Fast Online Convolutional Sparse Coding in Spatial Domain

Jinhui Xiong

3 April 2018

1 Motivation

Convolutional Sparse Coding learns translational invariant image features and it has been solved by modern approaches in Fourier domain owing to the improved computing efficiency. Recent algorithms tackles the problem in **batch mode**, where all training images are involved in each iteration, and these approaches are expensive in both space and time, lacking of scalability on large datasets or streaming data, like videos. **Online learning** is a promising strategy alleviating the above issues arisen in batch approaches, which takes account of a single or a portion of the whole data in each training step, and incrementally update model variables. Herein, the required memory and computing sources is only dependent on the sample size in every observation, independent of training data size.

Exploiting Parseval's theorem to express convolution operator by multiplication in fourier domain simplifies heavy matrix inversion operation to cheap point-wise division, thus it shows tremendous improvements over spatial solvers with respect to running time. Little work has been conducted about looking into a faster solver in spatial domain. Solving CSC problem in spatial domain gets the benefits of getting rid of the periodic boundary condition assumption hold in fourier domain. In this work, we implemented an efficient online CSC spatial solvers, and it runs faster than recently published online CSC frequency solver.

2 Methodology

The image formation model for sparse image representation can be expressed as follows:

$$x \approx d_1 * z_1 + d_2 * z_2 + \ldots + d_K * z_K \tag{1}$$

where $x \in \mathbb{R}^{n \times n}$ is a $n \times n$ image, $d_k \in \mathbb{R}^{m \times m}$ is the k-th dictionary and z_k is the sparse code associated with k-th dictionary, and it has the same dimension as x.

Specifically, the CSC problem is trying to minimize

$$\frac{1}{2}\|x - \sum_{k=1}^{K} d_k * z_k\|_2^2 + \lambda \sum_{k=1}^{K} \|z_k\|_1, s.t. \|d_k\|_2^2 \leqslant 1 \tag{2}$$

Issues regarding to the model and the optimization problem:

- far more variables to represent a single image (K times more)
- very sparse distributed feature map (more than 99.5% of the entries are nearly 0)
- Solving d in fourier domain is not efficient
- non-convex optimization problem

In iterative solver, it doesn't converge to the local optimal position in each sub-iteration to avoid over-fitting and getting stuck in some bad local minimum.

2.1 Stochastic Batch CSC (SBCSC)

The key idea of the proposed algorithm comes from the observation that the images can be well reconstructed by a random subset of the sparse code. The objective function of the modified problem can be written as:

$$\frac{1}{2}\|x-(DM^T)(Mz)\|_2^2+\lambda\|Mz\|_1, s.t. \|d_k\|_2^2 \leqslant 1 \tag{3}$$

where M is the randomized sub-sampling matrix.

Algorithm 1 Stochastic Batch Convolutional Sparse Coding

while not converge **do**
  $D \leftarrow constructD(d_{1,2,\ldots,K})$
  for i=1 to numImages **do**
    draw(x_i, M_i);
    $z_i \leftarrow \text{argmin } \frac{1}{2}\|x_i-(DM_i^T)(M_i z)\|_2^2+\lambda\|M_i z\|_1$
  end for
  $Z \leftarrow constructZ(z_{1,2,\ldots K})$
  $d \leftarrow \text{argmin } \frac{1}{2}\|x-Zd\|_2^2+\lambda\|z\|_1$
end while

2.2 Stochastic Online CSC (SOCSC)

We can further tackle the proposed problem in online learning fashion for a gain of scalability.

$$\begin{aligned} z_t = & \quad argmin\ \frac{1}{2}\|x-(D_{t-1}M^T)(Mz)\|_2^2+\lambda\|Mz\|_1 \\ d_t = & \quad argmin\ \frac{1}{2}\|x-Z_t d\|_2^2 \end{aligned} \tag{4}$$

Algorithm 2 Stochastic Online Convolutional Sparse Coding

while not converge **do**
  $D \leftarrow constructD(d_{1,2,\ldots,K})$
  draw(x_t, M_t);
  $z_t \leftarrow \text{argmin } \frac{1}{2}\|x_t-(DM_t^T)(M_t z)\|_2^2+\lambda\|M_t z\|_1$
  $Z \leftarrow constructZ(z_t)$
  $B_t \leftarrow \frac{t-1}{t}B_{t-1}+\frac{1}{t}Z^T x_t$
  $C_t \leftarrow \frac{t-1}{t}C_{t-1}+\frac{1}{t}Z^T Z$
  $d \leftarrow \text{argmin } \frac{1}{2}\|C-Bd\|_2^2$
end while

