Survey of Dictionary Learning *

Kai-Chieh Hsu, Ching-Yao Chou and Chieh-Fang Teng

B03901026, F03943134 and D06943020

Abstract

Modeling data with linear combination of a *few* elements from a *learned* dictionary has been the focus of recent research in machine learning and signal processing. In this final project, we present the statistical guarantee and the state-of-the-art optimization algorithms of reconstructive dictionary learning (RDL) for restoration task and predictive dictionary learning (PDL) for classification/regression task. Besides, we introduce sparse coding algorithm which plays an important role in dictionary learning. In addition to organize the materials from the reference papers, we interpret in our own words and compare different works at the end of each section, such as generalization bound and optimization efficiency.

Index Terms — dictionary learning, sparse coding, generalization bound, stability, optimization efficiency

1 Introduction

1.1 Sparse Linear Model

Concretely, consider a signal $\mathbf{x} \in \mathbb{R}^n$. We say that it admits a sparse approximation over a *dictionary* $\mathbf{D} = [\mathbf{d}_1 \ \cdots \ \mathbf{d}_d] \in \mathbb{R}^{n \times d}$, with d columns refered as *atoms*, when one can find a linear combination of a "few" atoms from $\mathbf{D}$ that is "close" to the signal $\mathbf{x}$, as shown in 1. For a simple explanation of sparse linear model, the signal and images in Fig. 2(a) can be seemed as $\mathbf{x}$ and they can be composed by the different frequency signals and many small blocks in Fig. 2(b) which are seemed as $\mathbf{D}$.

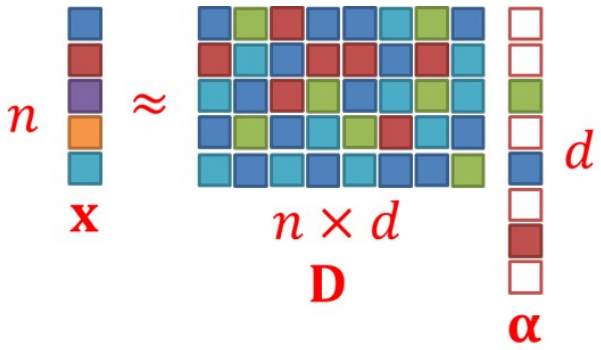
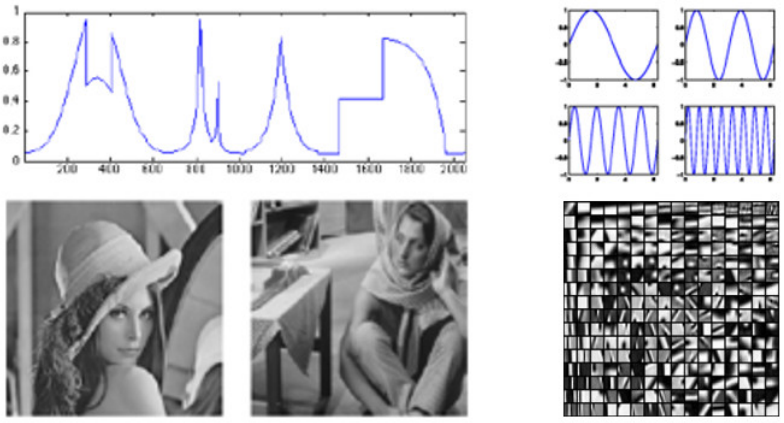


Figure 1: Signal $\mathbf{x}$ can be represented with a "few" atoms from dictionary $\mathbf{D}$

*This work is the final project of Team 16 with topic: Matrix Factorization



(a) Signal $\mathbf{x}$ (b) Dictionary $\mathbf{D}$

Figure 2: Examples of sparse linear model

1.2 Notations

Vectors are denoted by boldface lower case letters and matrices by boldface upper case. For $q \geq 1$, the ℓ_q-norm of vector $\mathbf{x} \in \mathbb{R}^n$ is defined as $\|\mathbf{x}\|_q \triangleq (\sum_{i=1}^n |x_i|^q)^{1/q}$, where x_i is the i^{th} entry of $\mathbf{x}$. The ℓ_0-norm of vector $\mathbf{x}$ is defined as the number of nonzero elements in $\mathbf{x}$. The inner product of vectors is denoted by $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^T\mathbf{y}$. Let $[d] := \{1, \cdots, d\}$ for $d \in \mathbb{N}$ and $supp(\boldsymbol{\alpha}) := \{j \in [d] : \alpha_j \neq 0\}$ for $\boldsymbol{\alpha} \in \mathbb{R}^d$. $\mathbf{X}$ is a matrix in $\mathbb{R}^{n\times m}$ and $\Lambda \subseteq [m]$, $\mathbf{X}_\Lambda$ is the matrix in $\mathbb{R}^{n\times|\Lambda|}$ whose columns are those of $\mathbf{X}$ indexed by Λ. Similarly, $\mathbf{x}_\Lambda$ is the sub-vector with elements indexed by Λ. We also denote $\xi_k(\mathbf{X})$ as the k^{th} eigenvalue of $\mathbf{X}$.

Throughout this paper, suppose that the sample $\mathbf{x} \in \mathbb{R}^n$ and assume training data set is of m samples concatenated to form a matrix, denoted as $\mathbf{X} = [\mathbf{x}_1 \ \cdots \ \mathbf{x}_m] \in \mathbb{R}^{n\times m}$. In addition, the dictionary is of dimension $\mathbf{D} = [\mathbf{d}_1 \ \cdots \ \mathbf{d}_d] \in \mathbb{R}^{n\times d}$ and the sparse coding vector of $\mathbf{x}$ depending on $\mathbf{D}$ is denoted by $\boldsymbol{\alpha}_\mathbf{D}(\mathbf{x}) \in \mathbb{R}^d$.

The rest of this paper is organized as follows: Section 2 illustrates the reconstructive dictionary learning framework with applications, generalization bound and optimization algorithm. Section 3 demonstrates the predictive dictionary learning framework with applications, generalization guarantee and optimization algorithm. Section 4 compares several algorithm to solve ℓ_1-regularization of sparse coding. Finally, section 5 draws our conclusion. Appendix A summarizes the glossary used in the whole work, while the Appendix B describes the work division of out team.

2 Reconstructive Dictionary Learning

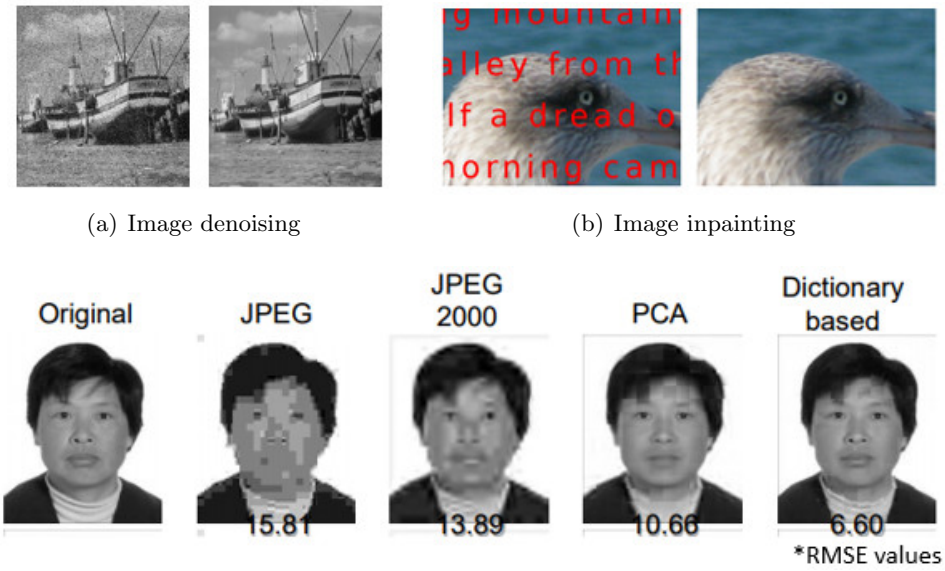
2.1 Problem Formulation and Applications

Learning the dictionary instead of using off-the-shelf (predefined) bases has been shown to dramatically improve the performance. Although some of the learned dictionary elements may sometimes "look like" wavelets (or Gabor filters), they can further tuned to the input signals, leading to much better results in practice. The goal of dictionary learning can be formulated as the following optimization problem below:

$$\min_{\mathbf{D},\mathbf{A}} \|\mathbf{X} - \mathbf{D}\mathbf{A}\|_F^2, \ \mathbf{A} = [\boldsymbol{\alpha}_\mathbf{D}(\mathbf{x}_1) \ \boldsymbol{\alpha}_\mathbf{D}(\mathbf{x}_2) \ ... \ \boldsymbol{\alpha}_\mathbf{D}(\mathbf{x}_m)], \ s.t. \ \|\boldsymbol{\alpha}\|_0 \leq k, \ \forall i \in [m] \tag{1}$$

The most common approach for (1) is to optimize between $\mathbf{A}$ (sparse coding) and $\mathbf{D}$ (dictionary update)

alternatively. The learning of compact representations adapted to restoration tasks has a variety of applications in image and medical signal processing, such as denoising, inpainting and compression as shown in Fig. 3.



