Some Extension of Sparse Principal Component Analysis

Thanh D. X. Duong and Hung V. Nguyen

Abstract—Given a covariance matrix, sparse principal component analysis (SPCA) considers the problem of maximizing the variance explained by a particular linear combination of the input variables where the number of nonzero coefficients is constrained. In some applications, the coefficients in this combination are required to be non-negative. Moreover, when loading an input variable is associated an individual cost, we need incorporate weights, which represent the loading cost of input variables, into sparsity constraint. And in this paper, we consider problems of SPCA with weighted sparsity constraint and/or non-negative sparsity constraint. These problems are reduced to solving some semi-definite programming ones via convex relaxation technique. Numerical results show that the method is efficient and reliable in practice.

Index Terms—Iterative re-weighting, non-negative constraint principal component analysis, principal component analysis, semi-definite relaxation, sparse principal component analysis.

I. INTRODUCTION

Sparse decompositions of data are required in many applications. In economics, sparsity increases the efficiency and reduces risk of a portfolio [7]; it also implies lower transaction cost in financial asset trading strategies. In computer vision, sparse decomposition is related to the extraction of some concerned pixels which are relevant parts from images [17]. In machine learning, sparsity is closely related to feature selection and to improved generalization of learning algorithms. And in biology, the sparsity is necessary for finding focalized local patterns hidden in gene expression data analysis [1].

Being first introduced by Pearson in [28], and developed independently by Hotelling in [6], principal component analysis (PCA) has now become a popular technique used to reduce multidimensional data sets to lower dimensions for analysis with applications throughout science and engineering, see [20]. This reduction is achieved by transforming to a new set of variables, the principal components, which are uncorrelated and ordered so that the first few retain most of the variation present in all of the original variables. It also can be performed via a singular value decomposition of the data matrix or an eigenvalue decomposition of the data covariance matrix.

A drawback of PCA is the lack of sparseness of the principal vectors - since the principal components are usually linear combinations of all variables and the loadings are typically non-zero - while sparse decompositions of data are required in many applications such as: sparsity increases the efficiency and reduces risk of a portfolio [7]; sparse decomposition is related to the extraction of some concerned pixels which are relevant parts from images [17]; sparsity is closely related to feature selection and to improved generalization of learning algorithms; sparsity is necessary for finding focalized local patterns hidden in gene expression data analysis [1]. This leads to appearance of several methods to find sparse principal components explaining most of the variance present in the data. To achieve this, it is necessary to sacrifice some of the explained variance and the orthogonality of the principal components. Rotation techniques [14] can be considered the first approach. Simple principal components [13] are studied by restricting the loadings to take values from a small set of allowable integers such as 0, 1 and -1. Simple thresholding technique [2] was an ad hoc way to deal with the problem, where the loadings with small absolute value are thresholded to zero. SCotTLASS [9] and SLRA [10], [11] were introduced to get modified principal components with possible zero loadings. ESPCA [21] used discrete spectral formulation based on variational eigenvalue bounds and an effective greedy strategy to give provably optimal solutions via branch-and-bound search.

SPCA [12] was proposed via a regression type optimization problem. And DSPCA [4], [16] is the current state-of-the-art method, which relaxes a hard cardinality constraint with a convex approximation. Recently, [26] discusses several options, comparing the variance vs. orthogonality/sparsity tradeoffs they imply. Finally, [30] shows that some naive approaches have the significantly worst convergence rates than the relaxation approach as in [4], [5], [16], [22].

Since the outputs of DSPCA [16] are not satisfied sparsity constraint - i.e. if we hope to find a principal component with less than $k$ non-zero entries, the output often contains more than $k$ non-zero entries - it is reasonable to add a post-processing technique to enhance the sparsity of the principal components. Recently, re-weighted $l_1$ minimization [3], [19] is a useful technique to enhancing sparsity output of the combinatorial optimization:

$$\min \|y\|_1 \quad \text{subject to } y = \Phi x,$$

where $\|y\|_1 = |\{i : x_i \neq 0\}|$. Replacing the linear equation constraints in the above combinatorial optimization by linear matrix inequality constraints, we get an approach to refine the sparsity of the principal components. This approach has been done in [22], where we add non-negative property to the principal components.

