Sparse Kernel Dictionary Learning

Cian O'Brien and Mark D. Plumbley
Centre for Vision, Speech and Signal Processing,
University of Surrey, UK.
{cj.obrien, m.plumbley}@surrey.ac.uk

***Abstract*—Dictionary Learning has proven to be a useful tool in many signal processing and machine learning applications, producing compelling results. An important use-case is as an unsupervised method of learning latent representations that can be used as the input features to a larger supervised classification system. The basic algorithm relies on inner-products between the input signals and a set of atomic dictionary elements and as such is amenable to kernel methods, which have been found to be very powerful in techniques such as non-linear Support Vector Machines and Kernel Regression. Based on previous kernel Dictionary Learning approaches, in this work we propose *Sparse Kernel Dictionary Learning* which provides significant gains in efficiency over its non-sparse counterpart. Additionally, we consider the online setting in which data arrive sequentially and demonstrate how sparse Dictionary Learning with kernels can be scaled up to extremely large datasets.**

***Keywords*—dictionary learning, sparse coding, kernel methods.**

I. Introduction

Many machine learning and signal processing applications rely on informative representations of signals. Dictionary learning has proven to be a successful strategy for learning parts-based representations which can be used as a feature extraction step in classification systems. We consider dictionary learning in non-linear spaces via the popular kernel trick, which while powerful introduces several technical difficulties in a large-data setting. In Sections (II)-(III) we introduce the dictionary learning algorithm and its kernel extension. In Section (V) we introduce our approach for learning with large amounts of data. We conclude with a set of experiments.

II. Dictionary Learning

Dictionary Learning (DL) aims to find a set of *atomic* or *basis* elements which can be used to efficiently represent a set of signals. For a collection $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ this process takes the form of the following optimization problem [1][2]

$$\min_{\mathbf{D}\in\mathcal{S},\{\mathbf{z}\}_{i=1}^n} \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}\|\mathbf{x}_i - \mathbf{D}\mathbf{z}_i\|_F^2 + \lambda\|\mathbf{z}_i\|_1 \qquad (1)$$

where $\mathbf{D}$ is a matrix whose columns contain the dictionary atoms, the collection $\{\mathbf{z}\}_{i=1}^n$ are the latent representations for each input signal and $\mathcal{S}$ is the set of matrices whose columns have at most unit norm. The ℓ_1-norm constraint on $\{\mathbf{z}\}_{i=1}^n$ is used to encourage *sparsity* in the representations, leading to vectors with few non-zero elements so that each input $\mathbf{x}$ can be represented as linear combination of only a few atoms from $\mathbf{D}$. In a classification context, the sparse codes $\mathbf{z}$ can be used as input features.

III. Kernel Methods and Dictionary Learning

Kernel methods have been used extensively in machine learning to deal with non-linear data distributions, most notably in non-linear Support Vector Machines (SVM). These approaches rely on the so-called "kernel trick" to *implicitly* transform a set of signals using a non-linear mapping Φ into a higher dimensional space in which different classes may be linearly separable. The key insight used is that many linear algorithms rely on the inner-product between vectors to calculate a notion of similarity. Therefore, if there exists an efficient method of calculating these inner-products in the new feature space we can summarize them using a kernel matrix (i.e. the Gram matrix in feature space), avoiding the need to explicitly calculate the mapped features – which can have possibly infinite row-dimension for some choices of Φ.

Kernel methods have been applied to DL by Van Nguyen *et al*. [3][4]; given a non-linear feature mapping $\Phi : \mathbf{x} \mapsto \Phi(\mathbf{x})$, they state the Kernel Dictionary Learning (KDL) problem as

$$\min_{\mathbf{C},\{\mathbf{z}\}_{i=1}^n} \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}\|\tilde{\mathbf{x}}_i - \tilde{\mathbf{X}}\mathbf{C}\mathbf{z}_i\|_F^2 \quad \text{such that} \quad \forall i\ \|\mathbf{z}_i\|_0 \leq c \qquad (2)$$

