigen-decomposition of $\mathbf{S}^T\mathbf{S}$, i.e., $\mathbf{S}^T\mathbf{S} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$ where $\mathbf{Q} = [\mathbf{q}_1, \ldots, \mathbf{q}_s]$ and $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_s)$ with $\lambda_1 \geq \ldots \lambda_s \geq 0$.
- Compute $\hat{\sigma}_t = \sqrt{\lambda_t}$ for $t = 1, \ldots, k$.
- Compute $\hat{\mathbf{u}}_t = \mathbf{S}\mathbf{q}_t / \hat{\sigma}_t$ for $t = 1, \ldots, k$.
- Return $\widehat{\mathbf{U}}_k = [\hat{\mathbf{u}}_1, \ldots, \hat{\mathbf{u}}_k]$ and $\widehat{\mathbf{\Sigma}}_k = \mathrm{diag}(\hat{\sigma}_1, \ldots, \hat{\sigma}_k)$.

Figure 2. Our fast approximate algorithm for large-scale LSI.

$$\leq \sqrt{k}\|\mathbf{A}\mathbf{A}^T - \mathbf{S}\mathbf{S}^T\|_F \ . \tag{18}$$

Applying the Cauchy-Schwartz inequality again and in addition the Hoffman-Wielandt inequality, we get

$$\begin{aligned}
&\left|\sum_{t=1}^{k} \sigma_t^2 - \sum_{t=1}^{k} \hat{\sigma}_t^2\right| \\
= &\left|\sum_{t=1}^{k} 1 \cdot \left(\sigma_t^2 - \hat{\sigma}_t^2\right)\right| \\
\leq &\sqrt{k}\sqrt{\sum_{t=1}^{k} \left(\sigma_t^2 - \hat{\sigma}_t^2\right)^2} \\
\leq &\sqrt{k}\sqrt{\sum_{t=1}^{m} \left(\sigma_t^2 - \hat{\sigma}_t^2\right)^2} \\
= &\sqrt{k}\sqrt{\sum_{t=1}^{m} \left(\sigma_t(\mathbf{A}\mathbf{A}^T) - \sigma_t(\mathbf{S}\mathbf{S}^T)\right)^2} \\
\leq &\sqrt{k}\|\mathbf{A}\mathbf{A}^T - \mathbf{S}\mathbf{S}^T\|_F \ .
\end{aligned} \tag{19}$$

Applying the triangle inequality to combine the above two inequalities, we get

$$\begin{aligned}
&\left|\sum_{t=1}^{k} \sigma_t^2 - \|\mathbf{A}^T\widehat{\mathbf{U}}_k\|_F^2\right| \\
= &\left|\left(\sum_{t=1}^{k} \sigma_t^2 - \sum_{t=1}^{k} \hat{\sigma}_t^2\right) + \left(\sum_{t=1}^{k} \hat{\sigma}_t^2 - \|\mathbf{A}^T\widehat{\mathbf{U}}_k\|_F^2\right)\right| \\
\leq &\left|\sum_{t=1}^{k} \sigma_t^2 - \sum_{t=1}^{k} \hat{\sigma}_t^2\right| + \left|\sum_{t=1}^{k} \hat{\sigma}_t^2 - \|\mathbf{A}^T\widehat{\mathbf{U}}_k\|_F^2\right| \\
= &\left|\sum_{t=1}^{k} \sigma_t^2 - \sum_{t=1}^{k} \hat{\sigma}_t^2\right| + \left|\|\mathbf{A}^T\widehat{\mathbf{U}}_k\|_F^2 - \sum_{t=1}^{k} \hat{\sigma}_t^2\right| \\
\leq &\ 2\sqrt{k}\|\mathbf{A}\mathbf{A}^T - \mathbf{S}\mathbf{S}^T\|_F \ .
\end{aligned} \tag{20}$$

