Efficient Shift-Invariant Dictionary Learning

Guoqing Zheng
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213, USA
gzheng@cs.cmu.edu

Yiming Yang
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213, USA
yiming@cs.cmu.edu

Jaime Carbonell
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213, USA
jgc@cs.cmu.edu

ABSTRACT

Shift-invariant dictionary learning (SIDL) refers to the problem of discovering a set of latent basis vectors (the dictionary) that captures informative *local patterns* at different locations of the input sequences, and a sparse coding for each sequence as a linear combination of the latent basis elements. It differs from conventional dictionary learning and sparse coding where the latent basis has the same dimension as the input vectors, where the focus is on global patterns instead of shift-invariant local patterns. Unsupervised discovery of shift-invariant dictionary and the corresponding sparse coding has been an open challenge as the number of candidate local patterns is extremely large, and the number of possible linear combinations of such local patterns is even more so. In this paper we propose a new framework for unsupervised discovery of both the shift-invariant basis and the sparse coding of input data, with efficient algorithms for tractable optimization. Empirical evaluations on multiple time series data sets demonstrate the effectiveness and efficiency of the proposed method.

CCS Concepts

•**Computing methodologies** → *Factor analysis; Learning latent representations;*

Keywords

dictionary learning; sparse coding; time series

1. INTRODUCTION

Sparse representation models, such as those in dictionary learning, have been proposed for obtaining both a succinct set of vectors as the new basis (also called the dictionary) from unlabeled input sequences, and for representing each sequence as a sparse linear combination of the basis elements, i.e., the sparse coding of the data. Successful applications include those in dimensionality reduction [6], image restoration [13, 2, 18], signal compression [15, 17] and compressed sensing [4, 9].

Conventional dictionary learning models assume that the induced basis is of the same dimensionality of the input data, thus the basis elements (vectors) and the input sequences (also vectors) are strictly aligned from coordinate to coordinate. This simplistic assumption enables efficient algorithms for finding the optimal basis from input data; however, it also significantly limits the scope of potential applications of those models. Consider the three-time-series example from [24] reproduced in Figure 1 for instance. Time series No. 1 and series No. 3 share the bumps (as plotted in red) which are similar in shape but appear at different locations. We would consider this pair (1 and 3) more similar than the other pairs but such similarity cannot be captured, obviously, using a global non-shift-invariant pattern (corresponding to a basis element) with the full span over the time series. On the other hand, such similarity could be captured if we allow the basis elements to have a shorter span and to occur at different locations in the time series. Such a desirable kind of basis is called *shift-invariant* basis, and finding such a basis from data is referred as solving the *shift-invariant dictionary learning* (SIDL) problem.

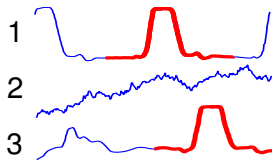


Figure 1: Importance of local patterns (plotted in red) versus that of the entire time series.

As a challenging problem with a potentially broad range of applications, SIDL has received increasing attention in recent machine learning research [8, 23, 14, 24]. Existing work and potential strengths/limitations can be grouped and outlined as follows: First, methods such as [24], which focus on exhaustive or heuristic search over all possible local pattern candidates, are either highly inefficient or sub-optimal, or both. Second, methods such as [7], which rely on the availability of sufficiently labeled data in order to learn discriminant local patterns, are not applicable when labeled data are lacking or are extremely sparse. Third, methods such as [8], which use convolution to model invariant shift of local patterns, has the drawback of not offering any method for sparse coding of input data. In other words, those methods are capable of learning shift-invariant local patterns as a new

KDD '16, August 13 - 17, 2016, San Francisco, CA, USA

ISBN 978-1-4503-4232-2/16/08...$15.00

DOI: http://dx.doi.org/10.1145/2939672.2939824

basis, but lack of the ability to combine the shift-invariant patterns in a sparse coding of the original data. Recall that sparse coding is a highly desirable for scalable data analysis tasks (including classification and clustering of time series), as well as for the interpretability of the analysis and the robustness of system predictions.

In this paper, we propose a new approach to SIDL which combines the strengths and addresses the aforementioned limitations of the previous methods. Specifically, our model allows the basis elements to be much shorter than the length of input series, to occur at different locations in the input series, and to be optimally combined into a sparse coding of observed data, thus representing each input series as as a sparse linear combination of the short basis elements. More importantly, both the shift-invariant basis and the sparse coding are *jointly optimized* in an *unsupervised learning* framework with our new algorithms for highly efficient computation. Empirical results on multiple times series demonstrate the effectiveness and efficiency of the proposed approach.

2. PRELIMINARIES

Suppose we have a set of n input data points with dimension p, $\{\mathbf{x}_i \in \mathbb{R}^p\}_{i=1}^n$, we want to learn a set of K basis elements, i.e., a dictionary, $\mathbf{D} = [\mathbf{d}_1, ..., \mathbf{d}_K]$ in $\mathbb{R}^{q \times K}$ and a sparse coding $\boldsymbol{\alpha}_i \in \mathbb{R}^K$ for each input data point $\mathbf{x}_i$. In classic sparse encoding and dictionary learning, the dimension of the basis is the same as that of the data point, i.e. $q = p$.

2.1 Sparse Coding

Given a *fixed* dictionary $\mathbf{D}$, for an input data point sparse encoding aims to find a sparse linear combination of the basis vectors as its representation. Formally, sparse coding with ℓ_1 regularization amounts to solving

$$\arg\min_{\boldsymbol{\alpha} \in \mathbb{R}^K} \frac{1}{2}\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1 \tag{1}$$

where λ is controls the balance between restoration error and coding sparsity. It is well known that ℓ_1 penalty yields a sparse solution, and the above LASSO problem can be efficiently solved with coordinate descent [19, 20]. Other sparsity penalties such as ℓ_0 regularization can be used well, however solving sparse coding with ℓ_0 penalty is often intractable. In this paper, we focus on the ℓ_1 regularized setting.

2.2 Dictionary Learning

When we lack a pre-existing dictionary a priori or we wish the dictionary to reflect directly the properties of the data, dictionary learning can induce both the sparse coding for the input data points and the dictionary by solving the following optimization problem

$$\begin{aligned} &\arg\min_{\substack{\mathbf{D} \in \mathbb{R}^{p \times K} \\ \boldsymbol{\alpha}_i \in \mathbb{R}^K}} \frac{1}{2}\sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i\|_2^2 + \lambda\|\boldsymbol{\alpha}_i\|_1 \\ &\text{s.t. } \|\mathbf{d}_k\|^2 \le c, \text{ for } k = 1, ..., K \end{aligned} \tag{2}$$

where c is a constant and the norm constraint on $\mathbf{d}_j$ is necessary to avoid degenerate solutions. The above problem is not convex in $(\mathbf{D}, \boldsymbol{\alpha})$ jointly, but is convex if we fix either variable. Hence alternate optimization between both sets of variables is a common approach to address dictionary learning.

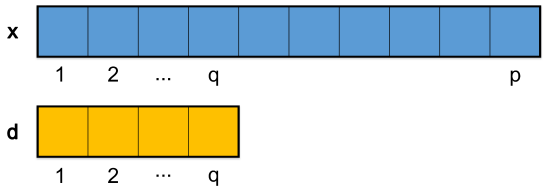
When $\mathbf{D}$ is fixed, the above problem is essentially a sparse coding as described in Section 2.1. When all $\boldsymbol{\alpha}$s are fixed, the above problem is quadratic programming with quadratic constraints (QCQP) and there exist many algorithms to solve it efficiently.

3. SHIFT-INVARIANT DICTIONARY LEARNING

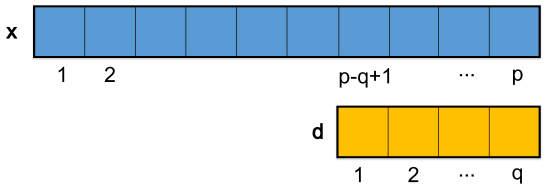
In this section, we present our shift-invariant dictionary learning (SIDL) to capture both the locality of representative patterns as well as preserve a sparse representations for the data points.

