Exploring local $f_{\rm NL}$ estimator based on the binned bispectrum

Casaponsa et al. 2013, (arXiv:1305.0671)

Biuse Casaponsa Galí
Observational cosmology and instrumentation group
IFCA, Santander

CSIC

Instituto de Física de Cantabria

UNIVERSIDAD DE CANTABRIA

Outline

Exploring local f_{NL} estimator based on the binned bispectrum
Casaponsa B., Barreiro R.B., Martínez-González E., Curto A., Bridges M., Hobson M.P.

2013, accepted for publication in MNRAS

arxiv:1105.6116

Local non-Gaussianity

Gaussianity

Standard inflationary model $\longmapsto$ Gaussian distribution of the anisotropies

Non-Gaussianity

Any deviation from normal probability distribution. Different processes can show different deviations.

local fnl parameter

$\phi = \phi_L + f_{NL}\left[\phi_L^2 - < \phi_L^2 >\right] \Rightarrow \frac{\Delta T}{T} = F(\phi, f_{NL}).$
Third order moments, as for example the bispectrum, are linearly dependant to fnl.

Local non-Gaussianity

Gaussianity

Standard inflationary model $\mapsto$ Gaussian distribution of the anisotropies

Non-Gaussianity

Any deviation from normal probability distribution. Different processes can show different deviations.

local fnl parameter

$\phi = \phi_L + f_{NL}\left[\phi_L^2 - <\phi_L^2>\right] \Rightarrow \frac{\Delta T}{T} = F(\phi, f_{NL})$.
Third order moments, as for example the bispectrum, are linearly dependant to fnl.

Very weak signal!

Method's efficiency:

- in terms of accuracy
- CPU time

Aim of this work

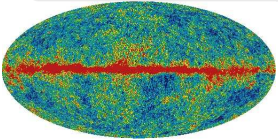
The binned bispectrum presented in Bucher et al. 2010 and Planck collaboration 2013, is one of the most efficient.

- We want to explore the requirements necessary to achieve this. (Might be useful for other bispectrum estimators.)
- Test utility of artificial intelligence techniques for non-Gaussianity analysis (NN).

Optimal Estimator

REAL DATA

f_{NL} estimators get more complicated when including the mask and anisotropic noise.



http://map.gsfc.nasa.gov/

The optimal estimator has been proposed by Babich (2005) and Creminelli et al. (2006) and successfully computed by Smith et al. (2009) and Komatsu et al.(2011) for WMAP-5year and WMAP-7year data. For larger data set as Planck, other approaches have been made.

$$\hat{f}_{NL} = \frac{1}{N}\sum_{l_i m_i} \langle a_{l_1 m_1} a_{l_2 m_2} a_{l_3 m_3}\rangle_1 \times \Big(C^{-1}_{l_1 m_1, l_4 m_4} C^{-1}_{l_2 m_2, l_5 m_5} C^{-1}_{l_3 m_3, l_6 m_6}\ a_{l_4 m_4} a_{l_5 m_5} a_{l_6 m_6} - 3\ C^{-1}_{l_1 m_1, l_2 m_2} C^{-1}_{l_3 m_3, l_4 m_4} a_{l_4 m_4} \Big) ,$$

where

$$N = \sum_{l_i m_i} \langle a_{l_1 m_1} a_{l_2 m_2} a_{l_3 m_3}\rangle_1 C^{-1}_{l_1 m_1, l_4 m_4} C^{-1}_{l_2 m_2, l_5 m_5} C^{-1}_{l_3 m_3, l_6 m_6} \times \langle a_{l_1 m_1} a_{l_2 m_2} a_{l_3 m_3}\rangle_1 , \quad (2)$$

High computational demand.

