A Fast Dictionary-Learning-based Classification Scheme Using Undercomplete Dictionaries

Saeed Mohseni-Sehdeh, Massoud Babaie-Zadeh
Electrical Engineering Department, Sharif Univesity of Technology
saeedmohseniph@gmail.com, mbzadeh@yahoo.com

Abstract

In dictionary-learning-based classification methods, a given data point is classified based on its representation over one or possibly more learned dictionaries. The goal is to find dictionaries that minimize the classification error. Previous works aimed to train dictionaries with representation and classification powers by using overcomplete dictionaries and sparse coding. These approaches are computationally expensive and do not scale readily to problems with high dimensional data. This paper presents a dictionary-learning-based classification method with the primary goal of classification and not representation. We propose to train multiple undercomplete dictionaries (one for each class of the problem). Each dictionary approximates the given test data, and the one with the lowest reconstruction error determines the class. Singular value decomposition (SVD) is used to obtain a straightforward algorithm for the resulted optimization problem. Simulation results show that our method achieves a higher accuracy compared with a number of successful sparse representation based classification methods, while having a significantly lower computational cost.

Keywords: Dictionary learning, supervised classification, undercomplete dictionary, singular value decomposition (SVD), gradient projection.

1. Introduction

In supervised classification, the goal is to learn the general patterns and structures of a particular multi-class dataset by using a subset of its labeled data, called training dataset [1]. The performance of these methods is then evaluated based on their classification accuracy on a testing dataset [1]. Su-

pervised classification has applications in a number of domains[1], e.g. text, audio, image, and video.

A supervised classification problem with C classes can be formulated as follows [1]: the inputs of the problem are two sets of labeled datasets: the training and testing datasets. These datasets can be modeled as two matrices: $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots \mathbf{y}_T] \in \mathbb{R}^{m \times T}$ for the training data, and $\mathbf{Y'} = [\mathbf{y}'_1, \mathbf{y}'_2, \dots \mathbf{y}'_{T'}] \in \mathbb{R}^{m \times T'}$ for the testing data, where each $\mathbf{y}_i$ and $\mathbf{y}'_j$ for $i \in \{1, 2, \dots, T\}$ and $j \in \{1, 2, \dots, T'\}$ is a sample data in $\mathbb{R}^m$. The labels of these datasets can be represented by two vectors: $\boldsymbol{\ell} = [l_1, l_2, \dots, l_T]^T \in \mathbb{R}^T$ and $\boldsymbol{\ell}' = [l'_1, l'_2, \dots, l'_{T'}]^T \in \mathbb{R}^{T'}$ for training and testing datasets, respectively, in which l_i and l'_j for $i \in \{1, 2, \dots, T\}$ and $j \in \{1, 2, \dots, T'\}$ are the labels of $\mathbf{y}_i$ and $\mathbf{y}'_j$, respectively, and their values are from $\{1, 2, \dots, C\}$. Based on the training data $\mathbf{Y}$ and its labels, the goal is to create and optimize a classification method $\mathcal{M}_{\boldsymbol{\theta}}(\mathbf{y})$ with a datapoint $(\mathbf{y})$ as input, its estimated labels as output and the internal parameters vector $\boldsymbol{\theta}$ that maximizes

$$\sum_{j=1}^{T'} \delta(l'_j - \mathcal{M}_{\boldsymbol{\theta}}(\mathbf{y}'_j)), \tag{1}$$

where δ: $\mathbb{R} \to \mathbb{R}$ is defined as

$$\delta(x) = \begin{cases} 1 & x = 0, \\ 0 & x \neq 0. \end{cases}$$

One of the approaches used for supervised classification is supervised classification based on dictionary learning (DL) [2, 3, 4, 5, 6, 7, 8]. In this approach, a labeled data point is classified based on its representation over one or possibly more learned dictionaries [2, 3, 4, 5, 6, 7, 8]. The representation of a datapoint $\mathbf{y} \in \mathbb{R}^m$ over a given dictionary $\mathbf{D} \in \mathbb{R}^{m \times n}$ is

$$\mathbf{x} = \underset{\hat{\mathbf{x}}}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{D}\hat{\mathbf{x}}\|_2. \tag{2}$$

In DL-based supervised classification, the idea is to leverage the information in $\mathbf{x}$ for classification purposes, and $\mathbf{D}$ should be learned from the information in the training dataset and its labels such that the classification accuracy is maximized [2, 3, 4, 5, 7, 8].

On the other hand, in traditional dictionary learning [9, 10, 11], the goal is to learn dictionaries that result in a sparse representation of the training

data. More specifically, one tries to solve

$$(\mathbf{D}, \mathbf{X}) = \underset{\hat{\mathbf{D}} \in \mathcal{D}, \hat{\mathbf{X}} \in \mathcal{X}}{\operatorname{argmin}} \|\mathbf{Y} - \hat{\mathbf{D}}\hat{\mathbf{X}}\|_F^2, \tag{3}$$

where $\mathbf{Y} \in \mathbb{R}^{m \times T}$ is the set of training signals, $\mathcal{D} \triangleq \{\mathbf{D} \in \mathbb{R}^{m \times n} : \forall i, \|\mathbf{d}_i\|_2^2 = 1\}$ where $n > m$, and $\mathcal{X} \triangleq \{\mathbf{X} \in \mathbb{R}^{n \times T} : \forall i, \|\mathbf{x}_i\|_0 \leq s\}$, in which $\|.\|_F$ denotes Frobenius norm. Although this approach results in dictionaries with high signal representation ability [9, 10, 11], it is not appropriate for learning dictionaries to be used in supervised classification. This is because in (3), the labels of the training data have not been used at all in training the dictionary. In other words, in learning dictionaries for classification, the main goal is to learn dictionaries with good discrimination abilities rather than good representation abilities. So, in learning dictionaries to be used in DL-based supervised classification [2, 3, 4, 5, 6, 7, 8], the label information should be incorporated in dictionary learning to find a $\mathbf{D}$ that maximizes the classification accuracy.

There are a number of attempts to incorporate the label information in dictionary learning and use the resulted dictionaries for supervised classification problems [2, 3, 4, 5, 6, 7, 8], which are all based on learning *overcomplete* dictionaries with both representation and discrimination power. Most of the previous attempts [4, 2, 5, 3] have treated the problem as two separate stages. In the first stage, they optimize a dictionary with representational power and then train a classifier based on it. Some approaches such as DKSVD [7], LCKSVD-1 and LCKSVD-2 [8], apply joint learning by using a unifying formulation for simultaneously training an overcomplete dictionary and a linear classifier. The authors of [12, 13] suggest to simultaneously learn an analysis dictionary along with a structured synthesis dictionary and a linear classifier in order to find the sparse representation of data more efficiently. In [14], a deep DL-method for image representation and classification is introduced. In this work, multiple dictionaries are learned in a multi-layer manner along with a softmax classifier for extracting hierarchical information hidden in data to have models with strong representational and discriminational abilities. In [15], DL and convolutional Neural Networks (CNN) [16] are integrated to improve representation learning. Another approach [2] trains *multiple* overcomplete dictionaries, one for each class of the dataset. It uses the reconstruction error as the classification criterion. More specifically, given a test data point $\mathbf{y}$, it solves (2) for every dictionary and calculates the estimates of the given test data point as $\mathbf{y}_i = \mathbf{D}_i\mathbf{x}_i$ for $i = \{1, 2, \ldots, C\}$,

where C is the number of classes in the problem. The one with minimum reconstruction error (*i.e.* minimum $e_i \triangleq \|\mathbf{y} - \mathbf{y}_i\|_2$) determines the label of the data.

The papers [17, 18, 19, 20] use DL for re-identification purposes. In these works, the problem is to re-identify a person in an image/video from one camera to an image/video in another camera. They learn a pair of transforms to match the feature spaces of both inputs and a pair of dictionaries for coding the features jointly in a formulation for both representation and discrimination. The coded features are then used for re-identifying the person. In [21, 22], representation learning is used for multi-view data classification. In [23] the DL is used for classification of multi-spectral images. The authors of [24] use DL for image annotation. This work proposes a DL algorithm that simultaneously learns a label embedding transform and an overcomplete dictionary. The annotations are then determined by the sparse representation of embedded testing data.