Unlike classic dictionary learning which enforces the same dimensionality for the basis and the data point ($q = p$), shift-invariant dictionary relaxes this constraint by $q \le p$, allowing the basis to slide along the support of the data point. For real problems, we may often set q to be much smaller than p; however we emphasize that classic DL is a special case of SIDL with the setting $q = p$.

For a data point $\mathbf{x}_i \in \mathbb{R}^p$ and shift-invariant basis $\mathbf{d}_k \in \mathbb{R}^q$, we introduce a variable t_{ik} to denote the location[1] where $\mathbf{d}_k$ is matched to $\mathbf{x}_i$, with $t_{ik} = 0$ indicating that $\mathbf{d}_k$ is aligned to the beginning of $\mathbf{x}_i$ and that $t_{ik} = p - q$ indicating the largest shift $\mathbf{d}_k$ can be aligned to $\mathbf{x}_i$ without running beyond the boundary of $\mathbf{x}_i$, as shown in Figure 3. Hence the possible values for t_{ik} are all integers[2] in $[0, p - q]$.



(a) Basis $\mathbf{d}$ is aligned to the beginning of $\mathbf{x}$, i.e. the shift variable $t = 0$



(b) Basis $\mathbf{d}$ is aligned to the end of $\mathbf{x}$, i.e. the shift variable $t = p - q$

Figure 2: The same basis with different shifts

Obviously, shifting a basis element $\mathbf{d}$ by an amount of t is

[1]The idea of shift invariant basis can also be applied to data with more than one "directions", such as image. For example, for images, the basis now tunrs to be a rectangular area and the shift is represented by two variables on each direction, the idea proposed in this paper can be easily extended to this case. For brevity, we focus on data with only one "direction" in this paper and leave the extensions as future work.

[2]As we only consider "discretized" data points, the shift can only take integer values.

equivalent to defining a new vector as

$$T_p(\mathbf{d}, t) \triangleq \mathbf{v} \in \mathbb{R}^p \tag{3}$$

where

$$\mathbf{v}_i = \begin{cases} \mathbf{d}_{i-t} & \text{if } 1 \le i - t \le q \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Given a set of input data points $\{\mathbf{x}_i\}_{i=1}^n$, shift-invariant dictionary learning aims to learn the dictionary $\mathbf{D} = [\mathbf{d}_1, ..., \mathbf{d}_K]$, the sparse coefficients $\{\boldsymbol{\alpha}_{ik}\}$ and the corresponding shifts $\{t_{ik}\}$ entirely from the data in an unsupervised way:

$$SIDL: \quad \underset{\substack{\mathbf{D}\in\mathbb{R}^{q\times K} \\ \boldsymbol{\alpha}_i\in\mathbb{R}^K \\ \mathbf{t}_{ik}\in[0,p-q]}}{\arg\min} \frac{1}{2}\sum_{i=1}^n \left\| \mathbf{x}_i - \sum_{k=1}^K \boldsymbol{\alpha}_{ik} T(\mathbf{d}_k, t_{ik}) \right\|_2^2 + \lambda \sum_{i=1}^n \|\boldsymbol{\alpha}_i\|_1$$
$$\text{s.t. } \|\mathbf{d}_k\|^2 \le c, \text{ for } k = 1, ..., K \tag{5}$$

4. MODEL LEARNING FOR SIDL

In this section, we present an efficient algorithm to solve SIDL. Similar to classic DL, the SIDL problem is non-convex; hence, we employ an anternative optimization scheme.

4.1 Shift-invariant sparse coding

Given the dictionary $\mathbf{D}$ fixed, Problem (5) turns to solving for the basis matching location t_{ik} and the the coefficients $\boldsymbol{\alpha}$. Also note that with $\mathbf{D}$ fixed, Problem (5) can be decomposed to learning the coefficients and the basis shift for every $\mathbf{x}_i$ independently. Hence in this subsection we drop the subscript and simply use $\mathbf{x}$ to represent any single input data point from the data set for clarity.

To learn $\boldsymbol{\alpha}$ and $\{t_k\}$ for input $\mathbf{x}$, we adopt coordinate descent to estimate the coefficients $\boldsymbol{\alpha}$ and a greedy approach to estimate $\{t_k\}$. Minimizing over α_k and t_k with $\{\alpha_j\}_{j\neq k}$ and $\{t_j\}_{j\neq k}$ fixed:

$$\underset{\alpha_k, t_k}{\arg\min} \frac{1}{2}\|\alpha_k T(\mathbf{d}_k, t_k) + \sum_{j\neq k} \alpha_j T(\mathbf{d}_j, t_j) - \mathbf{x}\|^2 + \lambda|\alpha_k| \tag{6}$$

For this one-dimensional optimization problem, its solution for α_k (as a function of the optimal basis shift t_k^*) is

$$\begin{aligned} \alpha_k &= S_{\frac{\lambda}{\|\mathbf{d}_k\|^2}}\left(\frac{T(\mathbf{d}_k, t_k^*)^\top \widehat{\mathbf{x}}}{T(\mathbf{d}_k, t_k^*)^\top T(\mathbf{d}_k, t_k^*)}\right) \\ &= S_{\frac{\lambda}{\|\mathbf{d}_k\|^2}}\left(\frac{T(\mathbf{d}_k, t_k^*)^\top \widehat{\mathbf{x}}}{\|\mathbf{d}_k\|^2}\right) \end{aligned} \tag{7}$$

where $\widehat{\mathbf{x}} \triangleq \left[\mathbf{x} - \sum_{j\neq k} \alpha_j T(\mathbf{d}_j, t_j)\right]$ is the residue of fitting $\mathbf{x}$ with all basis except for $\mathbf{d}_k$ and $S(\cdot)$ is the shrinkage operator defined as

$$S_a(x) = \begin{cases} x - a & \text{if } x \ge a \\ 0 & \text{if } x \in (-a, a) \\ x + a & \text{if } x \le -a \end{cases} \tag{8}$$

The above solution for α_k depends on the shift t_k of the kth basis element, and since it can only take integer values from $[0, p-q]$, a naive approach is to enumerate all possible t_k, compute the corresponding α_k and pick the pair of (t_k^*, α_k^*) which yields the minimum objective defined in (6). However, by plugging Eq. (7) back to Problem (6) and with simple math manipulation, we arrive at the following theorem which is much more efficient to compute.

Algorithm 1 Shift-invariant sparse coding

Input: data point $\mathbf{x} \in \mathbb{R}^p$, dictionary $\mathbf{D} = [\mathbf{d}_1, ..., \mathbf{d}_K]$
Output: sparse coding $\boldsymbol{\alpha}^*$, matching offsets $\{t_k^*\}_{k=1}^K$
Initialize $\boldsymbol{\alpha}$ randomly
repeat
  for $k = 1$ **to** K **do**
    $\widehat{\mathbf{x}} \leftarrow \left[\mathbf{x} - \sum_{j\neq k} \alpha_j T(\mathbf{d}_j, t_j)\right]$
    $t_k \leftarrow$ Eq. (9)
    $\alpha_k \leftarrow$ Eq. (10)
  end for
until convergence

PROPOSITION 4.1. *The optimal solution for Problem (6) is*

$$t_k^* = \underset{t_k}{\arg\max} |T(\mathbf{d}_k, t_k)^\top \widehat{\mathbf{x}}| \tag{9}$$

and

$$a_k^* = \begin{cases} T(\mathbf{d}_k, t_k^*)^\top \widehat{\mathbf{x}} & \text{if } |T(\mathbf{d}_k, t_k^*)^\top \widehat{\mathbf{x}}| > \lambda \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

Proof. See Appendix A. □

Proposition 4.1 suggests that we only need to compute the dot product between the basis element and the segment from the input data point; all updates are exact which do not involve parameter tuning. Algorithm 1 outlines the suggested procedure for solving shift-invariant sparse coding.

