LA-UR- 10-00412

Approved for public release; distribution is unlimited.

Title: DICTIONARY CONSTRUCTION IN SPARSE METHODS FOR IMAGE RESTORATION

Author(s): Brendt Wohlberg

Intended for: Proceedings - IEEE International Conference for Image Processing

DICTIONARY CONSTRUCTION IN SPARSE METHODS FOR IMAGE RESTORATION

*Brendt Wohlberg**

Los Alamos National Laboratory
Los Alamos, NM 87545, USA

ABSTRACT

Index Terms— Image restoration, sparse optimisation, basis pursuit denoising

1. INTRODUCTION

Sparsity-based methods [1] have achieved very good performance in a wide variety of image restoration problems, including denoising [2], inpainting [3, 4], super-resolution [5], and source separation [6]. These methods are based on the assumption that the image to be reconstructed may be represented as a superposition of a few known components, and the appropriate linear combination of components is estimated by solving an optimisation such as Basis Pursuit De-Noising (BPDN) [7]

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\|\Phi\boldsymbol{\alpha} - \mathbf{s}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1, \qquad (1)$$

or the constrained problem

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_1 \text{ such that } \|\Phi\boldsymbol{\alpha} - \mathbf{s}\|_2 \leq \sigma, \qquad (2)$$

where Φ is an overcomplete *dictionary*, $\boldsymbol{\alpha}$ is the sparse representation, $\mathbf{s}$ is the vector to be reconstructed, and λ or σ are user-determined parameters. Given the resulting representation $\boldsymbol{\alpha}$, the reconstructed image is $\Phi\boldsymbol{\alpha}$. When the dictionary is known analytically and the corresponding matrix-vector product may be computed by a fast transform (e.g. curvelets and the Discrete Cosine Transform [6]), it is often possible to apply this framework to the image as a whole. More recently, however, there has been growing interest in dictionaries derived from actual image data, in which case no fast transform is available, and it is, for practical reasons, necessary to apply the sparse optimisation to individual image blocks.

When deriving the dictionary from image data, in the hope of obtaining dictionaries more representative of the specific image data of interest, the construction of the dictionary plays a critical role. Currently, the leading approaches are to either construct a very large dictionary using the blocks in a set of training images (e.g. [5]), or to construct a compact dictionary representing the variation within the full training set (or samples from the image to be restored) via an algorithm such as the K-SVD [2], which can be considered a generalisation of the k-means algorithm for Vector Quantization.

As an alternative, a new approach is also considered here, consisting of a dictionary tuned to each block to be reconstructed by searching the full set of available blocks for nearest neighbours. (This new approach may either be considered as a new method for constructing a dictionary for standard BPDN, or as a hybrid of BPDN and Matching Pursuit [1].)

2. EXPERIMENTS

The performance of the different approaches to dictionary construction were compared in a number of computational experiments for the denoising problem. In order for the results not to depend on the specific method for combining individual block estimates into an estimate of the full image to be denoised, performance was calculated using the mean reconstruction error over a randomly selected set of 2000 blocks from the test images. The comparisons are made for three different noise levels (additive Gaussian white noise scaled to give SNR levels of 5dB, 10dB, and 20dB), two different block sizes (5×5 and 8×8), and six different dictionary sizes (with number of components selected to be $1\times$, $2\times$, $4\times$, $8\times$, $16\times$, and $32\times$ the relevant block size, and denoted D1, D2, D4, D8, D16, and D32 respectively). When constructing a dictionary, a distinction is made between testing data (noisy blocks selected from the test image to be denoised itself), and training data (a distinct, nominally noise-free set of images from which blocks are extracted). The following dictionary construction methods are compared:

NN train For each block to be denoised, a dictionary with N components is populated by the N nearest neighbours of that block, determined using angular distance, in the noise-free training data.

NN train (nrm) The dictionary construction follows that for "NN train", but each component is normalised to have unit magnitude.

*This research was supported by the U.S. Department of Energy through the LANL/LDRD Program.

KSVD train The K-SVD is applied to the noise-free training data to construct a dictionary of the desired size.

NN test For each block to be denoised, a dictionary with N components is populated by the N nearest neighbours of that block, determined using angular distance, in the noisy testing data. (To avoid very poor denoising performance, it is vital that the block being denoised is itself is omitted from this dictionary.) Each dictionary component is normalised to have unit magnitude.

KSVD test The K-SVD is applied to the noisy testing data to construct a dictionary of the desired size.