(a) Image denoising

(b) Image inpainting

(c) Image compression

Figure 3: Dictionary learning has a variety of applications

2.2 Generalization Bound

In [1, 2, 3], they provide sample complexity estimations to uniformly control how much the empirical average deviates from the best function. [1] presents a general coding method where data drawn from a distribution Π on the unit ball of a Hilbert space and are represented by finite dimensional coding vectors as shown in Fig. 4. The reconstruction error is defined as below:

$$\ell_u(\mathbf{x}) = \min_{\boldsymbol{\alpha}\in\mathbb{R}^d} \|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}_{\mathbf{D}}(\mathbf{x})\|_2^2, \ s.t. \ \|\boldsymbol{\alpha}\|_1 \leq \lambda^{-1}$$

which means the deviation between original and the decoding vector and λ is the sparse constraints. Whenever the codebook is compact and $\mathbf{D}$ is bounded, this approach is justified by the following high-probability, uniform bound on the expected reconstruction error.

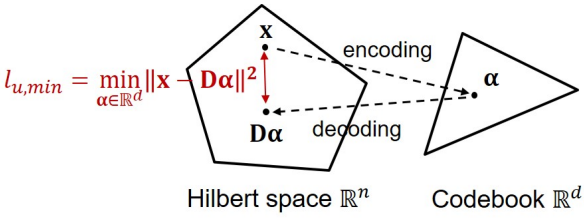


Figure 4: Data $\mathbf{x} \in \mathbb{R}^n$ in Hilbert space are represented via linear map ($\mathbf{D} \in \mathbb{R}^{n\times d}$) of prescribed set of code $\boldsymbol{\alpha} \in \mathbb{R}^d$

Theorem 1 (Generalization Error, Theorem 1.2 in [1]). *With probability at least $1-\delta$ in the observed data*

$(\mathbf{x}_1, ..., \mathbf{x}_m) \sim \Pi^m$, *we have for every* $\mathbf{D} \in \mathcal{D}$ *that*

$$\mathbb{E}_{\mathbf{x}\sim\Pi}\ \{\ell_u(\mathbf{x})\} - \frac{1}{m}\sum_{i=1}^{m} l_u(\mathbf{x}_i) \leq \frac{d}{\sqrt{m}}\Big(14\lambda^{-1} + \frac{b}{2}\sqrt{\ln\left(16m\lambda^{-2}\right)}\Big) + b\sqrt{\frac{\ln 1/\delta}{2m}}.$$

The upper bound of estimation error in Theorem 1 is mainly via two approaches in terms of the sample size m, the properties of the sets of codebook, and linear map $\mathbf{D} \in \mathcal{D}$. The first approach is based on a direct bound for the *Rademacher average* of the loss class induced by the reconstruction error. The second approach is to approximate the union with a finite union via *covering numbers*.

In [2], they develop generalization bounds on the quality of the learned dictionary for the constraints on the coefficient selection, as measured by the expected ℓ_2 error. For ℓ_1 regularized coefficient selection, they provide a generalization error bounds of order: $\mathcal{O}\left(\sqrt{\frac{nd\log(mk)}{m}}\right)$, which uses the covering number bound and a bounded differences concentration inequality. k is the ℓ_0 sparse constraint.

Theorem 2 (Generalization Error, Theorem 7 in [2])**.** *Let* $\lambda > e/4$*, with* Π *a distribution on* $\mathbb{S}^{n-1}$*. Then with probability at least* $1-e^{-x}$ *over the* m *samples drawn according to* Π*, for all* $\mathbf{D}$ *with unit length columns:*

$$\mathbb{E}_{\mathbf{x}\sim\Pi} l_{u,min}(\mathbf{x}) - \frac{1}{m}\sum_{i=1}^{m} l_{u,min}(\mathbf{x}_i) \leq \sqrt{\frac{nd\log(4\sqrt{m}k)}{2m}} + \sqrt{\frac{\mathbf{x}}{2m}} + \sqrt{\frac{4}{m}}.$$

The spirit of [3] is similar to [1, 2], and it focuses on the relation between the sample complexity and the empirical risk. The contribution of [3] is to generalize the penalty functions and data distributions. The class of penalty functions only need to be non-negative, lower semi-continuous, and coercive which is more generalized. Besides, they also relax the assumption of the training data beyond unit ball in [1] ,[2]. They also derive Lipschitz constant ρ from penalty function g for sharper bound.

Theorem 3 (Generalization Error, Theorem 1 in [3])**.** *Consider* $\rho > \rho_\Pi(\bar{g})$ *and define*

$$\beta \triangleq h \cdot \max\left(\log\frac{2\rho C}{c},\ 1\right),$$

$$Bound_m(g, \mathcal{D}, \mathcal{B}) \triangleq 3c\sqrt{\frac{\beta\log m}{m}} + c\sqrt{\frac{\beta + \mathbf{x}}{m}}.$$

Then, given $0 \leq \mathbf{x} \leq mT^2 - \beta\log m$*, we have: except with probability at most* $\Lambda_n(L) + 2e^{-\mathbf{x}}$*,*

$$\sup_{\mathbf{D}\in\mathcal{D}} |\hat{L}_{\mathbf{X}}(\mathbf{D}) - \mathbb{E}_{\mathbf{x}} l_{\mathbf{x}}(\mathbf{D})| \leq Bound_m(g, \mathcal{D}, \mathcal{B})$$

Note that $\Lambda_n(L)$ *is primarily characterized by the penalty function* g *and the class of probability distributions* Π*, while the constants* $C, h \geq 1$ *depend on the class of dictionaries* $\mathcal{D}$*, and* $c > 0, 0 < T < \infty$ *depend on the class of probability distributions* $\mathcal{B}$*.*

From Theorem 3, we can find out the order of estimation error is almost the same as [1, 2], which is proportional to $O\left(\sqrt{\frac{\log m}{m}}\right)$. For better understanding of [1, 2, 3], we compare this three works in Table 1.

Table 1: The comparison between different works of generalization bound

Work	A. Maurer & M. Pontil [1]	D. Vainsencher [2]	R. Gribonval [3]
Bound	$O\left(d\sqrt{\frac{\log(m\lambda^{-2})}{m}}\right)$	$O\left(\sqrt{\frac{nd\log(mk)}{m}}\right)$	$Bound_m\left(g,\mathcal{D},\mathcal{B}\right)=O\left(\sqrt{\frac{\log\rho\log m}{m}}\right)$
Sparse Constraints	indicator function		Extend to other penalty function
	$\|\boldsymbol{\alpha}\|_1 \leq \lambda^{-1}$	$\|\boldsymbol{\alpha}\|_0 \leq k$	
Data Distribution	Unit Ball		Extend to more complex model (sub-Gaussian)
Approach	Uniform Convergence		Extend to consider Lipschitzness

2.3 Optimization Algorithm

2.3.1 Method of Optimal Directions (MOD)

MOD is one of the first methods introduced to tackle the sparse dictionary learning problem. The core idea of this algorithm is to solve the minimization problem as depict in (1). This method follows closely to the K-means algorithm, with a sparse coding stage that uses either orthogonal matching pursuit (OMP) or focal underdetermined system solver (FOCUSS) followed by an update of the dictionary. The main contribution of MOD is the simple way of updating the dictionary which gives the optimal adjustment of the atoms in each iteration and provides better convergence properties than the old method. The process is shown below:

- **Sparse Coding** (*ℓ_0-based method*)
  The first step finds the coefficients given the dictionary referred to as sparse coding.

$$\boldsymbol{\alpha}_{\mathbf{D}_t}(\mathbf{x}_i) \triangleq \underset{\boldsymbol{\alpha}\in\mathbb{R}^d}{\operatorname{argmin}} \|\mathbf{x}_i - \mathbf{D}_t\boldsymbol{\alpha}\|_2^2, \text{ s.t. } \|\boldsymbol{\alpha}\|_0 \leq k, \ \forall i \in [m]$$

- **Update Dictionary** (*whole dictionary*)
  With the assumption that $\mathbf{A}_t$ is fixed, they seek an update of $\mathbf{D}_t$ such that error $\|\mathbf{X} - \mathbf{D}\mathbf{A}_t\|_F^2$ is minimized. By taking the derivative of error with respect to $\mathbf{D}$, the derived update approach is shown below:

$$\mathbf{D}_{t+1} = \mathbf{X}\mathbf{A}_t^T(\mathbf{A}_t\mathbf{A}_t^T)^{-1}, \ \mathbf{A}_t = [\boldsymbol{\alpha}_{\mathbf{D}_t}(\mathbf{x}_1) \ \boldsymbol{\alpha}_{\mathbf{D}_t}(\mathbf{x}_2) \ \cdots \ \boldsymbol{\alpha}_{\mathbf{D}_t}(\mathbf{x}_m)]$$

The above process is iteratively repeated until convergence. MOD has been proved to be a very efficient method for low-dimensional input data $\mathbf{x}$. However, due to the high complexity of the matrix-inversion operation, this shortcoming has inspired the development of other dictionary learning methods.

