Learning Dictionaries as Sums of Kronecker Products
Cassio Fraga Dantas, Rémi Gribonval, Renato R Lopes, Michele N da Costa

To cite this version:
Cassio Fraga Dantas, Rémi Gribonval, Renato R Lopes, Michele N da Costa. Learning Dictionaries as Sums of Kronecker Products. SPARS 2017 - Signal Processing with Adaptive Sparse Structured Representations workshop, Jun 2017, Lisbon, Portugal. . hal-01514044

HAL Id: hal-01514044
https://inria.hal.science/hal-01514044
Submitted on 25 Apr 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Learning Dictionaries as Sums of Kronecker Products

Cássio Fraga Dantas, Rémi Gribonval
INRIA Rennes-Bretagne Atlantique, France

Renato R. Lopes, Michele N. da Costa
University of Campinas, Brazil

Abstract—The choice of an appropriate dictionary is a crucial step in the sparse representation of a given class of signals. Traditional dictionary learning techniques generally lead to unstructured dictionaries which are costly to deploy and do not scale well to higher dimensional signals. In order to overcome such limitation, we propose a learning algorithm that constrains the dictionary to be a sum of Kronecker products of smaller sub-dictionaries. A special case of the proposed structure is the widespread separable dictionary. This approach, named SuKro, is evaluated experimentally on an image denoising application.

I. INTRODUCTION

Dictionary learning algorithms [1] generally lead to unstructured over-complete dictionaries which are very costly to operate with, limiting their applicability to relatively low-dimensional problems. In order to obtain computationally efficient dictionaries, some of the most recent works in the field employ parametric models in the training process, which produce structured dictionaries [2]–[8]. Among the countless possibilities, a promising one is learning separable dictionaries [2], which can be represented as the Kronecker product of two sub-dictionaries, i.e. \( D = B \otimes C \).

We propose a broader structure case of which the separable structure is a special case. It consists of a sum of separable terms, where the number of components serves as a fine tuner for the complexity-adaptability tradeoff:

\[
D = \sum_{r} B^{(r)} \otimes C^{(r)}. \tag{1}
\]

II. PROPOSED TECHNIQUE

Consider a matrix \( D \in \mathbb{R}^{m_1 \times m_2} \) which is the Kronecker product of two sub-matrices \( B \in \mathbb{R}^{m_1 \times m_1} \) and \( C \in \mathbb{R}^{m_2 \times m_2} \). The rearrangement operator, defined in [9] and here denoted \( \mathcal{R}(\cdot) \), reorganizes the elements of \( D \) in such a way that the rearranged matrix \( \mathcal{R}(D) \in \mathbb{R}^{m_1 \times m_2} \) has rank one and can be written as an outer product of the vectorized versions of \( B \) and \( C \):

\[
\mathcal{R}(D) = \text{vec}(B) \text{vec}(C)^T. \tag{2}
\]

Now, let us consider a sum of \( \alpha \) Kronecker products

\[
D = \sum_{r=1}^{\alpha} B^{(r)} \otimes C^{(r)} = \sum_{r=1}^{\alpha} D^{(r)}. \tag{3}
\]

After rearrangement, we obtain a rank-\( \alpha \) matrix, since each term \( D^{(r)} \) leads to a rank-1 matrix as follows

\[
\mathcal{R}(D) = \sum_{r=1}^{\alpha} \mathcal{R}(D^{(r)}) = \sum_{r=1}^{\alpha} \text{vec}(B^{(r)}) \text{vec}(C^{(r)})^T. \tag{4}
\]

Therefore, by using (4), we can add a low-rank regularization term to the original dictionary learning optimization problem in order to learn a dictionary as a sum of \( \alpha \) Kronecker products:

\[
\min_{D,X} \frac{1}{2} \|Y - DX\|_F^2 + \lambda \text{rank}(\mathcal{R}(D)) \tag{5}
\]

subject to \( \|x_i\|_0 \leq t \), \( \forall j \| d_j \|_2 = 1 \)

where the parameter \( \lambda \in \mathbb{R}^+ \) controls the rank penalty.

