Principal Component Analysis over encrypted data using homomorphic encryption

Hilder V. L. Pereira[1] **, Diego F. Aranha**[1]

[1] Institute of Computing (UNICAMP)
Av. Albert Einstein, 1251, 13083-852, Campinas-SP, Brazil
hilder@lasca.ic.unicamp.br,dfaranha@ic.unicamp.br,

Abstract. *We describe an algorithm to perform Principal Component Analysis (PCA) over encrypted data using homomorphic encryption. PCA is a fundamental tool for exploratory data analysis and dimensionality reduction, and thus a useful application for privacy-preserving computation in the cloud.*

1. Introduction

The increasingly intrusive behavior of governments and corporations and sensitive information leaks observed this year put into question the long-term viability of cloud computing as the prominent industry paradigm. Although this was an inherent risk since the introduction of cloud computing, the associated security and privacy issues of delegating computing to a third party became self-evident only recently.

A possible solution to accommodate these conflicting requirements is *computing over encrypted data*. In this model, data is encrypted by a transformation which conserves part of their structure and allows further execution of certain operations. Because of practical difficulties with *fully* homomorphic encryption that allows arbitrary computation in type and number of operations, a growing research area is dedicated to study *partially* homomorphic schemes and to adapt algorithms to work correctly in the encrypted domain. In this work, we propose an algorithm for performing PCA over encrypted data stored in the cloud. PCA is a fundamental step in data analysis and machine learning, thus a promising application for privacy-preserving computing. The proposed algorithm is non-interactive in nature and compatible with somewhat homomorphic encryption schemes.

2. Preliminaries

In this section, we recall basic Linear Algebra, without proof due to space constraints.

Definition 2.1 (Eigenvector and eigenvalue)**.** *Let X be a real matrix in $\mathbb{R}^{n\times n}$. We say that a scalar $\lambda \in \mathbb{R}$ is a eigenvalue of X if there exists a non-zero vector $v \in \mathbb{R}^n$ such that $Xv = \lambda v$. We also say that v is the eigenvector associated with λ and that (λ, v) is an eigenpair of X. Eigenvectors are invariant to multiplication by a scalar and the dominant eigenvalue of X is the one with the largest absolute value.*

Definition 2.2 (Shifting eigenpairs)**.** *We say that a procedure* shifts *the eigenvalues of a matrix X if it returns any matrix B such that the dominant eigenvalue of B is equal to the second dominant eigenvalue of X and their associated eigenvectors are the same. More formally, given X and dominant eigenpairs (λ_i, v_i), a function f shifts the eigenvalues of X if $f(X) = B \in \mathbb{R}^{n\times n}$ with dominant eigenpairs (λ_{i+1}, v_{i+1}).*

Theorem 2.3 (Spectral Theorem [Watkins 2005]). *Suppose $A \in \mathbb{R}^{n \times n}$ is symmetric. Then, it can be written as $A = UDU^T$, where U is a orthogonal matrix where each column is a normalized eigenvector and D is a diagonal matrix with the eigenvalues on the principal diagonal in an order correspondent to the columns of U. In other words, for $i \in \{1, 2, .., n\}$, the pair (D_{ii}, U_i) is an eigenpair, where U_i is the i-th column of U.*

Corollary 2.4 (Symmetric matrix as a sum). *Let $A \in \mathbb{R}^{n \times n}$ be symmetric and (λ_1, v_1), (λ_2, v_2), ..., (λ_n, v_n), eigenpairs of A, with $||v_i|| = 1$ for $i \in \{1, 2, .., n\}$. Then, A may be written as $A = \sum_{i=1}^{n} \lambda_i v_i v_i^T$.*

3. Principal Component Analysis

The problem of finding the principal components of a data matrix X is equivalent to the problem of finding the eigenvectors of its covariance matrix. In general, the i-th principal component is the i-th dominant eigenvector [Jolliffe 2002]. Hence, to project the data into a K-dimensional space, we have to find the K dominant eigenvectors.

3.1. Power Method

The *Power Method* is a simple iterative algorithm to find a dominant eigenvector of a given matrix. Let $A \in \mathbb{R}^{n \times n}$ be a real matrix. We sample a random initial vector $u \in \mathbb{R}^n$ and multiply A by u repeatedly, generating the sequence Au, A^2u, A^3u, ..., that converges to a dominant eigenvector. If we write the initial vector u as a linear combination of the eigenvectors $v_1, v_2, .., v_n$, we have:

$$A^k u = A^k(\alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 v_3 + ... + \alpha_n v_n) = \lambda_1^k \alpha_1 v_1 + \lambda_2^k \alpha_2 v_2 + ... + \lambda_n^k \alpha_n v_n.$$

Assuming that v_1 is a dominant eigenvector, we have $|\lambda_1^k| > |\lambda_i^k|$, for $i \in \{2, 3, .., n\}$. Therefore, if we divide both sides by λ_1^k, it converges to a multiple of v_1:

$$\frac{A^k u}{\lambda_1^k} = \alpha_1 v_1 + \frac{\lambda_2^k}{\lambda_1^k} \alpha_2 v_2 + ... + \frac{\lambda_n^k}{\lambda_1^k} \alpha_n v_n.$$

In order to avoid underflow and overflow in practice, it is common to divide the sequence by a scaling factor θ_k. The resulting algorithm for the Power Method can be found below:

```
1 powerMethod(A)
2     N = A.lines
3     u = randomVector(N)
4     for k = 1 to STEPS
5         u = Au / θ_k
6     return u
```

