LEARNING DICTIONARY VIA SUBSPACE SEGMENTATION FOR SPARSE REPRESENTATION

Jianzhou Feng, Li Song, Xiaokang Yang, and Wenjun Zhang

Institute of Image Comm. & Information Proc., Shanghai Jiaotong University, 200240, Shanghai, China

ABSTRACT

Sparse signal representation based on redundant dictionaries contributed to much progress in image processing in the past decades. But the common overcomplete dictionary model is not well structured and there is still no guideline for selecting the proper dictionary size. In this paper, we propose a new algorithm for dictionary learning based on subspace segmentation. Our algorithm divides the training data into subspaces and constructs the dictionary by extracting the shared basis from multiple subspaces. The learned dictionary is well structured and its size is adaptive to the training data. We analyze this algorithm and demonstrate its ability on some initial supportive experiments using real image data.

Index Terms— Dictionary learning, subspace segmentation, K-subspaces, K-SVD.

1. INTRODUCTION

As a strong and reliable model, sparse representation over redundant dictionary has been used to handle most image processing applications, ranging from denoising, restoration and super-resolution to compression, detection, and separation [1, 2, 3, 4]. It assumes that any image patch $y \in \mathbb{R}^N$, can be accurately approximated by a linear combination of a few atoms $\{d_m\}_{m\in\Lambda}$ in an overcomplete dictionary $\mathcal{D} = \{d_m\}_{m\in\Gamma}$ with $|\Gamma| \geq N \gg |\Lambda|$. Based on this model, lots of algorithms have been proposed [5, 6, 7].

Though the sparse representation model is successful, current dictionary learning methods still have spaces to be optimized. The first point is that the common model is not well structured. The number of atom selection choice $\binom{|\Gamma|}{|\Lambda|}$ is exponentially large. This makes large amounts of non-image patches can also be sparsely represented and leads to unstable signal estimation in image restoration applications. The second point is how to select the proper dictionary size. Too large size means computation and memory wasting, while too small size means the dictionary can't exploit all the image features. Nowadays algorithms mostly set it manually according to the experiment results on a certain data set. This solution is reasonable for some cases, but may be intolerable in other specific applications.

Recently, putting structure in sparsity has shown its power in solving the first problem. Yu *et al.* proposed in [3] the structured sparse model selection based on a family of learned orthogonal bases. Dong *et al.* proposed in [4] the adaptive sparse domain selection using a series of learned compact sub-dictionaries. Their idea is similar but the learning process is different. As for the second problem, it is unsolved and the dictionary size is still set manually in their work.

In this paper, we propose a novel dictionary learning framework, under which a structured dictionary can be learned and its size is adaptive to different kind of training data. For a given training set, we divide the set into subspaces and construct the dictionary by extracting the shared basis from multiple subspaces. Any image patch in the test set is assumed to belong to one of the learned subspaces. Our algorithm is different from [3, 4] in three aspects: (1) We propose a novel subspace segmentation method based on K-subspace clustering. (2) The number of subspaces is adaptive to the training set. (3) An atom can be shared by multiple subspaces which makes the dictionary more compact.

The remainder of the paper is organized as follows. Section 2 describes the proposed dictionary learning framework. Section 3 presents initial supportive experiment results on real image data. Section 4 concludes the paper.

2. SPARSE REPRESENTATION VIA SUBSPACE SEGMENTATION

2.1. Structured representation model

As mentioned in Section 1, the sparse representation model lacks structure because of too many atom selection choices. Reducing the feasible set of selection results is necessary for a structured model. So in this paper, we assume the selection result Λ to be a subset of Γ_k, where Γ_k is one of the K subsets of the dictionary index set Γ with $|\Gamma| \geq N \gg |\Gamma_k|$. Under this assumption, we can obtain K spaces S_k, spanned by $\Phi_k = \{d_m\}_{m\in\Gamma_k}$, and the image space $\mathcal{I}$ can be approximated by $\cup_{k=1}^{N} S_k$. A toy example of $K = 2$ is shown in Fig. 1. In this example, $\mathcal{D}$ is composed of 5 atoms with $\Gamma = \{1, 2, 3, 4, 5\}$. Among these atoms, four belong to Φ_1 with $\Gamma_1 = \{1, 2, 3, 4\}$ and three belong to Φ_2 with $\Gamma_2 = \{1, 2, 5\}$. Suppose $|\Lambda| = 3$, the number of atom selection choice de-

crease dramatically from $\binom{5}{3} = 10$ for the sparse representation model to $\binom{4}{3} + \binom{3}{3} = 5$ for the structured representation model.