4.2 Shift-invariant dictionary update

When the coefficients $\boldsymbol{\alpha}$ and basis shifts $\{t_{ik}\}$ are fixed, updating the dictionary requires solving the following optimization problem:

$$\underset{\mathbf{d}_1,...,\mathbf{d}_k\in\mathbb{R}^q}{\arg\min} \frac{1}{2}\sum_{i=1}^n \left\| \mathbf{x}_i - \sum_k^K \alpha_{ik} T(\mathbf{d}_k, t_{ik}) \right\|^2$$
$$\text{s.t. } \|\mathbf{d}_k\|^2 \le c, \forall k \in [1, K] \tag{11}$$

Coordinate descent is used to solve for $\mathbf{d}$ as well, when minimizing over $\mathbf{d}_k$ with $\{\mathbf{d}_j\}_{j\neq k}$ fixed:

$$\underset{\mathbf{d}_k\in\mathbb{R}^q}{\arg\min} \frac{1}{2}\sum_{i=1}^n \left\| \mathbf{x}_i - \alpha_{ik} T(\mathbf{d}_k, t_{ik}) - \sum_{j\neq k}^K \alpha_{ij} T(\mathbf{d}_k, t_{ij}) \right\|^2$$
$$\text{s.t. } \|\mathbf{d}_k\|^2 \le c \tag{12}$$

This is a least square problem with quadratic constraints, which we can solve via its dual problem. The Lagrangian is:

$$L(\mathbf{d}_k, u) = \frac{1}{2}\sum_{i=1}^n \|\widehat{\mathbf{x}}_i - \alpha_{ik} T(\mathbf{d}_k, t_{ik})\|^2 + u(\|\mathbf{d}_k\|^2 - c) \tag{13}$$

where $\widehat{\mathbf{x}}_i \triangleq \mathbf{x}_i - \sum_{j\neq k}^K \alpha_{ij} T(\mathbf{d}_k, t_{ij})$ is the residue for $\mathbf{x}_i$ and $u \ge 0$ is the Lagrangian multiplier. Minimizing $L(\mathbf{d}_k, u)$ over $\mathbf{d}_k$, we get an analytic form for $\mathbf{d}_k^*$ as

$$\mathbf{d}_k^* = \frac{\sum_{i=1}^n \alpha_{ik} \widehat{\mathbf{x}}_i^{[1+t_{ik}, q+t_{ik}]}}{2u + \sum_{i=1}^n \alpha_{ik}^2} \tag{14}$$

Algorithm 2 Shift-invariant dictionary update

Input: data point $\{\mathbf{x}_i\}_{i=1}^n$, sparse coding $\boldsymbol{\alpha}$, matching offsets $\{t_k\}_{k=1}^K$
Output: dictionary $\mathbf{D} = [\mathbf{d}_1, ..., \mathbf{d}_K]$
Initialize $\mathbf{D}$ randomly
repeat
  for $k = 1$ **to** K **do**
    $\widetilde{\mathbf{x}} \leftarrow \sum_{i=1}^n \alpha_{ik} \widehat{\mathbf{x}}_i^{[1+t_{ik}, q+t_{ik}]}$
    $d_k \leftarrow$ Eq. (15)
  end for
until convergence

Algorithm 3 SIDL

Input: data points $\{\mathbf{x}_i\}_{i=1}^n$, desired basis dimension q, desired number of basis K
Output: dictionary $\mathbf{D} = [\mathbf{d}_1, ..., \mathbf{d}_K]$, sparse coding $\{\boldsymbol{\alpha}_i\}_{i=1}^n$, basis shifting location $\{t_{ik}\}$
Initialize $\mathbf{D}$, $\{\boldsymbol{\alpha}_i\}_{i=1}^n$ and $\{t_{ik}\}$ randomly
repeat
  Sparse coding: call Alg. 1
  Dictionary updating: call Alg. 2
until convergence

where $\widehat{\mathbf{x}}_i^{[1+t_{ik}, q+t_{ik}]}$ denotes the slice of $\widehat{\mathbf{x}}_i$ from index $1+t_{ik}$ through $q+t_{ik}$. Eq. (14) suggests that the optimum $\mathbf{d}_k$ is a weighted average of the corresponding matching segments of all residues $\{\widehat{\mathbf{x}}_i\}$. Plugging it back to the Lagrangian and maximizing the dual problem we finally get

$$\mathbf{d}_k^* = \begin{cases} \frac{\sqrt{c}}{\|\widetilde{\mathbf{x}}\|}\widetilde{\mathbf{x}} & \text{if } \frac{\|\widetilde{\mathbf{x}}\|}{\sum_{i=1}^n \alpha_{ik}^2} \geq \sqrt{c} \\ \frac{1}{\sum_{i=1}^n \alpha_{ik}^2}\widetilde{\mathbf{x}} & \text{otherwise} \end{cases} \tag{15}$$

where $\widetilde{\mathbf{x}} \triangleq \sum_{i=1}^n \alpha_{ik} \widehat{\mathbf{x}}_i^{[1+t_{ik}, q+t_{ik}]}$. Refer to Appendix B for a detailed proof of Eq. (15). Algorithm 2 outlines the learning procedure for shift-invariant dictionary updating.

4.3 Complete algorithm

Given unlabeled input data points, SIDL alternates between optimization of the sparse coding and updating the dictionary. Algorithm 3 outlines the learning procedure for shift-invariant ditionary updating.

4.3.1 *Algorithm analysis*

The time complexity of SIDL consists of two parts, one for shift-invariant sparse coding and the other for shift-invariant dictionary update. For shift-invariant sparse coding (Algorithm 1), the cost for one outer iteration, i.e., finding the optimum shifting locations and basis coefficients for K basis, is $O(Kq(p-q+1))$, where $O(q)$ comes from the cost of computing inner product of two vectors of length q and $p-q+1$ is the possible integer scope for solving t_k by Proposition 4.1, hence the total complexity for one call to Algorithm 1 is $O(M_1 Kq(p-q))$ where M_1 is the maximum number of iterations allowed in Algorithm 1.

The core part of shift-invariant dictionary update (Algorithm 2) is the computing for $\mathbf{d}_k$, which takes $O(nq)$ operations to compute the $\widetilde{\mathbf{x}}$ and scale it to get $\mathbf{d}_k$ according to Eq. (15). Thus the total complexity for one call to Algorithm 2 is $O(M_2 Knq)$ where M_2 is the maximum number of iterations allowed in Algorithm 2.

In fact, we do not necessarily require Algorithm 1 and 2 to converge in every call from 3, since as long as the objective function decreases in both shift-invariant sparse coding and shift-invariant dictionary update, the entire algorithm is still guaranteed to converge. Hence M_1 and M_2 can be set to be quite small, e.g., 10 to 20 iterations in both Algorithm 1 and 2 would suffice. Therefore, the total time complexity for SIDL is $O(MKq(n+p-q))$, where M is the total number of iterations allowed or needed for Algorithm 3 to converge.

4.3.2 *Comparison to classical dictionary learning*

As we mentioned, classical dictionary learning is a special case of the proposed framework of SIDL by setting $q = p$, hence the time complexity for classical dictionary learning is $O(MKpn)$. By comparing the complexity of the proposed SIDL framework to classical DL, because

$$MKq(n+p-q) < MKnp \tag{16}$$

holds for any $q < \min(n, p)$, hence in terms of asympotic complexity, the proposed SIDL is more efficient than standard classical DL in terms of time complexity. If q is close to $\min(n, p)$, then the proposed SIDL will be of about the same compelexity in computation. Empirical timing of SIDL can also be found in Section 5.3.