The computational experiments described here were performed in Matlab, using publicly available codes where possible. The unconstrained minimum ℓ^1 problem (1) was solved using `l1_ls` [8]; in comparisons with a number of other publicly available codes for the same problem, this was found to give the most reliable convergence over a wide range of parameters, which was critical for these experiments where hand-tuning of parameters for each individual test case was not possible. The constrained minimum ℓ^1 problem (2) was solved using `spgl1` [9, 10], and the K-SVD was computed using `KSVD-Box`[2, 11]. It should be emphasised that the K-SVD algorithm has a number of free parameters, and while every effort was made to choose those giving the best performance in this application, it is possible that better choices are possible.

Two distinct data sets were used for these experiments:

Standard The test image consisted of the well-known "Cameraman" image, and the training image set consisted of the "Boat", "House", "Lena", "Peppers", and "Barbara" standard images included with the K-SVD implementation [11] as training data. Since the training image do not contain similar content to the test image, this is a more challenging data set for dictionaries constructed from the training data.

Face The test image consisted of a fixed image randomly selected from the AT&T Face Database [12], and the training image set consisted of 40 other images randomly selected (but excluding all images of the subject used for the test image) from the same database. Given the tightly-controlled nature of the data set, with both the training and testing images consisting of similar scenes, this is an ideal data set for dictionaries constructed from the training data.

3. RESULTS

In the first set of experiments, all test blocks are denoised using the unconstrained optimisation (1) over each of a set of fixed λ values. For the 10dB noise level and 8×8 blocks, the dependence of the mean reconstruction errors on λ are displayed in Figure 1 for the "Standard" data set, and in Figure 2 for the "Face" data set. (Since the best performance of "NN train" was found to be at much larger values of λ than the other methods, it was omitted from this set of experiments for practical reasons.) In both of these figures, to avoid clutter, only the best-performing dictionary size is plotted for each different dictionary construction method. The performance varation with dictionary size for the "Standard" data set is plotted for "KSVD test" in Figure 3 and for "NN train (nrm)" in Figure 4. The optimum reconstruction errors for each method, for all block sizes and noise levels, are presented in Tables 1 and 2. Note that, at each noise level, the best mean reconstruction error is provided by the "NN train (nrm)" dictionary construction method.

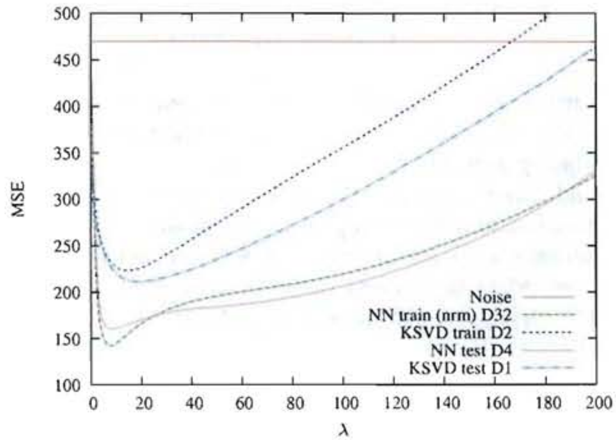

Fig. 1. Average reconstruction error for BPDN with various 8×8 block dictionaries constructed on the "Standard" data set, and tested at a noise level of 10dB.

SNR	5dB		10dB		20dB	
Blocks	5×5	8×8	5×5	8×8	5×5	8×8
Noise	1485	1486	470	470	47	47
NN train (n)	540	**319**	215	**143**	31	**27**
KSVD train	547	444	248	223	43	41
NN test	675	407	233	161	31	28
KSVD test	643	492	239	212	41	39

Table 1. Minimum (over λ) average reconstruction errors on the "Standard" data set.