In sparse representation-based classification (SRC) [6], an overcomplete dictionary is directly created from the training samples. It forms a dictionary as $\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2, \mathbf{D}_3, \dots \mathbf{D}_C]$ where each $\mathbf{D}_i$ for $i \in \{1, 2, \dots, C\}$ is a submatrix consisting of a subset of training samples corresponding to the i^{th} class. It assumes that a test sample belonging to the i^{th} class is well approximated by the subspace spanned by the columns of $\mathbf{D}_i$. By this assumption, the sparse representation of a test data of the i^{th} class over $\mathbf{D}$ is expected to have significant values corresponding to the atoms of $\mathbf{D}_i$ and zero or near-zero values otherwise. Using this assumption, SRC solves (5) for each test data to find the sparse representation. The sparse representation and the reconstruction error criterion is then used to classify the test data. Although SRC performs well for face recognition applications, it has a few practical drawbacks:

1. To improve the performance of this algorithm, one requires a large dictionary with a large number of training samples for each class [7]. But a large dictionary degrades the performance of sparse solvers [7]. It is also computationally expensive due to a large number of parameters (dictionary atoms).
2. In order to have a good performance, the dictionary atoms must be selected carefully to make sure the atoms span the subspace of each class of the dataset fairly well [7].

One of the problems of sparse representation based dictionary learning

methods for both representation and discrimination is their computational complexity. In order to find the sparse representation, one has to solve

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{x}\|_0 \quad \text{s.t.} \quad \mathbf{y} = \mathbf{D}\mathbf{x}, \tag{4}$$

where $\|.\|_0$ is the ℓ^0-norm (*i.e.* the number of non-zero entries), $\mathbf{y} \in \mathbb{R}^m$ is the test data and $\mathbf{D} \in \mathbb{R}^{m \times n}$ where $n > m$ is the dictionary. This problem has a unique solution under some mild conditions [25]. However, it is not computationally tractable[26]. So, it is usually replaced by [27]

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{x}\|_1 \quad \text{s.t.} \quad \mathbf{y} = \mathbf{D}\mathbf{x}, \tag{5}$$

and it is shown [28, 29, 30] that its solution is equal to the solution of (4), provided that it is sparse enough. The problem (5) can be solved in polynomial time [31]. Although finding the solution of (5) is tractable, it is still highly computationally demanding, especially when the data dimension is large.

In this paper, inspired by the idea of learning a separate dictionary for each class of the classification problem [2], and using the reconstruction error as the classification criterion [2, 6], we propose to learn *multiple undercomplete* dictionaries (one for each class of the dataset) to have an accurate and computationally efficient DL-based classification method. Our simulation results demonstrate that our method has a significantly lower computational cost compared with SRC [6], DKSVD [7], LCKSVD-1 and LCKSVD-2 [8], while it achieves even higher classification accuracies.

The main contributions of this paper can be viewed in two perspectives. Firstly, in contrast to previous DL-based supervised classification schemes that use overcomplete dictionaries (dictionaries with more columns than rows), our approach is based on undercomplete dictionaries (dictionaries with fewer columns than rows). To our best knowledge, undercomplete dictionaries have not previously been used for DL-based supervised classification problem. Undercomplete dictionaries have fewer parameters than overcomplete dictionaries, making them computationally more efficient. More importantly, as will be stated in Section 3, finding the representation of a signal over an undercomplete dictionary is easily computed by a linear operation, while for overcomplete dictionaries, (4) has to be solved, which is a computationally demanding task. As a result, our method has a significantly lower training and classification costs, while achieving even a better accuracy. These properties make our method suitable for applications where fast classification is

required, applications where constant learning is required (e.g. when new training data are acquired in real-time) and applications where computer storage is a constraint.

Secondly, an algorithm for learning optimal undercomplete dictionaries from a set of training data is introduced based on singular value decomposition.

The paper is organized as follows. In Section 2, the main idea is formulated, and then in Section 3 the final algorithm is developed. Finally, Section 4 is devoted to experimental results. Simulations are performed on a synthetic data, on MNIST dataset [32], and on Fashion MNIST dataset [33].

2. The Main Idea

Given a set of labeled training data, the goal is to train multiple undercomplete dictionaries (one for each class) to minimize the classification error based on the minimum reconstruction error criterion. More specifically, with $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots \mathbf{y}_T] \in \mathbb{R}^{m \times T}$ as the training dataset with corresponding labels $\boldsymbol{\ell} = [l_1, l_2, \dots, l_T]^T \in \mathbb{R}^T$, in which each l_i for $i \in \{1, 2, \dots, T\}$ is from $\{1, 2, \dots, C\}$, the goal is to simply learn a set of *undercomplete dictionaries* $\mathbf{D}_j \in \mathbb{R}^{m \times n}$, $(m > n)$ for $j = \{1, 2, \dots, C\}$ such that $\mathbf{D}_j$ minimizes the representation error of the training data of the j^{th} class. Classification criterion is then the reconstruction error. More precisely, the estimated label of a given datapoint $\mathbf{y}$ is

$$\underset{j}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{D}_j \mathbf{x}_j\|_2 \quad \text{for} \quad j \in \{1, 2, \dots, C\}, \tag{6}$$

in which $\mathbf{x}_j \in \mathbb{R}^n$ is the representation of $\mathbf{y}$ over $\mathbf{D}_j$, that is,

$$\mathbf{x}_j = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{D}_j \mathbf{x}\|_2 \quad \text{for} \quad j = \{1, 2, \dots, C\}. \tag{7}$$

For each class of the training data, the problem can be summarized as learning an undercomplete dictionary that minimizes the signal reconstruction error. This problem should be separately solved for each class of the dataset, resulting in a set of optimal undercomplete dictionaries, one for each class of the dataset.

Although an approach is already available for learning an undercomplete dictionary from a set of training signals [34], it has a sparsity constraint on the signal representation. So, this method does not utilize the full representational capability of an undercomplete dictionary since it uses only a limited

number of atoms for representing a signal. In fact, as is presented in the following section, the optimal representation over an undercomplete dictionary is not necessarily sparse. Thus, the following section provides an approach for learning undercomplete dictionaries without the sparsity constraint, and it ends up with an undercomplete dictionary learning algorithm.

3. Learning Undercomplete Dictionary

In this section, an approach for learning an undercomplete dictionary, for a particular class of the dataset, is presented. Our cost function for this dictionary learning problem is the total reconstruction error, that is,

$$E_c = \sum_{i=1}^{T_c} \|\mathbf{y}_i^c - \hat{\mathbf{y}}_i^c\|_F^2, \tag{8}$$

where $\mathbf{y}_i^c$ is the i^{th} element of $\mathbf{Y}_c = [\mathbf{y}_1^c, \mathbf{y}_2^c, \dots \mathbf{y}_{T_c}^c] \in \mathbb{R}^{m \times T_c}$, which is the dataset corresponding to the c^{th} class of the problem where $c \in \{1, 2, \dots, C\}$, and $\hat{\mathbf{y}}_i^c$ is the best approximate of $\mathbf{y}_i^c$ in terms of a linear combination of the columns of $\mathbf{D}_c \in \mathbb{R}^{m \times n}$. So, $\hat{\mathbf{y}}_i^c = \mathbf{D}_c \mathbf{x}_i^c$ where

$$\mathbf{x}_i^c = (\mathbf{D}_c^T \mathbf{D}_c)^{-1} \mathbf{D}_c^T \mathbf{y}_i^c, \tag{9}$$

provided that the columns of $\mathbf{D}_c$ are linearly independent [35] (equivalently, provided that $\mathbf{D}_c$ has non-zero singular values). We assume that this condition holds for $\mathbf{D}_c$ through the rest of this section. As (9) shows, $\mathbf{x}_i^c$ is not necessarily sparse. By using (9),

$$\begin{aligned} \|\mathbf{y}_i^c - \hat{\mathbf{y}}_i^c\|_2^2 &= \|\mathbf{y}_i^c - \mathbf{D}_c \mathbf{x}_i^c\|_2^2 \\ &= \|(\mathbf{I} - \mathbf{D}_c (\mathbf{D}_c^T \mathbf{D}_c)^{-1} \mathbf{D}_c^T) \mathbf{y}_i^c\|_2^2, \end{aligned} \tag{10}$$

and hence

$$E_c = \sum_{i=1}^{T_c} \|(\mathbf{I} - \mathbf{D}_c (\mathbf{D}_c^T \mathbf{D}_c)^{-1} \mathbf{D}_c^T) \mathbf{y}_i^c\|_2^2. \tag{11}$$

The goal is now finding a $\mathbf{D}_c$ that minimizes E_c.

