Approximating Eigenvectors with Fixed-Point Arithmetic: A Step Towards Secure Spectral Clustering

Lisa Steverink, Thijs Veugen, and Martin B. van Gijzen

Abstract We investigate the adaptation of the spectral clustering algorithm to the privacy preserving domain. Spectral clustering is a data mining technique that divides points according to a measure of connectivity in a data graph. When the matrix data are privacy sensitive, cryptographic techniques can be applied to protect the data. A pivotal part of spectral clustering is the partial eigendecomposition of the graph Laplacian. The Lanczos algorithm is used to approximate the eigenvectors of the Laplacian. Many cryptographic techniques are designed to work with positive integers, whereas the numerical algorithms are generally applied in the real domain. To overcome this problem, the Lanczos algorithm is adapted to be performed with fixed-point arithmetic. Square roots are eliminated and floating-point computations are transformed to fixed-point computations. The effects of these adaptations on the accuracy and stability of the algorithm are investigated using standard datasets. The performance of the original and the adapted algorithm is similar when few eigenvectors are needed. For a large number of eigenvectors loss of orthogonality affects the results.

1 Introduction

Computing eigenvectors of matrices has many important applications. One example is principal component analysis, a technique that is used to study large data sets such as those encountered in bioinformatics, data mining, chemical research, psychology,

L. Steverink · M. B. van Gijzen
Delft University of Technology, Delft, The Netherlands
e-mail: M.B.vanGijzen@TUDelft.nl

T. Veugen (✉)
TNO, Unit ICT, The Hague, The Netherlands

Cryptology Department, CWI, Amsterdam, The Netherlands
e-mail: thijs.veugen@tno.nl

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139, https://doi.org/10.1007/978-3-030-55874-1_112

and in marketing. Another example is the characterisation of DNA sequences [17] in bioinformatics. Large graphs have become an important data source for applications from social networks, mobile and web applications to biomedical research, providing great value in both business and scientific research. Particularly, spectral analysis of graphs gives important results pertinent to community detection, PageRank, and spectral clustering.

Especially when the matrix data are sensitive, security measures should be taken to overcome undesired leakage of data during the computation of eigenvectors. The data could be commercially sensitive, but also privacy sensitive, as is often the case with medical data. As data may be collected from different sources, and data processing is increasingly performed in the cloud or by external parties which are not allowed to learn the contents, techniques like data perturbation, homomorphic encryption [10], or secret sharing [1], are frequently used. Unfortunately, such cryptographic techniques are designed to work with integers, whereas the numerical algorithms that are used to compute eigenvectors are designed to work with real numbers. This means that these floating-point based algorithms have to be transformed to fixed-point based algorithms. This has a great influence on the accuracy and stability of the existing, often iterative, approaches.

In this paper, we investigate the effect of approximating eigenvectors with fixed-point arithmetic, and focus on the accuracy and stability of the adjusted numerical algorithms. Although we do not design the complete cryptographic protocols for computing eigenvectors in the encrypted domain, we pay attention to avoid complex operations on encrypted (or secret-shared) numbers, such as square roots and integer divisions [4, 15]. We perform spectral clustering, and compare the results of our adapted numerical algorithms in $\mathbb{Z}_N$ to the original algorithms in $\mathbb{R}$ on three datasets.

The paper is organized as follows. First, related work and preliminaries will be discussed. Then we present the adapted Lanczos algorithm that works on positive integers. Subsequently, the accuracy analysis of secure spectral clustering that makes use of both the original and adapted algorithm, is given. We end with the conclusions.

This paper is based on the research described in [14], which contains many additional algorithmic details and experimental results.

1.1 Related Work

Power methods are known in cryptography for computing square roots or dividing integers [5]. Although they can also be used to find eigenvectors, there is not much previous work done on the numerical analysis of finding eigenvectors in the integer domain. Nikolaenko et al. presented a privacy preserving way of factorising a matrix for recommendation purposes [8], by combining homomorphic encryption and garbled circuits. Erkin et al. designed a secure method for performing k-means clustering [2] by means of additively homomorphic encryption, but this does not require computing eigenvectors. Sharma and Chen [12, 13] recently

showed how spectral analysis could be securely done in the cloud, using additively homomorphic encryption and differential privacy. The focus of all related work is on the computational complexity, while we focus on accuracy, with complexity in mind.

