Robust Kernel Dictionary Learning Using a Whole Sequence Convergent Algorithm*

Huaping Liu[1,2,*], **Jie Qin**[1,2], **Hong Cheng**[3], **Fuchun Sun**[1,2]
[1]Department of Computer Science and Technology, Tsinghua University, Beijing, China
[2]State Key Lab. of Intelligent Technology and Systems, TNLIST, Beijing, China
[3]Center for Robotics, University of Electronic Science and Technology of China, Chengdu, China
*hpliu@tsinghua.edu.cn

Abstract

Kernel sparse coding is an effective strategy to capture the non-linear structure of data samples. However, how to learn a robust kernel dictionary remains an open problem. In this paper, we propose a new optimization model to learn the robust kernel dictionary while isolating outliers in the training samples. This model is essentially based on the decomposition of the reconstruction error into small dense noises and large sparse outliers. The outlier error term is formulated as the product of the sample matrix in the feature space and a diagonal coefficient matrix. This facilitates the kernelized dictionary learning. To solve the non-convex optimization problem, we develop a whole sequence convergent algorithm which guarantees the obtained solution sequence is a Cauchy sequence. The experimental results show that the proposed robust kernel dictionary learning method provides significant performance improvement.

1 Introduction

Sparse coding, which assumes that the signal can be sparsely represented by a dictionary, has become an active topic for scholars in the fields of machine learning and signal processing[Cheng *et al.*, 2009]. However, the linear reconstruction assumption which is explicitly imposed by many existing work is not valid for many practical signals[Cheng *et al.*, 2013][Yang *et al.*, 2015], particularly those well-modeled by manifolds[Hussein *et al.*, 2013][Ma *et al.*, 2014]. [Gao *et al.*, 2010] showed that one can use the kernel trick to perform sparse coding in the high-dimensional feature space instead of the input space in order to capture the non-linear structure of signals more efficiently. Kernel sparse coding has therefore extensive applications in many fields. On one hand, the high dimensionality of the feature space typically yields a more discriminative representation than input data space[Gao *et al.*, 2013][Nguyen *et al.*, 2013]. On the other hand, kernel trick provides effective strategy for sparse coding on Riemannian manifold[Harandi and Salzmann, 2015], of which structure is more complicated than the Euclidean space[Wang *et al.*, 2014].

Similar to the linear sparse coding, the dictionary learning problem of kernel sparse coding is also very important. If we map the training samples into a higher dimensional feature space by a proper mapping function, and learn the dictionary in this feature space, we expect a better linear representation. In [Gao *et al.*, 2010], the dictionary learning problem was tackled by gradient descent over the basis set in the input space directly. To compute the gradient, they had to resort to a specific kernel function such as the Gaussian kernel. Similarly, [Harandi *et al.*, 2012] and [Li *et al.*, 2013] addressed the dictionary learning problem on Symmetry-Positive-Definite (SPD) manifold, using Stein kernel and Log-Euclidean kernel, respectively. Very recently, [Zhang *et al.*, 2015] investigated the online dictionary learning on SPD manifold. All of those work deal with specified kernel and can limit their applicability considerably. In [Xie *et al.*, 2013], a nonlinear generalization of sparse coding and dictionray learning on manifold was proposed. This method is not based kernel and requires an extra affine constraint on the coding coefficients, which may be unexpected in practice. [Li and Ngom, 2012] revealed that the dictionary optimization only needs inner products of samples and this property can be used to kernelize linear sparse coding. Recently, some scholars proposed more principled kernel dictionary learning method[Anaraki and Hughes, 2013][Nguyen *et al.*, 2013][Liu *et al.*, 2014][Kim, 2014][Harandi and Salzmann, 2015]. Those methods are based on the basic conclusion that *the dictionary atoms in the feature space can be linearly reconstructed by the samples in the feature space*. As a result, the complicated dictionary learning problem in feature space is formulated as the rather simpler search of a coefficient matrix.

Nevertheless, existing kernel sparse coding and dictionary learning methods utilize square loss to measure the reconstruction capability. Such a loss function may be sensitive to the outliers[Liu *et al.*, 2015]. Although the above literatures proposed various methods for dictionary learning, none investigated the outliers which may lie in the training sample set. It should be noted that in [Nguyen *et al.*, 2013], the

*This work was supported in part by the National Key Project for Basic Research of China under Grant 2013CB329403; in part by the National Natural Science Foundation of China under Grants 61210013 and 61273256; and in part by the Tsinghua University Initiative Scientific Research Program under Grant 20131089295.

authors concerned the robustness but only added the corruptions on the testing samples and the training samples are still clean. *Therefore, the robust kernel dictionary learning problem, which aims to learn a dictionary in the feature space while isolating the outliers, has not been addressed*. As a comparison, the robust dictionary learning problem on Euclidean space has been extensively studied by many scholars. In [Chen and Wu, 2013], the error source decomposition technology was developed to solve the robust dictionary learning, and [Pan *et al.*, 2014] investigated the robust non-negative dictionaries learning. It is not clear how to extend their work to the kernel case. In [Nie *et al.*, 2013], a robust kernel dictionary selection method was developed for active learning. However, this work addressed the problem of dictionary selection, but not dictionary learning. In addition, [Kong *et al.*, 2011] addressed the robust nonnegative matrix factorization. [Xia *et al.*, 2012] investigated the robust kernel nonnegative matrix factorization. Both work did not consider the influence of sparsity.

In this work, we address the robust dictionary learning problem under the framework of kernel sparse coding. The main contributions are listed as follows:

1. A new optimization model is proposed to learn the robust kernel dictionary while isolating outliers in the training samples. This model is essentially based on the decomposition of the reconstruction error into small dense noises and large sparse outliers. The latter is formulated as the product of the sample matrix in the feature space and a diagonal coefficient matrix. This facilitates learning kernel dictionary.
2. A whole sequence convergent algorithm is developed to solve the non-convex robust kernel dictionary learning problem. The solution sequence is proved to be a Cauchy sequence and convergences to the critical point of the original optimization problem.
3. We perform empirical comparisons of the proposed method with the existing dictionary learning methods in the applications to image classification, which justifies that our method yields more robust results.

The rest of the organization of this paper is as follows: Section 2 gives the problem formulation, followed by the algorithm design and analysis in Section 3. Section 4 gives the experimental results.