3 Result

The algorithms were tested on city dataset with 10 training images and 4 test images, and each image has the size of 100×100. The dictionary size is $11 \times 11 \times 100$. They were compared with FFCSC (felix's implementation), OCSC (recent published online learning implementation solved in fourier domain).

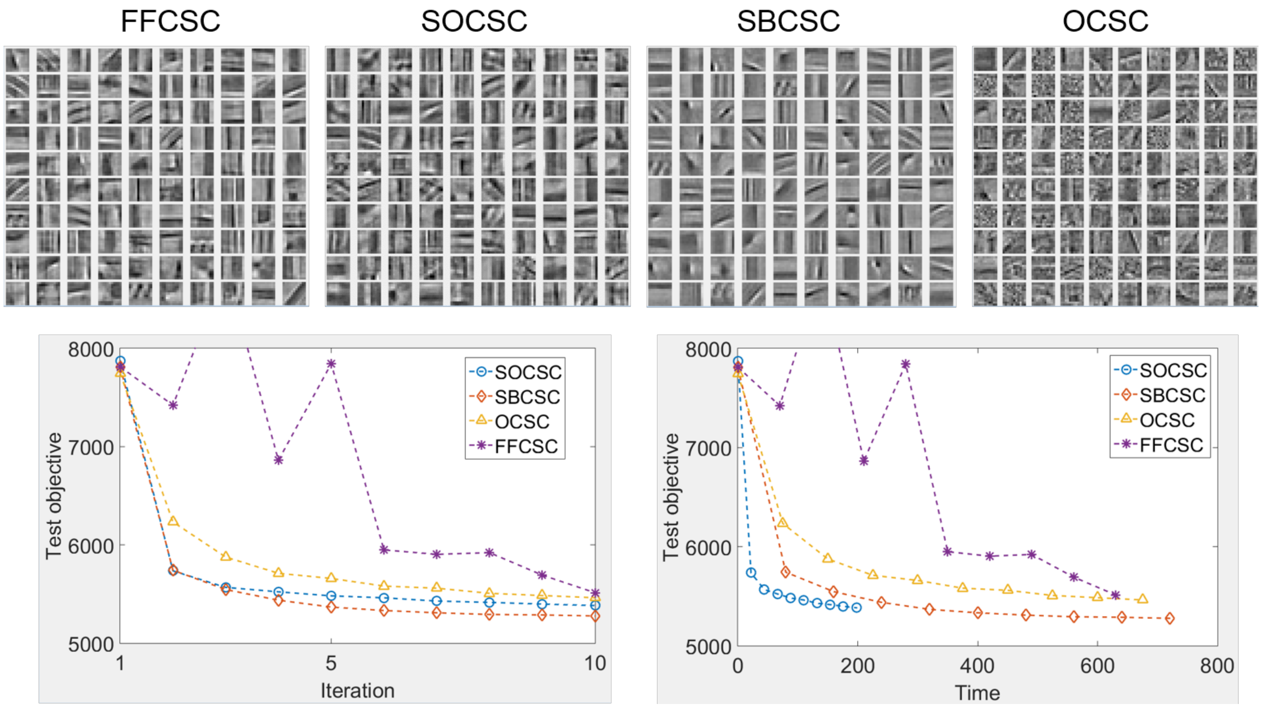


Figure 1:

In SOCSC implementation, we chose skipping factor of 0.9, indicating the random subsets contains 10% of the whole codes.The averaged time cost for each iteration is 20s, computing code takes half of it, computing dictionary and the other operations takes half of the remaining time, respectively.

