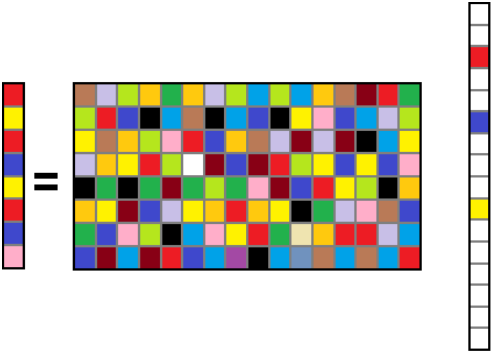
The Effect of Atom Replacement Strategies on Dictionary Learning

Paul Irofti

Politehnica University of Bucharest
Faculty of Automatic Control and Computers
Department of Automatic Control and Systems Engineering
Email: paul@irofti.net

International Traveling Workshop on Interactions Between Sparse Models and Technology
August 25th, 2016

Sparse Representation (SR)



$$y \approx Dx$$

The Dictionary Learning (DL) Problem

Given a data set $Y \in \mathbb{R}^{p \times m}$ and a sparsity level s, minimize the bivariate function

$$\begin{aligned} \underset{D,X}{\text{minimize}} \quad & \phi(D,X) = \|Y - DX\|_F^2 \\ \text{subject to} \quad & \|d_j\|_2 = 1,\ 1 \le j \le n \\ & \|x_i\|_0 \le s,\ 1 \le i \le m, \end{aligned} \tag{1}$$

where $D \in \mathbb{R}^{p \times n}$ is the dictionary (whose columns are called atoms) and $X \in \mathbb{R}^{n \times m}$ the sparse representations matrix.

Approach

Algorithm 1: Dictionary learning – general structure

```
1 Arguments: signal matrix $Y$, target sparsity $s$
2 Initialize: dictionary $D$ (with normalized atoms)
3 for $k = 1, 2, \ldots$ do
4     With fixed $D$, compute sparse representations $X$
5     With fixed $X$, update atoms $d_j$, $j = 1 : n$
```

The sparsity pattern provided by step 4 tells us:

- how many times each dictionary atom is used
- popular atoms help with classification and compression
- what about the ones that are rarely or never used?

Goal: Investigate strategies for replacing unused atoms.

DL Algorithms

K-SVD[1] solves the optimization problem in sequence

$$\min_{d_j, X_{j,\mathcal{I}_j}} \left\| \left(Y_{\mathcal{I}_j} - \sum_{\ell \neq j} d_\ell X_{\ell,\mathcal{I}_\ell} \right) - d_j X_{j,\mathcal{I}_j} \right\|_F^2 \qquad (2)$$

where all atoms excepting d_j are fixed.

This is seen as a rank-1 approximation and the solution is given by the singular vectors corresponding to the largest singular value.

- AK-SVD[2] uses one iteration of the power method
- SGK[3] refines only the atom d_j
- NSGK[4] uses differences with the previous values of D and X

[1] Aharon, Elad, and Bruckstein 2006.

[2] Rubinstein, Zibulevsky, and Elad 2008.

[3] Sahoo and Makur 2013.

[4] Sadeghi, Babaie-Zadeh, and Jutten 2013.

Replacement Strategies

Comments:

- the dictionary update sequence allows unused atoms to be replaced as soon as we encounter them
- numerical evidence shows that once an atom stops being used it never gets picked up again