3.2. Finding K principal components

Our strategy to find the principal components is to calculate the covariance matrix of the data and find the K dominant eigenvectors by repeatedly using the Power Method and a shifting procedure. Since the covariance matrix is symmetric, the following function works as a shifting procedure:

```
eigenShift(A, dominant eigenvector v)
    u = v/||v||
    return B = A − Auu^T
```

Theorem 3.1. *Let A be a $n \times n$ real symmetric matrix. Then the function* eigenShift *shifts the eigenvalues of A.*

Proof. Since the first operation of *eigenShift* is normalizing v, we have that v becomes equal to v_1, the normalized dominant eigenvector. Since $v_1^T v_1 = ||v_1||^2 = 1$, we have

$$Bv_1 = Av_1 - (Av_1 v_1^T)v_1 = Av_1 - Av_1(v_1^T v_1) = Av_1 - Av_1 = \lambda_1 v_1 - \lambda_1 v_1 = 0 \cdot v_1$$

which proves that v_1 is also an eigenvector of B but now associated with a new eigenvalue $\lambda_{new} = 0$. By Corollary 2.4, the matrix A may be written as $A = \sum_{i=1}^{n} \lambda_i v_i v_i^T$, and thus

$$B = \lambda_2 v_2 v_2^T + \lambda_3 v_3 v_3^T + ... + \lambda_n v_n v_n^T.$$

For all $i \in \{2, 3, .., n\}$, we have $Bv_i = \lambda_2 v_2 v_2^T v_i + ... + \lambda_i v_i v_i^T v_i + ... + \lambda_n v_n v_n^T v_i$. By Theorem 2.3, all the eigenvectors are orthogonal, so for $j \neq i$, the product $v_j^T v_i$ is equal to 0, and the product $v_i^T v_i$ is equal to 1. Then, $Bv_i = 0+0+...+0+\lambda_i v_i \cdot 1+0+...+0 = \lambda_i v_i$, which proves that all the other eigenpairs of A are also eigenpairs of B. Therefore, all the eigenvectors of A are also eigenvectors of B, the dominant eigenvector of A is associated with the eigenvalue $\lambda_{new} = 0$, and the second dominant eigenvalue of A is the dominant eigenvalue of B. □

In order to calculate the covariance matrix, we just have to set the mean of each variable (column of the data matrix) to zero and then make a matrix multiplication.

```
covarianceMatrix(X)
    N = X.lines
    P = X.columns
    for j = 1 to P
        μ = 0
        for i = 1 to N
            μ = μ + X[i][j]
        μ = μ/N
        /* Subtract the mean. */
        for i = 1 to N
            X[i][j] = X[i][j] − μ
    C = X^T * X
    return C
```

Our proposal for solving the PCA problem is the following:

```
PCA(X, new dimension K)
    C = covariance(A)
    pcs = ∅
    for i = 1 to K
        pc_i = powerMethod(C)
        C = eigenShift(C, pc_i)
        pcs = {pc_i} ∪ pcs
    return pcs
```

4. Homomorphic version

Employing a Somewhat Homomorphic Encryption (SWE) scheme such as [Bos et al. 2013] for privacy-preserving computation involves some restrictions on the operations that can be performed on the data. Usually, we can only make additions and a few multiplications over the ciphertexts, and general divisions are not viable. If encoding real numbers is possible using [Aono et al. 2015], we can also divide the ciphertexts by constants or any other known values (number n of elements submitted by the client, for example). Because of these restrictions, we have to modify the algorithms to remove divisions between ciphertexts and to minimize the number of consecutive multiplications.

For the Power Method, the value θ_k can be chosen as a constant, and the computation of the covariance matrix only employs a value known *a priori*, thus divisions can be performed between ciphertexts and plaintexts. The remaining obstacle is the *eigenShift* procedure. Since the components of the vectors are encrypted, we cannot normalize v dividing it by its norm (first operation of the *eigenShift*) as $B = A - A\frac{v}{||v||}\frac{v^T}{||v||}$.

However, Definitions 2.1 tell us that B and $||v||^2 B$ have the same eigenvectors. Hence:

$$||v||^2 B = ||v||^2 (A - A\frac{v}{||v||}\frac{v^T}{||v||}) = ||v||^2 A - Avv^T,$$

which means that we can compute $||v||^2 B$ from A and v without divisions between ciphertexts. Finally, using the relation between the inner product and the Euclidean norm, namely, $v^T v = ||v||^2$, the homomorphic version of the *eigenShift* function can be described as follows:

```
homomorphicShift(A, v)
    α = innerProduct(v, v)
    B = αA − Avv^T
    return B
```

This way, the entire Power Method can be computed over encrypted data.

5. Conclusion

Principal Component Analysis can be computed in a privacy-preserving way, by adapting all of the required steps in the Power Method to remove expensive divisions and employing a Somewhat Homomorphic Encryption scheme with a bounded number of multiplications. As far as we know, this is the first non-interactive proposal for performing PCA over encrypted data in the cloud.

References

Aono, Y., Hayashi, T., Phong, L. T., and Wang, L. (2015). Fast and secure linear regression and biometric authentication with security update. Cryptology ePrint Archive, Report 2015/692. `http://eprint.iacr.org/`.

Bos, J. W., Lauter, K., Loftus, J., and Naehrig, M. (2013). Improved security for a ring-based fully homomorphic encryption scheme. In *Cryptography and Coding (IMACC)*, pages 45–64. Springer.

Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer Series in Statistics.

Watkins, D. S. (2005). *Fundamentals of Matrix Computations*. Wiley, 2nd edition.