Notations: For a matrix $\mathbf{M}$. The matrix norm $||\mathbf{M}||_2$ is defined as $||\mathbf{M}||_2 = \sqrt{Tr(\mathbf{M}^T\mathbf{M})}$. The pseudo-norm $||\mathbf{M}||_0$ is defined as the number of the non-zero element in $\mathbf{M}$. The pseudo-norm $||\mathbf{M}||_{0,2}$ is defined as the number of the non-zero columns in $\mathbf{M}$. For a vector $\mathbf{u}$, we use $||\mathbf{u}||_2$ to denote its 2-norm, and $||\mathbf{u}||_0$ to count the number of the non-zero elements. The symbol $\mathbf{I}$ represents the identity matrix with compatible dimension.

2 Problem formulation

Given the N training samples $\{\mathbf{y}_i\}_{i=1}^N \subset \mathcal{M}$, where $\mathcal{M}$ is a Riemannian manifold which also can be a subset of a Euclidean space. If we map the training samples into a higher dimensional space by a proper mapping function, we expect a better linear representation. The linearity in feature space corresponds to the nonlinearity in the original space. To this end, we denote $\Phi(\cdot) : \mathcal{M} \to \mathcal{H}$ to be the implicit nonlinear mapping from $\mathcal{M}$ into a high-dimensional (maybe infinite dimensional) dot product space $\mathcal{H}$. This mapping function is associated with some kernel $\kappa(\mathbf{y}_i, \mathbf{y}_j) = \Phi^T(\mathbf{y}_i)\Phi(\mathbf{y}_j)$, where $\mathbf{y}_i, \mathbf{y}_j \in \mathcal{M}$. For convenience, we denote the dimension of $\mathcal{H}$ as d, which may be infinite.

The aim of dictionary learning is to empirically learn a dictionary adapted to the training sample set $\{\mathbf{y}_i\}_{i=1}^N$ that we want to sparsely code. Therefore we need to determine some atoms $\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_K \in \mathcal{H}$, where $K < N$ is the size of the dictionary, to sparsely represent each training sample in the feature space. By denoting $\Phi(\mathbf{Y}) = [\Phi(\mathbf{y}_1), ..., \Phi(\mathbf{y}_N)] \in R^{d\times K}$ and $\mathbf{D} = [\mathbf{d}_1, ..., \mathbf{d}_K] \in R^{d\times K}$, we can formulate the kernel dictionary learning problem as

$$\min_{\mathbf{C},\mathbf{D}} ||\Phi(\mathbf{Y}) - \mathbf{DC}||_2^2 + \lambda_1||\mathbf{C}||_0, \tag{1}$$

where $\mathbf{C} \in R^{K\times N}$ is the sparse coding matrix and λ_1 is the sparsity penalty parameter.

By using the mapping function $\Phi(\cdot)$, we can transform the problem on Riemannian manifold to the linear coding problem in feature space. This is the great advantage of the kernel sparse coding[Gao *et al.*, 2013][Harandi and Salzmann, 2015]. While achieving great success in nonlinear sparse coding, such a formulation admits challenge to the dictionary learning since the dictionary atoms $\mathbf{d}_j$ may be in infinite dimensional space. Fortunately, some recent literature pointed that the dictionary can be represented by $\mathbf{D} = \Phi(\mathbf{Y})\mathbf{A}$, where $\mathbf{A} \in R^{N\times K}$ is a coefficient matrix. This means that the dictionary atoms can be linearly reconstructed by the training samples in the feature space. This conclusion was proved in [Nguyen *et al.*, 2013] and [Kim, 2014]. Based on this formulation, the dictionary learning problem becomes

$$\min_{\mathbf{A},\mathbf{C}} ||\Phi(\mathbf{Y}) - \Phi(\mathbf{Y})\mathbf{AC}||_2^2 + \lambda_1||\mathbf{C}||_0. \tag{2}$$

To avoid the non-unique solution, we impose the constraint that $||\mathbf{a}_j||_2 = 1$, where $\mathbf{a}_j$ is the j-th column of $\mathbf{A}$. Such a formation provides significant convenience since the learning of dictionary becomes the search of the matrix $\mathbf{A}$ and provides a principled derivation for nonlinear dictionary learning and sparse coding that essentially reduces to linear problems for any type of kernel function.

However, the reconstruction error term in Eq.(2) is the squared loss function in the feature space and therefore it is very sensitive to the data outliers, which may dominate the objective function. To attenuate the influence of outlier, we introduce an error matrix $\mathbf{E} = [\mathbf{e}_1, \cdots, \mathbf{e}_N] \in R^{d\times N}$, where $\mathbf{e}_i \in \mathcal{H}$ for $i = 1, 2, \cdots, N$, to give

$$\min_{\mathbf{A},\mathbf{C},\mathbf{E}} ||\Phi(\mathbf{Y}) - \Phi(\mathbf{Y})\mathbf{AC} - \mathbf{E}||_2^2 + \lambda_1||\mathbf{C}||_0 + \lambda_2||\mathbf{E}||_{0,2} \tag{3}$$

where λ_2 is the parameter to penalize the column sparsity of $\mathbf{E}$. The motivation of this model is to isolate some few columns in $\Phi(\mathbf{Y})$, which cannot be well reconstructed by the dictionary. However, the challenge in solving model (3) lies in the fact that $\mathbf{e}_i$ is in the feature space and may be infinite

dimension. To this end, we try to represent $\mathbf{E}$ using the samples $\Phi(\mathbf{Y})$. Since the non-zero columns of $\mathbf{E}$ correspond to the columns of $\Phi(\mathbf{Y})$, we can represent

$$\mathbf{E} = \Phi(\mathbf{Y})\mathbf{R},$$

where $\mathbf{R} \in R^{N\times N}$ is a diagonal matrix. As a result, the problem of outlier search is transformed into the problem of determination of $\mathbf{R}$. Furthermore, we assume that there does not exist all-zero column in $\Phi(\mathbf{Y})$, i.e., $||\Phi(\mathbf{Y})||_{0,2} = N$, then we have

$$||\mathbf{E}||_{0,2} = ||\mathbf{R}||_{0,2} = ||\mathbf{r}||_0,$$

where $\mathbf{r} = \mathrm{diag}(\mathbf{R})$ is defined as a column vector accommodating the diagonal elements of the diagonal matrix $\mathbf{R}$.

