Principle Components Updates via Matrix Perturbations

Aiman Elragig, Hanan Dreiwi, Dung Ly, Idriss Elmabrook

Abstract—This paper highlights a new approach to look at online principle components analysis (OPCA). Given a data matrix $X \in \mathbb{R}^{m \times n}$ we characterise the online updates of its covariance as a matrix perturbation problem. Up to the principle components, it turns out that online updates of the batch PCA can be captured by symmetric matrix perturbation of the batch covariance matrix. We have shown that as $n \to n_0 \gg 1$, the batch covariance and its update become almost similar. Finally, utilize our new setup of online updates to find a bound on the angle distance of the principle components of $X$ and its update.

Keywords—Online data updates, covariance matrix, online principle component analysis (OPCA), matrix perturbation.

I. INTRODUCTION

Online learning has become an urge in many applications [5], [6]. As opposed to batch settings, in online learning we do not have an access to the whole data we are intended to learn from. Instead, we receive data points sequentially, one by one, and the ultimate goal is to determine an adaptive model capturing the overall trend of the whole data.

Many researchers have dedicated much of their work to develop online algorithms, using different approaches, to cope with such sequential updates of data [9]. However, most of the algorithms developed in the context of online learning are essentially based on tweaking the already known models for batch data, i.e., they become able to learn from data on-the-fly [7], [8], [18]–[20]. Despite of all evident successes in developing online models [1], [12], there is, however, a need to a rigorous theory which can be used to reveal some aspects, the exiting techniques fails to clarify.

One of the most important online learning problems, which has received tremendous amounts of investigation, is the online principle components analysis (OPCA) [2], [10]–[12]. Roughly speaking, PCA aims to extract the main modes of variation of the data around their mean through computing new variables called the principle components [9]. PCA has found applications in various fields such as face recognition [12], latent semantic indexing [16], sentiment analysis [13], industrial process modeling [14], astronomy [15], to name but few. Expensive computation is one of the main major challenges in using PCA technique as it often requires approximation which resulting in prediction errors.

This paper is a step forward to better understand online PCA update mechanism providing a new view to look at online updates of data. This paper is organised as follows. Section II will give some required mathematical concepts. Section III presents a detailed description of PCA as a technique with some exploration of the current related literatures. Section IV shows a formulation of online updates of data as a symmetric matrix perturbation. Section V provides the main result of the paper. In Section VI we use some matrix perturbation results to illustrate the significance of our setup.

II. MATHEMATICAL PRELIMINARIES

In this section we will briefly some basic mathematical concepts. For a vector $v \in \mathbb{R}^n$, its Euclidean norm is given by $||v|| := \sqrt{v^T v}$. For a matrix $A \in \mathbb{R}^{m \times n}$, the he spectral norm of $A$ is defined as $||A|| = \max\{||Av|| : ||v|| = 1\}$. The transpose of $A$ is another matrix obtained by exchanging its row with its columns and its often denoted by $A^T$. The matrix $A$ is said to be symmetric if, and only if $A = A^T$. The singular value decomposition of the matrix $A$ is given by $A = U \Sigma V^T$, where $U$ is an $m \times m$ orthonormal matrix, $\Sigma$ is diagonal matrix, and $V$ is an $n \times n$ orthonormal matrix. The diagonal entries of the diagonal matrix $\Sigma$, $\sigma_1, \sigma_2, \ldots, \sigma_n$, are known as the singular values of $A$. For a matrix $S \in \mathbb{R}^{m \times n}$, the scalar $\lambda$ is an eigenvalue of $S$ if there exists a non-zero vector $v$ such that $Av = \lambda v$. The vector $v$ is referred to as the eigenvector of $S$ corresponding to the eigenvalue $\lambda$. If $S$ is symmetric we can write $S = U \Lambda U^T$, where $U$ is an $m \times m$ orthonormal matrix and $\Lambda$ is a diagonal matrix whose entries are called the eigenvalues of the matrix $S$. Conventionally we often list the eigenvalues of $S$ in a non-increasing order namely, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$. It is worth to add here that $||S|| = \sigma_1 = \lambda_1$.