III. OPTIMIZATION FRAMEWORK

As typically done in the literature [10]–[12], we solve the problem in (5) by alternately minimizing on the variables \( D \) and \( X \). On the sparse coding step (minimization with respect to \( X \)), we use the existing Orthogonal Matching Pursuit (OMP) algorithm [13]. On the dictionary update step, we use the nuclear norm (denoted \( \|\cdot\|_* \)) as a convex relaxation of the rank operator [14], yielding

\[
\text{Dictionary update: } \min_D \frac{1}{2} \|DX - Y\|_F^2 + \lambda \|\mathcal{R}(D)\|_*. \tag{6}
\]

The Alternating Direction Method of Multipliers (ADMM) [15] can be employed to solve such problem. It introduces an auxiliary variable \( \tilde{D} = \mathcal{R}(D) \) and a Lagrangian multiplier matrix \( Z \), as shown in Algorithm 1. The partial update with respect to \( \tilde{D} \) (second step in Alg. 1) is the proximal operator associated to the nuclear norm. It consists in the singular value soft-thresholding operation [16].

Algorithm 1 Dictionary Update - ADMM

\[
\begin{align*}
\text{Initialize } & D_0, Z_0, \text{ while } \|Z_{k+1} - Z_k\|_F^2 < \text{tol do} \\
& D_{k+1} = D_k - \gamma (D_kX - Y)(X^T + \mu(D_k - \mathcal{R}^{-1}(D_k - Z_k))) \\
& \tilde{D}_{k+1} = \text{prox}_{\mu \|\cdot\|} (\mathcal{R}(D_{k+1}) + Z_k) \\
& Z_{k+1} = Z_k - (\tilde{D}_{k+1} - \mathcal{R}(D_{k+1})) \\
\text{end while} \\
& \text{Normalize columns of } D
\end{align*}
\]

IV. EXPERIMENTS

We use a patch-based image denoising application to validate the proposed algorithm. The simulation set-up is the same as in [17] and is summarized in Table I. We compare our results to the unstructured K-SVD [11] dictionary, the SeDiL [2] separable dictionary, which uses a single Kronecker product and a different optimization strategy, and the ODCT analytic dictionary.

Figure 1 shows the denoised image PSNR as a function of the number of separable terms in the dictionary, which can be controlled by the parameter \( \lambda \) in (6). Note that even with very few separable terms, the results are close to the K-SVD. SuKro may even outperform the one-term SuKro dictionary outperforms SeDiL. Also note that the one-term SuKro dictionary outperforms SuKro. Naturally, as the number of separable terms increases, so does the denoising performance, since the structure becomes more flexible. The drawback is the increase in the dictionary application complexity.

Figure 2 illustrates the complexity-performance tradeoff. Besides providing very good denoising performance with quite reduced complexity costs, the proposed technique has the merit of providing a range of options on this tradeoff curve.

V. CONCLUSION

The proposed dictionary structure leads to fast operators while keeping a considerable degree of flexibility. Such tradeoff can be controlled through the number of terms in the summation. The image denoising simulations have shown very promising results, specially for higher noise scenarios. In such cases, the proposed structure manages to overcome an unstructured dictionary in terms of both computational complexity and denoising performance.
Fig. 2. Performance (PSNR) vs. Complexity, with noise standard deviation.

Fig. 1. PSNR vs. rank(D) (i.e. the number of separable terms). σ is the noise standard deviation.

TABLE I
SIMULATION PARAMETERS

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample dimension (n)</td>
<td>64</td>
</tr>
<tr>
<td>Number of atoms (m)</td>
<td>256</td>
</tr>
<tr>
<td>Training samples (N)</td>
<td>40000</td>
</tr>
<tr>
<td>Step-size (γ)</td>
<td>6 × 10⁻⁹</td>
</tr>
<tr>
<td>Lagrangian penalty (µ)</td>
<td>10⁷</td>
</tr>
<tr>
<td>Convergence tolerance (tol)</td>
<td></td>
</tr>
<tr>
<td>Alternate optimization iterations (Niter)</td>
<td>100</td>
</tr>
<tr>
<td>Dictionary initialization (D₀)</td>
<td>ODCT</td>
</tr>
<tr>
<td>Size of sub-matrices B(r) and C(r)</td>
<td>(8 × 16)</td>
</tr>
<tr>
<td>Size of R(D)</td>
<td>(128 × 128)</td>
</tr>
</tbody>
</table>

REFERENCES