Therefore the robust kernel dictionary learning problem can be formulated as the following optimization problem:

$$\min ||\Phi(\mathbf{Y}) - \Phi(\mathbf{Y})\mathbf{AC} - \Phi(\mathbf{Y})\mathbf{R}||_2^2 + \lambda_1||\mathbf{C}||_0 + \lambda_2||\mathbf{r}||_0 \tag{4}$$

Please note that although the number of outlier is small, we cannot impose the sparsity on $\mathbf{R}$ since it is a diagonal matrix and therefore it is highly sparse. Instead, we encourage the sparsity of the vector which is formed by the diagonal elements. This shows that the outlier isolation term in (4), key to the robust dictionary learning, can be kernelized. Please note that the reconstruction error term now can be formulated as $||\Phi(\mathbf{Y})(\mathbf{I}-\mathbf{AC}-\mathbf{R})||_2^2 = Tr\{(\mathbf{I}-\mathbf{AC}-\mathbf{R})^T\mathbf{K}_{YY}(\mathbf{I}-\mathbf{AC}-\mathbf{R})\}$, where $\mathbf{K}_{YY} = \Phi^T(\mathbf{Y})\Phi(\mathbf{Y})$.

The model in (4) provides twofold roles: (1) Learn the dictionary coefficient $\mathbf{A}$ from the training sample set; and (2) Isolate the outliers. However, the model (4) is highly non-convex due to the coupling term $\mathbf{AC}$ and the zero norm terms imposed on $\mathbf{C}$ and $\mathbf{r}$. Neglecting the robustness problem, [Harandi and Salzmann, 2015] and [Nguyen *et al.*, 2013] developed method for dictionary learning. Both method are essentially based on the Method of Optimal Directions (MOD) method and Kernelized KSVD (K-KSVD) method. However, no whole sequence convergency can be guaranteed for such algorithms. That is to say, those algorithms at most can be proved that the functional value decreases at each iteration, but the solution sequence itself may not convergent. Motivated by the alternating proximal linearized method proposed by [Bolt *et al.*, 2013] and [Bao *et al.*, 2014], we develop a whole sequence convergent algorithm to solve the optimization problem in (4).

3 Algorithm design and analysis

To design the whole sequence convergent algorithm, we denote

$$\begin{cases} H(\mathbf{A},\mathbf{C},\mathbf{R}) = ||\Phi(\mathbf{Y}) - \Phi(\mathbf{Y})\mathbf{AC} - \Phi(\mathbf{Y})\mathbf{R}||_2^2 \\ F(\mathbf{A}) = \mathcal{I}_{\mathcal{A}}(\mathbf{A}) \\ G(\mathbf{C}) = \lambda_1||\mathbf{C}||_0 + \mathcal{I}_{\mathcal{C}}(\mathbf{C}) \\ Q(\mathbf{R}) = \lambda_2||\mathbf{Re}||_0 + \mathcal{I}_{\mathcal{R}}(\mathbf{R}), \end{cases}$$

where $\mathcal{I}_{\mathcal{A}}(\mathbf{A})$ denotes the indicator function of $\mathbf{A}$ which satisfies $I_{\mathcal{A}}(\mathbf{A}) = 0$ if $\mathbf{A} \in \mathcal{A}$ and $+\infty$ otherwise, where $\mathcal{A} = \{\mathbf{A} \in R^{N\times K} : ||\mathbf{a}_j||_2 = 1, j = 1, 2, \cdots, K\}$. Similarly, the symbol $I_{\mathcal{C}}(\mathbf{C})$ denotes the indicator function of $\mathbf{C}$, where $\mathcal{C} = \{\mathbf{C} \in R^{K\times N} : |\mathbf{C}_{ij}| \le c_m\}$. The symbol $I_{\mathcal{R}}(\mathbf{R})$ denotes the indicator function, where $\mathcal{R} = \{\mathbf{R} \in R^{N\times N} : |\mathbf{R}_{ii}| \le r_m \text{ for } i, j = 1, 2, \cdots, N, \text{ and } \mathbf{R}_{ij} = 0 \text{ for } i \ne j\}$. The values of c_m and r_m are simply set to serve as the upper bounds.

Motivated by the alternating proximal linearized method[Bao *et al.*, 2014], we need to alternatively update $\mathbf{C}$, $\mathbf{A}$ and $\mathbf{R}$. Before further processing, we formally define the proximal operator as

$$\mathrm{Prox}_{\mu}^{F}(\mathbf{U}) = \arg\min_{\mathbf{X}} F(\mathbf{X}) + \frac{\mu}{2}||\mathbf{U} - \mathbf{X}||_2^2. \tag{5}$$

In the following we introduce how to solve the three unknown matrices $\mathbf{C}$, $\mathbf{A}$ and $\mathbf{R}$. For conveniens, we use the superscript (k) to denote the values at the k-th iteration.

3.1 Calculating C

This step essentially realizes sparse coding. At the $(k+1)$ iteration, we are given $\mathbf{C}^{(k)}$, $\mathbf{A}^{(k)}$, $\mathbf{R}^{(k)}$, then we have

$$\mathbf{C}^{(k+1)} \in \mathrm{Prox}_{\mu_c^{(k)}}^{G}(\mathbf{C}^{(k)} - \frac{1}{\mu_c^{(k)}}\nabla_{\mathbf{C}}H(\mathbf{A}^{(k)}, \mathbf{C}^{(k)}, \mathbf{R}^{(k)})),$$

where $\mu_c^{(k)}$ is appropriately selected step size, and the gradient is calculated as

$$\nabla_{\mathbf{C}}H(\mathbf{A},\mathbf{C},\mathbf{R}) = -2\mathbf{A}^T\mathbf{K}_{YY}(\mathbf{I}-\mathbf{R}) + \mathbf{A}^T\mathbf{K}_{YY}\mathbf{AC}.$$

Denote $\mathbf{C}^* = \mathbf{C}^{(k)} - \frac{1}{\mu_c^{(k)}}\nabla_{\mathbf{C}}H(\mathbf{A}^{(k)}, \mathbf{C}^{(k)}, \mathbf{R}^{(k)})$, then we have

$$\mathbf{C}^{(k+1)} \in \underset{\mathbf{C}\in\mathcal{C}}{\mathrm{argmin}} \frac{\mu_c^{(k)}}{2\lambda_1}||\mathbf{C} - \mathbf{C}^*||_2^2 + ||\mathbf{C}||_0. \tag{6}$$