where $\tilde{\mathbf{X}}\mathbf{C} \in \mathcal{S}$, $\|\cdot\|_0$ is the ℓ_0 psuedo-norm which counts the number of non-zero elements in a vector, $c \in \mathbb{N}_+$ is a fixed sparsity level and we use the notation $\tilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_n] = [\Phi(\mathbf{x}_1), \dots, \Phi(\mathbf{x}_n)]$. In this formulation the complete dictionary is composed of a *base dictionary* $\tilde{\mathbf{X}}$ (consisting of the entire dataset) together with a *coefficient dictionary* $\mathbf{C}$. This setup is reminiscent of the "doubly sparse" approach of Rubenstein *et al*. [5] where an additional dictionary is learned over a base dictionary which has an efficient implementation. By representing the dictionary as a linear combination of the full set of signals, the resulting optimization problem depends on the kernel matrix $\mathcal{K}_{\mathbf{X}} \in \mathbb{R}^{n\times n}$ whose (i,j)-th entry is equal to $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \langle\Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j)\rangle$ for some kernel function κ, which makes the learning tractable since the mapped signals $\tilde{\mathbf{x}}$ are never explicitly evaluated – only their inner-products in feature space via κ. Examples of possible kernel functions κ include Radial Basis Function kernels $\kappa(\mathbf{x}, \mathbf{y}) = \exp(-\gamma\|\mathbf{x} - \mathbf{y}\|^2)$, Polynomial kernels $\kappa(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^\top\mathbf{y} + c)^d$ and the curiously named Sigmoid kernels $\kappa(\mathbf{x}, \mathbf{y}) = \tanh(\mathbf{x}^\top\mathbf{y} + c)$. The primary issue with KDL – and one which is shared by kernel methods in general – is that one is required to maintain the full kernel matrix which has dimension $(n \times n)$ and manipulating such large and dense matrices can be computationally expensive, making the technique impractical for moderately large data sets.

IV. Related Work

This work is primarily inspired by the Kernel Dictionary Learning algorithm of Van Nguyen *et al*., who also noted the link between their work and the double-sparsity model of Rubenstein *et al*. However they did not consider sparsity on the coefficient dictionary, which is useful when trying to derive an efficient algorithm based on the full signal kernel matrix. Motivated by the desire for a scalable approach, we make use of ideas from Online Dictionary Learning as proposed by Mairal and Bach (which is in turn is related to stochastic dictionary learning [6]). When combined with kernel methods, this has clear links with other online kernel approaches in the literature [7], for example Online Kernel Least Squares [8]. Kim proposed supervised KDL using online updates, however in their work the base dictionary is allowed to grow without bound since the new signal is included at every iteration, which limits its use in truly large scale learning (not to mention the lack of dictionary sparsity) [9]. Recently, Sulam *et al*. consider the problem of learning a sparse dictionary in large scale problems, however they rely on stochastic updates and don't consider the use of kernels [10].

Our contributions are (i) the explicit inclusion of sparsity in the coefficient dictionary based on the ℓ_1-norm penalty and (ii) the development of online kernel methods in a sparse DL framework using (iii) a novel selection condition for new signals which limits the growth of the base dictionary (disregarding noise, in practice it will be upper-bounded by the intrinsic dimension of the signals). Together, these approaches lead to a faster algorithm for fixed size problems and enable KDL to gracefully scale to extremely large datasets.

V. Sparse Kernel Dictionary Learning

In this section we introduce *Sparse Kernel Dictionary Learning* (sKDL), which aims to address the difficulties in scalability of Kernel Dictionary Learning. The main idea is that using the full data set as a base dictionary is highly redundant. Instead, we use a sparse combination which greatly increases the efficiency during learning. We impose sparsity on the coefficient dictionary by an ℓ_1-norm penalty on its columns

$$\min_{\mathbf{C},\{\mathbf{z}\}_{i=1}^n} \left(\frac{1}{n} \sum_{i=1}^{n} \frac{1}{2} \left\| \tilde{\mathbf{x}}_i - \tilde{\mathbf{X}}\mathbf{C}\mathbf{z}_i \right\|_F^2 + \lambda_1 \|\mathbf{z}_i\|_1 \right) + \lambda_2 \sum_{j=1}^{d} \|\mathbf{c}_j\|_1 \tag{3}$$

where $\tilde{\mathbf{X}}\mathbf{C} \in \mathcal{S}$ and $\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_d]$. Multiplication with sparse matrices is significantly faster than with dense matrices, potentially leading to a much more efficient algorithm. The minimization problem is non-convex in its arguments jointly, but convex in each when the other is fixed and therefore we take the standard approach in DL and iteratively optimize with respect to $\mathbf{C}$ and each $\mathbf{z}_i$ separately. The primary difficulty with the optimization is the non-differentiability of the ℓ_1-norm and to deal with this we make use of a standard tool in convex optimization, namely proximal gradient methods [11][12]. Given a function $\mathcal{F}$ which can be written as a sum of convex functions $\mathcal{F}(\mathbf{z}) = f(\mathbf{z}) + g(\mathbf{z})$ with g non-differentiable we can minimize $\mathcal{F}$ with respect to $\mathbf{z}$ by splitting the optimization and making use of the proximity operator of g. In our case, (3) can be minimized by applying the iterative scheme