In practice, there are many applications in which loading an input variable is associated an individual cost. In these cases, it is essential to incorporate weights, which represent the loading cost of input variables, into PCA. Weighted PCA
has thus been introduced and used in many applications such as learning from incomplete data in [29], giving an efficient search algorithm for motion data in [24], and face recognition in [27]. To solve this problem, we generalize the sparsity constraint with weighted sparsity constraint [5].

It is remarkable that we cannot obtain the desired sparsity of the output by using the re-weighted l1 minimization technique [3] even with linear equation constraints. To overcome this obstacle, we present a bisection algorithm exploiting truncation technique to find the principal component not only satisfying the desired sparsity constraint but also explaining the most variance [23].

In practice, the loadings of principal components are required to be non-negative and/or associated an individual cost. In this paper, we consider problems of SPCA with weighted sparsity constraint and/or non-negative sparsity constraint. This paper is organized as follows. The next section is the main results, where we presents a method which directly incorporates a weighted (and/or non-negative) sparsity criterion in the PCA problem formulation, then apply relaxation method and a bisection algorithm to find the best principle components which exactly satisfy the desired sparsity constraints. Section 3 is devoted to compare the proposed method with existing methods on both artificial data and real-life data.

Notation. In this paper, we denote the set of symmetric matrices of size n by \( S^n \), the vector of ones by \( 1 \), the cardinality (number of non-zero elements) of a vector \( x \) by \( \text{Card}(x) \), and the number of non-zero coefficients in a matrix \( X \) by \( \text{Card}(X) \). For \( X \in S^n \), the notation \( X \succeq 0 \) means that \( X \) is positive semi-definite, \( |X| \) is the matrix whose elements are the absolute values of the elements of \( X \), and \( \text{Tr}(X) = X_{11} + X_{22} + \ldots + X_{nn} \), which is the sum of diagonal entries.

II. MAIN RESULTS

In this section, we review semi-definite relaxation method [4], [5], [16], [22], [23] for SPCA with weighted sparsity constraint and/or non-negative sparsity constraint, refine the cardinality constraint via re-weighted \( l_1 \) minimization technique, and present a bisection algorithm to find the principle component which satisfies the required sparsity constraint when sacrifice some of the explained variance. Then, we apply the problem to decompose a data matrix into sparse principle components.

A. Semi-definite Relaxation

Let \( A \in S^n \) be a covariance matrix, i.e. \( A \succeq 0 \), and \( w \in R^n \) be a weight vector with \( w_i > 0 \) for all \( i = 1, \ldots, n \). We consider a problem of maximizing the variance of vector \( x \in R^n \) while constraining its weighted cardinality (and/or non-negative loadings):

\[
\begin{align*}
\text{maximize} & \quad x^T A x, \\
\text{subject to} & \quad \|x\|_2 = 1, \\
& \quad \sum_{i,j} w_i \delta(x) \leq k, (x \geq 0)
\end{align*}
\]

where \( \delta : R \rightarrow [0, 1] \) is defined by \( \delta(x) = \begin{cases} 0, & \text{if } x = 0, \\ 1, & \text{if } x \neq 0, \end{cases} \) and the given positive number \( k \) restricts the number of non-zero entries of the solution, thus the following inequality should be hold: \( \min_{i \neq j} w_i \leq k \leq \sum_{i,j} w_i \).

Hence, by choosing \( w = 1 \), the weighted sparsity constraint \( \sum_{i,j} w_i \delta(x) \leq k \) collapses to the classical sparsity constraint \( \text{Card}(x) \leq k \). In the case of non-negative property is not required for the loadings, we can remove the last constraint, \( x \geq 0 \).

Using semi-definite relaxation techniques [5], the problem is approximated by the following problem:

\[
\begin{align*}
\text{maximize} & \quad \text{Tr}(AX), \\
\text{subject to} & \quad \text{Tr}(X) = 1, \\
& \quad \sum_{i,j} w_i \delta(x) \leq k, \\
& \quad X \succeq 0, (X \geq 0).
\end{align*}
\]

This means, we will solve the semi-definite problem (2) to get solution \( X \), and an approximation solution of (1) is the dominant eigenvector of \( X \) (or the non-negative parts of the dominant eigenvector of \( X \) if non-negative property is required).