2.3.2 K-SVD

Most works of dictionary learning are mainly focus on finding the best sparse signals respect to a given dictionary. However, in this paper they want to adapt dictionaries to achieve better sparse signal represen-

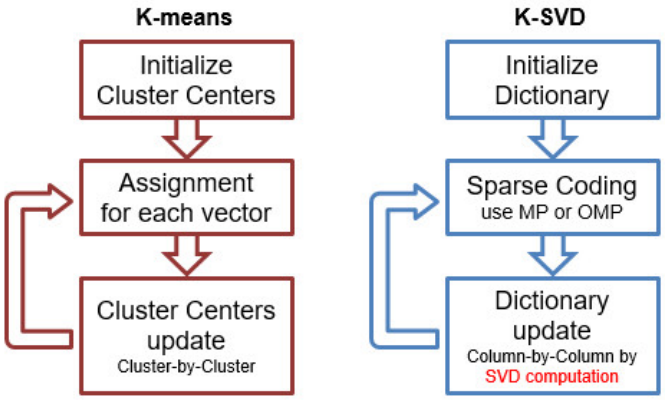


Figure 5: The comparison between K-means and K-SVD

tations. Therefore, they propose the K-SVD algorithm which generalizes the K-means clustering process. In K-means algorithm, each sample is only represented by one of the learned vectors ($\mathbf{d}_i,\ \forall i \in [d]$). However, in sparse representations, each sample is represented as a linear combination of the learned vectors. Therefore, sparse representations can be seemed as a generalization of the clustering problem. The comparison between K-means and K-SVD is shown in Fig. 5.
K-SVD is also an iterative method alternates between sparse coding of the samples based on the current dictionary and a process of updating the dictionary atoms to better fit the data. The detail of K-SVD is described below:

- **Sparse Coding** (*ℓ_0-based method*)
  The first step finds the coefficients given the dictionary which is the same as MOD.

$$\boldsymbol{\alpha}_{\mathbf{D}_t}(\mathbf{x}_i) \triangleq \underset{\boldsymbol{\alpha}\in\mathbb{R}^d}{\operatorname{argmin}} \|\mathbf{x}_i - \mathbf{D}_t\boldsymbol{\alpha}\|_2^2,\ \text{s.t. } \|\boldsymbol{\alpha}\|_0 \leq k,\ \forall i \in [m]$$

- **Update Dictionary** (*column-by-column update with active data*)
  The biggest difference between MOD and K-SVD is the method of dictionary update. In MOD, the whole dictionary is updated simultaneously. However, in K-SVD, they update the dictionary column-by-column by SVD computation which has smaller overhead than MOD. Besides, the update of the dictionary atoms is combined with an update of the sparse representation which accelerates the convergence rate. The update equation is described below and the schematic graph of dictionary update is shown in Fig. 6:

$$\min_{\mathbf{d}_v} \|\mathbf{X} - \mathbf{D}_t\mathbf{A}_t\|_F^2 = \min_{\mathbf{d}_v} \left\|\mathbf{X} - \sum_{j=1}^{d} \mathbf{d}_j\boldsymbol{\alpha}_j^T\right\|_F^2 = \min_{\mathbf{d}_v} \left\|\left(\mathbf{X} - \sum_{j\in[d],\ j\neq v} \mathbf{d}_j\boldsymbol{\alpha}_j^T\right) - \mathbf{d}_v\boldsymbol{\alpha}_v^T\right\|_F^2$$
$$\triangleq \min_{\mathbf{d}_v} \left\|\mathbf{E}_v - \mathbf{d}_v\boldsymbol{\alpha}_v^T\right\|_F^2,\quad \forall v \in [d]$$

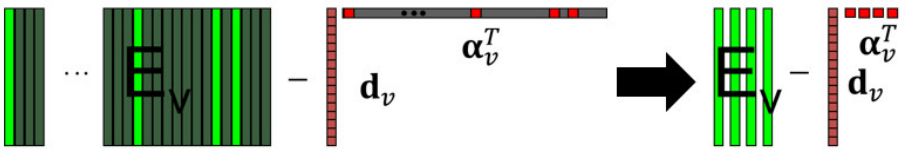


Figure 6: The schematic graph of K-SVD for dictionary update

This algorithm has been considered to be standard for dictionary learning and used in a variety of applications.

2.3.3 Online Dictionary Learning (ODL)

Unlike iterative *batch* based algorithm accessing the whole training dataset at each iteration in order to achieve empirical risk minimization, *online* dictionary learning (ODL) optimizes empirical risk with stochastic approximation, and thus ODL has low memory consumption, lower computational cost and scales up gracefully to large-scale data sets with millions of training samples

The algorithm is summarized in Algorithm 1. To prevent $\mathbf{D}$ from being arbitrarily large (which would lead to arbitrary small values of $\boldsymbol{\alpha}$), it is common to constrain its columns $\mathbf{d}_1, \cdots, \mathbf{d}_d$ to have ℓ_2-norm less than or equal to one. We will call $\mathcal{D}$ the convex set of matrices satisfying this constraints:

$$\mathcal{D} \triangleq \{\mathbf{D} \in \mathbb{R}^{n \times d} \text{ s.t. } \forall j \in [d],\ \|\mathbf{d}_j\|_2 \leq 1\}. \tag{2}$$

Algorithm 1 Online Dictionary Learning

Input:

- $\mathbf{x} \in \mathbb{R}^n \sim \Pi$ (a way to draw i.i.d samples from Π)
- $\lambda \in \mathbb{R}$ (regularization parameters)
- $\mathbf{D}_0 \in \mathcal{D}$ (initial dictionary)
- T (number of iterations)

Initialization: $\mathbf{A}_0 \in \mathbb{R}^{d \times d} \leftarrow 0, \mathbf{B}_0 \in \mathbb{R}^{n \times d} \leftarrow 0$ (reset the "past" information)
for $t = 1$ to T **do**

Step 0: Draw $\mathbf{x}_t$ from Π

Step 1: Sparse Coding: compute $\alpha_{\mathbf{D}}$ using ℓ_1-norm minimization (e.g. ISTA, FISTA).

$$\boldsymbol{\alpha}_t \leftarrow \underset{\boldsymbol{\alpha} \in \mathbb{R}^d}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{x}_t - \mathbf{D}_{t-1}\alpha\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1.$$

Step 2: Update $\mathbf{A}_t$, $\mathbf{B}_t$

$$\begin{aligned} \mathbf{A}_t &\leftarrow \mathbf{A}_{t-1} + \boldsymbol{\alpha}_t \boldsymbol{\alpha}_t^T, \\ \mathbf{B}_t &\leftarrow \mathbf{B}_{t-1} + \mathbf{x}_t \boldsymbol{\alpha}_t^T. \end{aligned}$$

Step 3: Update $\mathbf{D}_t$ using Algorithm 2, with $\mathbf{D}_{t-1}$ as warm restart, so that

$$\begin{aligned} \mathbf{D}_t &\leftarrow \underset{\mathbf{D} \in \mathcal{D}}{\operatorname{argmin}} \frac{1}{t} \sum_{i=1}^{t} \Big(\frac{1}{2}\|\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i\|_2^2 + \lambda\|\boldsymbol{\alpha}_i\|_1 \Big), \\ &\leftarrow \underset{\mathbf{D} \in \mathcal{D}}{\operatorname{argmin}} \frac{1}{t} \Big(\frac{1}{2}\operatorname{Tr}(\mathbf{D}^T\mathbf{D}\mathbf{A}_t) - \operatorname{Tr}(\mathbf{D}^T\mathbf{B}_t) \Big), \end{aligned}$$

where the expected loss is substituted by the surrogate loss.
end for
Return $\mathbf{D}_T$ (learned dictionary)

Algorithm 2 Dictionary Update

Input: $\mathbf{D}_{t-1} = [\mathbf{d}_1 \ \cdots \ \mathbf{d}_d] \in \mathbb{R}^{n \times d}, \mathbf{A}_{t-1} = [\mathbf{a}_1 \ \cdots \ \mathbf{a}_d] \in \mathbb{R}^{d \times d}, \mathbf{B}_{t-1} = [\mathbf{b}_1 \ \cdots \ \mathbf{b}_d] \in \mathbb{R}^{n \times d}$
for $j = 1$ to d **do**
Update the j-th column to optimize for the surrogate loss:

$$\mathbf{d}_j \leftarrow \Pi_{\mathcal{D}} \left[\frac{1}{\mathbf{A}[j,j]} (\mathbf{b}_j - \mathbf{D}\mathbf{a}_j) + \mathbf{d}_j \right]$$

end for
repeat until convergence
Return D (updated dictionary)

2.3.4 Comparison

The comparison between different dictionary learning algorithms is shown in Table 2. To reduce computation overhead, K-SVD utilize atom-by-atom dictionary update, and ODL further achieve expected minimization with stochastic gradient descent.

Table 2: The comparison between different dictionary learning algorithms

Algorithm	**MOD** [4]	**K − SVD** [5]	**ODL** [6]
Sparse Coding	l_0-based method (Matching Pursuit, Basis Pursuit)		l_1-based method (ISTA, FISTA)
Update Dictionary	whole dictionary	atom-by-atom	
Data Amount	whole dataset	active data only	single data only

3 Predictive Dictionary Learning

3.1 Problem Formulation and Applications

Unsupervised (Reconstructive) dictionary learning has also been used for other purposes than pure signal reconstruction, such as classification, but recent works have shown that better results can be obtained when the dictionary is tuned to the specific task (and not just data) it is intended for. A general efficient framework has been proposed [7, 8], and it is based on two-layer model (hypothesis). Compared with reconstructive dictionary learning, the generalization guarantee and the optimization have been proven much difficult.