In the second set of experiments, the known Mean Square Error (MSE) of each test image was used to set the value of σ in the constrained optimisation (2), and the resulting mean reconstruction errors are presented in Tables 3 and 4. At each noise level, the best mean reconstruction error is provided by the "NN train" dictionary construction method. (It is interesting to note that the dictionary normalisation of "NN train (nrm)" substantially reduces performance in this case.) The

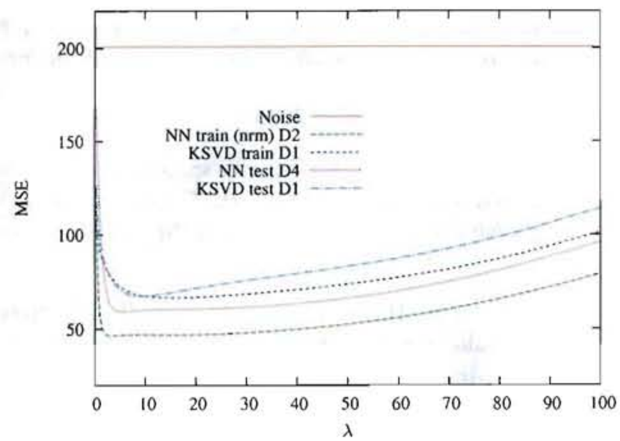

Fig. 2. Average reconstruction error for BPDN with various 8×8 block dictionaries constructed on the "Face" data set, and tested at a noise level of 10dB.

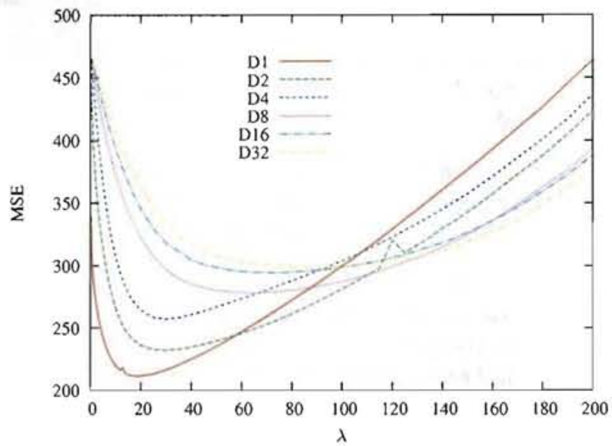

Fig. 3. Average reconstruction error for BPDN with various size KSVD dictionaries (8×8 blocks) constructed from "Standard" test data with SNR of 10dB.

SNR	5dB		10dB		20dB	
Blocks	5×5	8×8	5×5	8×8	5×5	8×8
Noise	634	635	201	201	20	20
NN train (n)	228	**111**	81	**46**	12	**10**
KSVD train	218	155	86	67	15	14
NN test	252	152	86	59	12	**10**
KSVD test	247	177	94	68	15	13

Table 2. Minimum (over λ) average reconstruction errors on the "Face" data set.

reconstruction error variation with dictionary size of the "NN train" method is displayed in Figure 5; note that the position of the minimum varies, but in all cases, the performance falls

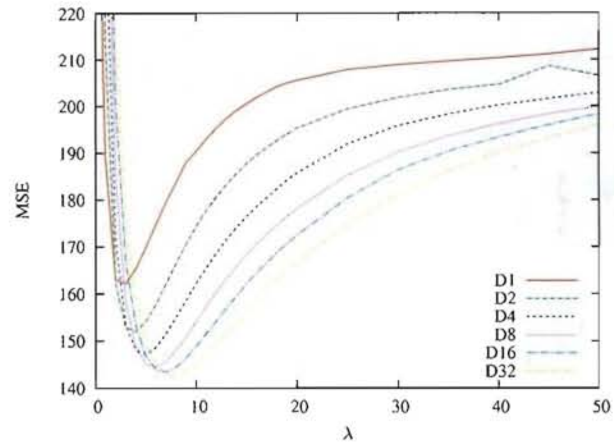

Fig. 4. Average reconstruction error for BPDN with various size NN train (normalised) dictionaries (8×8 blocks) constructed from "Standard" test data with SNR of 10dB.

off beyond some dictionary size.

SNR	5dB		10dB		20dB	
Blocks	5×5	8×8	5×5	8×8	5×5	8×8
Noise	1485	1483	470	469	47	47
NN train	549	**350**	213	**153**	39	**31**
NN train (n)	1081	636	396	242	52	38
KSVD train	714	591	291	257	58	50
NN test	1279	804	442	276	50	38
KSVD test	855	581	279	224	49	42
PCA proj.	436	369	170	162	32	39

Table 3. Average reconstruction errors on the "Standard" data set assuming known noise magnitude.

SNR	5dB		10dB		20dB	
Blocks	5×5	8×8	5×5	8×8	5×5	8×8
Noise	634	630	201	199	20	20
NN train	211	**134**	80	**55**	14	**12**
NN train (n)	463	261	163	86	19	13
KSVD train	293	206	111	81	18	14
NN test	479	298	166	102	18	13
KSVD test	340	250	124	89	17	15
PCA proj	156	148	67	72	14	14

Table 4. Average reconstruction errors on the "Face" data set assuming known noise magnitude.