1.2 Preliminaries

Spectral Clustering

The spectral clustering algorithm is able to find k, not necessarily convex clusters of similar points by mapping the data points to a k-dimensional space in which the similar points form convex sets. These convex sets can be clustered with a k-means algorithm. In spectral clustering, the dataset is represented as a graph G with weighted edges [16]. We aim to maximize the weights within the clusters, while the weights between clusters are low. A Laplacian matrix L is defined, which contains information about the connected components of G. The first k eigenvectors of Laplacian L approach indicator vectors of the connected components of G, and form convex clusters. Therefore, we are interested in finding the k eigenvectors of L that correspond to the k smallest eigenvalues. The complexity of computing the entire eigendecomposition of $L \in \mathbb{R}^{n \times n}$ is $O(n^3)$. Moreover, if the data set needs to be clustered into k clusters, only k eigenvectors are required. Therefore, we use numerical algorithms to approximate the k smallest eigenvalues and their corresponding eigenvectors.

The Lanczos Algorithm in $\mathbb{R}$

The Lanczos algorithm is used to reduce the Laplacian matrix L to a tridiagonal matrix T (the Ritz matrix) of which the eigenvalues (the Ritz values) approximate the eigenvalues of L. The Lanczos algorithm is shown in Algorithm 1 [3]. The inner product is indicated by a $\cdot$ between two vectors.

Algorithm 1: The Lanczos algorithm

Set $v_0 = \underline{0}$ and $\beta_1 = 1$.
Generate a random vector $v_1 \in (0, 1)^n \subset \mathbb{R}^n$.
for $j = 1, 2, \ldots, m - 1$ **do**
  $\alpha_j \leftarrow (Lv_j \cdot v_j)/(v_j \cdot v_j)$
  $r_j \leftarrow Lv_j - \alpha_j v_j - \beta_j v_{j-1}$
  $\beta_{j+1} \leftarrow \|r_j\|_2$
  $v_{j+1} \leftarrow r_j/\beta_{j+1}$
end
$\alpha_m \leftarrow (Lv_m \cdot v_m)/(v_m \cdot v_m)$

After m iterations, Algorithm 1 yields Ritz matrix T:

$$T = \begin{pmatrix} \alpha_1 & \beta_2 & & & 0 \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{m-1} & \alpha_{m-1} & \beta_m \\ 0 & & & \beta_m & \alpha_m \end{pmatrix}. \tag{1}$$

In exact arithmetic, the vectors $v_1, \ldots v_m$ form an orthonormal basis for the so-called Krylov subspace $\mathcal{K}_m(L, v_1)$ of dimension m, which is defined as

$$\mathcal{K}_m(L, v_1) = \text{span}\{v_1, Lv_1, \cdots, L^{m-1}v_1\}.$$

The eigenvalues of T are increasingly better estimates of the eigenvalues of L as its size grows. The extremal Ritz values are the first to converge in the spectrum of T.

Computing in the Integer Domain

Cryptographic techniques are designed to work on positive integers. Therefore, we translate the Lanczos algorithm to $\mathbb{Z}_N$, which is the set $\{0, 1, \ldots, N-1\}$ with modular arithmetic. Because of security requirements, N is an odd 2048-bit number. The domain $\mathbb{Z}_N$ forms the *message space* of messages that can be encrypted. Modular arithmetic is used on $\mathbb{Z}_N$. Integer division is defined as follows:

Definition 1 Let $a, b \in \mathbb{Z}$. The integer division $a \div b$ is defined as the integer q such that $a = qb + r$ with *remainder* $r \in \mathbb{Z}$, where $0 \leq r < b$.