3.1. Cost function simplification using singular value decomposition

To simplify the cost function in (11), we use singular value decomposition (SVD) [35]. The SVD of an arbitrary matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ is given as

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \tag{12}$$

where $\mathbf{U} \in \mathbb{R}^{m\times m}$ and $\mathbf{V} \in \mathbb{R}^{n\times n}$ are orthonormal (unitary) matrices, and $\mathbf{\Sigma} \in \mathbb{R}^{m\times n}$ is a diagonal matrix with non-negative diagonal values. By using the SVD of $\mathbf{D}_c$, that is $\mathbf{D}_c = \mathbf{U}_c\mathbf{\Sigma}_c\mathbf{V}_c^T$, (11) becomes

$$\begin{aligned} E_c &= \sum_{i=1}^{T_c} \|(\mathbf{I} - \mathbf{U}_c\mathbf{\Sigma}_c(\mathbf{\Sigma}_c^T\mathbf{\Sigma}_c)^{-1}\mathbf{\Sigma}_c^T\mathbf{U}_c^T)\mathbf{y}_i^c\|_2^2 \\ &= \sum_{i=1}^{T_c} \|(\mathbf{I} - \mathbf{U}_c\mathbf{B}_c\mathbf{U}_c^T)\mathbf{y}_i^c\|_2^2, \end{aligned} \tag{13}$$

where $\mathbf{B}_c \triangleq \mathbf{\Sigma}_c(\mathbf{\Sigma}_c^T\mathbf{\Sigma}_c)^{-1}\mathbf{\Sigma}_c^T$. It is easy to see that $\mathbf{B}_c \in \mathbb{R}^{m\times m}$ is of the form

$$\mathbf{B}_c = \begin{pmatrix} \mathbf{I}_n & \mathbf{0}_{n\times(m-n)} \\ \mathbf{0}_{(m-n)\times n} & \mathbf{0}_{(m-n)\times(m-n)} \end{pmatrix}, \tag{14}$$

and

$$\mathbf{B}_c^2 = \mathbf{B}_c = \mathbf{B}_c^T. \tag{15}$$

As (13) shows, the total reconstruction error (E_c) of an undercomplete dictionary $\mathbf{D}_c \in \mathbb{R}^{m\times n}$ depends only on its right singular matrix $\mathbf{U}_c$. By using this simplification, the optimization problem can be summarized as

$$\begin{aligned} &\underset{\mathbf{U}_c}{\text{minimize}}\{E_c \triangleq \sum_{i=1}^{T_c} \|(\mathbf{I} - \mathbf{U}_c\mathbf{B}_c\mathbf{U}_c^T)\mathbf{y}_i^c\|_2^2\} \\ &\text{s.t.} \quad \mathbf{U}_c^T\mathbf{U}_c = \mathbf{I} \end{aligned} \tag{16}$$

3.2. Optimization Algorithm

The optimization problem (16) is a constrained non-convex optimization problem [36]. To solve this problem, we use a Gradient Projection (GP) approach [37, Chapter 2]. Each iteration of GP consists of two steps: an iteration of gradient descent, followed by projection onto the feasible region.

The gradient descent step of our problem requires the derivative of E_c with respect to $\mathbf{U}_c$. This can be calculated as

$$\frac{dE_c}{d\mathbf{U}_c} = \frac{d}{d\mathbf{U}_c}\sum_{i=1}^{T_c}\|(\mathbf{I}-\mathbf{U}_c\mathbf{B}_c\mathbf{U}_c^T)\mathbf{y}_i^c\|_2^2, \tag{17}$$

$$= \frac{d}{d\mathbf{U}_c}\sum_{i=1}^{T_c}(\mathbf{y}_i^c)^T[\mathbf{I}-2\mathbf{U}_c\mathbf{B}_c\mathbf{U}_c^T+\mathbf{U}_c\mathbf{B}_c^2\mathbf{U}_c^T]\mathbf{y}_i^c, \tag{18}$$

$$= \frac{d}{d\mathbf{U}_c}\sum_{i=1}^{T_c}\|\mathbf{y}_i^c\|_2^2-(\mathbf{y}_i^c)^T\mathbf{U}_c\mathbf{B}_c\mathbf{U}_c^T\mathbf{y}_i^c, \tag{19}$$

$$= -\frac{d}{d\mathbf{U}_c}(\sum_{i=1}^{T_c}(\mathbf{y}_i^c)^T\sum_{j=1}^{n}\mathbf{u}_c^j(\mathbf{u}_c^j)^T\mathbf{y}_i^c), \tag{20}$$

$$= -2\mathbf{Y}_c\mathbf{Y}_c^T\mathbf{C}, \tag{21}$$

where (18) follows from the orthonormality of $\mathbf{U}_c$, and (19) follows from (15). In (20), $\mathbf{u}_c^j$ denotes the j^{th} column of $\mathbf{U}_c$. In (21), $\mathbf{C} \triangleq [\mathbf{U}_c^n \mid \mathbf{0}_{m\times(m-n)}]$, where $\mathbf{U}_c^n$ is the matrix composed of the first n columns of $\mathbf{U}_c$.

The projection step for our problem is the solution of the following problem: Given an arbitrary square matrix $\mathbf{A}$ with singular value decomposition $\mathbf{A}=\mathbf{U}_\mathbf{A}\mathbf{\Sigma}_\mathbf{A}\mathbf{V}_\mathbf{A}^T$, find $\hat{\mathbf{A}}$ such that

$$\hat{\mathbf{A}} = \underset{\tilde{\mathbf{A}}}{\operatorname{argmin}}\|\mathbf{A}-\tilde{\mathbf{A}}\|_F \quad \text{s.t} \quad \tilde{\mathbf{A}}^T\tilde{\mathbf{A}}=\mathbf{I}. \tag{22}$$

The solution of (22) is $\hat{\mathbf{A}}=\mathbf{U}_\mathbf{A}\mathbf{V}_\mathbf{A}^T$ [38, p. 327].

3.3. Choosing $\mathbf{\Sigma}_c$ and $\mathbf{V}_c^T$

As (16) shows, the cost function depends only on the right singular matrix of an undercomplete dictionary. In other words, given a specific orthonormal matrix $\mathbf{U}_c \in \mathbb{R}^{m\times m}$, regardless of the choice of a tall diagonal matrix $\mathbf{\Sigma}_c \in \mathbb{R}^{m\times n}$ with positive diagonal entries, and an orthonormal matrix $\mathbf{V}_c \in \mathbb{R}^{n\times n}$, any undercomplete dictionary formed by $\mathbf{D}_c=\mathbf{U}_c\mathbf{\Sigma}_c\mathbf{V}_c^T$ results in the same value of the total reconstruction error.

Therefore, solving (16) provides only the right singular matrix of the optimum undercomplete dictionary. To complete the solution, a diagonal $\mathbf{\Sigma}_c \in \mathbb{R}^{m\times n}$ with positive values along the main diagonal, and an orthonormal $\mathbf{V}_c \in \mathbb{R}^{n\times n}$ should be chosen. A strategy is to choose them randomly, but

the structure of the atoms in the final dictionary depends on this choice. A desired property of a dictionary is to have orthogonal atoms with unit ℓ^2-norms. In the following, Theorem 1 provides an approach to choose $\mathbf{\Sigma}_c$ such that the final dictionary has orthogonal atoms with unit ℓ^2-norms and shows that this is independent of $\mathbf{V}_c$. Then, Theorem 2 shows that the choice of $\mathbf{V}_c$ has no impact on the subspace spanned by the atoms of the final dictionary.

