공학석사 학위논문

이상치 탐지를 위한 적대적 사전 학습 알고리즘

Adversarial Dictionary Learning for Anomaly Detection

2020년 8월

서울대학교 대학원

기계공학부

백 종 혁

ABSTRACT

Adversarial Dictionary Learning for Anomaly Detection

by

Jonghyuk Baek

Department of Mechanical Engineering
Seoul National University

In this thesis, we propose a semi-supervised dictionary learning algorithm that learns representations of only non-outlier data. The presence of outliers in a dataset is a major drawback for dictionary learning, resulting in less than desirable performance in real-world applications. Our adversarial dictionary learning (ADL) algorithm exploits a supervision dataset composed of known outliers. The algorithm penalizes the dictionary expressing the known outliers well. Penalizing the known outliers makes dictionary learning robust to the outliers present in the dataset. The proposed method can handle highly corrupted dataset which cannot be effectively dealt with using conventional robust dictionary learning algorithms. We

empirically show the usefulness of our algorithm with extensive experiments on anomaly detection, using both synthetic univariate time-series data and multivariate point data.

Keywords: Sparse Representation, Dictionary Learning, Semi-Supervised Learning, Anomaly Detection

Student Number: 2018-21570

Contents

Abstract **i**

List of Tables **v**

List of Figures **vi**

1 Introduction **1**

1.1 Related Works 4

1.2 Contributions of This Thesis 5

1.3 Organization 6

2 Sparse Representation and Dictionary Learning **7**

2.1 Sparse Representation 7

2.1.1 Problem Definition of Sparse Representation 7

2.1.2 Sparse representation with l_0-norm regularization 10

2.1.3 Sparse representation with l_1-norm regularization 11

2.1.4 Sparse representation with l_p-norm regularization $(0 < p < 1)$ 12

2.2 Dictionary Learning 12

2.2.1 Problem Definition of Dictionary Learning 12
2.2.2 Dictionary Learning Methods 14

3 Adversarial Dictionary Learning **18**
3.1 Problem Formulation 18
3.2 Adversarial Loss 19
3.3 Optimization Algorithm 20

4 Experiments **25**
4.1 Data Description 26
4.1.1 Univariate Time-series Data 26
4.1.2 Multivariate Point Data 29
4.2 Evaluation Process 30
4.2.1 A Baseline of Anomaly Detection 30
4.2.2 ROC Curve and AUC 34
4.3 Experiment Setting 35
4.4 Results 36

5 Conclusion **43**

Bibliography **45**

국문초록 **50**

List of Tables

4.1 Table of multivariate point data properties. 29
4.2 Experiment settings. 36
4.3 AUC value result for univariate time-series data. 40
4.4 AUC value result for multivariate point data. 41

List of Figures

2.1 3D visualization of the intersection between the l_p-ball and the solution set of $Ax = b$. $p = 2$ (top left), $p = 1.5$ (top right), $p = 1$ (bottom left), and $p = 0.7$ (bottom right). $p \leq 1$ leads to a sparse solution. (image adopted from [16]) 9

4.1 Example of time-series used for training. $\sigma = 0.15$, # of anomaly 30, scale of anomaly 0.8, anomaly of length 3 shift. 27

4.2 Detailed plot of anomalies present in the time-series data; point anomaly (left) and sequence anomaly of length 3 (right). 28

4.3 Schematic diagram of the anomaly detection framework using the ADL. 31

4.4 Sliding window generation from time-series data. 33

4.5 Visualization of learned dictionary from sample univariate time-series data of $\sigma = 0.05$, anomaly of length-5 shift, number of anomaly 10, scale of anomaly 0.8. The outlier behavior of length-5 shift (marked as red dot circle) is not learned in ADL. 38

4.6 ROC curve of the anomaly detector using dictionaries presented in Figure 4.5. 39
4.7 Representative results from multivariate point dataset 42

1

Introduction

As the most representative methodology among linear representation methods, the sparse representation has attracted much attention from signal processing fields and has been applied to a variety of applications in recent years [1]. Originated from the theory of compressed sensing (CS), the concept of sparse representation is based on the rationale that if a signal is compressible, the original signal can be approximated using only a few measurement values [2]. From this rationale, the goal of sparse representation can be summarized as expressing a given signal as a linear combination of a small number of signals, taken from a reference database [3]. It has been demonstrated that many real-world signals and natural images can be represented with a sparse linear combination of some basis vectors. Further, the sparse representation of signal has been proven to be a powerful solution to a wide range of fields such as signal processing, computer vision, and machine learning, including tasks like image denoising [4], visual tracking [5], image classification [6], and anomaly detection [7].

The very first form of sparse representation method was to exploit a predesigned

set of transform functions, such as the fast Fourier transform (FFT) or wavelet transform and its variants. However, despite their simplicity and computational efficiency, the use of predesigned transform functions have an inherent problem that it is hard to manually design optimal transform function according to the signal characteristics. Their counterpart in learning scheme, the concept of learning transformation functions from the signal itself had emerged, which is called the dictionary learning method.

Dictionary learning aims to find a set of basis vectors (atoms), which is called a "dictionary" so that the given signals can be approximated as a sparse linear combination of basis elements [3]. A general framework for dictionary learning can be formulated as an optimization problem:

$$\min_{D\in\mathcal{C},\beta_i} \sum_{i=1}^{N} f(x_i - D\beta_i) + P(\lambda, \beta_i) \tag{1.0.1}$$

where $D \in \mathbb{R}^{n\times p}$ is a dictionary matrix and $\mathcal{C} = \{D = [d_1, d_2, \ldots, d_p] \mid d_i^T d_i \leq 1\}$. Here $d_i \in \mathbb{R}^n$ denotes the ith column of dictionary matrix, which is called "atom", and p denotes the number of atoms in a dictionary. N is the number of training samples, $x_i \in \mathbb{R}^n$ is the single n-dimensional sample from the dataset, and $\beta_i \in \mathbb{R}^p$ is the sparse representation corresponding to the ith sample. $f(x_i - D\beta_i)$ denotes a data fitting term. $P(\lambda, \beta_i)$ and λ are the regularization function and the regularization parameter for sparsity in representation β_i, respectively. The optimization problem jointly finds an optimal dictionary and sparse representation for given signals, which is achieved by minimizing the approximation error (first term) from the sparse representation while penalizing with sparsity in representation (second term). The scheme of learning basis vectors from signals can be also found in conventional linear representation methods such as Principle Component Analysis

(PCA) and Independent Component Analysis (ICA). However, unlike these methods dictionary learning does not impose the condition that the dictionary atoms be orthogonal, it allows much more flexibility in signal modeling.

Despite the usefulness in signal representation, dictionary learning usually suffers from the outliers present in the dataset. In fact, training a dictionary needs a large amount of data in practice. However, as the amount of training data grows larger, it becomes almost impossible to get an outlier-free dataset because there exists a limitation on human manual data labeling. The outliers present in the dataset affect the expressive ability of the learned dictionary if the presence of outliers is not regarded in advance in the learning procedure.

A dictionary corrupted with the outliers (i.e. a dictionary that also has an expressive ability on outliers) is not desirable in practical applications. For example, anomaly detection using sparse representation requires a dictionary well adapted to the non-anomalous signals. Then, the anomaly detection can be conducted by observing the residuals which cannot be reconstructed well using a given dictionary with the designated sparsity level in representation. However, as mentioned above, expressive ability on outliers (anomalies) from the corrupted dictionary make the chance of approximating even anomalous signals, while keeping the sparsity level in representation low. In another application of the Background Subtraction (BGS) for foreground segmentation, the dictionary learning can be exploited to model the background of the video sequence. However, video records often contain both background regions and foreground pixels; which is an outlier signal we do not want to model. This can lead to an inaccurate background model which results in poor foreground segmentation performance.

1.1 Related Works

Various attempts have been made to make dictionary learning robust to the outliers present in the dataset, and they can be grouped into the set of algorithms called the *robust dictionary learning*. Authors in [8] exploits l_1-norm for both data fitting term and sparsity penalty term in optimization (see equation 1.0.1), which is known to be robust for non-Gaussian noise contamination. This formulation improves either computational efficiency on sparse representation step and robustness on outliers, along with the guarantee of the existence of the global optima. [9] further enhances the robustness of dictionary learning using capped l_1-norm on data fitting term, which saturates to a constant value when the scale of l_1-norm is over the designated threshold. The non-convexness inherited from the capped l_1-norm is addressed and an efficient algorithm that finds local optimal solutions is suggested. Authors in [10] use the same modification on objective function as [9] with l_1-norm but they had a different approach solving it, a modified version of K-SVD, the representative of l_0-norm regularized dictionary learning algorithms. Authors in [11] presented a new data fitting loss called Gaussian fidelity, which guarantees a stable solution in the presence of outliers. Many other works proposed robust dictionary learning algorithms based on modification on linear approximation loss [12][13].

However, the approaches of merely modifying the data fitting loss function f to be robust can fail when the outliers are abundant in a dataset and they are not well distinguishable from the inlier ones just using the l_p-norms. Moreover, they do not exploit any additional domain-specific information of data, such as the representative patterns of frequently occurring outliers (e.g. semiconductor defect pattern or ECG anomaly pattern) or some supervision results from the human

practitioner, even when they are available to use.

1.2 Contributions of This Thesis

In this thesis, we propose a novel dictionary learning algorithm that rejects outliers using a semi-supervised learning scheme. The semi-supervised learning we refer to here is the learning setting where most data are unlabeled and a small portion of data is labeled.

Different from the existing robust dictionary algorithms with the same purpose, we exploit additional information of outliers provided, not just modifying the loss functions to be robust on some misbehaving data. To be more specific, given a small set of data labeled as an outlier we add an additional loss function to the conventional dictionary learning problem, called an *adversarial loss*. The adversarial loss takes the outlier labeled sample set as an input and penalizes the expressive ability of the dictionary on the outliers, acting as a barrier function in the optimization process.