We present in this section a formulation for learning a dictionary in a *supervised* way for classification or regression tasks, which is also finding a good data representation. Given a dictionary $\mathbf{D}$ obtained using reconstructive dictionary learning presented in previous section, a vector $\mathbf{x} \in \mathbb{R}^n$ can be represented as a sparse vector $\boldsymbol{\alpha}_{\mathbf{D}}(\mathbf{x})$. We want to predict the variable $\mathbf{y}$ from $\mathbf{x}$, assuming they are associated. We can now use the sparse vector $\boldsymbol{\alpha}_{\mathbf{D}}(\mathbf{x})$ as a feature representation of a signal $\mathbf{x}$ in a classical empirical risk minimization formulation:

$$\min_{\mathbf{W} \in \mathcal{W}} \sum_{i=1}^{m} \Big(\ell_s(\mathbf{y}_i, \mathbf{W}, \boldsymbol{\alpha}_{\mathbf{D}}(\mathbf{x}_i)) \Big) + \frac{\gamma}{2} \|\mathbf{W}\|_F^2,$$

where $\mathbf{W}$ are model parameters which we want to learn, $\mathcal{W}$ is a convex set, γ is a regularization parameter, and $\ell_s : \mathcal{Y} \times \mathbb{R} \mapsto [0, b]$, $b > 0$ is a convex loss function that measures how well one can predict $\mathbf{y}$ by observing $\boldsymbol{\alpha}_{\mathbf{D}}(\mathbf{x})$ given the model parameters $\mathbf{W}$. For instance, it can be square, logistic, or hinge loss from support vector machines. The subscript s of ℓ_s indicates here that the loss is adapted to a *supervised* learning problem.

So far, the dictionary $\mathbf{D}$ is obtained in an unsupervised way. We now introduce the predictive (task-driven) dictionary learning formulation, that consists of *jointly* learning $\mathbf{W}$ and $\mathbf{D}$ by solving

$$\min_{\mathbf{D}\in\mathcal{D},\mathbf{W}\in\mathcal{W}} \sum_{i=1}^{m} \Big(\ell_s(\mathbf{y}_i, \mathbf{W}, \boldsymbol{\alpha}_{\mathbf{D}}(\mathbf{x}_i))\Big) + \frac{\gamma}{2}\|\mathbf{W}\|_F^2,$$

where $\mathcal{D}$ is a set of constraints defined in (2).

3.2 Generalization Guarantee

This subsection mainly utilizes the result in [7]. To provide generalization guarantee, we first define some useful properties. If the dictionary and the data set are of these properties, we can induce some useful lemmas and theorems to upper bound the estimation error.

3.2.1 Conditions

Definition 1 (Optimal Condition)**.** *Let $\Lambda \subseteq [d]$ denote the active index set which the corresponding atoms in dictionary are chosen, namely $\alpha_j \neq 0$ iff $j \in \Lambda$. Then, we have the optimal condition for sparse coding as*

$$\begin{cases} \langle \mathbf{d}_j, res(\mathbf{x}, \mathbf{D})\rangle = sgn(\alpha_j)\lambda, & j \in \Lambda \\ |\langle \mathbf{d}_j, res(\mathbf{x}, \mathbf{D})\rangle| < \lambda & j \notin \Lambda \end{cases}.$$

Definition 2 (k-incoherence)**.** *For $k \in [d]$ and $\mathbf{D} \in \mathcal{D}$, the k-incoherence $\mu_k(\mathbf{D})$ is defined as the minimum eigenvalue among k-atom sub-dictionaries of $\mathbf{D}$. Formally,*

$$\mu_k(\mathbf{D}) = \min\{\sigma_k(\mathbf{D}_\Lambda) : \Lambda \subseteq [d],\ |\Lambda| = k\},$$

where $\sigma_k(\mathbf{A})$ is the k^{th} eigenvalue of $\mathbf{A}$.

Definition 3 (k-sparsity)**.** *If every point $\mathbf{x}_i,\ \forall i \in [m]$ of $\mathbf{X}$ satisfies $\|\boldsymbol{\alpha}_{\mathbf{D}}(\mathbf{x}_i)\|_0 \leq k$, then $\boldsymbol{\alpha}_{\mathbf{D}}$ is $k-$sparse on $\mathbf{X}$.*

Definition 4 (k-margin)**.** *Given a dictionary $\mathbf{D}$ and a data set $\mathbf{X} \in \mathbb{R}^{n\times m}$ with points $\mathbf{x}_i \in \mathcal{B}_{\mathbb{R}^n},\ i \in [m]$, the k-margin of $\mathbf{D}$ on $\mathbf{x}_i$ and data set $\mathbf{X}$ is*

$$margin_k(\mathbf{D}, \mathbf{x}_i) := \max_{\substack{\mathcal{I}\subseteq[d] \\ |\mathcal{I}|=d-k}} \min_{j\in\mathcal{I}} \{\lambda - |\langle \mathbf{d}_j,\ \mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_{\mathbf{D}}(\mathbf{x}_i)\rangle|\},$$
$$margin_k(\mathbf{D}, \mathbf{X}) := \min_{i\in[m]} margin_k(\mathbf{D}, \mathbf{x}_i).$$

3.2.2 Main Results

Different from reconstructive dictionary learning, one only need to ensure the stability of $res(\mathbf{x}, \mathbf{D})$ to dictionary perturbations. However, in the sense of predictive dictionary learning, the complexity hinges upon the stability of sparse code, which needs extra properties to ensure generalization bound.

We first introduce Lemma 1 to shift the analysis of difference between empirical risk and statistical risk to difference between two independent empirical risks.

Lemma 1 (Symmetrization by Ghost Sample, Lemma 1 in [7]). *Let $\mathcal{F}(Z_m, \mathbf{X}''_m) \subset \mathcal{F}$ be a random subclass which can depend on both a labeled m-point data set Z_m and an unlabeled m-point data set X''_m. With another labeled m-point data set Z'_m, if $m \geq (\frac{b}{\xi})^2$, then*

$$\begin{aligned} &Pr_{Z_m, \mathbf{X}''_m}\left\{\exists f \in \mathcal{F}(Z_m, \mathbf{X}''_m),\ \hat{L}_{s,Z_m}(f) - L_s(z,f) \geq \xi\right\} \\ &\leq 2Pr_{Z_m, Z'_m, \mathbf{X}''_m}\left\{\exists f \in \mathcal{F}(Z_m, \mathbf{X}''_m),\ \hat{L}_{s,Z_m}(f) - \hat{L}_{s,Z'_m}(f) \geq \frac{\xi}{2}\right\}. \end{aligned} \tag{3}$$

With Lemma 1, we can provide Proposition 1 with $\mathbf{X}''_m$ chosen as an empty matrix and $\mathcal{F}(Z_m, \mathbf{X}''_m)$ as $\left\{f \in \mathcal{F}_\mu : margin_k(\mathbf{D}, \mathbf{X}) \geq \iota \triangleq \sqrt{\frac{387\epsilon}{\lambda}}\right\}$.

Proposition 1 (Generalization Bound, Proposition 1 in [7]). *If $m \geq (\frac{b}{\xi})^2$, then*

$$\begin{aligned} &Pr_{Z_m}\left\{\exists f \in \mathcal{F}_\mu : margin_k(\mathbf{D}, \mathbf{X}) \geq \iota \text{ and } \hat{L}_{s,Z_m}(f) - L_s(z,f) \geq \xi\right\} \\ &\leq 2Pr_{Z_m, Z'_m}\left\{\exists f \in \mathcal{F}_\mu : margin_k(\mathbf{D}, \mathbf{X}) \geq \iota \text{ and } \hat{L}_{s,Z_m}(f) - \hat{L}_{s,Z'_m}(f) \geq \frac{\xi}{2}\right\}. \end{aligned} \tag{4}$$

We express the RHS of (4) using event $\mathcal{A}$

$$\mathcal{A} = \left\{Z_m, Z'_m : \exists f \in \mathcal{F}_\mu : margin_k(\mathbf{D}, \mathbf{X}) \geq \iota \text{ and } \hat{L}_{s,Z_m}(f) - \hat{L}_{s,Z'_m}(f) \geq \frac{\xi}{2}\right\}. \tag{5}$$

We then divide $\mathcal{A}$ into there are at least or at most ψ points of ghost sample without guarantee of stable sparse code and define event $\mathcal{C}$ as

$$\mathcal{C} = \left\{Z_m, Z'_m : \exists f \in \mathcal{F}_\mu : margin_k(\mathbf{D}, \mathbf{X}) \geq \iota \text{ and } \nexists \tilde{\mathbf{X}} \subseteq_\psi \mathbf{X}' : margin_k(\mathbf{D}, \tilde{\mathbf{X}}) > \frac{1}{3} margin_k(\mathbf{D}, \mathbf{X})\right\}, \tag{6}$$

where $\tilde{\mathbf{X}} \subseteq_\psi \mathbf{X}'$ stands for $\tilde{\mathbf{X}}$ is a subset of $\mathbf{X}'$ with at most ψ points removed. We now only need to bound $Pr\{\mathcal{C}\} + Pr\{\mathcal{A} \cap \bar{\mathcal{C}}\}$ with a simple fact:

$$Pr\{\mathcal{A}\} = Pr\{\mathcal{A} \cap \mathcal{C}\} + Pr\{\mathcal{A} \cap \bar{\mathcal{C}}\} \leq Pr\{\mathcal{C}\} + Pr\{\mathcal{A} \cap \bar{\mathcal{C}}\}. \tag{7}$$

To upper bound the RHS of (7), we now present Theorem 4: the stability result of LASSO, which is the fundamental theorem of Lemma 2 and 3.