Theorem 1. *Given an orthonormal matrix $\mathbf{U} \in \mathbb{R}^{m\times m}$, by choosing a tall diagonal matrix $\mathbf{\Sigma} \in \mathbb{R}^{m\times n}$ with 1's on the main diagonal, and an arbitrary orthonormal matrix $\mathbf{V} \in \mathbb{R}^{n\times n}$, the undercomplete dictionary $\mathbf{D} = \mathbf{U\Sigma V}^T$ has orthogonal atoms with unit ℓ^2-norms.*

Proof. Given an undercomplete dictionary $\mathbf{D} \in \mathbb{R}^{m\times n}$, the entries of the main diagonal of $\mathbf{D}^T\mathbf{D}$ are squares of the ℓ^2-norms of the atoms of $\mathbf{D}$, and off-diagonal values are the inner products of the dictionary atoms. Choosing $\mathbf{\Sigma} \in \mathbb{R}^{m\times n}$ as in the theorem results in $\mathbf{\Sigma}^T\mathbf{\Sigma} = \mathbf{I}$. So

$$\mathbf{D}^T\mathbf{D} = (\mathbf{V\Sigma}^T\mathbf{U}^T)(\mathbf{U\Sigma V}^T) = \mathbf{V\Sigma}^T\mathbf{\Sigma V}^T = \mathbf{VV}^T = \mathbf{I}, \tag{23}$$

which shows that the columns of D are orthogonal, and are of unit ℓ^2-norms. □

Theorem 2. *Given an orthnormal matrix $\mathbf{U} \in \mathbb{R}^{m\times m}$ and choosing a tall diagonal matrix $\mathbf{\Sigma} \in \mathbb{R}^{m\times n}$ with non-negative values along the main diagonal, regardless of the choice of an orthonormal matrix $\mathbf{V} \in \mathbb{R}^{n\times n}$, the atoms of the dictionary $\mathbf{D} = \mathbf{U\Sigma V}^T$ span the same subspace.*

Proof. The result is simply established by noting that from $\mathbf{D} = \mathbf{U\Sigma V}^T$, every column of $\mathbf{D}$ is a linear combination of the columns of $\mathbf{U\Sigma}$.

□

Putting all the results of this section together, the final algorithm of learning an undercomplete dictionary for the c^{th} class of data is obtained as summarized in Algorithm 1. In Algorithm 1, $\mathbf{D}_c^0$ is the initial dictionary, *i.e.* the starting point of the algorithm, which is chosen randomly, $\mathbf{I}_{mn} \in \mathbb{R}^{m\times n}$ is a matrix with ones along the main diagonal and zeros elsewhere, in which $n < m$ is the number of dictionary atoms. This number can be chosen based on the available computational capability or by fine tunning to obtain the best classification rate on the testing data. Moreover, μ and N are the user-selected step-size and the number of iterations of the Gradient Descent step of the Gradient Projection algorithm, and $\mathbf{C}^{i-1} \triangleq [\mathbf{U}_{c,n}^{i-1} \mid \mathbf{0}_{m\times(m-n)}]$, in which $\mathbf{U}_{c,n}^{i-1}$ is a matrix composed of the first n columns of $\mathbf{U}_c^{i-1}$.

Algorithm 1 Undercomplete Dictionary Learning Algorithm.

Inputs:
$\mathbf{Y}_c \in \mathbb{R}^{m\times T}$, μ, n, N
Initialize:
$\mathbf{D}_c^0$
Calculations:
$\mathbf{U}_c^0, \mathbf{\Sigma}_c^0, (\mathbf{V}_c^0)^T \leftarrow \text{SVD}(\mathbf{D}_c^0)$
for $i = 1 : N$ **do**
$\hat{\mathbf{U}}_c^i \leftarrow \mathbf{U}_c^{i-1} + \mu \mathbf{Y}_c \mathbf{Y}_c^T \mathbf{C}^{i-1}$
$\mathbf{U}_c^i \leftarrow proj(\hat{\mathbf{U}}_c^i)$
end for
$\mathbf{D}_c \leftarrow \mathbf{U}_c^N \mathbf{I}_{mn} (\mathbf{V}_c^0)^T$
return $\mathbf{D}_c$

4. Simulations

In this section, simulation results on a synthetic dataset, the standard MNIST dataset [32] and the Fashion-MNIST dataset [33] are presented. In theses simulations, the classification accuracy, the training time, the classification time and the mean square error (MSE) are reported. The classification accuracy measures how well our model predicts the label of the samples in the testing dataset and is calculated in percent as

$$\frac{\text{Number of samples classifed correctly}}{\text{Total number of samples}} \times 100.$$

The training time is the average required time for a model to be trained, and the classification time is the average runtime required to classify a single test data, and they are used as a rough measure of computational complexity. The MSE assesses the representational ability of the resulted dictionaries. Given a set of datapoints $[\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T]$ and a dictionary $\mathbf{D}$, the MSE is defined as

$$\frac{1}{T} \sum_{i=1}^{T} \|\mathbf{y}_i - \hat{\mathbf{y}}_i\|_2^2,$$

in which $\hat{\mathbf{y}}_i = \mathbf{D}(\mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}\mathbf{y}_i$ is the best approximation of $\mathbf{y}_i$ in terms of a linear combination of the columns of $\mathbf{D}$ in the ℓ^2-norm sense.

The simulations are done using MATLAB R2018b environment on a system with Windows 10 operating system with a 4.2 GHz Intel core i7-7700K

CPU and 16 GB of RAM. Our simulations will demonstrate the performance of our method compared with SRC[6], DKSVD[7], LCKSVD-1[8], LCKSVD-2[8] and structured analysis and discriminative dictionary learning (ADDL)[13].

4.1. Simulation on Synthetic Data

For the synthetic dataset, a classification problem with ten classes is defined. In order to create the training dataset of the i^{th} class with T samples, three random matrices $\mathbf{A}_i \in \mathbb{R}^{m\times s}$, $\mathbf{B}_i \in \mathbb{R}^{s\times T}$ and $\mathbf{C}_i \in \mathbb{R}^{m\times T}$ with independent identically distributed (i.i.d) entries drawn from a probability distribution $\mathcal{N}(0,1)$ are generated. The dataset for the i^{th} class is then created as

$$\mathbf{Y}_i = \mathbf{A}_i\mathbf{B}_i + \alpha\mathbf{C}_i, \tag{24}$$

in which m and s are fixed to be 100 and 30 for these simulations. Assuming α to be zero, by creating the dataset in this manner, the datapoints of each class (*i.e.* the columns of $\mathbf{Y}_i$) lie in a particular s-dimensional subspace of $\mathbb{R}^m$, and therefore they are theoretically separable. The parameter α is then used to investigate the effects of noisy training data on our algorithm by controlling the additive noise variance.

Figure 1 depicts the MSE of the first four undercomplete dictionaries for the classification problem with synthetic data for various step-sizes μ (learning rates). The number of training data for each class is 1000, and each dictionary has 50 atoms, and 20 iterations of the GP is performed. As Fig. 1 shows, our algorithm is fairly robust to the choice of the step-size. It converges to a solution with $\mu = 0.01$ and $\mu = 1$ with relatively the same rate.

Figures 2 and 3 demonstrate the classification accuracy of our method on synthetic data for different step-sizes and a different number of training data for a various number of dictionary atoms per class. A set of 1000 test data (100 for each class) is used to calculate the accuracy in these simulations. As Fig. 2 shows, our method is again robust to the choice of the step-size.

Figure 4 shows the performance of our method when the given training data is noisy. This simulation is performed by changing the value of α in (24). For this simulation, $\alpha \in \{10, 20, 30\}$. The noise-free case is also added for comparison. As Fig. 4 shows, our method is fairly robust to the additive white Gaussian noise.