Our algorithm has an advantage in that we can explicitly provide information on misbehaving data that we do not want the dictionary to express it. In many cases, the definition of the term *outlier* itself is ambiguous. Even intermittent data and data that deviate much from the most frequently observed data can be also considered as inliers, depending on the needs and choices of the practitioner. So it is necessary to have a "reference" which signal to learn and which to avoid. Our formulation with a new loss function tackles down this need, at the same time improving the robustness of the dictionary learning.

Although it is not the first time to exploit the scheme of semi-supervised learning in dictionary learning [14][15][13], the concept of using the supervision data as

a reference for misbehaving signals to improve the robustness of the dictionary learning, has not been proposed yet.

1.3 Organization

In Chapter 2, we review the classic framework of dictionary learning for sparse representation. First, we explain the mathematical formulations of the sparse representation problem. The algorithms dealing with each formulation will be presented along with. The following shows a detailed outline of the dictionary learning problem, including representative formulations and details in methods dealing with.

In chapter 3, we present the Adversarial Dictionary Learning (ADL), the robust dictionary learning algorithm which uses the semi-supervised learning scheme. Starting from a new formulation of the problem, we explain the philosophy of our method and show how our problem differs from the conventional approach. The optimization procedure for the problem and the resulting algorithm will then provided, with an explanation of each process.

Chapter 4 reports the experiments and analysis of the results. We test our algorithm with data of two types: synthetic univariate time-series data and natural multidimensional point data. The resulting dictionaries are evaluated with anomaly (outlier) detection performance, which is highly dependant on the robustness of a dictionary learning algorithm on outliers. The anomaly detection performance is quantified using the AUC of the receiver operating characteristic (ROC) curve.

Finally, we conclude our thesis in Chapter 5 with a summary of our research and a discussion on the main results.

2

Sparse Representation and Dictionary Learning

In this chapter, we review the dictionary learning for sparse representation. We first begin with the frameworks of sparse representation problems in Section 2.1. Then those of the dictionary learning problems will be introduced subsequently. Representative algorithms for each problem will be briefly introduced either.

2.1 Sparse Representation

2.1.1 Problem Definition of Sparse Representation

The general framework of sparse representation is to represent the observed signal with the linear combination of some given samples, which is called atoms. Then, the coefficients of the linear combination can be retrieved and used as a sparse representation solution.

Let $D = [d_1, d_2, \ldots, d_p] \in \mathbb{R}^{n \times p}$ where $d_i \in \mathbb{R}^n$ $(n < p)$ be a set of atoms,

or an over-complete dictionary matrix. The problem of representing an observed signal $x \in \mathbb{R}^n$ with linear combination of atoms can be expressed as the following equation:

$$x = d_1\beta_1 + d_2\beta_2 + \cdots + d_p\beta_p, \tag{2.1.1}$$

where $\beta_i \in \mathbb{R}$ is the coefficient for i^{th} atom of dictionary. Letting $\beta = [\beta_1, \beta_2, \ldots, \beta_p]^T$, above can be rewritten into compact form,

$$x = D\beta. \tag{2.1.2}$$

However, the equation cannot be solved alone due to the over-completeness of the dictionary ($n < p$). The problem represents an underdetermined linear system of equations, which has more unknowns than equations. Thus the problem of finding representation β is ill-posed with no unique solution. To alleviate the ill-posedness, imposing some regularization on solution β can be a practical solution. In dictionary learning, the sparsity of representation is used for the regularization.

After defining a proper "desirability" measure for a solution $J : \mathbb{R}^n \to \mathbb{R}$, the problem of finding a solution under the regularization can be formulated as:

$$\min_{\beta} J(\beta) \quad \text{s.t.} \quad x = D\beta. \tag{2.1.3}$$

The regularization function J governs the characteristic of the resulting solution. The typical choice for the sparsity in the solution is to use l_p-norm with $0 \leq p \leq 1$. Figure 2.1 presents the simple geometric example showing the effect of l_p-norm regularization on the sparsity of the solution.

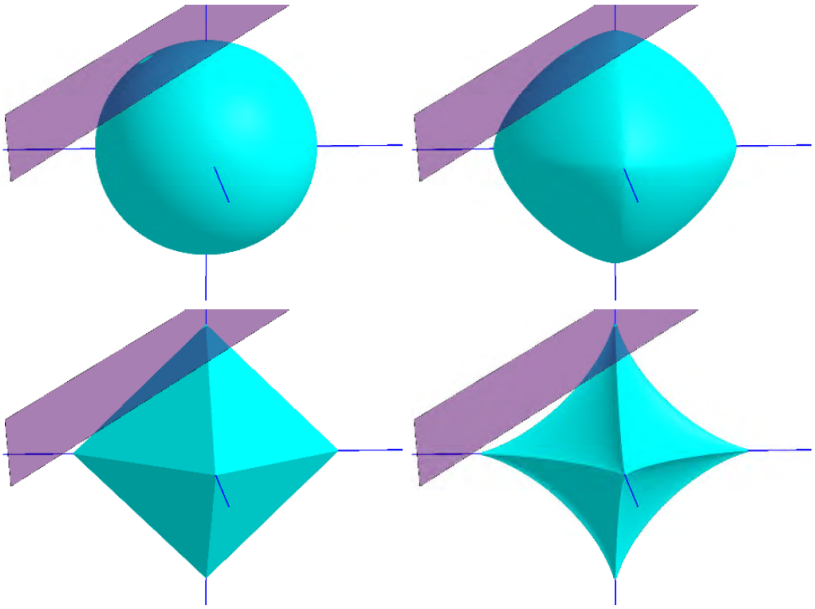


Figure 2.1: 3D visualization of the intersection between the l_p-ball and the solution set of $Ax = b$. $p = 2$ (top left), $p = 1.5$ (top right), $p = 1$ (bottom left), and $p = 0.7$ (bottom right). $p \leq 1$ leads to a sparse solution. (image adopted from [16])

2.1.2 Sparse representation with l_0-norm regularization

With l_0-norm regularization, the equation 2.1.3 is converted to the following optimization problem:

$$\beta^* = \arg\min_{\beta} \|\beta\|_0 \quad \text{s.t.} \quad x = D\beta, \tag{2.1.4}$$

where $\|\cdot\|_0$ counts the number of nonzero elements in a vector. If just k atoms participate in signal representation, the problem can be converted into another equivalent form:

$$x = D\beta \quad \text{s.t.} \quad \|\beta\|_0 \leq k, \tag{2.1.5}$$

which is called the k-sparse approximation problem. Here k denotes positive integer number. Considering the small and bounded noise present in observed signal, we can modify the model of equation 2.1.2 to:

$$x = D\beta + s, \tag{2.1.6}$$

where $s \in \mathbb{R}^n$ denotes a bounded energy noise term, i.e. $\|s\|_2 < \epsilon$. The approximate solution of equation 2.1.4 and 2.1.5 can be obtained with following relaxed optimization problems:

$$\beta^* = \arg\min_{\beta} \|\beta\|_0 \quad \text{s.t.} \quad \|x - D\beta\|_2^2 \leq \epsilon, \tag{2.1.7}$$

or

$$\beta^* = \arg\min_{\beta} \|x - D\beta\|_2^2 \quad \text{s.t.} \quad \|\beta\|_0 \leq k. \tag{2.1.8}$$

Finally, these problems are converted into the most general form of sparse representation problem by Lagrange multiplier theorem.

$$\beta^* = \arg\min_{\beta} \|x - D\beta\|_2^2 + \lambda\|\beta\|_0. \tag{2.1.9}$$

Although the sparse representation problem with l_0-norm regularization makes the solution of linear approximation problem to be sparse explicitly, the problem suffers from the NP-hardness inherited from the l_0-norm [17][18]. The representative algorithm dealing with l_0-norm regularized linear approximation problem is a group of Greedy algorithms consists of Matching Pursuit (MP) algorithm and Orthogonal Matching Pursuit (OMP) algorithm. Instead of directly solving the optimization problem, they obtain an approximate solution of the equation 2.1.4.

2.1.3 Sparse representation with l_1-norm regularization

One common approach dealing with the NP-hardness of the l_0-norm regularized linear approximation problem is to relax the l_0-norm constraint to the l_1-norm constraint.

$$\beta^* = \arg\min_{\beta} \|\beta\|_1 \quad \text{s.t.} \quad x = D\beta, \tag{2.1.10}$$

or

$$x = D\beta \quad \text{s.t.} \quad \|\beta\|_1 \leq \tau. \tag{2.1.11}$$

Here τ denotes small positive scalar value. It has been revealed in literature [19] that l_1-norm regularization also gives equal solution to the solution from the l_0-norm with full probability, if the solution sought for is sparse enough. The optimization problems can be reformulated as the same way in section 2.1.2 assuming bounded energy observation noise:

$$\beta^* = \arg\min_{\beta} \|\beta\|_1 \quad \text{s.t.} \quad \|x - D\beta\|_2^2 \leq \epsilon, \tag{2.1.12}$$

$$\beta^* = \arg\min_{\beta} \|x - D\beta\|_2^2 \quad \text{s.t.} \quad \|\beta\|_1 \leq \tau, \tag{2.1.13}$$

or

$$\beta^* = \arg\min_{\beta} \|x - D\beta\|_2^2 + \lambda\|\beta\|_1. \tag{2.1.14}$$

This formulation is called the lasso problem [20], which is a well-known problem in statistics. As the optimization problem is convex, there exist various algorithms that guarantee a globally optimal solution in polynomial times, such as Gradient Projection (GP), Homotopy, Iterative Shrinkage-Thresholding (IST), and Alternating Direction Method (ADM) and several more, along with heuristic greedy algorithms including Least Angle Regression (LARS) [21] and OMP.