III. PRINCIPLE COMPONENT ANALYSIS

Principle components analysis (PCA) is a very common method for reducing the dimensionality of data [23]–[26]. To have an idea about PCA we first discuss a related
concept namely, the covariance matrix. Given the data set of \( m \)-dimensional data points \( x_i \), we can defined the matrix

\[
X = \begin{pmatrix}
x_1 \\
x_2 \\
\vdots \\
x_m
\end{pmatrix}.
\]

Lets assume that each columns in the matrix \( X \) corresponds to a data point, whereas each row refers to all measurement of a particular value \( x_i \). Now the covariance matrix of the data \( X \) can be defined as follows:

\[
\text{Cov}(X) = \frac{1}{n-1}XX^T.
\]

It's easy to see the following properties of \( \text{Cov}(X) \):

- The main diagonal entries of \( \text{Cov}(X) \) show the variance of particular measurement features.
- The off-diagonal entries of \( \text{Cov}(X) \) represent the covariance between the measurement types.
- \( \text{Cov}(X) \) is square symmetric \( m \times m \) matrix

The matrix \( \text{Cov}(X) \) gives all correlations between all pairs of data in \( X \). The principle components of the data \( X \) are the eigenvectors of the covariance matrix \( \text{Cov}(X) \). In other words they are the singular vectors of the matrix \( X \).

Often, PCA is performed in batch mode in which all the training data are ready for processing (Computing PCA). In the batch setting there will be no information to be used in learning once all the whole available data are trained [27]. In contrast, in online learning data arrive sequentially and in every time instance we experience new data and that require an adaptive PCA algorithm to cope with such setting.

Typically PCA is implemented by using either the eigenvalue decomposition of the covariance matrix \( \text{Cov}(X) \) or the SVD of the data matrix \( X \), after centering the data.

In dimensionality reduction, for instance, we require to project the data into a low dimensional subspace and this can be easily handled using the SVD of the data matrix \( X \). The dimension of the choose subspace, say \( k \), should be small enough so that it reduces the data significantly and in the same time it should be large enough to retain the main variation characteristics of the data. In mathematical terms the best rank-\( k \) \((X_k)\) representation of the matrix \( X \) is given by

\[
X_k = U_k \Sigma_k V_k^T = \sum_{i=1}^k \sigma_i u_i v_i^T
\]

Updating the above decomposition as new data arrived is computationally expensive given the exponential growth of data in modern applications. Hence efficient (i.e., accurate and fast) PCA algorithms are in high demand. Over the years a large number of solutions has emerged from fields. A numerical resolution to what is so-called secular equations is one of the elegant cinoutation of PCA computations [28–30]. In this approach it has been shown that updating the PCA is equivalent of finding the zeros of a rational function. In [31], PCA updates has been formulated as a sum of the eigenvalue decomposition of a diagonal matrix and a rank-1 matrix. Incremental SVD is one of the most significant approach for having an approximate low rank PCA [32]. Incremental PCA (IPCA) has been around for more than a decades and have shown efficient performance in various applications [2], [32], [34]. The candid covariance-free incremental PCA (CCPCA) developed in [2] is one of the most recent and effective PCA algorithm which can compute the principle components of a stream of samples incrementally without estimating the covariance matrix. For a detailed literature on online pca we refer the reader to [2], [9].

IV. DATA STREAMS AS PERTURBATIONS

Let’s assume that \( X \) be a data matrix with \( m \) rows and \( n \) columns. Also assume that \( X \) is formed by stacking data points column by column. In online setting, \( X \) will be updated with new data points, i.e., new columns. So, we assume that \( X \) is updated with the data point (vector) \( x_{m+1} \). Accordingly, we have the following new data matrix

\[
\hat{X} = \left( X_{m \times n} \quad x_{m+1} \right)_{m \times (n+1)}
\]

Now the covariance matrix of the new data matrix can be obtained as follows

\[
\frac{1}{n} \hat{X} \hat{X}^T = \frac{1}{n} \left( X \quad x \right) \left( X \quad x \right)^T = \frac{n-1}{n} XX^T + \frac{1}{n} xx^T.
\]

In other words we can write

\[
\text{Cov}(\hat{X}) = \frac{n-1}{n} \text{Cov}(X) + \frac{1}{n} xx^T.
\]

Looking at Equation (1), it is straightforward to see that

- The matrices \( \frac{n-1}{n} \text{Cov}(X) \) and \( \text{Cov}(X) \) share the same eigenvectors.
- if \( \lambda \) is an eigenvalue of \( \text{Cov}(X) \) then, \( \frac{n-1}{n} \lambda \) is an eigenvalue of \( \text{Cov}(X) \).