$$\mathbf{z}_t = \mathcal{P}\big(\mathbf{z}_{t-1} - \nabla_{\mathbf{z}} f(\mathbf{z}_{t-1})\big) \tag{4}$$

where $\mathcal{P}$ is the proximity operator for the ℓ_1-norm and $f = \sum_{i=1}^{n} \frac{1}{2}\|\tilde{\mathbf{x}}_i - \tilde{\mathbf{X}}\mathbf{C}\mathbf{z}_i\|_F^2$. For the coefficients $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_n]$ this corresponds to a kernelized version of the well-known Iterative Shrinkage-Thresholding Algorithm (kISTA) [13]. The proximity operator for the λ-regularized ℓ_1-norm is given by the entry-wise soft-thresholding function $\mathcal{P}_\lambda(x) = \text{sign}(x)(|x| - \lambda)_+$ which leads to following updates for $\mathbf{z}$ and the j-th dictionary atom $\mathbf{c}_j$

$$\mathbf{z} \leftarrow \mathcal{P}_{\lambda_1}\big(\mathbf{z} + \mathbf{C}^\top(\boldsymbol{\mathcal{K}}_{\mathbf{X}} - \boldsymbol{\mathcal{K}}_{\mathbf{X}}\mathbf{C}\mathbf{z})\big) \tag{5}$$

$$\mathbf{c}_j \leftarrow \mathcal{P}_{\lambda_2}\big(\mathbf{c}_j + (\boldsymbol{\mathcal{K}}_{\mathbf{X}} - \boldsymbol{\mathcal{K}}_{\mathbf{X}}\mathbf{c}_j\mathbf{z}_{\bar{j}})\mathbf{z}_{\bar{j}}^\top\big) \tag{6}$$

where $\boldsymbol{\mathcal{K}}_{\mathbf{X}}$ is the signal kernel matrix and $\mathbf{z}_{\bar{j}}$ is the j-th row of $\mathbf{Z}$. We iteratively update each $\mathbf{z}_i$ with fixed $\mathbf{C}$ until convergence, followed by $\mathbf{C}$ updates with fixed $\mathbf{Z}$ (after each projection step we normalize its columns – if necessary – so that $\tilde{\mathbf{X}}\mathbf{C} \in \mathcal{S}$). The convergence of both update schemes can be accelerated by similar analogy to the Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) [14], the details of which we omit for space.

A. An Online Algorithm

The proposed approach successfully learns a sparse dictionary for KDL in the case where we have access to the whole collection of signals and can store it in memory. In this section we consider the problem of sKDL when we don't have access to all of data at once – rather, we assume that each signal arrives sequentially which may be the case when using very large data sets. This setup seems at odds with standard kernel approaches (including sKDL) which rely on the Representor Theorem [15] to guarantee that the solution can be written as a linear combination of *all* of the input samples and in this setup kernel K-SVD as it is stated cannot be used.

To address this issue, we develop a selection criteria based on which new samples are either added or not to a base dictionary $\tilde{\mathbf{D}}$. First however we outline an algorithm for online updating of $\mathbf{C}$; specifically, we assume that at each time step t we are presented with one signal $\mathbf{x}_i$ from the data set $\mathcal{X}$. Next we introduce a majorizing function for the total expected cost given by

$$\mathcal{F}_t(\mathbf{C}) = \left(\frac{1}{t} \sum_{i=1}^{t} \frac{1}{2} \left\| \tilde{\mathbf{x}}_i - \tilde{\mathbf{D}}\mathbf{C}\mathbf{z}_i \right\|_F^2 + \lambda_1 \|\mathbf{z}_i\|_1 \right) + \lambda_2 \sum_{j=1}^{d} \|\mathbf{c}_j\|_1 \tag{7}$$

where $\mathbf{z}_i$ is computed using the previous dictionary and $\tilde{\mathbf{D}}$ is a base dictionary whose columns consist of selected mapped input samples which play a role somewhat analogous to that of SVM support vectors. To minimize this expression with respect to $\mathbf{c}_j$ we again use the method of proximal splitting; writing $\mathcal{F}_t = f_t + g_t$ as before and taking derivatives we have