2.1.4 Sparse representation with l_p-norm regularization $(0 < p < 1)$

As in other cases, the optimization problem for l_p-norm $(0 < p < 1)$ regularized linear approximation problem is considered as:

$$\beta^* = \arg\min_{\beta} \|\beta\|_p^p \quad \text{s.t.} \quad \|x - D\beta\|_2^2 \leq \epsilon, \tag{2.1.15}$$

or

$$\beta^* = \arg\min_{\beta} \|x - D\beta\|_2^2 + \lambda\|\beta\|_p^p. \tag{2.1.16}$$

The main drawback of the l_p-norm regularized optimization problem is the non-convexity of the problem. Although there is no guarantee for the existence of global optima and the convergence property is hard to analyze, there are several algorithms which empirically works in practice. [22]. Iteratively Reweighted l_1 minimization (IRL1), Iteratively Reweighted Least Squares (IRLS), and Iteratively Thresholding Method (ITM) are the representatives.

2.2 Dictionary Learning

2.2.1 Problem Definition of Dictionary Learning

In the previous section, we assumed the dictionary; the key ingredient for obtaining a sparse representation solution, is given as a constant. Dictionary learning

aims to find a faithful and effective dictionary that well approximates a specific set of signals.

From the notations of the literature [23], given a set of data $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$, the general framework of dictionary learning is expressed as minimization of the following optimization function:

$$\min_{D \in \mathcal{C}, \beta_i} \sum_{i=1}^{N} f(x_i - D\beta_i) + P(\lambda, \beta_i). \tag{2.2.17}$$

$\mathcal{C}$ is the convex constraint on dictionary matrix which purpose is to prevent dictionary atoms from diverging, defined as $\mathcal{C} = \{D = [d_1, d_2, \ldots, d_p] \in \mathbb{R}^{n \times p} \mid d_i^T d_i \leq 1\}$. Here p is the number of atoms (i.e. d_i) in a dictionary. $\beta_i \in \mathbb{R}^p$ denotes sparse representation (or sparse code) of the ith signal $x_i \in \mathbb{R}^n$. p can be seen as either the number of atoms and the dimension of the sparse representation. f is a function penalizing signal approximation error and P is a regularization function that controls the degree of sparsity of the representation.

The problem jointly finds a dictionary matrix D whose atoms well represent the given set of data $\mathcal{X}$ and corresponding sparse representation set β_i. The choice for f and P depends greatly on the purpose of the user. The function itself is identical to the augmented version of the problem in section 2.1 if we set f as $\|\cdot\|_2^2$ and P as $\lambda\|\cdot\|_p$ $(0 \leq p \leq 1)$. Letting p as zero and setting f as l_2-norm is the most common setting for dictionary learning in many works of literature. As it is not possible to introduce all dictionary methods for all formulations, we will introduce a few representative methods for the most common problem setting.

2.2.2 Dictionary Learning Methods

The most common form of dictionary learning problem can be written as:

$$\min_{D\in\mathcal{C},\beta_i} \sum_{i=1}^{N} \|x_i - D\beta_i\|_2^2 + \lambda\|\beta_i\|_0. \tag{2.2.18}$$

As the problem is NP-hard due to l_0-norm, it cannot be solved directly. There are two mainstream methods dealing with this problem, one is the greedy strategy and the other is the convex relaxation.

Greedy Strategy

The representative method of greedy strategy is K-SVD [4]. The K-SVD solves the following optimization problem:

$$\begin{aligned} &\min_{D,B} \frac{1}{2}\|X - DB\|_F^2 \\ &\text{s.t.} \quad \|\beta_i\|_0 \leq k, \text{ for } i = 1, 2, \ldots, N. \end{aligned} \tag{2.2.19}$$

Here $X = [x_1, x_2, \ldots, x_N] \in \mathbb{R}^{n\times N}$ is the data matrix whose column represents a single sample and $B = [\beta_1, \beta_2, \ldots, \beta_N] \in \mathbb{R}^{p\times N}$ is a matrix form of sparse representation over the entire dataset. k is a positive integer value that constrains the degree of sparsity of the representation.

K-SVD algorithm solves the problem by iteratively solving the decomposed problems, one is sparse coding step and dictionary update step. The sparse coding step is conducted by fixing the dictionary matrix in the above problem and is formulated as follows:

$$\begin{aligned} &\min_{\beta_i\in\mathbb{R}^p} \|x_i - D\beta_i\|_2^2 \quad \text{s.t.} \quad \|\beta_i\|_0 \leq k, \\ &\text{for} \quad i = 1, 2, \ldots, N. \end{aligned} \tag{2.2.20}$$

The approximate solution for this subproblem can be obtained by the pursuit algorithms presented in section 2.1.2 if k is small enough. After obtaining sparse representation with a fixed dictionary, dictionary update problem while fixing the representation is formulated as follows:

$$\min_{D} \|X - DB\|_F^2. \tag{2.2.21}$$

The solution is straightforward, $D^* = XB^\dagger = XB^T(BB^T)^{-1}$. However, the computational complexity of the problem is $O(n^3)$ due to the inverse operation. K-SVD improves efficiency by updating each column of the dictionary while letting other columns as a constant vector. After freezing $p-1$ dictionary columns and letting only ith column as an optimization variable, the cost function can be reformulated as:

$$\begin{aligned} \|X - DB\|_F^2 &= \|X - \sum_{j=1}^{p} d_j \beta_T^j\|_F^2 \\ &= \|(X - \sum_{j \neq i} d_j \beta_T^j) - d_i \beta_T^i\|_F^2 \\ &= \|E_i - d_i \beta_T^i\|_F^2. \end{aligned} \tag{2.2.22}$$

Here β_T^j denotes jth row of matrix B. Then the optimal d_i and β_T^i for the cost function are obtained by solving SVD for the error matrix E_i, finding the closest rank-1 matrix approximating E_i. To not violate sparsity constraint $\|\beta_i\|_0 \leq k$ during the update, only vector elements of nonzero value in β_T^i are updated, letting zero value elements in the previous step unchanged. Restricting E_i by choosing only the columns affected, the smaller matrix E_i^R is obtained and is decomposed as $E_i^R = U\Delta V^T$. Then d_i and β_T^i are updated to the first column of U and the first column of V expanded to the original size, respectively. The procedure is repeated for $i = 1, 2, \ldots, p$.

Convex Relaxation

Another approach dealing with the equation 2.2.18 is to relax the l_0-norm regularization to l_1-norm regularization, which either induces sparsity on representation but also convex. The relaxed dictionary learning problem is written as:

$$\min_{D\in\mathcal{C},B} \|X - DB\|_F^2 + \lambda\|B\|_1. \tag{2.2.23}$$

The problem itself is not convex but it can be if we separate optimization variables into D and B independently. The practical approach is to iteratively update D and B while letting another variable fixed.

As in greedy strategy, the problem is decomposed into two subproblems:

$$\begin{gathered}\min_{\beta_i\in\mathbb{R}^p} \|x_i - D\beta_i\|_2^2 + \lambda\|\beta_i\|_1 \\ \text{for} \quad i = 1, 2, \ldots, N,\end{gathered} \tag{2.2.24}$$

and

$$\min_{D\in\mathcal{C}} \sum_{i=1}^{N} \|x_i - D\beta_i\|_2^2. \tag{2.2.25}$$

The first problem is a linear approximation problem with l_1-norm regularization, of which solvers are introduced in section 2.1.3. The second problem can be further decomposed into convex vector optimization problems:

$$\begin{gathered}\min_{D(k,:)} \sum_{i=1}^{N} (x_{ik} - D(k,:)\beta_i)^2 \\ \text{for} \quad k = 1, 2, \ldots, n.\end{gathered} \tag{2.2.26}$$

Here $D(k,:)$ denotes kth row of D and $x_{ik} \in \mathbb{R}$ denotes kth element of ith data vector x_i. The convex constraint $\mathcal{C}$ on dictionary matrix D in the original problem can be resolved by normalizing each column of D when the norm is larger than

one after the dictionary update. The formulation for the dictionary update step is almost the same as that of greedy strategy, but as the sparsity constraint is not defined straightforwardly in this setup, the heuristic approach used in greedy strategy cannot be applied.

Our method that will be introduced in the next chapter focuses on the convex relaxation approach over the greedy strategy because the problem can be easily decomposed into the convex vector optimization problems and it has much more flexibility in modification than the greedy strategy.

3

Adversarial Dictionary Learning

This chapter presents the Adversarial Dictionary Learning; a robust dictionary learning algorithm using supervision data. The philosophy of our algorithm can be summarized as: obtain a dictionary that is robust to the outliers exploiting data instances designated as *not-to-learn* examples. We will first introduce the problem statement of our algorithm. More details about Then the optimization framework and the algorithm details for the problem will be followed.

3.1 Problem Formulation

Given a dataset $\mathcal{X} = \{x_1, x_2, \dots x_N\}$ (training set) and a set of data labeled as an outlier $\mathcal{X}^- = \{x_1^-, x_2^-, \dots x_M^-\}$ (supervision set) where $x_i, x_j^- \in \mathbb{R}^n$, we propose the following optimization problem:

$$\min_{D \in \mathcal{C}} \{ \min_{\beta_i \in \mathbb{R}^p} \frac{1}{N} \sum_{i=1}^{N} (\|x_i - D\beta_i\|_1 + \lambda \|\beta_i\|_1) - w \cdot \mathcal{G}(D, X^-) \}. \tag{3.1.1}$$

Here $X^- = [x_1^-, x_2^-, \ldots, x_M^-] \in \mathbb{R}^{n \times M}$ is a matrix form of labeled outlier dataset. $D \in \mathbb{R}^{n \times p}$ is an overcomplete dictionary, i.e. $p > n$. $\lambda \in \mathbb{R}$ is a regularization parameter that controls the sparsity level in representation. $\mathcal{G}$ is an adversarial loss, which is the main component in our formulation. A detailed explanation of $\mathcal{G}$ will be given in the next section. $w \in \mathbb{R}$ is a weight that governs the effect of the adversarial loss in optimization. From this optimization problem, we can see that if we set w as zero, the problem becomes l_1-norm based robust dictionary learning presented in [8].

For the term *label* we refer to, we cover both cases the outlier label is given to each data sample (i.e. $\mathbb{R}^n \to [0, 1]$) and the label is given each dimension of data (i.e. $\mathbb{R}^n \to [0, 1]^n$). The examples of the latter case include abnormal pixel regions in image data and anomaly in time series data. We assume the label information for each single data in $\mathcal{X}^-$ is given.

