

Acceleration of the Sparse Modeling Imaging Tool for ALMA Radio Interferometric Data

George Kosugi,¹ Takeshi Nakazato,¹ and Shiro Ikeda²

¹*National Astronomical Observatory of Japan, Osawa 2-21-1, Mitaka, Tokyo 181-8588, Japan; george.kosugi@nao.ac.jp*

²*The Institute of Statistical Mathematics, Midori 10-3, Tachikawa, Tokyo 190-8562, Japan*

Abstract. Sparse modeling is widely used in image processing, signal processing, and machine learning recently. Thanks to the research and progress in statistical mathematics along with the evolution of computational power, the technique is applicable to the radio imaging for the data obtained with the ALMA (Atacama Large Millimeter-submillimeter Array). We've developed a new imaging tool based on the sparse modeling approach and it was experimentally implemented on the Common Astronomy Software Application (CASA) which is an official reduction software for the ALMA data. However, if the image size is large, e.g., 1K x 1K pixels, the data processing time gets longer, say several to ten hours, even with the latest mid-range server computers. Here we present a possible measure to greatly reduce the processing time.

1. Introduction

Radio interferometric imaging using the sparse modeling approach was originally developed for VLBI (Very Long Baseline Interferometry) observation data (Honma et al. 2014). Several simulated data were used to evaluate the method for years (Kuramochi et al. 2018). To automatically determine the most realistic solution from the infinite number of possible solutions, the cross-validation (CV) technique was introduced (Akiyama et al. 2017). We've developed a new interferometric imaging tool Python module for Radio Interferometry Imaging with Sparse Modeling (PRIISM, Nakazato et al. 2019) implemented on CASA, a standard data reduction application for ALMA data.

However, the new imaging technique with sparse modeling is computationally intense even for the latest CPUs. Furthermore, the CV process requires an order of magnitude more calculations. Reduction of its processing time is essential to utilize the technique with real ALMA data.

2. Process Overview

The most statistically realistic image can be derived by solving the following formula with the iterative process.

$$\mathbf{x} = \mathit{argmin}[\|\mathbf{v} - F(\mathbf{x})\|^2 + \lambda_1 \|\mathbf{x}\| + \lambda_{TSV} TSV(\mathbf{x})] \quad (1)$$

subject to $\mathbf{x} \geq 0$, where \mathbf{x} is image, \mathbf{v} is visibility, $F()$ is Fourier transform, $T_{SV}()$ is Total Square Variation function. Two regularization parameters λ_1 and λ_{TSV} control sparseness and smoothness, respectively. The equation can be solved iteratively, and the calculation is terminated when the image is converged. For each λ_1 and λ_{TSV} , we run the CV process to find the best probable $(\lambda_1, \lambda_{TSV})$ combination (Figure 1).

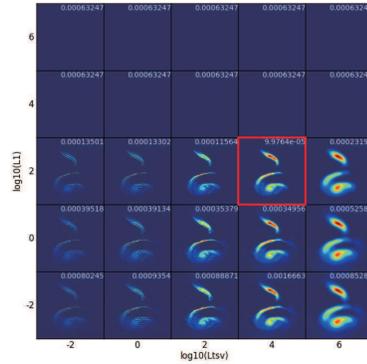


Figure 1. Sample chart resulting from the CV process. In this case, numerical values of 10^{-2} , 10^0 , 10^2 , 10^4 , and 10^6 were assigned to λ_1 and λ_{TSV} , and equation (1) was solved iteratively for every $(\lambda_1, \lambda_{TSV})$ combination. The most probable image can be obtained with $\lambda_1 = 10^2$ and $\lambda_{TSV} = 10^4$ (red square).

3. What's the Cross-Validation (CV) Process?

To choose the best probable regularization parameters λ_1 and λ_{TSV} , the CV process is introduced. In the CV process, visibility data is first divided into N (say 10) groups (N -fold CV), and then the process is run with $N-1$ groups of data (training set) to find a solution. We then apply the solution to the rest of the group (validation set) and calculate the deviation from the fit. The process is repeated with every combination of data groups; that means the process is repeated N times. Finally, we average all the deviation values derived by the process above. The smaller the averaged deviation is, the better the solution is presumed to be.

For every $(\lambda_1, \lambda_{TSV})$ combination, the whole CV process above is applied to determine the best λ_1 and λ_{TSV} , namely choose the combination of having the lowest averaged deviation. $\lambda_1 = 10^2$ and $\lambda_{TSV} = 10^4$ was selected in Figure 1 (in a red square frame). As one can imagine, this is really a CPU intensive process.

4. How to Accelerate the Process

Solving equation (1) is an iterative process. The resulting image is improved and converged gradually. Figure 2 shows how the image gets converged as the iteration increases. If the iteration cycle can be terminated earlier, the shorter the processing time goes. As is seen in Figure 2, the image is converged rapidly even in early cycles, and

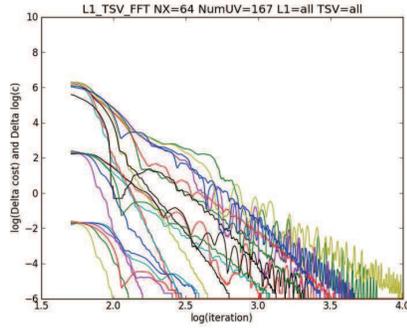


Figure 2. Image convergence curves for every $(\lambda_1, \lambda_{TSV})$ combination. The vertical axis represents the convergence: difference of the cost (in the square bracket of the right side of equation (1)) between adjacent iteration cycles. The lower the difference of the cost is, the more the image is converged. The horizontal axis shows the number of iteration cycles with a logarithmic scale.