Figures 5 and 6 depict the performance of our method in comparison with SRC [6], DKSVD [7], LCKSVD-1, LCKSVD-2 [8] and ADDL[13]. Figure 5

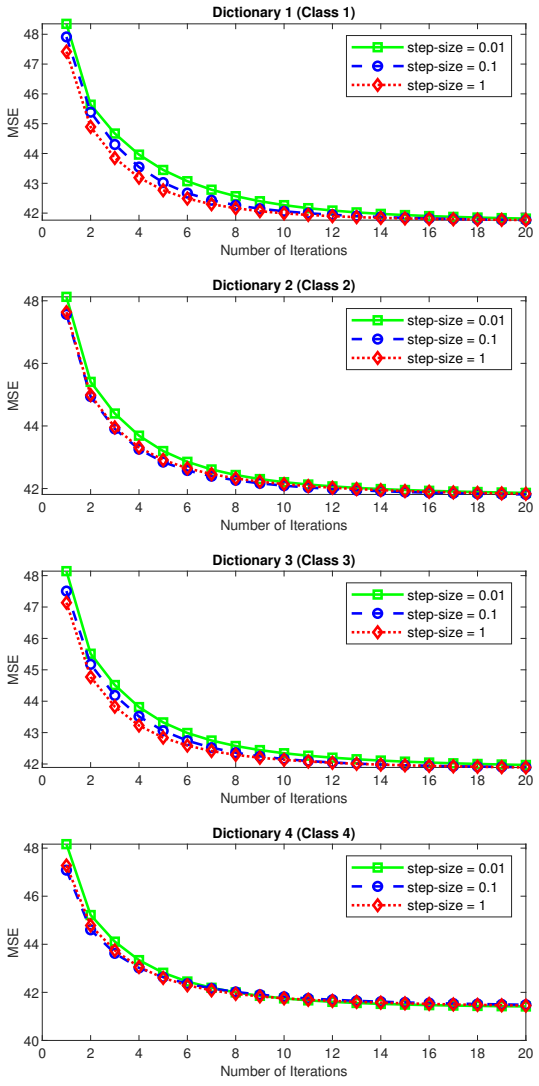

Figure 1: MSE for the first four undercomplete dictionaries for classification problem with synthetic data of Section 4.1 for various step-sizes. The number of training samples is 1000 for each class, the number of the atoms of each dictionary is 50. The GP's step-size is 0.1.

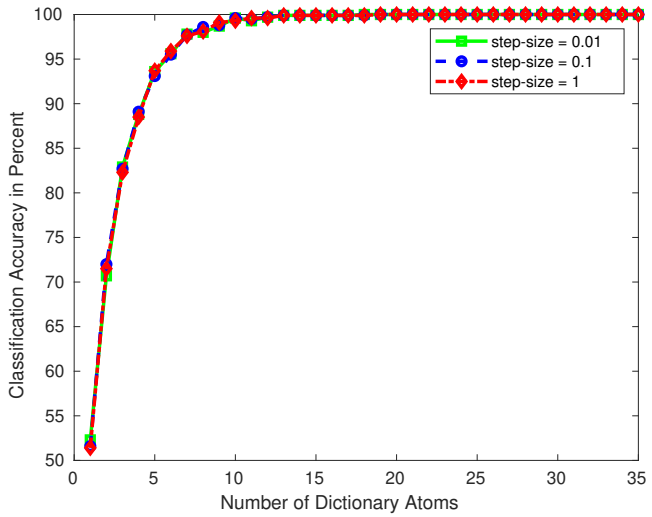

Figure 2: Classification Accuracy for various step-sizes and various number of dictionary atoms per class. The number of training samples is 1000 for each class, the number of testing samples is 100 for each class. The GP's step-size is 0.1

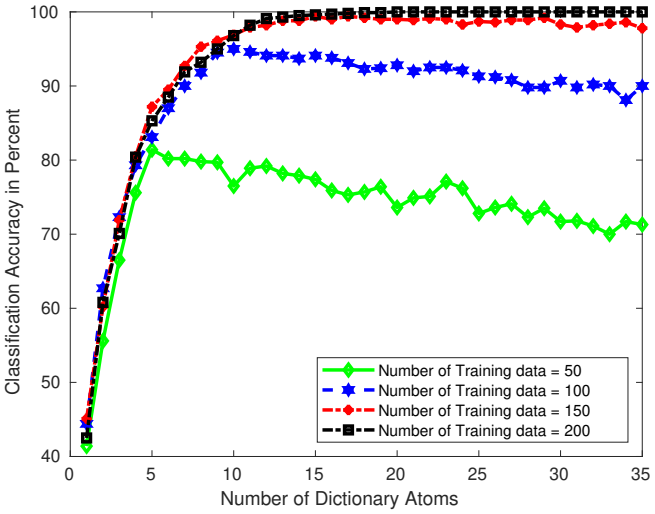

Figure 3: Classification Accuracy for various number of training data and various number of dictionary atoms per class. The number of training samples $\in [5, 10, 15, 20, 25, 30]$ for each class, the number of testing samples is 100 for each class. The GP's step-size is 0.1.

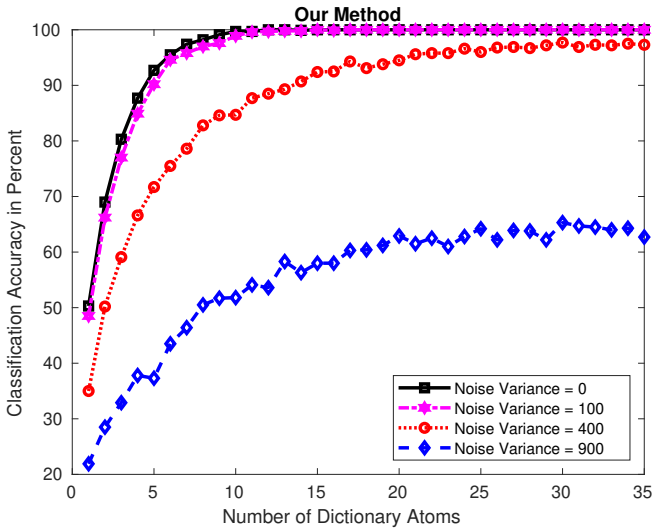

Figure 4: Classification Accuracy for various number of dictionary atoms per class with noisy training data. The noise is additive white Gaussian. The number of training samples is 1000 for each class and the number of testing samples is 100 for each class. The GP's step-size is 0.1

demonstrates the classification accuracy of our method in comparison with these methods for various dictionary sizes. As discussed, the dictionary in SRC, DKSVD, LCKSVD-1 and LCKSVD-2 is overcomplete. Therefore, there is a lower limit for the number of atoms in each submatrix $\mathbf{D}_i$. Our method does not have such a lower limitation. As Fig. 5 shows, our method has better accuracy with much fewer dictionary atoms. Figure 6 depicts the average time required to classify a single test data, which shows that our method is faster by around 4 orders of magnitude than SRC and by more than 1 order of magnitude than DKSVD, LCKSVD-1 and LCKSVD-2. The main reason for this difference is that in our method, finding the representation of the test data is achieved by projection matrix multiplication, while in SRC, solving an ℓ^1-norm minimization problem as (5) is required. This operation is highly more time consuming than a single matrix multiplication. DKSVD, LCKSVD-1 and LCKSVD-2 are faster than SRC because they use orthogonal matching pursuit (OMP) [39] for finding the sparse representation of a signal, but they are still slower than our proposed method. As Fig. 6 shows, the classification time of our method and ADDL are very close. However our method has a significantly higher classification rate according to Fig. 5 Another observation from Fig. 6 is that the rate of problem complexity growth with respect to the number of parameters of our method is less than SRC, DKSVD, LCKSVD-1 and LCKSVD-2. Figure 7 depicts the required training time of our method compared with DKSVD, LCKSVD-1 and LCKSVD-2, and ADDL. Each method is trained on the training dataset until convergence (which was 10 iterations for DKSVD, LCKSVD-1, LCKSVD-2 and our method).