3.2 Adversarial Loss

The purpose of the adversarial loss is to constrain the expressive ability of the optimized dictionary, not to obtain adequate sparse representation from even outlier data. The adversarial loss gives penalty when sparse representation obtained with dictionary fits given data well under appropriate sparsity level. To do so, we construct the adversarial loss as follows:

$$\mathcal{G}(D, X^-) = \sum_{k=1}^{n} \{ \frac{1}{M} \sum_{j=1}^{M} \log[(x_{jk}^- - D(k,:)\gamma_j^*)^2] \}, \tag{3.2.2}$$

where

$$\gamma_j^* = \underset{\gamma_j \in \mathbb{R}^p}{\arg\min} \|x_j^- - D\gamma_j\|_1 + \lambda \|\gamma_j\|_1. \tag{3.2.3}$$

Here M is the number of outlier labeled data. λ in equation 3.2.3 is the same value as in equation 3.1.1. The loss is designed to maximize the reconstruction error from the best sparse representation obtained, only for the outlier data. The logarithm enclosing the reconstruction error term acts as a barrier function that prevents the dictionary from expressing outlier data. Further, the gradient rapidly saturates as the reconstruction error for the outlier goes higher, lessening the effect of the function in optimization when the error is moderately high.

For the case where the outlier label is given with respect to each data's dimension, the adversarial loss can be further modified as:

$$\mathcal{G}(D, X^-) = \sum_{k=1}^{n} \{ \frac{1}{L_k^I} \sum_{j=1}^{M} I(x_{jk}^-) \cdot \log[(x_{jk}^- - D(k,:)\gamma_j^*)^2] \}. \tag{3.2.4}$$

Here, $I : \mathbb{R} \to [0, 1]$ is an indicator function, $I(x_{jk}^-) = 1$ when x_{jk}^- contains outlier and 0 otherwise. L_k^I is sum of the number of outlier spots in X^-, seen from data dimension k, i.e. $L_k^I = \sum_{j=1}^{M} I(x_{jk}^-)$. The indicator function has an essential role in preventing adversarial loss from attacking even inlier data patterns. We will keep this formulation and in the case where the outlier label is given w.r.t each single data, we set $L_k^I = M$ and $I(x_{jk}^-) = 1$ for all $k = 1, 2, \ldots, n$ if x_j^- is labeled as an outlier.

3.3 Optimization Algorithm

For optimization, we adopt the same strategy as the traditional dictionary learning method: the convex relaxation approach. We solve and update D and $\{B, C\}$ alternatively until the convergence. C denotes the matrix version of optimal sparse

code γ of outlier data. The sparse coding step is formulated as:

$$\beta_i^* = \arg\min_{\beta_i \in \mathbb{R}^p} \|x_i - D\beta_i\|_1 + \|\beta_i\|_1 \tag{3.3.5}$$

$$\text{for} \quad i = 1, 2, \ldots, N, \tag{3.3.6}$$

and

$$\gamma_j^* = \arg\min_{\gamma_j \in \mathbb{R}^p} \|x_j^- - D\gamma_j\|_1 + \|\gamma_j\|_1 \tag{3.3.7}$$

$$\text{for} \quad j = 1, 2, \ldots, M. \tag{3.3.8}$$

As the dictionary is fixed and the loss functions are defined separately with respect to two datasets, the sparse representation for each can be obtained by two independent optimization problems.

Using the approach presented in [8], the problem can be re-wrapped into linear l_1 approximation problem:

$$\beta_i^* = \arg\min_{\beta_i \in \mathbb{R}^p} \left\| \begin{bmatrix} x \\ 0 \end{bmatrix} - \begin{bmatrix} D \\ \lambda I \end{bmatrix} \beta_i \right\|_1 . \tag{3.3.9}$$

This problem shows an overdetermined linear system and it is guaranteed that this problem has a global optima [24]. The problem can be converted into linear programming (LP) and easily solved. It is the same for the case of γ_j.

The dictionary update step is formulated as:

$$\begin{aligned} &\min_{D \in \mathcal{C}} \frac{1}{N} \|X - DB\|_1 \\ &\quad - w \cdot \sum_{k=1}^{n} \{ \frac{1}{L_k^I} \sum_{j=1}^{M} I(x_{jk}^-) \cdot \log[(x_{jk}^- - D(k,:)\gamma_j)^2] \}. \end{aligned} \tag{3.3.10}$$

The optimization problem 3.3.10 can be further decomposed as we did in dictionary learning problem with convex relaxation approach so we can update each

row of dictionary $D(k,:)$ independently:

$$\begin{aligned}\min_{D(k,:)\in\mathbb{R}^p} & \frac{1}{N}\sum_{i=1}^{N}|x_{ik}-D(k,:)\beta_i| \\ & - w\cdot\frac{1}{L_k^I}\sum_{j=1}^{M} I(x_{jk}^-)\cdot\log[(x_{jk}^- - D(k,:)\gamma_j)^2] \\ & \text{for} \quad k=1,2,\ldots,n.\end{aligned} \tag{3.3.11}$$

To deal with non-differentiability of the problem, inspired by [23] we introduce the scheme of iterative reweighted least-squares (IRLS) [25]. The IRLS scheme can be implemented by adding weight term and changing the absolute value function in the first term into the square function. The modified subproblem can be written as:

$$\begin{aligned}\min_{D(k,:)\in\mathbb{R}^p} & \frac{1}{N}\sum_{i=1}^{N} w_i^k(x_{ik}-D(k,:)\beta_i)^2 \\ & - w\cdot\frac{1}{L_k^I}\sum_{j=1}^{M} I(x_{jk}^-)\cdot\log[(x_{jk}^- - D(k,:)\gamma_j)^2],\end{aligned} \tag{3.3.12}$$

where

$$w_i^k = \frac{1}{\sqrt{(x_{ik}-D(k,:)\beta_i)^2+\delta}}. \tag{3.3.13}$$

Here δ is a small positive value preventing weight diverging when $(x_{ik}-D(k,:)\beta_i)$ goes to zero. The optimization is done by updating $D(k,:)$ and w_i^k alternatively, setting each other fixed until the convergence. The entire optimization process can be written as the following pseudo-code. We will use the notation $g_k(D,X^-)$ for the second term in equation 3.3.12 for simplicity.

We can see that the problem 3.3.12 is a vector optimization problem and is locally convex. Due to the sum of the negative log terms in the objective, it can be seen that the optimization variable domain is divided by infinite-loss hyperplanes.

For each divided region, the problem is convex and there exists a local optima. We exploit the quasi-newton method, BFGS for the dictionary update.

As the infinite-loss hyperplane generated from the outlier data make the solution to be stuck in local minima too early in the dictionary learning procedure, we do not use all the outlier labeled data at once, but we randomly sample a small amount of outlier data at the start of each iteration and use them for dictionary update. We denote this set of data as $X'^{-} = [x'^{-}_1, x'^{-}_2, \ldots, x'^{-}_{M'}] \in \mathbb{R}^{n \times M'}$ where M' is an integer number less than the total sample number in supervision dataset M.

The interesting intuition for the problem 3.3.12 is that this can be seen as a log-barrier version of the inequality constrained convex optimization problem, constraining reconstruction error for outlier data not to be zero. The equivalent problem can be written as:

$$\begin{aligned} &\min_{D(k,:)} \frac{1}{N} \sum_{i=1}^{N} |x_{ik} - D(k,:)\beta_i| \\ &\text{s.t.} \quad (x^{-}_{jk} - D(k,:)\gamma_j)^2 \geq 0 \\ &\qquad \text{for all} \quad j,k \quad x^{-}_{jk} \text{ contains outlier.} \end{aligned} \tag{3.3.14}$$

Algorithm Adversarial Dictionary Learning

Input: $X = [x_1, x_2, \ldots, x_N] \in \mathbb{R}^{n \times N}$, $X^- = [x_1^-, x_2^-, \ldots, x_M^-] \in \mathbb{R}^{n \times M}$, adversarial weight $w > 0$, outlier sampling number $M'(< M)$, regularization parameter $\lambda > 0$

Output: dictionary D, sparse code B

Initialize D with a random matrix with $d_i^T d_i \leq 1$
Initialize $w_i^k = 1$ for all $i = 1, 2, \ldots, N$ and $k = 1, 2, \ldots, n$
repeat
  (1) Sparse coding
  sample M' samples from X^- and construct X'^-
  for $i \in \{1, 2, \ldots, N\}$ **do**
    $\beta_i \leftarrow \arg\min_{\beta_i \in \mathbb{R}^p} \|x_i - D\beta_i\|_1 + \lambda\|\beta_i\|_1$
  end for
  for $j \in \{1, 2, \ldots, M'\}$ **do**
    $\gamma_j \leftarrow \arg\min_{\gamma_j \in \mathbb{R}^p} \|x'^-_j - D\gamma_j\|_1 + \lambda\|\gamma_j\|_1$
  end for
  (2) Dictionary update
  for $k \in \{1, 2, \ldots, n\}$ **do**
    repeat
      $D(k,:) \leftarrow \arg\min_{D(k,:) \in \mathbb{R}^p} \frac{1}{N} \sum_{i=1}^{N} w_i^k (x_{ik} - D(k,:)\beta_i)^2 - w \cdot g_k(D, X^-)$
      update $w_i^k (i = 1, 2, \ldots, N)$
    until convergence
  end for
until convergence
return D, B

4

Experiments

In this chapter, we apply our dictionary learning algorithm to anomaly detection (or outlier detection) problems. The task of anomaly detection can be defined as *determining data instances that stand out as being dissimilar to all others* [26].

If the dictionary is trained to represent even outlier data, the performance of the anomaly detection task will be greatly degraded. Typically, anomaly detection with dictionary learning is conducted in a supervised way, i.e. dictionary is learned from the outlier-free dataset. However, as mentioned in the introduction, in a real environment usually it is almost impossible to get an outlier-free dataset. Therefore, a robust dictionary learning method that learns only inlier data behavior is necessary for the anomaly detection task and how robustly the dictionary is learned is directly related to the performance of the detection.