$$f_t = \frac{1}{t} \sum_{i=1}^{t} \frac{1}{2} \text{tr}\big(\tilde{\mathbf{x}}_i\tilde{\mathbf{x}}_i^\top - 2\mathbf{C}^\top\kappa(\mathbf{D}, \mathbf{x}_i)\mathbf{z}_i^\top + \mathbf{C}^\top\boldsymbol{\mathcal{K}}_{\mathbf{D}}\mathbf{C}\mathbf{z}_i\mathbf{z}_i^\top\big) \tag{8}$$

$$\Rightarrow \nabla_{\mathbf{c}_j} f_t = -\big(\mathbf{a}_j - \boldsymbol{\mathcal{K}}_{\mathbf{D}}\mathbf{c}_j\mathbf{b}_{\bar{j}j}\big) \tag{9}$$

where the matrices $\mathbf{A} = \sum_{i=1}^{t} \kappa(\mathbf{D}, \mathbf{x}_i)\mathbf{z}_i^\top$ and $\mathbf{B} = \sum_{i=1}^{t} \mathbf{z}_i\mathbf{z}_i^\top$ keep track of the past information and $\mathbf{b}_{\bar{j}j}$ is the j-th diagonal entry of $\mathbf{B}$. Finally we arrive at the update equation

for $\mathbf{c}^j$ in the online setting

$$\mathbf{c}_j \leftarrow \mathcal{P}_{\lambda_2}\big(\mathbf{c}_j + \eta_t(\mathbf{a}_j - \mathcal{K}_{\mathbf{D}}\mathbf{c}_j\mathbf{b}_{\bar{j}j})\big) \tag{10}$$

where $\eta_t = \frac{1}{t}$ is the learning rate.

We now address the issue of how to select the base dictionary $\tilde{\mathbf{D}}$. From the incoming data we aim to choose a set of "landmark" points in real-time which are best suited for representing the signals seen so far. To see how to do this, we consider the structure of the given composite dictionary $\tilde{\mathbf{D}}\mathbf{C}$; the main observation is that the coefficients in $\mathbf{C}$ select a linear combination of atoms from the base dictionary $\tilde{\mathbf{D}}$ and therefore *the composite dictionary will always lie in the column-span of the base dictionary*. This means that $\tilde{\mathbf{D}}$ defines a potential subspace in which the data reconstructions reside, with $\mathbf{C}$ selecting the most important "directions".

With this in mind, we propose a selection step in which incoming signals are added (or not) based on how well they are represented by the current column space of $\tilde{\mathbf{D}}$ – signals which have small magnitude after projection represent subspaces which are poorly captured by the current base dictionary and should be added to the model. We propose to include the signal $\mathbf{x}_i$ if the following condition holds

$$\big\|\tilde{\mathbf{D}}(\tilde{\mathbf{D}}^\top\tilde{\mathbf{D}})^{-1}\tilde{\mathbf{D}}^\top\tilde{\mathbf{x}}_i\big\|_F^2 \leq \tau\|\tilde{\mathbf{x}}_i\|_F^2 \tag{11}$$

$$\Rightarrow \kappa(\mathbf{x}_i,\mathbf{D})\mathcal{K}_{\mathbf{D}}^{-1}\kappa(\mathbf{D},\mathbf{x}_i)\kappa(\mathbf{x}_i,\mathbf{x}_i)^{-1} \leq \tau \tag{12}$$

for some threshold $\tau \in (0,1)$. In fact, such a strategy may encourage the composite dictionary to be low-rank since minor deviations of the data from the underlying manifold will be ignored based on the size of the τ parameter and such low-rank constraints can be useful from a de-noising perspective.

Algorithm 1 Online Sparse Kernel Dictionary Learning

Input: Signals $\mathcal{X}$, sparsity parameters λ_1 and λ_2, threshold τ.
Output: Learned Dictionary $\mathbf{DC}$.

Initialization : Set $\mathbf{A},\mathbf{B} = \mathbf{0}$. Initialize $\mathbf{C}$ randomly and normalize. Initialize $\mathbf{D}$ using random signals from $\mathcal{X}$.