The (i,j)-element of $\mathbf{C}^{(k+1)}$ can then be easily obtained as

$$\mathbf{C}_{ij}^{(k+1)} = \begin{cases} \min\{\mathbf{C}_{ij}^*, c_m\} & \mathbf{C}_{ij}^* \ge \sqrt{2\lambda_1/\mu_c^{(k)}} \\ 0 & \text{Otherwise.} \end{cases} \tag{7}$$

3.2 Calculating A

This step tries to update the dictionary coefficient matrix. Since $\mathbf{A}$ is column normalization, we can update it column by column as $\mathbf{a}_j^{(k+1)} \in \mathrm{Prox}_{\mu_{a_j}^{(k)}}^{F(\tilde{\mathbf{A}}^{(k)})}(\mathbf{a}_j^{(k)} - \frac{1}{\mu_{a_j}^{(k)}}\nabla_{\mathbf{a}_j}H(\tilde{\mathbf{A}}^{(k)}, \mathbf{C}^{(k+1)}, \mathbf{R}^{(k)}))$, where

$$\begin{cases} \bar{\mathbf{A}}^{(k)} = [\mathbf{a}_1^{(k+1)}, \cdots, \mathbf{a}_{j-1}^{(k+1)}, \mathbf{a}_j, \mathbf{a}_{j+1}^{(k)} \cdots, \mathbf{a}_K^{(k)}] \\ \tilde{\mathbf{A}}^{(k)} = [\mathbf{a}_1^{(k+1)}, \cdots, \mathbf{a}_{j-1}^{(k+1)}, \mathbf{a}_j^{(k)}, \mathbf{a}_{j+1}^{(k)} \cdots, \mathbf{a}_K^{(k)}]. \end{cases}$$

In addition, $\mu_{a_j}^{(k)}$ is appropriately selected step-size and the gradient is calculated as $\nabla_{\mathbf{a}_j}H(\mathbf{A},\mathbf{C},\mathbf{R}) = -2\mathbf{K}_{YY}(\mathbf{I}-\mathbf{R})\mathbf{C}^T\mathbf{q}_j + 2\mathbf{K}_{YY}\mathbf{ACC}^T\mathbf{q}_j$, where $\mathbf{q}_j$ is a K-dimensional vector of which the j-th element is 1 and the other elements are zeros.

Denote $\mathbf{a}_j^* = \mathbf{a}_j^{(k)} - \frac{1}{\mu_{a_j}^{(k)}}\nabla_{\mathbf{a}_j}H(\tilde{\mathbf{A}}^{(k)}, \mathbf{C}^{(k+1)}, \mathbf{R}^{(k)})$, then we have

$$\mathbf{a}_j^{(k+1)} \in \underset{||\mathbf{a}||_2=1}{\mathrm{argmin}} ||\mathbf{a} - \mathbf{a}_j^*||_2^2, \tag{8}$$

and the solution is easily obtained as $\mathbf{a}_j^{(k+1)} = \mathbf{a}_j^*/||\mathbf{a}_j^*||_2$. After obtaining $\mathbf{a}_j^{(k+1)}$ for $j = 1, 2, \cdots, K$, we can naturally obtain the dictionary coefficient matrix solution $\mathbf{A}^{(k+1)} = [\mathbf{a}_1^{(k+1)}, \cdots, \mathbf{a}_K^{(k+1)}]$.

3.3 Calculating R

Given $\mathbf{C}^{(k+1)}$, $\mathbf{A}^{(k+1)}$ and $\mathbf{R}^{(k)}$, the value of $\mathbf{R}$ can be updated as $\mathbf{R}^{(k+1)} \in \text{Prox}_{\mu_r^{(k)}}^{Q}(\mathbf{R}^{(k)} - \frac{1}{\mu_r^{(k)}}\nabla_{\mathbf{R}} H(\mathbf{A}^{(k+1)}, \mathbf{C}^{(k+1)}, \mathbf{R}^{(k)}))$, where $\mu_r^{(k)}$ is appropriately selected step size and the gradient is calculated as

$$\nabla_{\mathbf{R}} H(\mathbf{A}, \mathbf{C}, \mathbf{R}) = -2\mathbf{K}_{YY} + 2\mathbf{K}_{YY}\mathbf{A}\mathbf{C} + 2\mathbf{K}_{YY}\mathbf{R}.$$

Denote $\mathbf{R}^* = \mathbf{R}^{(k)} - \frac{1}{\mu_r^{(k)}}\nabla_{\mathbf{R}} H(\mathbf{A}^{(k+1)}, \mathbf{C}^{(k+1)}, \mathbf{R}^{(k)})$, then we have

$$\mathbf{R}^{(k+1)} \in \underset{\mathbf{R}\in\mathcal{R}}{\text{argmin}} \frac{\mu_r^{(k)}}{2\lambda_2}||\mathbf{R} - \mathbf{R}^*||_2^2 + ||\mathbf{r}||_0. \quad (9)$$

The solution can be obtained as

$$\mathbf{R}_{ii}^{(k+1)} = \begin{cases} \min\{\mathbf{R}_{ii}^*, r_m\} & \mathbf{R}_{ii}^* \geq \sqrt{2\lambda_2/\mu_r^{(k)}} \\ 0 & \text{Otherwise.} \end{cases} \quad (10)$$

3.4 Selection of step sizes

The step-sizes $\mu_c^{(k)}$, $\mu_{a_j}^{(k)}$, $\mu_r^{(k)}$ should be determined during the solving procedure. To this end, we need to calculate the Lipschitz constants $L_c^{(k)}$, $L_{a_j}^{(k)}$ and $L_r^{(k)}$ which satisfy $||\nabla_{\mathbf{C}} H(\mathbf{A}^{(k)}, \mathbf{C}_1, \mathbf{R}^{(k)}) - \nabla_{\mathbf{C}} H(\mathbf{A}^{(k)}, \mathbf{C}_2, \mathbf{R}^{(k)})||_2 \leq L_c^{(k)}||\mathbf{C}_1 - \mathbf{C}_2||_2$, $||\nabla_{\mathbf{a}_j} H(\mathbf{a}_{j1}, \mathbf{C}^{(k)}, \mathbf{R}^{(k)}) - \nabla_{\mathbf{a}_j} H(\mathbf{a}_{j2}, \mathbf{C}^{(k)}, \mathbf{R}^{(k)})||_2 \leq L_{a_j}^{(k)}||\mathbf{a}_{j1}^{(k)} - \mathbf{a}_{j2}^{(k)}||_2$, and $||\nabla_{\mathbf{R}} H(\mathbf{A}^{(k)}, \mathbf{C}^{(k)}, \mathbf{R}_1) - \nabla_{\mathbf{R}} H(\mathbf{A}^{(k)}, \mathbf{C}^{(k)}, \mathbf{R}_2)||_2 \leq L_r^{(k)}||\mathbf{R}_1 - \mathbf{R}_2||_2$, respectively. By simple calculation, we can select $L_c^{(k)} = ||\mathbf{A}^{(k)T}\mathbf{K}_{YY}\mathbf{A}^{(k)}||_2$, $L_{a_j}^{(k)} = 2||\mathbf{K}_{YY}||_2||\mathbf{C}^{(k)}\mathbf{C}^{(k)T}\mathbf{q}_j||_2$, and $L_r^{(k)} = 2||\mathbf{K}_{YY}||_2$.