Binned bispectrum

A reduced bispectrum is defined as:

$$b_{l_1 l_2 l_3} = \int T_{\ell_1} T_{\ell_2} T_{\ell_3} d\Omega \ , \tag{3}$$

where $T_\ell(\vec{n}) = \sum_m a_{\ell m} Y(\vec{n})$.
The binned reduced bispectrum is then

$$b_{abc} = \sum_{\ell_1 \in I_a} \sum_{\ell_2 \in I_b} \sum_{\ell_3 \in I_c} b_{\ell_1 \ell_2 \ell_3}, \tag{4}$$

where I_n are bins in ℓ.
Binned maps can be constructed as: $T_a = \sum_{\ell \in I_a} T_\ell$.

Then b_{abc} is constructed from $T_a T_b T_c$ (number of spherical harmonic transformation drmatically reduced!).

$$b_{abc} = \sum_i \frac{4\pi}{N} T_a T_b T_c(i) \tag{5}$$

(Bucher et al. 2010, MNRAS, 407, 2193)

$f_{\rm NL}$ estimators

Approximated maximum-likelihood estimator (AMLE)

$$f_{\rm NL} = \sum_{abc,def} \frac{\langle b_{abc}\rangle^1 C^{-1}_{abc,def} b^{obs}_{def}}{\sum_{abc,def} \langle b_{abc}\rangle^1 C^{-1}_{abc,def} \langle b_{def}\rangle^1} \,. \tag{6}$$

Approximated maximum-likelihood estimator with diagonal covariance matrix (AMLED)

$$f_{\rm NL} = \sum_{abc} \frac{\langle b_{abc}\rangle^1 b^{obs}_{abc}/\mathrm{var}(b_{abc})}{\sum_{def} (\langle b_{def}\rangle^1)^2/\mathrm{var}(b_{def})} \tag{7}$$

Neural network estimator (NNE)

$$f_{\rm NL} = \sum_{abc} w_{abc} b_{abc} + \theta \,. \tag{8}$$

Those estimators are compared using WMAP-7yr data and realisations with WMAP-7yr characteristics.

f_{NL} estimators

Approximated maximum-likelihood estimator (AMLE)

$$f_{\mathrm{NL}} = \sum_{abc,def} \frac{\langle b_{def}\rangle^1 C^{-1}_{def,abc} b^{obs}_{abc}}{\sum_{abc,def} \langle b_{abc}\rangle^1 C^{-1}_{abc,def} \langle b_{def}\rangle^1} \,. \tag{9}$$

Approximated maximum-likelihood estimator with diagonal covariance matrix (AMLED)

$$f_{\mathrm{NL}} = \sum_{abc} \frac{\langle b_{abc}\rangle^1 b^{obs}_{abc}/\mathrm{var}(b_{abc})}{\sum_{def} (\langle b_{def}\rangle^1)^2/\mathrm{var}(b_{def})} \tag{10}$$

Neural network estimator (NNE)

$$f_{\mathrm{NL}} = \sum_{abc} \mathbf{w_{abc}} b^{obs}_{abc} + \theta \,. \tag{11}$$

Realistic case

However, when dealing with real data those estimators are suboptimal due to the anisotropies given by the mask and the noise. There are several things that one might try to solve it:

- **Add a linear term**.
  remembering that $b_{abc} = \sum_i \frac{4\pi T(i)_a T(i)_b T(i)_c}{N_{pix}}$ a linear term of the form
  $$b^L_{abc} = b_{abc} - \sum_i \frac{4\pi \langle T_a T_b \rangle T_c - \langle T_a T_c \rangle T_b - \langle T_b T_c \rangle T_a}{N_{pix}} \tag{12}$$
  can be subtracted
- **Mean subtraction**.
  In Curto et al. 2009,2010,2011 and Donzelli et al. 2012 the mean value of the wavelet and needlet coefficients is subtracted. Then:
  $$b^{MS}_{abc} = \sum_i \frac{4\pi}{N} (T_a - \bar{T}_a)(T_b - \bar{T}_b)(T_c - \bar{T}_c) \tag{13}$$
- **Inpainting**.
  Fill the masked region by some simulated signal, specially the point sources and the galactic mask edges.
  $$b^I_{abc} = b_{abc}(from\ inpainted\ maps) \tag{14}$$

Those effects will be tested and all possible combinations.