4.2. Simulation on MNIST and Fashion-MNIST datasets

In this part, simulation results of our method on the MNIST [32] and Fashion-MNIS datasets are presented. MNIST is a dataset of handwritten digits (0-9) and Fashion-MNIST is a dataset of fashion objects in ten categories. Each dataset consists of 60000 training samples and 10000 testing samples. Each sample is a 28×28 image. Figures 8 and 9 show 16 samples of each dataset. Figures 10 and 11 demonstrate the classification accuracy of our method compared with SRC, DKSVD, LCKSVD-1, LCKSVD-2 and ADDL for various number of dictionary atoms per class for MNIST and Fashion-MNIST datasets, respectively. For these simulations, each image sample in each dataset is transformed to a vector by placing its columns sequentially. The classification accuracy for these simulations is obtained

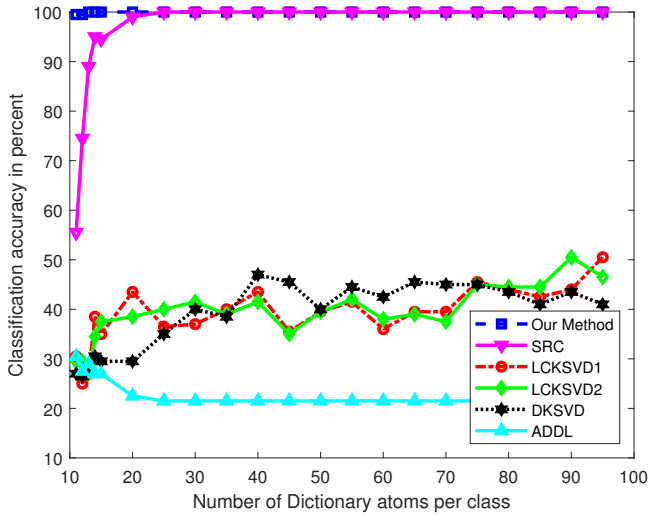

Figure 5: Classification Accuracy of our method in comparison with SRC, DKSVD, LCKSVD-1, LCKSVD-2, and ADDL for various number of dictionary atoms per class. The number of training samples is 1000 for each class. The number of testing samples is 100 for each class. The step-size of GP is 0.1.

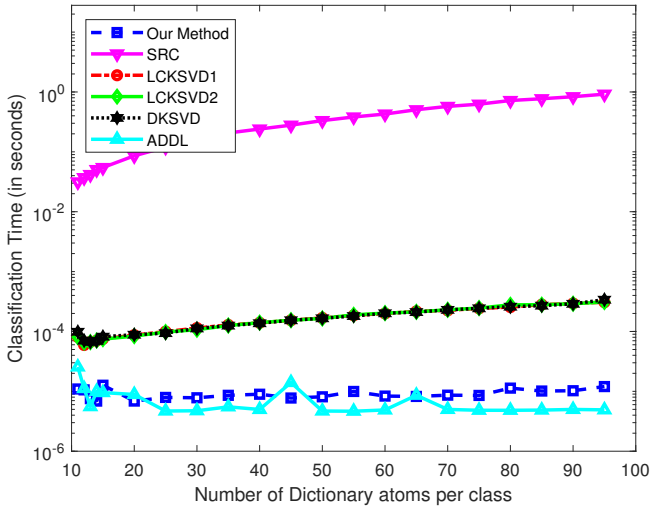

Figure 6: Required time to classify a single test data for various number of dictionary atoms per class in logarithmic scale. The number of training samples is 1000 for each class, and the number of testing samples is 100 for each class. The step-size of GP is 0.1.

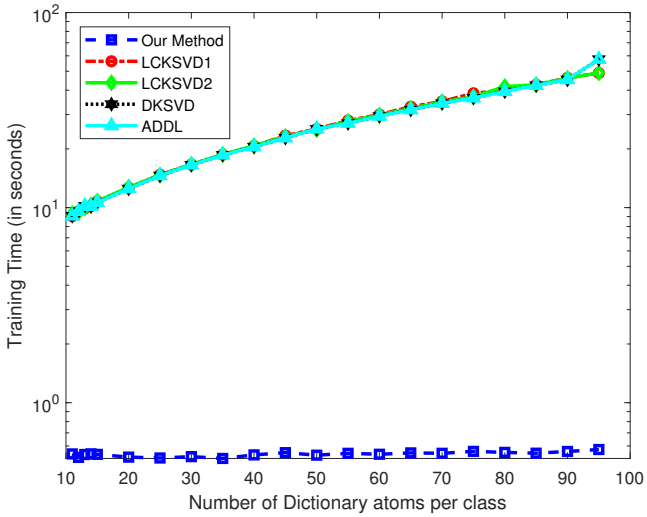

Figure 7: Required training time for various number of dictionary atoms per class in logarithmic scale. The number of training samples is 1000 for each class, and the number of testing samples is 100 for each class. The step-size of GP is 0.1.

by evaluating 200 randomly selected samples from the testing data of each dataset. In these simulations, the step-size of GP is 0.1 and each method is trained until convergence. Figures 10 and 11 depict the classification accuracy of our method compared with SRC, DKSVD, LCKSVD-1, LCKSVD-2 and ADDL. They also show that the methods based on overcomplete dictionaries require a minimum number of atoms for each class to ensure that their resulted dictionary is overcomplete, while our method does not have such a limitation and can operate with very small number of atoms starting from one. These simulations also demonstrate that increasing the number of dictionary atoms does not always increase the classification performance. This is because a more complex model (*i.e.* with more trainable parameters) is more vulnerable to overfitting [16], that is, although it would fit to the training data better, it may fail to generalize to the testing data.

Figures 12 and 13 depict the classification time of our method compared with SRC, DKSVD, LCKSVD-1, LCKSVD-2 and ADDL for MNIST and Fashion-MNIST datasets, respectively, and Figures 14 and 15 demonstrate the required training time of our method compared with DKSVD, LCKSVD-1, LCKSVD-2 and ADDL on both datasets (note that SRC has no training).

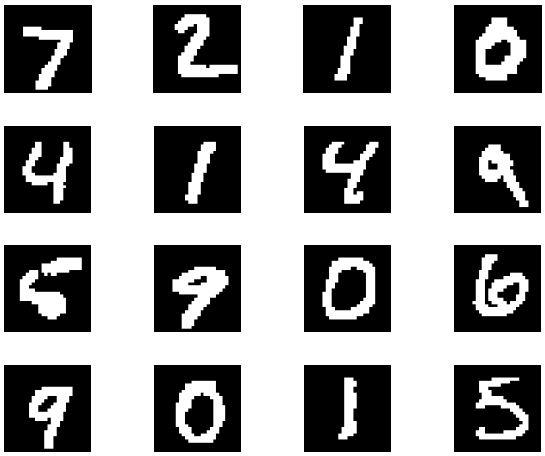

Figure 8: 16 samples of the MNIST dataset.

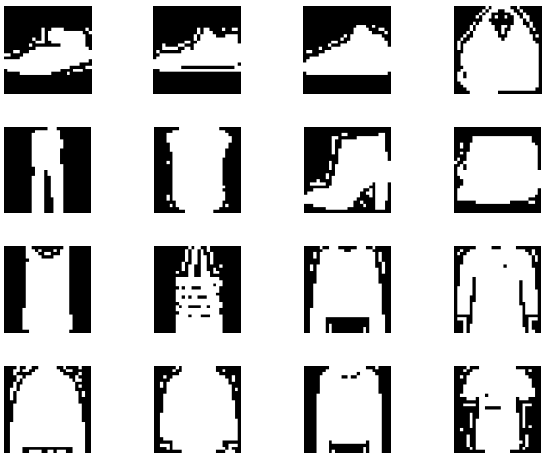

Figure 9: 16 samples of the Fashion-MNIST dataset.

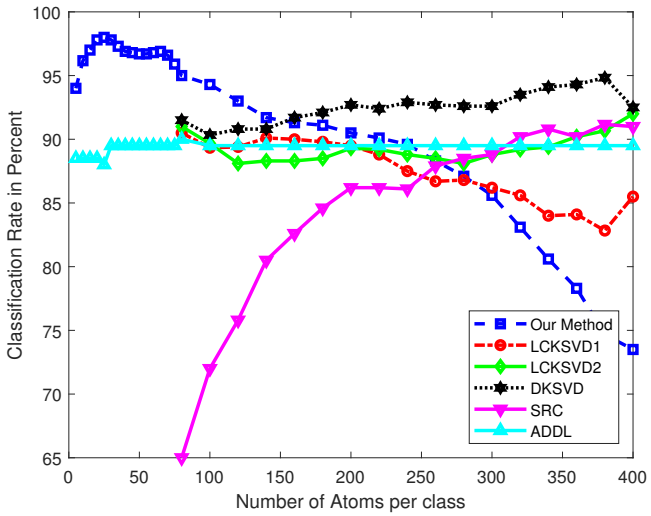

Figure 10: Classification Accuracy of our method in comparison with SRC, DKSVD, LCKSVD-1, LCKSVD-2 and ADDL on MNIST dataset for various number of dictionary atoms per class. The number of testing samples is 200. The step-size of GP is 0.1.