After obtaining the values of $L_c^{(k)}$, $L_{a_j}^{(k)}$, and $L_r^{(k)}$, we can determine the step-sizes as $\mu_c^{(k)} = \max(\rho L_c^{(k)}, \mu_{\min})$, $\mu_{a_j}^{(k)} = \max(\rho L_{a_j}^{(k)}, \mu_{\min})$ and $\mu_r^{(k)} = \max(\rho L_r^{(k)}, \mu_{\min})$, where $\rho > 1$ and $\mu_{\min}$ is a prescribed upper bound. In our work, we set $\rho = 1.1$ and $\mu_{\min} = 0.1$.

From the above derivation we can clearly see that the sequences $\{L_c^{(k)}\}$, $\{L_{a_j}^{(k)}\}$ and $\{L_r^{(k)}\}$ are all bounded sequence because the sequences $\{\mathbf{C}^{(k)}\}$ and $\{\mathbf{A}^{(k)}\}$, and the value of $\mathbf{K}_{YY}$ are bounded.

3.5 Algorithm analysis

The iteration procedure developed in the above sections presents excellent property which is summarized in the following theorem.

Theorem 1: The solution sequence $\{\mathbf{A}^{(k)}, \mathbf{C}^{(k)}, \mathbf{R}^{(k)}\}$, which is generated by the iteration procedure in Eqs.(7), (8) and (10), is a Cauchy sequence and converges to a critical point of (4).

Proof: The proof is essentially based on the results in [Bao *et al.*, 2014] and [Bolt *et al.*, 2013]. Here we provide a straightforward sketch proof. First, the objective functions $H(\mathbf{A}, \mathbf{C}, \mathbf{R})$, $F(\mathbf{A})$, $G(\mathbf{C})$ and $Q(\mathbf{R})$ are obviously semi-algebraic functions and therefore the whole objective function in (4), which can be written as $H(\mathbf{A}, \mathbf{C}, \mathbf{R}) + F(\mathbf{A}) + G(\mathbf{C}) + Q(\mathbf{R})$, is a semi-algebraic function, and therefore satisfy the *Kurdyka-Lojasiewicz* property. Secondly, the sequence $\{\mathbf{A}^{(k)}, \mathbf{C}^{(k)}, \mathbf{R}^{(k)}\}$ is bounded and the step-sizes $\mu_c^{(k)}$, $\mu_{a_j}^{(k)}$ and $\mu_r^{(k)}$ are bounded. Thirdly, $\nabla H(\mathbf{A}, \mathbf{C}, \mathbf{R}) = (-2\mathbf{K}_{YY}(\mathbf{I}-\mathbf{R})\mathbf{C}^T + 2\mathbf{K}_{YY}\mathbf{A}\mathbf{C}\mathbf{C}^T, -2\mathbf{A}^T\mathbf{K}_{YY}(\mathbf{I}-\mathbf{R}) + \mathbf{A}^T\mathbf{K}_{YY}\mathbf{A}\mathbf{C}, -2\mathbf{K}_{YY} + 2\mathbf{K}_{YY}\mathbf{A}\mathbf{C} + 2\mathbf{K}_{YY}\mathbf{R})$ has Lipschitz constant on any bounded set. Therefore, by adopting the results of Theorem 6.1 in [Bao *et al.*, 2014], the solution sequence is a Cauchy sequence and converges to a critical point of (4).

The conclusion in Theorem 1 clearly illustrates that the sequence $\{\mathbf{A}^{(k)}, \mathbf{C}^{(k)}, \mathbf{R}^{(k)}\}$ is a sequence whose elements become arbitrarily close to each other as the sequence progresses. Therefore, the developed algorithm achieves whole sequence convergence. Such a property provides important practical values since the number of iterations does not need to be determined empirically[Bao *et al.*, 2014].

4 Experimental results

In this section we present several experimental results demonstrating the effectiveness of the proposed robust kernel dictionary learning.

4.1 Synthetic Data

The synthetic data is used to illustrate the outlier rejection role of the proposed method. Similar to the setting in [Nguyen *et al.*, 2013], we learn the dictionary from 500 data sample set $\{\mathbf{y}_i\}_{i=1}^{500}$ that is generated from a 2-dimensional parabola $\{\mathbf{y}_i = [y_{i,1}, y_{i,2}]^T \in R^2 | y_{i,2} = y_{i,1}^2\}$. To corrupt the data, we replace $\mathbf{y}_j$ for $j = \{120, 220, 320, 420\}$ as newly generated random vectors. Therefore 4 outliers are incorporated into the sample set. We then preprocess all the samples to have unit L_2 norm,

We use a polynomial kernel of degree 2 and set the size of the dictionary to be 20. For K-KSVD, the sparsity is set to 5. For the proposed method, we solve the optimization problem in Eq.(2) with $\lambda_1 = 0.01$ and $\lambda_2 = 0.001$.

Fig.1 shows the results. The left panel shows the convergence behavior of the proposed method. It shows that the increment indeed converges to zero after several iterations. We also show the convergence behavior of the sparse coding coefficient matrix of K-KSVD method in the right-top panel, which illustrates that the generated sequence does not converge to zero. This validates that K-KSVD has at most subsequence convergence, while the proposed method has whole sequence convergence. Finally, we show the elements of the vector $\mathbf{r}$ which is the diagonal elements of $\mathbf{R}$ in the right-bottom panel. It clearly shows that the imposed outlier can be reliably isolated, while K-KSVD has no capability to isolate outliers. The results clearly indicate the effectiveness of the proposed algorithm.