This chapter aims to show the usefulness and the robustness of our algorithm by presenting qualitative and quantitative results of the anomaly detection task. We conducted experiments applying various dictionary learning algorithms along

with the proposed method. For the greedy strategy based dictionary learning algorithm, we took l_1-K-SVD [10], which is the robust version of the existing K-SVD method. For the convex relaxation based approach, we took the basic approach presented in section 2.2.2 and l_1 approximation based robust dictionary learning algorithm (we will call this RDL) [8]. Note that if we set the adversarial weight to zero, our method is identical to the method proposed in [8].

4.1 Data Description

Data we used in our experiment can be categorized into two groups, the univariate time-series data and the multivariate point data.

4.1.1 Univariate Time-series Data

The univariate time-series data is a one-dimensional point data that changes and is acquired over time. Inspired by the Yahoo Webscope S5 dataset [27], we generated synthetic univariate time-series consist of a trend, seasonality, white noise, and point or sequence anomaly. We controlled the number, length, and scale of the anomaly and the level of white noise and examined the effects on the dictionary learning and anomaly detection performance.

The static settings are; time-series length $N = 1500$, trend $T(t) = 0.3{*}\sin\frac{t}{2*N}2\pi$, and seasonality $S(t) = 0.3 * \sin\frac{t}{30}2\pi + 0.06 * (\sin\frac{t}{20}2\pi + \sin\frac{t}{12}2\pi)$. The resulting time series can be written as follows:

$$\begin{aligned} U(t) &= T(t) + S(t) + \epsilon + s \\ \text{where} \quad & t = 1, 2, \ldots, N \end{aligned} \tag{4.1.1}$$

where $\epsilon \sim N(0, \sigma)$ is Gaussian white noise and $s \sim N(c, \frac{c}{100}) * [1, -1]/\sqrt{l}$ is an

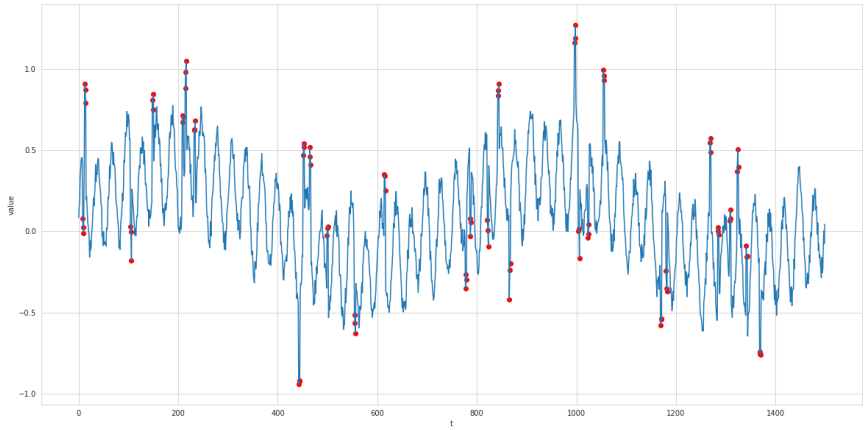


Figure 4.1: Example of time-series used for training. $\sigma = 0.15$, # of anomaly 30, scale of anomaly 0.8, anomaly of length 3 shift.

anomalous residual for the outlier data point generation, which is added at a random time with a designated length l; length 1 for the point anomaly and sequence (shift) anomaly otherwise. Here c denotes the scale of the anomalous residual.

For the controlled variables, we set white noise $\sigma = [0.050, 0.100, 0.150]$, scale of anomaly $c = [0.4, 0.6, 0.8]$, number of anomalies present in time-series $[10, 30, 50]$ and length of anomaly $l = [1, 3, 5]$.

In the dictionary learning procedure, as like the previous literature [7][28] we use a sliding window of designated length for the training data to take account of the temporal behavior of the time-series. A detailed explanation of the sliding window will be provided in chapter 4.2. We use a sliding window of length 20 and in our experiment. Now we have a 20-dimensional dataset that has both contaminated and uncontaminated data instance.

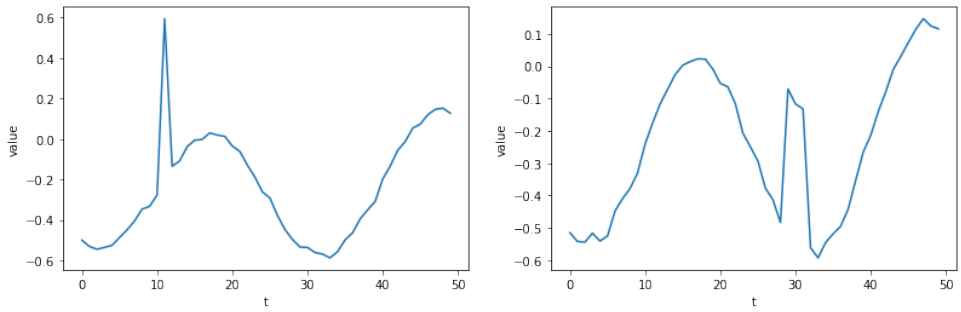


Figure 4.2: Detailed plot of anomalies present in the time-series data; point anomaly (left) and sequence anomaly of length 3 (right).

For the supervision data for ADL, as we know the exact time-series behavior in advance, we generated additional time-series windows containing anomalies with the same setting we used. The test time-series is constructed in the same way as the training data, but there is a difference in anomaly scale, position, and exact noise value due to the randomness in a generation.

Note that in time-series, the anomalous spot is designated to each time. So if we use a fixed size sliding window of the time-series as a data, a label for the outlier (anomaly) can be assigned to each dimension of the data. In our algorithm for time-series data, the dimension-wise outlier label is provided in the training stage (i.e. the time index showing anomalous behavior within the sliding window is given) and we use the adversarial loss of the form 3.2.4. For the other training data, which is a majority in the training step, the anomaly labels are unavailable in the training stage.

Table 4.1: Table of multivariate point data properties.

Dataset	N	n	# outliers (%)
Cardio	1831	21	176 (9.6%)
Breastw	683	9	239 (35%)
Ionosphere	351	33	126 (36%)
Satellite	6435	36	2036 (32%)
Vowels	1456	12	50 (3.4%)
Pima	768	8	268 (35%)
Mammography	11183	6	260 (2.3%)

4.1.2 Multivariate Point Data

Multivariate point data is data with multiple attributes with no correlation along with the time or data index. We used labeled natural datasets from the Outlier Detection Datasets (ODDS) [29]. Datasets with nominal and binary attributes are excluded. The dataset used are: *Cardio,* Wisconsin Breast Cancer (*Breastw*), *Ionosphere, Satellite, Vowels, Pima*, and *Mammography*. Table 4.1 provides the properties of datasets used in experiments.

Among the dataset, outlier labeled data are divided into two groups, the contamination set and the supervision set. The contamination set is concatenated with the non-outlier dataset and further divided into the training set and the test set. We used 20 percent of contamination set as a test set. The supervision set (i.e. $\mathcal{X}^-$) is fed to the ADL's adversarial loss and used for training. In the experiment, about 20 percent of outlier labeled data is used as supervision data.

As the label is given data instance-wise, we use the adversarial loss of form

3.2.2. The labels are available for only supervision data and test data (for evaluation) and not given for the training data.

4.2 Evaluation Process

To evaluate the outlier rejection performance (i.e. robustness on outliers) of dictionary learning algorithms, we employ the anomaly detection task which can act as an indirect measure for the robustness on outliers.

4.2.1 A Baseline of Anomaly Detection

As mentioned in chapter 2, dictionary learning aims to find a faithful and effective dictionary that well approximates a specific set of signals and not for the out-of-the-set signal. So we assume that if the dictionary successfully learned the given signal, the approximation result of the outlier data under the sparsity constraint will be worse than that of the inlier data. Inspired by the classification method presented by [30], we formulated the anomaly detection framework.

Given a new test sample $y \in \mathbb{R}^n$ and the sparsity regularization parameter $\lambda \in \mathbb{R}$, we first compute the sparse representation $\beta^* \in \mathbb{R}^p$ via problem 3.3.9. Then we reconstruct the original sample as $\hat{y} = D\beta^* \in \mathbb{R}^n$. Then we examine the residual between the original sample and the reconstructed sample. We define a scoring function for anomaly detection:

$$\begin{aligned} r_a(y) =& \|y - \hat{y}\|_2 = \|y - D\beta^*\|_2, \\ \text{where} \quad \beta^* =& \arg\min_{\beta \in \mathbb{R}^p} \|y - D\beta\|_1 + \|\beta\|_1. \end{aligned} \tag{4.2.2}$$