Moreover,

\[
\lim_{n \to \infty} \sigma(\text{Cov}(\hat{X})) = \sigma(\text{Cov}(X)),
\]

where \( \sigma \) stands for a matrix spectrum.

- \( \frac{1}{n} xx^T \) is symmetric matrix with a norm equals to \( \frac{||x||^2}{n} \)

To make a concrete insight for Equation (1), we consider the case when \( m = 2 \), i.e., a two dimensional feature space. Figure 1 depicts Equation (1) through its action on an arbitrary vector \( v \) (which can be an eigenvector of \( \text{Cov}(X) \)).

It's clear that the eigendata of \( \text{Cov}(X) \) eventually converges to the ones of \( \text{Cov}(X) \) as we are exposed to a sufficient amount of the data stream. The word “sufficient” suggests imposing certain threshold on the number of data above which new data arrivals
Given a data matrix \( X \in \mathbb{R}^{n \times m} \) and a date point \( x \) then

1. Up to the principle components, the online updates of the batch data \( \text{Cov}(X) \) with the data stream point \( x \) can be viewed as

\[
\text{Cov}(\bar{X}) = \text{Cov}(X) + \Delta, \quad \text{where} \quad \Delta = \frac{1}{n}xx^T . \tag{2}
\]

2. We have

\[
l_n \xrightarrow[n \rightarrow n_0 \gg 1]{} \text{Cov}(\bar{X}) = \text{Cov}(X). \tag{2}
\]

VI. SOME APPLICATIONS OF PROPOSITION V

In this section we make use of some results from symmetric perturbation theory to characterise some bounds on the inclination of the updated first principle component of \( \text{Cov}(\bar{X}) \), and its corresponding eigenvalue. The analysis is identical for other principle components.

1. Bound on the Eigenvalues

Following Weyl Theorem (See, Appendix B), and considering equation (1) we have

\[
\lambda_1 (\text{Cov}(\bar{X})) = \lambda_1 \left( \text{Cov}(\bar{X}) = \frac{n-1}{n} \text{Cov}(X) + \Delta \right) \leq \frac{n-1}{n} \lambda_1 (\text{Cov}(X)) + \lambda_1 (\Delta)
\]

\[
= \frac{n-1}{n} \lambda_1 (\text{Cov}(X)) + \frac{||x||^2}{n}.
\]

As the smallest eigenvalue of \( xx^T \) is zero, we can also show that

\[
\lambda_1 (\text{Cov}(\bar{X})) \geq \frac{n-1}{n} \lambda_1 (\text{Cov}(X)),
\]

thus we have

\[
\frac{n-1}{n} \lambda_1 (\text{Cov}(X)) \leq \lambda_1 (\text{Cov}(\bar{X})) \leq \frac{n-1}{n} \lambda_1 (\text{Cov}(X)) + \frac{||x||^2}{n}
\]

The above inequity says that the principle components of the updated data will be always greater or equal to the batch one!

2. Bound on Eigenspaces Inclination

Assume that \( V_1 \) and \( V_2 \) are the subspaces generated by any corresponding principle components of \( \text{Cov}(X) \) and \( \text{Cov}(\bar{X}) \). Therefore, based on Davis-Kahan Theorem (See, Appendix B), the angle distance \( \theta \) between the eigenspace generated by the principle components of \( X \) and \( \bar{X} \) is bounded by some value depending on the length of the new added data point \( x \). Namely, we have

\[
\sin 2\theta \leq \frac{2||\Delta||}{\delta} = \frac{2||x||^2}{n\delta}.
\]

where \( \delta \) is gap between the eigenvalues of \( \text{Cov}(X) \).