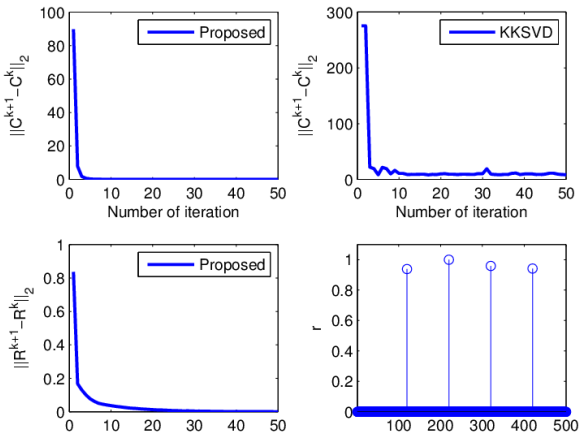

Figure 1: The results of the synthetic example.

4.2 Dataset

In this section we use 4 publicly available datasets to compare several dictionary learning methods. For USPS dataset, we use the vector feature and Gaussian kernel $\kappa(\mathbf{y}_i, \mathbf{y}_j) = \exp(-\gamma||\mathbf{y}_i - \mathbf{y}_j||_2^2)$. For the rest three datasets, we calculate specified feature vector $\mathbf{f}_{u,v}$ for each pixel at (u, v) in one image and use the popular Region Covariance Descriptor (RCovD)[Hussein *et al.*, 2013] to represent the image. Since RCovDs do not lie in Euclidean space, we use the Log-Euclidean kernel $\kappa(\mathbf{y}_i, \mathbf{y}_j) = \exp(-\gamma||\text{logm}(\mathbf{y}_i) - \text{logm}(\mathbf{y}_j)||_2^2)$[Li *et al.*, 2013]. By such settings we show that the kernel dictionary learning can be adapted different kernel forms. In all of the experiments, the parameter is set as $\gamma = 0.02$. In the following we list some details about the datasets.

1. USPS digit recognition[Nguyen *et al.*, 2013]: This dataset contains 10 classes of 256-dimensional handwritten digits. For each class, we randomly select $N = 500$ samples for training and 200 samples for testing. This setting was suggested in [Nguyen *et al.*, 2013] for K-KSVD method.
2. Virus image classification[Harandi and Salzmann, 2015]: This dataset includes 15 different classes. Each class has 100 images of size 41×41. For each class, we randomly select $N = 80$ samples for training and 20 samples for testing. At each pixel (u, v) of an image, we compute the 25-dimensional feature vector $\mathbf{f}_{u,v} = [I_{u,v}, |\partial I/\partial u|, |\partial I/\partial v|, |\partial^2 I/\partial u^2|, |\partial^2 I/\partial v^2|, |G_{u,v}^{0,0}|, \cdots, |G_{u,v}^{4,5}|]^T$, where $I_{u,v}$ is the intensity value, $G_{u,v}^{o,s}$ is the response of a 2D Gabor wavelet with orientation o and scale s, and $|\cdot|$ denotes the magnitude of a complex value. Therefore, we generated 20 Gabor filters at 4 orientations and 5 scales.
3. Kylberg texture classification[Kylberg, 2011]: This dataset includes 28 classes. Each class has 160 samples which are resized as 128×128. For each class, we randomly select $N = 80$ samples for training and 80 samples for testing. At each pixel (u, v) of an image, we compute the 5-dimensional feature vector $\mathbf{f}_{u,v} = [I_{u,v}, |\partial I/\partial u|, |\partial I/\partial v|, |\partial^2 I/\partial u^2|, |\partial^2 I/\partial v^2|]^T$, where $I_{u,v}$ is the intensity value.
4. UCMERCED scene classification[Yang and Newsam, 2010]: This dataset includes 21 challenging scene categories with 100 samples per class. For each class, we randomly select $N = 80$ samples for training and 20 samples for testing. At each pixel (u, v) of an image, we compute the 15-dimensional feature vector $\mathbf{f}_{u,v} = [\mathbf{f}_{R,u,v}^T, \mathbf{f}_{G,u,v}^T, \mathbf{f}_{B,u,v}^T]^T$, where $\mathbf{f}_{C,u,v}^T = [I_{C,u,v}, |\partial I_C/\partial u|, |\partial I_C/\partial v|, |\partial^2 I_C/\partial u^2|, |\partial^2 I_C/\partial v^2|]$ and I_C is the intensity image for the C channel and $C \in \{R, G, B\}$ represents one of the color channel.

In all of the above datasets, the training/testing split is randomly repeated for 10 times and the average results are reported. In addition, we generate corresponding corrupt datasets by selecting $\rho\%$ training samples in each class and replace them with the corresponding outlier images. The outlier image is produced by overlaying a rand noise block of which area is a half of the original image. The value of ρ is selected from $\{0, 10, 20, 30, 40, 50\}$.

For comparison, we designed the following dictionary learning methods:

1. randDict: This method just randomly selects K atoms subset $\{\mathbf{s}_1, \cdots, \mathbf{s}_K\}$ from the training sample set $\{\mathbf{y}_i\}_{i=1}^N$to construct the dictionary.
2. K-Means method: For vector descriptor, we just use the conventional k-means clustering method to get the K dictionary atoms. For RCovD, we first map the covariance matrices to the linear space by matrix logarithm, in which the clustering is performed and the results are then mapped back to the Symmetry-Positive-Definite manifold. As a result, we also get the dictionary as $\{\mathbf{s}_1, \cdots, \mathbf{s}_K\}$.
3. GRADient method. For the dataset using Gaussian kernel, we follow the method in [Gao *et al.*, 2010] to learn the dictionary. For the datasets using Log-Euclidean kernel, we follow the method in [Li *et al.*, 2013] to learn the dictionary. The dictionary is also denoted as $\{\mathbf{s}_1, \cdots, \mathbf{s}_K\}$.
4. K-KSVD method: This method was originally developed in [Nguyen *et al.*, 2013]. It utilizes the kernelized KSVD method to learn the dictionary. In this case, the obtained dictionary is denoted as $\Phi(\mathbf{Y})\mathbf{A} \in R^{d \times K}$. The default parameter suggested in [Nguyen *et al.*, 2013] are used for this implementation.