Without loss of generality, we can assume that $\mathbf{A}_{(j)}$, $j = 1, \ldots, s$ are the largest columns of $\mathbf{A}$, i.e., $\mathbf{S} = \left[\mathbf{A}_{(1)}, \ldots, \mathbf{A}_{(s)}\right]$. Then applying the triangle inequality again, we get

$$\begin{aligned}
&\|\mathbf{A}\mathbf{A}^T - \mathbf{S}\mathbf{S}^T\|_F \\
= &\left\|\sum_{t=1}^{n} \mathbf{A}_{(j)}\mathbf{A}_{(j)}^T - \sum_{t=1}^{s} \mathbf{A}_{(j)}\mathbf{A}_{(j)}^T\right\|_F \\
= &\left\|\sum_{t=s+1}^{n} \mathbf{A}_{(j)}\mathbf{A}_{(j)}^T\right\|_F \\
\leq &\sum_{t=s+1}^{n} \left\|\mathbf{A}_{(j)}\mathbf{A}_{(j)}^T\right\|_F \\
= &\sum_{t=s+1}^{n} \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{m} \left(A_{it}A_{jt}\right)^2} \\
= &\sum_{t=s+1}^{n} \sqrt{\left(\sum_{i=1}^{m} A_{it}^2\right)\left(\sum_{j=1}^{m} A_{jt}^2\right)}
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{t=s+1}^{n}\sum_{i=1}^{m} A_{it}^2 \\
&= \sum_{i=1}^{m}\sum_{j=s+1}^{n} A_{ij}^2 \\
&= \sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij}^2 - \sum_{i=1}^{m}\sum_{j=1}^{s} A_{ij}^2 \\
&= \|\mathbf{A}\|_F^2 - \|\mathbf{S}\|_F^2 \\
&= (1-p)\|\mathbf{A}\|_F^2 \ , \qquad (21)
\end{aligned}
$$

Finally, assembling all the above inequalities yields the theorem,

$$
\begin{aligned}
& \|\mathbf{A} - \mathbf{D}^*\|_F^2 \\
= \ & \|\mathbf{A}\|_F^2 - \|\mathbf{A}^T\widehat{\mathbf{U}}_k\|_F^2 \\
= \ & \left(\|\mathbf{A}\|_F^2 - \sum_{t=1}^{k}\sigma_t^2\right) + \left(\sum_{t=1}^{k}\sigma_t^2 - \|\mathbf{A}^T\widehat{\mathbf{U}}_k\|_F^2\right) \\
= \ & \left(\sum_{t=1}^{r}\sigma_t^2 - \sum_{t=1}^{k}\sigma_t^2\right) + \left(\sum_{t=1}^{k}\sigma_t^2 - \|\mathbf{A}^T\widehat{\mathbf{U}}_k\|_F^2\right) \\
= \ & \left(\sum_{t=k+1}^{r}\sigma_t^2\right) + \left(\sum_{t=1}^{k}\sigma_t^2 - \|\mathbf{A}^T\widehat{\mathbf{U}}_k\|_F^2\right) \\
= \ & \|\mathbf{A} - \mathbf{A}_k\|_F^2 + \left(\sum_{t=1}^{k}\sigma_t^2 - \|\mathbf{A}^T\widehat{\mathbf{U}}_k\|_F^2\right) \\
\leq \ & \|\mathbf{A} - \mathbf{A}_k\|_F^2 + \left|\sum_{t=1}^{k}\sigma_t^2 - \|\mathbf{A}^T\widehat{\mathbf{U}}_k\|_F^2\right| \\
\leq \ & \|\mathbf{A} - \mathbf{A}_k\|_F^2 + 2\sqrt{k}\|\mathbf{A}\mathbf{A}^T - \mathbf{S}\mathbf{S}^T\|_F \\
\leq \ & \|\mathbf{A} - \mathbf{A}_k\|_F^2 + 2\sqrt{k}(1-p)\|\mathbf{A}\|_F^2 \\
= \ & \min_{\substack{\mathbf{D}\in\mathbb{R}^{m\times n} \\ rank(\mathbf{D})\leq k}} \|\mathbf{A} - \mathbf{D}\|_F^2 + 2\sqrt{k}(1-p)\|\mathbf{A}\|_F^2 \ . \qquad (22)
\end{aligned}
$$

□ □

The above theorem also justifies our intuition that the best strategy to construct the sketch matrix $\mathbf{S}$ is to select the largest columns from $\mathbf{A}$ thus the proportion $p = \|\mathbf{S}\|_F^2/\|\mathbf{A}\|_F^2$ is large and consequently the error bound is small.