Theorem 4 (Sparse Coding Stability, Theorem 1 in [7]). *Let dictionaries $\mathbf{D}, \tilde{\mathbf{D}} \in \mathcal{D}$ satisfy $\mu_k(\mathbf{D}), \mu_k(\tilde{\mathbf{D}}) \geq \mu \geq 0$, $\|\mathbf{D} - \tilde{\mathbf{D}}\|_2 \leq \varepsilon$ and $\mathbf{x} \in \mathcal{B}_{\mathbb{R}^n}$. Suppose that there exists an index set $\mathcal{I} \subseteq [d]$, $|\mathcal{I}| = d - k$ such that*

$$|\langle \mathbf{d}_j,\ \mathbf{x} - \mathbf{D}\boldsymbol{\alpha}_\mathbf{D}(\mathbf{x})\rangle| < \lambda - \tau,\ \forall j \in \mathcal{I}, \tag{8}$$

for

$$\varepsilon \leq \frac{\tau^2 \lambda}{43}. \tag{9}$$

The following stability bound holds:

$$\|\boldsymbol{\alpha}_\mathbf{D}(\mathbf{x}) - \boldsymbol{\alpha}_{\tilde{\mathbf{D}}}(\mathbf{x})\|_2 \leq \frac{3\varepsilon\sqrt{s}}{\lambda\mu}. \tag{10}$$

In addition, if $\varepsilon = \frac{\tau'^2 \lambda}{43}$ for $\tau' < \tau$, then $\forall j \in \mathcal{I}$:

$$\left|\langle \tilde{\mathbf{d}}_j,\ \mathbf{x} - \tilde{\mathbf{D}}\boldsymbol{\alpha}_{\tilde{\mathbf{D}}}(\mathbf{x})\rangle\right| < \lambda - (\tau - \tau'). \tag{11}$$

Condition (8) suggests we ensure the optimal condition (Definition 1) with a margin $\tau > 0$, and condition (9) suggests that permissible radius of perturbation (PRP). Therefore, (10) indicates the perturbation of sparse coding is controlled by a constant factor times the dictionary perturbation, where the constant factor relies on k-sparsity, k-incoherence and ℓ_1-regularization coefficient. In addition, (11) maintains under small perturbation of dictionary will not change that a certain set of $d-k$ samples will remain inactive in the new sparse coding. In summary, some stability and margin are sustained after perturbation of dictionary, and thus same active set is guaranteed.

For imminent propositions, we first provide the covering number of $\mathcal{D}_\mu \triangleq \{\mathbf{D} \in \mathcal{D} : \mu_k(\mathbf{D}) \geq \mu\}$ and $\mathcal{F}_\mu = \{f = (\mathbf{D}, \mathbf{W}) \in \mathcal{F} : \mathbf{D} \in \mathcal{D}_\mu\}$.

Proposition 2 (Proposition 3 in [7]). *The proper ϵ-covering number of $\mathcal{D}_\mu$ is bounded by $(8/\epsilon)^{nd}$*

Proposition 3 (Proposition 4 in [7]). *The product of proper ϵ-covering number of $\mathcal{D}_\mu$ and $\mathcal{W}$ is bounded by*

$$\left(\frac{r}{2}\right)^k \left(\frac{8}{\epsilon}\right)^{(n+1)k} exp\left(\frac{-m\varpi^2}{2b^2}\right).$$

Lemma 2 (Good Ghost, Lemma 5 in [7]). *Fix $\mu, \lambda > 0$ and $k \in [d]$. With probability at least $1-\delta$ over two m-sample data set $X_m, X'_m \sim \Pi^m$, for any $\mathbf{D} \in \mathcal{D}_\mu$, for which k-sparse$(\boldsymbol{\alpha}_\mathbf{D}(\mathbf{X}))$ is satisfied, at least $m-\psi$ points $\tilde{\mathbf{X}} \subseteq \mathbf{X}'$ satisfy $margin_k(\mathbf{D}, \tilde{\mathbf{X}}) > \frac{1}{3} margin_k(\mathbf{D}, \mathbf{X})$ for*

$$\psi \triangleq nd \log \frac{3096}{margin_k^2(\mathbf{D}, \tilde{\mathbf{X}})\lambda} + \log(2m+1) + \log\frac{1}{\delta}. \tag{12}$$

Lemma 2 can be derived from Theorem 4 and Proposition 3. If we denote $Pr\{\mathcal{C}\} = \delta'$, Lemma 2 suggests the number of bad points in the ghost sample, which can then be used in following Lemma 3.

Lemma 3 (Large Deviation on Good Ghost, Lemma 6 in [7]). *Define $\varpi \triangleq \frac{\xi}{2} - (2\rho\theta + \frac{b\psi}{m})$ and $\theta \triangleq \frac{\epsilon}{\lambda}(1 + \frac{3r\sqrt{k}}{\mu})$. Then,*

$$Pr\left\{\mathcal{A} \cap \bar{\mathcal{C}}\right\} \leq \left(\frac{r}{2}\right)^k \left(\frac{8}{\epsilon}\right)^{(n+1)k} exp\left(\frac{-m\varpi^2}{2b^2}\right). \tag{13}$$

Proof: We first construct a event $\mathcal{G} \supseteq \mathcal{A} \cap \bar{\mathcal{C}}$ as

$$\mathcal{G} \triangleq \left\{ Z_m,\ Z'_m\colon \begin{array}{l} \exists f \in \mathcal{F}_\mu : margin_k(\mathbf{D}, \mathbf{X}) \geq \iota \text{ and} \\ \exists \tilde{\mathbf{X}} \subseteq_\psi \mathbf{X}' : margin_k(\mathbf{D}, \tilde{\mathbf{X}}) > \frac{1}{3} margin_k(\mathbf{D}, \mathbf{X}) \text{ and} \\ \hat{L}_{s,Z_m}(f) - \hat{L}_{s,Z'_m}(f) \geq \frac{\xi}{2} \end{array} \right\}.$$

It is equivalent bound the large deviation under the random sub-hypothesis class

$$\tilde{\mathcal{F}}(\mathbf{X}, \mathbf{X}') \triangleq \left\{ \begin{array}{l} \exists f \in \mathcal{F}_\mu : margin_k(\mathbf{D}, \mathbf{X}) \geq \iota \text{ and} \\ \exists \tilde{\mathbf{X}} \subseteq_\psi \mathbf{X}' : margin_k(\mathbf{D}, \tilde{\mathbf{X}}) > \frac{1}{3} margin_k(\mathbf{D}, \mathbf{X}) \end{array} \right\}.$$

Let $\mathcal{F}_\epsilon = \mathcal{D}_\epsilon \times \mathcal{W}_\epsilon$, where $\mathcal{D}_\epsilon$ is the ϵ-cover of $\mathcal{D}$ and $\mathcal{W}_\epsilon$ is the ϵ-cover of $\mathcal{W}$. Consider $f = (\mathbf{D}, \mathbf{W}) \in \tilde{\mathcal{F}}(\mathbf{X}, \mathbf{X}')$ and $f' = (\mathbf{D}', \mathbf{W}') \in \mathcal{F}_\epsilon$ be the closest cover function of f. If ϵ is small enough to satisfy Theorem 4, it is guaranteed that with at least $m-\psi$ points of $\mathbf{X}'$ and all points of $\mathbf{X}$

$$\begin{aligned} &|\langle \mathbf{W}, \boldsymbol{\alpha}_\mathbf{D}(\mathbf{x})\rangle - \langle \mathbf{W}', \boldsymbol{\alpha}_{\mathbf{D}'}(\mathbf{x})\rangle| \leq |\langle \mathbf{W} - \mathbf{W}', \boldsymbol{\alpha}_\mathbf{D}(\mathbf{x})\rangle| + |\langle \mathbf{W}', \boldsymbol{\alpha}_\mathbf{D}(\mathbf{x}_i) - \boldsymbol{\alpha}_{\mathbf{D}'}(\mathbf{x})\rangle| \\ &\leq \frac{\epsilon}{\lambda} + r\frac{3\epsilon\sqrt{k}}{\lambda\mu} = \frac{\epsilon}{\lambda}\left(1 + \frac{3r\sqrt{k}}{\mu}\right) \triangleq \theta. \end{aligned} \tag{14}$$