B. Cardinality Constraint Refinement

Let \( x_i \) be the approximation solution of (1). It is clear that \( x_i \) does not satisfying cardinality constraint \( \text{Card}(x) \leq k \) in general. Hence, we consider the following cardinality constraint refinement problem:

\[
\begin{align*}
\text{minimize} & \quad \sum_{i=1}^k w_i \delta(x_i), \\
\text{subject to} & \quad \|x\|_2 = 1, \\
& \quad x^T A x \succeq c, (x \geq 0),
\end{align*}
\]

where \( c = x_i^T A x_i \).

Let \( X = xx^T \succeq 0 \), the second order constraint \( x^T A x \succeq c \), and the non-convex constraint \( \|x\|_2 = 1 \) are transformed into linear constraints since \( \text{Tr}(AX) = x^T A x \) and \( \text{Tr}(X) = \|x\|_2 \).

Hence, using Lemma 2.1 in [5], the lifting procedure [1], [8], [16], [25] for semi-definite relaxation gives an equivalent problem of (3) as follows:

\[
\begin{align*}
\text{minimize} & \quad \sum_{i,j} w_i \delta(X_{ij}), \\
\text{subject to} & \quad \text{Tr}(X) = 1, \\
& \quad \text{Tr}(AX) \geq c, \\
& \quad X \succeq 0, (X \geq 0), \text{rank}(X) = 1
\end{align*}
\]

Next, apply truncation technique [1], [25] which drops the constraint \( \text{rank}(X) = 1 \), we can relax the problem (6) as follows:
\[
\text{minimize } \sum_{i,j=1}^{d} w_{ij} \delta(X_{ij}), \\
\text{subject to } \text{Tr}(X) = 1, \\
\text{Tr}(AX) \geq c, \\
X \succeq 0, (X \succeq 0). 
\] (5)

Using the re-weighted \( l_1 \) minimization which is recent technique to enhancing sparsity, see [3], we consider the following relaxation of the problem (5):

\[
\text{minimize } \text{Tr}(W^T X), \\
\text{subject to } \text{Tr}(X) = 1, \\
\text{Tr}(AX) \geq c, \\
X \succeq 0, (X \succeq 0). 
\] (6)

where \( W > 0 \) is positive weight matrix. We use a simple iterative algorithm that alternates between estimating \( X \) and redefining the weights [3] to give an extension of DSPCA (called EDSPCA in what follows):

1. Set the iteration count \( m \) to zero and solve DSPCA problem (4) to get the solution \( X^{(0)} \).
2. Update the weights:

\[
\text{for each } i, j = 1, n \quad w_{ij}^{(n+1)} = \frac{1}{X^{(m)}_{ij} + \varepsilon}. 
\] (7)

3. Solve the weighted problem (6) to get the solution \( X^{(n)} \).
4. Terminate when \( \text{Card}(x^{(n)}) \leq k \) or \( m \) attains a specified maximum number of iterations \( m_{\text{max}} \), where \( x^{(n)} \) is the dominant eigenvector of \( X^{(n)} \). Otherwise, increment \( m \) and go to step 2.

The semi-definite programming (SDP) problem (2) and (6) can be solved efficiently using interior-point solvers such as SEDUMI [31] or SDPT3 [32]. And we should set a truncation of \( X_{ij} \) to zero and solve DSPCA while \( \text{var}_{\text{var}} - \text{var}_{\text{low}} > \text{var}_{\text{error}} \)

\[
\text{set } \text{var}_{\text{new}} := 1/2(\text{var}_{\text{up}} + \text{var}_{\text{low}}) \\
\text{run the cardinality constraint refinement process with } c_{\ast} := \text{var}_{\text{new}} \text{ to get solution } x_{\ast} \\
\text{if } \text{Card}(x_{\ast}) \leq k \text{ then update } \text{var}_{\text{low}} := \text{var}_{\text{new}} \\
\text{else} \\
\quad \text{+ update } \text{var}_{\text{up}} := \text{var}_{\text{new}} \\
\quad x_{\ast} := \text{truncation of } x_{\ast} \\
\text{return the optimal principle component } x_{\ast}. 
\]

III. NUMERICAL EXPERIMENTS

In this section, we will compare the effectiveness of EDSPCA with the other methods mentioned in the introduction. We perform the test on an artificial data set proposed by H. Zou, T. Hastie, R. Tibshirani [12] and a well-known real-life benchmark data set - Pit Props data.