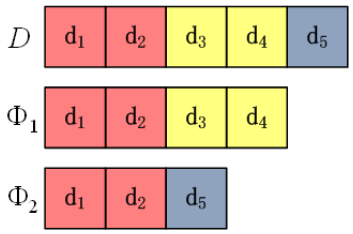

Fig. 1. Toy example

Based on this model, the dictionary learning algorithm contains two components: segmenting the imagery data to $\{S_k\}$ and constructing the dictionary using the basis Φ_k, which is obtained from S_k.

2.2. Subspace segmentation algorithm

Many clustering methods developed in statistics or machine learning can be used to solve the segmenting problem (e.g., expectation maximization, K-means, K-subspaces [8] and GPCA [9]). In this paper, we modify on K-subspaces to better fit the imagery data.

2.2.1. The K-subspaces clustering algorithm

To be clear, we recall the K-subspaces clustering algorithm in Fig. 2. It iteratively assign points to the nearest subspace (cluster assignment) and, for a given cluster, it computes a subspace that minimizes the sum of the squares of distance to all points of that cluster (cluster update).

1. **Initialization**
   Start with a random collection $\{S_1, \cdots, S_K\}$ of K subspaces of dimension d, where $S_k \subset \mathbb{R}^N$. Each subspace S_k is represented by one of its orthonormal basis, U_k (represented as a N-by-d matrix).
2. **Cluster Assignment**
   We define an operator $P_k = I_{N\times N} - U_k U_k^T$ for each subspace S_k. Each sample y_i is assigned a new label $L(y_i)$ such that
   $$L(y_i) = argmin_k \|P_k y_i\|_2^2 \quad (1)$$
3. **Cluster Update**
   Let S_k' be the set of samples labeled as k. We apply SVD (Singular Value Decomposition) to samples in S_k to update the basis U_k. Stop when $S_k' = S_k$ for all k. Otherwise, go to Step 2.

Fig. 2. The K-subspaces clustering algorithm.

2.2.2. The modified K-subspaces clustering algorithm

As shown in section 2.1, the structured representation model allows different subspace S_k to have its own dimension d_k, which is not assumed in the K-subspace clustering algorithms. This assumption fit the prior of imagery data better, because smooth patches are usually lie in a low dimension subspace while texture ones lie in subspaces with higher dimension. So we modify all the three steps of the K-subspaces clustering algorithm under this assumption.

1. Set $Y = \{y_i\}$, $\delta = \delta_0$, $k = 0$.
2. If $|Y| < S_{min}$, $K = k$ and stop. Else set $\delta = \delta + \tau$, until $max_{y_i \in Y}(|\Omega(y_i)|) \geq S_{min}$.
3. Set $i' = argmax_{y_i \in Y}(|\Omega(y_i)|)$ and $\Omega = \Omega(y_{i'})$.
4. Apply K-subspaces to Ω with $K = 2$ and $d = M$. The segmentation result is $\{\Omega_1, \Omega_2\}$.
5. Compute the sum of the squares of approximation error $\{E^2, E_1^2, E_2^2\}$ for $\{\Omega, \Omega_1, \Omega_2\}$. If $min(|\Omega_1|, |\Omega_2|) \geq S_{min}$ and $(E_1^2 + E_2^2) < \alpha E^2$,
   $$\Omega = \begin{cases} \Omega_1 & \frac{E_1^2}{|\Omega_1|} < \frac{E_2^2}{|\Omega_2|} \\ \Omega_2 & \text{otherwise} \end{cases} \quad (2)$$
   and go to Step 4.
6. Set $k = k + 1$.
7. Apply SVD to $\Omega = USV'$, set U_k as the first d_k column of U, where d_k is the minimum dimension that can make the mean of the squares of the approximation error below Δ^2.
8. Set $S_k = \{y_i \big| \|P_k y_i\|_2^2 < \Delta^2, y_i \in Y\}$.
9. Set $Y = Y - S_k$ and go to Step 2.