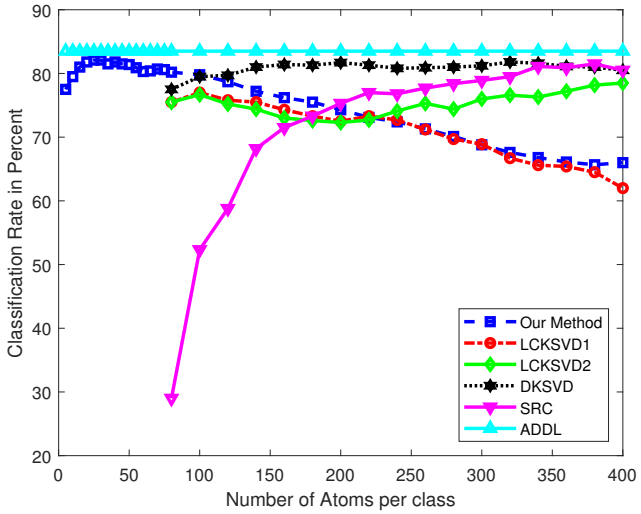

Figure 11: Classification Accuracy of our method in comparison with SRC, DKSVD, LCKSVD-1, LCKSVD-2 and ADDL on Fashion-MNIST dataset for various number of dictionary atoms per class. The number of testing samples is 200. The step-size of GP is 0.1.

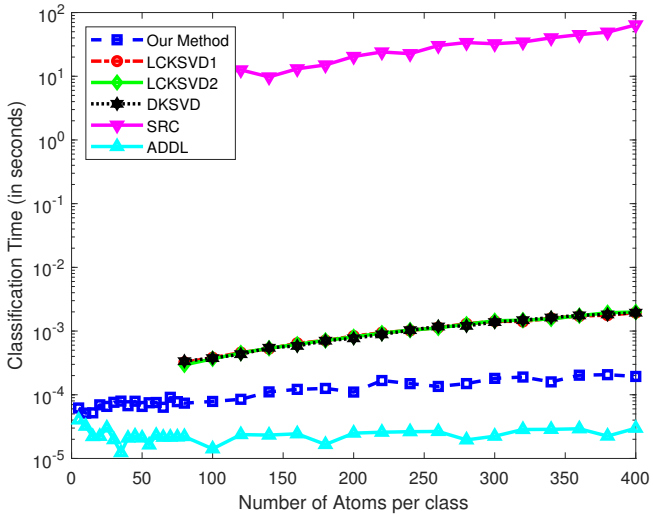

Figure 12: Classification time of our method in comparison with SRC, DKSVD, LCKSVD-1, LCKSVD-2 and ADDL on MNIST dataset for various number of dictionary atoms per class. The number of testing samples is 200. The step-size of GP is 0.1.

As these figures show, our method has a significantly lower computational cost in both training and classification phases compared with SRC, DKSVD, LCKSVD-1 and LCKSVD-2. The reasons for this is that our method is based on undercomplete dictionaries rathar than overcomplete dictionaries, thus there are less parameters involved in computations. Furthermore, finding the representation of data over an undercomplete dictionary is readily achieved by a matrix multiplication, while in the sparse representation based methods, problem (4) should be solved, which is computationally demanding. For these reasons, our method has a highly less computational volume in both training and testing phases. As the Figs. 12, 14, 13 and 15 show, ADDL is slightly faster in training and testing time compared with our method. However, the simulations show that our method outperforms ADDL in classification rate on both synthetic and MNIST datasets.

5. Conclusions

DL-based classification methods with both representation and classification goals suffer from the computational complexity. In this paper, a DL-

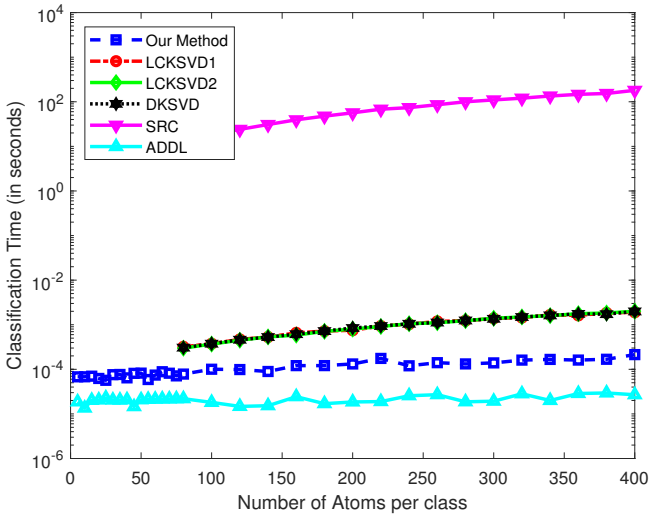

Figure 13: Classification time of our method in comparison with SRC, DKSVD, LCKSVD-1, LCKSVD-2 and ADDL on Fashion-MNIST dataset for various number of dictionary atoms per class. The number of testing samples is 200. The step-size of GP is 0.1.

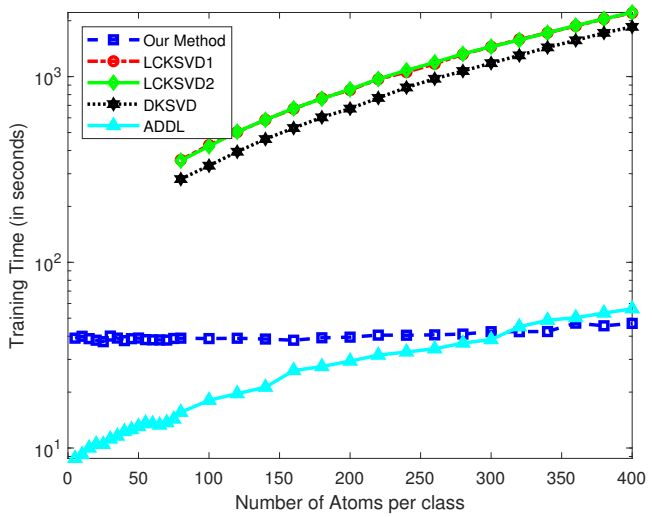

Figure 14: Training time of our method in comparison with DKSVD, LCKSVD-1, LCKSVD-2 and ADDL on MNIST dataset for various number of dictionary atoms per class. The number of testing samples is 200. The step-size of GP is 0.1.

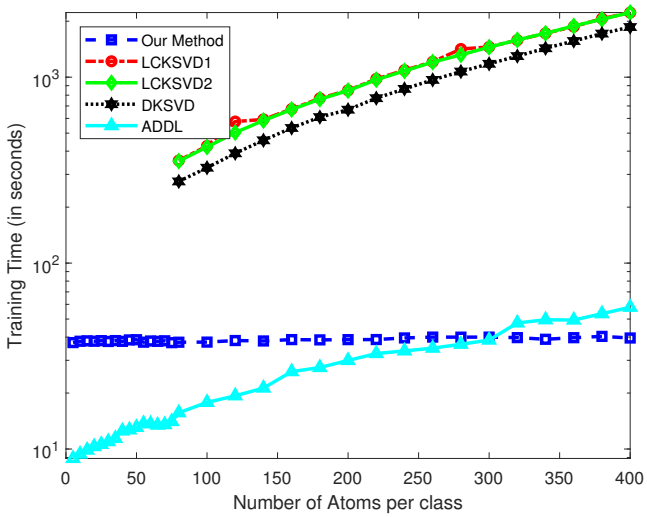

Figure 15: Training time of our method in comparison with DKSVD, LCKSVD-1, LCKSVD-2 and ADDL on Fashion-MNIST dataset for various number of dictionary atoms per class. The number of testing samples is 200. The step-size of GP is 0.1.

based classification method was introduced, which is based on undercomplete dictionaries and has the sole purpose of classification. We used singular value decomposition to provide an algorithm for the resulted optimization problem. Simulation results demonstrated that our method has a significantly lower computational cost compared with a number of related methods, while achieving even better classification accuracies.

