Fast NSR: an optimized Non-linear Stochastic deconvolution for large data sets and clinical analyses

Denis Peruzzo[1], Danilo Benozzo[1], Gianluigi Pillonetto[1], and Alessandra Bertoldo[1]
[1]*Department of Information Engineering, University of Padova, Padova, PD, Italy*

Target audience: Scientists and clinicians with interest in perfusion-MRI.

Purpose: Non-linear Stochastic Regularization (NSR) [1] is a non parametric deconvolution method that allows to estimate reliable Cerebral Blood Flow (CBF) values, physiological residue functions and the dispersion component in data from Dynamic Susceptibility Contrast – Magnetic Resonance Imaging (DSC-MRI). On the other hand, NSR has been shown to be time consuming and sensitive to the starting points, reducing its applicability. Here, we present fast NSR, an optimization of the original NSR algorithm that overcomes the problems of the original implementation.

Methods: NSR models the residue function as the convolution between a deterministic function, modeling the dispersion, and a stochastic function, which uses the exponential of a Brownian motion to describe the non dispersed component of the residue function. Both the deterministic and the stochastic functions are described by hyper-parameters that must be quantified in each voxel via maximum likelihood strategy [1]. Fast NSR introduces a preprocessing step that identifies the optimal starting points for each voxel and fixes the Brownian motion for the voxel level analysis. Firstly, features describing the concentration kinetic are derived for each voxel, i.e. the area under the curve, the maximum concentration, the time to peak, and three slopes describing the peak raise, the peak drop and the recirculation. Secondly, an automatic cluster analysis is performed on those features to divide the image voxels in subsets of voxels with similar kinetics. Thirdly, the mean kinetic is computed for each cluster and analyzed multiple times using the original NSR algorithm and different sets of starting points. Fourthly, the cluster best estimate is selected on the basis of physiological and fit constraints. Finally, the voxel level analysis is performed using the corresponding cluster best estimate to fix the Brownian motion and to set up the starting point.

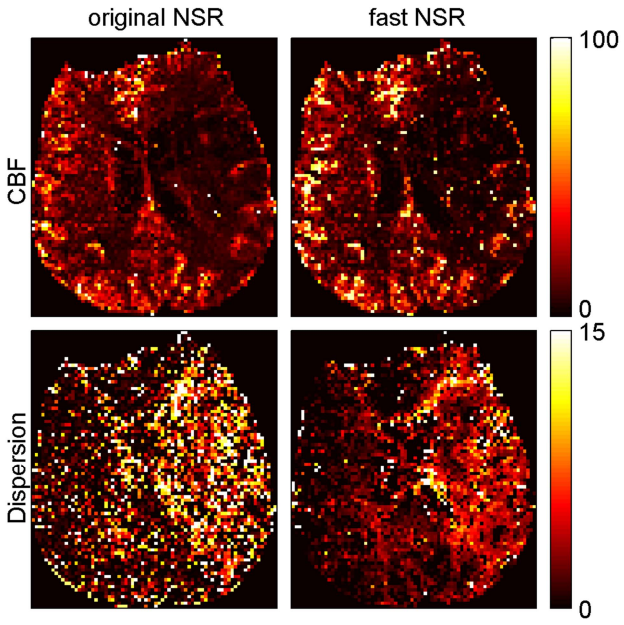

Fig 1: example of the perfusion and dispersion maps obtained in a subject with a severe stenosis on the right

Original NSR and fast NSR are compared on a data set of 11 subjects with severe atherosclerotic unilateral stenosis of the internal carotid artery. DSC-MRI protocol: 1.5T GE scanner, TE=51 ms, TR=1560 ms, voxel size 0.18x0.18x5 mm^3, 128x128x12 voxels per volume, 48 volumes. The Arterial Input Function (AIF) is extracted using the automatic method described in [2].

	Original NSR	fast NSR
Preprocessing	no	cluster analysis
Starting points	single	multiple
Starting point optimization	on the acquisition sequence	on the subject data
Comp. time	~14 days	~2.5 hours
outlier voxels	~20%	~1%

Tab 1: summary of the features of original NSR and fast NSR.

Results: Figure 1 reports the perfusion and dispersion maps obtained in a subject using both NSR implementations. Both methods highlight the perfusion deficit and the large dispersion component in the stenotic hemisphere. Table 1 summarizes the different set-up and the performances of the two NSR implementations. All analyses are performed on a standard personal computer (dual core 2.4 GHz CPU).

Discussion: CBF maps obtained using the two methods are comparable and well highlight the pathologic condition. The dispersion map obtained using fast NSR is smoother and easier to analyze than the original NSR one. On the other hand, optimized starting points allow fast NSR to strongly reduce the computational time and the amount of outlier voxels (i.e. voxels not correctly processed) compared to original NSR.

Conclusion: NSR is now applicable also in large data sets and in clinical contexts where an accurate estimate of CBF and dispersion is required.

References: [1] Zanderigo et al., IEEE Trans Biomed Eng 56:1287-97,(2009); [2] Peruzzo et al. Comput Methods Programs Biomed. 104(3):e148-57 (2011)