Fig. 3. The initialization step

Firstly, in the **initialization** step, we aim to determine the subspaces number K, the segmentation $\{S_1, \cdots, S_K\}$ and the subspaces dimensions $\{d_1, \cdots, d_K\}$ as close to the true ones as possible. Most of our modification is applied to this step because K-subspaces only converges to some local minimum, which makes the initialization result very important. For each subspace S_k, we assume there exists a neighborhood $\Omega(y_0) = \{y_i \big| \|y_i - y_0\| \leq \delta\}$, where y_0 is a sample and δ is a small number, $\Omega(y_0)$ is a subset of S_k and share the same basis with S_k. Such a neighborhood $\Omega(y_0)$ is likely to exist because experiment results show the effective dimension of neighborhoods are low in most cases [7] and similar samples should admit similar decompositions [2]. So the task of our initialization step is equal to finding all these neighborhoods, which is shown in Fig. 3. In the algorithm, Y contains all the unsegmented samples, its size decreases when a new subspace is segmented out. In each turn, we first search for the largest neighborhood $\Omega = \Omega(y_{i'})$ in Y. Its size is ensured to be no

less than S_{min}, the minimum size of subspaces, by enlarging the neighborhood radius δ. The original value of δ is δ_0 and the increase step is τ. Then we apply K-subspaces to Ω with $K = 2$ and $d = M$ to judge if Ω contains samples from more than one subspace, which is expressed in Step 4 and 5. The judgement mainly focus on if the approximation error drop obviously after segmentation. If so, we update Ω to be one of $\Omega_j, j \in \{1, 2\}$ and apply the above judging and updating process iteratively. We use $K = 2$ for simplicity. Once Ω is determined, for Ω and S_k share the same basis, d_k, U_k, S_k can easily be obtained by SVD, projection and grouping.

Secondly, in the **Cluster Assignment** step, we replace the label function by

$$L'(y_i) = argmin_k \|P_k y_i\|_2^2 + \lambda d_k. \tag{3}$$

The left part restricts the approximation error and the right part makes the representation as sparse as possible.

Finally, in the **Cluster Update** step, we update basis U_k to have d_k columns based on applying SVD to samples in S'_k.

2.3. Dictionary constructing

Given a segmentation $\{S_1, \cdots, S_K\}$ and its dimension set $\{d_1, \cdots, d_K\}$, constructing the dictionary $\mathcal{D}$ is equal to finding the basis shared by multiple subspaces. We still use the toy example shown in Fig.1 for a simple explanation. In it, we are aimed to divide $\mathcal{D}$ as $\mathcal{D} = (\mathcal{B}_1|\mathcal{B}_2|\mathcal{B}_3)$ with $\Phi_1 = (\mathcal{B}_1|\mathcal{B}_2)$ and $\Phi_2 = (\mathcal{B}_1|\mathcal{B}_3)$. The standard orthogonal basis Φ_k and U_k computed from S_k should span the same space. So we have

$$\begin{cases} (\mathcal{B}_1|\mathcal{B}_2) = \Phi_1 = Q_1 U_1 \\ (\mathcal{B}_1|\mathcal{B}_3) = \Phi_2 = Q_2 U_2 \end{cases}, \tag{4}$$

where Q_1 and Q_2 are two orthogonal matrices. Using matrix theory, we obtain $\mathcal{B}_1 = U_2 V_d$, where $U_1' U_2 = USV'$ is the SVD result and V_d is the d columns of V whose corresponding singular value is 1. In practice, we relax the singular value threshold from 1 to a little smaller parameter β. Using $\mathcal{B}_1, U_1$ and U_2, $\mathcal{B}_2$ and $\mathcal{B}_3$ can be easily obtained.