Therefore,

$$\frac{1}{m}\sum_{i=1}^{m}\left|\ell_s(y_i,\langle \mathbf{W},\boldsymbol{\alpha}_{\mathbf{D}}(\mathbf{x}_i)\rangle)\ell_s(y_i,\langle \mathbf{W}',\boldsymbol{\alpha}_{\mathbf{D}'}(\mathbf{x}_i)\rangle)\right| \leq \rho\theta$$
$$\frac{1}{m}\sum_{i=1}^{m}\left|\ell_s(y_i,\langle \mathbf{W},\boldsymbol{\alpha}_{\mathbf{D}}(\mathbf{x}_i)\rangle)\ell_s(y_i,\langle \mathbf{W}',\boldsymbol{\alpha}_{\mathbf{D}'}(\mathbf{x}_i)\rangle)\right| \leq \rho\theta + \frac{b\psi}{m}. \tag{15}$$

the difference between the losses of f and f' on the double sample will be at most $2\rho\theta + \frac{b\psi}{m}$. If we denote ν as the absolute deviation between the loss of f on original sample and ghost sample, then the loss of f' on original sample and ghost sample will be at least $\nu - (2\rho\theta + \frac{b\psi}{m})$ Consider $\nu > \frac{\xi}{2}$, then the target we want to bound turns to

$$Pr_{Z_m,Z'_m}\left\{\exists f' \in \mathcal{F}_\epsilon,\ \hat{L}_{s,Z_m}(f') - \hat{L}_{s,Z'_m}(f') \geq \frac{\xi}{2} - \left(2\rho\theta + \frac{b\psi}{m}\right) \triangleq \varpi\right\}. \tag{16}$$

We then apply Hoeffding's Inequality under the case of a single f' based on the fact that $\ell_s(y_i, f'(\mathbf{x}_i)) - \ell_s(y'_i, f'(\mathbf{x}'_i)) \in [-b, b]$ and thus we have

$$Pr_{Z_m,Z'_m}\left\{\hat{L}_{s,Z_m}(f') - \hat{L}_{s,Z'_m}(f') \geq \varpi\right\} \leq exp\left(\frac{-m\varpi^2}{2b^2}\right). \tag{17}$$

With Proposition 3 and union bound, we have

$$Pr_{Z_m,Z'_m}\left\{\exists f' \in \mathcal{F}_\epsilon,\ \hat{L}_{s,Z_m}(f') - \hat{L}_{s,Z'_m}(f') \geq \varpi\right\} \leq \left(\frac{r}{2}\right)^k\left(\frac{8}{\epsilon}\right)^{(n+1)k} exp\left(\frac{-m\varpi^2}{2b^2}\right). \tag{18}$$

Therefore, we complete the proof by

$$Pr\left\{\mathcal{A}\cap\bar{\mathcal{C}}\right\} \leq Pr_{Z_m,Z'_m}\left\{\exists f' \in \mathcal{F}_\epsilon,\ \hat{L}_{s,Z_m}(f') - \hat{L}_{s,Z'_m}(f') \geq \varpi\right\} \leq \left(\frac{r}{2}\right)^k\left(\frac{8}{\epsilon}\right)^{(n+1)k} exp\left(\frac{-m\varpi^2}{2b^2}\right). \tag{19}$$

Theorem 5 (Overcomplete Learning Bound, Theorem 3 in [7])**.** *With probability at least $1-\delta$ over a $Z_m \sim P^m$, for any $k \in [d]$ and any $f = (\mathbf{D}, \mathbf{W}) \in \mathcal{F}$ satisfying k-sparse($\boldsymbol{\alpha}_{\mathbf{D}}(\mathbf{X})$) and*

$$m > \frac{387}{margin_k^2(\mathbf{D},\mathbf{X})\lambda}, \tag{20}$$

the generalization error $\hat{L}_{s,Z_m}(\mathbf{D},\mathbf{W}) - L_s(z,\mathbf{D},\mathbf{W})$ is

$$O\left(b\sqrt{\frac{nd\log m + \log\frac{1}{\delta}}{m}} + \frac{b}{m}nd\log\frac{1}{margin_k^2(\mathbf{D},\mathbf{X})\lambda} + \frac{\rho}{m}\frac{r\sqrt{k}}{\lambda\mu_k(\mathbf{D})}\right). \tag{21}$$

Proof: With Proposition 1, Lemma 2 and 3, if we choose $Pr\{\mathcal{C}\} = \delta' = \left(\frac{r}{2}\right)^k\left(\frac{8}{\epsilon}\right)^{(n+1)k} exp\left(\frac{-m\varpi^2}{2b^2}\right) \triangleq \frac{\delta}{4}$ and $\epsilon = \frac{1}{m}$, yielding

$$Pr_{Z_m}\left\{\exists f \in \mathcal{F}_\mu : margin_k(\mathbf{D},\mathbf{X}) \geq \iota \text{ and } \hat{L}_{s,Z_m}(f) - L_s(z,f) \geq 2\left(\varpi + 2\rho\theta + \frac{b\psi}{m}\right)\right\} \leq \delta, \tag{22}$$

with

$$\varpi = b\sqrt{\frac{2((n+1)d\log 8m + d\log\frac{r}{2} + \log\frac{\delta}{4})}{m}},$$
$$\rho\theta = \frac{2\rho}{m\lambda}\left(1 + \frac{3r\sqrt{k}}{\mu}\right),$$
$$\frac{b\psi}{m} = \frac{b}{m}\left(nd\log\frac{3096}{margin_k^2(\mathbf{D},\mathbf{X})\lambda} + \log(2m+1) + \log\frac{\delta}{4}\right). \tag{23}$$

We now discuss the upper bound of estimation error derived in Theorem 5. The blue term represents the difference between loss of $f' \in \mathcal{F}_\epsilon$ on original sample and ghost sample, which somehow reflects the generalization error based on the ϵ-cover function. In addition, this term matches the same order of reconstructive dictionary learning [2] and in Table 1. The purple term demonstrates the good ghost sample points, which we can use ρ-Lipschitz to bound the ℓ_s function. However, the green term is used to bound those bad ghost sample points, where the number of such points is directly from Lemma 2. Both purple and green terms represent the estimation error via ϵ-cover of the space of dictionary and hypothesis class, mainly relying on the stability of sparse coding under some dictionary and classifier perturbations as quantified by Theorem 4. We finally make a remark that the sample requirement in (20) is from we set the ϵ-cover to be $\epsilon = \frac{1}{m}$ and $\epsilon = \frac{(\frac{1}{3}margin_k(\mathbf{D},\mathbf{X})^2\lambda}{43}$ to satisfy PRP condition in (9). Therefore, for the sample lower bound and purple and green term are determined primarily by Theorem 4.

3.3 Optimization Algorithm

This subsection focuses on the optimization of task-driven dictionary learning (TDDL). The method is a projected first-order stochastic gradient descent algorithm (summarized in Algorithm 3). Instead of empirical risk minimization, TDDL adopts expected risk minimization ($\mathbb{E}_{\mathbf{y},\mathbf{x}}[\ell_s(\mathbf{y},\mathbf{W},\boldsymbol{\alpha}_\mathbf{D}(\mathbf{x}))]$), and the expectation is taken with respect to the unknown probability distribution $P(\mathbf{x},\mathbf{y})$ of the data. Hence, stochastic gradient descent algorithms are used by cycling over a random permuted training set in practice. Even though the sparse coefficient $\boldsymbol{\alpha}_\mathbf{D}(\mathbf{x})$ are obtained by solving a non-differentiable optimization problem, ℓ_s is differentiable on $\mathcal{W}\times\mathcal{D}$ (uniformly Lipschitz continuous, optimality conditions of the elastic-net), and one can compute its gradient because it has been shown that the loss function admits a first-order Taylor expansion. Furthermore, the algorithm has been shown to converge to stationary points with satisfied assumptions.

Algorithm 3 Stochastic Gradient Descent Algorithm for Task-Driven Dictionary Learning

Input:

- $(\mathbf{x}, \mathbf{y}) \sim P$ (a way to draw i.i.d samples of P)
- $\lambda, \gamma \in \mathbb{R}$ (regularization parameters)
- $\mathbf{D}_0 \in \mathcal{D}, \mathbf{W}_0 \in \mathcal{W}$ (initial dictionary and weights)
- T (number of iterations), η (learning rate parameter)

for $t = 1$ to T **do**

Step 0: Draw $(\mathbf{x}_t, \mathbf{y}_t)$ from P.