5. EXPERIMENTS

In this section, we evaluate the proposed Shift-Invariant Dictionary Learning (SIDL) from two aspects. One is to investigate its performance as a data reconstruction algorithm in reconstructing unseen data with the dictionary learned from training data, and the other is to evaluate the quality of learned sparse representations as features for downstream tasks, specifically for classification.

5.1 Data sets

Table 1: Dataset information

Dataset	#(Training)	#(Testing)	Length	#(classes)
Trace	100	100	275	4
synthetic_control	300	300	60	6
ECGFiveDays	23	861	136	2
Gun_Point	50	150	150	2
MedicalImages	381	760	99	10
Chlorine.	467	3840	166	3

Before presenting our experimental results, we first list the data sets on which SIDL is performed. We use several real-world time-series data sets in all evaluations from the UCR time series archives[3] [5], which are listed below with brief descriptions.

- **Trace:** Synthetic dataset designed to simulate failures in nuclear power plants;
- **Synthetic Control**: Synthetically generated charts by the process in [1];
- **ECGFiveDays**: Measurements of the abnormality of heartbeat recorded by an electrode;
- **GunPoint**: Video tracking data of a human test subject's right hand, recording whether or not he/she is pulling out a gun;

[3] http://www.cs.ucr.edu/~eamonn/time_series_data/

- **MedicalImages**: Histograms of pixel intensity of medical images of different human body regions;
- **ChlorineConcentration**: Chlorine concentration in drinking water versus time collected from eight households.

Table 1 lists related statistics about the data sets mentioned above.

5.2 SIDL on data reconstruction

We first investigate how SIDL can be used as a data reconstruction method by learning the shift-invariant basis on the training set and use the learned dictionary to encode the signals from the test set. We report the reconstruction error on all data sets, as well how its performance changes with different values of the sparsity regularization parameter λ, basis vector length q and dictionary size K.

We train SIDL with different parameters on the training portion of each data set and use the learned dictionary to reconstruct the testing data. Specifically, on the training data, we use SIDL the learn the shift-invariant dictionary $\mathbf{D}$, and then given the testing data points, we solve for their sparse codings with $\mathbf{D}$ fixed. The Mean Squared Error (MAE) for reconstruction is measured as

$$MAE = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} \left\| \mathbf{x}_i - \sum_{k=1}^{K} \boldsymbol{\alpha}_{ik}^* T(\mathbf{d}_k, t_{ik}^*) \right\|^2 \quad (17)$$

where $\boldsymbol{\alpha}_{ik}^*$ and t_{ik}^* are obtained by solving the shift invariant sparse coding problem as described in Section 4.1 with $\mathbf{d}_k$s trained from the training data points.

Figure 3 presents the complete reconstruction performance of SIDL on all time series data sets with different parameter configurations. It's worth noting that when $\frac{q}{p}$ equals 1, our proposed model reduces to classical dictionary learning. The dictionary sizes in Figure 3 are all set to be $K = 50$ as we have examined different values of K and observed similar patterns, so we omit the detailed graphs.

On five out of the six datasets, we can see that SIDL actually achieves (i.e. when $\frac{q}{p} < 1$) significantly lower reconstruction error than classical dictionary learning, *given the same number of basis elements and same sparsity regularization strength*. This suggests that not only SIDL produces a better dictionary to generalize to unseen data but also it results in smaller dictionary (since our basis elements are shorter than those of classical dictionary learning). Results from Figure 3 demonstrate that SIDL can be an effective alternative to yield unsupervised sparse representations compared to classical dictionary learning.

The above results show the reconstruction performance of SIDL doing a grid search of the parameters. When applied pratically, the optimal choice of $\frac{q}{p}$ and the degree of sparsity regularization λ can be obtained via cross-validation on splits from the training set, as we will show in the evaluation for using codings from SIDL as features for classification in Section 5.4.

5.3 Computational efficiency of SIDL

We plot the running time of SIDL with different parameter setting in Figure 4. As disscussed in Section 4.3.2, when the length of the basis is quite smaller than that of the input signal, SIDL will be much faster than classical DL, such as $q = 0.1p$ and $p = 0.25p$ in Figure 4. When q gets closer to p, as we can see from 4, the running time for SIDL is getting closer to that of DL, especially for larger K.

We also plot how the training objective of SIDL and DL decreases in terms of iterations. We set the convergence threshold, which is defined as the relative function objective change, to be 10^{-5}, i.e. convergence is achieved when

$$\frac{|F^{(i+1)} - F^{(i)}|}{F^{(i)}} \leq 10^{-5} \quad (18)$$

where $F^{(i)}$ is the objective function for training after the ith iteration. It's clear to see when $\lambda = 1$, all the SIDL runs converge in fewer iterations than DL, except for $q = 0.75p$ which takes about 200 more iterations to converge. When $\lambda = 10$, which encourages sparser solutions, all the SIDL runs converge in fewer iterations than DL. These imply that the proposed SIDL method can be at least as efficient as classical DL, with the flexibility to model shift-invariant basis present in the data.

It's also worth emphasizing that, from Figure 4, although the training loss minimum DL achieved is slightly better than those of SIDLs, this does not necessarily mean that the basis found by DL is any better than those found by SIDL. In fact, as we have shown in the data restruction and we will show in the classification tasks, the basis found by SIDL actually are better than those found by DLs.

5.4 Sparse coding from SIDL as features for classification

In this section, we evaluate the encodings output by SIDL as feature representations for time series classification tasks. For all data sets, we train the dictionary on the training set and apply it onto the test data points to get their encodings. The dictionary size K, the length of basis element q and the sparsity regularization parameter λ are chosen via 3-fold cross validation on the training set. We compare the classification results using sparse coding output by SIDL versus using raw input as features. Below is a list of classification algorithms we used for the evaluation.

- *1-Nearst Neighbor with Euclidean distance* (1NN-Euclidean). It is widely accepted in the time series mining community that 1NN with eucledian distance [22] is a strong though naive baseline. We run this algorithm on the raw representation of the time series.
- *1-Nearst Neighbor with Dynamic Time Warping* (1NN-DTW). It is by far a strong benchmark method for time series classfication. The distance of two time series is computed by performing DTW on the pair.
- *Support Vector Machine with raw representation* (SVM-Raw). This method trains an SVM classifier based on the raw representation of time series, without any transformation or processing of the input signals. We use the libsvm package to implement this method [3, 21].
- *Support Vector Machine with classical dictionary learning features* (SVM-DL). This method trains an SVM classifier based on the encodings from classical dictionary learning and then makes predictions on the encodings of testing data.
- *Support Vector Machine with SIDL* (SVM-SIDL). We train SVM on the shift invariant sparse codings given