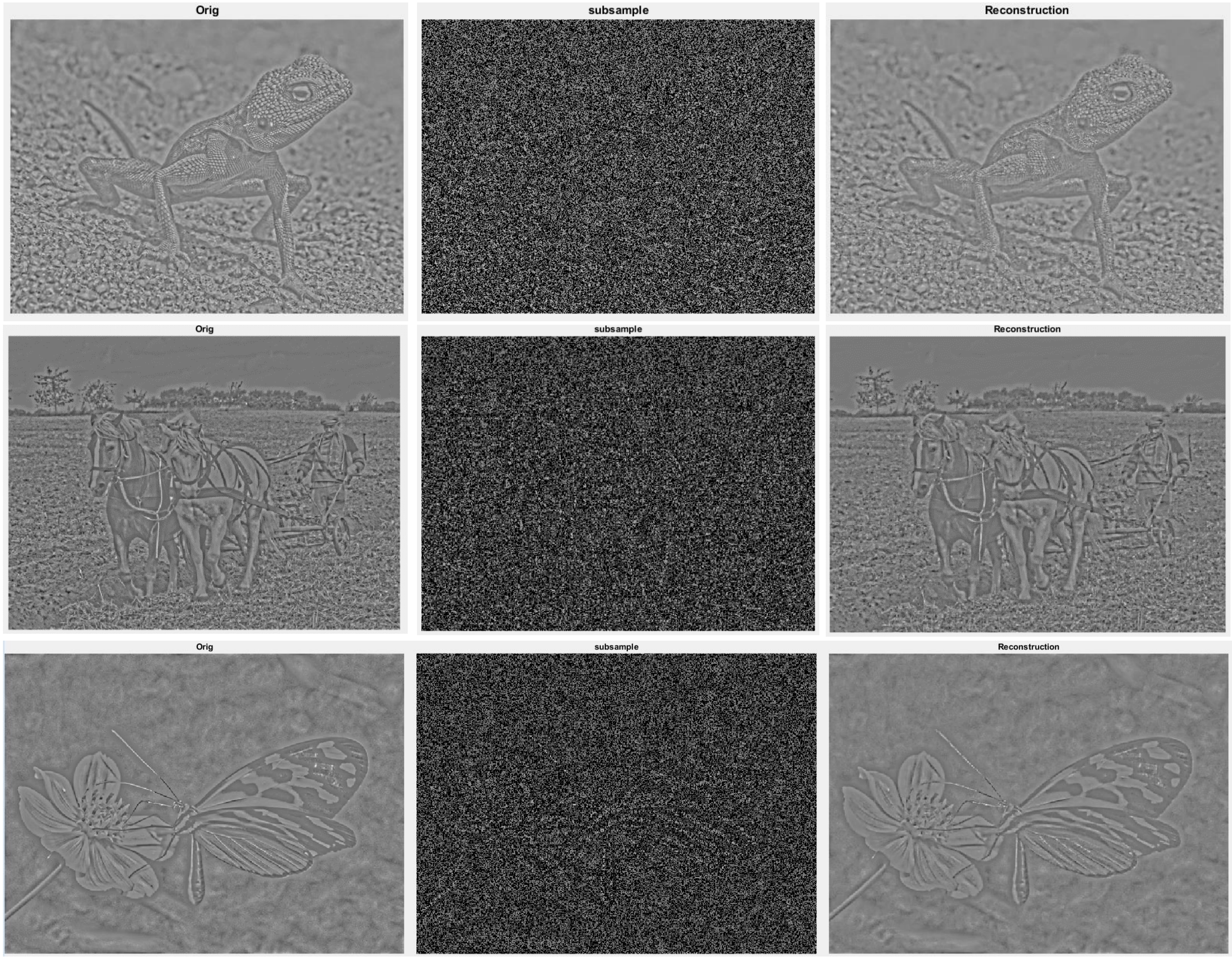
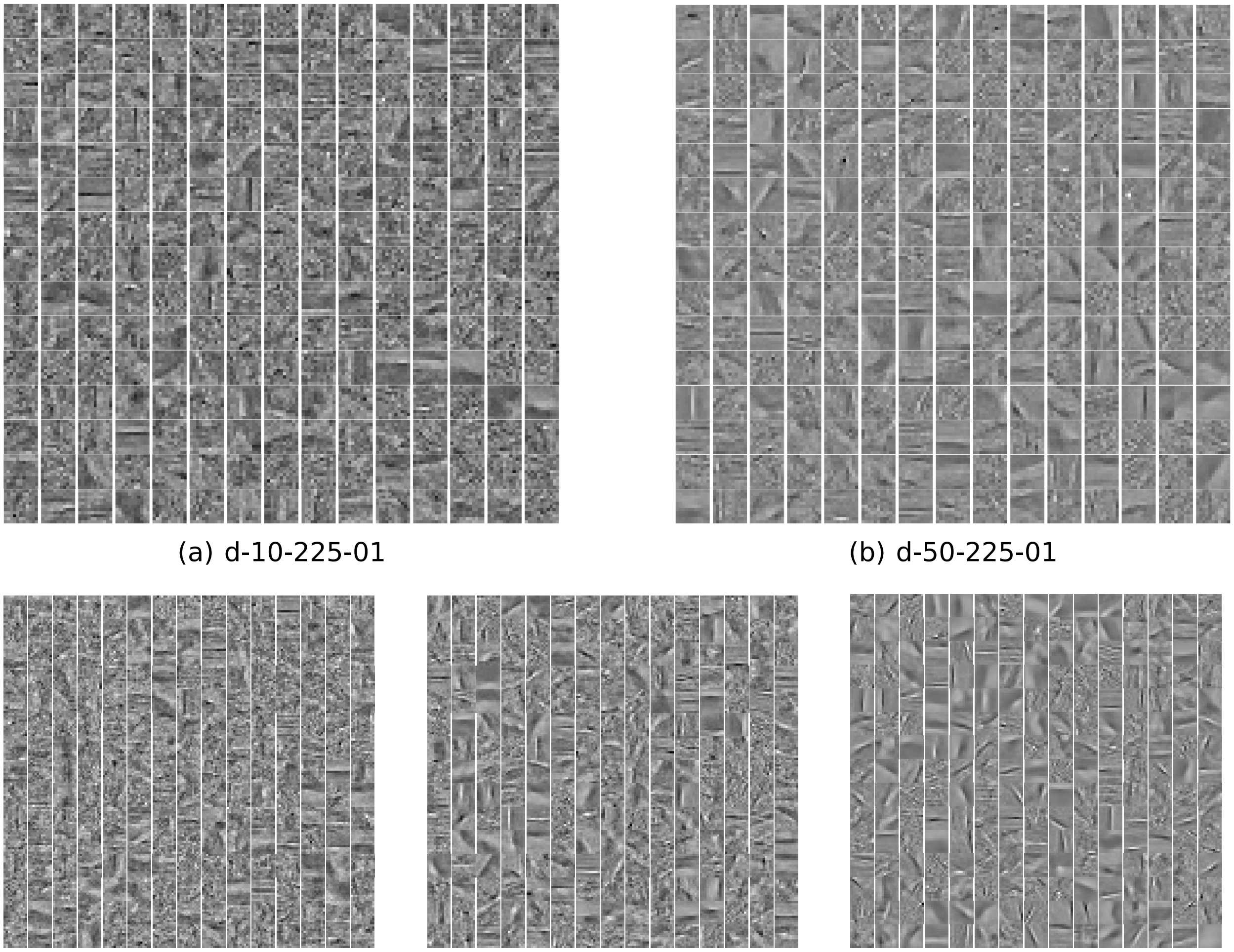

Orig
subsample
Reconstruction
Orig
subsample
Reconstruction
Orig
subsample
Reconstruction

Figure 2:



(a) d-10-225-01 (b) d-50-225-01

(c) d-10-225-005 (d) d-50-225-005 (e) d-270-225-005

Image	1	2	3
d-CSC	24.79	23.78	28.90
d-CCSC	25.41	24.25	29.47
d-10-100-01	25.20	24.21	29.29
d-50-100-01	25.39	24.34	29.50
d-10-225-01	25.37	24.40	29.54
d-50-225-01	25.51	24.52	29.70
d-10-225-005	25.31	24.34	29.47
d-50-225-005	25.50	24.46	29.74
d-270-225-005	**25.80**	**24.72**	**30.03**

Table 1: Quantitative analysis for 2D image inpainting

4 follow-ups

- Test on large datasets (high resolution images and sample size)
- Evaluation metrics (test set objective, reconstruction problems (PSNR,

SSIM))

- Advantages of solving the problem in spatial domain and in stochastic processes
- convergence prove