A. Artificial Data

To show the effectiveness of EDSPCA, we consider the simulation example proposed by H. Zou, T. Hastie, R. Tibshirani [12], which was used to test the other methods. In this example, there are three hidden factors which are better than simple thresholding method. Moreover, the output of DSPCA also satisfy the sparsity constraint \( m=4 \) (having 4 non-zero entries). Thus, the sparsity constraint refinement post-processing of DSPCA is not required.

Now, we consider the results of DSPCA when choosing the sparsity constraint \( k=5 \) in Table 1. The outputs of DSPCA

- the upper bound for the explained variance \( \text{var}_{\text{up}} := x_{\ast}^T A x_{\ast} \),
- the truncated principal component entries \( x_{ij} := x_{ij} \) for all \( i \neq I \) and \( x_{ij} := 0 \) for all \( i \in I \) - where \( I \) is the set of indexes of \( k \) absolutely largest entries of \( x_{ij} \),
- the lower bound for the explained variance \( \text{var}_{\text{low}} := x_{\ast}^T A x_{\ast} \).

• the upper bound for the explained variance \( \text{var}_{\text{up}} := x_{\ast}^T A x_{\ast} \),
• the truncated principal component entries \( x_{ij} := x_{ij} \) for all \( i \neq I \) and \( x_{ij} := 0 \) for all \( i \in I \) - where \( I \) is the set of indexes of \( k \) absolutely largest entries of \( x_{ij} \),
• the lower bound for the explained variance \( \text{var}_{\text{low}} := x_{\ast}^T A x_{\ast} \).

• the upper bound for the explained variance \( \text{var}_{\text{up}} := x_{\ast}^T A x_{\ast} \),
• the truncated principal component entries \( x_{ij} := x_{ij} \) for all \( i \neq I \) and \( x_{ij} := 0 \) for all \( i \in I \) - where \( I \) is the set of indexes of \( k \) absolutely largest entries of \( x_{ij} \),
• the lower bound for the explained variance \( \text{var}_{\text{low}} := x_{\ast}^T A x_{\ast} \).

To avoid the simulation randomness, the exact covariance matrix which is an infinity amount of data generated from the above model is used to compute principal components using the different approaches. The variance of the three underlying factors is nearly the same (290, 300 and 283.8, respectively). Since the first two are associated with four variables while the last one is associated with only two variables, \( V_1 \) and \( V_2 \) are almost equally important, and they are both significantly more important than \( V_3 \). In [12], the different approaches. The variance of the three above model is used to compute principal components using the different approaches. The variance of the three underlying factors is nearly the same (290, 300 and 283.8, respectively). Since the first two are associated with four variables while the last one is associated with only two variables, \( V_1 \) and \( V_2 \) are almost equally important, and they are both significantly more important than \( V_3 \). In [12], the first two principal components explain 99.6% of the total variance. In [16], by choosing the sparsity constraint \( m=4 \), DSPCA gives the same results as SPCA and SCOTLASS which are better than simple thresholding method. Moreover, the output of DSPCA also satisfy the sparsity constraint \( m=4 \) (having 4 non-zero entries). Thus, the sparsity constraint refinement post-processing of DSPCA is not required.

Now, we consider the results of DSPCA when choosing the sparsity constraint \( k=5 \) in Table 1. The outputs of DSPCA...
do not satisfy the sparsity constraint when both the first and second principal components have 6 non-zero entries. Hence, the sparsity constraint refinement post-processing is needed. With the same explained variance, the first two principal components of EDSPCA (with threshold $= 10^{-2}$, $\varepsilon = 10^{-2}$, $\text{var error} = 0.5\%$) satisfies the the sparsity constraint after solving 21 and 20 SDP problems, respectively.