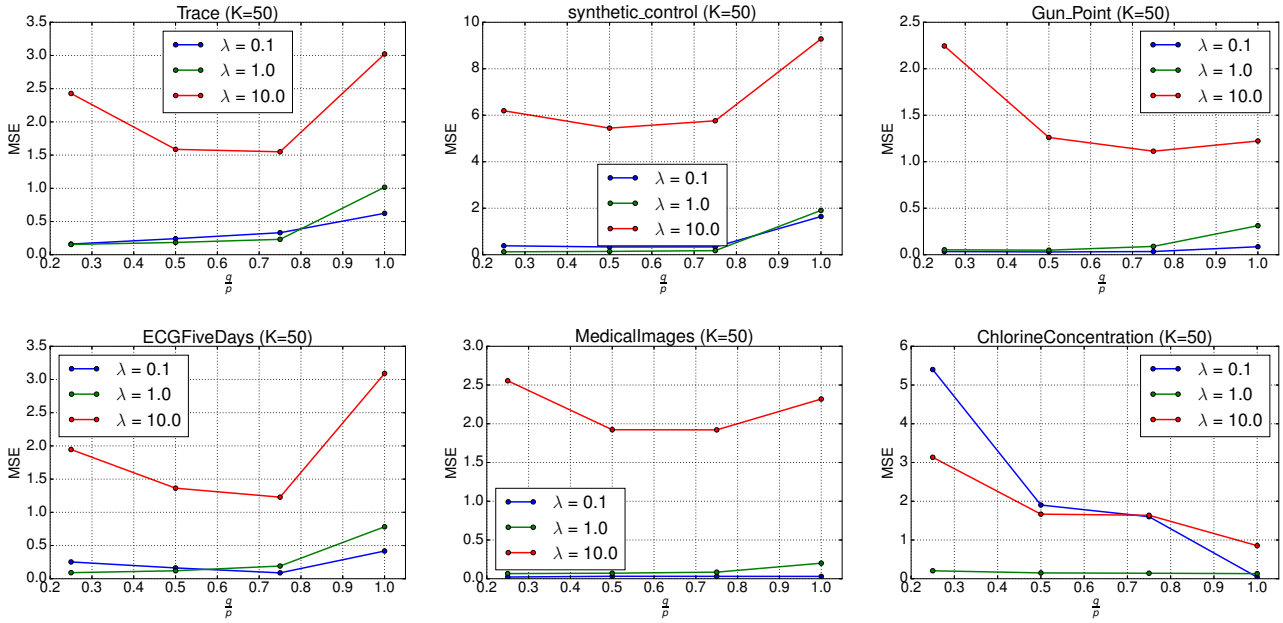
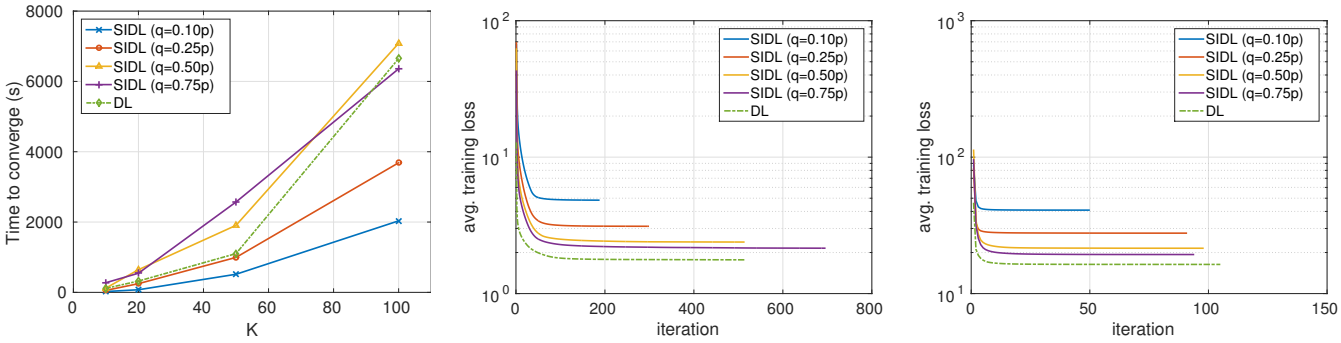


Figure 3: Reconstruction performance of SIDL w.r.t sparsity regularization parameter λ and the relative length of the basis element $\frac{q}{p}$ and a fixed dictionary size of $K = 50$. Note that $\frac{q}{p} = 1$ denotes the classical dictionary learning setting.



(a) Time taken for SIDL with different q and DL to converge (convergence threshold $= 10^{-5}$) when trained on the **Trace** data set with sparsity regularization parameter $\lambda = 1$.

(b) Training loss versus iteration for SIDL and DL when trained on the **Trace** data set with sparsity regularization parameter $\lambda = 1$. (convergence threshold $= 10^{-5}$)

(c) Training loss versus iteration for SIDL and DL when trained on the **Trace** data set with sparsity regularization parameter $\lambda = 10$. (convergence threshold $= 10^{-5}$

Figure 4: Running time comparisons

by SIDL, and report the performance on the encodings of the testing data points with the learned shift-invariant dictionary.

Table 2 presents the classification accuracies of various classification methods of SIDL on all data sets, with comparison to the baseline method. One interesting result to point out is that, without sparse encodings (such as DL and SIDL), classification with raw time series representations works always worse than naive 1NN method with Euclidean distance except on the **ECGFiveDays** data set. This suggests for data with temporal dependencies, such as time series, using raw representation as features for SVM is not a good idea. Nevertheless, sparse codings of the sequential data does help improve classification, compared to SVM-Raw.

SVM-SIDL achieves either the best or second best results on five data sets out of six, demonstrating the benefits of modeling shift-invariant local patterns in the data and though classical dictionary learning alleviate the problem suffered by SVM-Raw, it still lacks the flexibility to model local patterns. This further validates our motivation that a method that models local patterns in sequential data will better represent and explain the data.

As also validated by previous literature, 1NN-DTW is indeed a strong baseline method for time series classification, achieving three times of best and once of second best ac-

Table 2: Classification accuracies (Best results are printed in both bold and italic; second best results are printed in bold only. All parameters of SIDL are selected via 3-fold cross validation on splits of the training data with the following range $K \in \{10, 20, 50, 100\}, \lambda \in \{0.1, 1, 10, 100\}, q \in \{0.1p, 0.25p, 0.5p, 0.75p\}$. The same parameter ranges are also used for parameter tunning for DL. The parameter C for all linear SVMs are chosen through cross validation from {0.001, 0.01, 0.1,1,10,100,1000})

Dataset	1NN-Euclidean	1NN-DTW	SVM-Raw	SVM-DL	SVM-SIDL
Trace	0.760	***1.000***	0.430	0.680	**0.950**
synthetic_control	0.880	***0.993***	0.930	0.960	**0.967**
GunPoint	0.913	0.907	0.747	**0.933**	***0.953***
ECGFiveDays	0.797	0.768	**0.965**	0.497	***0.999***
MedicalImages	0.684	***0.737***	0.649	0.653	**0.688**
Chlorine.	***0.650***	**0.648**	0.533	0.533	0.533

curacies over all data sets, while the proposed SVM-SIDL twice of best and three times of second best. *It's worth mentioning that the proposed method is formulated to reduce reconstruction error, instead of classification error*. Besides, it's worth pointing out that nearest neighbor based methods suffer from expensive computational burden from pairwise DTW between the test instance and each of all training examples when doing prediction.

Also it is worth mentioning that SVM-SIDL performs best on the "toughest" dataset in the collection, i.e., **ECGFiveDays**, with a large margin over the baseline methods, including 1NN-DTW. This again reinforces the benefits of modeling shift-invariant local patterns for sequential data mining.

5.5 Learned Dictionary

In this section, we showcase the basis elements learned by SIDL without any human supervision. Figure 5 presents sample time series from each of the four classes and plots the learned basis elements together with the time series for the **Trace** data set. The two largest basis of each time series (in terms of the absolute value of the coefficient of that basis) are plotted with their shift locations in Figure 5. Also the degree of sparsity of the resulting codings (propotions of zero coefficients) for each time series are shown in the figure titles. It can be observed from the plots that the learned basis do capture local and discriminative patterns of the time series from different classes. This further validates the idea of representing time series with local basis elements and the proposed method can be used to efficiently learn them, even in the absence of true class labels. All the basis elements are learned with random initialization at the start of SIDL; it is possible that the algorithm might work even better given meaningful initializations.

Similarly, sample time series for **Gun Point** are plotted in Figure 6. Again, though the dictionary and the encodings are learned in an unsupervised fashion, it is clear that the learned representations for the time series do represent meaningful local patterns and more importantly, these patterns are relevant indicators for classification, as shown empirically by the example time series.

More sample time samples for **ECGFiveDays** are shown in Figure 7.

6. CONCLUSIONS AND FUTURE WORK

This paper presents a new framework for shift-invariant dictionary learning to capture local patterns from input signals. This framework relies on the assumption that the basis vectors may be be shorter than the full input signal. In other words useful temporal patterns may be embedded in different locations of a longer time series. We also presenet an efficient learning algorithm to estimate the shift-invariant sparse coding as well as a set of effective basis vectors. In our experiments on benchmark time series datasets for classification evaluations, the proposed method produces basis vectors that are more useful for signal reconstruction, exhibiting a lower reconstruction error. These same basis vectors also produce comparable or even more accurate classifications than several state-of-the-art baseline methods..