It is clear that for large enough \( n \), the two principle components will coincide, meaning that there will be no significant change in the maximum direction of variation of data. This observation is really an interesting! As it shows that the deviation of the existing principle component will not exceed the value \( \frac{||x||^2}{n\delta} \). As we have mentioned in Section IV this could be used to infer a threshold while ruining PCA-based algorithms, that is, we only process certain amount of the data and ignore the rest.

VII. EXPERIMENTS

In this section we have used the \textit{iris} data set (see, [17]) to show the convergence of the dominant eigenvalue of \( \text{Cov}(\bar{X}) \) to the one of \( \text{Cov}(X) \) as we experience large amount of data (i.e., \( n \gg 1 \)). In Figure 2, the left subplot shows this convergence, whereas the right subplot depicts the bound on the deviation of the first components.

VIII. CONCLUSION

In this paper we contextualized online updates of data as a matrix perturbation problem. Up to principle components, we showed that online updating of data is equivalent to perturbing a scalar multiple of the batch covariance matrix by a symmetric matrix. It has been shown that as \( n \rightarrow n_0 \gg 1 \), the updated covariance matrix will become almost similar to the batch one. Our setup encourages the use of a wide range of matrix perturbation theoretic results in the context of online learning. Symmetric perturbation, in particular, is one of the simplest classes of matrix perturbations which has characteristics that makes it very useful to reveal many aspects of online learning.
\[ B \begin{align*} \lambda_{\min} &= \lambda_n(A) \leq \lambda_{n-1}(A) \leq \lambda_{n-2}(A) \\ &\vdots \leq \lambda_2(A) \leq \lambda_1(A) = \lambda_{\max}, \\
\lambda_{\min} &= \lambda_n(B) \leq \lambda_{n-1}(B) \leq \lambda_{n-2}(B) \\ &\vdots \leq \lambda_2(B) \leq \lambda_1(B) = \lambda_{\max}, \\
\lambda_{\min} &= \lambda_n(A + B) \leq \lambda_{n-1}(A + B) \leq \lambda_{n-2}(A + B) \ldots \leq \lambda_2(A + B) \leq \lambda_1(A + B) = \lambda_{\max}, \\
\text{then for each } k = 1, 2, \ldots, n, \text{ we have} \\
\lambda_k(A) + \lambda_n(B) &\leq \lambda_k(A + B) \leq \lambda_k(A) + \lambda_1(B). \quad \square \\ \end{align*} \]

B. Davis-Kahan, [3]

Let \( X \) and \( \Delta \) be symmetric matrices. Assume that \( X = Q\Lambda Q^T = Q\text{diag}(\lambda_i)Q^T \) and \( X + \Delta = \tilde{Q}\text{diag}(\tilde{\lambda})\tilde{Q}^T \).

Also let’s write

\[ Q = (q_1, q_2, \ldots, q_n) \text{ and } \tilde{Q} = (\tilde{q}_1, \tilde{q}_2, \ldots, \tilde{q}_n). \]

Let \( \theta_i \) denote the acute angle between the \( q_i \) and \( \tilde{q}_i \), then,

\[ \sin 2\theta_i \leq \frac{2\Delta}{\min_{i \neq j} |\lambda_i - \lambda_j|} \]

provided that \( \min_{i \neq j} |\lambda_i - \lambda_j| \neq 0. \]

APPENDIX

A. Weyl’s Theorem [44]

Assume that \( A \) and \( B \) are two Hermitian matrices. Assume that their eigenvalues are ordered in a increasing order as follows.

\[ \lambda_{\min} = \lambda_n(A) \leq \lambda_{n-1}(A) \leq \lambda_{n-2}(A) \\ &\vdots \leq \lambda_2(A) \leq \lambda_1(A) = \lambda_{\max}, \\
\lambda_{\min} = \lambda_n(B) \leq \lambda_{n-1}(B) \leq \lambda_{n-2}(B) \\ &\vdots \leq \lambda_2(B) \leq \lambda_1(B) = \lambda_{\max}, \\
\lambda_{\min} = \lambda_n(A + B) \leq \lambda_{n-1}(A + B) \leq \lambda_{n-2}(A + B) \ldots \leq \lambda_2(A + B) \leq \lambda_1(A + B) = \lambda_{\max}, \]

REFERENCES