Fixed-point arithmetic is used to represent fractions as signed integers [4]. By multiplying fractions with 10^d, a signed integer is obtained, where d is the scaling parameter that determines the number of decimals that will be stored. Scaling fractions with 10^d has implications for the operations in the integer domain. To preserve the scaling parameter 10^d when dividing two numbers, the numerator is first multiplied by 10^d. We assume that each integer division on numbers in fixed-point arithmetic has this implicit additional multiplication. Moreover, we define the fixed-point arithmetic multiplication operations as follows:

Definition 2 Let a and b be fixed-point integers. The fixed-point integer multiplication $*$ is defined as

$$a * b = ab10^{-d}.$$

Indeed, multiplying $a10^{-d}$ and $b10^{-d}$ gives $ab10^{-2d} = (a * b)10^{-d}$, so $a * b$ is the scaled version of the product. The operator $*$ is also used to denote fixed-point matrix multiplications. Finally, $\langle v_j, v_j \rangle$ denotes the inner product or a matrix-vector

product that makes use of the fixed-point integer multiplication. The following map ψ can encode signed integers (with absolute value less than $N/2$) as positive integers (less than N):

$$\psi : \{-(N-1)/2, \ldots, 0, \ldots, (N-1)/2\} \longrightarrow \mathbb{Z}_N, \tag{2}$$

$$x \longmapsto x \mod N. \tag{3}$$

Informally stated, the upper half of the domain $\mathbb{Z}_N$ is used to represent the negative integers of maximum bit length 2047. Using these definitions, we adapt the Lanczos algorithm to the integer domain. All computations in this algorithm are performed modulo N. In the algorithm, "mod N" will be omitted.

2 Lanczos Algorithm on Integers

The standard Lanczos algorithm in Algorithm 1 incorporates a normalization of the Lanczos vectors (see line 7). However, the square root operation (within line 6) is expensive in a finite field [7]. Therefore, we propose to perform an unnormalized version of the Lanczos algorithm [9] in the integer domain. Due to this lack of normalization, the entries of v_j tend to grow as the algorithm progresses. Thus, there is a danger of overflow of message space $\mathbb{Z}_N$. The unnormalized Lanczos algorithm in $\mathbb{Z}_N$ is given in Algorithm 2. The entries of starting vector v_1 are chosen randomly from (0, 1) and scaled by 10^d to integers. The Laplacian matrix L contains integer values and is unscaled. Note that this alternative Lanczos algorithm yields an unsymmetric matrix T, because the β_j from Algorithm 1 are now constants:

$$T = \begin{pmatrix} \alpha_1 & \gamma_2 & & & 0 \\ 10^d & \alpha_2 & \gamma_3 & & \\ & 10^d & \alpha_3 & \gamma_4 & \\ 0 & & \ddots & \ddots & \ddots \end{pmatrix}. \tag{4}$$

The above algorithm computes basis vectors $v_1 \cdots v_m$ for the Krylov subspace, and a matrix T_m whose eigenvalues (called Ritzvalues) converge to the eigenvalues of L. Additionally, [14] explains how to use this information to compute the Ritz values and corresponding Ritz vectors (approximating the eigenvectors) in the integer domain.

Algorithm 2: Unnormalized fixed-point Lanczos algorithm in $\mathbb{Z}_N$

$v_0 \leftarrow \underline{0}$ and $\beta_1 \leftarrow 0$
$\gamma_1 \leftarrow 0$
Generate a random vector $v_1 \in \{1, \ldots, 10^d\}^n$
for $j = 1, 2, \ldots, m-1$ **do**
  $L_j \leftarrow \langle L, v_j \rangle$
  $\alpha_j \leftarrow \langle v_j, L_j \rangle \div \langle v_j, v_j \rangle$
  $\beta_{j+1} \leftarrow 1$
  $v_{j+1} \leftarrow L_j - \alpha_j * v_j - \gamma_j * v_{j-1}$
  $\gamma_{j+1} \leftarrow \beta_{j+1} \langle v_{j+1}, v_{j+1} \rangle \div \langle v_j, v_j \rangle$
end
$L_m \leftarrow \langle L, v_m \rangle$
$\alpha_m \leftarrow \langle v_m, L_m \rangle \div \langle v_m, v_m \rangle$