There are several promising directions in extending the work. First, the formulation of SIDL does not make any assumptions about the input signals, such as smoothness, sparsity or bounded norms [10]. If such assumptions are true for the input signals, we can exploit these characteristics to further improve the proposed method. Second, temporal patterns may compress or stretch in time, and hence incorporating dynamic time warping (DTW) [16, 11] into the framework may further improve the process by normalizing the basis vectors. Third, it is possible to extend the framework to take advantage of supervision such as partial labeling of the input signals [12], to help identify local patterns that are pertinent to the learning task. In addition, one may also want to investigate initialization strategies of the dictionaries and sparse codings based on priors (such domain knowledge) about the input signal to improve the resulting coding quality.

Acknowledgements

We thank the anonymous reviewers for their helpful comments.

7. REFERENCES

[1] R. J. Alcock and Y. Manolopoulos. Time-series similarity queries employing a feature-based approach. In *7-th Hellenic Conference on Informatics, Ioannina*, pages 27–29, 1999.

[2] C. Bao, J. Cai, and H. Ji. Fast sparsity-based orthogonal dictionary learning for image restoration. In *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*, pages 3384–3391, 2013.

[3] C. Chang and C. Lin. LIBSVM: A library for support vector machines. *ACM TIST*, 2(3):27, 2011.

[4] X. Chen, Z. Du, J. Li, X. Li, and H. Zhang. Compressed sensing based on dictionary learning for extracting impulse components. *Signal Processing*, 96:94–109, 2014.

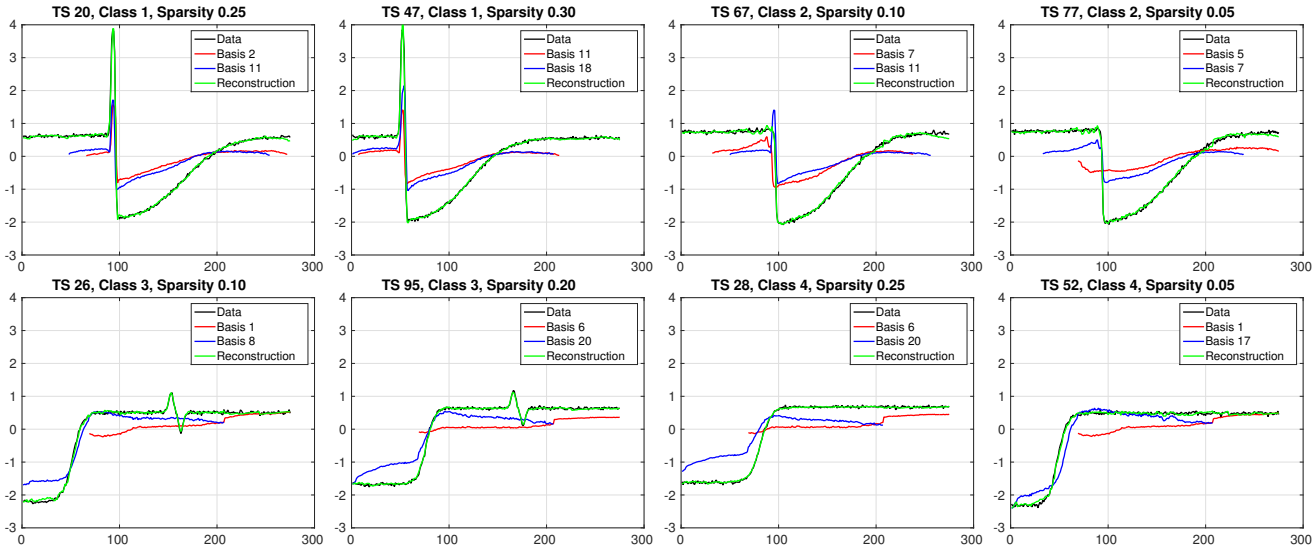


Figure 5: Sample time series in different classes from the *Trace* dataset (plotted in black). The reconstructed signal with the learned basis is plotted in green. The most 2 active basis are also shown in their matching location with the time series in red and blue.

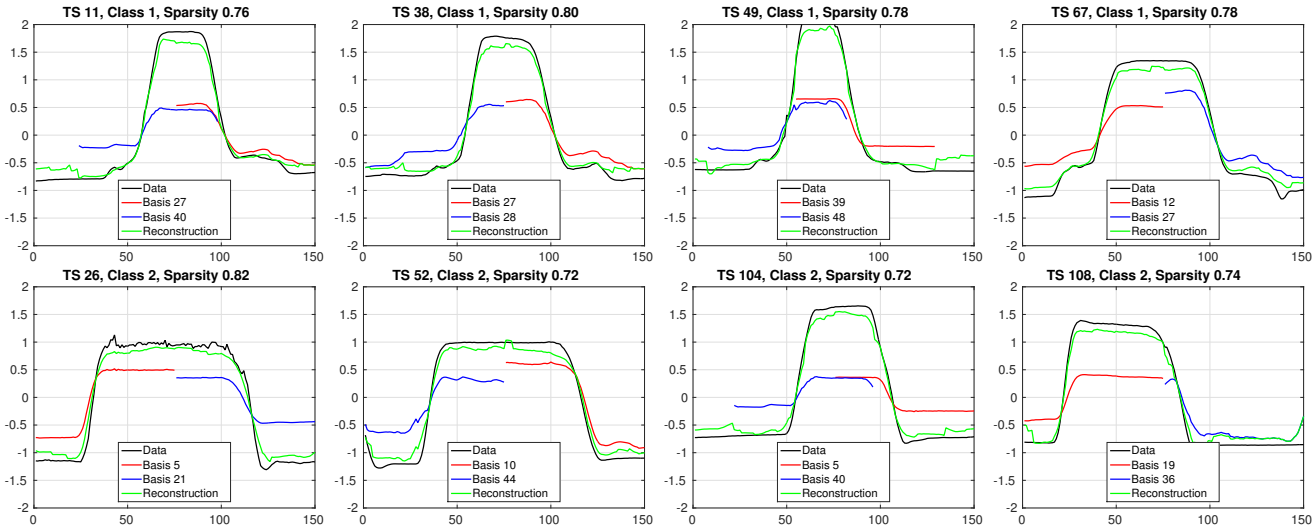


Figure 6: Sample time series in different classes from the *Gun Point* dataset (plotted in black). The reconstructed signal with the learned basis is plotted in green. The most 2 active basis are also shown in their matching location with the time series in red and blue.

[5] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista. The ucr time series classification archive, July 2015. www.cs.ucr.edu/~eamonn/time_series_data/.

[6] I. Gkioulekas and T. E. Zickler. Dimensionality reduction using the sparse linear model. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain.*, pages 271–279, 2011.

[7] J. Grabocka, N. Schilling, M. Wistuba, and L. Schmidt-Thieme. Learning time-series shapelets. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 392–401, 2014.

[8] R. B. Grosse, R. Raina, H. Kwong, and A. Y. Ng. Shift-invariance sparse coding for audio classification. In *UAI 2007, Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence, Vancouver, BC, Canada, July 19-22, 2007*, pages 149–158, 2007.

[9] Y. Huang, J. Paisley, Q. Lin, X. Ding, X. Fu, and X. Zhang. Bayesian nonparametric dictionary learning for compressed sensing MRI. *IEEE Transactions on Image Processing*, 23(12):5007–5019, 2014.