B. Pit Props Data

The pit props data (consisting of 180 observations and 13 measured variables) was introduced in [18] and is another benchmark example used to test SPCA. All simple thresholding [2], SCoTLASS [9], SPCA [12], and DSPCA [16] have been tested on this data set. As reported in [12], SPCA performs better than SCoTLASS in the sense that it identifies principal components with 7, 4, 4, 1, 1, and 1 non-zero loadings respectively - while explaining nearly the same variance as SCoTLASS, the result SPCA of is much sparser; and better than simple thresholding in the sense that it explains more variance. As reported in [16], DSPCA performs better than SPCA in the sense that it identifies principal components with 6, 2, 3, 1, 1, and 1 non-zero loadings (with respect to sparsity constraint 5, 2, 2, 1, 1, and 1).

Here, we want to compare the results of EDSPCA - using the same sparsity constraint (5, 2, 2, 1, 1, and 1) - with those of DSPCA. The results are given in Table 2 with threshold $= 10^{-2}$, $\varepsilon = 10^{-10}$, $\text{var error} = (0.5/(2^{i-1}))\%$ for $i=1,…,6$. While explaining 76.0% variance - nearly the same as DSPCA (77.3%) - the first six principal components of EDSPCA satisfies the sparsity constraint after solving 5, 1, 16, 1, 1, and 1 SDP problems respectively. It is also remarkable that these results are better than ESPCA (75.9%) in [21]. However, we can see that there is an overlap between the first principal component and the third principal component on entry "ringbut". Hence, it is reasonable to think about a better sparsity constraint as 4, 2, 2, 1, 1, and 1. The outputs for this case are displayed in the left haft of Table 3, where EDSPCA also explains a large amount of the variance - 74.3% - by solving 12, 1, 1, 1, and 1 SDP problems respectively. Finally, with the less sparsity results (5, 2, 3, 1, 1, and 1) than DSPCA , the results of EDSPCA in the right haft of Table 3 explains more variance than DSPCA (78.5% compared with 77.3%). The first six principal components of EDSPCA satisfy the sparsity constraint after solving 5, 1, 15, 1, 1, and 1 SDP problems respectively.

Table I: The First Two Principal Components of DSPCA and EDSPCA with k=5 on the Artificial Data

<table>
<thead>
<tr>
<th>Variable</th>
<th>PC1</th>
<th>PC2</th>
<th>PC3</th>
<th>Var.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSPCA, PC1</td>
<td>-0.49</td>
<td>0.49</td>
<td>0.10</td>
<td>0.49</td>
</tr>
<tr>
<td>DSPCA, PC2</td>
<td>0</td>
<td>-0.49</td>
<td>0</td>
<td>0.49</td>
</tr>
<tr>
<td>EDSPCA, PC1</td>
<td>-0.49</td>
<td>0.49</td>
<td>0</td>
<td>0.49</td>
</tr>
<tr>
<td>EDSPCA, PC2</td>
<td>0</td>
<td>-0.49</td>
<td>0</td>
<td>0.49</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variable</th>
<th>PC1</th>
<th>PC2</th>
<th>PC3</th>
<th>Var.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSPCA, PC1</td>
<td>-0.56</td>
<td>0</td>
<td>-0.48</td>
<td>0.48</td>
</tr>
<tr>
<td>DSPCA, PC2</td>
<td>-0.58</td>
<td>0</td>
<td>-0.49</td>
<td>0.49</td>
</tr>
<tr>
<td>EDSPCA, PC1</td>
<td>-0.56</td>
<td>0</td>
<td>-0.48</td>
<td>0.48</td>
</tr>
<tr>
<td>EDSPCA, PC2</td>
<td>-0.58</td>
<td>0</td>
<td>-0.49</td>
<td>0.49</td>
</tr>
</tbody>
</table>

Table II: The First Three Principal Components of DSPCA and EDSPCA with Sparsity Constraint 5,2,2,1,1 and 1 on the Pit Props Data