Then the decision of anomaly is done by thresholding the score value of the data

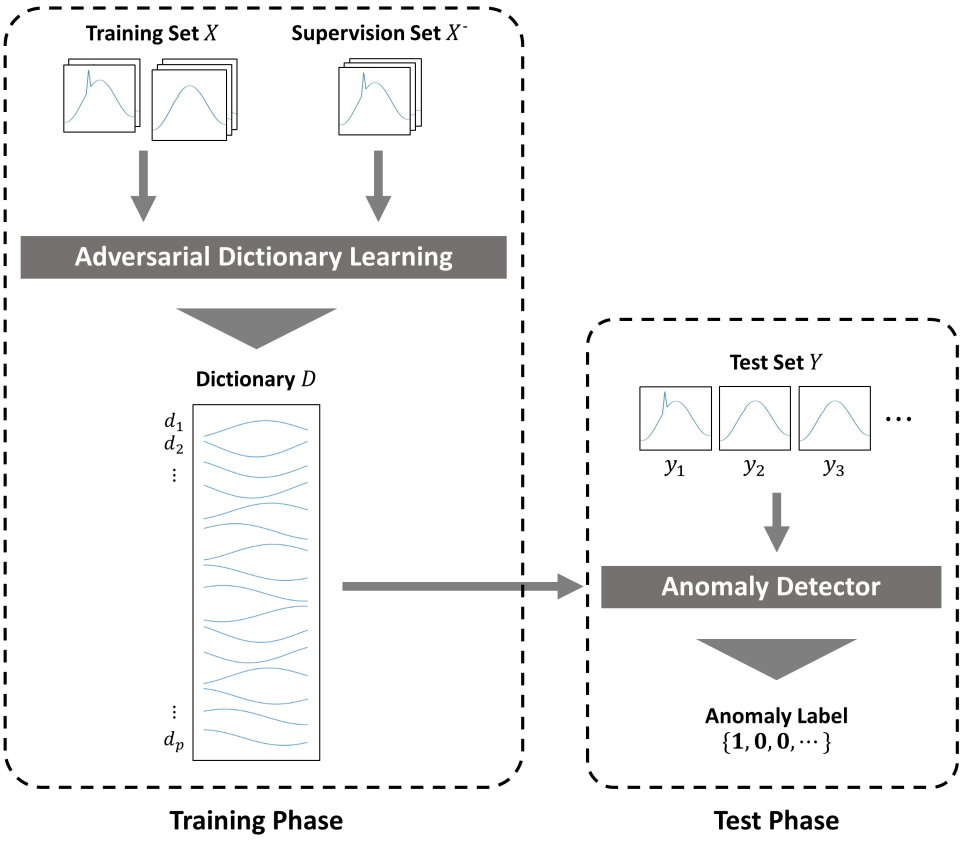


Figure 4.3: Schematic diagram of the anomaly detection framework using the ADL.

instance:

$$AD(y) = \begin{cases} 1, & \text{if } r_a(y) \geq \tau. \\ 0, & \text{otherwise.} \end{cases} \tag{4.2.3}$$

Here $AD(y) : \mathbb{R}^n \to [0, 1]$ denotes an anomaly detector; 1 means data instance is anomalous and 0 otherwise. For the case where the outlier label is given in each dimension (sliding window of time-series), the score is evaluated for each dimension independently. In the case of time-series data, there can be an overlapping area in the time axis between sliding windows, according to the settings. So we calculate and average the error for all sliding windows containing a value at time t. The anomaly score for time-series with sliding window data can be formulated as:

$$\begin{aligned} r_a^{ts}(t) =& \frac{1}{h} \sum_{i=t-h+1}^{t} (y_{i,t+1-i} - \hat{y}_{i,t+1-i})^2 \\ =& \frac{1}{h} \sum_{i=t-h+1}^{t} (y_{i,t+1-i} - (D\beta_i^*)_{t+1-i})^2, \\ \text{where} \quad \beta_i^* =& \underset{\beta_i \in \mathbb{R}^p}{\arg\min} \|y_i - D\beta_i\|_1 + \|\beta_i\|_1. \end{aligned} \tag{4.2.4}$$

Here y_i is an ith test data and $y_{i,j}$ is the jth dimension value of y_i. h is the size of the sliding window and we set the window's step size as 1. We let the time-series' time index is an integer value starts from zero and ith sliding window of time-series y_i covers the time index of $i \sim (i + h)$. Then the anomaly detector for the time-series can be defined similarly:

$$AD^{ts}(t) = \begin{cases} 1, & \text{if } r_a^{ts}(t) \geq \tau. \\ 0, & \text{otherwise,} \end{cases} \tag{4.2.5}$$

where $t = 1, 2, \ldots, T$.

The entire framework of anomaly detection using ADL is expressed in figure

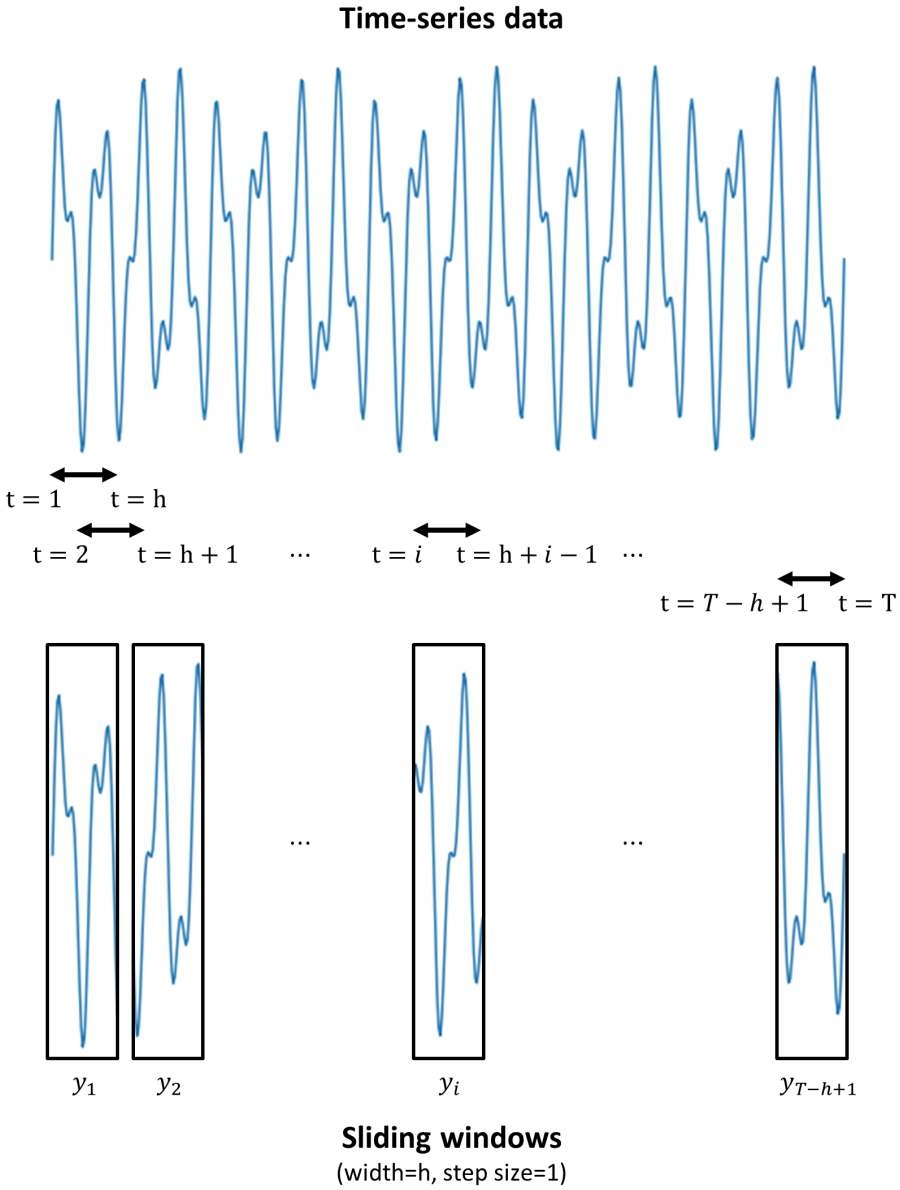


Figure 4.4: Sliding window generation from time-series data.

4.3. For the other dictionary learning algorithms, only the diagram of the training phase (left box) should be modified not to use the supervision data as an input.

4.2.2 ROC Curve and AUC

A receiver operating characteristic (ROC) curve is a plot that illustrates the performance of a binary classifier [31]. In classification problem using only two classes (in our case outlier and inlier classes), each instance in a dataset is mapped to one element of the set $\{p, n\}$ of positive and negative class labels. If the classification is done by thresholding a user-defined score, the classification result will vary depending on the selected value of the threshold. ROC visualizes the effectiveness of the classifier by plotting the true positive rate (TPR) against the false positive rate (FPR) by changing the threshold value used to the classifier. TPR is the proportion of instances (samples) correctly classified as positive among positive instances. FPR is the proportion of data instances classified as a negative label among positive instances.

After we obtain the ROC curve from the classifier, we can obtain a quantified evaluation result using AUC. AUC is basically a value of area under the ROC curve and a larger value means a better classifier. In particular, AUC of 0.5 means a meaningless classifier and that of 1 means a perfect classifier.

The anomaly detection can be seen as a two-class classification problem. Setting anomaly label 1 as a positive label and 0 as a negative label, we can evaluate the performance of the anomaly detector qualitatively. We will provide qualitative comparison results for several dictionary learning algorithms along with our method in the results section.

4.3 Experiment Setting

We set regularization parameter λ for the sparsity as 1.5 in all experiments. The number of dictionary atoms p is chosen according to the data dimension n, to have value about 1.2∼1.6 times of n. Value for the adversarial weight w is selected empirically for each type of dataset, from 0.018 to 0.48. We found some tendency in well-performing weight for the time-series, however.

$$w = 0.0012 * (s_a - 0.1) * n_a. \tag{4.3.6}$$

s_a is the average scale of anomalous deviation, i.e. l_2-norm value of mean error between inlier data and outlier data. n_a is the number of anomalies present in the training dataset. Although this information is not explicitly given in practice, we expect the domain-specific knowledge like the frequency of anomaly occurrence or the representative pattern of normal and abnormal data can be used.

For the univariate time-series data, we generated the supervision dataset X^- with the same size as the training data. In optimization, 10% of data is sampled and used for adversarial loss. For the multivariate point data, 20% of all outlier labeled data is used as a supervision data. All the other data are used as training data. About 0.5% to 10% of the supervision dataset is used each iteration, according to the size of the supervision dataset.

As the scale and mean of data instance affects the effective sparsity when obtaining sparse representation, all the data used are normalized to have zero mean and unit deviation before the training and evaluation.

The entire settings are presented in Table 4.2. p is the number of atoms in dictionary and k is the number of nonzero coefficients in representation for the constraint in K-SVD algorithm (only for l_1-K-SVD). λ is a regularization parameter for l_1 approximation problem. w is a weight for the adversarial loss. M' is the

Table 4.2: Experiment settings.