Neural networks

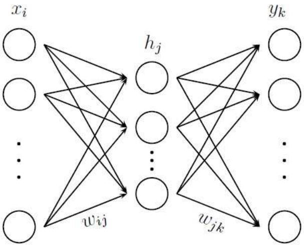


Figure 1. Schematic of a 3-layer feed-forward neural network.

nodes

$$y_k = \sum_j w_{kj} h_j + \theta_k ,$$

where h_j is

$$h_j = tanh(\sum_i w_{ji} x_i) + \theta_j$$

$$y_k = \sum_j w_{kj} \left(\tanh(\sum_i w_{ji} x_i) + \theta_j \right) + \theta_k$$

Neural networks

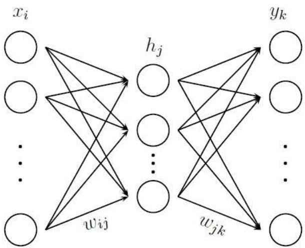


Figure 1. Schematic of a 3-layer feed-forward neural network.

nodes

$$y_k = \sum_j w_{kj} h_j + \theta_k ,$$

where h_j is

$$h_j = tanh(\sum_i w_{ji} x_i) + \theta_j$$

$$y_k = \sum_j w_{kj} \left(\tanh(\sum_i w_{ji} x_i) + \theta_j \right) + \theta_k$$

NN training

Supervised training for a feedforward network

We train the network with a known set of inputs and outputs, $\mathbf{x^t}$ and $\mathbf{y^t}$. We choose an optimization function (Ex. mse,rmse,χ^2,. . .) . The optimization function is only dependent of the network parameters.

$$Err = \sum_{t,k} (y_k^{(net),t} - y_k^{(t)})^2$$

minimize this function (using conjugates gradient methods, gradient descent method, etc.)

We have used a neural network code with $Q = \alpha S - \chi^2$, where S is the entropy (Gull & Skilling 1999). Following the maximum entropy trajectory to find the optimal solution. In any case we need to find w_{lm} and $\theta_n \mapsto y_k \sim y_k^{real}$.

Then simulations with a given $f_{\rm NL}$ are generated and the binned bispectrum components are computed.

- INPUTS $\mapsto b_{abc}$
- OUTPUTS $\mapsto f_{\rm NL}$
- TRAINING $\mapsto w_{abc}, \theta$

$$f_{\rm NL} = \sum_{abc} w_{abc} b_{abc} + \theta \ . \quad (15)$$

Lower bound

Our efficiency goal for a realistic case is:

$$\sigma^2_{fh} = \left[f_{sky} \sum_{\ell_1 \leq \ell_2 \leq \ell_3} \frac{\left(\langle B_{\ell_1 \ell_2 \ell_3} \rangle^1 \right)^2}{\Delta C_{\ell_1} C_{\ell_2} C_{\ell_3}} \right]^{-1} \tag{16}$$

where Δ takes values 1, 2 or 6 when all ℓ's are different, two are equal, or all are the same and f_{sky} is the fraction of the sky available. (Noise and beam contributions are added).

$$\boxed{\sigma_{fh} \sim 21}$$

Approximated maximum-likelihood estimator (AMLE)

$$f_{\rm NL} = \sum_{abc,def} \frac{\langle b_{def} \rangle^1 C^{-1}_{def,abc} b^{obs}_{abc}}{\sum_{abc,def} \langle b_{abc} \rangle^1 C^{-1}_{abc,def} \langle b_{def} \rangle^1} \,. \tag{17}$$

Approximated maximum-likelihood estimator with diagonal covariance matrix (AMLED)

$$f_{\rm NL} = \sum_{abc} \frac{\langle b_{abc} \rangle^1 b^{obs}_{abc} / \mathrm{var}(b_{abc})}{\sum_{def} (\langle b_{def} \rangle^1)^2 / \mathrm{var}(b_{def})} \tag{18}$$