Step 1: Sparse Coding: compute $\boldsymbol{\alpha}_t$ using ℓ_1-norm minimization (e.g. ISTA, FISTA).

$$\boldsymbol{\alpha}_t \leftarrow \underset{\boldsymbol{\alpha} \in \mathbb{R}^d}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{x}_t - \mathbf{D}_{t-1}\boldsymbol{\alpha}\|_2^2 + \lambda\|\alpha\|_1.$$

Step 2: Compute the active set:

$$\Lambda \leftarrow \{j \in [d] : \boldsymbol{\alpha}_t[j] \neq 0\}.$$

Step 3: Compute $\boldsymbol{\beta}_t^*$: Set $\boldsymbol{\beta}_{\Lambda^c}^* = 0$ and

$$\boldsymbol{\beta}_\Lambda^* = (\mathbf{D}_\Lambda^T \mathbf{D}_\Lambda)^{-1} \nabla_{\boldsymbol{\alpha}_\Lambda} \ell_s(\mathbf{y}_t, \mathbf{W}_{t-1}, \boldsymbol{\alpha}_t).$$

Step 4: Update the parameters by a projected gradient step

$$\begin{aligned}
\mathbf{D}_t &\leftarrow \Pi_{\mathcal{D}}\left[\mathbf{D}_{t-1} - \eta(-\mathbf{D}_{t-1}\boldsymbol{\beta}_t^*\boldsymbol{\alpha}_t^T + (\mathbf{x}_t - \mathbf{D}_{t-1}\boldsymbol{\alpha}_t)\boldsymbol{\beta}_t^{*T})\right], \\
\mathbf{W}_t &\leftarrow \Pi_{\mathcal{W}}\left[\mathbf{W}_{t-1} - \eta(\nabla_{\mathbf{W}} \ell_s(\mathbf{y}_t, \mathbf{W}_{t-1}, \boldsymbol{\alpha}_t) + \gamma \mathbf{W}_{t-1})\right],
\end{aligned}$$

where $\Pi_{\mathcal{W}}$ and $\Pi_{\mathcal{D}}$ are respectively orthogonal projections on the sets $\mathcal{W}$ and $\mathcal{D}$.

end for

return $\mathbf{D}_T$, $\mathbf{W}_T$ (learned dictionary and weights)

4 Sparse Coding Optimization Algorithm

In section 2, we introduce how to co-optimize $\mathbf{D}$ and $\mathbf{A}$, while in section 3, we further co-optimize $\mathbf{W}$. However, there is a crucial step in reconstructive or predictive dictionary learning: how to attain sparse coding. In this section, we will introduce proximal gradient descent in ℓ_1-norm regularization and its extension with Nesterov-like acceleration. We also will compare the convergence rate of proximal gradient descent and sub-gradient descent.

4.1 Proximal Gradient Descent

We first formulate the ℓ_1-norm regularization (a.k.a. LASSO expression) as

$$h(\boldsymbol{\alpha}) \triangleq \frac{1}{2}\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1 \triangleq f(\boldsymbol{\alpha}) + g(\boldsymbol{\alpha}). \tag{24}$$

Since $g(\boldsymbol{\alpha})$ is not differential everywhere, it's very intuitive we can solve LASSO using sub-gradient descent (Sub-GD). However, Sub-GD suffers from slow convergence rate of order $\mathcal{O}(T^{-1/2})$, which is mentioned in Lecture 4 class note [9].

One way to avoid slow convergence rate is using proximal gradient descent, which utilizes the proximal operator to iteratively solve the sub-problem, so it is much more computationally efficient than the original problem. The proximal gradient descent of LASSO use second-order approximation upon $f(\boldsymbol{\alpha})$ and replace $\boldsymbol{\nabla}^2 f$ with $\frac{1}{\eta_t}\mathbf{I}$, which in iteration t can be formulated as below:

$$\boldsymbol{\alpha}_{t+1} = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \left\{ f(\boldsymbol{\alpha}_t) + \langle \boldsymbol{\alpha} - \boldsymbol{\alpha}_t, \boldsymbol{\nabla} f(\boldsymbol{\alpha}_t) \rangle + \frac{1}{2\eta_t} \|\boldsymbol{\alpha} - \boldsymbol{\alpha}_t\|_2^2 + g(\boldsymbol{\alpha}) \right\}, \tag{25}$$

where η_t can be viewed as step size. After neglecting constant terms, we rewrite (25) as

$$\boldsymbol{\alpha}_{t+1} = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \left\{ \frac{1}{2\eta_t} \|\boldsymbol{\alpha} - (\boldsymbol{\alpha}_t - \eta_t \boldsymbol{\nabla} f(\boldsymbol{\alpha}_t))\|_2^2 + g(\boldsymbol{\alpha}) \right\}. \tag{26}$$

4.2 Iterative Shrinkage-Thresholding Algorithm

We now begin to introduce Iterative Shrinkage-Thresholding Algorithm (ISTA), which provides a shrinkage solution to (26) as

$$\boldsymbol{\alpha}_{t+1} = \mathcal{S}_{\lambda\eta_t} \left(\boldsymbol{\alpha}_t - \eta_t \boldsymbol{\nabla} f \left(\boldsymbol{\alpha}_t \right) \right), \tag{27}$$

where shrinkage operator is

$$\mathcal{S}_s(\boldsymbol{\alpha})_j = (|\alpha_j| - s)_+ sgn(\alpha_j), \ \forall j \in [d], \tag{28}$$

and

$$(x)_+ = \begin{cases} x, & x > 0 \\ 0, & x \leq 0 \end{cases}. \tag{29}$$

Fig. 7 shows the such shrinkage operation of one entry in $\boldsymbol{\alpha}$, which somehow reflects the influence of ℓ_1-norm regularization, suppressing the small value of α_j to zero.

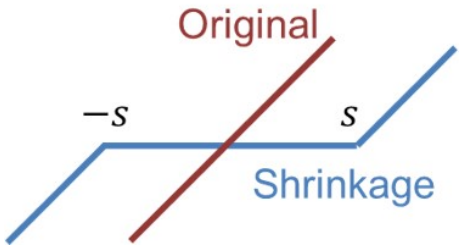


Figure 7: Shrinkage operation of one entry in $\boldsymbol{\alpha}$ (28)

The ISTA algorithm can now be summarized in Algorithm 4, mainly relying on (27) with a constant step size $\eta_t = \frac{1}{\beta}$, $\forall t$, where β is the smoothness of $f(\boldsymbol{\alpha})$. ISTA increase the convergence rate from $\mathcal{O}(T^{-1/2}) \rightarrow \mathcal{O}(T^{-1})$ [11].

Algorithm 4 Iterative Shrinkage-Thresholding Algorithm (ISTA) with constant step size

Input:

- Step size: $\eta = \frac{1}{\beta}$
- Threshold: ζ
- Dictionary: $\mathbf{D}$
- Encoded Signal: $\mathbf{x}$

Output:

- Sparse encoding signal: $\boldsymbol{\alpha}_{\mathbf{D}}(\mathbf{x})$

Algorithm:

Step 0: Pick $\boldsymbol{\alpha}_1 \in \mathbb{R}^d$ randomly.

Step 1: At iteration t ($t \geq 1$), compute

$$\boldsymbol{\alpha}_{t+1} = \mathcal{S}_{\lambda/\beta}\left(\boldsymbol{\alpha}_t - \frac{1}{\beta}\boldsymbol{\nabla} f\left(\boldsymbol{\alpha}_t\right)\right).$$

If $|h(\boldsymbol{\alpha}_{t+1}) - h(\boldsymbol{\alpha}_t)| \leq \zeta$, return $\boldsymbol{\alpha}_t$; else, repeat Step 1 and $t \leftarrow t+1$.

4.3 Fast Iterative Shrinkage-Thresholding Algorithm

Just like Nesterov accelerates GD with well-structure point, which extend the concept of *momentum*, similar idea can be adopted to ISTA, resulting in Fast Iterative Shrinkage-Thresholding Algorithm (FISTA). The FISTA algorithm is summarized in Algorithm 5, which consists of mainly three operations in each iteration: shrinkage operation, coefficient update and new well-structure point. FISTA increase the convergence rate from $\mathcal{O}(T^{-1}) \rightarrow \mathcal{O}(T^{-2})$ [11].

Algorithm 5 Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) with constant step size

Input:

- Step size: $\eta = \frac{1}{\beta}$
- Threshold: ζ
- Dictionary: $\mathbf{D}$
- Encoded Signal: $\mathbf{x}$

Output:

- Sparse encoding signal: $\boldsymbol{\alpha}_{\mathbf{D}}(\mathbf{x})$

Algorithm:

Step 0: Pick $\boldsymbol{\nu}_1 = \boldsymbol{\alpha}_1 \in \mathbb{R}^d$ randomly, $\kappa_1 = 1$.

Step 1: At iteration t ($t \geq 1$), compute

$$\begin{aligned}
\boldsymbol{\alpha}_{t+1} &= \mathcal{S}_{\lambda/\beta}\left(\boldsymbol{\nu}_t - \frac{1}{\beta}\boldsymbol{\nabla} f\left(\boldsymbol{\nu}_t\right)\right), \\
\kappa_{t+1} &= \frac{1+\sqrt{1+4\kappa_t^2}}{2}, \\
\boldsymbol{\nu}_{t+1} &= \boldsymbol{\alpha}_{t+1} + \frac{\kappa_t - 1}{\kappa_{t+1}}(\boldsymbol{\alpha}_{t+1} - \boldsymbol{\alpha}_t).
\end{aligned}$$

If $|h(\boldsymbol{\alpha}_{t+1}) - h(\boldsymbol{\alpha}_t)| \leq \zeta$, return $\boldsymbol{\alpha}_t$; else, repeat Step 1 and $t \leftarrow t+1$.

4.4 Optimization Efficiency

Table 3 summarizes the difference between Sub-GD, ISTA and FISTA.

Table 3: Comparison between Sub-GD, ISTA and FISTA.

Algorithm	Sub-GD [9]	ISTA [10]	FISTA [11]
Approach	GD	Proximal GD	
Acceleration	X	X	Nesterov-like
Convergence Rate	$\mathcal{O}(T^{-1/2})$	$\mathcal{O}(T^{-1})$	$\mathcal{O}(T^{-2})$

Fig. 8 shows the convergence rate of Sub-GD, ISTA and FISTA, demonstrating the influence of proximal gradient descent over Sub-GD and the effect of Nesterov-like acceleration.