Dataset	p	k	λ	w	$M'(\%)$
Synthetic TS	32	10	1.5	eqn 4.3.6	10
Cardio	32	10	1.5	0.0384	10
Breastw	16	3	1.5	0.0864	10
Ionosphere	40	10	1.5	0.0576	10
Satellite	50	10	1.5	0.960	0.5
Vowels	20	5	1.5	0.0192	10
Pima	14	5	1.5	0.1152	10
Mammography	16	3	1.5	0.1920	10

number of supervision data used in each iteration (only for ADL).

All the algorithms are implemented and performed with Python on Intel (R) Core (TM) i7-7700 CPU @ 3.60GHz with 32GB memory. CVXPY [32][33] is used for optimization.

4.4 Results

The experiments on synthetic time-series data focus on the effect of the scale and frequency of the anomaly along with the scale of white noise. Then the applicability and superiority of our algorithm on real-world data are provided by the experiments on real multivariate point data. We compared the anomaly detection performance of our algorithm with other three dictionary learning algorithms, DL (convex relaxation), RDL, l_1-K-SVD.

Results on Univariate Time-series Data

As we cannot show all the dictionary obtained from every setting, we illustrate the representative results of dictionary learning. Figure 4.5 shows the plot of learned dictionary atoms from a single time-series sample. The outlier time-series signal added to the sample is the shift of length 5, and we qualitatively compare the robustness on the outliers by expecting the dictionary atoms learned. It can be seen that dictionary from our method shows the least amount of anomalous patterns, relative to the other methods. l_1-K-SVD shows the moderate performance on robustness, but some smoothed outlier pattern still remains in the dictionary. DL and RDL learned a dictionary whose majority of atoms has an anomalous pattern.

ROC curve for anomaly detector using the learned dictionaries in Figure 4.5 is shown in Figure 4.6. The larger area under the curve (AUC) means the better classifier. AUC of the anomaly detector is the highest when using a dictionary from the proposed method. The dependency between the amount of learned anomaly pattern and the anomaly detection performance can be found in figures.

We evaluated AUC with a total of 81 settings as explained in section 4.1.1. The results are shown in Table 4.3. The AUC value for each algorithm is averaged for each control variables to verify the effect of the setting. Our method shows better performance over other algorithms especially when the scale of the anomaly and the number of anomalies present in the dataset are high. When the number of the anomaly was 10, the average AUC of l_1-K-SVD was higher (which means better) than that of ours. The common property is that it is easier to detect anomalies when the scale of anomaly is high, the number of anomalies present in the training dataset is low, and the scale of white noise is low.

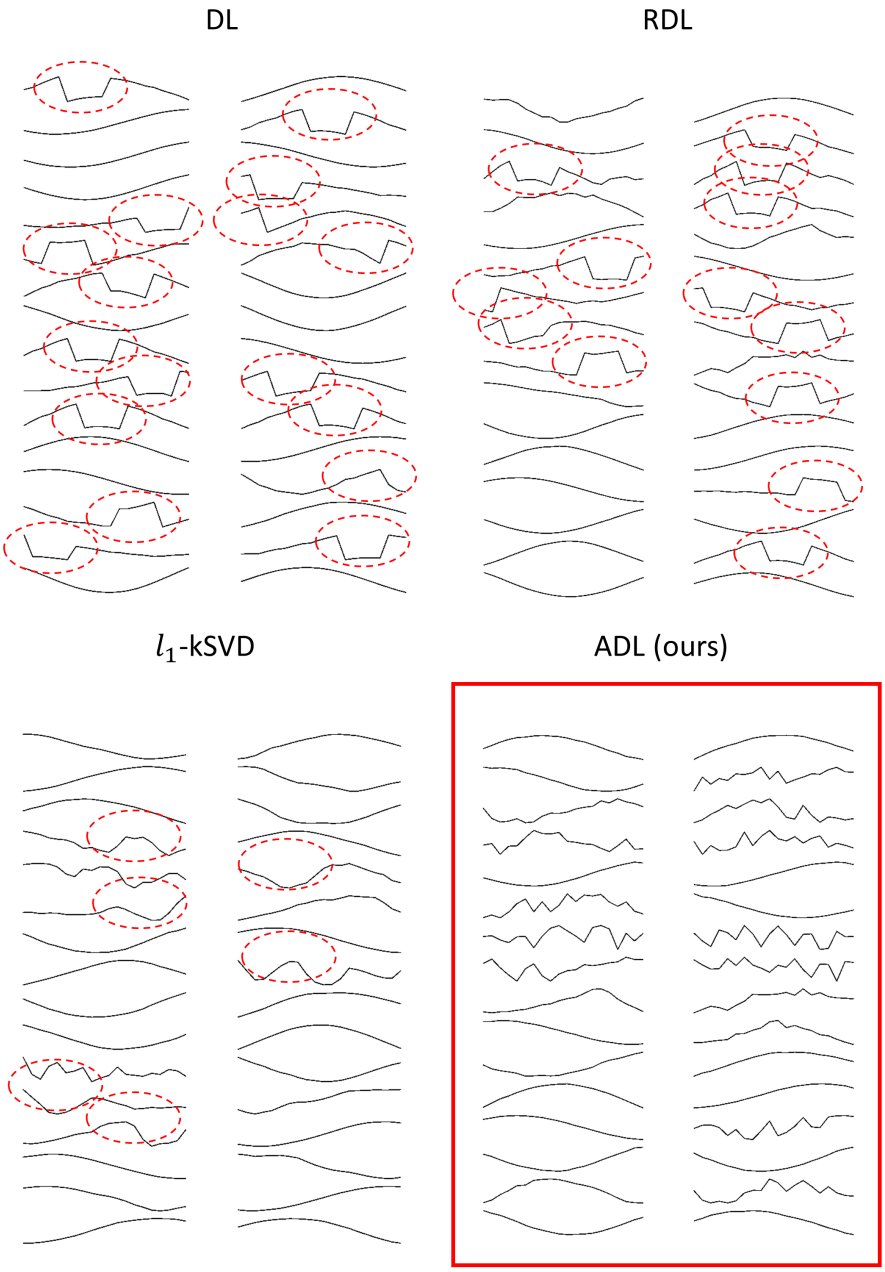


Figure 4.5: Visualization of learned dictionary from sample univariate time-series data of $\sigma = 0.05$, anomaly of length-5 shift, number of anomaly 10, scale of anomaly 0.8. The outlier behavior of length-5 shift (marked as red dot circle) is not learned in ADL.

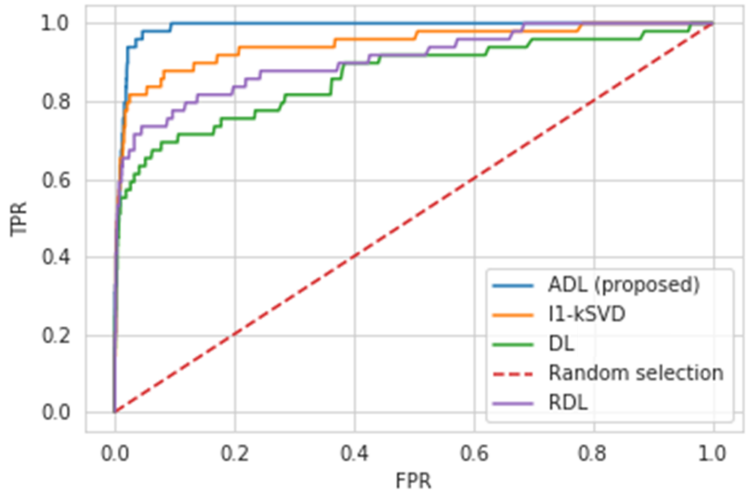


Figure 4.6: ROC curve of the anomaly detector using dictionaries presented in Figure 4.5.

Table 4.3: AUC value result for univariate time-series data.

Anomaly scale	**ADL (ours)**	**l_1-K-SVD**	**RDL**	**DL**
0.4	**0.912±0.101**	0.907±0.103	0.863±0.124	0.814±0.124
0.6	**0.951±0.061**	0.943±0.069	0.898±0.098	0.848±0.097
0.8	**0.971±0.041**	0.959±0.053	0.919±0.080	0.889±0.082

Anomaly number	**ADL (ours)**	**l_1-K-SVD**	**RDL**	**DL**
10	0.960±0.065	**0.964±0.059**	0.924±0.083	0.872±0.104
30	**0.941±0.076**	0.932±0.082	0.880±0.108	0.840±0.108
50	**0.932±0.083**	0.914±0.091	0.875±0.114	0.839±0.106

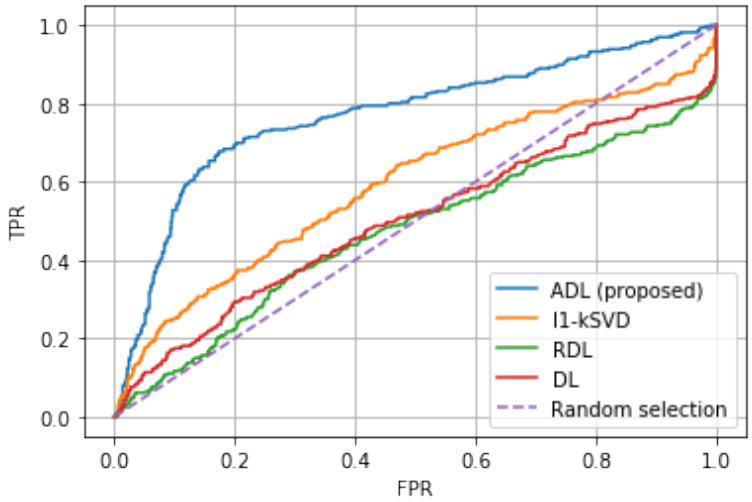
Noise scale	**ADL (ours)**	**l_1-K-SVD**	**RDL**	**DL**
0.05	**0.963±0.052**	0.956±0.054	0.922±0.075	0.886±0.078
0.10	**0.945±0.074**	0.938±0.077	0.890±0.105	0.851±0.107
0.15	**0.926±0.093**	0.914±0.100	0.868±0.121	0.814±0.107

Table 4.4: AUC value result for multivariate point data.