Our dictionary constructing algorithm extends the toy example. Firstly, we initialize the dictionary $\mathcal{D}^{(1)}$ as U_1. Then we update $\mathcal{D}$ according to subspaces S_k one after another. In the k^{th} turn, $\mathcal{D}^{(k-1)} = (\mathcal{B}_1|\cdots|\mathcal{B}_{n_{k-1}})$, where each group $\mathcal{B}_i$ is orthogonal and shared by different group of previous subspaces. Just like the toy example, we compute SVD for $\mathcal{B}_i' U_k$ and update $\mathcal{D}$ as

$$\begin{aligned} \mathcal{D}^{(k)} = (&\tilde{\mathcal{B}}_1|\cdots|\tilde{\mathcal{B}}_{n_{k-1}}| \\ &\mathcal{B}_1/\tilde{\mathcal{B}}_1|\cdots|\mathcal{B}_{n_{k-1}}/\tilde{\mathcal{B}}_{n_{k-1}}| \\ &U_k/\tilde{\mathcal{B}}_1/\cdots/\tilde{\mathcal{B}}_{n_{k-1}}) \end{aligned} \tag{5}$$

where $\tilde{\mathcal{B}}_i$ is the basis shared by S_k and the space spanned by $\mathcal{B}_i$. After all the subspaces have been processed, $\mathcal{D}$ is further updated using the codebook update stage of K-SVD [5], where each orthogonal basis $\mathcal{B}_i$ is treated as a whole.

2.4. Efficient implementation

A straight forward implementation of the method presented in the previous section is computationally demanding. We make our method more efficient using the following procedure.

- All the subspaces S_k are supposed to share the DC atom. So we remove the dc components of all the samples and then apply the presented algorithm. Adding this procedure can help to increase the distance between different subspaces so that the segmentation accuracy can be improved.
- There are usually massive amount of samples in Y, finding neighborhood for all the samples is computing costly. In the realization, we randomly select a subset of Y for initialization and assign all the samples in Y to the segmented subspaces. Initializing in this way may merge two similar subspaces together for lack of data. So we apply the same judging and updating process to each subspace as shown in Fig. 3 (step 4 and 5) with $\Omega = S_k$. The final value of $\Omega \subset S_k$ is segmented as a new subspace and $\Omega = S_k - \Omega$ is used for the next judging and updating. This process iterates until all the samples in S_k are assigned.
- During the iteration of cluster assignment and cluster update, if the samples in one of the subspace S_k is fewer than S_{min}, S_k will be eliminated and its samples are assigned to other subspaces in the next iteration. This can help to avoid overfitting.

3. EXPERIMENTAL RESULTS

The parameters used in all the experiments are: $N = 64$, $M = 10$, $S_{min} = 60$, $\delta_0 = 0$, $\tau = 10$, $\Delta = 5\sqrt{N}$, $\alpha = 0.9$, $\lambda = 75$ and $\beta = 0.95$.

3.1. Sparse representation

In this experiment, we choose 10000 patches of size 8×8 from a given 512×512 image. The subspace number K, the average number of atom used $L = \frac{\sum_{k=1}^{K} |S_k| d_k}{\sum_{k=1}^{K} |S_k|}$, the dictionary size n_K and the representation RMSE (root mean square error) ε are shown in Table 1. K and n_K are really adaptive to the training image and all the ε values are below 5 as we set $\Delta = 5\sqrt{N}$. All the parameters for *lena* is small for it is smooth and contains less texture patterns. As for *man*, it contains much more irregular texture patches which makes L being very large. Some dominant atoms learned from *barbara* is shown in Fig. 4 as an example.