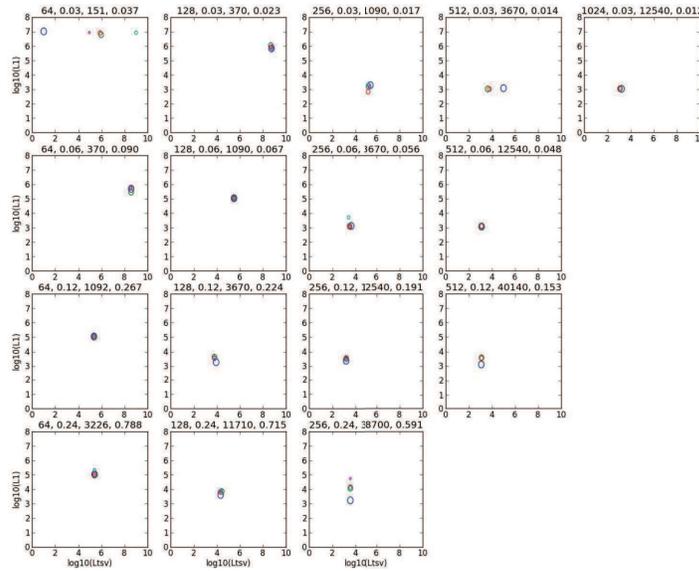


Figure 3. The most probable $(\lambda_1, \lambda_{TSV})$ combination derived from the CV process was plotted by changing the iteration cycle from 100 to 10000. From top to bottom, spatial resolution was changed from 0.03 to 0.24 arcsec/pixel. From right to left, size of the image was changed from 64x64 to 1024x1024. Ovals represent the CV result at a certain number of iteration cycles: largest Oval for 100, next largest oval for 300, middle sized oval for 1000, smaller oval for 3000, and the smallest oval for 10000 iterations. Numbers put on each boxes are image size (pixel), spatial resolution (arcsec/pixel), number of data on the uv-plane, and filling factor of the uv-plane, respectively.

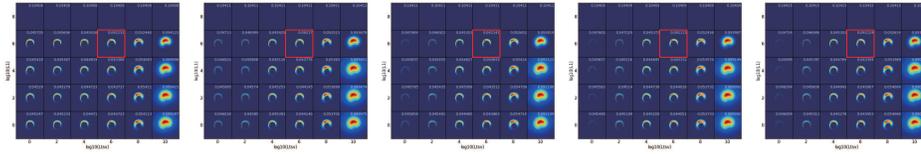


Figure 4. CV process with different iteration cycles: from left to right, 100, 300, 1000, 3000, and 10000 iterations, respectively. The vertical axis and the horizontal axis for each chart indicate λ_1 and λ_{TSV} , respectively.

in some cases it rebounds and oscillates in later cycles. But in general, the curve shows decreasing trend globally.

Figure 3 shows how the derived most probable λ_1 and λ_{TSV} are moved as the number of iteration cycle increases. In most cases, only small changes can be seen even if the iteration cycle is increased, and therefore, 100 iteration cycles is barely acceptable. For more safety, 300 or 1000 iteration cycles is enough for rough estimation of λ_1 and λ_{TSV} in the CV process.

Resulting charts from the CV process with different iteration cycles are shown in Figure 4. Most probable $(\lambda_1, \lambda_{TSV})$ combinations selected by the CV process were independent of the number of iteration cycles and identical in this case. The processing time was 29, 85, 279, 820, and 2740 sec for 100, 300, 1000, 3000, and 10000 iteration cycles, respectively. The iteration process was stopped when the difference of the cost between two adjacent cycles became smaller than a certain threshold value. Since the threshold is arbitrarily set, the value tends to be smaller so as to continue the iteration process until the image is fully converged. It requires a long time. However, if we split the whole process into two, namely, a light weighted CV process (small iteration cycle) only to determine the regularization parameters $(\lambda_1, \lambda_{TSV})$ combination and the final imaging iteration process (until the image is fully converged) with the derived parameter set $(\lambda_1, \lambda_{TSV})$, the whole processing time is greatly reduced. That is one of the practical solutions to accelerate the interferometric imaging process using PRIISM.

References

- Akiyama, K., Ikeda, S., Pleau, M., Fish, V. L., Tazaki, F., Kuramochi, K., Broderick, A. E., Dexter, J., Mościbrodzka, M., Gowanlock, M., Honma, M., & Doeleman, S. S. 2017, *AJ*, 153, 159. 1702.00424
- Honma, M., Akiyama, K., Uemura, M., & Ikeda, S. 2014, *PASJ*, 66, 95
- Kuramochi, K., Akiyama, K., Ikeda, S., Tazaki, F., Fish, V. L., Pu, H.-Y., Asada, K., & Honma, M. 2018, *ApJ*, 858, 56. 1802.05783
- Nakazato, T., Ikeda, S., Akiyama, K., Kosugi, G., Yamaguchi, M., & Honma, M. 2019, in *ADASS XXVIII*, edited by P. J. Teuben, M. W. Pound, B. A. Thomas, & E. M. Warner (San Francisco: ASP), vol. 523 of ASP Conf. Ser., 143