[10] W. Jiang, F. Nie, and H. Huang. Robust dictionary learning with capped l1-norm. In *Proceedings of the*

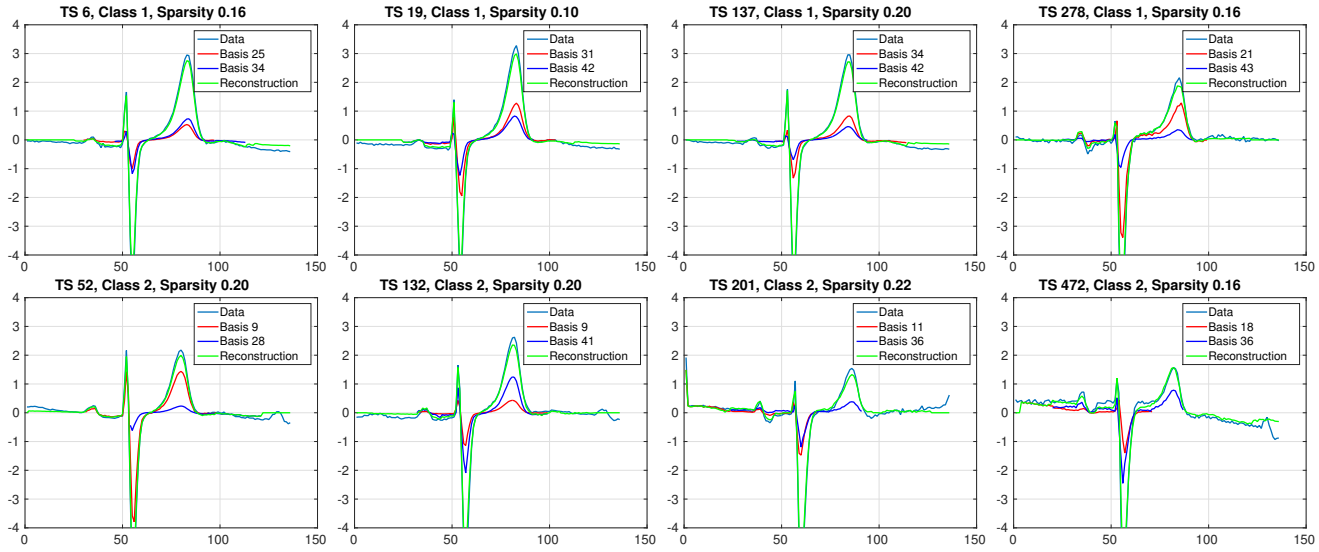


Figure 7: Sample time series in different classes from the *ECGFiveDays* dataset (plotted in black). The reconstructed signal with the learned basis is plotted in green. The most 2 active basis are also shown in their matching location with the time series in red and blue.

Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015, pages 3590–3596, 2015.

[11] E. J. Keogh and M. J. Pazzani. Scaling up dynamic time warping for datamining applications. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, Boston, MA, USA, August 20-23, 2000*, pages 285–289, 2000.

[12] J. Mairal, F. R. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. In *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008*, pages 1033–1040, 2008.

[13] J. Mairal, F. R. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-local sparse models for image restoration. In *IEEE 12th International Conference on Computer Vision, ICCV 2009, Kyoto, Japan, September 27 - October 4, 2009*, pages 2272–2279, 2009.

[14] A. Mueen, E. J. Keogh, and N. E. Young. Logical-shapelets: an expressive primitive for time series classification. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011*, pages 1154–1162, 2011.

[15] R. Rubinstein, M. Zibulevsky, and M. Elad. Double sparsity: learning sparse dictionaries for sparse signal approximation. *IEEE Transactions on Signal Processing*, 58(3):1553–1564, 2010.

[16] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 26(1):43–49, 1978.

[17] K. Skretting and K. Engan. Image compression using learned dictionaries by rls-dla and compared with k-svd. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 1517–1520. IEEE, 2011.

[18] C. Studer and R. G. Baraniuk. Dictionary learning from sparsely corrupted or compressed signals. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 3341–3344. IEEE, 2012.

[19] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. R. Statist. Soc. B*, 58:267–288, 1996.

[20] T. T. Wu and K. Lange. Coordinate descent algorithms for lasso penalized regression. *The Annals of Applied Statistics*, 2:1–21, 2008.

[21] Y. Wu and E. Y. Chang. Distance-function design and fusion for sequence data. In *Proceedings of the 2004 ACM CIKM International Conference on Information and Knowledge Management, Washington, DC, USA, November 8-13, 2004*, pages 324–333, 2004.

[22] X. Xi, E. J. Keogh, C. R. Shelton, L. Wei, and C. A. Ratanamahatana. Fast time series classification using numerosity reduction. In *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, pages 1033–1040, 2006.

[23] L. Ye and E. J. Keogh. Time series shapelets: a new primitive for data mining. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009*, pages 947–956, 2009.

[24] J. Zakaria, A. Mueen, and E. J. Keogh. Clustering time series using unsupervised-shapelets. In *12th IEEE International Conference on Data Mining, ICDM 2012, Brussels, Belgium, December 10-13, 2012*, pages 785–794, 2012.

APPENDIX

A. PROOF OF PROPOSITION 4.1

Proof. Plugging Eq. (7) back to Eq. (6), there are three cases:

- If $|T(\mathbf{d}_k,t_k)^\top\widehat{\mathbf{x}}|\le\lambda$, $\alpha_k^*=0$ and the corresponding value for Eq. (6) is $\frac{1}{2}\|\widehat{\mathbf{x}}\|^2$;

- If $T(\mathbf{d}_k,t_k)^\top\widehat{\mathbf{x}}>\lambda$, then $\alpha_k=\frac{T(\mathbf{d}_k,t_k)^\top\widehat{\mathbf{x}}-\lambda}{\|\mathbf{d}_k\|^2}$ and we now minimize the following objective over t_k:

$$\begin{aligned}
&\frac{1}{2}\left\|\frac{T(\mathbf{d}_k,t_k)^\top\widehat{\mathbf{x}}-\lambda}{\|\mathbf{d}_k\|^2}T(\mathbf{d}_k,t_k)-\widehat{\mathbf{x}}\right\|^2+\lambda\frac{T(\mathbf{d}_k,t_k)^\top\widehat{\mathbf{x}}-\lambda}{\|\mathbf{d}_k\|^2}\\
=&\frac{1}{2\|\mathbf{d}_k\|^4}\Big\{\left\|T(\mathbf{d}_k,t_k)T(\mathbf{d}_k,t_k)^\top\widehat{\mathbf{x}}\right\|^2+\|\mathbf{d}_k\|^4\|\widehat{\mathbf{x}}\|^2\\
&-2\lambda\|\mathbf{d}_k\|^2T(\mathbf{d}_k,t_k)^\top\widehat{\mathbf{x}}-2\|\mathbf{d}_k\|^2\widehat{\mathbf{x}}^\top T(\mathbf{d}_k,t_k)T(\mathbf{d}_k,t_k)^\top\widehat{\mathbf{x}}\\
&+\lambda^2\|\mathbf{d}_k\|^2+2\lambda\|\mathbf{d}_k\|^2\widehat{\mathbf{x}}^\top T(\mathbf{d}_k,t_k)+2\lambda\|\mathbf{d}_k\|^2\widehat{\mathbf{x}}^\top T(\mathbf{d}_k,t_k)\\
&-2\lambda^2\|\mathbf{d}_k\|^2\Big\}\\
=&\frac{1}{2\|\mathbf{d}_k\|^4}\Big\{-\|\mathbf{d}_k\|^2\widehat{\mathbf{x}}^\top T(\mathbf{d}_k,t_k)T(\mathbf{d}_k,t_k)^\top\widehat{\mathbf{x}}+\|\mathbf{d}_k\|^4\|\widehat{\mathbf{x}}\|^2\\
&+2\lambda\|\mathbf{d}_k\|^2T(\mathbf{d}_k,t_k)^\top\widehat{\mathbf{x}}-\lambda^2\|\mathbf{d}_k\|^2\Big\}\\
=&-\frac{\|T(\mathbf{d}_k,t_k)^\top\widehat{\mathbf{x}}-\lambda\|^2}{2\|\mathbf{d}_k\|^2}+\frac{1}{2}\|\widehat{\mathbf{x}}\|^2 \qquad (19)
\end{aligned}$$