Neural network estimator (NNE)

$$f_{\rm NL} = \sum_{abc} w_{abc} b^{obs}_{abc} + \theta \,. \tag{19}$$

Linear Term

Linear term.

remembering that $b_{abc} = \sum_i \frac{4\pi T(i)_a T(i)_b T(i)_c}{N_{pix}}$ a linear term of the form

$$b^L_{abc} = b_{abc} - \sum_i \frac{4\pi \langle T_a T_b \rangle T_c - \langle T_a T_c \rangle T_b - \langle T_b T_c \rangle T_a}{N_{pix}} \tag{20}$$

can be subtracted

Case	INP	LT	MS	Estimator	σ_g	$< f_{\rm NL} >^{Gauss}$	$(\sigma_{fh} - \sigma_g)/\sigma_{fh}(\%)$
1	No	Yes	No	AMLED	35.9	-0.3	60
				AMLE	24.3	0.1	9.3
				NN	23.6	0.6	4.8

Mean Subtraction

Mean subtraction.
Donzelli et al. 2012 have shown that for needlets and wavelets, subtracting the mean value of the wavelet or needlet coefficients is almost equivalent than adding the linear term (and that would be faster to compute). Then the $f_{\rm NL}$ estimators are now computed using:

$$b^{MS}_{abc} = \sum_i \frac{4\pi}{N}(T_a - \bar{T_a})(T_b - \bar{T_b})(T_c - \bar{T_c}) \tag{21}$$

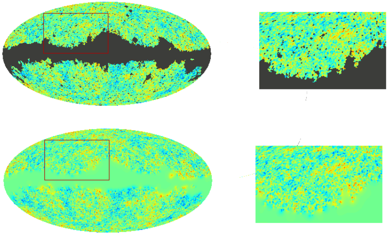
similarities with the linear term:

$$\begin{aligned}\frac{1}{N}\sum_i (T_{a,i} - \bar{T_a})(T_{b,i} - \bar{T_b})(T_{c,i} - \bar{T_c}) &= \frac{1}{N}\sum_i T_{a,i}T_{b,i}T_{c,i} \\ -\bar{T_a}\frac{1}{N}\sum_i T_{b,i}T_{c,i} - \bar{T_b}\frac{1}{N}\sum_i T_{a,i}T_{c,i} &- \bar{T_c}\frac{1}{N}\sum_i T_{a,i}T_{b_i} \\ &+2\bar{T_a}\bar{T_b}\bar{T_c}\end{aligned} \tag{22}$$

Case	INP	LT	MS.	Estimator	σ_g	$< f_{\rm NL} >^{Gauss}$	$(\sigma_{fh} - \sigma_g)/\sigma_{fh}(\%)$
2	No	No	Yes	AMLED	37.0	1.5	64
				AMLE	24.6	-0.4	8.0
				NN	23.6	0.4	4.8

Inpainting

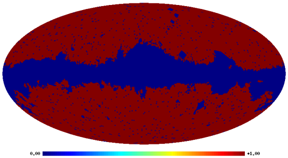
One begins with the map $T(\vec{x})$ and the binary mask $M(\vec{x})$. Then each pixel of the masked map $T' = T \times M$ with value zero is substituted by the average of its immediate neighbours, whether masked or not, using the HEALPIX subroutine *neighbours*.

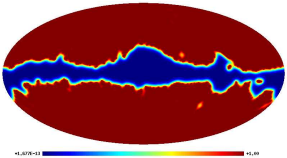


Case	INP	LT	MS	Estimator	σ_g	$< f_{\rm NL} >^{Gauss}$	$(\sigma_{fh} - \sigma_g)/\sigma_{fh}(\%)$
3	Yes	No	No	AMLED	107	3	300
				AMLE	32.7	-1	45
				NN	29.7	-0.3	32

Inpainting

One begins with the map $T(\vec{x})$ and the binary mask $M(\vec{x})$. Then each pixel of the masked map $T' = T \times M$ with value zero is substituted by the average of its immediate neighbours, whether masked or not, using the HEALPIX subroutine *neighbours*.