3 Accuracy Analysis

In order to investigate the influence of adapting the Lanczos algorithm to the integer domain, the performance of the algorithm in $\mathbb{R}$ and $\mathbb{Z}_N$ is compared. The performance is measured by computing the accuracy of the Ritz values and Ritz vectors, the clustering accuracy and a measure of compactness. The value of N is chosen to comprise 2048 bits. Therefore, we say that overflow occurs when a number becomes larger than 2047 bits, since we need one bit to represent negative numbers. The algorithms were implemented in Python 3.6 and tested on three real datasets. Three datasets from the UCI Machine Learning Repository were used to assess the spectral clustering algorithm in $\mathbb{Z}_N$: the Wisconsin Breast Cancer Dataset, the Yeast5 Dataset and the Yeast10 Dataset [6]. These datasets were chosen for their variety in size and number of clusters. Moreover, a suitable Laplacian could be constructed in the integer domain for these datasets. The Wisconsin Breast Cancer Dataset has size 699×9 and should be clustered into two clusters. The Yeast5 Dataset has size 384×17 and contains five clusters. Finally, the Yeast10 Dataset is a 1484×8 dataset in which ten clusters can be distinguished. Below, we only give the numerical results for the Wisconsin Breast Cancer Dataset. We refer to [14] for a complete description of the numerical results for the other two data sets.

The accuracy of the Ritz value θ_i to eigenvalue λ_i of L is assessed with the absolute error:

$$|\theta_i - \lambda_i|. \tag{5}$$

The accuracy of the corresponding Ritz vector $\tilde{u}_i$ to an eigenvector u_i of L is measured with the absolute cosine of the angle α between the vectors:

$$|\cos(\alpha)| = \frac{|\tilde{u}_i \cdot u_i|}{\|\tilde{u}_i\| \cdot \|u_i\|}. \tag{6}$$

The *silhouette value* is a measure of the compactness and separation of clusters [11]. The distance of the data point to other data points in the same cluster is compared to the distance to data points in other clusters. Formally, the silhouette value of data point i is defined as

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \tag{7}$$

where $a(i)$ is the average distance from point i to other points in the same cluster, and $b(i)$ is the minimum average distance from point i to points in a different cluster. The squared Euclidean distance is used in the computation of the silhouette value. From the above definition it follows that

$$-1 \leq s(i) \leq 1 \tag{8}$$

for each data point i. A positive silhouette value indicates that the data point is clustered well. From a negative silhouette value we conclude that a data point has been misclassified.

3.1 Wisconsin Breast Cancer Dataset

A scaling parameter $d = 6$ is required to obtain sufficient accuracy in $\mathbb{Z}_N$. Table 1 shows the relative accuracy of the first two Ritz values. Both in $\mathbb{R}$ and in $\mathbb{Z}_N$ the eigenvalues are approximated well. The accuracy is higher in $\mathbb{R}$. Furthermore, the cosine of the angle between the Ritz vectors and the exact eigenvectors is shown. The values show that the eigenvectors are approximated with high accuracy. Table 2 shows the cluster quality. Both in $\mathbb{R}$ and in $\mathbb{Z}_N$, the first two eigenvectors are approximated well enough to form the correct convex clusters. The maximum entry bit length is 51 in matrix T and 76 in matrix V.