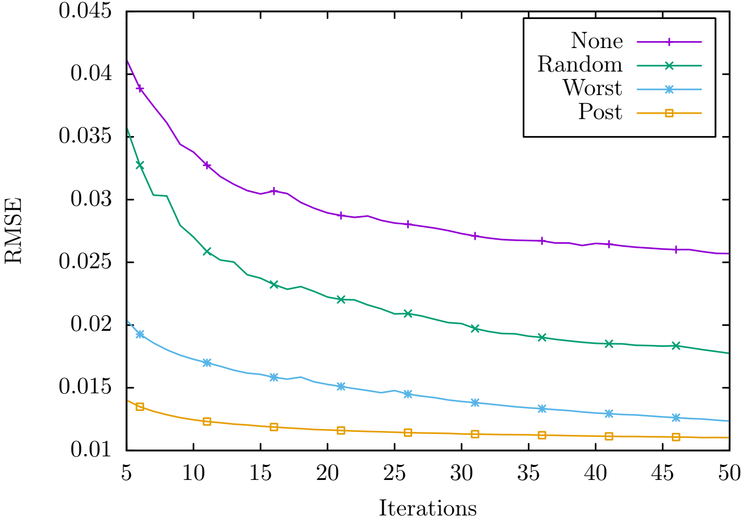
Strategies:

- given that we start with a random dictionary, substitute with a new randomly generated column ('Random')
- find the worst represented signal and make it a part of the dictionary ('Worst')
- perform bulk replacement using worst represented signals at the end of the dictionary refinement stage ('Post')

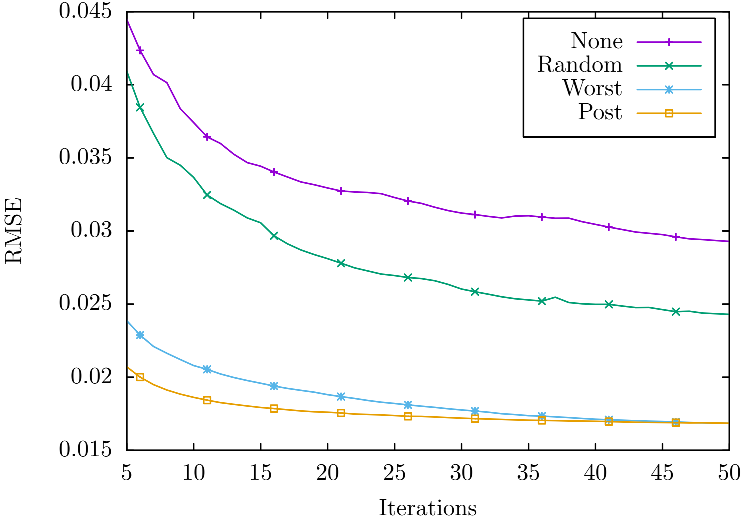
Final RMSE with various replacement strategies

n	Method	*Replacement*			
		None	Random	Worst	Post
128	K-SVD	0.029406	0.026723	**0.019096**	0.019355
	AK-SVD	0.029497	0.026876	**0.019134**	0.019369
	SGK	0.029402	0.026800	**0.019079**	0.019612
	NSGK	0.025004	0.024707	**0.020558**	0.019804
256	K-SVD	0.029146	0.023979	0.016773	**0.016730**
	AK-SVD	0.029291	0.024300	**0.016843**	0.016858
	SGK	0.029836	0.024201	0.016928	**0.016923**
	NSGK	0.022099	0.022590	0.020913	**0.017563**
512	K-SVD	0.025699	0.017744	0.012343	**0.011034**
	AK-SVD	0.025318	0.018016	0.012436	**0.011078**
	SGK	0.025617	0.017668	0.012579	**0.011072**
	NSGK	0.016795	0.016769	0.017951	**0.012128**

Error descent averaged over 10 runs for K-SVD ($n = 512$)



Error descent averaged over 10 runs for AK-SVD ($n = 256$)



Conclusions

Results:

- substitution has significant impact on approximation error
- experiments suggest that replacing unused atoms with the worst represented signals is the best approach
- inspired new research: B. Dumitrescu and **P. Irofti** (2016). "Low Dimensional Subspace Finding via Size-Reducing Dictionary Learning". In: *2016 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6

Size-Reducing Dictionary Learning

Inspiration:

- starting from "numerical evidence shows that once an atom stops being used it never gets picked up again"
- lead to "we can just remove the atom and shrink the dictionary"
- but how much can we squeeze the dictionary minimizing the SR complexity without representation quality loss?
- can we apply this to low-rank representation (LRR) through subspace tracking?