In practice, it could be more convenient to determine the value of parameter s based on a pre-fixed threshold of p rather than the other way around.

4.2 Computational Complexity

Let's examine our fast approximate algorithm for large-scale LSI (as shown in Figure 2) step by step.

Calculating the lengths of the columns of $\mathbf{A}$, i.e., $|\mathbf{A}_{(j)}| = \sqrt{\sum_{i=1}^{m} A_{ij}^2}$ for $j = 1, \ldots, n$, can be done in one pass over $\mathbf{A}$ and requires only $O(n)$ additional time and space. Picking the s largest columns from the n columns of $\mathbf{A}$ can be done using *selection algorithms* [4] which typically have $O(n + s\log s)$ computational complexity, but as we do not need those largest s columns $\mathbf{A}_{(j_1)}, \ldots, \mathbf{A}_{(j_s)}$ to be themselves ordered, the complexity can be further reduced to $O(n)$.

Then the construction of $\mathbf{S}$ takes $O(ms)$ additional time and space, and to get $\mathbf{S}^T\mathbf{S}$ we need $O(ms^2)$ additional time and space. Furthermore, the eigen-decomposition of the $s \times s$ matrix $\mathbf{S}^T\mathbf{S}$ takes $O(s^3)$ additional time and space [13, 11]. Moreover, computing $\hat{\sigma}_t = \sqrt{\lambda_t}$ for $t = 1, \ldots, k$ requires $O(k)$ additional time and space; computing $\hat{\mathbf{u}}_t = \mathbf{S}\mathbf{q}_t/\hat{\sigma}_t$ for $t = 1, \ldots, k$ requires $O(msk)$ additional time and space.

In summary, since s and k are constants, the overall computational complexity of our algorithm is only $O(m + n)$.

5 Experiments

We have implemented our algorithm in Matlab, and performed preliminary experiments on a real-world text dataset 20-newsgroups[1] [16]. There are totally 19928 documents and 62061 terms, therefore the document-term matrix $\mathbf{A}$ has $m = 19928$ rows and $n = 62061$ columns. Assuming the number of latent semantic concepts in this corpus to be 20, we use truncated SVD with $k = 20$ of $\mathbf{A}$ for LSI. The experimental observation is that selecting the $s = n/10$ columns with largest lengths, our fast approximate algorithm is an order of magnitude faster than the original LSI algorithm, while keeping the approximation error small as $1 - p = 0.1002$. In practice, we note that the fast approximate algorithm often performs much better than the theoretical error bound suggests.

6 Related Work

The exact solutions of truncated SVD are typically computed using iterative algorithms like the Lanczos method [18], but the computational complexity of such algorithms is too high to be practical on very large datasets.

Gorrell proposed an incremental algorithm for approximate truncated SVD which works in a neural-network-like fashion and requires much less resources [12]. Tang et al. proposed to reduce the cost of truncated SVD through document clustering and term selection [23]. However, those approximate algorithms do not come with a theoretical guarantee of error bounds.

Drineas et al. proposed a randomised algorithms for approximate truncated SVD which also reduces the computation on the given large matrix $\mathbf{A}$ to that on a small sketch

[1] http://www.csie.ntu.edu.tw/∼cjlin/libsvmtools/datasets/multiclass.html #news20

matrix consisting of randomly sampled columns from $\mathbf{A}$ according to a certain probability distribution [7, 8, 9], but their used sketch matrix, unlike ours, contains many duplicate columns and consequently impairs the algorithm's time and space efficiencies. Achlioptas and McSherry proposed an alternative entry-wise randomised algorithm for approximate truncated SVD based on the theory of random matrices [1].

7 Conclusions

We have presented a fast approximate algorithm for large-scale LSI that is conceptually simple and theoretically justified. Our main contribution is to show that the proposed algorithm has provable error bound and linear computational complexity.

We plan to conduct more experiments on real-world datasets in the future so as to empirically evaluate the effectiveness and efficiency of our proposed algorithm.

8 Acknowledgements

We thank the anonymous reviewers for their helpful comments.