1: **for** $t = 1$ to n **do**
2: Draw a signal $\mathbf{x}_i$.
3: Check condition (11). If it holds, add $\mathbf{x}_i$ to $\mathbf{D}$ and append a row of zeros to $\mathbf{C}$. Reset $\mathbf{A} \leftarrow \mathbf{0}$ and $\mathbf{B} \leftarrow \mathbf{0}$. Update $\mathcal{K}_{\mathbf{D}}$.
4: Perform kernel sparse coding using kISTA (5) by iterating
$\mathbf{z} \leftarrow \mathcal{P}_{\lambda_1}\big(\mathbf{z} + \mathbf{C}^\top(\kappa(\mathbf{D},\mathbf{x}_i) - \mathcal{K}_{\mathbf{D}}\mathbf{C}\mathbf{z})\big)$
5: Update $\mathbf{A}$ and $\mathbf{B}$
$\mathbf{A} \leftarrow \mathbf{A} + \kappa(\mathbf{D},\mathbf{x}_i)\mathbf{z}_i^\top$
$\mathbf{B} \leftarrow \mathbf{B} + \mathbf{z}_i\mathbf{z}_i^\top$
6: For each j, update $\mathbf{c}_j$ using (10) with the iterate
$\mathbf{c}_j \leftarrow \mathcal{P}_{\lambda_2}\big(\mathbf{c}_j + \eta_t(\mathbf{a}_j - \mathcal{K}_{\mathbf{D}}\mathbf{c}_j\mathbf{b}_{\bar{j}j})\big)$
and normalize.
7: **end for**
8: **return** Composite dictionary $\mathbf{DC}$.

VI. Experiments

Here we outline a set of experiments to quantify the performance of sKDL, with the primary focus being the *efficiency* and *scalability* of both the online and batch algorithms. Working with sparse instead of dense matrices can lead to a significant speedup when dealing with a large amount of matrix multiplications – indeed, as noted by Rubenstein *et al*. the main computational saving when using sparse dictionaries is in the sparse coding step, the complexity of which is dominated by the projection of the signals onto the dictionary via a matrix multiplication. In the first experiment, we examine the performance of the proposed ℓ_1 kernel sparse coding approach for various dictionary sparsity levels, while in the second experiment we demonstrate how the online algorithm can be used to successfully learn a useful kernel dictionary on a set of over one-million signals.

Sparsity	Dense	90%	99%
Speed Up	$1\times$	$\sim 1\times$	$\mathbf{1.2}\times$
Error (test set)	0.2291	0.2289	**0.2259**

Fig. 1. Final reconstruction error using a fixed learning time.

A. Kernel sparse coding using kISTA

The first experiment uses the MNIST data set, consisting of $60,000$ images of handwritten digits. Using a subset of $10,000$ randomly selected images from the full collection we trained several dictionaries with sparsity values $\lambda_2 \in \{0.0001, 0.001, 0.005\}$ using a polynomial kernel and 800 atoms. For each such dictionary, we record the encoding time and reconstruction error using kISTA and report the results as a fraction of the time recorded when using a dense dictionary (i.e. $\lambda_2 = 0$). In Figure (VI) we show the final approximation error achieved for each dictionary, using a fixed learning time. We find that using a sparse dictionary leads to reduced time for both dictionary learning and encoding of new signals and lower reconstruction error.

B. Online Sparse Kernel Dictionary Learning

For this experiment we demonstrate how the online algorithm allows for kernel dictionaries to be learned from extremely large sets of signals, even those which cannot be contemporaneously held in memory. For training signals we used the MIDI Aligned Piano Sounds (MAPS) collection – a set of recordings of classical performances using several different pianos [16]. Each audio file was processed using an Equivalent Rectangular Bandwidth transform on 23-ms windows with a frequency resolution of 512 bins resulting in a collection of over 1.2×10^6 frames, a size which would normally be infeasible for kernel methods. The behaviour of the online algorithm is shown in Figure (VI), where we observe consistent reduction in the cost function.

VII. Conclusion

While potentially powerful, kernel methods introduce a series of practical hurdles which limit their use in medium to large scale problems. Inspired by developments in sparse Dictionary Learning, we have outlined a sparse model for learning with kernels which offers a noticeable improvement in efficiency with no impact on performance. The proposed formulation results in an optimization problem involving an ℓ_1 penalty over the dictionary and coding-coefficients, for which