References

[1] C. C. Aggarwal, Data classification: Algorithms and applications (2014).

[2] J. Mairal, F. Bach, J. Ponce, G. Sapiro, A. Zisserman, Discriminative learned dictionaries for local image analysis, in: 2008 IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8. doi:10.1109/CVPR.2008.4587652.

[3] K. Huang, S. Aviyente, Sparse representation for signal classification, Advances in neural information processing systems 19 (2006).

[4] Y.-L. Boureau, F. Bach, Y. LeCun, J. Ponce, Learning mid-level features for recognition, in: 2010 IEEE computer society conference on computer vision and pattern recognition, IEEE, 2010, pp. 2559–2566.

[5] J. Mairal, M. Leordeanu, F. Bach, M. Hebert, J. Ponce, Discriminative sparse image models for class-specific edge detection and image interpretation, in: European conference on computer vision, Springer, 2008, pp. 43–56.

[6] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (2) (2009) 210–227. doi:10.1109/TPAMI.2008.79.

[7] Q. Zhang, B. Li, Discriminative k-svd for dictionary learning in face recognition, in: 2010 IEEE computer society conference on computer vision and pattern recognition, IEEE, 2010, pp. 2691–2698.

[8] Z. Jiang, Z. Lin, L. S. Davis, Learning a discriminative dictionary for sparse coding via label consistent k-svd, in: CVPR 2011, IEEE, 2011, pp. 1697–1704.

[9] R. Rubinstein, A. M. Bruckstein, M. Elad, Dictionaries for sparse representation modeling, Proceedings of the IEEE 98 (6) (2010) 1045–1057.

[10] M. Sadeghi, M. Babaie-Zadeh, C. Jutten, Dictionary learning for sparse representation: A novel approach, IEEE Signal Processing Letters 20 (12) (2013) 1195–1198.

[11] I. Tos, P. Frossard, et al., Dictionary learning—what is the right representation for my signal?, IEEE Signal Process. Mag. 28 (7) (2011) 27–38.

[12] Z. Zhang, W. Jiang, Z. Zhang, S. Li, G. Liu, J. Qin, Scalable block-diagonal locality-constrained projective dictionary learning, in: IJCAI International Joint Conference on Artificial Intelligence, Vol. 2019, AAAI Press, 2019, pp. 4376–4382.

[13] Z. Zhang, W. Jiang, J. Qin, L. Zhang, F. Li, M. Zhang, S. Yan, Jointly learning structured analysis discriminative dictionary and analysis multiclass classifier, IEEE transactions on neural networks and learning systems 29 (8) (2017) 3798–3814.

[14] Z. Zhang, Y. Sun, Z. Zhang, Y. Wang, L. Wu, M. Wang, Mdpl-net: Multi-layer dictionary learning network with added skip dense connections, in: 2020 IEEE International Conference on Data Mining (ICDM), IEEE, 2020, pp. 811–820.

[15] Z. Zhang, Y. Sun, Y. Wang, Z. Zha, S. Yan, M. Wang, Convolutional dictionary pair learning network for image representation learning, in: ECAI 2020, IOS Press, 2020, pp. 1642–1649.

[16] I. Goodfellow, Y. Bengio, A. Courville, Deep learning, MIT press, 2016.

[17] X. Zhu, X.-Y. Jing, X. You, W. Zuo, S. Shan, W.-S. Zheng, Image to video person re-identification by learning heterogeneous dictionary pair with feature projection matrix, IEEE Transactions on Information Forensics and Security 13 (3) (2017) 717–732.

[18] X. Zhu, X.-Y. Jing, L. Yang, X. You, D. Chen, G. Gao, Y. Wang, Semi-supervised cross-view projection-based dictionary learning for video-based person re-identification, IEEE Transactions on Circuits and Systems for Video Technology 28 (10) (2017) 2599–2611.

[19] X.-Y. Jing, X. Zhu, F. Wu, X. You, Q. Liu, D. Yue, R. Hu, B. Xu, Super-resolution person re-identification with semi-coupled low-rank discriminant dictionary learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 695–704.

[20] F. Ma, X.-Y. Jing, X. Zhu, Z. Tang, Z. Peng, True-color and grayscale video person re-identification, IEEE Transactions on Information Forensics and Security 15 (2019) 115–129.

[21] F. Wu, X.-Y. Jing, X. You, D. Yue, R. Hu, J.-Y. Yang, Multi-view low-rank dictionary learning for image classification, Pattern Recognition 50 (2016) 143–154.

[22] X. Jia, X.-Y. Jing, X. Zhu, S. Chen, B. Du, Z. Cai, Z. He, D. Yue, Semi-supervised multi-view deep discriminant representation learning, IEEE Transactions on Pattern Analysis and Machine Intelligence 43 (7) (2021) 2496–2509. doi:10.1109/TPAMI.2020.2973634.

[23] X.-Y. Jing, F. Wu, X. Zhu, X. Dong, F. Ma, Z. Li, Multi-spectral low-rank structured dictionary learning for face recognition, Pattern Recognition 59 (2016) 14–25, compositional Models and Structured Learning for Visual Recognition. doi:https://doi.org/10.1016/j.patcog.2016.01.023.
URL https://www.sciencedirect.com/science/article/pii/S0031320316000443

[24] X.-Y. Jing, F. Wu, Z. Li, R. Hu, D. Zhang, Multi-label dictionary learning for image annotation, IEEE Transactions on Image Processing 25 (6) (2016) 2712–2725.

[25] D. L. Donoho, M. Elad, Optimally sparse representation in general (nonorthogonal) dictionaries via $\ell 1$ minimization, Proceedings of the National Academy of Sciences 100 (5) (2003) 2197–2202.

[26] Z. Zhang, Y. Xu, J. Yang, X. Li, D. Zhang, A survey of sparse representation: algorithms and applications, IEEE access 3 (2015) 490–530.

[27] M. Elad, Sparse and redundant representations: from theory to applications in signal and image processing, Vol. 2, Springer, 2010.

[28] D. L. Donoho, For most large underdetermined systems of linear equations the minimal $\ell 1$-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) (2006) 797–829.

[29] E. J. Candes, J. K. Romberg, T. Tao, Stable signal recovery from incomplete and inaccurate measurements, Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences 59 (8) (2006) 1207–1223.

[30] E. J. Candes, T. Tao, Near-optimal signal recovery from random projections: Universal encoding strategies?, IEEE transactions on information theory 52 (12) (2006) 5406–5425.

[31] S. S. Chen, D. L. Donoho, M. A. Saunders, Atomic decomposition by basis pursuit, SIAM review 43 (1) (2001) 129–159.

[32] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (11) (1998) 2278–2324.

[33] H. Xiao, K. Rasul, R. Vollgraf, Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, arXiv preprint arXiv:1708.07747 (2017).

[34] D. I. Moody, S. P. Brumby, J. C. Rowland, C. Gangodagamage, Undercomplete learned dictionaries for land cover classification in multispectral imagery of arctic landscapes using cosa: clustering of sparse approximations, in: Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XIX, Vol. 8743, International Society for Optics and Photonics, 2013, p. 87430B.

[35] D. C. Lay, S. R. Lay, J. J. McDonald, Linear algebra and its applications, Pearson, 2016.

[36] S. Wright, J. Nocedal, et al., Numerical optimization, Springer Science 35 (67-68) (1999) 7.

[37] D. Bertsekas, Nonlinear Programming, Athena Scientific, 1999.

[38] G. H. Golub, C. F. Van Loan, Matrix computations, JHU press, 2013.

[39] Y. C. Pati, R. Rezaiifar, P. S. Krishnaprasad, Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition, in: Proceedings of 27th Asilomar conference on signals, systems and computers, IEEE, 1993, pp. 40–44.