References

[1] D. Achlioptas and F. McSherry. Fast computation of low-rank matrix approximations. *Journal of the ACM*, 54(2):Article 9, 2007.

[2] O. Alter, P. O. Brown, and D. Botstein. Singular value decomposition for genome-wide expression data processing and modeling. *Proceedings of the National Academy of Sciences (PNAS)*, 97(18):10101–10106, 2000.

[3] M. W. Berry, S. T. Dumais, and G. W. O'Brien. Using linear algebra for intelligent information retrieval. *SIAM Review*, 37(4):573–595, 1995.

[4] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press and McGraw-Hill, 2nd edition, 2001.

[5] N. Cristianini, J. Shawe-Taylor, and H. Lodhi. Latent semantic kernels. In *Proceedings of the 18th International Conference on Machine Learning (ICML)*, pages 66–73, Williamstown, MA, 2001.

[6] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science (JASIS)*, 41(6):391–407, 1990.

[7] P. Drineas, R. Kannan, and M. W. Mahoney. Fast monte carlo algorithms for matrices i: Approximating matrix multiplication. *SIAM Journal on Computing*, 36(1):132–157, 2006.

[8] P. Drineas, R. Kannan, and M. W. Mahoney. Fast monte carlo algorithms for matrices ii: Computing a low-rank approximation to a matrix. *SIAM Journal on Computing*, 36(1):158–183, 2006.

[9] P. Drineas, R. Kannan, and M. W. Mahoney. Fast monte carlo algorithms for matrices iii: Computing a compressed approximate matrix decomposition. *SIAM Journal on Computing*, 36(1):184–206, 2006.

[10] S. T. Dumais, T. A. Letsche, M. L. Littman, and T. K. Landauer. Automatic cross-linguistic information retrieval using latent semantic indexing. In *AAAI Symposium on CrossLanguage Text and Speech Retrieval. American Association for Artificial Intelligence*, pages 15–21, Palo Alto, CA, 1997.

[11] G. H. Golub and C. F. V. Loan. *Matrix Computations*. The Johns Hopkins University Press, 3rd edition, 1996.

[12] G. Gorrell. Generalized hebbian algorithm for incremental singular value decomposition in natural language processing. In *Proceedings of the 11st Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 97–104, Trento, Italy, 2006.

[13] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1990.

[14] P. Husbands, H. Simon, and C. H. Q. Ding. On the use of the singular value decomposition for text retrieval. In *Computational Information Retrieval*, pages 145–156, 2001.

[15] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[16] K. Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the 12th International Conference on Machine Learning (ICML)*, pages 331–339, Tahoe City, CA, 1995.

[17] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[18] C. D. Meyer. *Matrix Analysis and Applied Linear Algebra*. Society for Industrial and Applied Mathematics, Philadelphia, 2000.

[19] H. Murase and S. K. Nayar. Visual learning and recognition of 3-d objects from appearance. *International Journal of Computer Vision (IJCV)*, 14(1):5–24, 1995.

[20] C. H. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala. Latent semantic indexing: A probabilistic analysis. *Journal of Computer and System Sciences (JCSS)*, 61(2):217–235, 2000.

[21] S. Raychaudhuri, J. M. Stuart, and R. B. Altman. Principal components analysis to summarize microarray experiments: Application to sporulation time series. In *Proceedings of the 5th Pacific Symposium on Biocomputing (PSB)*, pages 452–463, Hawaii, 2000.

[22] G. W. Stewart and J.-G. Sun. *Matrix Perturbation Theory*. Academic Press, 1990.

[23] C. Tang, S. Dwarkadas, and Z. Xu. On scaling latent semantic indexing for large peer-to-peer systems. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 112–121, Sheffield, UK, 2004.

[24] O. G. Troyanskaya, M. Cantor, G. Sherlock, P. O. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman. Missing value estimation methods for dna microarrays. *Bioinformatics*, 17(6):520–525, 2001.

[25] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[26] D. Zhang and Y. Dong. Semantic, hierarchical, online clustering of web search results. In *Proceedings of the 6th Asia-Pacific Web Conference (APWeb)*, pages 69–78, Hangzhou, China, 2004.