<table>
<thead>
<tr>
<th>Methods</th>
<th>PC1</th>
<th>PC2</th>
<th>PC3</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSPCA</td>
<td>0.56</td>
<td>0.58</td>
<td>0.01</td>
</tr>
<tr>
<td>EDSPCA</td>
<td>0.56</td>
<td>0.58</td>
<td>0.01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sparsity</th>
<th>4-2-2</th>
<th>5-2-3</th>
</tr>
</thead>
<tbody>
<tr>
<td>varerr</td>
<td>PC1</td>
<td>PC2</td>
</tr>
<tr>
<td>topdiam</td>
<td>-0.54</td>
<td>0</td>
</tr>
<tr>
<td>length</td>
<td>-0.55</td>
<td>0</td>
</tr>
<tr>
<td>moist</td>
<td>0</td>
<td>0.71</td>
</tr>
<tr>
<td>testsg</td>
<td>0</td>
<td>0.71</td>
</tr>
<tr>
<td>ovensg</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ringtop</td>
<td>-0.26</td>
<td>0</td>
</tr>
<tr>
<td>ringbut</td>
<td>-0.01</td>
<td>0</td>
</tr>
<tr>
<td>bowmax</td>
<td>-0.37</td>
<td>0</td>
</tr>
<tr>
<td>bowdist</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>whorls</td>
<td>-0.36</td>
<td>0</td>
</tr>
<tr>
<td>clear</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>knots</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>diaknot</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

TABLE III: The First Three Principal Components of EDSPCA with Sparsity Constraint (4,2,2,1,1 and 1) and (5,3,1,1,1) on the Pit Props Data

Here, we want to compare the results of EDSPCA - using the same sparsity constraint (5, 2, 2, 1, 1, and 1) - with those of DSPCA. The results are given in Table 2 with threshold $= 10^{-2}$, $\varepsilon = 10^{-10}$, $\text{var error} = (0.5/(2^{i-1}))\%$ for $i=1,…,6$. While explaining 76.0% variance - nearly the same as DSPCA (77.3%) - the first six principal components of EDSPCA satisfies the sparsity constraint after solving 5, 1, 16, 1, 1, and 1 SDP problems respectively. It is also remarkable that these results are better than ESPCA (75.9%) in [21]. However, we can see that there is an overlap between the first principal component and the third principal component on entry "ringbut". Hence, it is reasonable to think about a better sparsity constraint as 4, 2, 2, 1, 1, and 1. The outputs for this case are displayed in the left haft of Table 3, where EDSPCA also explains a large amount of the variance - 74.3% - by solving 12, 1, 1, 1, and 1 SDP problems respectively. Finally, with the less sparsity results (5, 2, 3, 1, 1, and 1) than DSPCA , the results of EDSPCA in the right haft of Table 3 explains more variance than DSPCA (78.5% compared with 77.3%). The first six principal components of EDSPCA satisfy the sparsity constraint after solving 5, 1, 15, 1, 1, and 1 SDP problems respectively. Figure 1 shows the cumulative number of non-zero loadings and the cumulative explained variance of EDSPCA compared with DSPCA and SPCA.

![Cumulative cardinality and percentage of total variance explained versus number of principal components for SPCA, DSPCA and EDSPCA with sparsity constraint (4,2,2,1,1 and 1) and (5,3,1,1,1) on the pit props data.](image-url)
IV. CONCLUSIONS AND PERSPECTIVE

The application specific solution will be discussed elsewhere since we want to keep our method general. By re-weighted 1\_1 minimization technique and a bisection algorithm, EDSPCA attempted to add an post processing to re-weighted l 1 minimization technique and a bisection elsewhere since we want to keep our method general. By re-weighted function in (7) is also one of our priorities. Problems, see [15], [16]. Finally, finding an efficient re-weighted function in (7) is also one of our priorities.

ACKNOWLEDGMENT

The authors thank University of Pedagogy, Ho Chi Minh city, Viet Nam for financial support.

REFERENCES


He is currently leader of Quantitative Strategies and Analysis at VietFund Management. Concurrently, he is affiliate leader of Alternative Research at John von Neumann Institute - Vietnam National University HCM and a certified member of Professional Risk Managers' International Association. Dr. Duong was at Japan Advanced Institute of Science and Technology doing research on statistical machine learning in 2010.

Hung V. Nguyen was born in Vietnam, in 1981. Nguyen received B.S, Maths & Informatics (2003), MS. Computer Science (2007), all with high distinction from HCM University of Science. Since 2009, he has been a PhD candidate in Computer Vision of Japan Advanced Institute of Science and Technology.

He is currently a senior lecturer of Pedagogy University, Ho Chi Minh city, Vietnam. Concurrently, he is a research assistant at Japan Advanced Institute of Science and Technology.