As to the proposed method, we fix the regularization parameters $\lambda_1 = 0.001$ and $\lambda_2 = 0.0001$ and the maximum iteration number is 100. Please note that the dictionaries in K-KSVD and the proposed method are learned in the feature space, while randDict, K-Means and GRAD methods learn the dictionaries in the input space. Nevertheless, the relation $\{\mathbf{s}_i\}_{i=1}^K \subset \{\mathbf{y}_i\}_{i=1}^N$ holds for randDict only.

Because the dictionary learning is essentially non-convex problem, for GRAD, K-KSVD and the proposed one, we randomly initialize the dictionary ten times and pick the one with

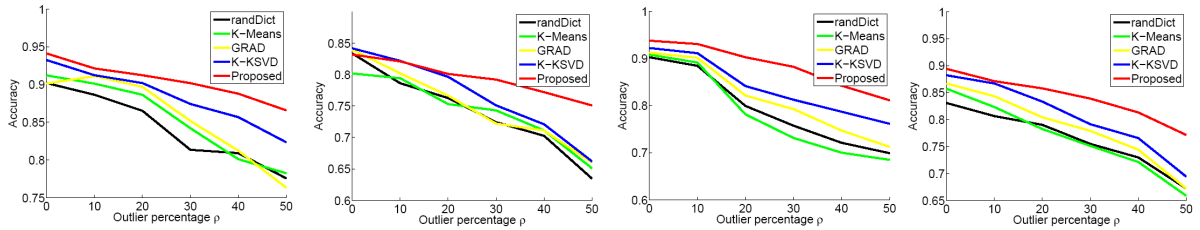

Figure 2: The results on the public datasets. From left to right: USPS, Virus, Kylberg, UCMERCED.

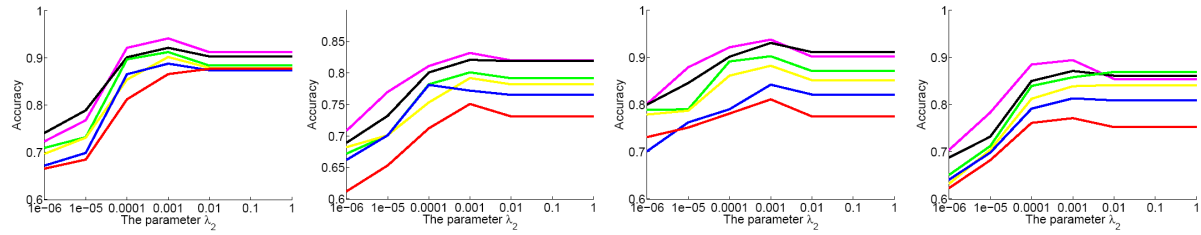

Figure 3: The influences of λ_2. From left to right: USPS, Virus, Kylberg, UCMERCED. Legend: $\rho = 0$, $\rho = 10$, $\rho = 20$, $\rho = 30$, $\rho = 40$, $\rho = 50$.

minimum reconstruction error over the training set. In all our experiments, we fix the size of the dictionary to be $K = 30$. Since all of the compared methods except the proposed one does not equip outlier rejection capability, we resort the classification task to evaluate their performance. We use similar approaches as in [Nguyen *et al.*, 2013] for classification. The algorithm starts with learning a dictionary for each class. In particular, we aggregate all training images for each category into $\mathbf{Y}_c = \{\mathbf{y}_{c,1}, \cdots, \mathbf{y}_{c,N_c}\}$, where c represents the c-th class and N_c is the number of the samples in the c-th class. Let $\mathbf{S}_c = \{\mathbf{s}_{c,1}, \cdots, \mathbf{s}_{c,K}\}$ denotes the learned dictionary in the original space for the $c-$th class. Such a dictionary can be obtained by using the methods of randDict, K-Means, and GRAD. For K-KSVD and the proposed one, the dictionaries are learned in the feature space and therefore we use $\mathbf{D}_c = \Phi(\mathbf{Y}_c)\mathbf{A}_c$ to denote the learned kernel dictionary for the c-th class, where $\mathbf{A}_c$ is the learned coefficient matrix. Given a query sample $\mathbf{z}$, we first perform the spare coding for each $\mathbf{D}_c$ to get the sparse code $\mathbf{c}_c$. The sparse setting is the same as the training phase. The reconstruction error for the c-th class is denoted as $r_c = ||\Phi(\mathbf{z}) - \Phi(\mathbf{S}_c)\mathbf{c}_c||_2^2$ for randDict, K-Means and Grad, and $r_c = ||\Phi(\mathbf{z}) - \Phi(\mathbf{Y}_c)\mathbf{A}_c\mathbf{c}_c||_2^2$ for K-KSVD and the proposed one. Finally, The test sample is simply classified to the class that gives the smallest reconstruction error.

The classification accuracies on the 4 datasets are reported in Fig.2. It is not surprising that all the curves are monotonically decreasing along the increasing of ρ. From those results we summarize the following observations: (1)K-Means and GRAD do not show significant advantages over randDict. In some especial cases when ρ is large, their performances are even worse than randDict. The main reason is that the dictionary atoms in K-Means and GRAD are learned but not selected and all of atoms are contaminated by the outliers. This means the learning procedures in k-means clustering and gradient learning are strongly affected by the outliers. On the contrary, randDict selects some existing training samples to be the dictionary atoms and therefore some atoms may be clean samples. (2) The performance of the proposed method and K-KSVD is consistently better than that of the other methods. The reason is partially due to the fact that both methods learn dictionaries in the feature space, but not the input space. (3) When the outlier level ρ is small, the performance difference between the proposed method and K-KSVD is not significant. However, with the increasing of ρ, the advantage of the proposed method becomes obvious. The intrinsic reason is that the proposed dictionary learning method explicitly isolates the outliers in the training samples. In this sense, the proposed method learns a cleaner dictionary.

Empirically the proposed algorithm works well when the parameter λ_2 is in the interval $[10^{-5}, 10^{-3}]$. When λ_2 is too large, the outlier cannot be isolated. On the contrary, when λ_2 is too small, too many training samples may be mistakenly isolated as outliers. To study the influence of λ_2, we fix $\lambda_1 = 0.001$ and vary λ_2 from 10^{-6} to 1 and run the algorithm on the 4 datasets. In Fig.3 we show the classification accuracies corresponding to different noise levels. The results show that the proposed algorithm is not very sensitive to the parameter λ_2 and a properly-designed robust term indeed plays important role in dictionary learning. In addition, the role of outlier isolation diminishes when λ_2 is larger than 10^{-2}.