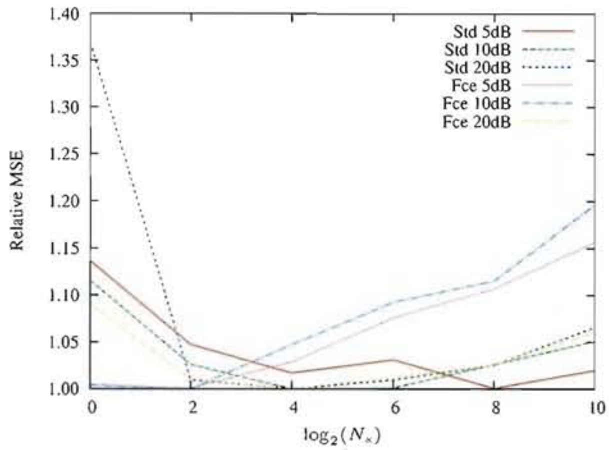

Fig. 5. Variation of relative reconstruction errors (each MSE curve is divided by its minimum to allow comparison on the same graph) with dictionary size factor (actual dictionary size is $64N_s$) for the "NN train" method with 8×8 blocks.

4. CONCLUSIONS

Considering that the K-SVD constructs a dictionary which has been optimised for mean performance over a training set, it is not too surprising that better performance can be achieved by selecting a custom dictionary for each individual block to be reconstructed. The nearest neighbour dictionary construction can be understood geometrically as a method for estimating the local projection into the manifold of image blocks [13], whereas the K-SVD dictionary makes more sense within a source-coding framework (it is presented as a generalisation of the k-means algorithm for constructing a VQ codebook), is therefore, it could be argued, less appropriate in principle, for reconstruction problems. One can, of course, motivate the use of the K-SVD in reconstruction application on practical grounds, avoiding the computational expense of constructing a different dictionary for each block to be denoised.

Since the performance of the nearest neighbour dictionary decreases when the dictionary becomes sufficiently large, this method is also superior to the approach of utilising the entire training set as a dictionary (and this can also be understood within the image block manifold model). In practical terms, the tradeoff is between the computational cost of a nearest neighbour search (which can be achieved very efficiently), or of increased cost at the sparse optimisation.

5. REFERENCES

[1] A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM Review*, vol. 51, no. 1, pp. 34–81, 2009.

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

[3] B. Wohlberg, "Inpainting with sparse linear combinations of exemplars," in *Proc. IEEE Int. Conf. Acoustics Speech Sig. Proc.*, Taipei, Taiwan, Apr. 2009, pp. 689–692.

[4] B. Shen, W. Hu, Y. Zhang, and Y.-J. Zhang, "Image inpainting via sparse representation," in *Proc. IEEE Int. Conf. Acoustics Speech Sig. Proc.*, April 2009, pp. 697–700.

[5] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution as sparse representation of raw image patches," in *Proc. IEEE Int. Conf. Comput. Vision Patt. Recogn.*, June 2008, pp. 1–8.

[6] Jean-Luc Starck, Michael Elad, and David L. Donoho, "Image decomposition via the combination of sparse representation and a variational approach," *IEEE Trans. Image Process.*, vol. 14, no. 10, pp. 1570–1582, Oct. 2005.

[7] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33–61, 1998.

[8] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An interior-point method for large-scale l1-regularized least squares," *IEEE J. Sel. Top. Sign. Proces.*, vol. 1, no. 4, pp. 606–617, 2007.

[9] E. van den Berg and M. P. Friedlander, "SPGL1: A solver for large-scale sparse reconstruction," June 2007, `http://www.cs.ubc.ca/labs/scl/spgl1`.

[10] E. van den Berg and M. P. Friedlander, "Probing the pareto frontier for basis pursuit solutions," *SIAM J. Sci. Comput.*, vol. 31, no. 2, pp. 890–912, 2008.

[11] R. Rubinstein, "KSVDBox v10," `http://www.cs.technion.ac.il/~ronrubin/Software/ksvdbox10.zip`, May 2009.

[12] "AT&T face database," `http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html`, 2002.

[13] G. Peyré, "Manifold models for signals and images," *Comput. Vision Image Understanding*, vol. 113, no. 2, pp. 249–260, 2009.