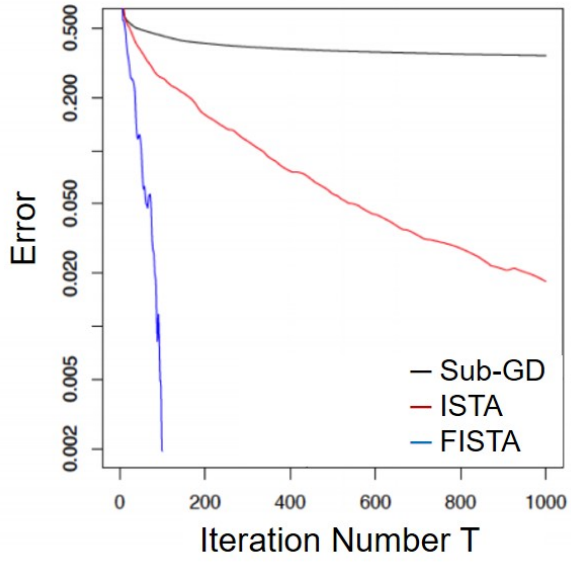


Figure 8: Convergence rate of Sub-GD, ISTA and FISTA (Figure 8.2 in [12])

5 Conclusions

In this work, we introduce the statistical guarantee and optimization of reconstructive and predictive dictionary learning respectively and analyze the converge rate of different algorithms for LASSO optimization of sparse coding. In Section 2, several generalization bounds of reconstructive dictionary learning are compared for 1) less parameter dependency thus sharper bound and 2) more general sparse constraint/data distribution cases. Furthermore, different optimization algorithms such as MOD, K-SVD, ODL are discussed in reducing computation overhead by 1) atom-by-atom dictionary update and 2) stochastic process. In Section 3, we further discuss predictive dictionary learning which utilizes back-propagation to consider the influence of labels. Stability of sparse coefficient under perturbed dictionary is discussed and the upper bound of estimation error is further derived. Moreover, TDDL algorithm is introduced to efficiently co-optimize the dictionary and classifier under supervised loss. In Section 4, proximal gradient descent on LASSO problem is introduced and the shrinkage operator and Nesterov-like acceleration are further discussed in ISTA and FISTA algorithm. The convergence rate of different algorithms are also analyzed.

Appendices

A Glossary

Notation	Description
$[m]$	$\{1, 2, \cdots, m\}$
Π	Marginal Probability measured over space $\mathcal{B}_{\mathbb{R}^n}$
P	Joint Probability measured over $\mathcal{B}_{\mathbb{R}^n} \times \mathcal{Y}$
$\mathbf{X}$	Unlabeled m-sample training data set
ℓ_u	Unsupervised loss function
ℓ_s	Supervised loss function
Z_m	Labeled m-sample training data set
$\mathbf{X}''$	Unlabeled m-sample ghost data set
Z'_m	Labeled m-sample ghost data set
$r\mathcal{B}_{\mathbb{R}^n}$	The ball in $\mathbb{R}^d$ with radius r
$\mathcal{D}$	The space of dictionaries $(\mathcal{B}_{\mathbb{R}^n})^d$
$\mathcal{W}$	The space of linear hypothesis class equal to $r\mathcal{B}_{\mathbb{R}^n}$
$\mathbf{d}_j$	The j^{th} atom of dictionary $\mathbf{D}$
$\boldsymbol{\alpha}_{\mathbf{D}}(\mathbf{x})$	Sparse coding vector of $\mathbf{x}$ depending on $\mathbf{D}$
$res(\mathbf{x}, \mathbf{D})$	Residual error of original and reconstructive signal: $\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}_{\mathbf{D}}(\mathbf{x})$
$\mu_k(\mathbf{D})$	k-incoherence: the minimum eigenvalue among k-atom sub-dictionaries of $\mathbf{D}$
k-sparse$(\boldsymbol{\alpha}_{\mathbf{D}}(\mathbf{X}))$	k-sparsity: it's true if $\|\boldsymbol{\alpha}_{\mathbf{D}}(\mathbf{x}_i)\|_0 \leq k,\ \forall i \in [m]$
$margin_k(\mathbf{D}, \mathbf{x}_i)$	$\max_{\substack{\mathcal{I}\subseteq[d] \\ \lvert\mathcal{I}\rvert=d-k}} \min_{j\in\mathcal{I}} \{\lambda - \lvert\langle \mathbf{D}_j,\ \mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_{\mathbf{D}}(\mathbf{x}_i)\rangle\rvert\}$
$margin_k(\mathbf{D}, \mathbf{X})$	$\min_{i\in[m]} margin_k(\mathbf{D}, \mathbf{x}_i)$
$\mathcal{D}_\mu$	$\{\mathbf{D} \in \mathcal{D} : \mu_k(\mathbf{D}) \geq \mu\}$
$\mathcal{F}$	$\{f \triangleq \mathbf{x} \mapsto \langle \mathbf{W}, \boldsymbol{\alpha}_{\mathbf{D}}(\mathbf{x})\rangle : \mathbf{D} \in \mathcal{D},\ \mathbf{W} \in \mathcal{W}\}$
$\mathcal{F}_\mu$	$\{f = (\mathbf{D}, \mathbf{W}) \in \mathcal{F} : \mathbf{D} \in \mathcal{D}_\mu\}$
ψ	The maximum number of bad samples in ghost data set
$\tilde{\mathbf{X}} \subseteq_\psi \mathbf{X}'$	$\tilde{\mathbf{X}}$ is a subset of $\mathbf{X}'$ with at most ψ points removed
$\mathcal{A}$	$\left\{Z_m, Z'_m : \exists f \in \mathcal{F}_\mu : margin_k(\mathbf{D}, \mathbf{X}) \geq \iota \text{ and } \hat{L}_{s,Z_m}(f) - \hat{L}_{s,Z'_m}(f) \geq \frac{\xi}{2}\right\}$
$\mathcal{C}$	$\left\{Z_m, Z'_m : \exists f \in \mathcal{F}_\mu : margin_k(\mathbf{D}, \mathbf{X}) \geq \iota \text{ and } \nexists \tilde{\mathbf{X}} \subseteq_\psi \mathbf{X}' : margin_k(\mathbf{D}, \tilde{\mathbf{X}}) > \frac{margin_k(\mathbf{D}, \mathbf{X})}{3}\right\}$
$\mathcal{D}_\epsilon$	The ϵ-cover of $\mathcal{D}$
$\mathcal{W}_\epsilon$	The ϵ-cover of $\mathcal{W}$
$\mathcal{F}_\epsilon$	$\mathcal{D}_\epsilon \times \mathcal{W}_\epsilon$
$\mathcal{S}_s(\boldsymbol{\alpha})_j$	Shrinkage operator of α_j with threshold s: $(\lvert\alpha_j\rvert - s)_+ sgn(\alpha_j)$

B Work Division

Name	Student ID	Work Assignment
Kai-Chieh Hsu	B03901026	Section 1.2, Section 3.2 and Section 4
Ching-Yao Chou	F03943134	Section 1.1, Section 2.3, Section 3.1 and Section 3.3
Chieh-Fang Teng	D06943020	Section 2.1, Section 2.2 and Section 2.3.2

References

[1] A. Maurer and M. Pontil, "K-dimensional coding schemes in hilbert spaces," in *IEEE Trans. Inf. Theory*, vol. 56, no. 11, pp. 5839-5846, Nov. 2010.

[2] D. Vainsencher, S. Mannor, and A. M. Bruckstein, "The sample complexity of dictionary learning," in *Journal Machine Learning Research*, vol. 12, pp. 3259-3281, 2011.

[3] R. Gribonval, R. Jenatton, F. Bach, M. Kleinsteuber and M. Seibert, "Sample complexity of dictionary learning and other matrix factorizations," in *IEEE Trans. Inf. Theory*, vol. 61, no. 6, pp. 3469-3486, June 2015.

[4] K. Engan, S. O. Aase and J. H. Husoy, "Method of optimal directions for frame design," in *1999 IEEE Int. Conf. Acoustics, Speech, and Signal Processing. Proceedings*, vol. 5, pp. 2443-2446, 1999.

[5] M. Aharon, M. Elad and A. Bruckstein, "K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation," in *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311-4322, Nov. 2006.

[6] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proc. IEEE Int. Conf. Machine Learning*, 2009.

[7] N. A. Mehta and A. G. Gray, "On the sample complexity of predictive sparse coding,"arXiv:1202.4050 [cs.LG], 2012.

[8] J. Mairal, F. Bach and J. Ponce, "Task-Driven Dictionary Learning," in *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 791-804, Apr. 2012.

[9] I-Hsiang Wang, "Mathematical Principles of Machine Learning-Lecture 4: Stability", 2018. [Online]. Available: http://homepage.ntu.edu.tw/~ihwang/Teaching/Sp18/MLNotes/Lecture04_v1.pdf

[10] I. Daubechies, M. Defrise, and C. D. Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," in *Commun. Pure Appl. Math.*, vol. 57, pp. 1413-1457, 2004.

[11] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," in *SIAM Journal Imaging Sciences*, vol. 2, pp. 183-202, 2009.

[12] Ryan Tibshirani, "Convex Optimization-Lecture 8", 2015. [Online]. Available: http://www.stat.cmu.edu/~ryantibs/convexopt-S15/scribes/08-prox-grad-scribed.pdf