Table 1 The absolute error of the two smallest Ritz values ($\lambda_1 = 2.92700358$ and $\lambda_2 = 9.03710093\text{e}4$) and the accuracy of the corresponding Ritz vectors for the Wisconsin Breast Cancer dataset. Parameters: $d = 6$, $m = 6$

i	$\lvert\theta_i - \lambda_i\rvert\ \mathbb{R}$	$\lvert\theta_i - \lambda_i\rvert\ \mathbb{Z}_N$	$\lvert\cos\alpha\rvert\ \mathbb{R}$	$\lvert\cos\alpha\rvert\ \mathbb{Z}_N$
1	1.3157e−11	1.1549e−4	1.00000000	1.00000000
2	1.5449e−6	2.9566e−5	1.00000000	1.00000000

Table 2 Cluster quality of the Wisconsin Breast Cancer dataset. Parameters: $k = 2$, $d = 6$, $m = 6$

	Lanczos $\mathbb{R}$	Lanczos $\mathbb{Z}_N$
Cluster accuracy	95.85%	95.85%
Silhouette value	0.9118	0.9118

4 Conclusions

We conclude that a few of the smallest eigenvalues of the Laplacian could be approximated well in the integer domain. The accuracy of the algorithm in $\mathbb{R}$ and $\mathbb{Z}_N$ is similar, and the eigenvectors that correspond to the computed eigenvalues are approximated with high accuracy. For a small number of clusters, a good performance of the spectral clustering algorithm is achieved. As a higher number of clusters requires more iterations of the Lanczos algorithm, the loss of orthogonality may affect the accuracy of the spectral clustering algorithm in both domains, see [14].

References

1. Ben-David, A., Nisan, N., Pinkas, B.: FairplayMP - a secure multi-party computation system. In: ACM CCS (2008)
2. Erkin, Z., Veugen, T., Toft, T., Lagendijk, R.L.: Privacy-preserving user clustering in a social network. In: IEEE International Workshop on Information Forensics and Security (2009)
3. Golub, G., Van Loan, C.: Matrix Computations. Johns Hopkins University Press (1996)
4. Hoogh de, S.J.A.: Design of large scale applications of secure multiparty computation: secure linear programming. Ph.D. thesis, Eindhoven University of Technology (2012)
5. Jakobsen, T.: Secure multi-party computation on integers (2006)
6. Lichman, M.: UCI machine learning repository. http://archive.ics.uci.edu/ml (2013)
7. Liedel, M.: Secure distributed computation of the square root and applications. In: International Conference on Information Security Practice and Experience, pp. 277–288. Springer (2012)
8. Nikolaenko, V., Ioannidis, S., Weinsberg, U., Joye, M., Taft, N., Boneh, D.: Privacy-preserving matrix factorization. In: Proceedings of the 2013 ACM SIGSAC conference on Computer and communications security, pp. 801–812. ACM (2013)
9. Paige, C.C.: The computation of eigenvalues and eigenvectors of very large sparse matrices. Ph.D. thesis, University of London (1971)
10. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Proceedings of Eurocrypt 1999, *Lecture Notes in Computer Science*, vol. 1592, pp. 223–238. Springer-Verlag (1999). citeseer.ist.psu.edu/article/paillier99publickey.html
11. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics **20**, 53–65 (1987)
12. Sharma, S., Chen, K.: Privategraph: a cloud-centric system for spectral analysis of large encrypted graphs. In: IEEE 37th International Conference on Distributed Computing Systems, pp. 2507–2510. IEEE Computer Society (2017)
13. Sharma, S., Powers, J., Chen, K.: Privacy-preserving spectral analysis of large graphs in public clouds. In: Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security, pp. 71–82. ACM (2016)
14. Steverink, M.L.: Secure spectral clustering: the approximation of eigenvectors in the integer domain. Master's thesis, Delft University of Technology (2017). http://resolver.tudelft.nl/uuid:284fc7f2-440d-4435-ae04-fea83d12c12f
15. Veugen, T.: Encrypted integer division and secure comparison. International Journal of Applied Cryptography **3**, 166–180 (2014)
16. Von Luxburg, U.: A tutorial on spectral clustering. Statistics and computing **17**(4), 395–416 (2007)
17. Yu, H.J., Huang, D.S.: Graphical representation for DNA sequences via joint diagonalization of matrix pencil. IEEE Journal of Biomedical and Health Informatics **17**(3), 503–511 (2013)