Size-Reducing Dictionary Learning

Atoms can be reduced indirectly by acting on the representations.

We add a penalty term to Equation (1)

$$\phi(D, X) = \frac{1}{2}\|Y - DX\|_F^2 + \frac{\mu_1}{2}\|X\|_F^2 + \mu_2 \sum_{i=1}^{n} \|x_i^T\|_2, \qquad (3)$$

where x_i^T is the i-the row of X.

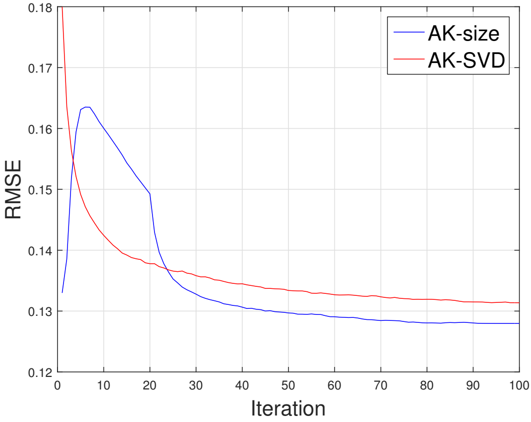
The first penalty term is a regularization[5] meant to discourage almost linearly dependent atoms.

The second penalty term is a group sparsity inducer: if the entire row i of X is zero, then atom d_i is unused and can be removed.

[5] Dai, Xu, and Wang 2012.

DL size-reducing AK-SVD

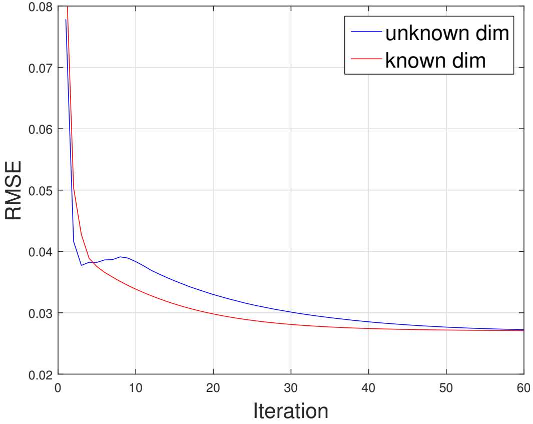
Size-reducing AK-SVD can be used not only for finding an economical number of atoms, but also as an initialization method.



After 20 iterations (100 → 54 atoms) switch to plain AK-SVD.

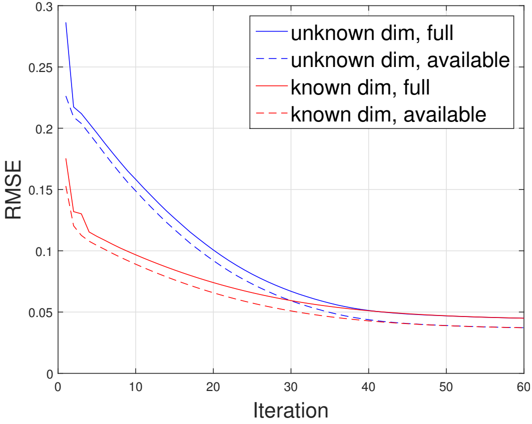
Low-dimensional subspace finding, full data

- $p = 50$, $m = 2000$ and true dimension $n = 4$
- start with 100 atoms, run with $s = 8$ and $s = 4$
- dictionary reduced to 4 atoms in less than 10 iterations
- RMSE: 0.271 ($s = 8$), 0.270 ($s = 4$), 0.268 (SVD)



Low-dimensional subspace finding, incomplete data

- same dimensions, missing ratio $\rho = 0.5$
- true subspace dimension is reached in about 5 iterations
- dashed lines: error on the available data only



Thank You!

Questions?