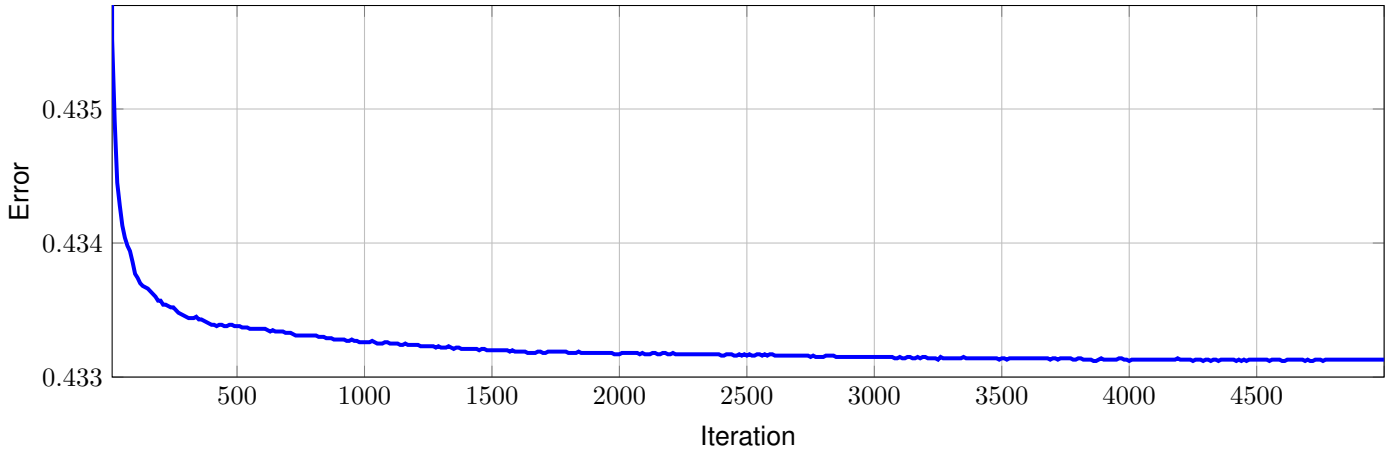

Fig. 2. Online Sparse Dictionary learning with Radial Basis Function kernel and mini-batch size of 64. Here we plot the reconstruction error on a separate set of $10,000$ signals as a function of the number of signals processed. In total, more that $300,000$ signals were used during dictionary learning.

we derive an efficient optimization scheme in both batch and online settings based on proximal splitting. Experiments on signals from two different domains validate the approach in terms of scalability. For future work, we plan to apply KDL to signal processing tasks such as Automatic Music Transcription and de-noising.

ACKNOWLEDGEMENT

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7-PEOPLE-2013-ITN) under grant agreement n° 607290 SpaRTaN.

REFERENCES

[1] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Advances in neural information processing systems*, 2006, pp. 801–808.

[2] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by v1?" *Vision research*, vol. 37, no. 23, pp. 3311–3325, 1997.

[3] H. Van Nguyen, V. M. Patel, N. M. Nasrabadi, and R. Chellappa, "Kernel dictionary learning," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 2021–2024.

[4] ——, "Design of non-linear kernel dictionaries for object recognition," *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 5123–5135, 2013.

[5] R. Rubinstein, M. Zibulevsky, and M. Elad, "Double sparsity: Learning sparse dictionaries for sparse signal approximation," *IEEE Transactions on signal processing*, vol. 58, no. 3, pp. 1553–1564, 2010.

[6] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image processing*, vol. 15, no. 12, pp. 3736–3745, 2006.

[7] J. Kivinen, A. J. Smola, and R. C. Williamson, "Online learning with kernels," *IEEE transactions on signal processing*, vol. 52, no. 8, pp. 2165–2176, 2004.

[8] Y. Engel, S. Mannor, and R. Meir, "The kernel recursive least-squares algorithm," *IEEE Transactions on signal processing*, vol. 52, no. 8, pp. 2275–2285, 2004.

[9] S.-J. Kim, "Online kernel dictionary learning," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2015, pp. 103–107.

[10] J. Sulam, B. Ophir, M. Zibulevsky, and M. Elad, "Trainlets: Dictionary learning in high dimensions," *IEEE Transactions on Signal Processing*, vol. 64, no. 12, pp. 3180–3193, 2016.

[11] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-point algorithms for inverse problems in science and engineering*. Springer, 2011, pp. 185–212.

[12] D. P. Bertsekas, "Incremental gradient, subgradient, and proximal methods for convex optimization: A survey," *Optimization for Machine Learning*, vol. 2010, pp. 1–38, 2011.

[13] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on pure and applied mathematics*, vol. 57, no. 11, pp. 1413–1457, 2004.

[14] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM journal on imaging sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[15] B. Schölkopf, R. Herbrich, and A. J. Smola, "A generalized representer theorem," in *International Conference on Computational Learning Theory*. Springer, 2001, pp. 416–426.

[16] V. Emiya, R. Badeau, and B. David, "Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1643–1654, 2010.