3.2. Image denoising

For a given 512×512 image I and the noisy version $I' = I + \varepsilon$, where $\varepsilon_{ij} \sim N(0, \sigma^2)$ is white noise, we collect all the 8×8

Table 1. Dictionary learning results.

image	K	L	n_K	ε
lena	14	7.65	95	4.23
barbara	18	13.37	193	4.53
boat	21	13.32	205	4.86
peppers	18	7.87	147	4.33
man	14	24.22	211	4.87

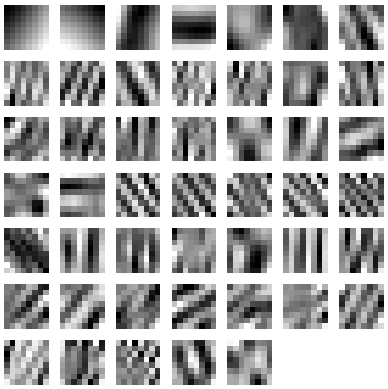

Fig. 4. Example of learnt atoms.

patches to be $\{y_i\}$ and $\{y_i'\}$. The denoising RMSE is defined as $\sqrt{\frac{\sum_i \|y_i - \hat{y}_i\|_2^2}{N|\{y_i\}|}}$, where $\hat{y}_i$ is the denoised result of y_i'. We randomly select 10000 patches from $\{y_i\}$ as the training data. Our denoise process contains two steps. The first is assigning y_i' to a subspace S_k according to equation (3) with $\lambda = 3\sigma^2$. The second is setting the projection of y_i' onto S_k as $\hat{y}_i$. In order to better fit the denoising application, a recalculating process is inserted into the learning algorithm. Each d_k is set to be $argmax_i(S(i,i) > \sigma)$ before dictionary constructing, where S is the singular matrix of $S_k = USV'$. We also apply K-SVD and SMSS on the same training set for comparison. For K-SVD, we set $\mathcal{D} \in \mathbb{R}^{64\times 256}$, sparsity $T_0 = 10$ and denoise y_i' as in [1]. For SSMS, we implement it ourselves using the training set for basis adaption according to [3]. The RMSE of the three algorithms on different images are listed in Table 2, which verify the superiority of our newly built algorithm. We should emphasize that this result is an initial support for the proposed method as the RMSE is computed directly from the image patches. Here we do not consider the overlap of patches, which is a non-trivial factor for the whole image denoising.

4. SUMMARY

In this paper, we proposed a new algorithm for dictionary learning. The learned dictionary is strongly structured with its size adaptive to the training data. Initial supportive experiments showed its superiority and potential in image processing related applications.

Table 2. Denoising RMSE under different σ, the column from left to right represents K-SVD [1], SSMS [3] and the proposed algorithm respectively. The smallest one is bolded.

image	$\sigma = 10$			$\sigma = 20$		
lena	6.20	6.50	**5.60**	9.18	9.63	**8.30**
barbara	7.48	7.84	**6.59**	11.52	11.93	**10.22**
boat	7.55	7.94	**6.93**	11.18	11.67	**10.31**
peppers	6.52	6.76	**6.03**	9.22	9.72	**8.33**
man	8.36	9.03	**8.10**	12.40	13.11	**12.01**

Acknowledgement

This work was supported in part by NSFC (60702044, 60902020, 60902073), 973 Program (2010CB731401, 2010CB731406), the 111 project and STCSM (10511500802).

References

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

[2] J. Mairalo, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *Proc. ICCV*, 2009.

[3] G. Yu, G. Sapiro, and S. Mallat, "Image modeling and enhancement via structured sparse model selection," in *Proc. ICIP*, 2010.

[4] W.S. Dong, L. Zhang, G. Shi, and X. Wu, "Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization," *IEEE Trans. on Image Processing to appear*.

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

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

[7] M. Elad, M. A. T. Figueiredo, and Y. Ma, "On the role of sparse and redundant representations in image processing," *Proceedings of the IEEE*, pp. 972–982, 2010.

[8] J. Ho, M. H. Yang, J. Lim, and D.Kriegman, "Clustering appearances of objects under varying illumination conditions," in *Proc. CVPR*, 2003.

[9] R. Vidal, Y.Ma, and J. piazzi, "A new gpca algorithm for clustering subspaces by fitting, differentiating and dividing polynomials," in *Proc. CVPR*, 2004.