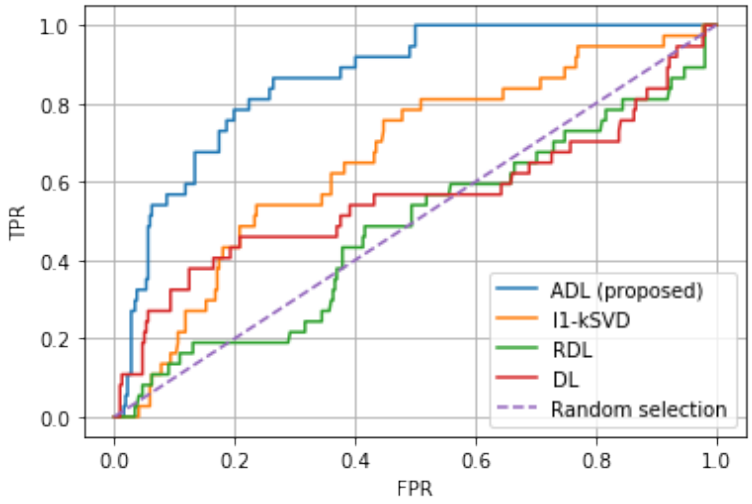
Dataset	**ADL (ours)**	**l_1-K-SVD**	**RDL**	**DL**
Cardio	**0.811±0.091**	0.559±0.103	0.495±0.040	0.542±0.082
Breastw	**0.786±0.071**	0.765±0.071	0.729±0.063	0.667±0.114
Ionosphere	0.982±0.014	0.976±0.022	0.979±0.017	**0.983±0.016**
Satellite	**0.759±0.048**	0.571±0.021	0.474±0.005	0.513±0.025
Vowels	**0.896±0.025**	0.845±0.082	0.881±0.032	0.791±0.060
Pima	**0.575±0.063**	0.426±0.062	0.457±0.045	0.528±0.028
Mammography	**0.780±0.045**	0.653±0.086	0.656±0.065	0.655±0.155

Results on Multivariate Point Data

The evaluation result for multivariate point data is presented in Table 4.4. Each evaluation is conducted 5 times with the same experiment settings. Our method reports better performance especially on *Cardio* and *Satellite* datasets. For the *Ionosphere* dataset, our method is slightly outperformed by the DL method but the overall performance for the dataset was high enough. As mentioned previously, our algorithm is based on the RDL method and if we set adversarial loss w to be zero, our method becomes identical to the RDL. We can see the explicit improvement of our method compared to the RDL, which is an evidence that the adversarial loss is indeed effective at rejecting outliers. Figure 4.7 shows the representative ROC results from the multivariate point experiments.



(a) ROC curve of the anomaly detector for *Satellite* dataset.



(b) ROC curve of the anomaly detector for *Cardio* dataset.

Figure 4.7: Representative results from multivariate point dataset

5

Conclusion

In this thesis, we proposed a new dictionary learning algorithm that robustly learns the representations of only inlier data. Different from the existing dictionary learning algorithms for the contaminated dataset, the distinguishing feature of our method is that it uses a sample out-of-class dataset in the learning procedure. The loss function is designed to make a dictionary not to obtain good quality (i.e. approximate given signal under designated sparsity constraint) sparse representation for only outlier data. This scheme is implemented by penalizing the approximation error for the outlier data along with optimizing the original dictionary learning problem.

Our method is particularly advantageous when the outlier signal is generated with some patterns. Experiments on natural multivariate point data suggest that the signal modeling ability can be greatly improved by using a small amount of supervision data (labeled outlier data). Further, our method leaves an opportunity for the practitioner to designate which class of signals not to learn, without

manually eliminating the not-to-learn samples in the dataset. However, if there exists no explicit pattern on outliers so the supervision dataset cannot represent the out-of-class data well, the algorithm does not show the dramatic improvement.

We expect the performance of the algorithm to be further improved if the amount of supervision data grows in time and well generalizes the outlier signal. Our algorithm uses a negative logarithm function that acts as a barrier function for the approximation of outlier data. Therefore if the number of supervision data used in each iteration M' is large, the solution tends to fall into the local minima too early. This problem should be further investigated. The computational time is another issue: our method requires far much time than greedy strategy based algorithms. If we can implement the scheme of adversarial loss using supervision data to the greedy strategy based dictionary learning, the time efficiency will be greatly improved.

Bibliography

[1] Zheng Zhang, Yong Xu, Jian Yang, Xuelong Li, and David Zhang. A survey of sparse representation: algorithms and applications. *IEEE access*, 3:490–530, 2015.

[2] David L Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.

[3] Ivana Tosic and Pascal Frossard. Dictionary learning. *IEEE Signal Processing Magazine*, 28(2):27–38, 2011.

[4] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311–4322, 2006.

[5] Tianzhu Zhang, Bernard Ghanem, Si Liu, and Narendra Ahuja. Robust visual tracking via multi-task sparse learning. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2042–2049. IEEE, 2012.

[6] Jianchao Yang, Kai Yu, Yihong Gong, and Thomas Huang. Linear spatial pyramid matching using sparse coding for image classification. In *2009 IEEE Conference on computer vision and pattern recognition*, pages 1794–1801. IEEE, 2009.

[7] Naoya Takeishi and Takehisa Yairi. Anomaly detection from multivariate time-series with sparse representation. In *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2651–2656. IEEE, 2014.

[8] Cong Zhao, Xiaogang Wang, and Wai-Kuen Cham. Background subtraction via robust dictionary learning. *EURASIP Journal on Image and Video Processing*, 2011:1–12, 2011.

[9] Wenhao Jiang, Feiping Nie, and Heng Huang. Robust dictionary learning with capped l1-norm. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[10] Subhadip Mukherjee, Rupam Basu, and Chandra Sekhar Seelamantula. ℓ1-k-svd: A robust dictionary learning algorithm with simultaneous update. *Signal Processing*, 123:42–52, 2016.

[11] Asif Iqbal and Abd-Krim Seghouane. An α-divergence-based approach for robust dictionary learning. *IEEE Transactions on Image Processing*, 28(11):5729–5739, 2019.

[12] Naiyan Wang, Jingdong Wang, and Dit-Yan Yeung. Online robust non-negative dictionary learning for visual tracking. In *Proceedings of the IEEE international conference on computer vision*, pages 657–664, 2013.

[13] Hua Wang, Feiping Nie, Weidong Cai, and Heng Huang. Semi-supervised robust dictionary learning via efficient l-norms minimization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1145–1152, 2013.

[14] Qiang Zhang and Baoxin Li. Discriminative k-svd for dictionary learning in face recognition. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2691–2698. IEEE, 2010.

[15] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman. Discriminative learned dictionaries for local image analysis. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[16] Michael Elad. *Sparse and redundant representations: from theory to applications in signal and image processing*. Springer Science & Business Media, 2010.

[17] Balas Kausik Natarajan. Sparse approximate solutions to linear systems. *SIAM journal on computing*, 24(2):227–234, 1995.

[18] Edoardo Amaldi, Viggo Kann, et al. On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science*, 209(1):237–260, 1998.

[19] David L Donoho. For most large underdetermined systems of linear equations the minimal 1-norm solution is also the sparsest solution. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 59(6):797–829, 2006.

[20] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[21] Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, et al. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

[22] Qin Lyu, Zhouchen Lin, Yiyuan She, and Chao Zhang. A comparison of typical ℓp minimization algorithms. *Neurocomputing*, 119:413–424, 2013.

[23] Cewu Lu, Jiaping Shi, and Jiaya Jia. Online robust dictionary learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 415–422, 2013.

[24] Emmanuel J Candes and Terence Tao. Decoding by linear programming. *IEEE transactions on information theory*, 51(12):4203–4215, 2005.

[25] Nicolai Bissantz, Lutz Dümbgen, Axel Munk, and Bernd Stratmann. Convergence analysis of generalized iteratively reweighted least squares algorithms on convex function spaces. *SIAM Journal on Optimization*, 19(4):1828–1845, 2009.

[26] Raghavendra Chalapathy and Sanjay Chawla. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*, 2019.

[27] N Laptev and S Amizadeh. Yahoo anomaly detection dataset s5. *URL http://webscope. sandbox. yahoo. com/catalog. php*, 2015.

[28] Barbara Pilastre, Loic Boussouf, Stéphane d'Escrivan, and Jean-Yves Tourneret. Anomaly detection in mixed telemetry data using a sparse representation and dictionary learning. *Signal Processing*, 168:107320, 2020.

[29] Shebuti Rayana. Odds library. *URL http://odds.cs.stonybrook.edu*, 2016.

[30] John Wright, Allen Y Yang, Arvind Ganesh, S Shankar Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE transactions on pattern analysis and machine intelligence*, 31(2):210–227, 2008.

[31] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[32] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

[33] Akshay Agrawal, Robin Verschueren, Steven Diamond, and Stephen Boyd. A rewriting system for convex optimization problems. *Journal of Control and Decision*, 5(1):42–60, 2018.

국문초록

본 논문에서는 이상치가 아닌 데이터의 희소 표현만을 학습하는 준지도 사전 학습 알고리즘을 제안한다. 데이터셋에 섞여 있는 이상치는 사전 학습의 주요한 문제로, 실제 문제에 적용 시 바람직하지 않은 성능을 초래한다. 본 연구에서 제안하는 적대적 사전 학습(ADL) 알고리즘은 이상치 데이터로 구성된 감독 데이터셋을 학습에 이용한다. 우리의 알고리즘은 주어진 이상치 데이터를 잘 표현하는 사전에 페널티를 주고, 이것은 사전이 학습 데이터셋에 섞여 있는 이상치에 강건하게 학습되도록 한다. 제안된 방법은 기존의 사전 학습 방법들과 비교해 이상치의 비중이 높은 데이터셋에서도 효과적으로 사전을 학습해 낸다. 이 연구에서는 인공적인 단변량 시계열 데이터와 다변량 점 데이터에 대한 이상치 탐지 실험을 통해 알고리즘의 유용성을 경험적으로 검증한다.

주요어: 희소 표현, 사전 학습, 준지도학습, 이상치 탐지

학번: 2018-21570