Case	INP	LT	MS	Estimator	σ_g	$< f_{\rm NL} >^{Gauss}$	$(\sigma_{fh} - \sigma_g)/\sigma_{fh}(\%)$
3	Yes	No	No	AMLED	107	3	300
				AMLE	32.7	-1	45
				NN	29.7	-0.3	32

Linear Term + Inpainting

To improve the results we now use inpainted maps and subtract the linear term. And the same for the mean subtraction.

Case	INP	LT	MS	Estimator	σ_g	$< f_{\rm NL} >^{Gauss}$	$(\sigma_{fh} - \sigma_g)/\sigma_{fh}(\%)$
4	Yes	Yes	No	AMLED	**22.7**	0.7	0.9
				AMLE	**23.3**	0.7	3.5
				NN	**22.4**	0.7	0.4
5	Yes	No	Yes	AMLED	31.5	0.7	40
				AMLE	24.0	0.7	6.7
				NN	23.1	0.5	2.7

Looks better but still not 21! (lower bound, Komatsu et al. 2011)

Final results

With accurate realisations (Elsner and Wandelt 2009 http://planck.mpa-garching.mpg.de/cmb/fnl-simulations/) the lower bound is reached.
(Linear term subtracted and inpainting done to the maps).

Estimator	σ_{fh}	σ_g	$< f_{\rm NL} >^{Gauss}$	$f_{\rm NL}^{map}$	$\Delta f_{\rm NL}$
AMLED	21.3	21.7	-0.2	33.4	3±2
AMLE		22.4	-0.1	39.8	3±2
NN		21.4	0.5	44.2	4±2

Table: Results for inpainted Gaussian realizations. Model estimated and neural network trained using Elsner & Wandelt simulations (set 2). The columns from left to right are: the estimator used, the Fisher σ computed analytically, the dispersion and mean value of $\hat{f}_{\rm NL}$ for 1,000 Gaussian simulations. Followed by the $f_{\rm NL}$ value found for WMAP-7yr data and the contribution expected by the unresolved point sources ($\Delta f_{\rm NL}$).

Three estimators comparison

The computational demand of the estimators is directly related to the number of realisations used.

- AMLE $\mapsto$ realisations to estimate covariance matrix.
- AMLED $\mapsto$ realisations to estimate $\mathrm{var}(b_{abc})$.
- NNE $\mapsto$ realisations required to train the network.

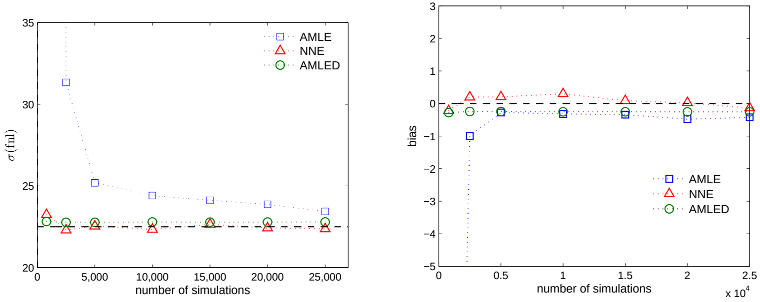


Figure: Comparison of the efficiency (top) and bias (bottom) of the three estimators with respect to the number of simulations used to construct the estimator. For reference, the optimal values for the dispersion and bias (dashed black line) are also shown.

Conclusions

- All three estimators are close to optimal if linear term is subtracted and inpainting performed.
- The effect of the linear term is small if the full covariance matrix is taken into account or NN are used.
- Mean subtraction almost equivalent to the linear term subtraction only for AMLE and NN.
- The effect of the inpainting is small if the full covariance matrix is taken into account or NN are used.
- The NN estimator simplifies the analysis with respect to the AMLE.
- AMLED is very effective and fast when these considerations are taken into account.
- Neural networks can be useful to avoid large complicated steps (matrix inversions, model estimation).

Thanks