where we extensively used the fact

$$T(\mathbf{d}_k,t_k)^\top T(\mathbf{d}_k,t_k)=\|\mathbf{d}_k\|^2 \qquad (20)$$

Since $T(\mathbf{d}_k,t_k)^\top\widehat{\mathbf{x}}>\lambda$, Eq. (19) is monotonically decreasing in $T(\mathbf{d}_k,t_k)$, therefore to minimize Eq. (19) we have

$$t_k^*=\arg\max_{t_k}T(\mathbf{d}_k,t_k)^\top\widehat{\mathbf{x}} \qquad (21)$$

- If $T(\mathbf{d}_k,t_k)^\top\widehat{\mathbf{x}}<-\lambda$, then $\alpha_k=\frac{T(\mathbf{d}_k,t_k)^\top\widehat{\mathbf{x}}+\lambda}{\|\mathbf{d}_k\|^2}$ and likewise we have:

$$\begin{aligned}
&\frac{1}{2}\left\|\frac{T(\mathbf{d}_k,t_k)^\top\widehat{\mathbf{x}}+\lambda}{\|\mathbf{d}_k\|^2}T(\mathbf{d}_k,t_k)-\widehat{\mathbf{x}}\right\|^2-\lambda\frac{T(\mathbf{d}_k,t_k)^\top\widehat{\mathbf{x}}+\lambda}{\|\mathbf{d}_k\|^2}\\
=&\frac{1}{2\|\mathbf{d}_k\|^4}\Big\{\left\|T(\mathbf{d}_k,t_k)T(\mathbf{d}_k,t_k)^\top\widehat{\mathbf{x}}\right\|^2+\|\mathbf{d}_k\|^4\|\widehat{\mathbf{x}}\|^2\\
&+2\lambda\|\mathbf{d}_k\|^2T(\mathbf{d}_k,t_k)^\top\widehat{\mathbf{x}}-2\|\mathbf{d}_k\|^2\widehat{\mathbf{x}}^\top T(\mathbf{d}_k,t_k)T(\mathbf{d}_k,t_k)^\top\widehat{\mathbf{x}}\\
&+\lambda^2\|\mathbf{d}_k\|^2-2\lambda\|\mathbf{d}_k\|^2\widehat{\mathbf{x}}^\top T(\mathbf{d}_k,t_k)-2\lambda\|\mathbf{d}_k\|^2\widehat{\mathbf{x}}^\top T(\mathbf{d}_k,t_k)\\
&-2\lambda^2\|\mathbf{d}_k\|^2\Big\}\\
=&\frac{1}{2\|\mathbf{d}_k\|^4}\Big\{-\|\mathbf{d}_k\|^2\widehat{\mathbf{x}}^\top T(\mathbf{d}_k,t_k)T(\mathbf{d}_k,t_k)^\top\widehat{\mathbf{x}}+\|\mathbf{d}_k\|^4\|\widehat{\mathbf{x}}\|^2\\
&-2\lambda\|\mathbf{d}_k\|^2T(\mathbf{d}_k,t_k)^\top\widehat{\mathbf{x}}-\lambda^2\|\mathbf{d}_k\|^2\Big\}\\
=&-\frac{\|T(\mathbf{d}_k,t_k)^\top\widehat{\mathbf{x}}+\lambda\|^2}{2\|\mathbf{d}_k\|^2}+\frac{1}{2}\|\widehat{\mathbf{x}}\|^2 \qquad (22)
\end{aligned}$$

Since $T(\mathbf{d}_k,t_k)^\top\widehat{\mathbf{x}}<-\lambda$, Eq. (22) is monotonically increasing in $T(\mathbf{d}_k,t_k)$, therefore to minimize Eq. (22) we have

$$t_k^*=\arg\min_{t_k}T(\mathbf{d}_k,t_k)^\top\widehat{\mathbf{x}} \qquad (23)$$

Aggregating the above three cases, since no matter which range $T(\mathbf{d}_k,t_k)^\top\widehat{\mathbf{x}}$ lies in, Eq. (19) and Eq. (22) are both upper bounded by $\frac{1}{2}\|\widehat{\mathbf{x}}\|^2$, we arrive at the statement in Proposition 4.1. □

B. PROOF OF EQUATION (15)

Proof. Plugging Eq. (14) of $\mathbf{d}_k$ back to the Lagrangian Eq. (13), we have

$$\begin{aligned}
L(u)=&\frac{1}{2}\sum_{i=1}^n\left\|\widehat{\mathbf{x}}_i-\alpha_{ik}T\left(\frac{\widetilde{\mathbf{x}}}{2u+\sum_{j=1}^n\boldsymbol{\alpha}_{jk}^2},t_{ik}\right)\right\|^2\\
&+u\left(\frac{\widetilde{\mathbf{x}}^\top\widetilde{\mathbf{x}}}{(2u+\sum_{j=1}^n\boldsymbol{\alpha}_{jk}^2)^2}-c\right) \qquad (24)
\end{aligned}$$

Taking the derivative of L w.r.t u yields

$$\begin{aligned}
L'(u)=&-\frac{\sum_{i=1}^n 2\boldsymbol{\alpha}_{ik}^2}{\left(2u+\sum_{i=1}^n\boldsymbol{\alpha}_{jk}^2\right)^3}\widetilde{\mathbf{x}}^\top\widetilde{\mathbf{x}}+\frac{2}{\left(2u+\sum_{i=1}^n\boldsymbol{\alpha}_{ik}^2\right)^2}\widetilde{\mathbf{x}}^\top\widetilde{\mathbf{x}}\\
&+\frac{-2u+\sum_{i=1}^n\boldsymbol{\alpha}_{ik}^2}{\left(2u+\sum_{i=1}^n\boldsymbol{\alpha}_{ik}^2\right)^3}\widetilde{\mathbf{x}}^\top\widetilde{\mathbf{x}}-c\\
=&\frac{1}{\left(2u+\sum_{i=1}^n\boldsymbol{\alpha}_{ik}^2\right)^2}\widetilde{\mathbf{x}}^\top\widetilde{\mathbf{x}}-c \qquad (25)
\end{aligned}$$

From the KKT conditions, we also have $u\ge0$ and

$$u\left(\frac{1}{\left(2u+\sum_{i=1}^n\boldsymbol{\alpha}_{ik}^2\right)^2}\widetilde{\mathbf{x}}^\top\widetilde{\mathbf{x}}-c\right)=0 \qquad (26)$$

Hence

- if $\frac{\|\widetilde{\mathbf{x}}\|}{\sum_{i=1}^n\boldsymbol{\alpha}_{ik}^2}\ge\sqrt{c}$, $L'(u)$ can reach 0 when

$$u=\frac{1}{2}\left(\frac{\|\widetilde{\mathbf{x}}\|^2}{\sqrt{c}}-\sum_{i=1}^n\boldsymbol{\alpha}_{ik}^2\right)\ge0 \qquad (27)$$

and the corresponding $\mathbf{d}_k$ is

$$\mathbf{d}_k=\frac{\sqrt{c}}{\|\widetilde{\mathbf{x}}\|}\widetilde{\mathbf{x}} \qquad (28)$$

- If $\frac{\|\widetilde{\mathbf{x}}\|}{\sum_{i=1}^n\boldsymbol{\alpha}_{ik}^2}<\sqrt{c}$, $L'(u)$ will always be negative, hence to maximize the Lagrangian, $u=0$ and the corresponding $\mathbf{d}_k$ is

$$\mathbf{d}_k=\frac{1}{\sum_{i=1}^n\boldsymbol{\alpha}_{ik}^2}\widetilde{\mathbf{x}} \qquad (29)$$

□