5 Conclusions

In this paper, we propose an optimization model for robust kernel dictionary learning which has capability to isolate the outliers. A whole sequence convergent algorithm is developed to solve the non-convex optimization problem and the

experimental results show that the proposed robust kernel dictionary learning method provides significant performance improvement.

References

[Anaraki and Hughes, 2013] Farhad Pourkamali Anaraki and Shannon M. Hughes. Kernel compressive sensing. *ICIP*, pages 494–498, 2013.

[Bao *et al.*, 2014] Chenlong Bao, Hui Ji, Yuhui Quan, and Zuowei Shen. l_0 norm based dictionary learning by proximal methods with global convergence. *CVPR*, pages 1–8, 2014.

[Bolt *et al.*, 2013] Jerome Bolt, Shoham Sabach, and Marc Teboulle. Proximal alternating linearized minimizing for nonconvex and nonsmooth problems. *Mathematical Programming*, pages 459–494, 2013.

[Chen and Wu, 2013] Zhouyuan Chen and Ying Wu. Robust dictionary learning by error sourse decomposition. *ICCV*, pages 2216–2223, 2013.

[Cheng *et al.*, 2009] Hong Cheng, Zicheng Liu, and Jie Yang. Sparsity induced similarity measure for label propagation. *ICCV*, pages 1–8, 2009.

[Cheng *et al.*, 2013] Hong Cheng, Zicheng Liu, Lu Yang, and Xuewen Chen. Sparse representation and learning in visual recognition: Theory and applications. *Signal Processing*, pages 1408–1425, 2013.

[Gao *et al.*, 2010] Shenghua Gao, Ivor Wai-Hung Tsang, and Liang-Tien Chia. Kernel sparse representation for image classification and face recognition. *ECCV*, pages 1–14, 2010.

[Gao *et al.*, 2013] Shenghua Gao, Ivor Wai-Hung Tsang, and Liang-Tien Chia. Sparse representation with kernels. *IEEE Trans. on Image Processing*, pages 423–434, 2013.

[Harandi and Salzmann, 2015] Mehrtash Harandi and Mathieu Salzmann. Riemannian coding and dictionary learning: Kernels to the rescue. *CVPR*, 2015.

[Harandi *et al.*, 2012] Mehrtash Harandi, Conrad Sanderson, Richard Hartley, and Brian Lovell. Sparse coding and dictionary learning for symmetric positive definite matrices: A kernel approach. *ECCV*, pages 216–229, 2012.

[Hussein *et al.*, 2013] Mohamed E. Hussein, Marwan Torki, Mohammad A. Gowayyed, and Motaz El-Saban. Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. *IJCAI*, pages 2466–2472, 2013.

[Kim, 2014] Minyoung Kim. Efficient kernel sparse coding vis first-order smooth optimization. *IEEE Trans. on Neural Networks and Learning Systems*, 25(8):1447–1459, August 2014.

[Kong *et al.*, 2011] Deguang Kong, Chris Ding, and Heng Huang. Robust nonnegtive matrix factorization using l21-norm. *CIKM*, pages 673–682, 2011.

[Kylberg, 2011] Gustaf Kylberg. The kylberg texture dataset v. 1.0. *External report (Blue series) 35, Centre for Image Analysis, Swedish University of Agricultural Sciences and Uppsala University, Uppsala, Sweden,*, 2011.

[Li and Ngom, 2012] Yifeng Li and Alioune Ngom. Fast kernel sparse representation approaches for classification. *ICDM*, 2012.

[Li *et al.*, 2013] Peihua Li, Qilong Wang, Wangmeng Zuo, and Lei Zhang. Log-euclidean kernels for sparse represenation and dicionary learning. *ICCV*, pages 1601–1608, 2013.

[Liu *et al.*, 2014] Baodi Liu, Yuxiong Wang, Bin Shen, and Yujin Zhang. Self-explanatory sparse representation for image classification. *ECCV*, pages 600–616, 2014.

[Liu *et al.*, 2015] Huaping Liu, Yunhui Liu, and Fuchun Sun. Robust exemplar extraction using structured sparse coding. *IEEE Trans. on Neural Networks and Learning Systems*, 2015.

[Ma *et al.*, 2014] Rui Ma, Huaping Liu, Fuchun Sun, Qingfen Yang, and Meng Gao. Linear dynamic system method for tactile object classification. *Science China Information Sciences*, pages 1–11, 2014.

[Nguyen *et al.*, 2013] Hien Nguyen, Vishas Patel, Nasser Nasrabadi, and Rama Chellappa. Design of non-linear kernel dictionaries for object recognition. *IEEE Trans. on Image Processing*, pages 5123–5135, 2013.

[Nie *et al.*, 2013] Feiping Nie, Hua Wang, Heng Huang, and Chris Ding. Early active learning via robust representation and structured sparsity. *IJCAI*, pages 1572–1578, 2013.

[Pan *et al.*, 2014] Qihe Pan, Deguang Kong, Chris Ding, and Bin Luo. Robust non-negative dicionrarty learning. *AAAI*, pages 2027–2033, 2014.

[Wang *et al.*, 2014] Ling Wang, Hong Cheng, Zicheng Liu, and Ce Zhu. A robust elastic net approach for feature learning. *Journal of Visual Communication and Image Representation*, pages 313–321, 2014.

[Xia *et al.*, 2012] Zhichen Xia, Chiris Ding, and Edmond Chow. Robust kernel nonnegative matrix factorization. *ICDMW*, pages 522–529, 2012.

[Xie *et al.*, 2013] Yuchen Xie, Jeffrey Ho, and baba Vemuri. On a nonlinear generalization of sparse coding and dictionrary learning. *ICML*, pages 1–8, 2013.

[Yang and Newsam, 2010] Yi Yang and Shawn Newsam. Bag-of-visual-words and spatial extensions for land-use classification. *ACM Int. Conf. Adv. Geogr. Inf. Syst.*, pages 270–279, 2010.

[Yang *et al.*, 2015] Lu Yang, Hong Cheng, Jianan Su, and Xuelong Li. Pixel-to-model distance for robust background reconstruction. *IEEE Trans. on Circuits and Systems for Video Technology*, 2015.

[Zhang *et al.*, 2015] Shengping Zhang, Shiva Kasiviswanathan, Pong C Yuen, and Mehrtash Harandi. Online dictionary learning on symmetric positive definite manifolds with vision applications. *AAAI*, pages 1–6, 2015.